applied sciences

IMPACT FACTOR 2.5

CITESCORE 5.5

Article

Data-Driven Approaches for Predicting and Forecasting Air Quality in Urban Areas

Cosmina-Mihaela Rosca, Madalina Carbureanu and Adrian Stancu

Special Issue

Smart City and Informatization, 2nd Edition

Edited by

Prof. Dr. António Pereira, Dr. Nuno Alexandre Ribeiro Costa and Prof. Dr. Antonio Fernández-Caballero

MDPI

https://doi.org/10.3390/app15084390

applied sciences

MDPI

Article

Data-Driven Approaches for Predicting and Forecasting Air Quality in Urban Areas

Cosmina-Mihaela Rosca [1], Madalina Carbureanu [1] and Adrian Stancu [2,*]

[1] Department of Automatic Control, Computers, and Electronics, Faculty of Mechanical and Electrical Engineering, Petroleum-Gas University of Ploiesti, 39 Bucharest Avenue, 100680 Ploiesti, Romania; cosmina.rosca@upg-ploiesti.ro (C.-M.R.); mcarbureanu@upg-ploiesti.ro (M.C.)

[2] Department of Business Administration, Faculty of Economic Sciences, Petroleum-Gas University of Ploiesti, 39 Bucharest Avenue, 100680 Ploiesti, Romania

* Correspondence: astancu@upg-ploiesti.ro

Abstract: Air quality (AQ) is one of the most important urban environment indicators for the quality of life. The paper proposes a software solution for predicting and forecasting the air quality index (AQI) in urban areas. The study integrates pollutant factors (CO, NO_2, SO_2, $PM_{2.5}$), meteorological parameters (temperature, humidity, wind speed), and traffic data to determine air quality. For this purpose, 19 predictive models were developed and compared: 12 machine learning algorithms, 7 deep learning, and 1 forecasting model based on structural component analysis. The Random Forest Regression model, customized within the study, achieved the best results, with an R^2 score of 99.59%, an MAE of 0.22%, an MAPE of 0.68%, and an OP (Overall Precision) score of 95.61%. It was subsequently validated on unseen data and recorded a mean deviation of 0.58%. For short-term AQI forecasting (5 days), the AQIF model achieved an R^2 of 71.62%, an MAE of 0.4%, and an MAPE of 0.9%. The proposed solution was integrated into a web application with IoT infrastructure and real-time alert mechanisms. Future directions include expanding the dataset and optimizing hyperparameters for the deep learning models to increase accuracy, as well as integrating PM_{10} and O_3 factors, along with the degree of industrialization and demographic level.

Keywords: AI methods; machine learning; IoT infrastructure; deep learning; urban area; air quality; pollution control; Azure Communication Service; Azure SQL Server

Academic Editors: Antonio Fernández-Caballero, Nuno Alexandre Ribeiro Costa and António Pereira

Received: 26 March 2025
Revised: 11 April 2025
Accepted: 15 April 2025
Published: 16 April 2025

Citation: Rosca, C.-M.; Carbureanu, M.; Stancu, A. Data-Driven Approaches for Predicting and Forecasting Air Quality in Urban Areas. *Appl. Sci.* **2025**, *15*, 4390. https://doi.org/10.3390/app15084390

1. Introduction

The principal parameters monitored as pollutants are carbon monoxide (CO), nitrogen dioxide (NO_2), particulate matter (PM) with diameters of 2.5 μm and smaller ($PM_{2.5}$), and sulfur dioxide (SO_2). These pollutants originate from distinct sources, have specific health impacts, and interact with environmental factors that affect human health and ecosystems.

CO primarily arises from incomplete combustion processes, especially those related to vehicular traffic and industrial activities. Munfarida and Arida [1] demonstrate a direct correlation between the number of vehicles and increased CO levels. Chronic exposure to CO is associated with long-term health conditions, particularly in communities with limited access to healthcare services [2]. These findings highlight the importance of traffic management in urban planning [3,4].

NO_2 is another major urban pollutant, primarily released through vehicle emissions and industrial processes. It is a known cause of respiratory diseases and contributes to the formation of ground-level ozone (O_3), which presents additional health risks [5]. The study

by Zhang et al. [6] suggests that NO_2 exposure is particularly harmful in urban areas with high traffic emissions.

$PM_{2.5}$ consists of fine particles capable of penetrating deep into the lungs and entering the bloodstream, resulting in severe health consequences such as respiratory and cardiovascular diseases [7,8]. Annually, many urban areas frequently surpass this 10 $\mu g/m^3$ threshold [7]. The link between $PM_{2.5}$ exposure and health outcomes is well established, with studies demonstrating that higher $PM_{2.5}$ levels correlate with increased mortality and hospitalizations for respiratory ailments [9,10].

SO_2 contributes to respiratory complications and exacerbates pre-existing conditions like asthma [11]. While SO_2 concentrations have decreased in several regions due to stricter environmental regulations, its health impacts persist, particularly in areas with intensive industrial activity. The interaction of SO_2 with other pollutants, such as NO_2 and $PM_{2.5}$, complicates the evaluation of AQ, as these substances often coexist and may synergistically affect health [12,13].

The COVID-19 pandemic offered a unique chance to witness how decreased human activity impacted air pollution [14]. Global lockdowns led to notable decreases in CO, NO_2, SO_2, and $PM_{2.5}$ levels across various regions, underscoring the influence of human activities on AQ [15,16]. The long-term consequences of these reductions on public health and AQ management remain unclear. Sarmadi et al. [17] analyzed 87 cities during the pandemic. The study found that AQI-$PM_{2.5}$, AQI-PM_{10}, and AQI-NO_2 decreased by −7.36%, −17.52%, and −20.54% in 2020 compared to 2019, following the introduction of restriction measures. This trend reversed in 2021, when the values of these parameters increased by +4.25%, +9.08%, and +7.48%. Another study conducted in Monterrey shows a decrease in the concentrations of SO_2 (−41.9%), PM_{10} (−30.5%), and $PM_{2.5}$ (−25.6%) during the quarantine period, compared to the average of the years 2017–2019. O_3 increased by +15% during quarantine and +2.2% during the reopening period [18]. Another study on 21 cities highlighted that daily NO_2 concentrations decreased by between 3% and 58% during the quarantine, but increased subsequently after the lifting of restrictions [19]. The pandemic highlighted the importance of sustainable urban planning and pollution control initiatives to reduce the health impacts of air pollution in the future [20,21].

1.1. AI Used in Air Quality Measurement

Measures to combat pollution have increasingly incorporated artificial intelligence (AI), smart devices, and the Internet of Things (IoT) to enhance monitoring, prediction, and control of environmental pollutants, considering compatibility concepts between technologies [22]. In recent years, data-driven AQ models have garnered attention due to their capacity to utilize large datasets and advanced computational methods for predicting pollution levels. These models harness concepts such as machine learning (ML) and more restrictive deep learning (DL) techniques to increase prediction and forecasting accuracy [23].

The data-driven AQ modeling methodology includes applying ML and DL techniques. For example, Long Short-Term Memory (LSTM) networks, a recurrent neural network component (RNN), are used for forecasting time-series of AQ parameters. The LSTM networks capture temporal dependencies in AQ data, resulting in better predictive performance than traditional statistical methods [24,25]. The hybrid models produce promising results by incorporating multiple data sources, including meteorological and geographical information, to enhance AQ forecasts [26].

Murad et al. [27] present the uncertainty quantification in AQ forecasting through the probabilistic DL methods, such as those utilizing Gaussian processes and quantile regression decision-making in environmental management.

Multiple datasets, including Automatic Identification System (AIS) data for maritime emissions, were studied in refining AQ models. Incorporating AIS data allows researchers to estimate emissions from shipping activities and their impact on local AQ, offering a holistic perspective on pollution sources [28]. Many geospatial datasets use regression models to enhance air pollution mapping across urban environments, enabling targeted intervention strategies [29].

Recent innovations in model architectures, such as diffusion convolutional recurrent neural networks, have shown superior performance in regional predictions of pollutants like O_3 and PM [30]. These models leverage graph-based techniques to capture spatial relationships between monitoring stations.

ML has also emerged as a powerful tool for AQ prediction [31]. It leverages complex datasets to enhance predictive accuracy beyond traditional statistical methods. One notable approach is ensemble methods, such as Random Forests, which have demonstrated improved prediction accuracy by incorporating atmospheric circulation patterns and key meteorological factors like wind speed, temperature, and humidity [32,33]. For example, Sekula et al. [32] mention that integrating a vertical gradient between neighboring pressure fields enhances PM_{10} forecasts and that atmospheric dynamics influence AQ predictions. In the paper by Hamami and Dahlan [34], a LightGBM model was developed that achieved a root mean square error (RMSE) of approximately 1.0 for $PM_{2.5}$ concentration forecasts, showcasing the effectiveness of tree-based models in this context.

Badrakh and Choimaa [35] applied the LSTM networks to forecast air pollution parameters, utilizing historical weather data to enhance prediction accuracy. There is growing interest in hybrid DL frameworks, combining multiple neural network architectures to improve performance in AQ forecasting tasks [36]. The advancements in DL have led to the introduction of a Diffusion Convolutional Recurrent Neural Network, which models the spatial and temporal dependencies of air pollutants [30]. This model outperformed traditional approaches by utilizing graph neural networks to represent the complex relationships present in environmental data. Huang and Kuo [37] show the advantages of combining convolutional neural networks (CNNs) with LSTMs for PM forecasting. Liu et al. [38] highlight the importance of integrating meteorological factors with real-time industrial emissions data to improve AQ forecasts' accuracy. This view is supported by Akdi et al. [39], who applied recursive neural networks to model $PM_{2.5}$ emissions, demonstrating the possibility of integrating the DL models to capture the intricate dynamics of AQ variations [40].

Traditional empirical models frequently struggle with accuracy because they cannot adjust to fluctuating environmental conditions. Conversely, ML methods analyze intricate, non-linear interactions among the variables affecting air quality [41].

Feature selection techniques in ML models, a fundamental step in model development, have been shown to optimize performance by identifying the most relevant predictors of AQ. Fang et al. [42] introduced a regional feature selection-based ML system that utilizes AQ observations to improve forecasting accuracy across China.

1.2. IoT Employed in Air Quality Measurement

Multiple studies have demonstrated that ML algorithms successfully predict air quality index (*AQI*) and pollutant concentrations utilizing a range of meteorological and environmental variables.

Integrating IoT technologies into AQ monitoring systems has transformed data collection, analysis, and dissemination. IoT-based systems enable continuous, real-time monitoring of air pollutants, which permits timely responses to pollution incidents and informs public health policies in real-time. Some platforms [43] can be accessed online

and offer daily forecasting information regarding the weather parameters for the next 7 days. By accessing the URL's parameters platform [43], it can be identified that the $PM_{2.5}$ concentration is currently 8.5 $\mu g/m^3$ (10 April 2025).

IoT devices are integrated into AQ monitoring systems due to their capacity for remote operation without human intervention [44]. These devices automatically collect and analyze data, replacing the fixed stations that are costly and limited in spatial coverage [45,46]. For example, low-cost sensors can be deployed in diverse environments, facilitating a more comprehensive understanding of pollution distribution across various regions [47,48]. This is especially important in indoor AQ, where people spend a significant portion of their time [49–51]. Modern sensors measure a wide range of pollutants, including PM, NO_2, and volatile organic compounds (VOCs) [52]. These sensors provide real-time feedback on AQ fluctuations, enabling citizens to participate in measurement initiatives and fostering greater community awareness of air pollution [53–55]. Integrating ML algorithms with IoT systems enhances data analysis capabilities, allowing for predictive AQ modeling based on historical and real-time data [56].

An air quality monitoring system (AQMS) using Global System for Mobile Communications (GSM) communication is presented, enabling real-time measurement of CO_2, CO, NO_2, and SO_2 via electrochemical and infrared sensors. The system features wireless data transmission, a user-friendly Graphical user interface (GUI) for gas concentration visualization, and periodic calibration to ensure accuracy, addressing design challenges from prior studies [57]. A heuristic recurrent air quality predictor (RAQP) is proposed by [58], using meteorology- and pollution-related variables to infer $PM_{2.5}$ concentrations. By addressing non-linear declines in correlations over time, RAQP recursively applies a one-hour prediction model, outperforming state-of-the-art accuracy methods for multi-hour forecasts. Massive-scale AQ sensing with millions of sensors offers fine spatial-temporal monitoring. Challenges include calibration and deployment feasibility, explored via a Helsinki testbed [59].

Another technical solution in AQ is using mobile sensing networks, such as sensors attached to vehicles or drones [60,61]. Data collected through these networks are uploaded to cloud platforms, ensuring accessibility for the public and policymakers [62,63].

1.3. Paper Contributions

The research introduces this unique combination of features into the specialized literature for AQ studies as a novel element. This statement is based on a search in the Web of Science platform that identifies all works containing the terms related to the features proposed in this research TS = ((“air quality” OR “AQ” OR “AQI”) AND “CO” AND “NO_2” AND “SO_2” AND “$PM_{2.5}$” AND “temperature” AND “humidity” AND “wind speed” AND “traffic” AND (“machine learning” OR “ML”)). The query returned zero results in the specialized literature, meaning no other research has addressed the issue of *AQI* using the same input elements for an ML model.

The contributions of the study are mentioned below:

- Implementation and integration of six customized ML algorithms, six pre-implemented ML models using ML.NET, and seven customized DL algorithms designed to address the specific characteristics of AQ data;
- Comparative analysis of ML versus DL methodologies, including the identification of the best approach based on performance metrics using an innovative parameter proposed by the authors;
- Integration of a second model for window timeframe forecasting integrated into a user-friendly application, including a prediction model and a forecasting model;

- Proposing a decision-making framework based on the implemented models, using the Azure Communication Service for data-driven actions in AQ management.

The paper consists of five sections. Section 2 describes importing data for the analyzed area (Bronx, New York) utilized in developing the AQ monitoring model. This section also explains the implementation and integration of the customized ML and DL algorithms explicitly tailored for AQ predictions, including a novel framework for real-time monitoring based on the customized models and relevant meteorological data. A case study is presented to detail the application of the proposed models in predicting AQ parameters under various environmental conditions. A comprehensive description of the real-time AQ monitoring prototype, which utilizes an IoT infrastructure to enhance community engagement, is provided in this section. This section also proposes an AI-based decision-making framework for AQ management. Section 3 presents the testing phase of the proposed AQ model and integrates the results into a new real-time monitoring application. The section includes examples of the proposed methodology for evaluating AQ predictions. Section 4 outlines the discussions, limitations of the study, and possible directions for future research in AQ monitoring and predictive analytics. Section 5 provides the conclusions of the paper.

2. Materials and Methods

According to the Office of the New York State Comptroller [64], between 1980 and 2017, the Bronx's population increased by 26%, reaching nearly 1.5 million people. Public and private investments supported this growth, which fueled the creation of new businesses and thousands of jobs. For ten years ending in 2017, the Bronx added 40,900 private sector jobs and was the only New York City borough that did not lose jobs during the Great Recession. The Bronx has set employment records for ten consecutive years [64]. An extended transportation network, including seven significant highways and seven subway lines, has further bolstered the Bronx's economy. New ferry services and Metro-North rail stations are also being developed to improve connectivity with Manhattan. This industrial growth and increased mobility have contributed to the Bronx's economic resilience, encouraging further research into the area's development. The borough saw a 24% drop in businesses between 1975 and 1984, but since then, the number has risen by 35%, reaching almost 18,000 enterprises by 2017 [64]. This increasing population, industrialization, and transportation improvements justify research interest in the Bronx's AQ due to industrialization and traffic patterns.

In this study, we use data from the United States Environmental Protection Agency (EPA) through its comprehensive outdoor AQ database, which offers daily measurements of various air pollutants across different locations in the United States. Daily statistics related to traffic levels and meteorological conditions in Bronx, New York, were imported from the Open NY (the New York State's open data platform [65]) and Open Meteo API Portal (OMAP) [66]. These data sources were employed to establish key variables influencing *AQI* predictions, helping to develop the model by integrating traffic volumes, weather patterns, CO, NO_2, SO_2, and $PM_{2.5}$. The units of measurement for AQ and weather factors are as follows: temperature in °C, CO is measured in parts per million (ppm), NO_2, SO_2, and $PM_{2.5}$ are measured in micrograms per cubic meter ($\mu g/m^3$), wind speed in kilometers per hour (km/h), humidity in percentage (%), and vehicular traffic in number of cars. The data correspond to the period from 1 January 2020 to 31 December 2023, representing a large volume of data that allows us to obtain a model training with a high accuracy in values prediction.

2.1. Data Import Process Description

These datasets were integrated using API requests to the EPA's online database [67], which provided individual CSV files for each pollutant parameter. We employed a C# script to align and correlate the data from these CSV files, facilitating their import into a centralized database.

The database was an SQL Server hosted locally for local training of the custom-made algorithm. An entry was created for each day of 2020–2023 to populate the database. The import process involved reading data from a CSV file through an API request containing daily records for the selected AQ parameters. Four API requests were made for the complete CO, NO_2, SO_2, and $PM_{2.5}$ data import.

In the same database, records were extracted from a CSV file provided by Open NY, containing data for the same period detailing the number of vehicles that passed through two access points in Bronx County. Additionally, temperature, humidity, and wind speed data for the same dates were imported from the OMAP system.

The import procedure is as follows:

- The table AirQualityDataBronx is populated with values corresponding to the data from 1 January 2020 to 31 December 2023, and NULL values will be used for each column;
- For each analyzed parameter, an API request to the EPA is made:
    - Using the CsvHelper library, the C# script first reads the CSV file containing the AQ value for the first parameter analyzed;
    - Each record from the CSV is parsed and transformed into an AirQualityRecord object, capturing the values for attributes such as the date, parameter type, and the corresponding *AQI* value for the parameter;
    - The script establishes a connection to the SQL database;
    - An SQL UPDATE command is run to modify the existing records in the AirQualityData table based on the new date;
    - The global *AQI* is computed as the maximum value for the AQ_CO, AQ_NO_2, AQ_SO_2, AQ_$PM_{2.5}$;
- A second API Request to Open NY is made to obtain a traffic dataset in the Bronx, representing the total number of cars transiting the county. To calculate the unit traffic data, the hourly traffic volume for each monitoring station in the Bronx was analyzed. At the source [63], the available dataset provided information on the number of vehicles passing through each station (broken down by inbound and outbound directions) at each hour of the day. The monitored locations are not explicitly mentioned, but the number of vehicles provided the information to compute the daily traffic volume for each station. The data are imported into the DailyTrafficData table.
- A third Request to the OMAP is made to retrieve the temperature, humidity, and wind speed values for each day in the 2020–2023 interval. The data are imported into the BronxWeatherDataHistory table;
- The data from the tables AirQualityDataBronx, DailyTrafficData, and BronxWeatherDataHistory are imported into a single table named BronxStats;
- The records for which one of the monitored parameters had a NULL value were at random, and we removed them due to missing data in the CSV files for several days, ensuring data integrity for analysis.

Figure 1 presents a map of the stations' locations for data collection, including the contaminant measurement station and the meteorological measurement station.

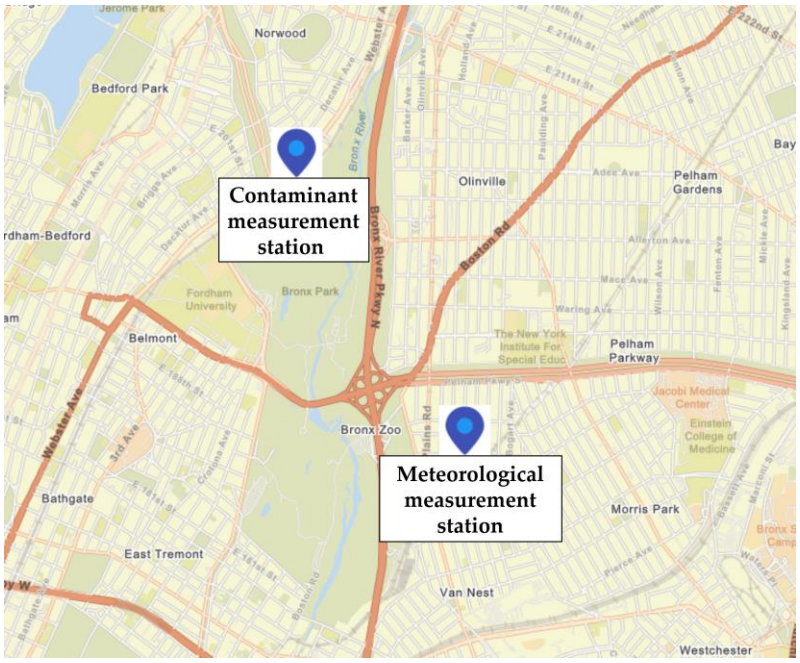

Figure 1. A map screenshot containing the contaminants and meteorological station location [68].

Each pollutant has a formula that converts its concentration into an *AQI* value. According to the referenced paper [69], the general form of the *AQI* is expressed as mentioned in Equation (1):

$$AQI = I_0 + (I_1 - I_0) \times \frac{(C - C_0)}{(C_1 - C_0)} \quad (1)$$

where

AQI is the air quality index;

C—the pollutant concentration ($\mu g/m^3$ for $PM_{2.5,}$ SO_2 and NO_2, ppm for CO, °C for temperature, % for humidity, km/h for wind speed and units for vehicular traffic);

C_0 and C_1—the breakpoint concentrations that define the range for a specific *AQI* category;

I_0 and I_1—the corresponding *AQI* values for those breakpoints.

However, the models analyzed in this research include, in addition to pollutants (CO, NO_2, SO_2, and $PM_{2.5}$), other factors (temperature, humidity, wind speed, and vehicular traffic) that were not previously considered in calculating *AQI*. The authors of this study believe that besides the established factors found in the literature for calculating *AQI*, weather conditions and traffic are additional factors that should be incorporated into the *AQI* formula.

Figure 2 illustrates the correlations between *AQI* values and their corresponding levels of health concern. It provides a comprehensive overview of how varying *AQI* levels relate to potential health effects, highlighting the significance of monitoring AQ for public health awareness [70].

The *AQI* equation does not explicitly include meteorological parameters or traffic conditions, which influence AQ and, in particular situations, affect *AQI* values. Weather conditions influence the assessment of air pollutant concentration. For instance, humidity can impact the dispersion and retention of PM in the air. The wind can help disperse pollutants, thereby affecting their concentration.

Secondly, traffic is a significant source of air pollutants such as NO_2, $PM_{2.5}$, and PM_{10}. Vehicle emissions contribute to elevated levels of these pollutants, especially in urban areas with heavy traffic. Therefore, while the *AQI* formula may not directly incor-

porate meteorological and traffic parameters, they should be analyzed to interpret *AQI* values accurately.

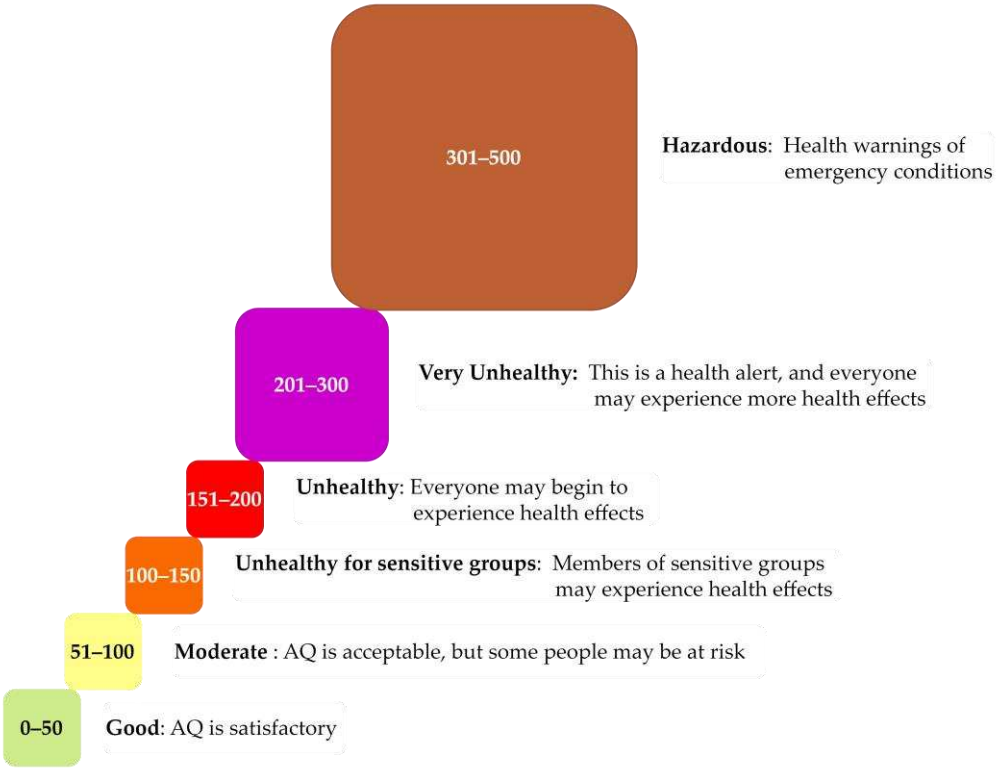

Figure 2. Correlation between *AQI* values and corresponding levels of concern [70].

The predictive algorithms provide a framework for analyzing AQ dynamics, ensuring informed decision-making in public health interventions. In the following analyzed models, the parameters CO, NO_2, SO_2, $PM_{2.5}$, temperature, wind speed, humidity, and vehicular traffic are named *features*. The corresponding *AQI* value is named *label*. Figure 3 presents the BronxStats table that provides the model entries. A total of 1305 records have been imported for the period from 1 January 2020 to 31 December 2023.

Figure 3 shows the correlations for each field from the source tables obtained through the imports and the final table BronxStats utilized in the model training tests. These correlations highlight the relationships between the input features and the label, which format impacts model performance.

The initial step in the training process is to have the data well-structured and formatted, as it ensures that the algorithms interpret the patterns within the dataset correctly. Properly formatted data allows for better handling of missing values, normalization, and feature engineering as part of the training preparation process, ultimately enhancing the model's predictive capabilities. This is particularly important as it directly affects the accuracy of the predictions made by the customized models.

The following sections analyze multiple models from various categories. Initially, algorithms that enable label prediction based on input values were examined. Next, another category of algorithms that facilitate forecasting for the next five days was evaluated.

This approach focuses on immediate predictions derived from current conditions and then extends to forecasts anticipating future outcomes. This dual analysis serves different purposes. The first will be used in an IoT infrastructure to report the current *AQI*, while the second will be used to identify a future trend for emergent solutions in case of *AQI* risk value.

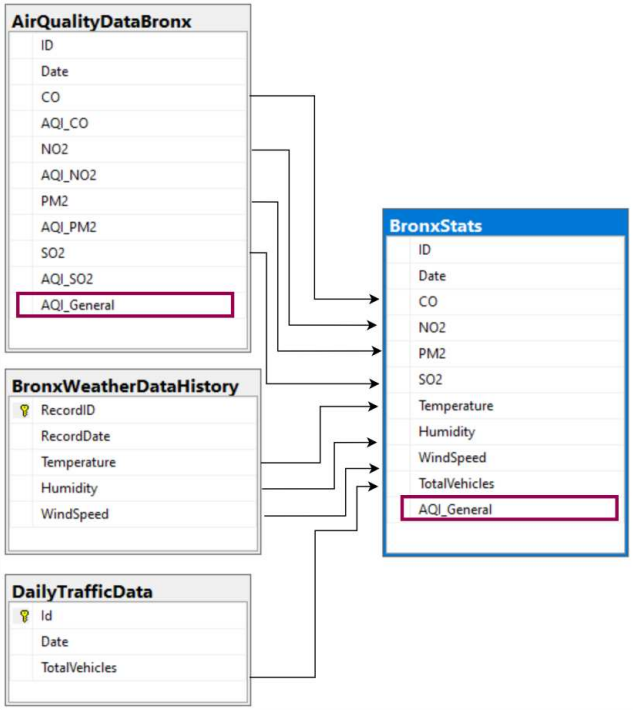

Figure 3. Correlations between features and labels in the training dataset.

Figure 4 presents the generic algorithm testing flow for model training and metrics evaluation. The same steps are applied for each algorithm presented for *AQI* prediction. In Figure 4, data are initially loaded into the TrainTestSplit. Next, the dataset is split into training sets (80%)—TrainSet and test sets (20%)—TestSet. In the pipeline step, the algorithm concatenates features and applies one of several ML or DL algorithms. In the following step, the model is fitted using the TrainSet. This trains the model by feeding it the data and applying the selected algorithm. Finally, the process evaluates the trained model on the TestSet. This step measures the model's performance using the metrics explained in the following subsection.

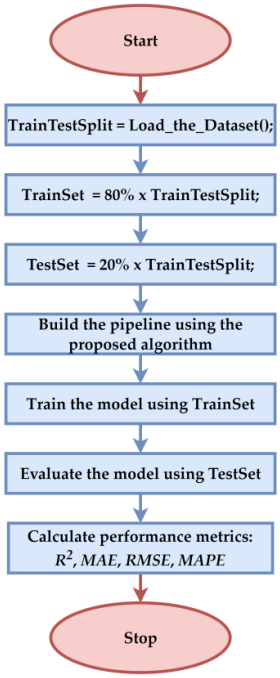

Figure 4. Logical Flow for Testing and Evaluating ML and DL Model.

2.2. ML Custom-Made Versus Pre-Implemented Algorithms for AQI

This study implements six customized and six ML.NET pre-implemented algorithms to train a model that can compute the *AQI* based on the input values presented in Figure 5. The Results section conducts a comparative study on the *AQI* solution's custom-made versus pre-implemented ML.NET algorithm.

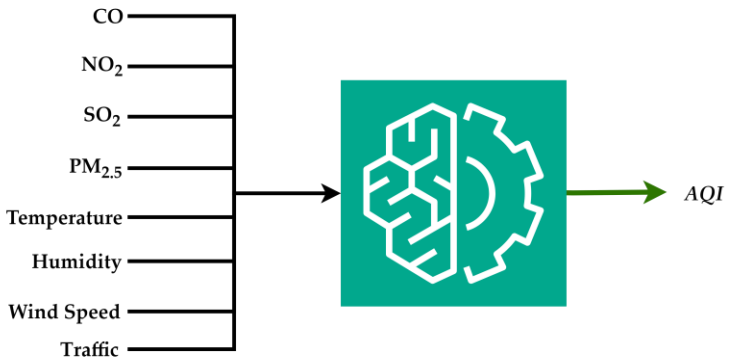

Figure 5. Conceptual diagram of the prediction model.

The custom-made ML models use the regression algorithms with all hyperparameters/parameters set by empirical tests and synthesized in Table 1, according to the algorithm employed, including [71]:

1. Random Forest Regression (RFR) uses decision tree decomposition in the training process, and next, it uses the mean prediction for each tree [72]. In the evaluated RFR, the hyperparameters/parameters were set as follows: the number of estimators (*n_estimators*) to 500, the maximum features in each tree (*max_features*) to 10, random state (*random_state*) to 0 and, the decision tree maximum depth (*max_depth*) was set to 15;
2. Decision Tree Regression (DTR) is used for regression tasks, where the data are split into subsets based on feature values, creating a tree-like model of decisions. This approach allows for the modeling of complex relationships without requiring linear assumptions. Decision trees are valued for their interpretability and ability to handle numerical and categorical data. They can also manage missing values, making them versatile in practical applications [73,74]. In this case, the hyperparameters/parameters were set as follows: the decision tree maximum depth (*max_depth*) was set to 5 (sufficient depth to ensure an accurate prediction, with low execution time), the random state (*random_state*) to 42, the minimum number of samples per split (*min_sample_split*) was 4, the minimum number of samples per leaf (*min_samples_leaf*) was set to 2, and the *criterion* hyperparameter was set to "*squared_error*", and the tree was split using minimum mean squared error (MSE);
3. Linear Regression (LR) represents one of the simplest forms of regression analysis. This method assumes a straight-line relationship and is widely used due to its simplicity and ease of interpretation. The algorithm does not perform well in some situations when complex patterns are not identified, which leads to underfitting [75]. In this case, the algorithm calculates the parameter intercept (*intercept_*) or bias (whose value is 7.014022494795359), the coefficients (*coef_*) or weights for each of the eight features (respectively [−5.140072451758598, 0.3317304406559781, 3.1989070961349095, −0.1828119410953855, −0.03925884240140581, 0.00782994619292085, 0.00233905 795529826, $1.0876253401024537 \times 10^{-6}$]), and a single CPU was used to evaluate the model;

4. The K-nearest neighbors (KNN) algorithm estimates the target variable's value by averaging the k-nearest neighbors in the feature space. This technique helps capture local patterns in the data and can adapt to various data distributions. Nonetheless, it tends to be computationally demanding, particularly with extensive datasets, since it necessitates calculating distances to every training sample [76]. We set the hyperparameters/parameters as follows: the algorithm uses 5 neighbors (*n_neighbors*), with all neighbors weighted equally (*weights* = "*uniform*"), 2 power parameters ($p = 2$) for the Minkowski metric, and a single CPU;
5. Support Vector Regression (SVR) is a regression algorithm employed to approximate the output, knowing the input [77]. For SVR, we set the hyperparameters/parameters as follows: the regularization parameter (*C*) to 1.0 value, the kernel type was employed as a Radial Basis Function-RBF (*kernel* = "*rbf*"), the *epsilon* value (tolerance) to 0.1, and the kernel coefficient, and the gamma parameter value (*gamma* = "*scale*") was calculated using the input features;
6. Gradient Boosting Regression (GBR) method constructs an ensemble sub-trees based on the errors provided by the predecessor trees. Gradient boosting is known for its ability to model complex relationships [78]. In this case, the hyperparameters/parameters setup contains a learning rate of 0.1 ("*learning_rate*": 0.01), the number of estimators was set to 500 ("*n_estimators*": 500), the maximum depth of each regression estimator was set to 4 ("*max_depth*": 4), the minimum number of samples per split was set to 5 ("*min_samples_split*": 5), and it was used a squared error type loss function ("*loss*": "*squared_error*").

Table 1. Hyperparameters/parameters tuning for ML Models.

Model	Hyperparameters/Parameters	Value
RFR	n_estimators	500
	random_state	0
	max_depth	15
	max_features	10
DTR	max_depth	5
	min_samples_split	4
	min_samples_leaf	2
	criterion	MSE
	random_state	42
LR	features weights	[−5.14, 0.33, 3.19, −0.18, −0.03, 0.007, 0.002, 0]
	intercept (bias)	7.01
KNN	n_neighbors	5
	weights	uniform
	p	2
	metric	minkowski
SVR	kernel	rbf
	C	1.0
	epsilon	0.1
	gamma	scale
GBR	n_estimators	500
	max_depth	4
	min_samples_split	5
	learning_rate	0.01
	loss	squared_error

Table 1 lists the tuning hyperparameters/parameters for ML models. These hyperparameters/parameters optimize the performance of the analyzed ML algorithms. The values are extracted from the Python 3.9 software and TensorFlow library used for the customized algorithms.

The six above-mentioned algorithms were specifically tailored to address the characteristics of the labels analyzed. At this stage, 1305 entries have been utilized for training.

The pre-implemented ML.NET analyzed models are as follows [79,80]:

1. Fast Forest and Fast Tree are specific implementations of decision tree algorithms optimized for speed. Fast Forest combines multiple decision trees to improve predictive performance while maintaining computational efficiency. Fast Tree is designed to build trees quickly and make predictions, making it suitable for large datasets [81,82];
2. FastTreeTweedie is a modified version of the FastTree tree algorithm. When predicting *AQI*, which varies widely and may not follow a normal distribution, FastTreeTweedie is expected to provide a more accurate estimate by accommodating the characteristics of the *AQI* data, particularly when dealing with extreme values;
3. Generalized Additive Models (GAM) increase linear models by incorporating smooth functions to capture non-linear relationships between independent and dependent variables. This flexibility makes GAMs a powerful tool for modeling complex data while retaining interpretability [75];
4. LbfgsPoissonRegression treated *AQI* as count data, especially in cases of high pollution levels, capturing the frequency of extreme AQ events;
5. Online Gradient Regression involves updating regression models incrementally as new data become available, making it suitable for dynamic environments where data stream continuously. Online gradient regression algorithms are designed to minimize computational costs while maintaining predictive performance [83].

The algorithms' performance was assessed through standard metrics [84]:

- R^2 quantifies the proportion of variance in the *AQI* explained by the predictive models based on feature input values. An R^2 score of 1 indicates that the model perfectly predicts *AQI* based on the selected pollutants, whereas a score of 0 suggests that the model fails to account for any variability in the *AQI*;
- *Mean Absolute Error (MAE)* measures the average magnitude of the absolute errors in predicting the *AQI* from input values. It provides insight into the average discrepancy between predicted *AQI* values and actual observations, offering a straightforward metric for assessing the accuracy of the AQ forecasts;
- *Mean Absolute Percentage Error (MAPE)* evaluates the accuracy of the *AQI* predictions as a percentage of the actual *AQI* values. *MAPE* is particularly useful in this context, as it allows for a relative assessment of prediction accuracy across different levels of air pollution, accommodating the variations in pollutant concentrations;
- Median Absolute Error (*MedAE*) indicates the median absolute errors between predicted and actual *AQI* values. This metric offers a measure of prediction accuracy less affected by outliers, especially during extreme pollution events that could distort the results.

These indicators enable comparative analysis between different customized models. Assessing performance through metrics such as R^2, *MAE*, *MAPE*, and *MedAE* identifies which modeling approach (ML or DL) yields the most accurate predictions of AQ.

2.3. DL Custom-Made Algorithms for AQI

This study also implemented seven customized solutions for the analyzed parameters. In AQ prediction, various artificial neural network (ANN) architectures (such as the one

proposed by the authors in Figure 6) offer distinct advantages for modeling the relationships between air pollutant concentrations and the corresponding *AQI*.

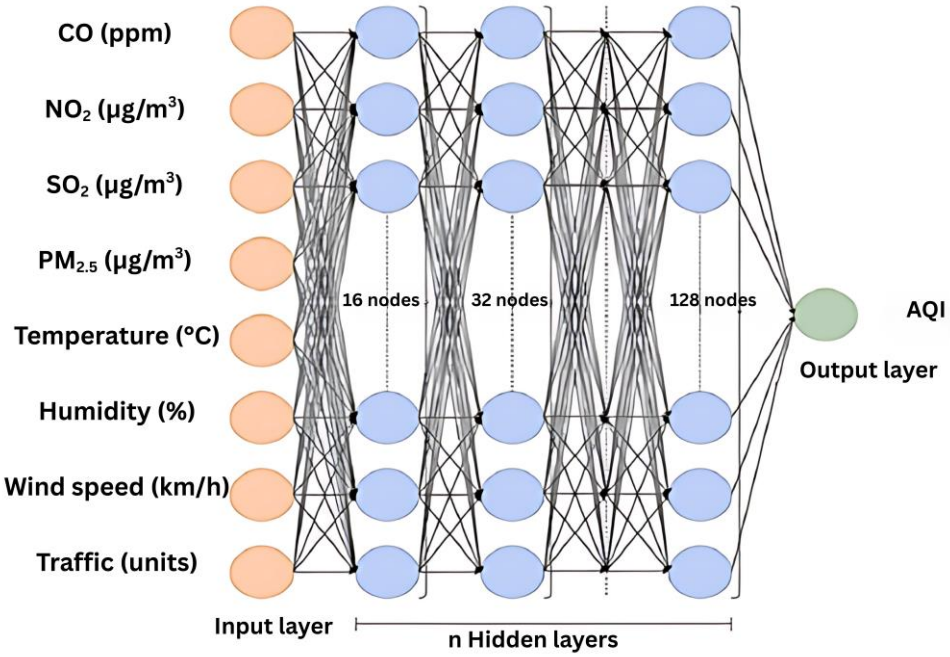

Figure 6. The ANN proposed architecture.

The same features are used for this prediction model, as depicted in Figure 5. As for the DL tailored models, the parameters and hyperparameters/parameters were set by empirical tests. The neural network types employed in the analysis are as follows:

1. Feedforward Neural Networks (FNNs) represent the most basic category of ANNs, characterized by non-cyclical node connections. In FNNs, data flow unidirectionally, moving from input nodes through any hidden nodes and ultimately reaching the output nodes. This architecture is specific for tasks where the relationship between input and output is straightforward and does not depend on previous inputs [85]. In this case, the hyperparameters that define the FNN architecture are nine dense layers (the first dense layer representing the network input layer), each dense layer having a specific number of neurons (respectively: 7, 8, 16, 32, 64, 128, 256, 256, 64), using Rectified Linear Unit (ReLu)-type activation function, one dropout layer to prevent overfitting (dropout rate 0.2), and one output layer (a dense one) with just one neuron, without activation function. Other hyperparameters are the employed optimizer was "Adam" with a learning rate of 0.001, the loss function for regression tasks was MSE, the number of epochs used for network training was set to 1000, batch size was 32, the validation split was set to 0.1 (10% of the data is used for validation during training), and the connections between layers are made in a feedforward manner (connected sequentially). The model parameters are the neurons connections weights (for instance, for the first dense layer) the weights shape is (1, 7), with associated weight matrix [0.618999, 0.47214946, 0.2781748, 0.753425, 0.06874061, 0.7170182, 0.35377005], and the bias for each neuron (for instance, the biases shape for the first dense layer) (7) and the corresponding biases vector is [−0.0030382, −0.00307668, 0.01217235, −0.00328257, −0.00333186, −0.00342433 0.00128003];
2. Temporal Convolutional Networks (TCNs) utilize convolutional layers to process sequential data, offering an alternative to RNNs. TCNs are characterized by their ability to capture long-range dependencies through dilated convolutions, which allow for a larger receptive field without increasing the number of parameters [86]. The

hyperparameters that define the TCNs architecture are: eight 1D Convolutional layers (the first layer representing the input layer-input shape, with a standard kernel size of 2 and causal type padding), each one with a specific number of output filters (4, 8, 16, 32, 64, 128, 256, 64) and ReLu-type activation function applied to each Conv1D layer, one dropout layer to prevent overfitting (dropout rate 0.2), one flatten layer used to transform the convolutional output into a one-dimensional vector, and an output layer (a dense one) with just one neuron. Another hyperparameter setup was as follows: the kernel size was set to 2, the number of filters was 4, 8, 16, 32, 64, 128, 256, 64 (increasing and then decreasing), the dilation rate was set to 1 for the first Conv1D layer and 2 for the rest of them, the dropout rate was 0.2, activation function was ReLu-type, the padding was "causal". The hyperparameters learning rate for the "Adam" type optimizer, the loss function for regression tasks, the number of epochs used for network training, the batch size, the validation split, and the connections between layers were set to the same values as in the FNNs. The model parameters are the neurons connections weights (for instance, for the first Conv1D layer), weights of shape are (2, 1, 4) with associated weight matrix [[−0.585093, −0.42209578, −0.10097494, 0.88292724]; [−0.65418446, 0.56988806 0.11362608, 0.11557351]], and the bias for each neuron (for instance, the biases shape for the first Conv1D layer) is (4), and the corresponding biases vector is [0, 0.05134838, −0.0748504, 0.0022911];

3. Recurrent Neural Networks (RNNs) handle sequential data by incorporating cycles in their architecture, allowing information to persist. This feature enables RNNs to maintain a form of memory. Standard RNNs face challenges with long-term dependencies because of problems such as vanishing and exploding gradients [87,88]. In this case, the hyperparameters that define the RNN architecture are three SimpleRNNs layers (the first one representing the input layer-input shape that defines the shape of the input data for the RNN layers, with 32 RNN units with a return sequence; the second one with 128 units with a return sequence and the third one with 256 units without a return sequence, the output of each RNN layer using ReLu-type activation function), a dropout layer to prevent overfitting with a dropout rate of 0.2, and an output layer (a dense one) with just one neuron. Other hyperparameters have the same values as those of FNNs and TCNs. The model parameters are the neurons connections weights (for instance, for the first SimpleRNNslayer) the weights shape is (1, 32) with associated weight matrix [−0.37684393, −0.3347999, 0.12421719, 0.04884235, 0.40717593, −0.2224836, 0.11467709, −0.44855782,. . ., 0.24549635, −0.4644161, 0.14050183, −0.421698, 0.4127007, 0.20661259, −0.17216238, −0.0187266], and the biases shape for each neuron (for instance, the biases shape for the first SimpleRNN layer) is (32) with the corresponding vector bias [0.20325282, 0.01771547, 0.27512175, . . ., 0.12933277, 0.10233897, −0.00640139];
4. LSTMs are a form of RNN featuring memory cells and gating mechanisms that manage the flow of information. This architecture is used for tasks requiring the retention of information over extended periods, such as speech recognition and natural language processing [87,89,90]. In this case, the hyperparameters that describe this type of network architecture are two LSTM layers (the first LSTM layer contains the input shape, each one with 100 neurons, with ReLu-type used activation function, first LSTM layer with a return sequence and the second one without a return sequence), two dropout layers (with a dropout rate of 0.2 to prevent overfitting), and one output layer (a dense one, with just one neuron). Other hyperparameters have the same values as FNNs, TCNs, and RNNs. The model parameters are the neuron connection weights (for instance, for the last layer, the dense one), the weight matrix is (100, 1)

(respectively, [[−0.1917704], . . ., [0.03949241]]), and the bias shape of the dense layer is (1), a 1D array ([0.34271002]).

5. General Regression Neural Networks (GRNNs) are FNNs designed for regression tasks. They utilize a kernel-based approach to estimate the output for a given input, used for problems where the relationship between input and output is not strictly linear [85,91]. The hyperparameters that define the FNN architecture are five dense layers (hidden layers—dense ones, the first one representing the input layer with 64 units, the second one with 128 units, the third and the fourth layer with 256 units, and the fifth layer with 64 units, all dense layers using Radial Basis Function (RBF) activation function), one dropout layer to prevent overfitting with dropout rate 0.2 and, and an output layer (a dense one) with just one neuron. Other hyperparameters have the same values as FNNs, TCNs, RNNs, and LSTMs. Regarding the parameters, the important ones are the neurons connections weights (for instance, for the first dense layer) weight shape is (64, 128) with the corresponding weights matrix ([[0.09693588, 0.11530785, 0.04327405, . . ., −0.01015339, −0.00465284, −0.16290204], . . ., [0.18151447, −0.00127745, 0.11786038, . . ., −0.06098074, 0.0883746, −0.14132085]]) and bias shape (for the same first dense layer) that is (128), with the corresponding bias array [0.01425829, −0.019097, 0.02077781, 0.05533804, −0.02579543, −0.05466483, . . ., 0.03854505, −0.08813311, −0.0760707, 0.03319431, −0.03673444, −0.02343925, −0.01249062, −0.04072443].
6. Time-Delay Neural Networks (TDNNs) are designed to process temporal data by incorporating delays in the input layer. This architecture allows TDNNs to adapt unstructured data speech recognition, image processing, and time series analysis [92]. In this case, the hyperparameters that define the TDNN architecture are an input layer (the input shape that is determined by the number of the time step and the number of features in the training data), five hidden layers (dense ones, first one with 64 units, second one with 128 units, the third and the fourth one with 256 units, and the fifth one with 64 units, each one using ReLu-type activation function), a dropout layer to prevent overfitting (dropout rate of 0.2), a flatten layer used to convert the output of the last dense layer to a one-dimensional vector necessary for the output layer and, one output layer with just one unit (a dense one). Other hyperparameters are the same and with the same values as in the case of FNNs, TCNs, RNNs, LSTMs, and GRNNs, to which it is added the number of time steps (*time_steps*) that was set to 4. Regarding the model parameters, for instance, for the first dense layer, the weights of the shape are (64, 128) with the associated matrix weights ([[−0.18985237, −0.23790528, −0.26162702, . . ., −0.21098962, −0.34780186, 0.00897838], . . ., [0.23329817, −0.03779398, 0.08295902 . . . −0.13313578, 0.11072736, −0.10848673]]) and biases shape (128), with the associated biases 1D array [$-6.65585697 \times 10^{-2}$, $2.33586300 \times 10^{-2}$, $-8.02270137 \times 10^{-3}$, $-6.08175024 \times 10^{-2}$, . . ., $-3.14048007 \times 10^{-2}$, $-1.27918571 \times 10^{-1}$, $-8.52884054 \times 10^{-2}$, $-2.51952391 \times 10^{-2}$].
7. Deep Belief Networks (DBNs) are associated with complex hierarchical data representations. DBNs capture intricate relationships between features and the *AQI* for AQ prediction, improving forecasting performance [93,94]. This type of network architecture presents the next hyperparameters: an input layer (that corresponds to the number of features in the dataset), a sequence of autoencoders that successively apply transformations to the data, each one encoding previous data representations, respectively, four encoding layers that perform data compression (dense ones, with ReLu-type activation function), one decoding layer to reconstruct the input data (a dense one, with "sigmoid" type activation function), one dropout layer to prevent overfitting (dropout rate 0.2), and an output layer (a dense one, with just one unit). Other hyperparameters are the same and with the same values as in the case

of FNNs, TCNs, RNNs, LSTMs, and GRNNs, to which added the data autoencoders encoding dimensions that are [1024, 64, 512, 32], the autoencoders training (number of epochs for autoencoders training was set to 50, batch size 32 and, the validation split was set to 0.1). Regarding the model parameters, for instance, for the second layer (first encoder layer of the autoencoder) the weights shape is (1024, 64) with the corresponding matrix weights [[−0.01099487, 0.05026998, −0.04905977, . . ., −0.06741306, 0.15472981, −0.25435853], . . ., [−0.04147308, −0.02906811, −0.00823338, . . ., −0.05285922, −0.1270246, −0.180068]] and biases shape is (64) with the corresponding bias vector [−0.02391353, −0.0217452, −0.02374031, −0.02275715, −0.00915561, −0.02361129, . . ., −0.02615629, −0.0149049, −0.0102234, −0.08077513].

Table 2 contains the tuning hyperparameters/parameters for DL-tailored models.

Table 2. Hyperparameters tuning for DL Models.

Hyperparameters/Parameters	Model FNNs	TCNs	RNNs	LSTMs	GRNNs	TDNNs	DBNs
Layers number	9 (Dense); 1 (Dropout);	8 (Conv1D); 1 (Dropout); 1 (Flatten);	3 (SimpleRNNs); 1 (Dropout);	2 (LSTM); 2 (Dropout);	5 (Dense); 1 (Dropout);	5 (Dense); 1(Dropout); 1 (Flatten);	4 encoders (dense); 1 decoder (dense) 1(Dropout);
Input layer	1 (Dense)	1 (Conv1D)	1 (SimpleRNNs)	1 (LSTM)	1 (Dense)	1 (InputLayer)	1 (InputLayer)
Output layer	1 (Dense)	1 (Dense)	1 (Dense)	1 (Dense)	1 (Dense)	1 (Dense)	1 (Dense)
Input layer size	7	4	32	100	64	(4, 8); *time_steps* = 4; *number of features* = 8;	8
Output layer size	1	1	1	1	1	1	1
Number of neurons per layer	[7, 8, 16, 32, 64, 128, 256, 256, 64]	[4, 8, 16, 32, 64, 128, 256, 64]	[32, 128, 256]	[100, 100]	[64, 128, 256, 256, 64]	[64, 128, 256, 256, 64]	[8, 1024, 64, 512, 32, 0, 1]
Dropout rate	0.2	0.2	0.2	[0.2, 0.2]	0.2	0.2	0.2
Optimizer	Adam	Adam	Adam	Adam	Adam	Adam	Adam
Loss Function	MSE	MSE	MSE	MSE	MSE	MSE	MSE
Epochs	1000	1000	1000	1000	1000	1000	1000
Batch Size	32	32	32	32	32	32	32
Learning rate	0.001	0.001	0.001	0.001	0.001	0.001	0.001
Activation function	ReLu	ReLu	ReLu	ReLu	RBF	ReLu	ReLu (encoder layers); Sigmoid (decoder layers);
Connections between layers	connected sequentially	connected sequentially	connected sequentially	connected sequentially	connected sequentially	fully connected (dense) connection	connected sequentially
Weights shape (selection)	(1, 7)	(2, 1, 4)	(1, 32)	(100, 1)	(64, 128)	(64, 128)	(1024, 64)
Biases shape (selection)	(7)	(4)	(32)	(1)	(128),	(128)	(64)

In the case of these neural network models, the comparative analysis is conducted on the same metrics, such as R^2, *MAE*, *MAPE*, and *MedAE*. Comparing the performance of different models will facilitate the selection of the most suitable model for addressing AQ issues.

To establish a unified metric for evaluating the algorithms' performance, we propose the formula shown in Equation (2), where the R^2 parameter is assigned a weight of 40%, *MAPE* 30%, and both *MAE* and *MedAE* are weighted at 15% each.

$$OP_k = 0.4 \times R_k^2 + 0.3 \times (1 - MAPE_k) + 0.15 \times (1 - MAE_k) + 0.15 \times (1 - MedAE_k) \quad (2)$$

where OP_k—the unified comparing measure for the algorithm, k, representing the overall precision;

R_k^2—the coefficient of the algorithm, k, should approach 1 for the perfect performance;

$MAPE_k$, MAE_k, $MedAE_k$—the error metrics for algorithm k that should be as close to 0 as possible.

R^2 is highly important (40%) as it comprehensively measures how well the model explains the data variance. *MAPE* (30%) is prioritized for its focus on the percentage-based error, which helps in understanding relative prediction errors across different scales. *MAE* and *MedAE* (15% each) contribute by offering insights into the absolute errors of the model in handling outliers. These metrics provide a balanced and multi-dimensional model performance evaluation, ensuring accuracy.

Initially, we analyzed the possibility of distributing the weights equally, but the metrics are not equally relevant in analyzing the model's performance. For this reason, we assigned the highest percentage to the parameter that illustrates the direct performance of the model without significantly diminishing the importance of the other parameters, namely 40%. Subsequently, the remaining 60% could have been distributed evenly, but again, the parameters are not equally important. Therefore, we assigned 30% to *MAPE* because it provides valuable context for deviations in prediction accuracy. Finally, the remaining 30% was distributed evenly between *MAE* and *MedAE*, as both parameters analyze extreme deviations and potential outliers. We did not consider these two parameters as important as R^2 or *MAPE* because they play a complementary role in the model analysis. In summary, the authors acknowledge the empirical nature of these values and agree that better weight distributions may exist. However, the proposed 40:30:15:15 ratio was considered a starting point for this research.

2.4. Forecasting Algorithm for AQI

The research builds a tailored predictive model based on *AQI* features after determining the most accurate model for predicting *AQI* (MA-AQI). This model was designed to refine *AQI* predictions, improving accuracy and adapting the prediction framework to specific dataset characteristics or external influencing factors like seasonal variations and vehicular traffic patterns. The MA-AQI model corresponds to the value prediction scenario. In this scenario, the predicted *AQI* value requires all the values for the input features. However, the value for the input features is not available, so the model is available only for real-time readings and *AQI* prediction.

The *AQI* forecasting (AQIF) model incorporates additional features represented by the data series. The historical dataset enables the model to analyze trends and variations in vehicular traffic intensity in conjunction with weather conditions. This behavior affects accurate *AQI* forecasting in an urban environment like Bronx County.

The AQIF uses the ForecastBySsa algorithm, described in the paper by Ayilu et al. [95]. The analyzed parameters conceptually depicted in Figure 7 are as follows [96]:

- The input column contains the historical *AQI* values used to train the model. The accuracy of predictions relies on the quality and quantity of the historical data between 1 January 2020 and 31 December 2023;

- The windowSize determines how many past observations are considered to predict future values. Multiple tests were provided for this parameter to obtain the best value for the accuracy of the forecasting;
- The horizon parameter specifies how many future days the model will forecast. This value is set to 5 days;
- The output column contains the forecasted *AQI* values for the next five days, predicted after the model is trained. These values are displayed in the proposed environmental monitoring and public health application.

Figure 7. Conceptual Diagram of the AQIF.

The AQIF model's evaluation included the same performance metrics as the previously mentioned ML models, R^2, *MAE*, *MAPE*, and *MedAE*.

2.5. Real-Time Prediction and Forecasting Application

The Real-Time Prediction and Forecasting (RTPF) application aims to inform the public and authorities about the current *AQI* and forecasted values for the next five days. This information enables timely measures to prevent adverse effects on the population.

Implementing the RTPF involves addressing two key integration challenges:

- The locally trained MA-AQI and AQIF models need to be hosted in the cloud to ensure they are accessible;
- A reliable data acquisition system must be established to use real-time data as inputs for the MA-AQI model. This involves collecting data from various sensors and devices and centralizing it within an IoT infrastructure. There are two primary approaches to achieve this:
    - Data can be entered into a cloud-hosted database via API requests. This method facilitates data storage and retrieval while ensuring that the information remains up-to-date and accessible for analysis;
    - Another solution involves employing an IoT hub service like Azure IoT Hub. This service platform connects IoT devices, enabling real-time data ingestion, management, and analysis.

The RTPF application displays the collected data using the ASP.NET Web Application environment. It shows the current parameters along with the *AQI* generated by the MA-AQI model. The predicted values from the AQIF model for the next five days are displayed in the application's graphical interface.

Figure 8 depicts the conceptual diagram of the prediction and forecasting system process. The data are collected in real-time from multiple sources in the Bronx using an IoT infrastructure. The system utilizes models to predict and forecast AQ using the mentioned parameters. A detailed breakdown of the components and their interactions for the proposed IoT infrastructure is provided below.

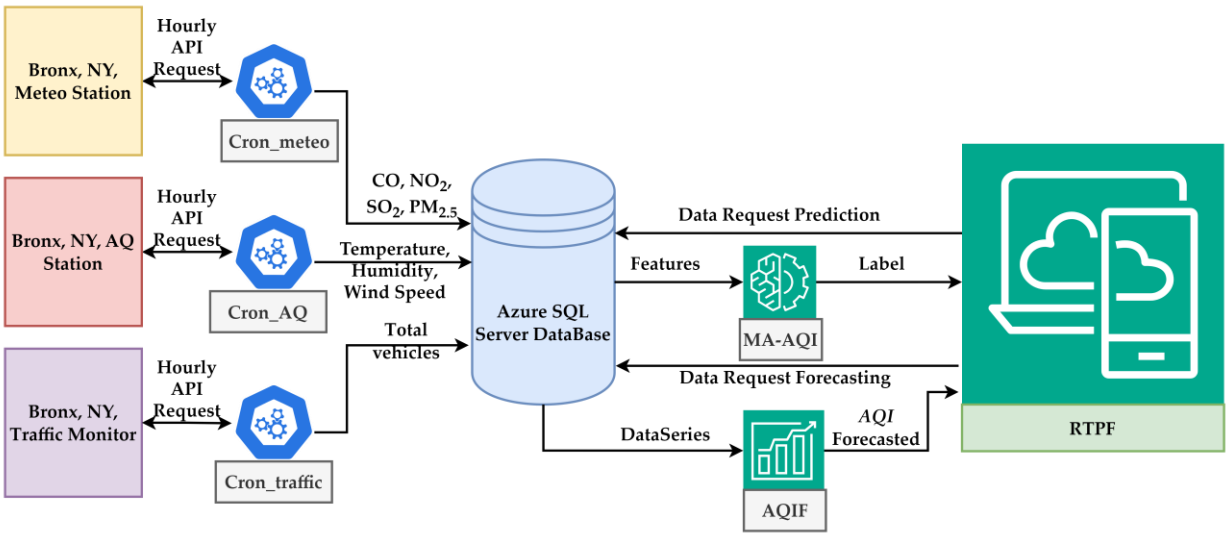

Figure 8. The conceptual diagram for the proposed IoT infrastructure.

1. Data Collection Sources:
   - Meteo Station. This component retrieves hourly data on weather parameters. The data are collected via an API when the cron job Cron_meteo runs hourly on the API. The result produced by the API is sent to the SQL Server Database hosted in Azure Cloud;
   - AQ Station. AQ parameters such as CO, NO_2, SO_2, and $PM_{2.5}$ are gathered from the AQ monitoring station. These data are also transmitted via an API to the central database hourly when the Cron_AQ job is executed;
   - Traffic Monitor. The number of vehicles is collected in real-time through an API and sent to the Azure SQL Server when the Cron_traffic hourly execution is released.
2. The central repository Azure SQL Server Database receives and stores all the data from the three primary sources (Meteo Station, AQ Station, and Traffic Monitor). The stored data include air pollutants, weather conditions, and traffic information, which the prediction and forecasting models then utilize.
3. The MA-*AQI* model's prediction receives the collected features (pollutant levels, weather data, and traffic counts) from the Azure SQL Database and predicts the *AQI* as the label. This is used to determine the current state of AQ based on the data inputs.
4. Forecasting Model AQIF forecasts the future *AQI* based on historical data series from the Azure SQL Database. It predicts the future *AQI* trend, potentially for proactive measures or alerts.
5. RTPF collects the predicted *AQI* from MA-AQI and the forecasted *AQI* from AQIF. This platform provides a real-time dashboard where users view the current and predicted AQ data on different devices (laptops, tablets, or mobile devices).

The basic RTPF procedure includes the following stages:

- Data Integration. All collected environmental and traffic data are integrated into a central Azure SQL Server database for further processing;
- Real-Time Prediction. The system predicts current *AQI* based on real-time data using the MA-AQI model;
- Forecasting. The system forecasts future *AQI* trends using historical data.

APIs continuously feed data into the system through the corresponding cron jobs, which run the API hourly. The results of prediction and forecasting are displayed to users in real-time.

Figure 9 displays a screenshot of the RTPF application user interface, indicating the 37 February 2024 date. The current temperature is 6.07 °C, the humidity is 85.70%, and the wind direction is 29.40°. The traffic volume is 62,543 vehicles, and the predicted *AQI* is 55.

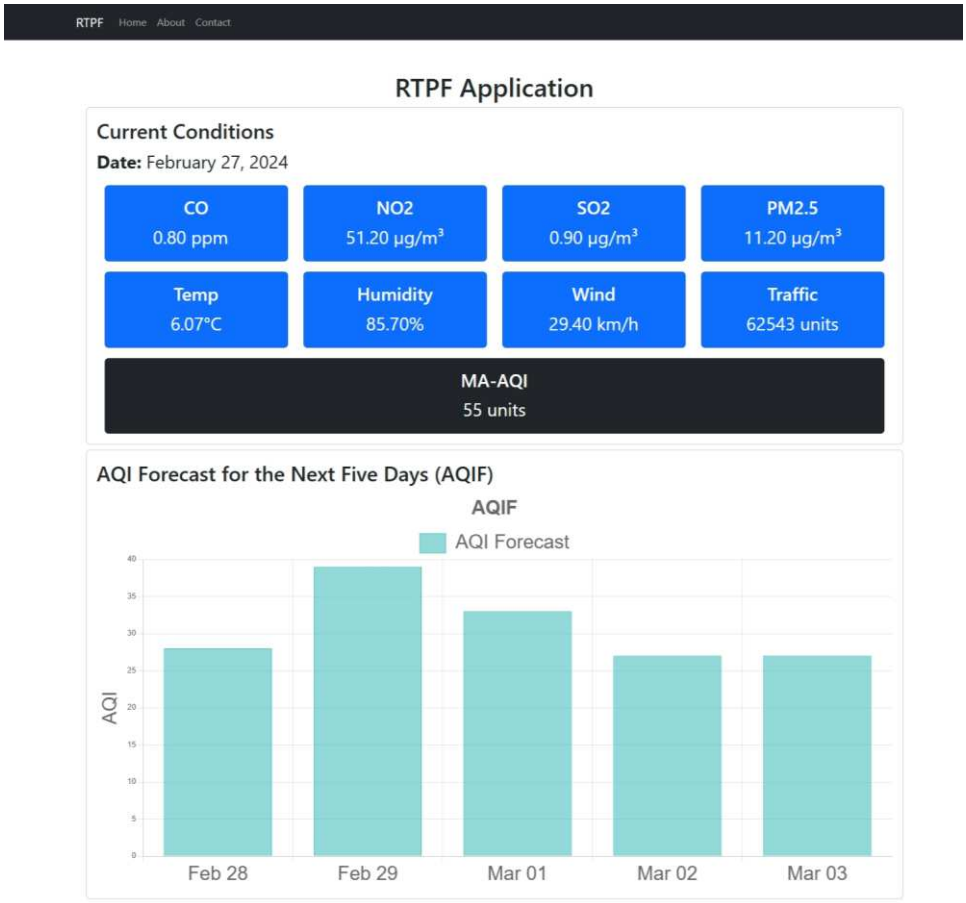

Figure 9. The graphical user interface of the RTPF application.

The *AQI* forecast values for the upcoming five days, i.e., 28, 29 February and 1, 2, and 3 March, are as follows: 27, 35, 33, 26, and 25, respectively. These values are presented in a bar chart format, clearly comparing the projected *AQI* over this period. The bar chart presents the *AQI* trend, with a peak of 50 on 1 March, suggesting a potential increase in air pollution levels on that day. This visualization aids users in quickly grasping the expected trends and making informed decisions regarding outdoor activities and health precautions based on the AQ forecast.

2.6. Emergency Response Protocol Integration for Exceeding Safe AQI Levels

Based on the AQIF model, the population and authorities are informed about the evolution of *AQI* values up to five days before. This enables proactive measures to be proposed when the forecasted *AQI* values exceed 100, corresponding to the "Unhealthy for Sensitive Groups" category, as outlined in Figure 2. *AQI* values may be suddenly altered due to a spontaneous, unforeseen event. The MA-AQI model will detect the *AQI* value in these cases and propose an alternative emergency assessment protocol.

The proposed intervention protocols (PIP) based on *AQI* categories are integrated using Azure services to enhance AQ management. The PIP is integrated using Azure

Communication Services, which facilitates real-time communication through automated SMS and email notifications. This service enables the seamless integration of smart notifications, community action alerts, and public information campaigns, ensuring that relevant stakeholders receive timely alerts and updates about AQ events.

Figure 10 presents the Azure Communication Service integrated for PIP. After testing, the two SMS and email services were implemented using C# language, including a custom message for the population.

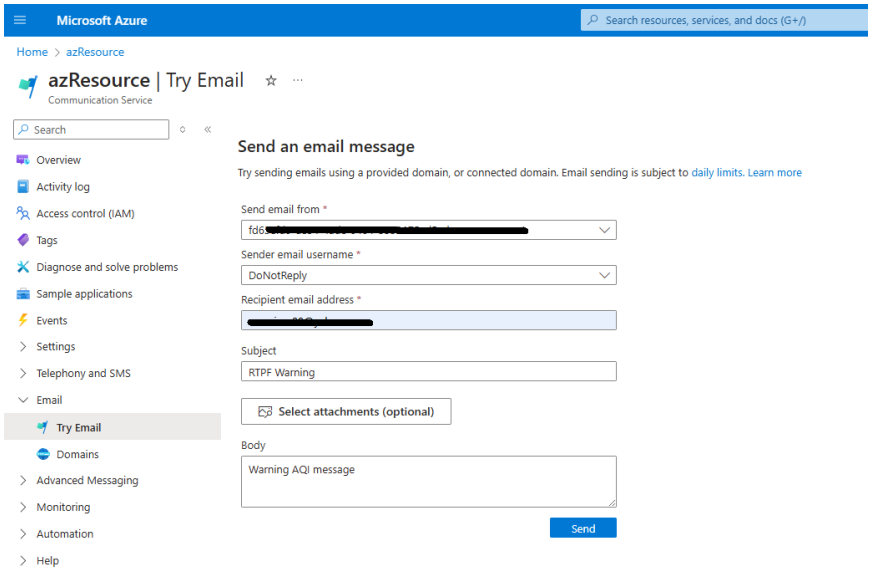

Figure 10. Azure communication services for PIP. (Note: * stands for required field).

Using these communication channels enhances communities' responsiveness to AQ conditions, contributing to the protection of public health. Emails will inform the public by delivering detailed information, updates, and general announcements to a broad audience through the warning message category. At the same time, SMS messages will be reserved for urgent notifications for immediate, time-sensitive alerts. This last trigger is associated with the alert category.

The flowchart presented in Figure 11 provides a decision-making process for handling *AQI* notifications based on real-time data from the MA-AQI model. It outlines the steps to determine whether to send informative emails or SMS alerts, depending on the *AQI* levels.

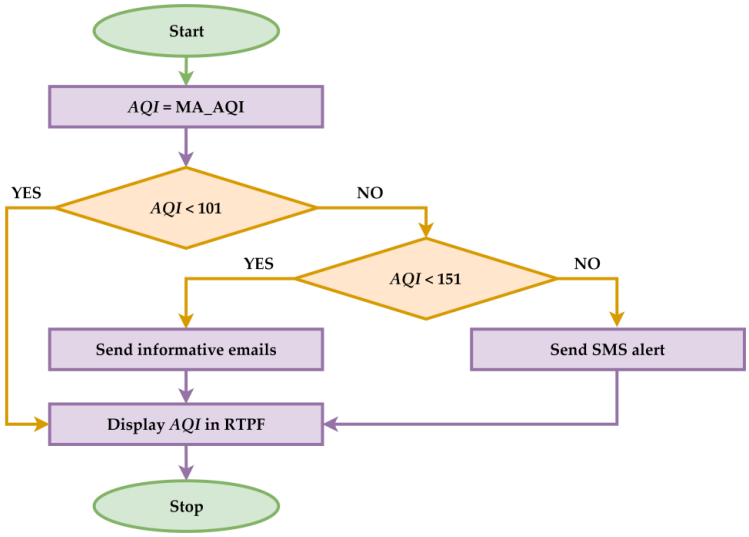

Figure 11. Decision flow for *AQI* notifications based on real-time data using PIP.

The process triggers informative emails if the *AQI* is between 101 and 150, indicating moderate *AQI*. When the *AQI* exceeds 150, indicating an unhealthy *AQI*, an SMS alert is sent instead of emails due to urgency. Afterward, the *AQI* is displayed in the real-time platform.

This system ensures communication emails for general updates and SMS for immediate alerts when AQ reaches critical levels.

3. Results

This section presents the results of the twelve implemented and tailored ML models for this research, the results from the tests for the seven DL algorithms implemented and customized by the authors, and a comparative analysis of these *AQI* prediction algorithms. Next, the outcomes of the forecasting algorithm are discussed, along with a concrete example of the RTPF application in use and a detailed overview of an alert event.

3.1. ML and DL Model Results

This study initially focused on implementing and evaluating several ML and DL algorithms to enhance the accuracy of *AQI* predictions. The research includes a comparative analysis of six implemented and customized ML algorithms, seven implemented and customized DL algorithms, and six pre-implemented ML (ML.NET) models.

For each of the six ML regression algorithms (RFR, DTR, LR, KNR, SVR, and GBR) that were implemented in Python 3.9 and customized by the authors for the analyzed problem (*AQI* prediction) was achieved, first of all, a much needed primary data processing (missing values and outliers removing, normalization, data splitting into training, evaluation, validation sets, etc.), followed by the configuration of each algorithm hyperparameters/parameters(such as the number of estimators, random state, maximum depth, learning rate, kernel type, number of neighbors, the minimum number of samples per split, intercept, maximum leaf nodes, loss function, etc.) for training and validating processes, parameters that are different from one model to another. Also, the best parameters were identified for each analyzed ML regression algorithm through an intensive trial-and-error testing process.

The study explored seven DL models after identifying the top-performing ML model (implemented and customized by the authors or pre-implemented (ML.NET).

Regarding the analyzed DL algorithms (FNNs, TCNs, RNNs, LSTMs, GRNNs, TDNNs, and DBNs) that were implemented in Python 3.9 and customized for the analyzed problem (*AQI* prediction), in addition to a much-needed data preprocessing step, the following actions were taken:

- For each customized DL algorithm (unlike the original algorithm), a dropout layer was added that ignores twenty percent of the data for network overfitting prevention, and in the case of some algorithms (for instance, for RNNs and LSTMs) was integrated a return sequence for better performances. Also, the input layer (the input shape) for some types of networks (for instance, for FNNs and GRNNs) was integrated into the network-dense layers. In contrast, for TCNs, the input shape was incorporated into the convolutional layers, with a standard kernel size of 2 and a causal type padding;
- For each customized DL algorithm, the number of layers (dense layers) of densely interconnected nodes (that are using ReLu or RBF), the number of nodes for each layer (for instance, FNNs contain nine layers of densely interconnected nodes that start at seven nodes and doubles in number up to two hundred and fifty-six nodes), the number of 1D Convolutional layers (for TCNs) that supplies the output filter starting from a number of four nodes to a number of two hundred and fifty-six nodes, and the number of flattening layers (for TCNs, such a layer being used to flatten the output to

the correct size) were established. Also, the number of SimpleRNN layers (for RNNs, where the first layer contains thirty-two nodes with a return sequence, the second one has one hundred and twenty-eight nodes with a return sequence, and the third one includes two hundred and fifty-six nodes without a return sequence), the number of final layers that contains just one output (for instance, for DBNs and TCNs), and the number of layers of autoencoders (for example, for DBNs).

Regarding the results supplied by the implemented and customized DL methods (FNNs, TCNs, RNNs, LSTMs, GRNNs, TDNNs, and DBNs), for correct comparison of them, we established the same number of trainable parameters (around one hundred and twenty thousand) that indicate the global parameters of the entire neural network (with the architecture, from the authors' point of view, presented in Figure 6). Respectively, for the components of each neuron in the proposed network and for the analyzed problem (*AQI* prediction), the models were trained in one thousand epochs using the Adaptive Moment Estimation (ADAM) algorithm and the MSE function. The number of epochs used for training each DL method was established through experimental testing, respectively, through a trial-and-error testing process.

The primary objective of this exploration was to ascertain whether these advanced DL techniques, implemented and tailored by the authors, could yield superior results compared to the traditional ML (implemented and customized by the authors or pre-implemented ML) approaches. By leveraging DL's capabilities to capture complex patterns and relationships within large datasets, we aimed to determine if these models could provide more accurate predictions of *AQI* values, ultimately contributing to better AQ management and public health outcomes.

Various performance metrics were utilized to assess the models, *including* R^2, *MAE*, *MAPE*, and *MedAE*. Table 3 presents the algorithms' results for a training set of 912 entries and an evaluation set of 393 records. The total number of analyzed entries extracted from the custom SQL Server database is 1305, corresponding to the *AQI* monitoring interval between 1 January 2020 and 31 December 2023.

A comparative discussion regarding the best performance algorithm for ML versus DL is conducted regarding performance metrics and *OP* indicators.

RFR is the top performer among the fully customized algorithms. It achieves an impressive R^2 of 0.9944, which indicates that it explains 99.44% of the variance in *AQI* values. It has a low *MAE* of 0.2237 and an *MAPE* of 0.0068. These metrics demonstrate RFR's exceptional accuracy in predicting AQ. Its *OP* is set to 0.95618, classifying the RFR algorithm as the most performant in *AQI* prediction.

In contrast, all the other implemented and customized ML algorithms exhibit a low-performance value for the *OP* indicator, as all indicated a negative value, suggesting inadaptability to the *AQI* problem.

A negative R^2 value indicates that the model is not suitable for the type of dataset and is unable to model the *AQI* problem for the selected features. Usually, a negative R^2 may indicate overfitting, underfitting, or inadequate data that include noisy features. However, the fact that the RFR model generates an R^2 with an accuracy of 0.9944 suggests that the other models are unsuitable for the *AQI* problem, given the selected features.

Among the implemented DL algorithms, TCNs demonstrate the best performance, achieving an impressive R^2 of 0.9934, a low *MAPE* of 0.0172, and an exceptionally low *MAE* of 0.5235. The *OP* indicator set to 0.86275 indicates a high accuracy in predicting *AQI* values compared to the other implemented and tailored DL algorithms.

Compared to the two, RFR is the superior choice for *AQI* prediction, with a higher R^2 and lower *MAE* and *MAPE* than TCNs. This comparison illustrates the strengths of traditional ML models in achieving optimal performance for *AQI* prediction. Overall, the

findings highlight the importance of tailored algorithm selection, where RFR demonstrates superior performance in *AQI* prediction compared to the other models.

Table 3. Performance results of *AQI* prediction algorithms on training and evaluation sets.

Algorithm	R^2	*MAE* [%]	*MAPE* [%]	*MedAE* [%]	*OP*
Implemented and customized ML algorithms					
RFR	0.9944	0.2237	0.0068	0.0399	0.95618
DTR	0.9791	1.3196	0.0475	1.0153	0.627155
LR	0.9186	3.1369	0.1070	2.6027	0.0744
KNR	−0.175	12.155	0.4481	11.0000	−3.07768
SVR	0.9902	1.1273	0.0391	1.2282	0.631025
GBR	0.9940	0.4259	0.0181	0.1696	0.902845
ML.NET integrated algorithms					
FastForest	0.9482	2.7009	3.3538	10.1051	−1.94776
FastTree	0.9634	0.9753	2.8193	2.6264	−0.400685
FastTreeTweedie	0.9744	0.7510	2.3555	2.0555	−0.137865
Gam	0.9079	2.8480	4.4729	10.2693	−2.346305
LbfgsPoissonRegression	−1.5016	19.6176	23.3069	68.3076	−20.18149
OnlineGradientDescent	−4.0409	29.6999	33.0849	78.9704	−27.242375
Implemented and customized DL algorithms					
FNNs	−0.0026	11.8325	0.4199	11.3324	−3.001745
TCNs	0.9934	0.5235	0.0172	0.3395	0.86275
RNNs	0.9957	0.7009	0.0235	0.6142	0.793965
LSTMs	0.9788	1.5709	0.0661	1.4351	0.52079
GRNNs	0.9939	0.8900	0.0317	0.7995	0.734625
TDNNs	−0.1785	12.483	0.4254	10.417	−3.03402
DBNs	0.9973	0.5520	0.0182	0.4556	0.84232

Data in Table 3 outline that the DL models' prediction performance was inferior to that of the ML models. This can be explained by the dataset of 1305 records, which may be insufficient for the generalization of DL-type predictions. Due to computational constraints and time limitations, we did not fully explore every possible hyperparameter combination, which could explain the relatively lower performance. Future research will detail each model individually, but the present paper provides an overview of the most suitable algorithm for the particularities of the *AQI*.

The authors explain that RFR, as an ML model, outperforms all DL models because of the dataset size and complexity. DL models require large datasets. However, our dataset includes 1305 records. Extensive data could not capture the urbanization image, which has inherent implications for *AQI*. As a result, RFR outperforms DL models because they are easier to understand and tune (having fewer hyperparameters/parameters than DL models), are faster to train, are less prone to overfitting and robust to outliers, they perform well with less data than DL models, and maybe the most important fact is that they can handle complex data and non-linear relationships between features.

3.2. Model Validation

The best-performing model, RFR, is validated by real-time monitoring of the MA-AQI system. The validation process spans from 1 March 2024 to 1 July 2024. The input feature values were acquired in real-time and fed into the MA-AQI model for *AQI* prediction during this period. The model's output values were then compared with the official *AQI* values reported by authorities. As shown in Figure 12, comparing the predicted *AQI* values generated by the RFR model, and the official *AQI* measurements reveals a significant

correlation. The average deviation between the expected and actual reported values was 0.58%, indicating that the model's predictions were highly accurate and closely aligned with real-world data. This strong performance further supports the MA-AQI model in predicting AQ based on real-time data inputs.

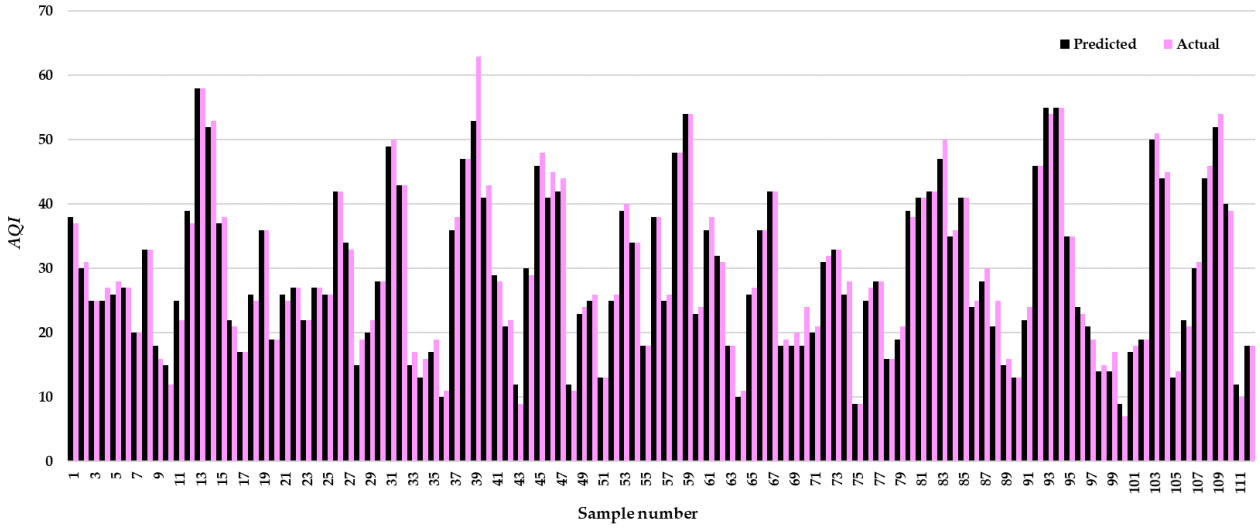

Figure 12. Comparison of predicted and actual *AQI* values between 1 March 2024 and 1 July 2024.

The chart presented in Figure 12 compares predicted *AQI* values with actual *AQI* measurements. The bar chart shows that the predicted values reproduce the real data with fidelity. The accuracy of the MA-AQI model demonstrated its applicability in production or commercial scenarios. The slight variations between the two sets of values are minimal, indicating that the predictions generated by the model are reliable for real-world applications. Variations of 1 or 2 units in *AQI* values are not a significant concern, as such minor differences fall within the acceptable margin of error for AQ measurements and do not substantially impact the interpretation of *AQI*.

3.3. Forecasting Model Results

In the forecasting model ForecastBySsa, 400 tests were conducted, with the value of windowSize iteratively adjusted from 1 to 400. Various performance metrics were calculated for each iteration, focusing on the R^2, which measures the model's accuracy. Figure 13 illustrates the evolution of R^2 concerning different windowSize values. The analysis determined that the optimal R^2 value was achieved when windowSize was set to 325. This indicates that the model performs best with this specific window size. The employed C# script is available as a Supplementary Material (ForecastingScript.txt).

In Figure 13, the red value indicates the best R^2 value for the windowSize variation. The indicator corresponds to the windowSize of 325. The AQIF model with this indicator achieved the highest R^2 of 0.7162, indicating moderate accuracy. Low errors are reflected by an *MAE* of 0.4, *MedAE* of 0.4, and *MAPE* of 0.9%, demonstrating its ability to forecast *AQI* with minimal deviation.

Negative R^2 values observed for other windowSize values indicate suboptimal parameter tuning. Inappropriate windowSize choices lead to overfitting or underfitting, resulting in poor predictions and negative R^2.

Expanding the dataset can improve the forecasting component. The authors introduced this component into the research with a demonstrative purpose, aiming to illustrate the concept of urban-level information dissemination.

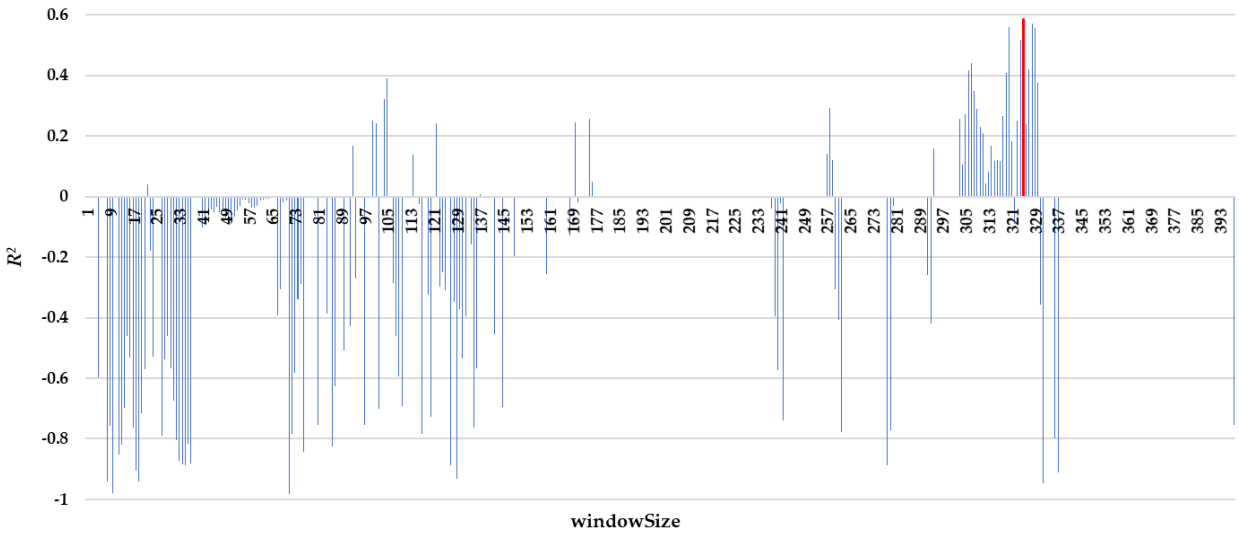

Figure 13. Evolution of R^2 Based on windowSize Adjustments in the AQIF model.

3.4. PIP Warnings and Alerts

When the *AQI* is between 101 and 150, indicating moderate warning according to Figure 2, PIP sends users a warning message via email. For instance, on 30 June 2023, the calculated *AQI* value was 138 in the Bronx. This exceeded the threshold for sensitive groups, prompting a warning message. Figure 14 shows the RTPF interface, where the exceedance of the *AQI* value can be observed. The interface highlights the elevated *AQI* levels and provides users with real-time updates and recommendations based on current AQ conditions. Notably, in Figure 14, the information level is marked in yellow on the RTPF platform. Figure 15 presents the content of the notification message sent by Azure Communication Services via email.

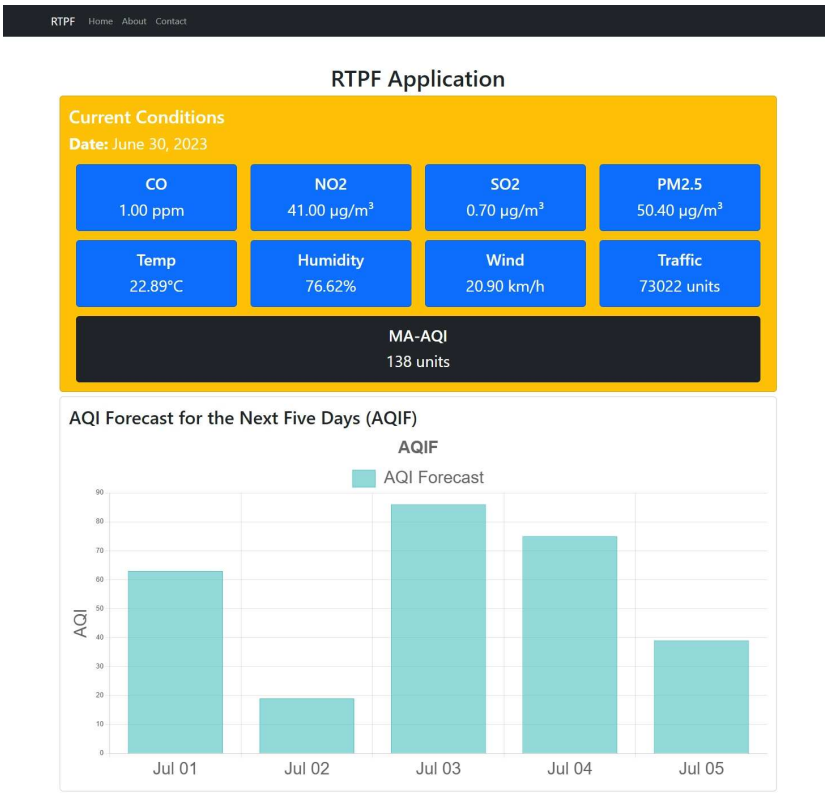

Figure 14. Example of the RTPF graphical interface—warning message.

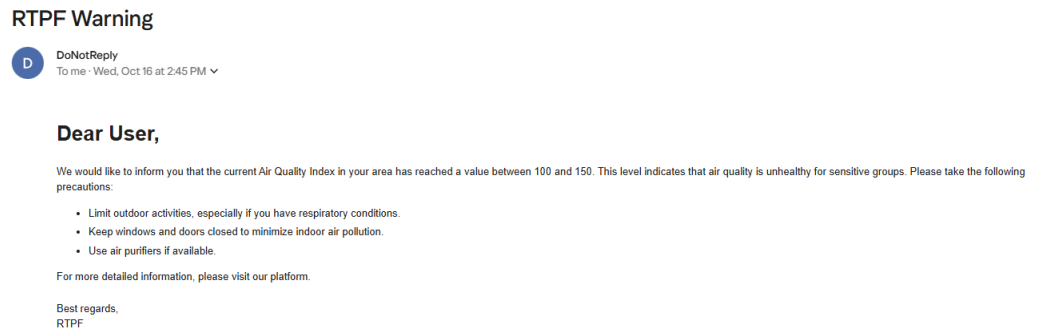

Figure 15. Content of the email notification message.

On 7 June 2023, in the Bronx, the calculated *AQI* value was 194, which should trigger the transmission of an SMS message via Azure Communication Service. Figure 16 displays the graphical interface of the RTPF application, while Figure 17 presents a screenshot of the sent SMS message. In this case, the notification box is highlighted in red due to the elevated alert level, emphasizing the situation's urgency.

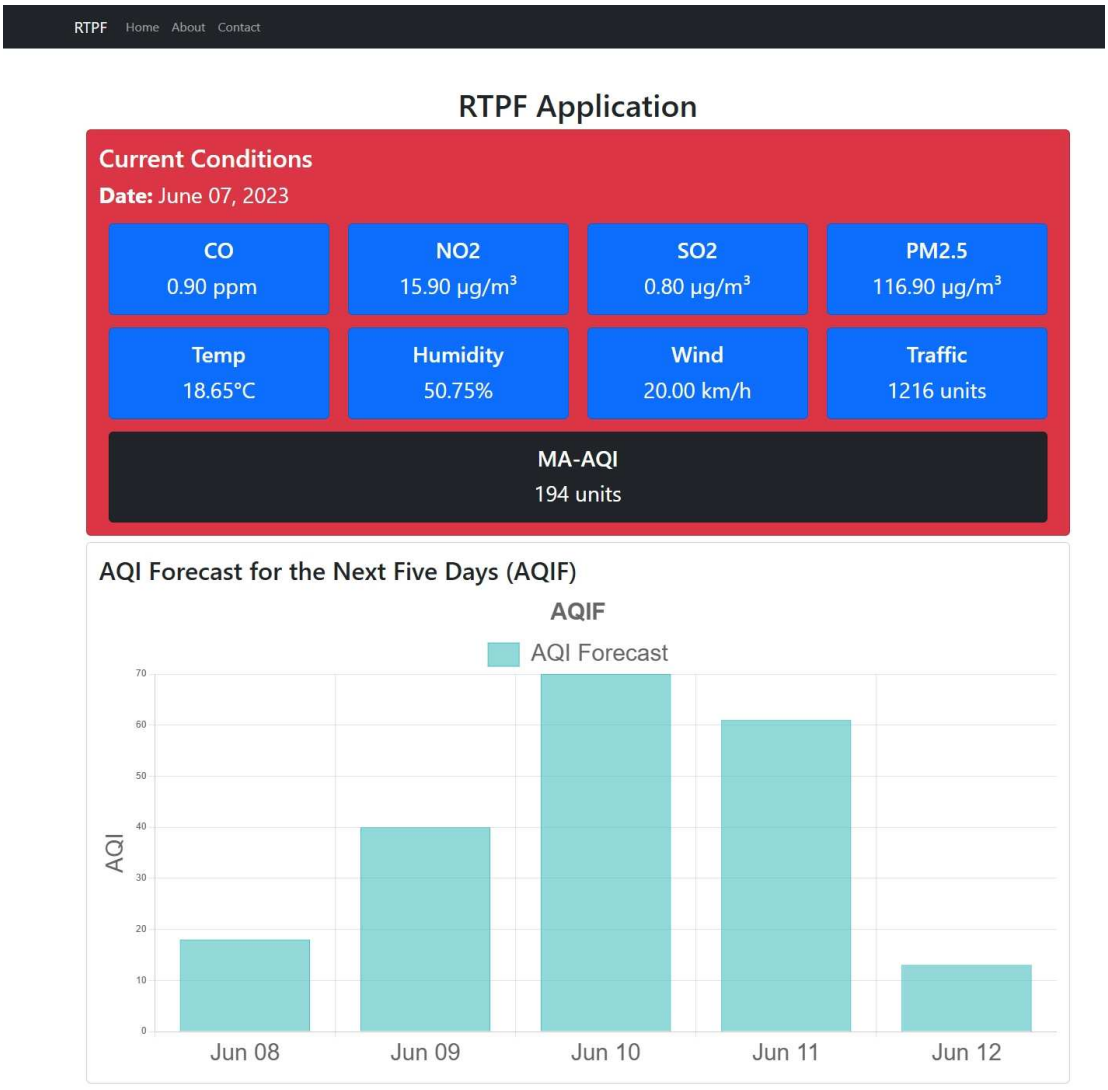

Figure 16. Example of the RTPF graphical interface—alert message.

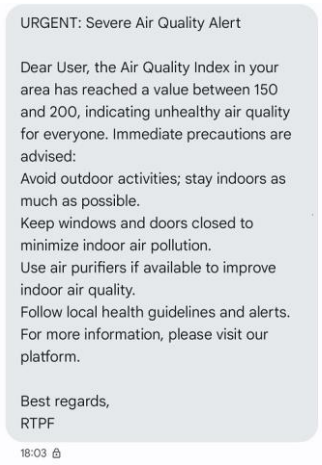

Figure 17. Content of the SMS notification message.

The MQ-AQI model has demonstrated practical usability due to its high *OP* of 95.61%. This level of precision was achieved through the customized design and implementation of the predictive algorithm RFR. Additionally, the AQIF model has shown an R^2 accuracy of 71.62%, indicating its effectiveness in forecasting future *AQI* data.

The RTPF application underscores practical usability by integrating the PIP component, which relies on the MA-AQI and AQIF models. These models empower authorities and the public to take informed actions regarding AQ, enhancing community awareness and promoting proactive measures for a healthier environment.

4. Discussion

This research aimed to develop an *AQI* prediction model based on a unique combination of parameters in the literature, including CO, NO_2, SO_2, $PM_{2.5}$, temperature, humidity, wind speed, and vehicular traffic. The model was trained on data corresponding to the period from 1 January 2020 to 31 December 2023, for the Bronx, but it is designed to be universally applicable in other regions with similar data. The exclusion of PM_{10} is justified by its partial overlap with $PM_{2.5}$, a subcategory of PM_{10}. This indicator is considered more dangerous to health because it penetrates deeper into the respiratory system. For this reason, numerous studies opt to use only $PM_{2.5}$ as a representative indicator of suspended particles. O_3 is a secondary pollutant whose formation depends on complex photochemical reactions between other pollutants (precursors) such as NO_2 and VOCs, under the influence of solar radiation. Due to this complexity, O_3 modeling requires a separate approach, with specific input factors different from those integrated in this research. Moreover, in the Bronx area, O_3 concentrations are influenced by the regional transport of pollutants, making it difficult to directly correlate them with local traffic sources or specific weather conditions.

The paper evaluated 19 prediction algorithms, which were hyperparameterized to achieve the best results. In the end, it was determined that RFR achieved the best results. The validation was carried out from 1 March 2024 to 1 July 2024, with an accuracy deviation of 0.58%. The model metrics were an R^2 score of 99.59%, an MAE of 0.22%, an MAPE of 0.68%, and an OP score of 95.61%. For a comparative analysis of the results, Panaite et al. [97] analyze NO, NO_2, and CO on $PM_{2.5}$, which obtained an R^2 of 82.76%, while MA-AQI obtained 99.59%.

The AQIF model recorded an R^2 of 71.62%, demonstrating the ability to anticipate *AQI* trends over five days but still leaving room for further improvements. Chan et al. [98] explore the $PM_{2.5}$ and PM_{10} parameters through time series. The results of this research indicate R^2 values of 79% while analyzing two parameters. Unlike this research, AQIF obtained an R^2 of 71.62% while simultaneously analyzing eight parameters. In the paper

by Cican et al. [99], two models that used NO_2 as input were investigated, and the results showed R^2 values below 80%. These comparative results are summarized in Table 4.

Table 4. Number of analyzed parameters and R^2 value for AQIF and other studies.

Number of Analyzed Parameters	R^2 (%)	Reference
8	71.62	AQIF
4	82.76	[97]
2	79	[98]
1	<80	[99]

The models MA-AQI and AQIF were integrated into a dedicated real-time prediction and forecasting application, RTPF. An automated PIP notification system was also implemented, which generates alerts via email or SMS based on the estimated *AQI* values.

The first limitation of this research is that the dataset includes only four years (2020–2023), which may limit the generalizability of the DL models. At the same time, no variation in hyperparameters was analyzed to identify the optimal setup for each model. For this research, the default values for the hyperparameters were employed. This may have impacted the performance of the DL models, combined with the limited number of samples in the dataset.

While the MA-AQI model demonstrates high predictive accuracy, the AQIF model exhibits a lower R^2 accuracy, indicating certain limitations in its capacity for long-term forecasting. This constraint is particularly evident when attempting to predict more complex, evolving AQ patterns, highlighting areas where the model's precision could be improved. The models were customized based on data from the Bronx, New York, which restricts the generalizability of the findings. Using data from a single location with specific industrial activity levels and environmental characteristics may limit the model's applicability to regions with different pollution profiles or activity types.

The authors wanted to include the level of industrialization as a parameter of the model in their research. Still, when conducting the project, it was impossible to find a source from which they could extract data on the degree of industrialization. Additionally, the authors wanted to use the demographic level in the area of the analyzed city, but they faced the same problem with the data source. In the future, the authors wish to develop a new, improved model that includes these additional parameters. Future research will also include PM_{10} and O_3.

Future research will focus on improving the accuracy of the AQIF model in capturing complex long-term AQ patterns. Improvements will incorporate additional environmental parameters that better reflect the evolving nature of AQ influencers over time. Furthermore, we aim to propose the development of a new IoT device specifically designed to monitor a broader range of ecological parameters. This advanced device would support a more extended data collection, improving the predictive capabilities of AQ models by capturing previously unmeasured environmental factors.

5. Conclusions

This research proposes novel features for the *AQI* model and sets out to develop a system for predicting and forecasting AQ by integrating ML and DL models into an IoT infrastructure. The main research question focused on whether a customized approach to predictive modeling could improve AQ monitoring and forecasting accuracy. Developing and deploying the MA-AQI and AQIF models demonstrated that combining tailored algorithms improves AQ prediction and future trend forecasting.

The MA-AQI model achieved a high *OP* of 95.61%, providing a reliable solution for real-time AQ monitoring. The AQIF model, while achieving a lower R^2 accuracy of 71.62%, demonstrated its potential for longer-term forecasting. Integrating these models into the RTPF platform, combined with the PIP component for decision-making, highlights the practical application of our system in real-world scenarios, enabling both authorities and the public to take informed actions. Two informative messages were integrated using Azure Communication Service, for the interval between 101 and 150 sending a warning email message and for the *AQI* interval between 151 and 200 sending an SMS email alert.

Our PIP alert system stands out by generating notifications based on *AQI*-specific measurements and incorporating additional parameter readings that influence air quality. Furthermore, it provides warning alerts based on forecasting predictions, a capability that current alert systems lack, making our solution proactive.

Reflecting on the research process, the study successfully addressed the challenges of integrating complex ML and DL models into an IoT framework. However, further work is needed to improve the accuracy of the AQIF model and extend data analysis to another industrial area.

The study presents advancements in *AQI* monitoring and prediction, integrating data analysis, ML, DL, and IoT framework for a comprehensive approach to AQ management. The research begins with the import and analysis of AQ data, aimed at identifying parameters involved in the AQ monitoring.

A core contribution is the analysis and development of six customized ML algorithms specifically designed to address the unique features of AQ data and improve predictive accuracy. Additionally, six pre-existing ML models were seamlessly integrated using ML.NET, enriching the predictive framework with diverse algorithmic approaches. To further bolster prediction capabilities, the study implemented and incorporated seven customized DL algorithms tailored for AQ forecasting, accounting for AQ data's intricate patterns and time dependencies.

In addition to model development, the study offers a thorough comparative analysis of ML and DL approaches. By introducing a novel performance metric proposed by the authors, the analysis reveals the strengths and limitations of each methodology, identifying the optimal strategy based on key performance indicators.

Another notable aspect of the study is integrating a dual-model forecasting system, which allows for adaptable and accurate AQ predictions over variable timeframes. This model flexibility is complemented by embedding the predictive framework within an IoT infrastructure for real-time data integration and AQ monitoring.

This study developed a user-friendly application incorporating predictive and forecasting capabilities, making AQ insights readily accessible for end-users. Finally, a comprehensive decision-making framework was proposed, leveraging Azure Communication Services to drive AQ management actions. This framework supports informed decision-making processes, ensuring that the AQ management actions are responsive and data-driven, significantly contributing to advancements in AQ monitoring and response systems.

In conclusion, this study provides a comprehensive framework for advancing AQ monitoring and prediction through innovative data integration, customized ML and DL models, and seamless IoT integration. It ultimately supports data-driven decision-making and AQ management.

Supplementary Materials: The following supporting information can be downloaded at: https://www.mdpi.com/article/10.3390/app15084390/s1, ForecastingScript.txt.

Author Contributions: Conceptualization, C.-M.R. and M.C.; methodology, C.-M.R., M.C. and A.S.; software, C.-M.R. and M.C.; validation, C.-M.R., M.C. and A.S.; formal analysis, C.-M.R. and A.S.; investigation, M.C. and A.S.; resources, A.S.; data curation, C.-M.R.; writing—original draft preparation, C.-M.R., M.C. and A.S.; writing—review and editing, A.S.; visualization, A.S.; supervision, A.S. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Petroleum-Gas University of Ploiesti, Romania.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

%	Percentage
$\mu g/m^3$	Micrograms per cubic meter
ADAM	Adaptive Moment Estimation
AI	Artificial intelligence
AIS	Automatic Identification System
ANN	Artificial neural network
AQ	Air quality
AQI	Air quality index
AQIF	Air Quality Index Forecasting
AQMS	Air quality monitoring system
CNN	Convolutional neural network
CO	Carbon monoxide
DBN	Deep Belief Network
DL	deep learning
DTR	Decision Tree Regression
EPA	Environmental Protection Agency
FNN	Feedforward Neural Network
GAM	Generalized Additive Models
GBR	Gradient Boosting Regression
GRNN	General Regression Neural Network
GSM	Global System for Mobile Communications
GUI	Graphical user interface
IoT	Internet of Things
km/h	kilometers per hour
KNN	K-nearest neighbors
LR	Linear Regression
LSTM	Long Short-Term Memory
MA-AQI	Most accurate model for predicting AQI
MAE	Mean Absolute Error
MAPE	Mean Absolute Percentage Error
MedAE	Median Absolute Error
ML	Machine learning
MSE	Mean squared error
NO_2	Nitrogen dioxide
O_3	Ozone
OMAP	Open Meteo API Portal
OP	Overall precision

PIP	Proposed intervention protocols
PM	Particulate matter
PM_{10}	Particulate matter with diameters of 10 μm and smaller
$PM_{2.5}$	Particulate matter with diameters of 2.5 μm and smaller
ppm	Parts per million
RAQP	Recurrent air quality predictor
RBF	Radial Basis Function
ReLu	Rectified Linear Unit
RFR	Random Forest Regression
RMSE	Root mean square error
RNN	Recurrent neural network
RTPF	Real-Time Prediction and Forecasting
SMS	Short Message Service
SO_2	Sulfur dioxide
SVR	Support Vector Regression
TCN	Temporal Convolutional Network
TDNN	Time-Delay Neural Network
VOC	Volatile organic compound

References

1. Munfarida, I.; Arida, V. Air Pollution Assessment in the Main Roads of Surabaya-Indonesia During Post COVID-19. *Int. J. Environ. Sustain. Soc. Sci.* **2023**, *4*, 664–671. [CrossRef]
2. Chong, C. Carbon Monoxide Pollution and Limited Health Service Access in Third-world Countries. *J. Glob. Ecol. Environ.* **2024**, *20*, 17–27. [CrossRef]
3. Rosca, C.-M.; Stancu, A.; Neculaiu, C.-F.; Gortoescu, I.-A. Designing and Implementing a Public Urban Transport Scheduling System Based on Artificial Intelligence for Smart Cities. *Appl. Sci.* **2024**, *14*, 8861. [CrossRef]
4. Panait, M.; Voica, M.C.; Rădulescu, I. Approaches Regarding Environmental Kuznets Curve in the European Union from the Perspective of Sustainable Development. *Appl. Ecol. Environ. Res.* **2009**, *17*, 6801–6820. [CrossRef]
5. Wang, Y.; Li, C.; Ruan, Z.; Ye, R.; Yang, B.; Ho, H.C. Effects of Ambient Exposure to Nitrogen Dioxide on Outpatient Visits for Psoriasis in Rapidly Urbanizing Areas. *Aerosol Air Qual. Res.* **2022**, *22*, 220166. [CrossRef]
6. Zhang, Y.; Yang, D.; Wu, R.; Li, Y.; Yang, X.; Wang, R.; Xu, H. Characterization of roadside air pollutants: An artery road of Lanzhou city in northwest China. In Proceedings of the 8th International Symposium on Vehicle Emission Supervision and Environment Protection, Wuhan, China, 23–24 June 2022; p. 01039. [CrossRef]
7. Srivastava, S.; Kumar, A.; Bauddh, K.; Gautam, A.S.; Kumar, S. 21-Day Lockdown in India Dramatically Reduced Air Pollution Indices in Lucknow and New Delhi, India. *Bull. Environ. Contam. Toxicol.* **2020**, *105*, 9–17. [CrossRef]
8. Rendana, M.; Komariah, L.N. The relationship between air pollutants and COVID-19 cases and its implications for air quality in Jakarta, Indonesia. *J. Pengelolaan Sumberd. Alam Dan Lingkung.* **2021**, *11*, 93–100. [CrossRef]
9. Olmo, N.R.S.; Do Nascimento Saldiva, P.H.; Braga, A.L.F.; Lin, C.A.; De Paula Santos, U.; Pereira, L.A.A. A review of low-level air pollution and adverse effects on human health: Implications for epidemiological studies and public policy. *Clinics* **2011**, *66*, 681–690. [CrossRef]
10. Rosca, C.M.; Popescu, M.; Patrascioiu, C.; Stancu, A. Comparative Analysis of pH Level Between Pasteurized and UTH Milk Using Dedicated Developed Application. *Rev. Chim.* **2019**, *70*, 3917–3920. [CrossRef]
11. Anwar, M.Z.; Rindarjono, M.G.; Ahmad. The impact of transportation growth on the increase SO_2 and NO_2 gases in Surakarta City during 2013–2020. In Proceedings of the International Conference on Anthropocene, Global Environmental Change and Powerful Geography, Online, Indonesia, 27 September 2022; p. 012028. [CrossRef]
12. Wyche, K.P.; Nichols, M.; Parfitt, H.; Beckett, P.; Gregg, D.J.; Smallbone, K.L.; Monks, P.S. Changes in ambient air quality and atmospheric composition and reactivity in the South East of the UK as a result of the COVID-19 lockdown. *Sci. Total Environ.* **2021**, *755*, 142526. [CrossRef]
13. Apostu, S.A.; Gigauri, I.; Panait, M.; Martín-Cervantes, P.A. Is Europe on the Way to Sustainable Development? Compatibility of Green Environment, Economic Growth, and Circular Economy Issues. *Int. J. Environ. Res. Public Health* **2023**, *20*, 1078. [CrossRef] [PubMed]
14. Stancu, A. Impact of COVID-19 Pandemic on International Agricultural Trade in European Countries. In *Shifting Patterns of Agricultural Trade*; Erokhin, V., Tianming, G., Andrei, J.V., Eds.; Springer: Singapore, 2021; pp. 263–284. [CrossRef]

15. Akan, A.P.; Coccia, M. Changes of Air Pollution between Countries Because of Lockdowns to Face COVID-19 Pandemic. *Appl. Sci.* **2022**, *12*, 12806. [CrossRef]
16. Alzayed, M.A.; Aljame, M.; Ahmad, I. Investigating the impact of COVID-19 lockdowns on environmental health in Kuwait. *J. Eng. Res.* **2023**, *11*, 317–328. [CrossRef]
17. Sarmadi, M.; Rahimi, S.; Rezaei, M.; Sanaei, D.; Dianatinasab, M. Air quality index variation before and after the onset of COVID-19 pandemic: A comprehensive study on 87 capital, industrial and polluted cities of the world. *Environ. Sci. Eur.* **2021**, *33*, 134. [CrossRef] [PubMed]
18. Schiavo, B.; Morton-Bermea, O.; Arredondo-Palacios, T.E.; Meza-Figueroa, D.; Robles-Morua, A.; García-Martínez, R.; Valera-Fernández, D.; Inguaggiato, C.; Gonzalez-Grijalva, B. Analysis of COVID-19 Lockdown Effects on Urban Air Quality: A Case Study of Monterrey, Mexico. *Sustainability* **2022**, *15*, 642. [CrossRef]
19. Benchrif, A.; Wheida, A.; Tahri, M.; Shubbar, R.M.; Biswas, B. Air quality during three COVID-19 lockdown phases: AQI, $PM_{2.5}$ and NO_2 assessment in cities with more than 1 million inhabitants. *Sustain. Cities Soc.* **2021**, *74*, 103170. [CrossRef]
20. Keller, C.A.; Evans, M.J.; Knowland, K.E.; Hasenkopf, C.A.; Modekurty, S.; Lucchesi, R.A.; Oda, T.; Franca, B.B.; Mandarino, F.C.; Díaz Suárez, M.V.; et al. Global impact of COVID-19 restrictions on the surface concentrations of nitrogen dioxide and ozone. *Atmos. Chem. Phys.* **2021**, *21*, 3555–3592. [CrossRef]
21. Jacob, D.; Stowe, S.; Babarinde, I.; Sharma, A.; Christopher, A.; Vilcassim, M.J.R. The Impact of COVID-19 Related Changes on Air Quality in Birmingham, Alabama, United States. *Int. J. Environ. Res. Public Health* **2022**, *19*, 3168. [CrossRef]
22. Rosca, C.M. Convergence Catalysts: Exploring the Fusion of Embedded Systems, IoT, and Artificial Intelligence. In *Engineering Applications of AI and Swarm Intelligence*; Yang, X.-S., Ed.; Springer Nature: Singapore, 2025; pp. 69–87. [CrossRef]
23. Rosca, C.M.; Gortoescu, I.A.; Tanase, M.R. Artificial Intelligence—Powered Video Content Generation Tools. *Rom. J. Pet. Gas Technol.* **2024**, *V (LXXVI)*, 131–144. [CrossRef]
24. Waseem, K.H.; Mushtaq, H.; Abid, F.; Abu-Mahfouz, A.M.; Shaikh, A.; Turan, M.; Rasheed, J. Forecasting of Air Quality Using an Optimized Recurrent Neural Network. *Processes* **2022**, *10*, 2117. [CrossRef]
25. Tao, Q.; Liu, F.; Li, Y.; Sidorov, D. Air Pollution Forecasting Using a Deep Learning Model Based on 1D Convnets and Bidirectional GRU. *IEEE Access* **2019**, *7*, 76690–76698. [CrossRef]
26. Zhang, T.; Dick, R.P. Image-Based Air Quality Forecasting Through Multi-Level Attention. In Proceedings of the IEEE International Conference on Image Processing, Bordeaux, France, 16–19 October 2022; pp. 686–690. [CrossRef]
27. Murad, A.; Kraemer, F.A.; Bach, K.; Taylor, G. Probabilistic Deep Learning to Quantify Uncertainty in Air Quality Forecasting. *Sensors* **2021**, *21*, 8009. [CrossRef]
28. Kodak, G. An Investigation on the Use of Air Quality Models in Ship Emission Forecasts. *J. Intell. Transp. Syst. Appl.* **2024**, *7*, 15–30. [CrossRef]
29. Weissert, L.F.; Alberti, K.; Miskell, G.; Pattinson, W.; Salmond, J.A.; Henshaw, G.; Williams, D.E. Low-cost sensors and microscale land use regression: Data fusion to resolve air quality variations with high spatial and temporal resolution. *Atmos. Environ.* **2019**, *213*, 285–295. [CrossRef]
30. Wang, D.; Wang, H.-W.; Lu, K.-F.; Peng, Z.-R.; Zhao, J. Regional Prediction of Ozone and Fine Particulate Matter Using Diffusion Convolutional Recurrent Neural Network. *Int. J. Environ. Res. Public Health* **2022**, *19*, 3988. [CrossRef]
31. Rosca, C.-M.; Stancu, A. Earthquake Prediction and Alert System Using IoT Infrastructure and Cloud-Based Environmental Data Analysis. *Appl. Sci.* **2024**, *14*, 10169. [CrossRef]
32. Sekula, P.; Ustrnul, Z.; Bokwa, A.; Bochenek, B.; Zimnoch, M. Random Forests Assessment of the Role of Atmospheric Circulation in PM10 in an Urban Area with Complex Topography. *Sustainability* **2022**, *14*, 3388. [CrossRef]
33. Vu, T.V.; Shi, Z.; Cheng, J.; Zhang, Q.; He, K.; Wang, S.; Harrison, R.M. Assessing the impact of clean air action on air quality trends in Beijing using a machine learning technique. *Atmos. Chem. Phys.* **2019**, *19*, 11303–11314. [CrossRef]
34. Hamami, F.; Dahlan, I.A. Air Quality Classification in Urban Environment using Machine Learning Approach. In Proceedings of the International Conference on Disaster Management and Climate Change, Surakarta, Indonesia, 9 April 2021; p. 012004. [CrossRef]
35. Badrakh, O.; Choimaa, L. Air quality predictions of Ulaanbaatar using machine learning approach. In Proceedings of the International Symposium on Grids & Clouds, Taipei, Taiwan, 22–26 March 2021; pp. 1–7. [CrossRef]
36. Du, S.; Li, T.; Yang, Y.; Horng, S.-J. Deep Air Quality Forecasting Using Hybrid Deep Learning Framework. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 2412–2424. [CrossRef]
37. Huang, C.-J.; Kuo, P.-H. A Deep CNN-LSTM Model for Particulate Matter (PM2.5) Forecasting in Smart Cities. *Sensors* **2018**, *18*, 2220. [CrossRef]
38. Liu, Y.; Wang, P.; Li, Y.; Wen, L.; Deng, X. Air quality prediction models based on meteorological factors and real-time data of industrial waste gas. *Sci. Rep.* **2022**, *12*, 9253. [CrossRef]
39. Akdi, Y.; Gölveren, E.; Ünlü, K.D.; Yücel, M.E. Modeling and forecasting of monthly PM2.5 emission of Paris by periodogram-based time series methodology. *Environ. Monit. Assess.* **2021**, *193*, 622. [CrossRef] [PubMed]

40. Rosca, C.-M.; Stancu, A.; Popescu, M. The Impact of Cloud Versus Local Infrastructure on Automatic IoT-Driven Hydroponic Systems. *Appl. Sci.* **2025**, *15*, 4016. [CrossRef]
41. Liu, Q.; Cui, B.; Liu, Z. Air Quality Class Prediction Using Machine Learning Methods Based on Monitoring Data and Secondary Modeling. *Atmosphere* **2024**, *15*, 553. [CrossRef]
42. Fang, L.; Jin, J.; Segers, A.; Lin, H.X.; Pang, M.; Xiao, C.; Deng, T.; Liao, H. Development of a regional feature selection-based machine learning system (RFSML v1.0) for air pollution forecasting over China. *Geosci. Model Dev.* **2022**, *15*, 7791–7807. [CrossRef]
43. IQAir. Air Quality in The Bronx. Available online: https://www.iqair.com/us/usa/new-york/the-bronx (accessed on 10 April 2025).
44. Rosca, C.M.; Stancu, A.; Ariciu, A.V. Algorithm for child adoption process using artificial intelligence and monitoring system for children. *Internet Things* **2024**, *26*, 101170. [CrossRef]
45. Bajpai, A.; Girish Kumar, T.P.; Sreenivasan, G.; Srivastav, S.K. System Design, Automatic Data Collection Framework and Embedded Software Development of Internet of Things (IoT) for Air Pollution Monitoring of Nagpur Metropolis. *J. Indian Soc. Remote Sens.* **2024**, *52*, 2347–2359. [CrossRef]
46. Martín-Baos, J.Á.; Rodriguez-Benitez, L.; García-Ródenas, R.; Liu, J. IoT based monitoring of air quality and traffic using regression analysis. *Appl. Soft Comput.* **2022**, *115*, 108282. [CrossRef]
47. Shahid, I.; Shahzad, M.I.; Tutsak, E.; Mahfouz, M.M.K.; Al Adba, M.S.; Abbasi, S.A.; Rathore, H.A.; Asif, Z.; Chen, Z. Carbon based sensors for air quality monitoring networks; middle east perspective. *Front. Chem.* **2024**, *12*, 1391409. [CrossRef]
48. Kumar, Y.J.N.; Chandan, R.; Somanini, S.H.; Vadtya, S.; Dangi, D. Ecosense: An IoT System for Detecting Suitable and Sustainable Living Conditions. In Proceedings of the 15th International Conference on Materials Processing and Characterization, Newcastle, UK, 5–8 September 2023; p. 01085. [CrossRef]
49. Marques, G.; Pitarma, R. A Cost-Effective Air Quality Supervision Solution for Enhanced Living Environments through the Internet of Things. *Electronics* **2019**, *8*, 170. [CrossRef]
50. Jo, J.; Jo, B.; Kim, J.; Kim, S.; Han, W. Development of an IoT-Based Indoor Air Quality Monitoring Platform. *J. Sens.* **2020**, *2020*, 749764. [CrossRef]
51. Rosca, C.-M.; Stancu, A. A Comprehensive Review of Machine Learning Models for Optimizing Wind Power Processes. *Appl. Sci.* **2025**, *15*, 3758. [CrossRef]
52. Shrestha, R.; Maharjan, M.; Sharma, M. Air Pollution Monitoring System Using Micro Controller Atmega 32A and MQ135 Gas Sensor at Chandragiri Municipality of Kathmandu City. *Semicond. Sci. Inf. Devices* **2022**, *4*, 35–44. [CrossRef]
53. Zheng, T.; Bergin, M.H.; Johnson, K.K.; Tripathi, S.N.; Shirodkar, S.; Landis, M.S.; Sutaria, R.; Carlson, D.E. Field evaluation of low-cost particulate matter sensors in high- and low-concentration environments. *Atmos. Meas. Tech.* **2018**, *11*, 4823–4846. [CrossRef]
54. Cowell, N.; Chapman, L.; Bloss, W.; Srivastava, D.; Bartington, S.; Singh, A. Particulate matter in a lockdown home: Evaluation, calibration, results and health risk from an IoT enabled low-cost sensor network for residential air quality monitoring. *Environ. Sci. Atmos.* **2023**, *3*, 65–84. [CrossRef]
55. Rosca, C.-M.; Stancu, A.; Tănase, M.R. A Comparative Study of Azure Custom Vision Versus Google Vision API Integrated into AI Custom Models Using Object Classification for Residential Waste. *Appl. Sci.* **2025**, *15*, 3869. [CrossRef]
56. Tooki, O.O.; Tamasi, M.A.; Ohemu, M.F.; Ogunkeyede, O.Y.; Abolade, R.O. Implementation of a sustainable real-time air quality monitoring system using the Internet of Things for Kaduna metropolis, Nigeria. *LAUTECH J. Eng. Technol.* **2024**, *18*, 122–127. [CrossRef]
57. Phala, K.S.E.; Kumar, A.; Hancke, G.P. Air Quality Monitoring System Based on ISO/IEC/IEEE 21451 Standards. *IEEE Sens. J.* **2016**, *16*, 5037–5045. [CrossRef]
58. Gu, K.; Qiao, J.; Lin, W. Recurrent Air Quality Predictor Based on Meteorology- and Pollution-Related Factors. *IEEE Trans. Ind. Inform.* **2018**, *14*, 3946–3955. [CrossRef]
59. Motlagh, N.H.; Lagerspetz, E.; Nurmi, P.; Li, X.; Varjonen, S.; Mineraud, J.; Siekkinen, M.; Rebeiro-Hargrave, A.; Hussein, T.; Petaja, T.; et al. Toward Massive Scale Air Quality Monitoring. *IEEE Commun. Mag.* **2020**, *58*, 54–59. [CrossRef]
60. Zhang, D.; Woo, S.S. Real Time Localized Air Quality Monitoring and Prediction Through Mobile and Fixed IoT Sensing Network. *IEEE Access* **2020**, *8*, 89584–89594. [CrossRef]
61. Rosca, C.-M.; Rădulescu, G.; Stancu, A. Artificial Intelligence of Things Infrastructure for Quality Control in Cast Manufacturing Environments Shedding Light on Industry Changes. *Appl. Sci.* **2025**, *15*, 2068. [CrossRef]
62. Hawari, H.F.; Zainal, A.A.; Ahmad, M.R. Development of real time internet of things (IoT) based air quality monitoring system. *Indones. J. Electr. Eng. Comput. Sci.* **2019**, *13*, 1039. [CrossRef]
63. Sunarno; Purwanto; Suryono. The Monitoring System of Sulfur Dioxide Gas Using a Web-based Wireless Sensor. In Proceedings of the 13th International Interdisciplinary Studies Seminar, Malang, Indonesia, 30–31 October 2019. [CrossRef]
64. DiNapoli, T.P.; Bleiwas, K.B. *An Economic Snapshot of the Bronx*; Office of the New York State Comptroller: New York, NY, USA, 2019. Available online: https://www.osc.ny.gov/files/reports/osdc/pdf/report-4-2019.pdf (accessed on 8 October 2024).

65. State of New York. Transportation. 2024. Available online: https://data.ny.gov/browse?category=Transportation&sortBy=relevance&page=1&pageSize=20 (accessed on 9 October 2024).
66. Open-Meteo. Open-Meteo API. Available online: https://archive-api.open-meteo.com/v1/era5?latitude=40.8527&longitude=73.8770&start_date=2020-01-01&end_date=2023-01-31&hourly=temperature_2m,relative_humidity_2m,wind_speed_10m (accessed on 10 October 2024).
67. U.S. Environmental Protection Agency. Outdoor Air Quality Data. 2024. Available online: https://www.epa.gov/outdoor-air-quality-data/download-daily-data (accessed on 12 October 2024).
68. U.S. Environmental Protection Agency. Interactive Map of Air Quality. Available online: https://gispub.epa.gov/airnow/?contours=none&monitors=ozonepm&basemap=basemap_1&xmin=-8234563.212844915&xmax=-8216294.763084781&ymin=4988514.852448748&ymax=4996426.084876254 (accessed on 10 April 2025).
69. Gangwar, A.; Singh, S.; Mishra, R.; Prakash, S. The State-of-the-Art in Air Pollution Monitoring and Forecasting Systems Using IoT, Big Data, and Machine Learning. *Wirel. Pers. Commun.* **2023**, *130*, 1699–1729. [CrossRef]
70. American Lung Association. Air Quality Index. Available online: https://www.lung.org/clean-air/outdoors/air-quality-index (accessed on 20 October 2024).
71. Roșca, C.-M.; Cărbureanu, M. A Comparative Analysis of Sorting Algorithms for Large-Scale Data: Performance Metrics and Language Efficiency. In Proceedings of the Emerging Trends and Technologies on Intelligent Systems. ETTIS 2024. Lecture Notes in Networks and Systems, Noida, India, 27–28 March 2024; pp. 99–113. [CrossRef]
72. Belgiu, M.; Drăguţ, L. Random forest in remote sensing: A review of applications and future directions. *ISPRS J. Photogramm. Remote Sens.* **2016**, *114*, 24–31. [CrossRef]
73. Sammut, C.; Webb, G.I. *Encyclopedia of Machine Learning and Data Mining*, 2nd ed.; Springer: New York, NY, USA, 2017. [CrossRef]
74. Ashoor Mahani, E.; Ziarati, K. A Consolidated Tree Structure Combining Multiple Regression Trees with Varying Depths, Resulting in an Efficient Ensemble Model. *Informatica* **2023**, *47*, 17–34. [CrossRef]
75. Wood, S.N. *Generalized Additive Models*, 2nd ed.; Chapman and Hall/CRC: New York, NY, USA, 2017. [CrossRef]
76. Alamgir, M.S.M.; Sultana, M.N.; Chang, K. Link Adaptation on an Underwater Communications Network Using Machine Learning Algorithms: Boosted Regression Tree Approach. *IEEE Access* **2020**, *8*, 73957–73971. [CrossRef]
77. Cocchi, G.; Galli, L.; Galvan, G.; Sciandrone, M.; Cantù, M.; Tomaselli, G. Machine learning methods for short-term bid forecasting in the renewable energy market: A case study in Italy. *Wind Energy* **2018**, *21*, 357–371. [CrossRef]
78. Elith, J.; Leathwick, J.R.; Hastie, T. A working guide to boosted regression trees. *J. Anim. Ecol.* **2008**, *77*, 802–813. [CrossRef]
79. Microsoft. Microsoft.ML.Trainers.FastTree Namespace. Available online: https://learn.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.fasttree?view=ml-dotnet (accessed on 4 November 2024).
80. Rosca, C.-M.; Stancu, A. Fusing Machine Learning and AI to Create a Framework for Employee Well-Being in the Era of Industry 5.0. *Appl. Sci.* **2024**, *14*, 10835. [CrossRef]
81. Zhou, M.; Chen, C.; Peng, J.; Luo, C.-H.; Feng, D.Y.; Yang, H.; Xie, X.; Zhou, Y. Fast Prediction of Deterioration and Death Risk in Patients With Acute Exacerbation of Chronic Obstructive Pulmonary Disease Using Vital Signs and Admission History: Retrospective Cohort Study. *JMIR Med. Inform.* **2019**, *7*, e13085. [CrossRef]
82. Phillips, N.D.; Neth, H.; Woike, J.K.; Gaissmaier, W. FFTrees: A toolbox to create, visualize, and evaluate fast-and-frugal decision trees. *Judgm. Decis. Mak.* **2017**, *12*, 344–368. [CrossRef]
83. Zhang, H.; Yang, Q.; Shao, J.; Wang, G. Dynamic Streamflow Simulation via Online Gradient-Boosted Regression Tree. *J. Hydrol. Eng.* **2019**, *24*, 04019041. [CrossRef]
84. Rosca, C.-M. New Algorithm to Prevent Online Test Fraud Based on Cognitive Services and Input Devices Events. In *Proceedings of Third Emerging Trends and Technologies on Intelligent Systems. ETTIS 2023. Lecture Notes in Networks and Systems*; Noor, A., Saroha, K., Pricop, E., Sen, A., Trivedi, G., Eds.; Springer Nature: Singapore, 2023; Volume 730, pp. 207–219. [CrossRef]
85. Sun, F.; He, M.; Gao, Q. Double-paralleled ridgelet neural network with IFPSO training algorithm. In Proceedings of the International Conference on Electrical and Control Engineering, Yichang, China, 16–18 September 2011; pp. 4468–4471. [CrossRef]
86. Banerjee, I.; Ling, Y.; Chen, M.C.; Hasan, S.A.; Langlotz, C.P.; Moradzadeh, N.; Chapman, B.; Amrhein, T.; Mong, D.; Rubin, D.L.; et al. Comparative effectiveness of convolutional neural network (CNN) and recurrent neural network (RNN) architectures for radiology text report classification. *Artif. Intell. Med.* **2019**, *97*, 79–88. [CrossRef]
87. Hindarto, D. Comparison of RNN Architectures and Non-RNN Architectures in Sentiment Analysis. *sinkron* **2023**, *8*, 2537–2546. [CrossRef]
88. Ororbia, A.; Elsaid, A.; Desell, T. Investigating recurrent neural network memory structures using neuro-evolution. In Proceedings of the GECCO '19: Genetic and Evolutionary Computation Conference, Prague, Czech Republic, 13–17 July 2019; pp. 446–455. [CrossRef]
89. Shewalkar, A.; Nyavanandi, D.; Ludwig, S.A. Performance Evaluation of Deep Neural Networks Applied to Speech Recognition: RNN, LSTM and GRU. *J. Artif. Intell. Soft Comput. Res.* **2019**, *9*, 235–245. [CrossRef]

90. Gryshmanov, E.; Kalimulin, T.; Zakharchenko, I. Prognosis Method of Unfavorable Airborne Events during Flight Based On Convolutional and Recurrent Neural Networks. *Adv. Inf. Syst.* **2019**, *3*, 104–108. [CrossRef]
91. Puchalski, B. Neural Approximators for Variable-Order Fractional Calculus Operators (VO-FC). *IEEE Access* **2022**, *10*, 7989–8004. [CrossRef]
92. Ji, X.A.; Orosz, G. Trainable Delays in Time Delay Neural Networks for Learning Delayed Dynamics. *IEEE Trans. Neural Netw. Learn. Syst.* **2025**, *36*, 5219–5229. [CrossRef]
93. Zhang, Y.; Liu, F. An Improved Deep Belief Network Prediction Model Based on Knowledge Transfer. *Future Internet* **2020**, *12*, 188. [CrossRef]
94. Rosca, C.-M.; Stancu, A.; Iovanovici, E.M. The New Paradigm of Deepfake Detection at the Text Level. *Appl. Sci.* **2025**, *15*, 2560. [CrossRef]
95. Ayilu, J.; Dhivya, S.; Saravanane, M. Performance Analysis of Machine Learning Singular Spectrum Analysis for Forecasting Air Contamination. In Proceedings of the International Conference on System, Computation, Automation and Networking, Puducherry, India, 17–18 November 2023; pp. 1–7. [CrossRef]
96. Microsoft. TimeSeriesCatalog.ForecastBySsa Method. Available online: https://learn.microsoft.com/en-us/dotnet/api/microsoft.ml.timeseriescatalog.forecastbyssa?view=ml-dotnet (accessed on 2 November 2024).
97. Panaite, F.A.; Rus, C.; Leba, M.; Ionica, A.C.; Windisch, M. Enhancing Air-Quality Predictions on University Campuses: A Machine-Learning Approach to PM2.5 Forecasting at the University of Petroșani. *Sustainability* **2024**, *16*, 7854. [CrossRef]
98. Chan, K.; Matthews, P.; Munir, K. Time Series Forecasting for Air Quality with Structured and Unstructured Data Using Artificial Neural Networks. *Atmosphere* **2025**, *16*, 320. [CrossRef]
99. Cican, G.; Buturache, A.-N.; Mirea, R. Applying Machine Learning Techniques in Air Quality Prediction—A Bucharest City Case Study. *Sustainability* **2023**, *15*, 8445. [CrossRef]