ISSN: 2633-4828 Vol. 5 No.4, December, 2023

International Journal of Applied Engineering & Technology

A HYBRID MACHINE LEARNING REGRESSION FRAMEWORK FOR AIR QUALITY PREDICTION WITH META-HEURISTIC APPROACH

Dr.R. Merlin Packiam, Ms. M. Ellakkiya and Ms. V. Infine Sinduja
Department of Computer Science, Cauvery College for Women (A), Bharathidasan University,Trichy,Tamil Nadu, India
merlin.ca@cauverycollege.ac.in, elakkiya.ca@cauverycollege.ac.in and infinesinduja.ca@cauverycollege.ac.in

ABSTRACT
Without oxygen, it is impossible to understand how humanity would survive. Modern human culture has had constant advancements that have a negative impact on the quality of the air. Everyday transportation, industrial, and domestic operations churn up dangerous contaminants in our surroundings. In the modern day, air quality monitoring and forecasting have become crucial tasks, particularly in developing nations like India. The big data and machine learning based prediction technologies have been shown to be more effective than conventional methods for researching these contemporary threats. The current study examines six years' worth of air pollution data from 23 Indian cities to analyze and forecast air quality. The dataset has undergone thorough preprocessing, and the correlation analysis has been used to identify essential features. This research paper's goal is to examine various big-data and machine learning-based strategies for forecasting air quality. It also sheds light on some of the difficulties and the requirements for further study.

Keywords: Air quality, Big data, Machine learning, Meta-Heuristic.

1. INTRODUCTION
The recent rise in concern over environmental contamination is a result of the economic and urban expansion of cities. In addition to other issues, air pollution significantly affects human health by increasing the risk of several fatal diseases. Air pollution causes 21% of deaths due to pneumonia, 20% due to stroke, 34% owing to ischemic heart disease, 19% due to chronic obstructive pulmonary disease (COPD), and 7% due to lung cancer [1]. Because of the dangers of air pollution, there is a strong desire to successfully anticipate air pollution events ahead of time in order to notify the public and limit the number of deaths caused by air pollution. Air quality forecast models are widely used in highly developed and heavily populated cities and regions around the world. The World Health Organization (WHO) claims that air pollution is a "silent killer" that kills over seven million people early each year.

The COVID-19 epidemic began in early 2020, drastically altering most people's lifestyles and resulting in significant reductions in air pollution, resulting in cleaner ambient air. Carbon monoxide (CO), nitrogen dioxide (NO2), and PM10 levels declined during the partial lockdown measures in February 2020, whereas ozone (O3) levels raised when NO2 levels decreased in a VOC-controlled setting in Rio de Janeiro, Brazil [3]. Furthermore, the lockdown measures resulted in a considerable fall in NO2 concentrations in European countries such as France, Germany, Italy, and Spain, as well as a significant decrease in both NO2 and PM2 levels in China [4].

Meanwhile, predicting air quality got more difficult because the overall seasonal and temporal trend of air pollution altered; thus, new methodologies must be studied to accurately anticipate air quality in the future. In this work, the use of big data analysis and machine leaning regression models are applied to successfully improve the forecasting the air quality.

2. LITERATURE REVIEW
During a study in Cambridge, UK, an innovative framework was used to examine several forecast models such as statistical approaches, machine learning, and neural networks by combining pollution concentration, urban traffic, aerial images, and meteorological conditions [6]. Weather normalized models are used to measure air pollution using various methods like as machine learning and deep learning, and then to compare their performance to that of GB [7].

In one study, the air quality index (AQI) was predicted using a support vector machine (SVM). With a high coefficient of determination (R2) and a low sum square error (SSE) and mean sum square error (MSSE), the results were effective and accurate [8]. In Taiwan, researchers applied machine learning algorithms to anticipate PM using prior air quality datasets. The results reveal that machine learning models outperformed classical deterministic models in terms of air quality forecasting, as measured by model performance indices such as R2, root mean square error (RMSE), mean absolute error (MAE), and mean square error (MSE) [9].

In the forecast of AQI, RF, SVM, adaptive boosting (AdaBoost), artificial neural networks (ANNs), and stacking ensemble exhibited promising results, with stacking ensemble having the best results for R^2 and RMSE and AdaBoost having the greatest performance for MAE [10]. The use of RF, CART, and logistic regression, along with variable selections by an expert group and two automatic selection methods, revealed that the variables selected with expert knowledge outperformed the automatic methods in the air quality forecast of PM_{10} concentration levels in Bogotá, Columbia [11].

Extreme gradient boosting (XGB) and Random Forest (RF) were utilized to anticipate the hourly air quality in Delhi, India [12]. In a research to predict air quality in Beijing, GB outperformed XGB in terms of prediction accuracy and operational efficiency [13]. In Rio de Janeiro, Brazil, RF and SVM were used to investigate the association between air pollutants (O3, NOx, and CO) and meteorological factors (wind speed, solar radiation, temperature, and relative humidity) and successfully forecast the level of ozone concentrations with a high R2 [14].

Recommendation system is the fault information estimation of the reviews and the unrelated recommendations of the best selling or the better quality product. The complete feedback of the product is estimated by propose the Novel Product Feature-based Opinion Score Estimation (NPF-OSE) process is examined by compared with the traditional approaches in terms of subsequent parameters, precision, recall, F-measure, MAE and the RMSE. EFCFM approach produced the superior accuracy recommendation to the customers. [15]

3. MATERIAL AND METHODS

Air quality prediction using machine learning and big data is a popular and important application of data science. The goal is to build predictive models that can accurately forecast air quality levels in different regions, which can help people make informed decisions about their daily activities and prevent potential health risks. When building machine learning algorithms for air quality forecasting, it is important to select a suitable dataset that contains relevant features and target variables. The dataset should include concentration levels of various air pollutants, such as particulate matter, ozone, nitrogen oxides, and sulfur dioxide. These concentration levels can be measured in micrograms per cubic meter (µg/m³). The dataset should also include meteorological factors, such as wind speed and direction, temperature, humidity, and precipitation. These factors can significantly affect pollutant dispersion and should be included in any air quality forecast model. Since air quality varies over time, the dataset should include time series data, with measurements taken at regular intervals, such as hourly, daily, or weekly. This will allow the machine learning algorithms to capture the temporal patterns of air quality. The dataset should also include spatial information, such as the location of the monitoring station or the region being monitored. This information can help the machine learning algorithms understand the spatial variability of air quality. The target variable for air quality forecasting machine learning algorithms is usually the concentration levels of one or more air pollutants at a future time point. For example, the target variable may be the concentration levels of PM2.5 (particulate matter with a diameter of less than 2.5 micrometers) 24 hours in advance. The dataset should be of high quality, with accurate measurements and minimal missing data. Data cleaning and preprocessing may be required before feeding the data into the machine learning algorithms. In conclusion, a suitable air quality forecast dataset for machine learning algorithms should include pollutant concentration levels, meteorological factors, time series data, spatial information, a target variable, and high-quality data [16]. By selecting a suitable dataset, machine learning algorithms can be trained to accurately predict air quality levels, which can help representatives and the public take appropriate measures to protect their health and the environment. Here are some key steps involved in developing an air quality prediction model using machine learning and big data:

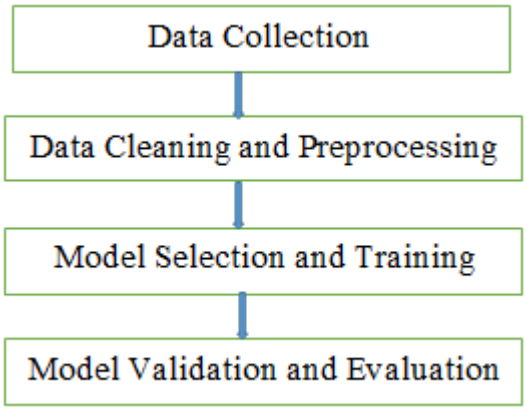

Figure 1. Air Quality Prediction Model

4. MACHINE LEARNING BASED METHODS

Machine Learning (ML) Regression techniques are the most common methods to forecast air quality.

4.1 Multiple Linear Regression (MLR):

Multiple Linear Regression is a supervised machine learning algorithm that is used to predict a continuous outcome variable based on multiple predictor variables. Here are the steps to implement the Multiple Linear Regression algorithm:

1. **Collect and preprocess data:** Collect the data that you want to use for the Multiple Linear Regression algorithm and preprocess it. This involves removing missing values, removing outliers, scaling the data, and encoding categorical variables.
2. **Split the data into training and testing sets:** Divide the data into two sets – the training set and the testing set. The training set is used to train the model, and the testing set is used to evaluate the model's performance.
3. **Fit the Multiple Linear Regression model:** Use the training data to fit the Multiple Linear Regression model. The goal of this step is to find the coefficients of the predictor variables that best predict the outcome variable.
4. **Make predictions:** Once the model is trained, use it to make predictions on the testing data. The goal is to evaluate how well the model performs on data it hasn't seen before.
5. **Evaluate the model:** Calculate various evaluation metrics to measure the performance of the model. Some common metrics include mean squared error (MSE), mean absolute error (MAE), and R-squared.
6. **Improve the model:** If the model's performance is not satisfactory, you can try improving it by adding or removing predictor variables, adjusting the model's hyperparameters, or using a different algorithm altogether.
7. **Use the model to make predictions on new data:** Once you are satisfied with the model's performance, you can use it to make predictions on new data.

These are the general steps for implementing the Multiple Linear Regression algorithm [17].

4.2 Auto-Regressive Integrated Moving Average (ARIMA):

The Auto-Regressive Integrated Moving Average (ARIMA) algorithm is a widely used time series forecasting model that incorporates both autoregression and moving average techniques. Here are the general steps to implement an ARIMA model:

1. **Stationarity Check:** Check if the time series data is stationary. Stationarity means the statistical properties of the time series do not change over time, i.e., mean, variance, and covariance remain constant over time.

2. **Differencing:** If the data is not stationary, perform differencing to make the data stationary. Differencing means taking the difference between consecutive observations in the time series.
3. **Autocorrelation and Partial Autocorrelation Analysis:** Determine the order of the Autoregressive (AR) and Moving Average (MA) terms in the ARIMA model. This can be done using autocorrelation and partial autocorrelation plots.
4. **Model Fitting:** Fit the ARIMA model to the data using the determined values of p, d, and q. This involves estimating the coefficients of the model.
5. **Model Validation:** Check the accuracy of the ARIMA model by comparing the predicted values with the actual values using metrics such as Mean Squared Error (MSE) and Root Mean Squared Error (RMSE).
6. **Forecasting:** Use the ARIMA model to forecast future values of the time series.

Overall, the ARIMA algorithm involves a combination of data preparation, statistical analysis, and model fitting to produce accurate time series forecasts [18].

4.3 Support Vector Regression (SVR):

Support vector regression (SVR) is a supervised machine learning algorithm that can be used for regression tasks. It is a variant of Support Vector Machines (SVM) and uses the same basic principles of finding a hyperplane that separates the data into two classes. However, unlike SVM, which is used for classification tasks, SVR is used for regression tasks where the goal is to predict a continuous target variable. Here are the main steps involved in implementing SVR [19]:

1. **Data collection:** Collect data on the target variable and predictor variables.
2. **Data preparation:** Preprocess the data by cleaning and normalizing it, and split it into training and testing sets.
3. **Model training:** Train the SVR model using the training data. The SVR algorithm finds a hyperplane in a high-dimensional space that is close to as many data points as possible while also having a maximum margin from the data points. This hyperplane is then used to make predictions on new data.
4. **Model evaluation:** Evaluate the performance of the trained SVR model on the testing data. The evaluation metrics typically used for SVR are Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.
5. **Model tuning:** Tune the hyperparameters of the SVR model to improve its performance. The most important hyperparameters in SVR are the kernel type, regularization parameter, and kernel function parameters.
6. **Model deployment:** Use the trained SVR model to make predictions on new data.

SVR is a powerful algorithm for regression tasks, especially when dealing with nonlinear data. However, it can be computationally expensive and requires careful tuning of its hyperparameters.

4.4 Linear Regression (LR):

Linear regression is a supervised machine learning algorithm used for predicting a continuous target variable based on one or more predictor variables. Here are the basic steps involved in linear regression [20]:

1. **Data Collection:** Collect data on the target variable and predictor variables.
2. **Data Preparation:** Clean and preprocess the data, and split it into training and testing sets.
3. **Model Training:** Train the linear regression model using the training data. In linear regression, the model learns the weights (coefficients) for each predictor variable to minimize the difference between the predicted and actual target values.

4. **Model Evaluation:** Evaluate the performance of the trained model on the testing data. The evaluation metrics typically used for linear regression are Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and R-squared.
5. **Model Deployment:** Use the trained model to make predictions on new data.

5. HYBRID METAHEURISTIC REGRESSION MODEL FOR HANDLING BIG DATA

A hybrid metaheuristic regression model is a machine learning model that combines the strengths of multiple optimization algorithms to achieve better accuracy and performance in regression tasks. Metaheuristics are optimization algorithms that do not guarantee global optimum solutions, but they are efficient in finding good solutions in complex search spaces. In a hybrid metaheuristic regression model, multiple metaheuristic algorithms are used in combination to optimize the model parameters and improve the model's performance. Some commonly used metaheuristic algorithms for regression tasks include genetic algorithms, particle swarm optimization, ant colony optimization, simulated annealing, and differential evolution.

The hybridization of these algorithms can be done in various ways, such as combining their search strategies, using their operators interchangeably, or using their results sequentially. The goal is to leverage the strengths of each algorithm and compensate for their weaknesses, leading to a more efficient and accurate regression model. Overall, a hybrid metaheuristic regression model can be a powerful tool for solving complex regression problems where traditional optimization methods may not be sufficient [21].

5.1 Hybrid Genetic Algorithm (GA) regression model

Here are the general steps involved in building a hybrid genetic algorithm regression model:

1. **Select the regression technique:** Decide on the regression technique to be used based on the problem domain and data characteristics.
2. **Define the objective function:** Define an objective function that represents the fitness of the model on a validation set. This objective function should be optimized using the genetic algorithm.
3. **Define the parameter space:** Define the parameter space that will be searched by the genetic algorithm. This can include the number of hidden layers and neurons in a neural network, the type of activation function to use, and the regularization parameter.
4. **Initialize the population:** Create an initial population of candidate solutions randomly, each representing a set of model parameters.
5. **Evaluate fitness:** Evaluate the fitness of each candidate solution by computing its predictive accuracy on a validation set using the objective function.
6. **Apply genetic operators:** Apply genetic operators such as selection, crossover, and mutation to generate new candidate solutions and improve the overall fitness of the population.
7. **Evaluate convergence:** Check for convergence of the population's fitness and stop the algorithm if a stopping criterion is met, such as a maximum number of iterations.
8. **Select the best solution:** Select the best solution from the final population as the solution to the problem.
9. **Evaluate the solution:** Evaluate the solution's predictive accuracy on a test set to assess its generalization performance.
10. **Tune hyperparameters:** Tune the hyperparameters of the model, such as the population size, crossover probability, and mutation probability, to improve its performance.

These steps provide a general framework for building a hybrid genetic algorithm regression model. However, the specific details of the model will depend on the problem being solved and the data characteristics [22].

5.2 Hybrid Simulated Annealing (SA) algorithms regression model steps

Here are the general steps involved in building a hybrid simulated annealing algorithms regression model:

1. **Select the Regression Technique:** Decide on the regression technique to be used based on the problem domain and data characteristics.
2. **Define the Objective Function:** Define an objective function that represents the fitness of the model on a validation set. This objective function should be optimized using the simulated annealing algorithm.
3. **Define the Parameter Space:** Define the parameter space that will be searched by the simulated annealing algorithm. This can include the number of hidden layers and neurons in a neural network, the type of activation function to use, and the regularization parameter.
4. **Initialize the Solution:** Create an initial solution randomly, representing a set of model parameters.
5. **Evaluate fitness:** Evaluate the fitness of the initial solution by computing its predictive accuracy on a validation set using the objective function.
6. **Apply Simulated Annealing:** Apply the simulated annealing algorithm to generate new candidate solutions and improve the overall fitness of the solution. This involves iteratively perturbing the current solution and accepting or rejecting the new solution based on a probability distribution that depends on the current temperature.
7. **Evaluate convergence:** Check for convergence of the solution's fitness and stop the algorithm if a stopping criterion is met, such as a maximum number of iterations or the convergence of the solution's fitness.
8. **Select the best solution:** Select the best solution from the final iterations as the solution to the problem.
9. **Evaluate the solution:** Evaluate the solution's predictive accuracy on a test set to assess its generalization performance.
10. **Tune hyperparameters:** Tune the hyperparameters of the model to improve its performance.

These steps provide a general framework for building a hybrid simulated annealing algorithms regression model. However, the specific details of the model will depend on the problem being solved and the data characteristics [23].

5.3 Hybrid Differential Evolution (DE) algorithms regression model

Here are the general steps involved in building a hybrid differential evolution algorithms regression model:

1. **Select the regression technique:** Decide on the regression technique to be used based on the problem domain and data characteristics.
2. **Define the objective function:** Define an objective function that represents the fitness of the model on a validation set. This objective function should be optimized using the differential evolution algorithm.
3. **Define the parameter space:** Define the parameter space that will be searched by the differential evolution algorithm. This can include the number of hidden layers and neurons in a neural network, the type of activation function to use, and the regularization parameter.
4. **Initialize the population**: Create an initial population of candidate solutions randomly, each representing a set of model parameters.
5. **Evaluate fitness:** Evaluate the fitness of each candidate solution by computing its predictive accuracy on a validation set using the objective function.

6. **Apply differential evolution:** Apply the differential evolution algorithm to generate new candidate solutions and improve the overall fitness of the population. This involves iteratively creating new solutions by combining and mutating existing solutions.
7. **Evaluate convergence:** Check for convergence of the population's fitness and stop the algorithm if a stopping criterion is met, such as a maximum number of iterations or the convergence of the population's fitness.
8. **Select the best solution:** Select the best solution from the final population as the solution to the problem.
9. **Evaluate the solution:** Evaluate the solution's predictive accuracy on a test set to assess its generalization performance.
10. **Tune hyperparameters:** Tune the hyperparameters of the model, such as the population size, mutation rate, and crossover rate, to improve its performance.

These steps provide a general framework for building a hybrid differential evolution algorithms regression model. However, the specific details of the model will depend on the problem being solved and the data characteristics [24].

6. RESULT AND DISCUSSION

The methodology we use considers a wide range of data that was collected from several cities over a period of hours and days. The addition of this dataset will aid in the system's training. The recordings in this dataset came from a IoT gas sensor array that was purchased from the UCI Repository. It consists of one temperature sensor and eleven separate gas sensors. Using Jupyter notebook, we built various data analytics and machine learning algorithms in Python. The below results demonstrates that the relevant features taken into account for the prediction are correlated and can be used to train the model.

6.1 Performance Measures

The factors included as predictors in the training dataset for the machine learning and statistical models in the air quality forecast are shown in Table 1. To evaluate the performance of the air quality forecast models using the testing dataset, model performance measures such as R^2 (R-squared), Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and systematic error (BIAS) were utilized. The equations are defined as follows [25]:

R-squared (R^2): It is a statistical measure used to evaluate how well a regression model fits the data. It is a value between 0 and 1 that indicates the proportion of the variance in the dependent variable (Y) that is explained by the independent variables (X) in the model. R^2 is calculated as the ratio of the explained variance (ESS) to the total variance (TSS) of the dependent variable:

R2 = ESS / TSS

Where: ESS (Explained Sum of Squares) is the sum of squared differences between the predicted values and the mean of the dependent variable. TSS (Total Sum of Squares) is the sum of squared differences between the actual values and the mean of the dependent variable. R2 ranges from 0 to 1, with a value of 1 indicating that the model perfectly fits the data and explains all the variation in the dependent variable, and a value of 0 indicating that the model does not explain any of the variation in the dependent variable.

Root Mean Square Error (RMSE): It is a widely used metric for evaluating the accuracy of a regression model. It measures the difference between the actual and predicted values of the dependent variable and provides a measure of the typical error of the model in predicting the target variable. RMSE is calculated as the square root of the mean of the squared differences between the actual and predicted values:

RMSE = sqrt(mean((Y - Y_pred)^2))

Where:

- Y is the actual value of the dependent variable.
- Y_pred is the predicted value of the dependent variable.
- mean() is the mean function that calculates the average of the squared differences.

RMSE is expressed in the same units as the dependent variable and provides a measure of the average magnitude of the errors in the predictions. A lower value of RMSE indicates a better fit of the model to the data.

Mean Absolute Error (MAE): It is a metric used to evaluate the accuracy of a regression model. It measures the average absolute difference between the actual and predicted values of the dependent variable and provides a measure of the typical magnitude of the errors in the predictions.MAE is calculated as the mean of the absolute differences between the actual and predicted values:

MAE = mean(abs(Y - Y_pred))

Where:

- Y is the actual value of the dependent variable.
- Y_pred is the predicted value of the dependent variable.
- mean() is the mean function that calculates the average of the absolute differences.

MAE is expressed in the same units as the dependent variable and provides a measure of the average magnitude of the errors in the predictions. Unlike RMSE, MAE does not give more weight to large errors and is less sensitive to outliers.

Systematic Error (BIAS): In statistics, a systematic error, also known as bias, is an error in measurement that is consistent and predictable in a particular direction. It affects the accuracy and precision of a measurement or estimate, and it can result from various sources such as faulty equipment, flawed experimental design, or human error. Systematic errors can be either positive or negative, depending on whether they cause the measured values to be higher or lower than the true values. It is important to identify and minimize systematic errors in order to obtain reliable and valid results [25].

Following this, machine learning techniques are used for the categorization process. MLR, ARIMA, SVR and LR are only some of the effective machine learning regression classification approaches that are brought in for the classification process, as shown by the results of the metaheuristic based ML models illustrated below.

Table 1. Performance Analysis with Machine learning Algorithms

Algorithm Details	**Performance Metrics**			
	R2	RMSE	MAE	BIAS
MLR	0.94	7.8	4.3	0.95
ARIMA	0.98	6.8	3.9	0.89
SVR	0.99	7.5	4.3	0.97
LR	0.93	7.3	4.4	0.96

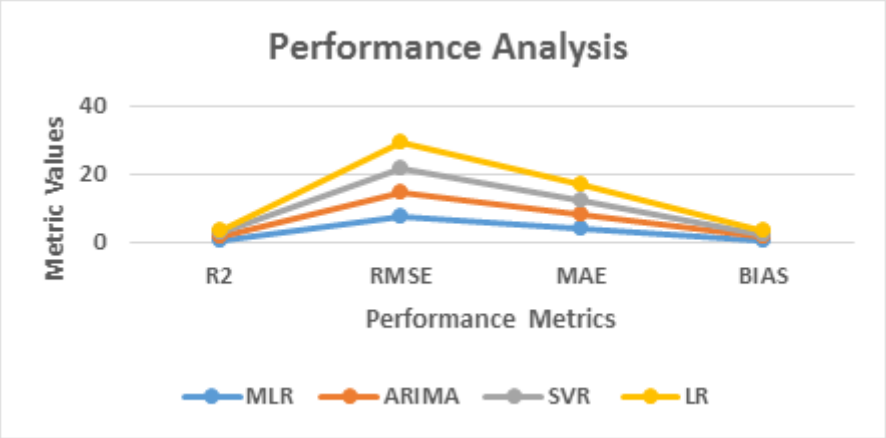

Figure 2: Performance Analysis with Machine learning Algorithms

Table 2. Performance Analysis of Machine learning Algorithms

Algorithm Details	Performance Metrics			
	R^2	RMSE	MAE	BIAS
GA-MLR	0.92	7.3	4.3	0.93
GA-ARIMA	0.90	6.6	3.8	0.87
GA-SVR	0.91	7.3	4.2	0.93
GA-LR	0.90	7.6	4.3	0.94
SA-MLR	0.91	7.2	4.1	0.92
SA-ARIMA	0.89	6.5	3.5	0.88
SA-SVR	0.90	7.0	3.9	0.89
SA-LR	0.89	7.5	4.1	0.93
DE-MLR	0.89	7.1	4.0	0.91
DE-ARIMA	0.88	6.3	3.2	0.85
DE-SVR	0.86	6.8	3.5	0.88
DE-LR	0.81	7.3	3.4	0.92

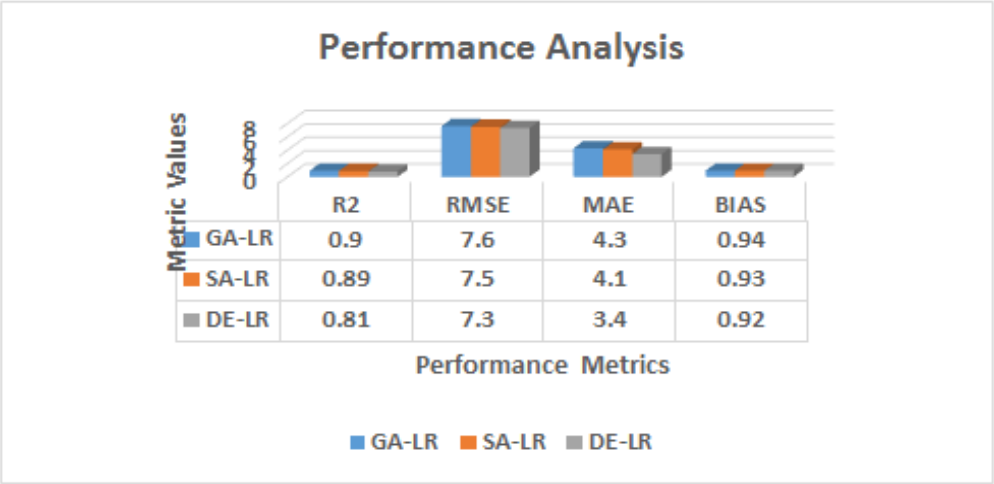

Figure 3: Performance Analysis of Machine learning Algorithms

The above table and figures shows that machine learning techniques such as MLR, ARIMA, SVR and LR are used for classification process, as shown by the results of the metaheuristic based ML models that is DE-LR algorithms shows better results when compared to others.

7. CONCLUSION

Vehicle emissions and industrial equipment that produce harmful pollutants such as particulate matter, carbon monoxide, and ozone pose a major health risk to humans are the primary sources of the present increase in air pollution in metropolitan areas. With the advancement of big data technologies, and machine learning techniques, real-time air quality monitor and evaluation is desirable for future smart cities. A range of multidimensional factors, such as location, time, season, and many more, determine the qualities of air. Because many specialists have recently employed big data analytics and machine learning regression approaches to consider, assess, and predict air quality, the development of an affordable, effective air quality monitoring and forecasting system that gathers data and delivers air pollution evaluations is required. The proposed LR models were trained and their parameters were modified using the DE strategy to achieve the best prediction accuracy.

8. ACKNOWLEDGEMENT

The author is grateful to the Management and Principal for their motivation and constant support. This research has been supported by the grant obtained under the scheme of Seed Money for Research Projects from Cauvery College for Women (Autonomous), Tiruchirappalli – 620 018, India.

9. REFERENCES

1. WHO. World Health Statistics 2021: Monitoring Health for the SDGs, Sustainable Development Goals; WHO: Geneva, Switzerland, 2021
2. Zaheer, J.; Jeon, J.; Lee, S.-B.; Kim, J.S. Effect of Particulate Matter on Human Health, Prevention, and Imaging Using PET or SPECT. Prog. Med. Phys. 2018, 29, 81.
3. Dantas, G.; Siciliano, B.; França, B.B.; da Silva, C.M.; Arbilla, G. The impact of COVID-19 partial lockdown on the air quality of the city of Rio de Janeiro, Brazil. Sci. Total Environ. 2020, 729, 139085.
4. Zambrano-Monserrate, M.A.; Ruano, M.A.; Sanchez-Alcalde, L. Indirect effects of COVID-19 on the environment. Sci. Total Environ. 2020, 728, 138813.
5. Fan, K.; Dhammapala, R.; Harrington, K.; Lamastro, R.; Lamb, B.; Lee, Y. Development of a Machine Learning Approach for Local-Scale Ozone Forecasting: Application to Kennewick, WA. Front. Big Data 2022, 5, 781309.
6. Saheer, L.B.; Bhasy, A.; Maktabdar, M.; Zarrin, J. Data-Driven Framework for Understanding and Predicting Air Quality in Urban Areas. Front. Big Data 2022, 5, 822573.
7. hau, P.N.; Zalakeviciute, R.; Thomas, I.; Rybarczyk, Y. Deep Learning Approach for Assessing Air Quality During COVID-19. Lockdown in Quito. Front. Big Data 2022, 5, 842455.
8. Leong, W.C.; Kelani, R.O.; Ahmad, Z. Prediction of air pollution index (API) using support vector machine (SVM). J. Environ. Chem. Eng. 2020, 8, 103208.
9. Doreswamy; Harishkumar, K.S.; Km, Y.; Gad, I. Forecasting Air Pollution Particulate Matter (PM2.5) Using Machine Learning Regression Models. Procedia Comput. Sci. 2020, 171, 2057–2066.
10. Liang, Y.C.; Maimury, Y.; Chen, A.H.L.; Juarez, J.R.C. Machine learning-based prediction of air quality. Appl. Sci. 2020, 10, 9151.
11. Martínez, N.M.; Montes, L.M.; Mura, I.; Franco, J.F. Machine Learning Techniques for PM 10 Levels Forecast in Bogotá. In Proceedings of the 2018 ICAI Workshops (ICAIW), Bogota, Colombia, 1–3 November 2018.

12. Juarez, E.K.; Petersen, M.R. A Comparison of Machine Learning Methods to Forecast Tropospheric Ozone Levels in Delhi. Atmosphere 2022, 13, 46.
13. Su, Y. Prediction of air quality based on Gradient Boosting Machine Method. In Proceedings of the 2020 International Conference on Big Data and Informatization Education (ICBDIE), Zhangjiajie, China, 23–25 April 2020; pp. 395–397.
14. De Oliveira, R.C.G.; Cunha, C.L.; Tôrres, A.R.; Corrêa, S.M. Forecasts of tropospheric ozone in the Metropolitan Area of Rio de Janeiro based on missing data imputation and multivariate calibration techniques. Environ. Monit. Assess. 2021, 193, 531.
15. Nahar K, Ottom MA, Alshibli F, Shquier MA (2020) Air quality index using machine learning—a jordan case study. COMPUSOFT, Int J Adv Comput Technol 9(9):3831–3840.
16. O. BOUAKLINE et al., "Prediction of daily PM10 concentration using machine learning," 2020 IEEE 2nd International Conference on Electronics, Control, Optimization and Computer Science (ICECOCS), Kenitra, Morocco, 2020, pp. 1-5.
17. J*, Ms. S., & Janita, Dr. S. (2019). Opinion Based Memory Access Algorithms using Collaborative Filtering in Recommender Systems. In International Journal of Innovative Technology and Exploring Engineering (Vol. 8, Issue 12, pp. 606–612). Blue Eyes Intelligence Engineering and Sciences Engineering and Sciences Publication - BEIESP. https://doi.org/10.35940/ijitee.l3274.1081219
18. J. K. Sethi and M. Mittal, "Analysis of Air Quality using Univariate and Multivariate Time Series Models," 2020 10th International Conference on Cloud Computing, Data Science & Engineering (Confluence), Noida, India, 2020, pp. 823-827, doi: 10.1109/Confluence47617.2020.9058303.
19. M. Kulkarni, A. Raut, S. Chavan, N. Rajule and S. Pawar, "Air Quality Monitoring and Prediction using SVM," 2022 6th International Conference On Computing, Communication, Control And Automation (ICCUBEA, Pune, India, 2022, pp. 1-4.
20. S. B. Sonu and A. Suyampulingam, "Linear Regression Based Air Quality Data Analysis and Prediction using Python," 2021 IEEE Madras Section Conference (MASCON), Chennai, India, 2021, pp. 1-7.
21. Hamzah A. Alsayadi1, Abdelaziz A. Abdelhamid, El-Sayed M. El-Kenawy, Abdelhameed Ibrahim, Marwa M. Eid. Improving the Regression of Air Quality Using Ensemble of MachineLearning Models.
22. Packiam, R. Merlin, and V. Sinthu Janita Prakash. "A Novel Integrated Framework Based on Modular Optimization for Efficient Analytics on Twitter Big Data." Information and Communication Technology for Intelligent Systems (2018): 213-224.
23. Journal of Artificial Intelligence and Metaheuristics (JAIM) Vol. 01, No. 02, PP. 08-16, 2022.
24. F. C. Tian, C. Kadri, L. Zhang, J. W. Feng, L. H. Juan and P. L. Na, "A Novel Cost-Effective Portable Electronic Nose for Indoor-/In-Car Air Quality Monitoring," 2012 International Conference on Computer Distributed Control and Intelligent Environmental Monitoring, Zhangjiajie, China, 2012, pp. 4-8.
25. Wei, H.; Li, S.; Jiang, H.; Hu, J.; Hu, J. Hybrid Genetic Simulated Annealing Algorithm for Improved Flow Shop Scheduling with Makespan Criterion. Appl. Sci. 2018, 8, 2621. Available from:
26. L. Cui et al., "Differential Evolution Algorithm With Tracking Mechanism and Backtracking Mechanism," in IEEE Access, vol. 6, pp. 44252-44267, 2018.
27. Lei, T.M.T.; Siu, S.W.I.;Monjardino, J.; Mendes, L.; Ferreira, F. Using Machine Learning Methods to Forecast Air Quality: A Case Study in Macao. Atmosphere 2022, 13, 1412.