sustainability

MDPI

Article

Effective Air Quality Prediction Using Reinforced Swarm Optimization and Bi-Directional Gated Recurrent Unit

Sasikumar Gurumoorthy [1], Aruna Kumari Kokku [2], Przemysław Falkowski-Gilski [3,*] and Parameshachari Bidare Divakarachari [4,*]

[1] Department of Computer Science and Engineering, J. J. College of Engineering and Technology, Trichy 620009, India; sasikumarg@jjcet.ac.in
[2] Department of Computer Science and Engineering, SRKR Engineering College, Chinaamiram, Bhimavaram 534204, India; arunakumarisatti@srkrec.ac.in
[3] Faculty of Electronics, Telecommunications and Informatics, Gdansk University of Technology, Narutowicza 11/12, 80-233 Gdansk, Poland
[4] Department of Electronics and Communication Engineering, Nitte Meenakshi Institute of Technology, Bengaluru 560064, India
* Correspondence: przemyslaw.falkowski@eti.pg.edu.pl (P.F.-G.); paramesh@nmit.ac.in (P.B.D.)

Abstract: In the present scenario, air quality prediction (AQP) is a complex task due to high variability, volatility, and dynamic nature in space and time of particulates and pollutants. Recently, several nations have had poor air quality due to the high emission of particulate matter ($PM_{2.5}$) that affects human health conditions, especially in urban areas. In this research, a new optimization-based regression model was implemented for effective forecasting of air pollution. Firstly, the input data were acquired from a real-time Beijing $PM_{2.5}$ dataset recorded from 1 January 2010 to 31 December 2014. Additionally, the newer real-time dataset was recorded from 2016 to 2022 for four Indian cities: Cochin, Hyderabad, Chennai, and Bangalore. Then, data normalization was accomplished using the Min-Max normalization technique, along with correlation analysis for selecting highly correlated variables (wind direction, temperature, dew point, wind speed, and historical $PM_{2.5}$). Next, the important features from the highly correlated variables were selected by implementing an optimization algorithm named reinforced swarm optimization (RSO). Further, the selected optimal features were given to the bi-directional gated recurrent unit (Bi-GRU) model for effective AQP. The extensive numerical analysis shows that the proposed model obtained a mean absolute error (MAE) of 9.11 and 0.19 and a mean square error (MSE) of 2.82 and 0.26 on the Beijing $PM_{2.5}$ dataset and a real-time dataset. On both datasets, the error rate of the proposed model was minimal compared to other regression models.

Keywords: air quality prediction; bi-directional gated recurrent unit; correlation analysis; Min-Max normalization technique; reinforced swarm optimization algorithm

Citation: Gurumoorthy, S.; Kokku, A.K.; Falkowski-Gilski, P.; Divakarachari, P.B. Effective Air Quality Prediction Using Reinforced Swarm Optimization and Bi-Directional Gated Recurrent Unit. *Sustainability* **2023**, *15*, 11454. https://doi.org/10.3390/su151411454

Academic Editor: Ali Elkamel

Received: 28 March 2023
Revised: 13 June 2023
Accepted: 21 July 2023
Published: 24 July 2023

1. Introduction

In recent decades, air pollution has been a serious environmental issue, and several developed and developing countries have suffered from heavy air pollution [1]. The identification of atypical pollution in the quantified concentrations of these compounds has been a significant problem for health [2]. Compared to other pollution, air pollution has a direct impact on people's health, and the major causes of air pollution are natural disasters, residential heating, exhaust from industries and factories, and the burning of fossil fuels [3,4]. Therefore, predicting the mass concentrations of air pollution is essential and plays a crucial role in atmospheric management decisions [5]. Additionally, existing epidemiological research studies state that $PM_{2.5}$ causes negative human health effects, like respiratory diseases and cardiovascular diseases [6,7]. Therefore, effective forecasting of air pollutant concentrations strengthen the prevention of air pollution, which helps in

achieving efficient environmental management [8]. In addition, it has great significance for government decision making and people's health [9]. Poor AQP not only affects the human physical condition but also produces a key impact on societal and economic controls [10].

Recently, several research studies have been carried out on AQP, but the majority of the existing studies face difficulty in predicting future air quality for the monitoring stations [11]. In this scenario, AQP is influenced by several factors, like dust, coal burning, industrial emissions, vehicle exhaust, spatial distribution, and time patterns [12]. According to atomic science research, the major factors for the dissipation and accumulation of atmospheric pollutants are weather conditions, regional transport, and local emissions. These factors are categorized into indirect and direct factors based on their impact on air quality [13]. Compared to traditional machine learning models, deep learning models have gained more attention among researchers, especially in time series analysis. Deep learning models are effective in exploring longer-term dependencies and implicit features from time series data for effective AQP. Yet, several deep learning models have problems like overfitting and the vanishing gradient problem in time series forecasting. Therefore, a new optimization-based regression model is proposed in this research in order to overcome the above-stated problems and to achieve better AQP.

The main contributions of our paper are given as follows:

- Initially, this study implemented a Min-Max normalization technique that efficiently preserves the relationship between the data values with low standard deviation. In time series forecasting, the Min-Max normalization technique forecast the next hour's concentration and reduces the effect of outliers by using different sizes of sliding windows.
- Then, we performed a correlation analysis to select the optimal meteorological variables (wind speed, temperature, dew point, wind direction, and historical $PM_{2.5}$) from the collected datasets.
- After that, an RSO algorithm was developed for selecting discriminative features from the selected meteorological variables. This action greatly reduces the model's complexity and computational time. The RSO algorithm is the integration of the BSO algorithm and reinforcement learning, which overcomes the optimization problems like poor convergence rate and local optima problems.
- Finally, the Bi-GRU model was used for effective forecasting of air quality, and its efficacy was tested using the performance measures, including *MSE*, *RMSE*, *MAE*, symmetric mean absolute percentage error (SMAPE), MAPE, and coefficient of determination (R^2).

The manuscript is organized as follows: The existing papers on the topic of AQP are reviewed in Section 2. The methodology details, numerical investigation, and conclusion of this research are mentioned in Sections 3–5, respectively.

2. Literature Review

For predicting the air quality in Tripoli [14], Esager and Ünlü proposed an evaluation of deep learning models for hourly $PM_{2.5}$ surface mass concentrations. Since the analyzed data are a time series, the Box–Jenkins methodology is generally used to model such a dataset. This study gave particular attention to the LSTM and GRU with CNN types of recurrent neural networks. The result analysis demonstrates the strong forecasting power of the used algorithms. This type of model's key benefit is that it does not call for the same exact assumptions that other traditional models do. These algorithms were also quite effective in simulating the data's nonlinear behavior.

Du et al. [15] implemented a hybrid deep learning architecture for effective air pollution forecasting. The implemented hybrid architecture, Bi-LSTM and a convolutional neural network (CNN), learns multivariate, temporal, and spatial correlation features from the collected time series data for effective forecasting of air quality. The experiments conducted on the two real-world datasets demonstrated that the implemented hybrid architecture was effective in dealing with $PM_{2.5}$ air pollution prediction with better accuracy. The integration

of deep learning models increased the time complexity and computational cost because it required an enormous amount of data to obtain satisfactory results.

Usually, the dynamics of air pollution is reflected by dissimilar factors, like rainfall, snowfall, wind speed, wind direction, humidity, and temperature. These factors increase the difficulty in understanding the changes that occurred in the air pollutant concentration. Tao et al. [16] integrated the CNN and Bi-GRU models for effective forecasting of air pollution. The experiments conducted on the UCI machine-learning repository Beijing $PM_{2.5}$ dataset demonstrated the effectiveness of the hybrid deep learning models, as they achieved better results than traditional models. As mentioned earlier, the integration of two deep learning models leads to high time complexity.

Ma et al. [17] used a Bi-LSTM network with transfer learning for forecasting air pollution in Anhui, China. The numerical results showed that the Bi-LSTM network with transfer learning achieved a 35% lower error rate than the existing models on a real-time dataset. The developed Bi-LSTM network with transfer learning was not scalable and was time-consuming while performing experiments on a real-time dataset.

Chang et al. [18] implemented a new aggregated LSTM network for effective air pollution forecasting. The aggregated LSTM network combines information about external pollution sources, stations nearby industrial areas, and the stations with local air pollution monitoring systems. Here, three LSTM models were aggregated in order to improve prediction accuracy, but it was a computationally complex process.

Castelli et al. [19] employed a machine learning technique called support vector regression (SVR) for forecasting air quality index (AQI) and pollutant levels. After the acquisition of time series data, data preprocessing (data transformation, outlier removal, and imputation of missing data) and feature engineering were accomplished. Finally, the air pollution prediction was carried out by utilizing the SVR technique. However, the SVR will underperform when the number of feature vectors for every data point exceeds the number of training samples.

Xayasouk et al. [20] integrated a deep autoencoder and an LSTM network for air pollution prediction. In addition to this, Wen et al. [21] combined a CNN and an LSTM network for effective forecasting of air pollution in China. Wang et al. [22] implemented a two-layer air pollution prediction model based on a GRU and an LSTM network. The numerical outcomes confirmed that the presented hybrid models obtained higher prediction performance than existing ones at different regional scales. The hybrid deep learning model has the ability to handle complex and large data, but it was computationally expensive.

Air pollution is becoming a serious problem due to the rapid growth of industrialization. In the present scenario, predicting air pollution is crucial in determining prevention measures for avoiding disasters. Zhang et al. [23] utilized a light gradient boosting technique for selecting discriminative features from real-time datasets. Further, the selected 500 feature vectors were given to the eXtreme Gradient Boosting (XGBoost) technique for air pollution forecasting.

Wang et al. [24] initially adopted the Hampel identifier and variational mode decomposition (VMD) technique for detecting and eliminating outliers from the acquired datasets. Then, the optimal feature vectors were selected from the denoised data by employing a sine-cosine algorithm, and finally, an extreme learning machine (ELM) was implemented for accurate forecasting of air pollution. Generally, standard machine learning techniques, such as XGBoost and ELM, exhibit outliers and overfitting problems when analyzing complex time series data.

The PM of the Turkish city Ankara was modeled using a hybrid deep learning methodology, which was analyzed by Akbal and Ünlü [25]. According to the WHO's criteria, PM levels were categorized to provide a prediction problem. Further, by using the ensemble machine learning methodology of random forest regression (RFR), extra tree regression (ETR), and multiple linear regression (MLR), the impact of various contaminants and meteorological variables on the prediction of PM has been examined. The findings indicated

that other substances, the Earth's surface temperature, wind speed, and PM's own lagged values were the most crucial predictor variables for PM.

Li et al. [26] employed the Hampel filter and least square support vector machine (SVM) regression for AQI forecasting. Maleki et al. [27] implemented an artificial neural network (ANN) for air pollution forecasting. However, the ANN was a simpler deep learning mode and required more training data to obtain satisfactory results. Mao et al. [28] implemented a temporal sliding LSTM network for effective prediction of air quality. The presented temporal sliding LSTM network achieved higher prediction results with strong atmospheric decision making.

Zhang et al. [29] integrated empirical mode decomposition (EMD) and a Bi-LSTM network for effective forecasting of AQI. Firstly, the EMD technique was employed for decomposing $PM_{2.5}$ time series data and extracting the amplitude and frequency features. Secondly, the obtained features were given to the Bi-LSTM network for AQI forecasting. The experiments conducted on the $PM_{2.5}$ and Beijing hourly datasets demonstrated the efficacy of the developed EMD-Bi-LSTM model by means of error rate. In the time series analysis, the Bi-LSTM network was slower and consumed more time for model training.

Zeinalnezhad et al. [30] integrated an adaptive neuro-fuzzy inference system (ANFIS) and semi-experimental nonlinear regression for predicting the concentration of important pollutants. However, the standard ANFIS models include a few problems, such as the curse of dimensionality, high computational expense, and loss of data interpretability.

Aarthi et al. [31] initially used a Min-Max normalization technique for filling in the missing attributes in the collected dataset, and then, the optimal attributes were selected from the preprocessed data by implementing a balanced spider monkey optimization (BSMO) algorithm. Based on the balancing factor, the BSMO algorithm selects the relevant attributes, which are given to the Bi-LSTM network for AQP. The developed BSMO algorithm efficiently finds the optimal solution but has a poor convergence rate. To highlight the aforementioned concerns and to achieve precise AQP, an effective optimization-based regression model (RSO and Bi-GRU) is introduced in this paper.

Several models have been examined to improve air quality, which is essential for preventing or reducing the consequences of pollution. We will be prompted by the air quality to be careful, and it may even motivate individuals to carry out their daily activities in less polluted areas. However, it is still challenging to analyze the data and provide improved outcomes. Air pollution forecasting is among the fields in which deep learning technologies have a substantial impact and penetration rise. The authors use complex and advanced methods to accurately anticipate the air quality. External factors, such as weather, geographic features, and temporal characteristics, must be taken into account. For pollution reduction, human health monitoring, and sustainability, an accurate air quality prediction model is necessary. Due to overfitting in the prediction model and local optima trap in feature selection, the current air quality forecast models (state-of-the-art methods) are inefficient.

3. Methodology

Eight pollutants, namely particulate matter (PM) 10, $PM_{2.5}$, ozone (O3), sulfur dioxide (SO2), nitrogen dioxide (NO2), carbon monoxide (CO), lead (Pb), and ammonia (NH3), act as major parameters in deriving the AQI of an area. While using the annual data, this research uses 24 lags in time series analysis. In this time series analysis, the proposed framework includes five phases:

(1) Dataset description—Beijing $PM_{2.5}$ dataset and a real-time dataset;
(2) Data normalization—Min-Max normalization technique;
(3) Correlation analysis;
(4) Feature optimization—RSO algorithm;
(5) Prediction—Bi-GRU model.

The diagram of the developed regression model is shown in Figure 1.

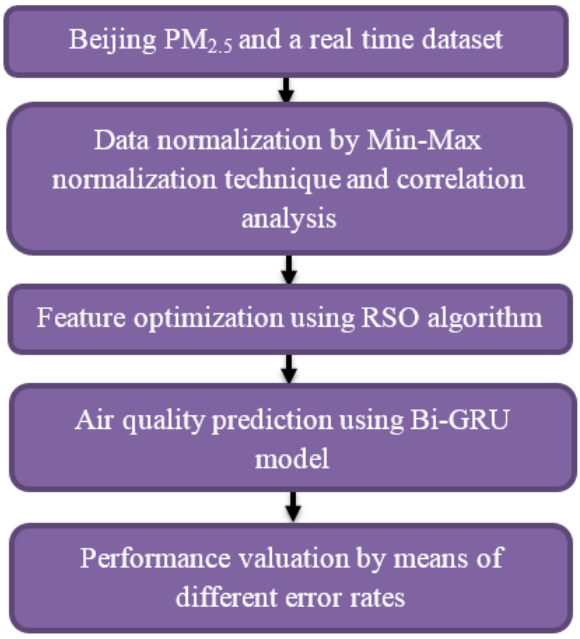

Figure 1. Schematic diagram of the developed regression model.

3.1. Dataset Description

The introduced optimization-based regression model's (RSO and Bi-GRU) performance was validated on a Beijing $PM_{2.5}$ dataset and a newer real-time dataset. The Beijing $PM_{2.5}$ dataset comprised $PM_{2.5}$ meteorological data, which were recorded from 1 January 2010 to 31 December 2014 [16]. Here, in this dataset, 70% of data are used for training, and the remaining 30% are used for testing. From this ratio (70:30), until 2 July 2013, the data has been trained. Then, the testing process started and lasted until 31 December 2014. This dataset has eight characteristics: wind speed, rainfall, wind direction, snowfall, dew point, $PM_{2.5}$ concentration, air pressure, and temperature. Among 43,800 rows, 30,000 rows were utilized as a training set, 8000 rows were utilized as a validation set, and the remaining 5800 rows were utilized as a testing set. In this dataset, the wind direction had four features (southwest, northwest, southeast, and northeast), which were encoded as float data (−10, 0, 10, and 20) [32].

Additionally, a newer real-time dataset was acquired from the central pollution control board for four Indian cities: Cochin, Hyderabad, Chennai, and Bangalore. In this collected dataset (two times a week during a 24 h time period), the pollutants were monitored, and 104 observations were provided annually [31].

3.2. Data Normalization

The acquired time series data were normalized by implementing a Min-Max normalization technique. This helps in removing the units in the acquired data or the impact of differing scales [33,34]. The Min-Max normalization technique is used for scaling the data values within a fixed range (zero to one). Initially, the Min-Max normalization technique subtracts the minimum value from data points X and further divides by its range. The formula of the Min-Max normalization technique X_norm is presented in Equation (1).

$$X\_norm = (X - X\_min)/(X\_max - X\_min) \quad (1)$$

In this scenario, the calculation of the normalization is performed only for the training set, and the validation set and the testing set are unknown. The actual $PM_{2.5}$ concentration in the test set is shown in Figure 2.

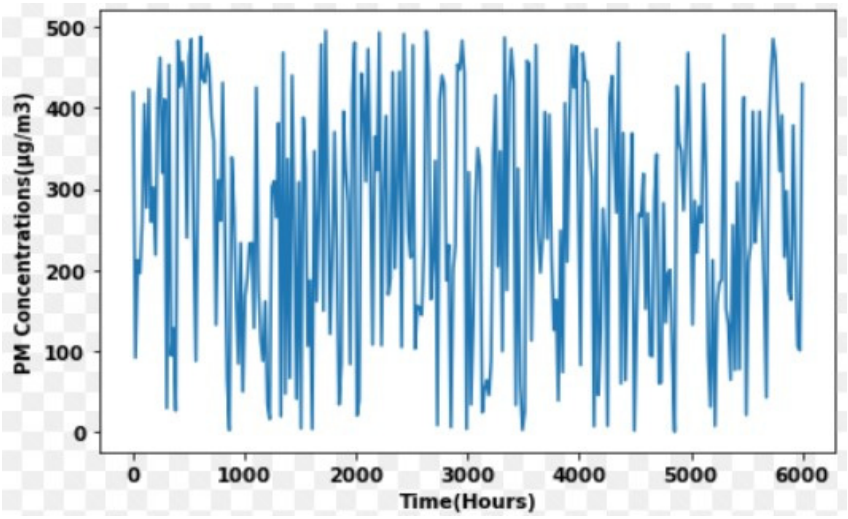

Figure 2. Actual $PM_{2.5}$ concentration in the test set.

3.3. Correlation Analysis

The correlation between all potential pairs of values in a table is shown in the matrix. It is an effective tool for compiling a sizable dataset and for locating and displaying data patterns. A correlation matrix simplifies the process of selecting different assets by tabulating their correlation with one another. It is vital to identify the correlations between PM concentrations and influencing factors for developing a good prediction model. It guarantees that the proposed regression model utilizes the efficient features for AQP. $PM_{2.5}$ is affected by several factors, but all the factors are important in effective AQP. On the other hand, the irrelevant/inactive factors affect the proposed model's performance by means of time complexity. Therefore, it is important to compute the correlation coefficients (CCs) for every factor that helps in selecting the optimal features for effective forecasting of air pollution. Let us consider characteristic time series data as $X = (x_1, x_2, \ldots, x_n)$ and other data as $Y = (y_1, y_2, \ldots, y_n)$. The CC between the factors r is computed as described in Equation (2).

$$r = \frac{n\sum_{i=1}^{n} x_i y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n\sum_{i=1}^{n} x_i^2 - \left(\sum_{i=1}^{n} x_i\right)^2}\sqrt{n\sum_{i=1}^{n} y_i^2 - \left(\sum_{i=1}^{n} y_i\right)^2}} \quad (2)$$

where $0 < r < 1$ indicates a positive correlation, $-1 < r < 0$ represents a negative correlation, and n represents the number of samples. The correlation is greater and the space between X and Y is limited if the absolute value of r is closer to 1.

The CCs between PM concentrations and every feature were calculated for the Beijing $PM_{2.5}$ dataset and a real-time dataset. Table 1 shows that the snowfall, wind direction, and dew point have positive correlations with $PM_{2.5}$ concentration, whereas the wind speed, temperature, rainfall, and air pressure have negative correlations with $PM_{2.5}$ concentration. Table 1 clearly shows that all variables are weakly correlated with each other, and it shows that there are no duplicate variables. The obtained meteorological variables are directly utilized as the input of the proposed optimization-based regression model. As specified in Table 1, the CCs of rainfall, snowfall, and air pressure are small, and the unrelated input increases the difficulty in learning useful features and the model's complexity. Therefore, the wind direction, temperature, dew point, wind speed, and historical $PM_{2.5}$ were chosen as the input for the proposed optimization-based regression model.

Table 1. CCs between PM concentrations and meteorological variables.

R	Rain	Snow	Wind Speed	Wind Direction	Pressure	Temperature	Dew Point	Pollution
Rain	1.00	−0.01	−0.01	−0.05	−0.07	0.05	0.13	−0.05
Snow	−0.01	1.00	0.02	0.01	0.07	−0.09	−0.03	0.02
Wind speed	−0.01	0.02	1.00	−0.20	0.18	−0.15	−0.30	−0.24
Wind direction	−0.05	0.01	−0.20	1.00	−0.16	0.18	0.23	0.19
Pressure	−0.07	0.07	0.18	−0.16	1.00	−0.79	−0.74	−0.06
Temperature	0.05	−0.09	−0.15	0.18	−0.79	1.00	0.82	−0.09
Dew point	0.13	−0.03	−0.30	0.23	−0.74	0.82	1.00	0.18
Pollution	−0.05	0.02	−0.24	0.19	−0.06	−0.09	0.18	1.00

3.4. Feature Optimization

From the selected variables, namely wind direction, temperature, dew point, wind speed, and historical $PM_{2.5}$, the important features were chosen by implementing the RSO algorithm, which is a combination of the bee swarm optimization (BSO) algorithm and reinforcement learning.

3.4.1. BSO Algorithm

The BSO algorithm [35] is one of the effective metaheuristic feature selection algorithms; it mimics the hierarchical task management, adaptation, and self-organization behavior of natural bees. The BSO is an iterative algorithm that resolves optimization problems by imitating the probabilistic decision-making mechanism and foraging behavior of bees for exploiting and selecting optimal food sources. First, the heuristic is utilized for generating the reference solution, which is considered as the reference for determining other solutions in the search space. In the BSO algorithm, the search space is defined as the distance that is inversely proportional to the flip parameter that helps in finding the convergence in the search process. In the local search, a bee agent is assigned to each of these solutions. Every bee's search result is saved to the dance table when it is completed. One of the solutions is picked to serve as the new reference solution in the following iteration. In order to avoid cycles, the reference solutions are kept in the dance table. From the dance table, the fittest and best solutions are passed to the congeners, which are further utilized for selecting the next reference solution.

In order to avoid congestion problems, the selected reference solutions are placed in a table "Tab". Then, a parameter (Chance-Max) is defined for avoiding the local optima problem. In the BSO algorithm, maximum chances are given to a bee agent in order to explore a reference solution. In the next step, intensification is performed if a better reference solution is found within the Chance-Max range; otherwise, diversification is carried out. The search stops after identifying the global optimal best solution or reaching the maximum number of iterations.

3.4.2. Reinforcement Learning

Reinforcement learning [36] tackles the issue of autonomous entities needing to learn control techniques with little or no data. It strengthens or reinforces the behavior. Since positive reinforcement does not require taking something away or imposing a negative consequence, people frequently find it simpler to accept than other teaching techniques. Additionally, it is far simpler to reward behaviors than to penalize them, which makes reinforcement generally a more effective tool. In machine learning, reinforced learning is called Q learning, and how a specific task is achieved is defined as a programming agent. In this scenario, $U = \{u_1, u_2, \ldots, u_n\}$ represents the set of states, and $V = \{v_1, v_2, \ldots, v_n\}$ specifies the set of actions. For each action a_i, a reward b_i is received, and this is performed in a set s_i. This algorithm maps $U \rightarrow V$ for maximizing the reward function, and it is mathematically specified in Equation (3).

$$B_{u,v}(i) = b_i + \alpha b_{i+1} + \alpha^2 b_{i+2} + \cdots \alpha^2 b_{i+n} \quad (3)$$

where the discount parameter is defined as α, which ranges between 0 and 1. Generally, the search agents tend towards the longer-term rewards when α is equal to 1. On the other hand, the search agents tend towards the immediate or shorter-term rewards when α is equal to 0. In residual learning, the temporal difference is an extensively utilized method that integrates the features of the Markov decision process and the Monte Carlo algorithm. In this scenario, the temporal difference method is used in the recursive Q learning for computing the immediate reward Q, and it is mathematically described in Equation (4).

$$Q(u_i, v_i) = b(u_i, v_i) + \alpha max Q(\beta(u_i, v_i), v_t) \tag{4}$$

where $\beta(u_i, v_i)$ indicates the resulting state and v_t represents another t^{th} action. Therefore, after modifying Equation (4), we obtain a new formula, expressed as Equation (5).

$$Q(u_i, v_i) = lr \times (b(u_i, v_i)) + (1 - lr) \times Q(u_i, v_i) + \alpha max Q(\beta(u_i, v_i), v_t) \tag{5}$$

where $lr \in [0, 1]$ and is represented as the learning rate. The pseudocode (Algorithm 1) of the reinforcement learning process is as follows:

Algorithm 1 Reinforcement Learning

1. Initialize table elements $Q(u, v) \rightarrow 0$
2. Initialize actions $V_i \; \forall_i = 1, 2, 3, \ldots n$
3. Initialize states $U_i \; \forall_i = 1, 2, 3, \ldots n$
4. **For** $k \leq n$ do
5. Presents state $\rightarrow u_k$
6. Present action $\rightarrow v_k$
7. Execute v_k over u_k
8. Immediate reward $\rightarrow b_k$ and the new state is obtained from u_t
9. $Q(u_k, v_k) \leftarrow b_k + \alpha \max(u_t, v_t)$
10. $k = k + 1$
11. Update $u_t \rightarrow u_k$
12. **End for**

3.4.3. RSO Algorithm

As specified earlier, the RSO algorithm [37] is the integration of the BSO algorithm and reinforcement learning, and it improves the learning process by making the agents learn from prior experiences. The main issue in the BSO algorithm is the absence of memory or intelligence in the local search, which results in local optima problems. This makes the BSO algorithm ineffective compared to other optimization algorithms. In order to highlight the aforementioned issue, the local search algorithm is replaced by Q learning.

In the context of feature selection, the deletion and inclusion of a feature from the optimal features is assumed as an action, the reward is considered as the selection of optimal features, and the improvement of AQP is considered as a secondary constraint. Let us assume $V_t = \{v_{t1}, v_{t2}, \ldots, v_{tn}\}$ is an action performed in the t^{th} iteration. The reward obtained in the set u_t leverages the prediction accuracy acc and the number of selected features Num in the feature subsets (selected variables). The reward is mathematically specified in Equation (6).

$$b_t \leftarrow \left\{ \begin{array}{c} Acc(u_t), if \; Acc(u_t) < Acc(u_{t+1}) \\ Acc(u_{t+1}) - Acc(u_t), if \; Acc(u_t) > Acc(u_{t+1}) \\ \frac{Acc(u_t)}{2}, if \; Num(u_t) > Num(u_{t+1}) \\ -\frac{Acc(u_t)}{2}, if \; Num(u_t) < Num(u_{t+1}) \end{array} \right\} \tag{6}$$

In this scenario, the RSO algorithm selects 4522 features from the selected variables: wind speed, wind direction, temperature, dew point, and historical $PM_{2.5}$, which are given to the Bi-GRU model for effective forecasting of air quality. The parameters considered in the RSO algorithm are mentioned as follows: the maximum number of iterations is equal

to 100, Chance-Max is equal to 5, flip is equal to 5, the number of bees is equal to 100, the learning rate is equal to 0.001, α is set to 0.2, and β is set to 0.1.

3.5. Air Quality Prediction

GRU has fewer gates than LSTM, which makes it less complicated. Sequential data's long-term dependencies can be successfully maintained using GRUs. They can also deal with the so-called short-term memory problem. The selected 4522 features were given to the Bi-GRU model for effective forecasting of air pollution [38]. The GRU model has reset and update gates for effective AQP; these gates reduce computational loss and gradient dispersion and enable the ability of longer-term memory. The update gate d_{TS} replaces forget and input gates of the LSTM network; it determines the retention degree of the prior information in the present forecasting, and it is mathematically presented in Equation (7).

$$d_{TS} = \sigma(W_d \times [h_{TS-1}, Fea_{TS}] + s_d) \tag{7}$$

where h_{TS-1} represents the hidden state at the prior time step $TS-1$; σ denotes the sigmoid activation function, which ranges between 0 and 1; Fea_{TS} indicates the input matrix at time step TS; and W_d and s_d denote the weight matrix and bias matrix of the update gate d_{TS} [39]. On the other hand, the reset gate p_{TS} controls the historical time series data, and it is mathematically specified in Equation (8).

$$p_{TS} = \sigma(W_p \times [h_{TS-1}, Fea_{TS}] + s_p) \tag{8}$$

where W_p and s_p denote the weight matrix and bias matrix of the reset gate p_{TS} [40]. Further, the candidate hidden state $\tilde{h}_{TS}$ is mathematically denoted in Equation (9).

$$\tilde{h}_{TS} = tanh\ (W_h \times [h_{TS-1} \odot p_{TS}, Fea_{TS}] + s_h) \tag{9}$$

where $\odot$ indicates dot multiplication operation, W_h and s_h represent the weight matrix and bias matrix of the memory cell state, and *tanh* denotes the tangent activation function. The linear interpolation between $\tilde{h}_{TS}$ and h_{TS-1} results in output h_{TS}, which is mathematically specified in Equation (10). The flow diagram of the Bi-GRU model is shown in Figure 3.

$$h_{TS} = (1 - d_{TS}) \odot h_{TS-1} + d_{TS} \odot \tilde{h}_{TS} \tag{10}$$

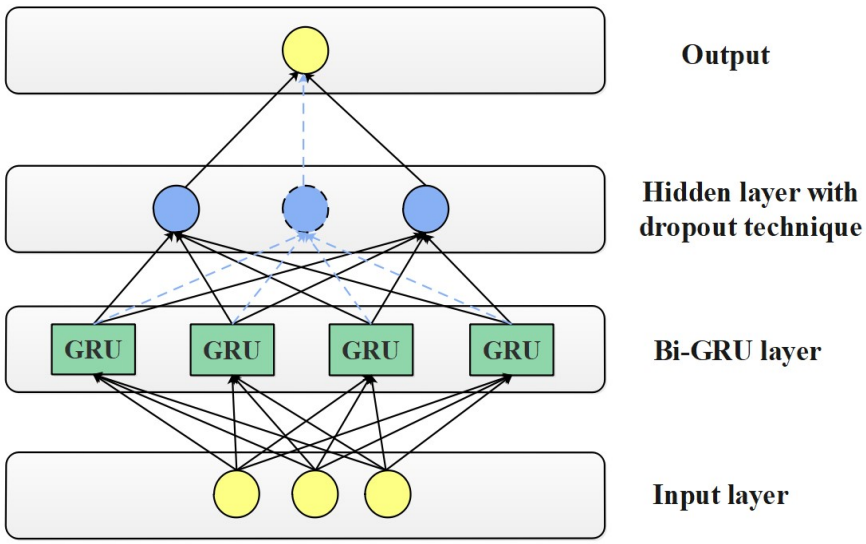

Figure 3. Flow diagram of the Bi-GRU model.

Generally, an effective prediction model is required for AQP for extracting complex variances and implicit features from sequence data. The conventional GRU model only extracts feature information from the forward direction, and it ignores the backward time series data. Therefore, the Bi-GRU model was used in this study, which proved to be effective in mining the knowledge between the meteorological variables from both backward and forward directions. The Bi-GRU model, composed of backward and forward GRUs, is shown in Figure 3. The backward GRU obtains future information from the input data, and the forward GRU captures past information from the input data. The Bi-GRU model O_{TS} is mathematically denoted in Equation (11).

$$O_{TS} = C\left(\overrightarrow{h}_{TS}, \overleftarrow{h}_{TS}\right) \tag{11}$$

where C indicates the output of two directions (summation function, average function, multiplication function, and so on), and $\overrightarrow{h}_{TS}$ and $\overleftarrow{h}_{TS}$ represent the hidden state of both forward and backward GRUs. The assumed parameters of the Bi-GRU model are represented as follows: the learning rate is equal to 0.001, the optimizer is set to Adam, the loss function is the MSE loss, the number of epochs is set to 100, the batch size is set to 50, the dropout rate is equal 0.5, the number of neurons (NoN) is set to 80, and look-back is set to 8. The numerical analysis of the proposed regression model (RSO and Bi-GRU) is discussed in Section 4.

4. Numerical Analysis

The proposed regression model (RSO and Bi-GRU) was simulated using a custom-built Python 3.7 software tool and tested on a computer with an NVidia GeForce RTX 3080, 128 GB of random-access memory (RAM), a Linux operating system, and an Intel core-i5 12th generation processor. The Beijing $PM_{2.5}$ dataset and a real-time dataset were utilized for evaluating the effectiveness of the proposed regression model (RSO and Bi-GRU), and the proposed model was compared with six existing regression models. For this article, all machine learning and deep learning models were trained on scikit-learn, TensorFlow, and Keras libraries. All the regression models were trained with a learning rate of 0.001, the Adam optimizer type, and an MSE loss function.

4.1. Performance Measures

The proposed model's (RSO and Bi-GRU) efficacy was evaluated using different loss functions, such as MAE, $SMAPE$, $RMSE$ and MSE. The MAE performance measure effectively reflected the actual situation of the forecasting error. In addition, the other performance measures, such as $RMSE$ and $SMAPE$, effectively evaluate the degree of data change and measure the prediction quality of the proposed model. On the other hand, the MSE is determined as the average or mean square difference between the estimated and actual values. The mathematical formulas of the performance measures MSE, MAE, $RMSE$, $SMAPE$, R^2, and $MAPE$ are stated in Equations (12)–(17).

$$MSE_{(z',z)} = \frac{1}{n}\sum_{i=1}^{n}\left(z'_i - z_i\right) \tag{12}$$

$$MAE_{(z',z)} = \frac{1}{n}\sum_{i=1}^{n}\left|z'_i - z_i\right| \tag{13}$$

$$RMSE_{(z',z)} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(z'_i - z_i\right)^2} \tag{14}$$

$$SMAPE_{(z',z)} = \frac{1}{n}\sum_{i=1}^{n}\frac{\left|z'_i - z_i\right|}{\left(z'_i - z_i\right)/2} \tag{15}$$

$$R^2 = 1 - \frac{\sum_i (z_i - \hat{z}_i)^2}{\sum_i (z_i - \bar{z})^2} \quad (16)$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{z_i - \hat{z}_i}{z_i}\right| \quad (17)$$

where n represents the number of samples, z_i' indicates the predicted time series value, and z_i denotes the measured time series value.

4.2. Experimental Setup with Ablation Analysis

In this scenario, six comparative regression models, namely SVR, random forest, recurrent neural network (RNN), LSTM, extra tree regression (ETR), and multiple linear regression (MLR), were developed for investigating the efficacy of the proposed model (GRU and Bi-LSTM). Here, all regression models were trained for 100 epochs with a batch size of 50. Additionally, a dropout layer with a probability of 0.5 was extensively applied between the layers in order to avoid the overfitting problem. The weight matrices were stored when the loss value of the previous epoch was higher compared to the present epoch. All regression models utilized an early stopping condition that stops the model training when the validation loss does not change within 10 training epochs. The Adam optimizer was utilized as the optimizer in the regression models because it iteratively updates its learning rate and effectively handles sparse gradients in noisy problems. Further, the Adam optimizer addresses two major concerns, including the local minima and convergence speed. Each data point in the testing set was verified by means of MAE, $SMAPE$, $RMSE$ and MSE after the trained models were obtained.

Numerous ablation experiments were performed in terms of MAE, $SMAPE$, $RMSE$ and MSE as specified in Tables 2–5. A couple of hyper-parameters, namely NoN and look-back, were tuned in the Bi-GRU model for achieving better prediction performance. The NoN indicates which neurons have a high prediction effect, and the look-back represents the previous time steps needed by the normalized data. Here, the NoNs were chosen from different candidate sets {256, 128, 80, 64, and 32}. Tables 2 and 3 represent the effect of the NoN on the Bi-GRU model for a Beijing $PM_{2.5}$ dataset and a real-time dataset. Table 2 indicates that the Bi-GRU model with 80 neurons obtained a minimum error rate with MAE of 9.11, $SMAPE$ of 0.16, $RMSE$ of 9.82, MSE of 2.82, $MAPE$ of 13.76, and R^2 of 2.45 on a Beijing $PM_{2.5}$ dataset. Correspondingly, as depicted in Table 3, the Bi-GRU model with 80 neurons obtained a lower error rate with MAE of 0.19, $SMAPE$ of 0.44, $RMSE$ of 0.48, MSE of 0.26, $MAPE$ of 16.59, and R^2 of 1.86 on a real-time dataset.

Table 2. Analyzing the effect of NoN on Bi-GRU model for a Beijing $PM_{2.5}$ dataset.

Neurons	MAE	SMAPE	RMSE	MSE	MAPE	R^2
256	23.40	5.68	32.80	6.53	16.42	3.12
128	19.43	4.53	28.27	5.49	14.23	2.90
80	9.11	0.16	9.82	2.82	13.76	2.45
64	10.92	2.33	12.12	3.90	12.48	2.01
32	14.55	3.48	19.30	4.55	11.59	1.84

Table 3. Analyzing the effect of NoN on Bi-GRU model for a real-time dataset.

Neurons	MAE	SMAPE	RMSE	MSE	MAPE	R^2
256	3.88	5.01	3.92	3.92	18.29	2.42
128	2.12	4.02	2.03	3.80	17.37	2.23
80	0.19	0.44	0.48	0.26	16.59	1.86
64	1.82	2.92	1.18	1.94	14.74	1.77
32	1.50	3.11	1.78	2.50	13.19	1.75

Table 4. Analyzing the effect of look-back on Bi-GRU model for a Beijing $PM_{2.5}$ dataset.

Look-Back	MAE	SMAPE	RMSE	MSE	MAPE	R^2
16	12.33	12.20	15.60	12.80	14.23	2.55
12	20.47	9.82	13.22	10.28	12.38	2.49
10	13.45	2.39	10.09	6.50	14.15	2.48
8	9.11	0.16	9.82	2.82	12.74	1.94
6	11.25	2.11	12.92	12.92	13.76	1.43
4	20.34	10.82	16.57	18.29	12.84	1.29

Table 5. Analyzing the effect of look-back on Bi-GRU model for a real-time dataset.

Look-Back	MAE	SMAPE	RMSE	MSE	MAPE	R^2
16	2.80	3.23	4.22	3.20	12.64	1.92
12	2.12	2.30	3.20	1.29	12.43	1.74
10	1.92	1.14	1.01	0.82	12.29	1.56
8	0.19	0.44	0.48	0.26	12.06	1.19
6	1.02	0.82	0.93	1.28	11.98	1.12
4	0.82	1.44	0.98	1.22	11.89	0.99

Tables 4 and 5 represent the effect of look-back on the Bi-GRU model for a Beijing $PM_{2.5}$ dataset and a real-time dataset, respectively. Tables 4 and 5 show that the Bi-GRU model with a look-back of 8 has obtained a minimum error rate in terms of *MAE*, *SMAPE*, *RMSE*, *MSE*, *MAPE*, and R^2. The optimal selection of look-back and NoN effectively fits the model on historical data for better AQP.

4.3. Analysis on a Beijing $PM_{2.5}$ Dataset

Quantitative analysis was performed on a Beijing $PM_{2.5}$ dataset by varying the regression models and the optimization algorithms. In this research, the importance of the normalization technique (Min-Max normalization technique) is specified in Table 6. The table clearly shows that preprocessing data using the Min-Max normalization technique accomplished better results in terms of *MAE* (9.11), *SMAPE* (10.16), *RMSE* (9.82), *MSE* (12.82), *MAPE* (12.27), and R^2 (0.78) on a Beijing $PM_{2.5}$ dataset. For non-preprocessed data, an *MAE* of 12.37, *SMAPE* of 14.27, *RMSE* of 20.38, *MSE* of 11.83, *MAPE* of 19.92, and R^2 of 0.43 were obtained. The above values demonstrate that the preprocessing of the data using the Min-Max normalization technique effectively preserves the relation between the original data values with limited standard deviations that effectively suppress the effect of outliers. On the other hand, the numerical analysis of different deep learning models on a Beijing $PM_{2.5}$ dataset is represented in Table 7. Compared to other regression models (SVR, random forest (RF), RNN, LSTM, ETR, MLR, GRU, and Bi-LSTM), the Bi-GRU model obtained better forecasting performance with the minimal *MAE* of 9.11, *SMAPE* of 0.16, *RMSE* of 9.82, *MSE* of 2.82, *MAPE* of 10.46, and R^2 of 0.84 on a Beijing $PM_{2.5}$ dataset. The graphical evaluation of different prediction models on a Beijing $PM_{2.5}$ dataset is shown in Figure 4.

The comparison of different feature selection algorithms on a Beijing $PM_{2.5}$ dataset is described in Table 8. As mentioned in Table 8, the RSO algorithm with Bi-GRU model obtained higher forecasting performance with minimal error rate compared to other optimization algorithms, such as the butterfly optimization algorithm (BOA), firefly optimization algorithm (FOA), whale optimization algorithm (WOA), genetic algorithm (GA), grey wolf optimization (GWO) algorithm, and particle swarm optimization (PSO) algorithm. The selection of optimal features by the RSO algorithm significantly decreases the computational time. The proposed regression model (RSO and Bi-GRU) consumed a computational time of 43.22 s, which is efficient in comparison to other combinations. The graphical evaluation of different optimization algorithms on a Beijing $PM_{2.5}$ dataset is shown in Figure 5. Additionally, the plot of actual and predicted $PM_{2.5}$ values and the boxplots of actual and predicted $PM_{2.5}$ values are presented in Figures 6–8. The boxplot of prediction errors from deep learning models is illustrated in Figure 9.

Table 6. Performance analysis of normalization technique on a Beijing $PM_{2.5}$ dataset.

Technique	MAE	SMAPE	RMSE	MSE	MAPE	R^2
Non-preprocessed data	12.37	14.27	20.38	11.83	19.92	0.43
Preprocessed data (Min-Max)	9.11	10.16	9.82	12.82	12.27	0.78

Table 7. Comparison of different deep learning models on a Beijing $PM_{2.5}$ dataset.

Models	MAE	SMAPE	RMSE	MSE	MAPE	R^2
SVR	13.12	0.21	23.22	12.14	18.92	0.76
Random forest	14.32	0.22	24.92	12.33	17.79	0.71
RNN	13.90	0.28	19.21	10.94	16.71	0.72
LSTM	12.12	0.25	15.42	9.13	15.54	0.75
ETR	11.94	0.24	14.26	8.12	15.51	0.77
MLR	11.35	0.22	15.34	9.94	14.47	0.76
GRU	11.22	0.21	16.39	9.82	12.42	0.79
Bi-LSTM	10.78	0.26	12.77	6.65	12.36	0.80
Bi-GRU	9.11	0.16	9.82	2.82	10.46	0.84

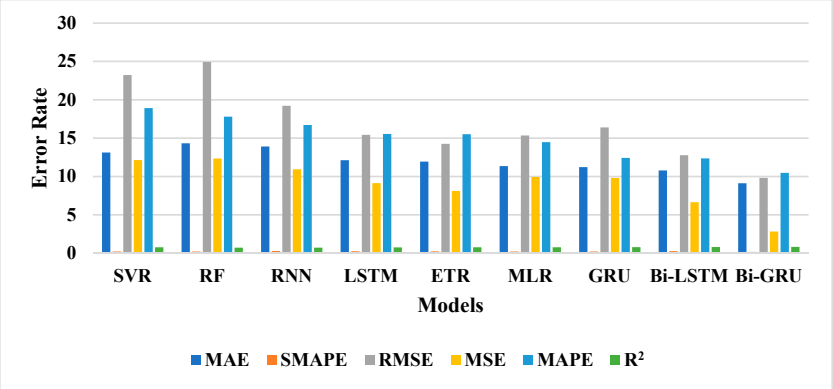

Figure 4. Graphical evaluation of different prediction models on a Beijing $PM_{2.5}$ dataset.

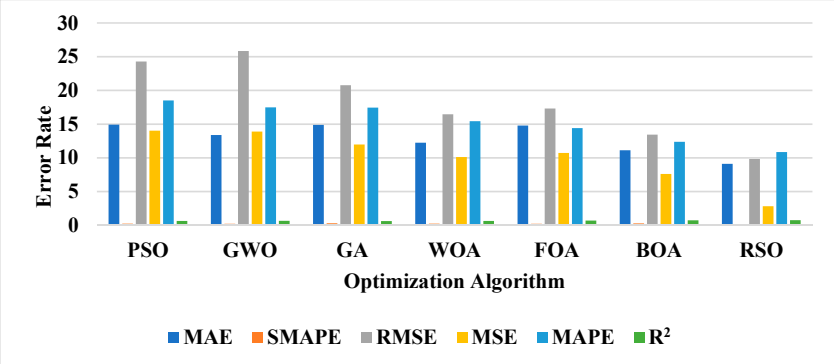

Figure 5. Graphical evaluation of different optimization algorithms on a Beijing $PM_{2.5}$ dataset.

Table 8. Comparison of different feature selection algorithms on a Beijing $PM_{2.5}$ dataset.

Optimization Algorithm	MAE	SMAPE	RMSE	MSE	MAPE	R^2
PSO	14.92	0.24	24.28	14.02	18.51	0.65
GWO	13.38	0.23	25.83	13.90	17.47	0.66
GA	14.88	0.29	20.77	11.98	17.45	0.63
WOA	12.23	0.25	16.46	10.12	15.42	0.64
FOA	14.78	0.22	17.30	10.72	14.40	0.69
BOA	11.12	0.28	13.44	7.60	12.38	0.73
RSO	9.11	0.16	9.82	2.82	10.86	0.86

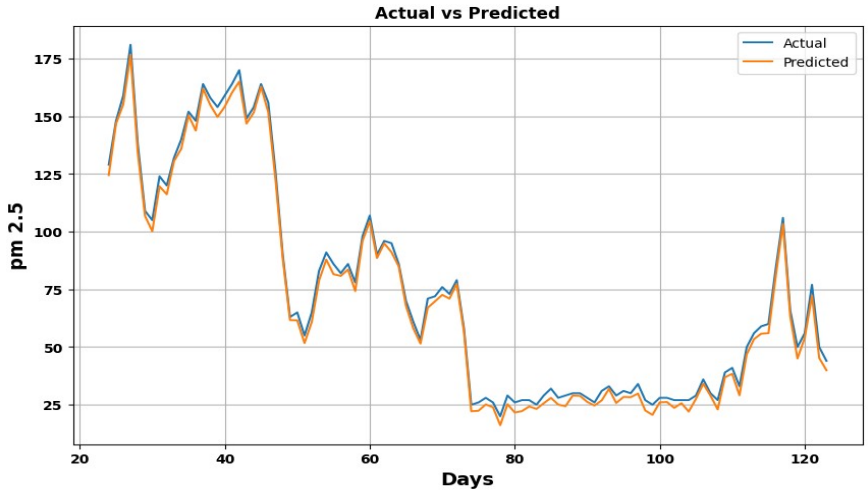

Figure 6. Plot of actual and predicted $PM_{2.5}$ values.

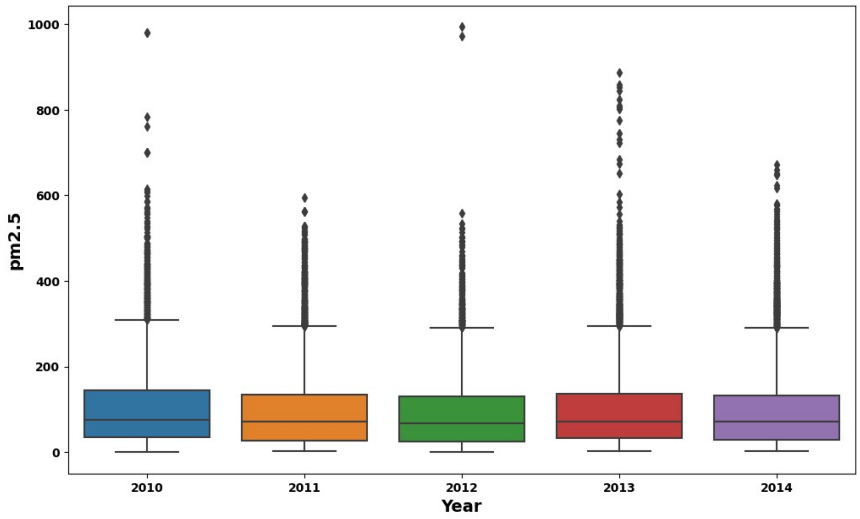

Figure 7. Boxplot of actual $PM_{2.5}$ values.

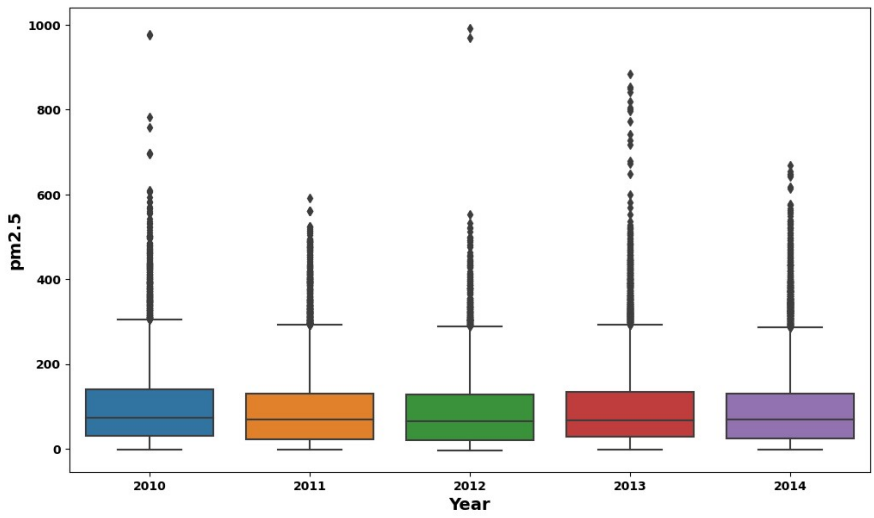

Figure 8. Boxplot of predicted $PM_{2.5}$ values.

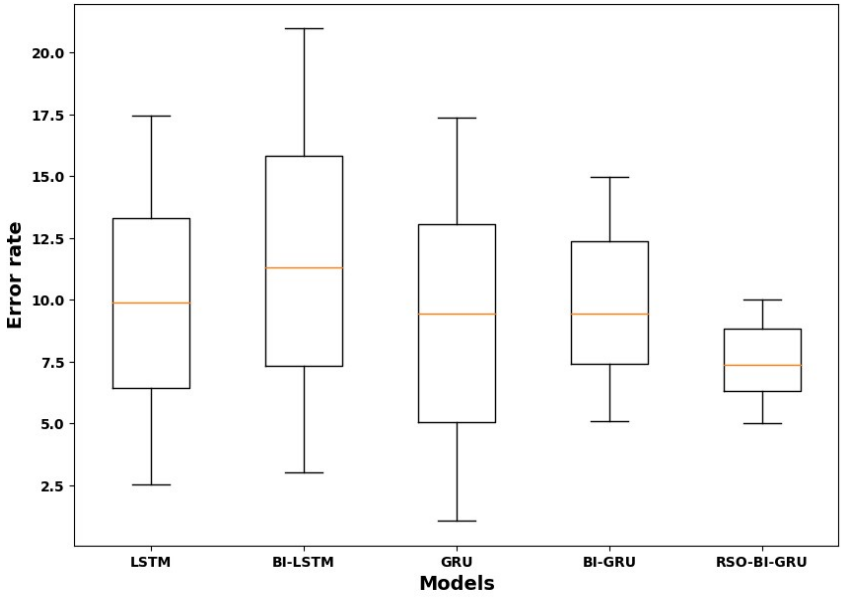

Figure 9. Boxplot of prediction errors from deep learning models.

4.4. Analysis on a Real-Time Dataset

This subsection presents the results of the quantitative analysis performed on a real-time dataset by varying the deep learning models and optimization algorithms. The importance of the normalization technique is clearly described in Table 9. The table clearly shows that the results of non-preprocessed data provide an *MAE* of 0.80, *SMAPE* of 0.76, *RMSE* of 0.78, *MSE* of 0.81, *MAPE* of 17.93, and R^2 of 0.46. Once the data are preprocessed using the Min-Max technique, minimal error rate values are obtained (*MAE* of 0.39, *SMAPE* of 0.44, *RMSE* of 0.48, *MSE* of 0.41, *MAPE* of 12.31, and R^2 of 0.74). As specified in Table 10, the Bi-GRU model achieved the minimal *MAE* of 0.19, *SMAPE* of 0.44, *RMSE* of 0.48, *MSE* of 0.26, *MAPE* of 10.98, and R^2 of 0.83, which are better when compared to those of the six others regression models. In time series forecasting, the Bi-GRU model uses special gates (reset and update gates) that reduce the computational loss, enable the ability of long-term memory, and reduce the gradient dispersion.

Table 9. Performance analysis of normalization technique on a real-time dataset.

Technique	MAE	SMAPE	RMSE	MSE	MAPE	R^2
Non-preprocessed data	0.80	0.76	0.78	0.81	17.93	0.46
Preprocessed data (Min-Max)	0.39	0.44	0.48	0.41	12.31	0.74

Table 10. Comparison of different deep learning models on a real-time dataset.

Models	MAE	SMAPE	RMSE	MSE	MAPE	R^2
SVR	1.28	1.11	1.45	1.76	14.83	0.64
Random forest	1.23	1.02	1.33	1.44	14.71	0.68
RNN	0.82	0.91	1.09	1.28	14.62	0.66
LSTM	0.76	0.86	1.12	0.88	14.54	0.69
ETR	0.71	0.81	0.94	0.65	13.51	0.70
MLR	0.67	0.79	0.77	0.54	13.49	0.71
GRU	0.54	0.77	0.62	0.43	12.46	0.76
Bi-LSTM	0.22	0.54	0.56	0.31	11.37	0.77
Bi-GRU	0.19	0.44	0.48	0.26	10.98	0.83

Correspondingly, the comparison of different feature optimization algorithms on a real-time dataset is described in Table 11. As shown, the combination of the RSO algorithm with the Bi-GRU model obtained a minimal error rate compared to other optimization

algorithms. The RSO algorithm effectively selects the optimal features from the highly correlated variables (wind speed, wind direction, temperature, dew point, and historical $PM_{2.5}$) with a better convergence rate. This action improves the prediction performance with limited computational time. In this scenario, the proposed regression model (RSO and Bi-GRU) consumed a computational time of 19.28 s on a real-time dataset.

Table 11. Comparison of different feature selection algorithms on a real-time dataset.

Optimization Algorithm	MAE	SMAPE	RMSE	MSE	MAPE	R^2
PSO	1.55	1.16	1.32	1.35	18.59	0.58
GWO	1.28	1.08	1.20	1.22	18.53	0.62
GA	0.77	0.98	1.02	1.04	17.50	0.59
WOA	0.64	0.84	0.82	0.84	16.46	0.57
FOA	0.50	0.90	0.60	0.48	15.14	0.64
BOA	0.25	0.53	0.51	0.30	13.02	0.70
RSO	0.19	0.44	0.48	0.26	11.37	0.81

4.5. Comparative Analysis

As reviewed in the literature section, Tao et al. [16] combined the CNN and Bi-GRU models for effective AQP. The experiments performed on a Beijing $PM_{2.5}$ dataset demonstrated the efficacy of the developed model. The developed CNN-Bi-GRU model obtained the *RMSE* of 14.53, *MAE* of 10.47, and *SMAPE* of 0.20 on a Beijing $PM_{2.5}$ dataset. Compared to this existing model, the proposed regression model (RSO and Bi-GRU) obtained better AQP with the *RMSE* of 9.82, *MAE* of 9.11, and *SMAPE* of 0.16 on a Beijing $PM_{2.5}$ dataset, as shown in Table 12. Aarthi et al. [31] used the Min-Max normalization technique, BSMO, and Bi-LSTM network for AQP. The developed model proved to be effective in AQP with an *MSE* of 0.31, *RMSE* of 0.56, and *MAE* of 0.22 on a real-time dataset. As specified in Table 13, the proposed regression model (RSO and Bi-GRU) obtained the minimal *MAE* of 0.19, *RMSE* of 0.48, and *MSE* of 0.26 on a real-time dataset, and these results are better than those of the existing model.

Table 12. Comparative results on a Beijing $PM_{2.5}$ dataset.

Models	MAE	SMAPE	RMSE
CNN and Bi-GRU [16]	10.47	0.20	14.53
RSO and Bi-GRU	9.11	0.16	9.82

Table 13. Comparative results on a real-time dataset.

Models	MAE	RMSE	MSE
BSMO and Bi-LSTM [31]	0.22	0.56	0.31
RSO and Bi-GRU	0.19	0.48	0.26

4.6. Discussion

As discussed earlier, feature selection and prediction are the two integral parts of this research. The selection of optimal features from the highly correlated variables, namely wind direction, temperature, dew point, wind speed, and historical $PM_{2.5}$, significantly increases the prediction performance with limited computational time and complexity. In this research, the RSO algorithm was utilized for feature selection, and the Bi-GRU model was implemented for AQP. Compared to other deep learning models, the Bi-GRU model utilizes reset and update gates for AQP, and these gates reduce the gradient dispersion and computational loss and enable the ability of long-term memory. Correspondingly, the RSO algorithm significantly selects the optimal features with a better convergence rate. The RSO algorithm has better exploration and exploitation abilities in achieving better feature selection performance. Additionally, the Diebold Mariano (DM) test was conducted for this manuscript to assess the superiority of the proposed regression model statistically. The DM test defines the loss differential between forecasts. Here, the probability *p* value

of the DM test was equal to 0.01, which shows that the proposed regression model is statistically efficient. The numerical study revealed that the suggested model on a Beijing $PM_{2.5}$ dataset and a real-time dataset produced values of 9.11 and 0.19 for *MAE* and 2.82 and 0.26 for *MSE*. On a Beijing $PM_{2.5}$ dataset and a real-time dataset, the suggested regression model (RSO and Bi-GRU) required the least amount of processing time, 43.22 and 19.28 s, respectively.

5. Conclusions

In this research, a new optimization-based regression model (RSO and Bi-GRU) was implemented for effective AQP. In the present scenario, effective AQP assists the government in controlling pollution. After collecting Beijing $PM_{2.5}$ and real-time data, normalization and correlation analysis were accomplished to eliminate the outliers and select the highly correlated variables: wind direction, temperature, dew point, wind speed, and historical $PM_{2.5}$. From the selected variables, the optimal and relevant features were selected by implementing the RSO algorithm. Finally, the selected features from the variables were given to the Bi-GRU model for AQP. Here, the proposed model's (RSO and Bi-GRU) performance was validated on a Beijing $PM_{2.5}$ dataset and a real-time dataset, and it was evaluated using different performance measures, such as MAE, SMAPE, RMSE, and MSE. The numerical analysis showed that the proposed model obtained MAE values of 9.11 and 0.19 and MSE values of 2.82 and 0.26 on a Beijing $PM_{2.5}$ dataset and a real-time dataset. Additionally, the proposed regression model (RSO and Bi-GRU) consumed minimal computational time of 43.22 and 19.28 s on a Beijing $PM_{2.5}$ dataset and a real-time dataset.

Still, the proposed regression model faces difficulty in analyzing real-time data due to their dynamic nature and high variability. Therefore, as an extension, hyper-parameter tuning was performed in the Bi-GRU model to further enhance the prediction efficiency. In addition to this, the present research work can be further extended by conducting both parametric and non-parametric statistical analysis using the Wilcoxon test, *t*-test, Z-test, etc. In upcoming research, the high-pollution Indian cities (Delhi and Ghaziabad) will be also considered in experiments.

Author Contributions: Investigation, resources, data curation, writing—original draft preparation, writing—review and editing, and visualization: S.G. Conceptualization and software: A.K.K. Validation, formal analysis, methodology, supervision, project administration, and funding acquisition: P.B.D. and P.F.-G. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Cabaneros, S.M.; Calautit, J.K.; Hughes, B.R. A review of artificial neural network models for ambient air pollution prediction. *Environ. Modell. Softw.* **2019**, *119*, 285–304. [CrossRef]
2. Harrou, F.; Kadri, F.; Khadraoui, S.; Sun, Y. Ozone measurements monitoring using data-based approach. *Process Saf. Environ. Prot.* **2016**, *100*, 220–231. [CrossRef]
3. Wu, Q.; Lin, H. A novel optimal-hybrid model for daily air quality index prediction considering air pollutant factors. *Sci. Total Environ.* **2019**, *683*, 808–821. [CrossRef] [PubMed]
4. Ameer, S.; Shah, M.A.; Khan, A.; Song, H.; Maple, C.; Islam, S.U.; Asghar, M.N. Comparative analysis of machine learning techniques for predicting air quality in smart cities. *IEEE Access* **2019**, *7*, 128325–128338. [CrossRef]
5. Amuthadevi, C.; Vijayan, D.S.; Ramachandran, V. Development of air quality monitoring (AQM) models using different machine learning approaches. *J. Ambient. Intell. Hum. Comput.* **2022**, *13*, 33. [CrossRef]
6. Hao, Y.; Tian, C. The study and application of a novel hybrid system for air quality early warning. *Appl. Soft Comput.* **2019**, *74*, 729–746. [CrossRef]

7. Dionova, B.W.; Mohammed, M.N.; Al-Zubaidi, S.; Yusuf, E. Environment indoor air quality assessment using fuzzy inference system. *ICT Express* **2020**, *6*, 185–194. [CrossRef]
8. Yuan, G.; Yang, W. Evaluating China's air pollution control policy with extended AQI indicator system: Example of the Beijing-Tianjin-Hebei Region. *Sustainability* **2019**, *11*, 939. [CrossRef]
9. Bikkina, S.; Andersson, A.; Kirillova, E.N.; Holmstrand, H.; Tiwari, S.; Srivastava, A.K.; Bisht, D.S.; Gustafsson, Ö. Air quality in megacity Delhi affected by countryside biomass burning. *Nat. Sustain.* **2019**, *2*, 200–205. [CrossRef]
10. Dairi, A.; Harrou, F.; Khadraoui, S.; Sun, Y. Integrated multiple directed attention-based deep learning for improved air pollution forecasting. *IEEE Trans. Instrum. Meas.* **2021**, *70*, 3520815. [CrossRef]
11. Yan, Y.; Li, Y.; Sun, M.; Wu, Z. Primary pollutants and air quality analysis for urban air in China: Evidence from Shanghai. *Sustainability* **2019**, *11*, 2319. [CrossRef]
12. Woo, J.H.; Kim, Y.; Kim, H.K.; Choi, K.C.; Eum, J.H.; Lee, J.B.; Lim, J.H.; Kim, J.; Seong, M. Development of the CREATE inventory in support of integrated climate and air quality modeling for Asia. *Sustainability* **2020**, *12*, 7930. [CrossRef]
13. Rahman, M.M.; Shafiullah, M.; Rahman, S.M.; Khondaker, A.N.; Amao, A.; Zahir, M.H. Soft computing applications in air quality modeling: Past, present, and future. *Sustainability* **2020**, *12*, 4045. [CrossRef]
14. Esager, M.W.M.; Ünlü, K.D. Forecasting air quality in Tripoli: An evaluation of deep learning models for hourly PM2.5 surface mass concentrations. *Atmosphere* **2023**, *14*, 478. [CrossRef]
15. 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]
16. 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]
17. Ma, J.; Li, Z.; Cheng, J.C.P.; Ding, Y.; Lin, C.; Xu, Z. Air quality prediction at new stations using spatially transferred bi-directional long short-term memory network. *Sci. Total Environ.* **2020**, *705*, 135771. [CrossRef]
18. Chang, Y.S.; Chiao, H.T.; Abimannan, S.; Huang, Y.P.; Tsai, Y.T.; Lin, K.M. An LSTM-based aggregated model for air pollution forecasting. *Atmos. Pollut. Res.* **2020**, *11*, 1451–1463. [CrossRef]
19. Castelli, M.; Clemente, F.M.; Popovič, A.; Silva, S.; Vanneschi, L. A machine learning approach to predict air quality in California. *Complexity* **2020**, *2020*, 8049504. [CrossRef]
20. Xayasouk, T.; Lee, H.; Lee, G. Air pollution prediction using long short-term memory (LSTM) and deep autoencoder (DAE) models. *Sustainability* **2020**, *12*, 2570. [CrossRef]
21. Wen, C.; Liu, S.; Yao, X.; Peng, L.; Li, X.; Hu, Y.; Chi, T. A novel spatiotemporal convolutional long short-term neural network for air pollution prediction. *Sci. Total Environ.* **2019**, *654*, 1091–1099. [CrossRef] [PubMed]
22. Wang, B.; Kong, W.; Guan, H.; Xiong, N.N. Air quality forecasting based on gated recurrent long short-term memory model in Internet of Things. *IEEE Access* **2019**, *7*, 69524–69534. [CrossRef]
23. Zhang, Y.; Zhang, R.; Ma, Q.; Wang, Y.; Wang, Q.; Huang, Z.; Huang, L. A feature selection and multi-model fusion-based approach of predicting air quality. *ISA Trans.* **2020**, *100*, 210–220. [CrossRef] [PubMed]
24. Wang, J.; Du, P.; Hao, Y.; Ma, X.; Niu, T.; Yang, W. An innovative hybrid model based on outlier detection and correction algorithm and heuristic intelligent optimization algorithm for daily air quality index forecasting. *J. Environ. Manag.* **2020**, *255*, 109855. [CrossRef] [PubMed]
25. Akbal, Y.; Ünlü, K.D. A deep learning approach to model daily particular matter of Ankara: Key features and forecasting. *Int. J. Environ. Sci. Technol.* **2021**, *19*, 5911–5927. [CrossRef]
26. Li, H.; Wang, J.; Li, R.; Lu, H. Novel analysis-forecast system based on multi-objective optimization for air quality index. *J. Clean. Prod.* **2019**, *208*, 1365–1383. [CrossRef]
27. Maleki, H.; Sorooshian, A.; Goudarzi, G.; Baboli, Z.; Birgani, Y.T.; Rahmati, M. Air pollution prediction by using an artificial neural network model. *Clean Technol. Environ. Policy* **2019**, *21*, 1341–1352. [CrossRef]
28. Mao, W.; Wang, W.; Jiao, L.; Zhao, S.; Liu, A. Modeling air quality prediction using a deep learning approach: Method optimization and evaluation. *Sustain. Cities Soc.* **2021**, *65*, 102567. [CrossRef]
29. Zhang, L.; Liu, P.; Zhao, L.; Wang, G.; Zhang, W.; Liu, J. Air quality predictions with a semi-supervised bidirectional LSTM neural network. *Atmos. Pollut. Res.* **2021**, *12*, 328–339. [CrossRef]
30. Zeinalnezhad, M.; Chofreh, A.G.; Goni, F.A.; Klemeš, J.J. Air pollution prediction using semi-experimental regression model and adaptive neuro-fuzzy inference system. *J. Clean. Prod.* **2020**, *261*, 121218. [CrossRef]
31. Aarthi, C.; Ramya, V.J.; Falkowski-Gilski, P.; Divakarachari, P.B. Balanced spider monkey optimization with Bi-LSTM for sustainable air quality prediction. *Sustainability* **2023**, *15*, 1637. [CrossRef]
32. Liang, X.; Zou, T.; Guo, B.; Li, S.; Zhang, H.; Zhang, S.; Huang, H.; Chen, S.X. Assessing Beijing's PM2.5 pollution: Severity, weather impact, APEC and winter heating. *Proc. R Soc. A* **2015**, *471*, 20150257. [CrossRef]
33. Mazziotta, M.; Pareto, A. Normalization methods for spatio-temporal analysis of environmental performance: Revisiting the Min-Max method. *Environmetrics* **2022**, *33*, e2730. [CrossRef]
34. Islam, M.J.; Ahmad, S.; Haque, F.; Reaz, M.B.I.; Bhuiyan, M.A.S.; Islam, M.R. Application of Min-Max normalization on subject-invariant EMG pattern recognition. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 2521612. [CrossRef]
35. Yang, N.C.; Mehmood, D. Multi-objective bee swarm optimization algorithm with minimum Manhattan distance for passive power filter optimization problems. *Mathematics* **2022**, *10*, 133. [CrossRef]

36. Fajri, Y.A.Z.A.; Wiharto, W.; Suryani, E. Hybrid model feature selection with the bee swarm optimization method and q-learning on the diagnosis of coronary heart disease. *Information* **2023**, *14*, 15. [CrossRef]
37. Wu, D.; Wang, S.; Liu, Q.; Abualigah, L.; Jia, H. An improved teaching-learning-based optimization algorithm with reinforcement learning strategy for solving optimization problems. *Comput. Intell. Neurosci.* **2022**, *2022*, 1535957. [CrossRef]
38. Yuan, Q.; Wang, J.; Zheng, M.; Wang, X. Hybrid 1D-CNN and attention-based Bi-GRU neural networks for predicting moisture content of sand gravel using NIR spectroscopy. *Constr. Build. Mater.* **2022**, *350*, 128799. [CrossRef]
39. Zhang, X.; Wu, Z.; Liu, K.; Zhao, Z.; Wang, J.; Wu, C. Text sentiment classification based on BERT embedding and sliced multi-head self-attention Bi-GRU. *Sensors* **2023**, *23*, 1481. [CrossRef]
40. Xu, H.; Zhang, A.; Xu, X.; Li, P.; Ji, Y. Prediction of particulate concentration based on correlation analysis and a Bi-GRU model. *Int. J. Environ. Res. Public Health* **2022**, *19*, 13266. [CrossRef]