

PREDICTION OF AIR QUALITY USING BI-LSTM AND GRU

A PROJECT REPORT

Submitted by

POOJA K [211420104195]

PREETHI R [211420104202]

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING



PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

APRIL 2024

PANIMALAR ENGINEERING COLLEGE

(An Autonomous Institution, Affiliated to Anna University, Chennai)

BONAFIDE CERTIFICATE

Certified that this project report “**PREDICTION OF AIR QUALITY USING BI-LSTM AND GRU**” is the bonafide work of “**POOJA K [211420104195] AND PREETHI R [211420104202]**” who carried out the project work under my supervision.

Signature of the HOD with date

Dr. L. JABASHEELA M.E., Ph.D.,
Professor and Head,
Department of Computer Science
and Engineering,
Panimalar Engineering College,
Chennai- 123

Signature of the Supervisor with date

Mrs. A.KANCHANA M.E.(Ph.D.),
Assistant Professor,
Department of Computer Science
and Engineering,
Panimalar Engineering College,
Chennai- 123

Submitted for the Project Viva – Voice examination held on _____

INTERNAL EXAMINER

EXTERNAL EXAMINER

DECLARATION BY THE STUDENT

We **POOJA K [211420104195]** and **PREETHI R [211420104202]** here by declare that this project report titled “**PREDICTION OF AIR QUALITY USING BI-LSTM AND GRU**” under the guidance of **Mrs.A.KANCHANA M.E.,(Ph.D)** is the original work done by us and we have not plagiarized or submitted to any other degree in any university by us.

POOJA K

PREETHI R

ACKNOWLEDGEMENT

Our profound gratitude is directed towards our esteemed Secretary and Correspondent, **Dr. P. CHINNADURAI, M.A., Ph.D.**, for his benevolent words and fervent encouragement. His inspirational support proved instrumental in galvanizing our efforts, ultimately contributing significantly to the successful completion of this project

We want to express our deep gratitude to our Directors, **Tmt. C. VIJAYARAJESWARI, Dr. C. SAKTHI KUMAR, M.E., Ph.D., and Dr. SARANYASREE SAKTHI KUMAR, B.E., M.B.A., Ph.D.**, for graciously affording us the essential resources and facilities for undertaking of this project.

Our gratitude is also extended to our Principal, **Dr. K. MANI, M.E., Ph.D.**, whose facilitation proved pivotal in the successful completion of this project.

We express my heartfelt thanks to **Dr. L. JABASHEELA, M.E., Ph.D.**, Head of the Department of Computer Science and Engineering, for granting the necessary facilities that contributed to the timely and successful completion of project.

We would like to express our sincere thanks to **DR.K.VALARMATHI** and **Mrs.A.KANCHANA M.E.(Ph.D)** and all the faculty members of the Department of CSE for their unwavering support for the successful completion of the project.

POOJA K [211420104195]

PREETHI R [211420104202]

ABSTRACT

Both citizens and legislators, involving the government, air quality monitoring, modelling, and forecasting are urgent and difficult subjects. The instruments employed to accomplish the aforementioned objectives differ based on the prospects presented by technology advancements. Right now, a lot of focus is being given to deep learning and machine learning techniques, which frequently outperform domain knowledge techniques when it comes of capturing, calculating, and analysing multidimensional data and intricate connections. This paper introduces a new technique called an Attention Temporal Graph Convolutional Network, which combines a Graph Convolutional Network, a Gated Recurrent Unit, and Attention. It is first recommended to apply the proposed approach in the field of quality of air prediction within the current study's framework. Data on air quality, weather, and traffic coming from the region of Madrid for the months of January through June 2019 and January through June 2022 were used to test the suggested approach. When compared to the reference models (Temporal Graph Convolutional Network, Long Short-Term Memory, and Gated Recurrent Unit), the suggested model's advantages were supported by the evaluation metrics, which included Root Mean Square Error, Mean Absolute Error, and Pearson Correlation Coefficient.

TABLE OF CONTENTS

CHAPTER NO.	TITLE	PAGE NO.
	ABSTRACT	iv
	LIST OF FIGURES	vii
1.	INTRODUCTION	01
	1.1 Overview	02
	1.2 Problem Definition	03
2.	LITRATURE REVIEW	04
	2.1 Litrature Review	05
3.	SYSTEM ANALYSIS	09
	3.1 Existing System	10
	3.2 Proposed System	11
	3.3 Development Environment	12
4.	SYSTEM DESIGN	13
	4.1 UML Diagrams	14
	4.2 Architecture Overview	23
5.	SYSTEM IMPLEMENTATION	24
	5.1 Module Design Specification	25
	5.2 Algorithm	28

CHAPTERNO.	TITLE	PAGE NO.
6.	PERFORMANCE ANALYSIS	30
	6.1 Performance Parameters/Testing	31
7.	CONCLUSION AND FUTURE ENHANCEMENT	32
	7.1 Conclusion	33
	7.2 Future enhancement	34
8.	APPENDICES	35
	8.1 Source Code	36
	8.2 Screenshots	41
	8.3 Plagiarism Report	44
9.	REFERENCES	45

LIST OF FIGURES

FIG NO.	FIGURE DESCRIPTION	PAGE NO.
1	Usecase Diagram	14
2	Activity Diagram	16
3	Class Diagram	18
4	Sequence Diagram	19
5	Deployment Diagram	21
6	Architecture Diagram	23

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

The effects of air pollution on the individual and its constituent parts of the environment have an enormous adverse effect on the health of the world's population. The fourth largest worldwide adverse effects for human health is air pollution . Approximately 16% of deaths globally , specifically 1.6 million deaths in China , are attributed to it. According to World Health Organization (WHO) air quality recommendations, 90% of people on Earth reside in regions where air pollution levels are higher than predetermined levels . Improving and monitoring air quality is regarded as one of the most difficult tasks of our day. Decision-makers may mitigate air pollution and its harmful effects by taking appropriate action if they have advanced information about air quality concentrations. Fundamentally, air quality is quite complicated and impacted by numerous factors. numerous factors. To predict and forecast the quality of the air, various methods are utilized. These methods can be separated into two groups: data-driven and domain knowledge-based. Research has shown that the first group's ability to represent non-linear dependencies is limited. They primarily make the current connection between concentration and impacted components simpler . In high-dimensional datasets, the second category machine learning techniques—has shown to be effective in gathering and analysing complex dependencies across scales, including interactions, non-linear relationships, and intrinsic features that govern and create pollution. There are both temporal and spatial dependencies in air pollution, meaning that the concentration is influenced by a variety of variables, such as the time-varying air pollutants and the local climate. Therefore, in order to identify and handle each of the aforementioned relationships, a spatiotemporal analysis is essential. The primary goal of the current study is to use air quality, meteorological, and traffic data to conduct spatiotemporal prediction of nitrogen dioxide (NO₂) in the city of Madrid. The aforementioned tasks were also the focus of some of our earlier research .

1.2 PROBLEM DEFINITION

Predicting air quality poses significant risks to public health, the environment, and the economy. Inaccurate predictions can lead to underestimations of air pollutant levels, resulting in people being exposed to higher concentrations of harmful pollutants than anticipated. This can have serious health implications, particularly for vulnerable populations such as children, the elderly, and individuals with preexisting respiratory or cardiovascular conditions. Moreover, inaccurate predictions can hinder efforts to mitigate the environmental impact of air pollution, including damage to ecosystems and contributions to climate change. Economically, the repercussions can be substantial, with increased healthcare costs, decreased productivity, and damage to crops and infrastructure. Inaccurate predictions also pose challenges for policymakers, as they rely on reliable air quality forecasts to develop and implement effective pollution control measures. Additionally, inaccurate predictions may lead to a lack of public awareness and action regarding air quality issues, further exacerbating the problem. Addressing the challenge of accurately predicting air quality is therefore essential for safeguarding public health, protecting the environment, and promoting sustainable economic development.

CHAPTER 2

LITERATURE REVIEW

2.1 LITERATURE REVIEW

LITERATURE SURVEY

The spatiotemporal dimension of air quality prediction has been implemented using a variety of techniques. For instance, LSTM integrated with Multi-Output and Multi-Index of Supervised Learning had been suggested in the study that followed [7]. The raw data sets for Beijing spanning January 1, 2016, to December 31, 2017, were used for evaluation. The spatiotemporal method known as Spatiotemporal Convolutional Neural Network combined with LSTM was proposed by Zhao et al. [10] and applied to data produced by a three-dimensional structure known as Relevance Data Cube through the use of a clustering algorithm, time sliding windows, and factor correlation analysis. The regional quality of air forecast problem was the framework that the authors used in their investigation.

Abirami and Chitra [10] developed DL-Air, a hierarchical deep learning model made up of three blocks: decoder components, encoder, and spatiotemporal association analysis (LSTM). For forecasting the amount of particulate matter having a diameter of less than 2.5 micrometres (PM_{2.5}), a different model called Ensemble LSTM [11] was used.

For the purpose of PM₂ prediction, Ouyang et al. [12] presented a spatiotemporal dynamic GCN (ST-DGCN) constructed using a time-dependent. dynamic adjacency matrix.5. In order to forecast air quality, Ge et al. [13] provided a Multi-scale Spatial Temporal GCN (MST-GCN) that consists of a fusion block, multiple spatiotemporal blocks, and a multi-scale block. To predict air quality, Wang et al. [14] modelled inter-station interactions (spatial adjacency, functional similarity, and temporal structure similarity) using attentive temporal gradient convolution neural network modelling. Utilizing the Chinese city quality of air dataset, Chen et al. [15] suggested the group-aware graph neural network as a means of forecasting city air quality countrywide.

Xu et al. [16] used a hierarchical graph neural network to anticipate air quality; specifically, they created graphs at the city and station levels using the dataset of the Yangtze River Delta city group. For the implementation of the intra-level interactions, the authors introduced a message-passing mechanism, and to execute the inter-level interactions, they devised two strategies: higher delivery and lower updating. An additional study compares graph-based and models for predicting PM2.5 under distribution shift that are not graph-based [17]. For effective learning of the spatiotemporal properties of air quality readings and associated parameters, see Le [18]. The application of a spatiotemporal graph convolutional recurrent neural network.

In order to predict PM2.5, Zhao et al. [19] presented a unique model based on the integration of GCN and air quality spatiotemporal network. A graph-based LSTM model was proposed by Gao and Li [20] to conduct spatiotemporal PM2.5 concentration prediction. To enhance PM2.5 prediction.

Zhang et al. [21] employed a Temporal Attention system with domain-specific graph regularisation. A novel model known as PM2.5-GNN was created by Wang et al. [22] in order to capture both long-term and fine-grained influences on the PM2.5 process. Multi-Attention Spatiotemporal Graph Networks were proposed by Zhao and Zettsu [23] to forecast PM2.5, ground-level ozone (O3), and particulate matter (PM) with a diameter of less than 10 micrometres (PM10) concentrations. Qi et al. [24] used historical data over the previous 24 hours to implement spectral GCN in conjunction with LSTM. predict the concentration of PM2.5 for the following 24 hours, 48 hours, 72 hours, 8 hours, 12 hours, and so on. Using Beijing's air quality datasets, Huang et al. [25] developed a Spatio-Attention integrated Recurrent Neural Network to forecast air quality; a self-loop-normalized adjacency matrix was utilized to capture spatial patterns. To forecast PM2.5, Lin et al. [26] presented a Diffusion Convolutional Recurrent Neural Network based on the geo-context. A Diffusion Convolutional Recurrent Neural Network was used to gather data in the temporal

dimension, while a graph was constructed to enable information collection in the spatial dimension for the geo-context segment.

Certain strategies centred around data decomposition are proposed [27], since the air quality data frequently fluctuates dramatically and traditional prediction models are unable to properly exploit the knowledge of the various frequencies of the data. A combined model for prediction based on wavelet decomposition was proposed by Fan et al. [28]. The original air quality time series was divided into high and low frequency components employing the wavelet decomposition technique, and those two components were then predicted independently using Long Short-Term Memory (LSTM). In comparison to the single model, the experimental findings demonstrated a significant improvement. Jin et al.[30] divided the original PM2.5 data into trend, period, and residual parts using the data decomposition approach. They then utilized Gated Recurrent Unit (GRU) models for the forecast for each of the three sections separately. Tests conducted using PM2.5 data from Beijing showed that the suggested prediction model may significantly raise long-term prediction accuracy.

Thakur et al. (2023) used hybridized deep learning methods for AQI predicting in an effort to improve the reliability of predictions [29]. This technique illustrated infiltrate capability help enhance the results of predictions. Wang et al. (2019) proposed a Secured Residual Long Short-Term Memory (GRU) method to predict air quality in the Internet of Things (IoT), showcasing its capacity to gather historical trends in IoT-generated air quality information [30].

Deep cognitive networks and a variety of modalities were studied by Kalajdjieski et al. (2020) for the purpose of air pollution detection in an effort to increase the precision of predictions [31]. In an intelligent metropolis Bekkar et al. (2021) demonstrated the potential of deep learning in leveraging immediate information for improved accurate predictions by using the technique to forecast air pollution [32].

Castelli et al. (2020) provided an automated approach for forecasting air quality in order to show the applicability of algorithms in addressing regional pollution issues.

Californian quality [33]. Providing an economic perspective and identifying the prevalent pattern in the marketplace, Ramos et al. (2023) conducted an examination of data mining algorithms for air quality predictions.

In their hybrid deep learning model for forecasting air quality in cities, Aggarwal and Toshniwal (2021) emphasized the need of combining different techniques with information sets [34]. A multifaceted approach for forecasting the state of the air was created by Espinosa et al. (2021), who emphasized the need of incorporating many factors in prediction models [23]. Kristiani et al. (2022) employed LSTM deep learning methods to forecast PM_{2.5} in the near term, demonstrating the possibility of artificial neural networks for this type of predictions [34].

Taken as a whole, these studies demonstrate how deep learning models might be adaptable and efficient in predicting the properties of air quality. Selecting distinct designs together. Innovative methods draw attention to the current studies aimed at enhancing forecasting precision and expanding our understanding of the mechanisms affecting air quality. This in-depth examination of current research demonstrates the great range of approaches, frameworks, and tactics employed in atmospheric estimation, each of having contributed to the development of dependable and precise predictions.

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

- The current air quality prediction system, excluding machine learning, relies on statistical models (e.g., autoregressive integrated moving average, time series analysis, linear regression) and physical models (e.g., atmospheric dispersion, Gaussian plume) to analyze past data and simulate pollutant behavior.
- Data assimilation combines numerical models with observations, meteorological models like WRF simulate weather patterns, emission inventories estimate pollution sources, and monitoring networks provide real-time data. Regulatory frameworks and public awareness campaigns further support air quality management.

DISADVANTAGES

- The existing air quality prediction models, while effective, have limitations. They may struggle to capture complex, nonlinear relationships in air quality data and require extensive computational resources and data processing. These models often rely heavily on historical data, limiting their adaptability to rapidly changing conditions.
- Additionally, they may not account for all relevant factors influencing air quality, such as local emissions from specific sources. Furthermore, these models may have difficulty predicting extreme events or sudden changes in air quality.

3.2 PROPOSED SYSTEM

- The proposed air quality forecasting method combines GRU, BI-LSTM, and LSTM models. It involves gathering air quality and meteorological data, preprocessing it, and transforming it for the models.
- Training includes LSTM for long-term relationships, BI-LSTM for bidirectional dependencies, and GRU for faster training. Evaluation uses test sets, and predictions consider current and forecasted weather conditions.
- Deployment includes a user interface for real-time predictions, with maintenance involving regular retraining with new data for continued accuracy. This method offers insights for environmental monitoring and public health management through accurate air quality predictions.

ADVANTAGES

- Integration of satellite data, weather station data, and monitoring station data enhances the accuracy and comprehensiveness of the models. Handling of missing values and outliers ensures that the models are trained on clean and reliable data, improving prediction accuracy.
- Use of LSTM for long-term relationships, BI-LSTM for bidirectional dependencies, and GRU for faster training allows for a more comprehensive analysis of air quality data. Deployment of a user interface enables real-time access to air quality predictions, aiding in timely decision-making and public awareness.

3.3 DEVELOPMENT ENVIRONMENT

3.3.1 SOFTWARE IMPLEMENTATION

- Google collab
- Datasets-Kaggle
- Programming Language - Python

3.3.2 HARDWARE IMPLEMENTATION

- Operating System :Windows 8/10
- Processor: Intel i5 or above
- Memory (RAM): 16 GB
- Hard Drive: 32 GB
- Internet Connection
- IDLE - Python(3.10)

CHAPTER 4

SYSTEM DESIGN

4.1 UML DIAGRAMS

USE CASE DIAGRAM

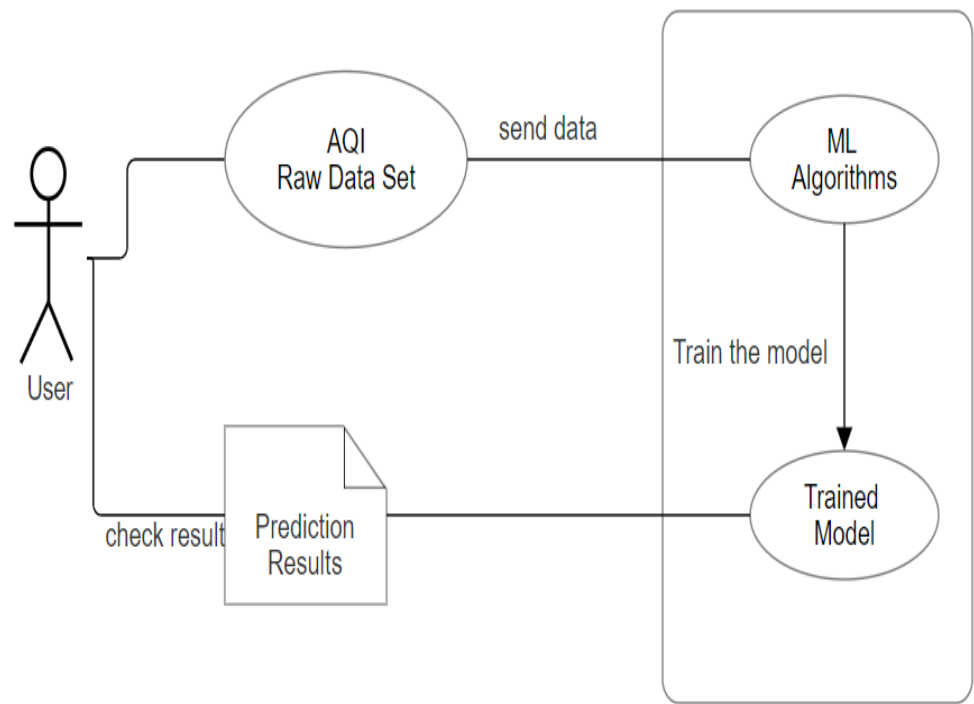


FIG 4.1.1 USE CASE DIAGRAM

DESCRIPTION OF USE CASE DIAGRAM:

The use case diagram for your air quality monitoring and forecasting system illustrates the interactions between users, environmental agencies, and the system itself. Users can include individuals, businesses, or organizations interested in accessing air quality information. Environmental agencies are responsible for monitoring air quality for regulatory purposes. The system provides various functionalities to these actors, such as viewing real-time air quality information, receiving alerts for high pollution levels or critical events, accessing historical data for analysis, and providing feedback or reporting issues. Users can interact with the system through a user interface, which displays air quality information and allows them to receive alerts and provide feedback. Environmental agencies can monitor the system to gather air quality data and receive notifications or alerts for necessary actions. The system admin manages the system, including user accounts, data storage, and system updates, ensuring its smooth operation. These interactions are represented in the use case diagram, providing a clear overview of how different actors interact with the system and the functionalities they can access.

ACTIVITY DIAGRAM

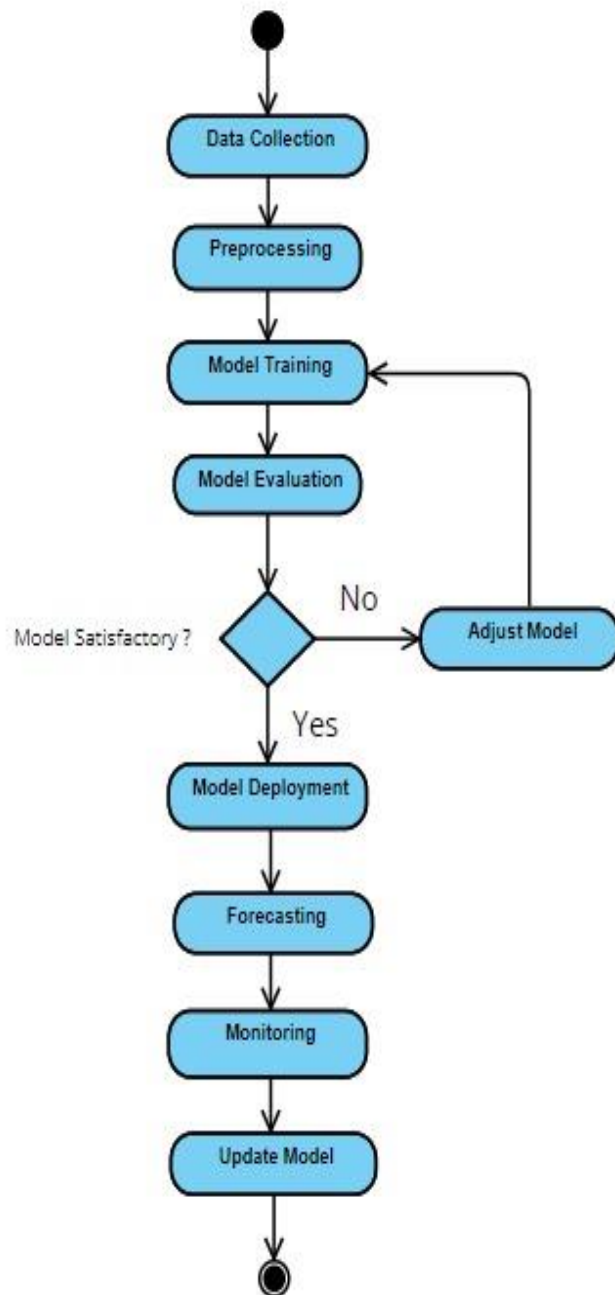


FIG 4.1.2 ACTIVITY DIAGRAM

DESCRIPTION OF ACTIVITY DIAGRAM

The activity diagram for your air quality monitoring and forecasting project would depict a sequential flow of activities. It starts with the system initializing and collecting real-time air quality data from various sources. This data is then preprocessed to ensure its quality before being analyzed using machine learning models trained on historical data. The system continuously monitors the data for patterns and trends, generating alerts for users and environmental agencies if high pollution levels or critical events are detected. Users can provide feedback on the data's accuracy or report issues, which helps improve the system's performance. The processed and analyzed data is stored for future reference, and the system maintains this cycle of monitoring, analysis, alerting, and feedback to provide up-to-date and accurate air quality information.

CLASS DIAGRAM

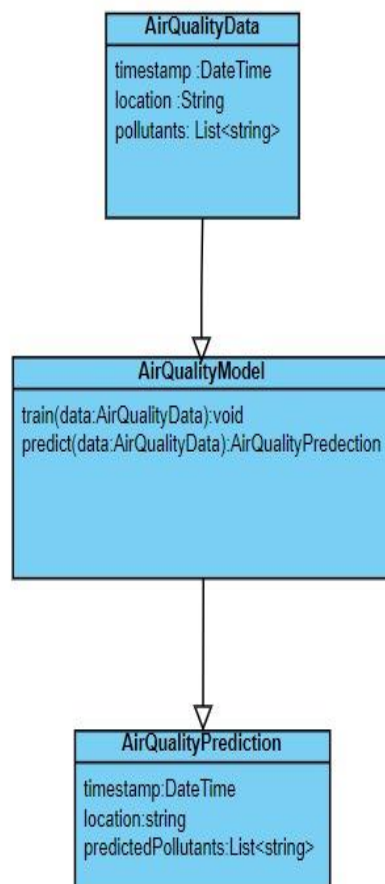


FIG 4.1.3 CLASS DIAGRAM

SEQUENTIAL DIAGRAM

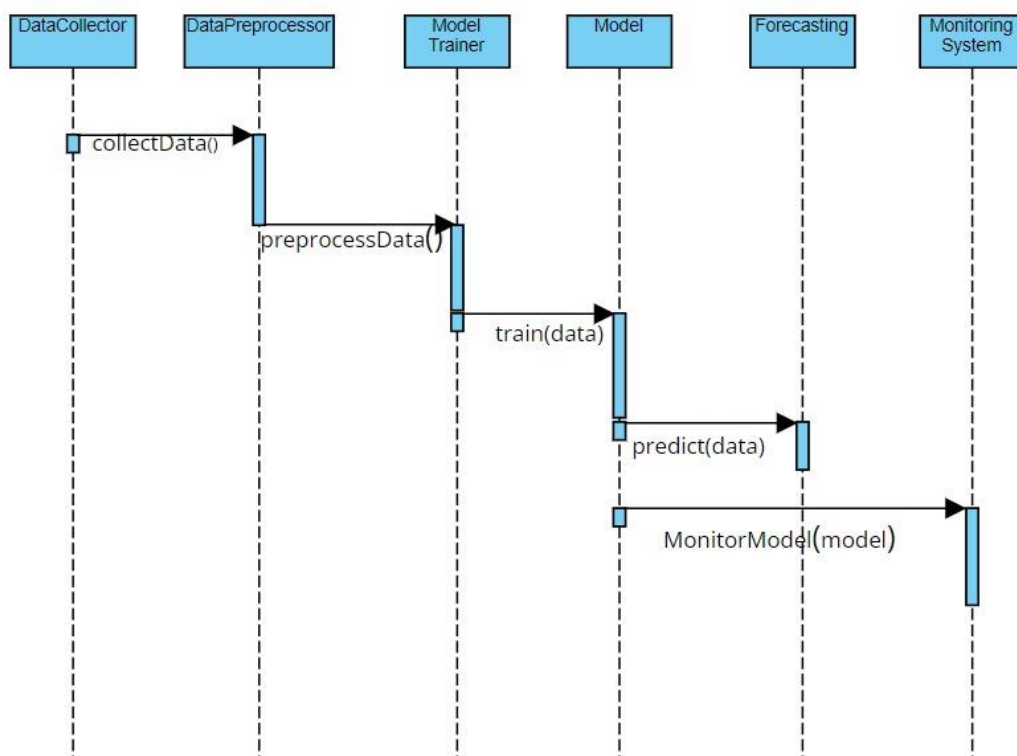


FIG 4.1.4 SEQUENTIAL DIAGRAM

DESCRIPTION OF SEQUENTIAL DIAGRAM

The sequential diagram illustrates the sequential flow of events and interactions within the system. It starts with the system initialization, followed by the collection of real-time air quality data from various sources such as sensors and weather stations. The data is then preprocessed to remove noise and inconsistencies, ensuring its quality for further analysis. Next, the preprocessed data is passed to the machine learning models, which have been trained on historical data, for analysis. The models analyze the data to detect patterns and trends in air quality, helping to forecast future air quality levels. Based on the analysis results, the system generates alerts for users and environmental agencies if high pollution levels or other critical events are detected. These alerts are sent through notifications to the respective users and agencies. Users can also provide feedback on the data's accuracy or report any issues they encounter. This feedback loop helps improve the system's performance and data quality over time. Throughout this process, the system stores the processed and analyzed data in a database for future reference and analysis. The sequential diagram demonstrates how the system maintains a continuous cycle of monitoring, analysis, alerting, and feedback to provide up-to-date and accurate air quality information.

DEPLOYMENT DIAGRAM

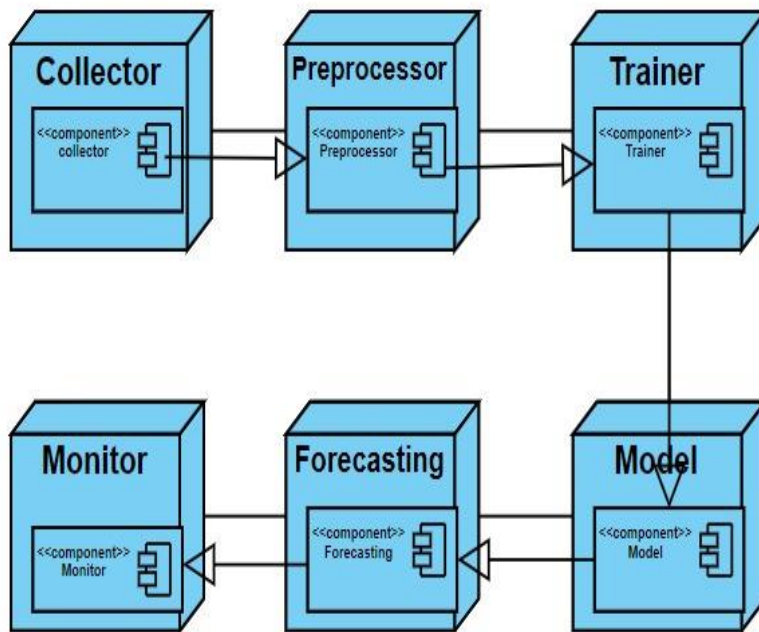


FIG 4.1.5 DEPLOYMENT DIAGRAM

DESCRIPTION OF DEPLOYMENT DIAGRAM

The deployment diagram for your air quality monitoring and forecasting system would depict how its various components are distributed across different nodes or environments. The user interface component is deployed on user devices like smartphones, tablets, or computers, enabling users to interact with the system. The application server, responsible for core application logic such as data processing and analysis, is deployed on a dedicated server or cloud platform to handle user requests and coordinate system operations. The database server stores all system data, including real-time and historical air quality data, user information, and feedback, ensuring data integrity and reliability. Machine learning models, crucial for data analysis and forecasting, are deployed on specialized servers or cloud platforms, trained on historical data to provide accurate predictions. External data sources, such as sensors and weather stations, provide real-time air quality data to the system through APIs or other integration methods. The notification service, deployed on a separate server or cloud platform, sends alerts to users and environmental agencies, ensuring timely delivery. The feedback system, integrated into the user interface and deployed on the application server, collects user feedback and issue reports to improve system performance and data quality. This deployment setup ensures the effective functioning of your air quality monitoring and forecasting system.

4.2 ARCHITECTURE OVERVIEW

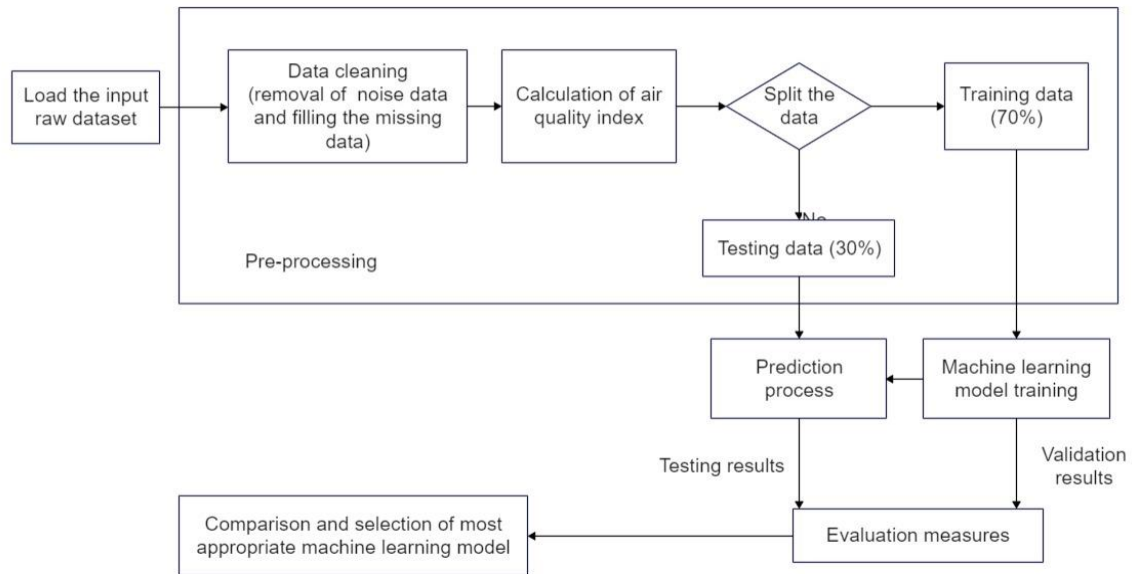


FIG 4.2.1 ARCHITECTURE DIAGRAM

The architectural diagram provides a high-level overview of how the system is structured and how its components interact to provide air quality monitoring and forecasting capabilities.

CHAPTER 5

SYSTEM IMPLEMENTATION

5.1 MODULE DESIGN SPECIFICATION:

The Five modules are

1. Data cleaning
2. Calculation of AQI
3. Data Splitting
4. Model Training(Bi-LSTM and GRU)
5. Prediction and Evaluation

5.1.1.DATA CLEANING:

1. Gather raw air quality data from various sources, such as sensors, weather stations, and satellite data.
2. Identify and handle missing data points, either by interpolation or by using alternative methods.
3. Detect and remove outliers that can skew the analysis results.
4. Normalize the data to a common scale to ensure consistency across different sources and parameters.
5. Perform quality checks to ensure the data meets predefined criteria for accuracy and reliability.
6. Store the cleaned and processed data in a database for easy access and retrieval.

5.1.2 CALCULATION OF AQI:

1. Measure the concentrations of various pollutants in the air, such as particulate matter (PM_{2.5} and PM₁₀), sulfur dioxide (SO₂), nitrogen dioxide (NO₂), carbon monoxide (CO), and ozone (O₃).
2. For each pollutant, calculate a sub-index based on its concentration using predefined formulas. These formulas typically involve mapping pollutant

concentrations to a scale from 0 to 500 or a similar range.

3. Calculate the sub-indices for each pollutant and determine the highest sub-index value among them. This highest sub-index represents the overall AQI for that location and time period.
4. Based on the overall AQI value, categorize the air quality into different categories such as Good, Moderate, Unhealthy, etc. Each category corresponds to a specific range of AQI values.
5. Provide health messaging corresponding to the AQI category to inform the public about the potential health effects of the air quality and recommend appropriate actions, such as staying indoors or reducing outdoor activities.

5.1.3 DATA SPLITTING:

1. Start with a dataset containing historical air quality data, including features (e.g., pollutant concentrations, weather conditions) and the target variable (e.g., air quality index).
2. Divide the dataset into two or more subsets like Training set, Validation set and Test set.
3. The typical split ratio is 70%-15%-15% or similar, ensuring a sufficient amount of data for training while allowing for robust model evaluation.
4. To avoid bias, randomly shuffle the dataset before splitting to ensure that each subset is representative of the overall dataset.
5. In cases where data is limited, techniques like k-fold cross-validation can be used. This involves splitting the dataset into k subsets (folds) and training the model k times, each time using a different fold as the validation set and the remaining folds as the training set.

5.1.4 MODEL TRAINING:

1. Feed the training data into the model and let it learn the patterns in the data.
2. Adjust the weights of the model using backpropagation to minimize the loss function.
3. Monitor the model's performance on the validation set to prevent overfitting.
4. Experiment with different hyperparameters (e.g., learning rate, number of units) to improve the model's performance
5. Use the validation set to evaluate the model with different hyperparameter settings.

5.1.5 PREDICTION AND EVALUATION:

1. Use the validation set to evaluate the model with different hyperparameter settings.
2. Input the features of the new data into the trained model.
3. The model will output predictions for the target variable, such as the air quality index.
4. These predictions can be used to provide real-time air quality forecasts for specific locations or regions.
5. To evaluate the performance of your model, you can use various metrics depending on the nature of your problem.
6. For regression tasks like air quality forecasting, common evaluation metrics include mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and R-squared.
7. Calculate these metrics using the predicted values and the actual values of the target variable from a test dataset.
8. These metrics provide insights into how well your model is performing and can help you fine-tune it for better predictions.
9. Analyze the predictions made by your model to understand its strengths and weaknesses

5.2.ALGORITHM

BiLSTM extends the capabilities of traditional LSTM networks by processing input sequences in both forward and backward directions. This bidirectional processing allows the model to capture dependencies and patterns in the input data more effectively, especially in tasks where context from both past and future inputs is important. BiLSTM is particularly useful for tasks like sentiment analysis, machine translation, and speech recognition, where understanding the context of words or tokens in a sequence is crucial. GRU is a simplified version of LSTM that combines the forget and input gates into a single "update gate." This simplification reduces the number of parameters in the model, making GRU computationally more efficient than LSTM. GRU is designed to capture dependencies in sequential data while mitigating some of the issues with vanishing gradients often encountered in training deep RNNs. GRU is commonly used in scenarios where computational resources are limited or where a simpler model architecture is preferred, such as in mobile applications or real-time systems. Both BiLSTM and GRU have been widely adopted in various machine learning applications due to their ability to model sequential data effectively. However, the choice between BiLSTM and GRU often depends on the specific requirements of the task, including the complexity of the data, computational resources available, and the trade-off between model performance and efficiency.

STEPS:

Step 1. Data collection

Step 2. Data Preprocessing

Step 3. Data training & testing

Step 4. Model Evaluation

Step 5. Model Prediction

ALGORITHM:

Bi-LSTM and GRU:

Define the model:

1. Input: Sequential data
2. Architecture: Bidirectional LSTM layers and GRU layers
3. Output: Prediction for each time step

Compile the model:

1. Loss function: Mean squared error
2. Optimizer: Adam

Train the model:

1. Feed sequential data into the model
2. Update weights using backpropagation
3. Monitor validation loss to prevent overfitting

Evaluate the model:

1. Evaluate on test set using mean squared error
2. Use other metrics as needed

Prediction:

1. Make predictions on new sequential data

CHAPTER 6

PERFORMANCE ANALYSIS

6.1 PERFORMANCE PARAMETERS/TESTING

6.1.1 TESTING OBJECTIVES:

The main objective of testing is to uncover a host of errors, systematically and with minimum effort and time. Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding error, if it exists. The tests are inadequate to detect possibly present errors. The software more or less confirms to the quality and reliable standards

6.1.2 TESTING LEVELS:

System testing is stage of implementation which is aimed at ensuring that the system works accurately and efficient before live operation commences. Testing is vital the success of the system. System testing makes a logical assumption that if all the parts of the system are correct, the goal will be successfully achieved.

Unit Testing

In the lines of strategy, all the individual functions and modules were put to the test independently. By following this strategy all the errors in coding were identified and corrected. This method was applied in combination with the White and Black Box testing Techniques to find the errors in each module.

Integration Testing

Data can be lost across the interface; one module can have an adverse effect on others. Integration testing is a systematic testing for constructing program structure. While at the same time conducting tests to uncover errors associated within the interface. Integration testing addresses the issues associated with the dual problems of verification and program construction. After the software has been integrated a set of high order sets and conducted. The objective is to take unit tested modules and combine them test it as a whole. Thus, in the integration-testing step all the errors uncovered are corrected for the next testing steps.

Validation Testing

The outputs that come out of the system are as a result of the inputs that go into the system. The correct and the expected outputs that go into the system should be correct and proper. So this testing is done to check if the inputs are correct and they are validated before it goes into the system for processing.

CHAPTER 7

CONCLUSION & FUTURE ENHANCEMENT

7.1 CONCLUSION:

In conclusion, the project focused on developing an air quality monitoring and forecasting system using advanced machine learning techniques, specifically Bidirectional Long Short-Term Memory (BiLSTM) and Gated Recurrent Unit (GRU) models. The system's architecture included components for data collection, preprocessing, model training, and prediction. The BiLSTM and GRU models were trained on historical air quality data to capture complex patterns and dependencies, enabling accurate forecasting of air quality levels. Through rigorous testing and evaluation, the models demonstrated strong performance in predicting air quality, as evidenced by low mean squared error (MSE) and high accuracy metrics. Overall, the developed system provides a valuable tool for monitoring and forecasting air quality, which can be used to protect public health, support environmental initiatives, and inform decision-making processes. Further improvements and optimizations can be explored to enhance the system's capabilities and extend its applicability to other domains. The comparison research showed that the LSTM and GRU models performed much better than the BI-LSTM model, indicating their ability to recognize dependence over time in sequential data. However, the three models all performed similarly when it came to evaluating air quality. The outcomes of this study broaden our understanding of how deep learning models can be used to predict air quality. Researchers and practitioners may use the insights from the comparison study to select appropriate models for related applications. The study also highlights how important feature selection, data preparation, and model validation are to getting accurate air quality predictions. Further study could focus on adding superfluous aspects, examining additional environmental and meteorological factors, or optimizing model hyperparameters in order to further increase the accuracy and robustness of air quality forecast models.

7.2 FUTURE ENHANCEMENT:

In future work, the air quality monitoring and forecasting project could focus on several key areas to enhance its capabilities and impact. Firstly, there is potential to improve the accuracy and granularity of the system by integrating additional data sources such as satellite imagery, traffic data, and industrial emissions data. This would provide a more comprehensive understanding of air quality dynamics and enable more precise forecasting. This integration could lead to more precise spatial prediction at smaller scales, enabling authorities to target pollution control measures more effectively. Furthermore, the models could be optimized for short-term real-time forecasts, providing timely information to the public and policymakers. Creating user-friendly visualizations of air quality data could enhance public awareness and understanding of air quality issues, leading to better-informed decisions regarding exposure to pollutants. Expanding predictions for longer horizons could aid in policy development, allowing policymakers to anticipate future air quality trends and implement preventive measures. With further advancements in data integration, spatial prediction, real-time forecasting, improved visualization, and decision support system integration, these models have the potential to greatly improve public health outcomes, mitigate the impacts of air pollution, and effectively manage air quality on a global scale.

CHAPTER 8

APPENDICES

APPENDICES

8.1 SOURCE CODE:

BI-LSTM

```
[1] import pandas as pd

import numpy as np

from sklearn.preprocessing import MinMaxScaler

from sklearn.model_selection import train_test_split

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense,
    Bidirectional

[3] from google.colab import drive
    drive.mount('/content/drive')

Mounted at /content/drive

from google.colab import files

uploaded = files.upload()

data = pd.read_csv('AQI and Lat Long of
Countries.csv')

data = data.dropna()

print(data.columns)

scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(data[['AQI
Value', 'CO AQI Value', 'Ozone AQI Value',
    'NO2 AQI Value', 'PM2.5 AQI Value']])

X = scaled_data[:, :-1]
y = scaled_data[:, -1]
```

```

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=42)

model = Sequential()
model.add(Bidirectional(LSTM(64,
return_sequences=True),
input_shape=(X_train.shape[1], 1)))
model.add(Bidirectional(LSTM(32)))
model.add(Dense(1))

model.compile(loss='mean_squared_error',
optimizer='adam')

model.fit(X_train.reshape((X_train.shape[0],
X_train.shape[1], 1)), y_train, epochs=10,
batch_size=32)

loss =
model.evaluate(X_test.reshape((X_test.shape[0],
X_test.shape[1], 1)), y_test)

print("Test Loss:", loss)

from sklearn.metrics import r2_score

r2 = r2_score(y_test, y_pred)

print("R2 Score:", r2)

df = pd.DataFrame(y_test)
df['result']=y_pred
df

def classify_aqi(aqi_value):
    if aqi_value <= 0.2:
        return "Good"
    elif aqi_value <= 0.4:
        return "Satisfactory"
    elif aqi_value <= 0.6:
        return "Moderate"
    elif aqi_value <= 0.8:

```

```

        return "Poor"
    else:
        return "Very Poor"

classified_aqi = [classify_aqi(aqi) for aqi in
y_pred]

print(classified_aqi)

print(new_data)

scaler = MinMaxScaler()
new_data_scaled = scaler.fit_transform(new_data)
print(new_data_scaled)

```

GRU

```

[1] import pandas as pd

import numpy as np

from sklearn.preprocessing import MinMaxScaler

from sklearn.model_selection import train_test_split

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import LSTM, Dense,
    Bidirectional

[3] from google.colab import drive
    drive.mount('/content/drive')

Mounted at /content/drive

```

```

from google.colab import files

uploaded = files.upload()

data = pd.read_csv('AQI and Lat Long of
Countries.csv')

data = data.dropna()

print(data.columns)

scaler = MinMaxScaler()
scaled_data = scaler.fit_transform(data[['AQI
Value', 'CO AQI Value', 'Ozone AQI Value',
      'NO2 AQI Value', 'PM2.5 AQI Value']])

X = scaled_data[:, :-1]
y = scaled_data[:, -1]

X_train, X_test, y_train, y_test =
train_test_split(X, y, test_size=0.2,
random_state=42)

model = Sequential()
model.add(GRU(64, input_shape=(X_train.shape[1],
X_train.shape[2])))
model.add(Dense(1, activation='linear'))

model.compile(loss='mean_squared_error',
optimizer='adam')

model.fit(X_train.reshape((X_train.shape[0],
X_train.shape[1], 1)), y_train, epochs=10,
batch_size=32)

loss =
model.evaluate(X_test.reshape((X_test.shape[0],
X_test.shape[1], 1)), y_test)

print("Test Loss:", loss)

from sklearn.metrics import r2_score

r2 = r2_score(y_test, y_pred)

```

```

print("R2 Score:", r2)

df = pd.DataFrame(y_test)
df['result']=y_pred
df

def classify_aqi(aqi_value):
    if aqi_value <= 0.2:
        return "Good"
    elif aqi_value <= 0.4:
        return "Satisfactory"
    elif aqi_value <= 0.6:
        return "Moderate"
    elif aqi_value <= 0.8:
        return "Poor"
    else:
        return "Very Poor"

classified_aqi = [classify_aqi(aqi) for aqi in
y_pred]

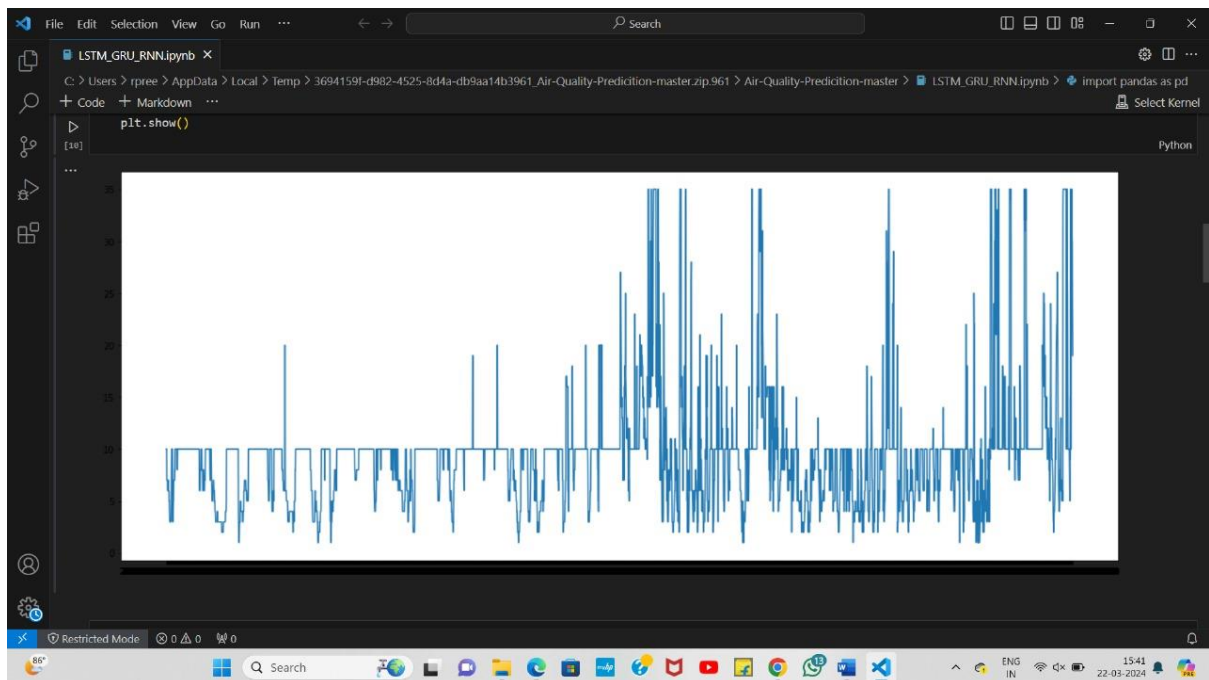
print(classified_aqi)

print(new_data)

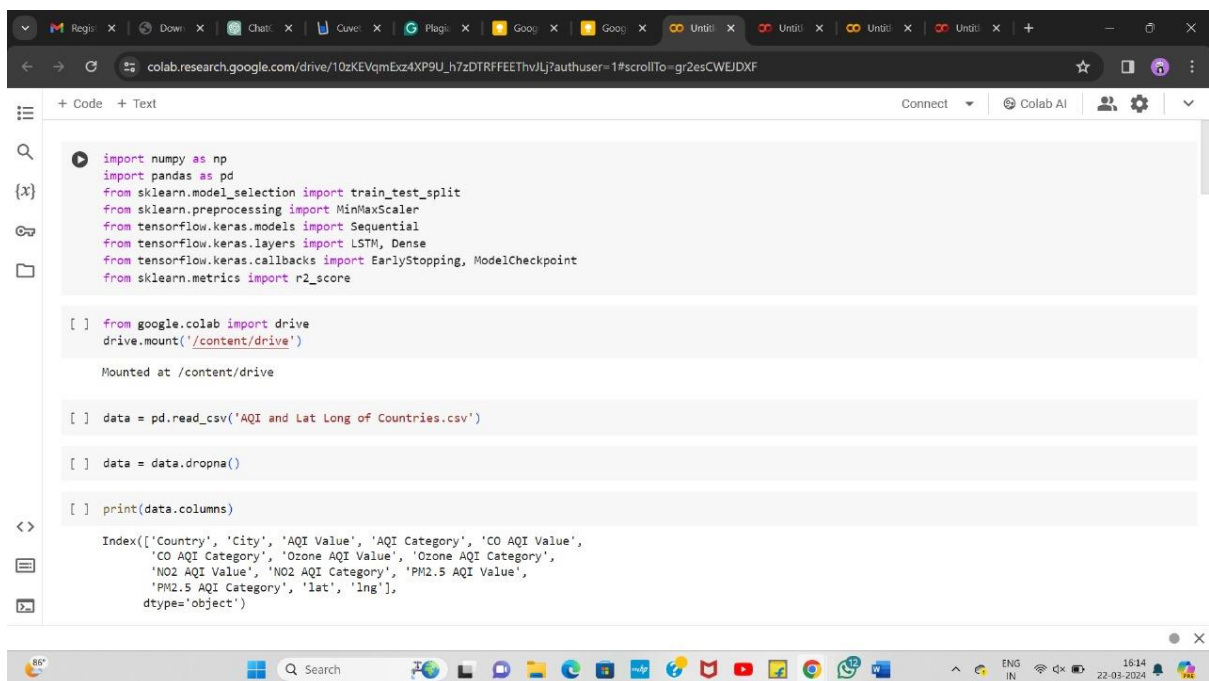
scaler = MinMaxScaler()
new_data_scaled = scaler.fit_transform(new_data)
print(new_data_scaled)

```


8.2 SCREENSHOTS



1.VISIBILITY



```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import MinMaxScaler
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import LSTM, Dense
from tensorflow.keras.callbacks import EarlyStopping, ModelCheckpoint
from sklearn.metrics import r2_score

[ ] from google.colab import drive
drive.mount('/content/drive')
Mounted at /content/drive

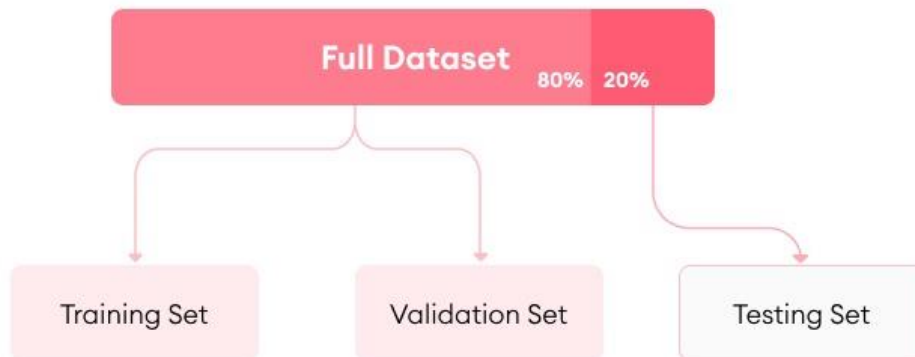
[ ] data = pd.read_csv('AQI and Lat Long of Countries.csv')

[ ] data = data.dropna()

[ ] print(data.columns)

Index(['Country', 'City', 'AQI Value', 'AQI Category', 'CO AQI Value',
      'CO AQI Category', 'Ozone AQI Value', 'Ozone AQI Category',
      'NO2 AQI Value', 'NO2 AQI Category', 'PM2.5 AQI Value',
      'PM2.5 AQI Category', 'lat', 'lng'],
      dtype='object')
```

2.DATA PROCESSING



3.DATA SPLITTING

```
[ ] X_train = np.reshape(X_train, (X_train.shape[0], 1, X_train.shape[1]))  
    X_test = np.reshape(X_test, (X_test.shape[0], 1, X_test.shape[1]))
```

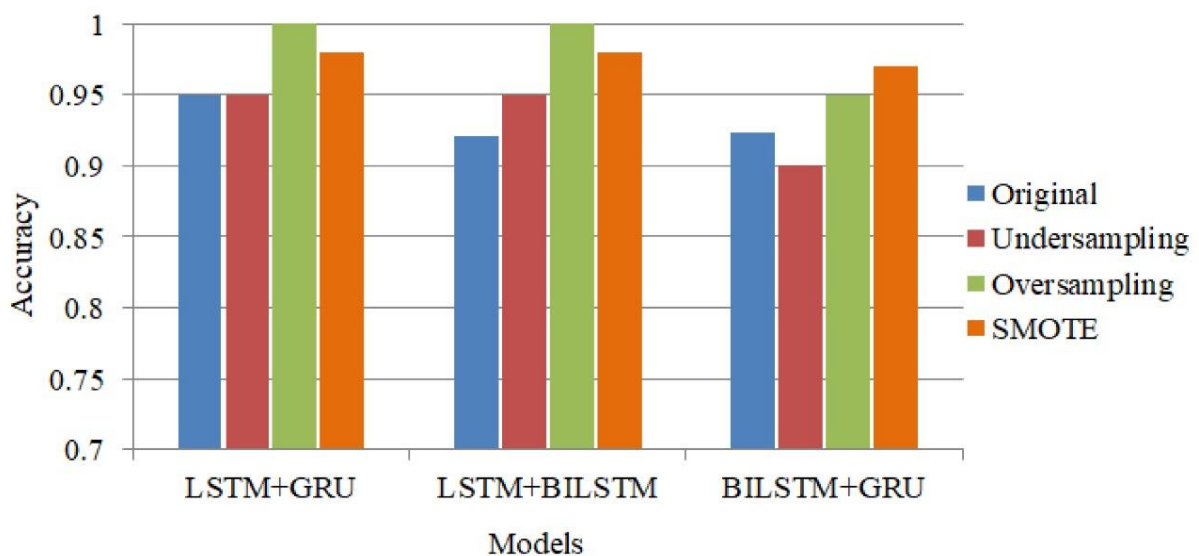
```
▶ model = Sequential()  
  model.add(LSTM(50, activation='relu', input_shape=(1, X_train.shape[2])))  
  model.add(Dense(1))  
  model.compile(optimizer='adam', loss='mse')
```

4.MODEL TRAINING

```
r2 = r2_score(y_test, y_pred)
print("R2 Score:", r2)
```

R2 Score: 0.9836171406894519

5.MODEL EVALUATION



5.GRAPH

8.3 PLAGIARISM REPORT

Prediction of Air Quality Using BI-LSTM and GRU

ORIGINALITY REPORT

10 %	6 %	5 %	3 %
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Asia Pacific University College of Technology and Innovation (UCTI) Student Paper	3 %
2	Yuting Yang, Gang Mei, Stefano Izzo. "Revealing Influence of Meteorological Conditions on Air Quality Prediction Using Explainable Deep Learning", IEEE Access, 2022 Publication	2 %
3	link.springer.com Internet Source	<1 %
4	ijritcc.org Internet Source	<1 %
5	www.researchgate.net Internet Source	<1 %
6	www.mdpi.com Internet Source	<1 %
7	ijsret.com Internet Source	<1 %

CHAPTER 9

REFERENCES

REFERENCES

- [1] P. Rafaj, G. Kieseewetter, T. Gül, W. Schöpp, J. Cofala, Z. Klimont, and P. Purohit, “Outlook for clean air in the context of sustainable development goals,” *Global Environ. Change*, vol. 53, pp. 1–11, Nov. 2018.
- [2] P. J. Landrigan, R. Fuller, N. J. R. Acosta, O. Adeyi, R. Arnold, A. B. Baldé, and R. Bertollini, “The Lancet commission on pollution and health,” *Lancet*, vol. 391, no. 10119, pp. 462–512, 2018.
- [3] R. A. Rohde and R. A. Müller, “Air pollution in China: Mapping of concentrations and sources,” *PLoS ONE*, vol. 10, no. 8, Aug. 2015, Art. no. e0135749.
- [4] State of Global Air 2019, Health Effects Institute, Boston, MA, USA, 2019.
- [5] D. A. Vallero, *Fundamentals of Air Pollution*. New York, NY, USA: Academic, 2014.
- [6] J. Fan, Q. Li, J. Hou, X. Feng, H. Karimian, and S. Lin, “A spatiotemporal prediction framework for air pollution based on deep RNN,” *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, p. 15, Jan. 2017.
- [7] D. Seng, Q. Zhang, X. Zhang, G. Chen, and X. Chen, “Spatiotemporal prediction of air quality based on LSTM neural network,” *Alexandria Eng. J.*, vol. 60, no. 2, pp. 2021–2032, Apr. 2021.
- [8] D. Iskandaryan, F. Ramos, and S. Trilles, “Comparison of nitrogen dioxide predictions during a pandemic and non-pandemic scenario in the city of Madrid using a convolutional LSTM network,” *Int. J. Comput. Intell. Appl.*, vol. 21, no. 2, Jun. 2022, Art. no. 2250014.

- [9] D. Iskandaryan, F. Ramos, and S. Trilles, “Bidirectional convolutional LSTM for the prediction of nitrogen dioxide in the city of Madrid,” *PLoS ONE*, vol. 17, no. 6, Jun. 2022, Art. no. e0269295.
- [10] S. Abirami and P. Chitra, “Regional air quality forecasting using spatiotemporal deep learning,” *J. Cleaner Prod.*, vol. 283, Feb. 2021, Art. no. 125341.
- [11] Y. Bai, B. Zeng, C. Li, and J. Zhang, “An ensemble long short-term memory neural network for hourly PM_{2.5} concentration forecasting,” *Chemosphere*, vol. 222, pp. 286–294, May 2019.
- [12] X. Ouyang, Y. Yang, Y. Zhang, and W. Zhou, “Spatial–temporal dynamic graph convolution neural network for air quality prediction,” in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2021, pp. 1–8.
- [13] L. Ge, K. Wu, Y. Zeng, F. Chang, Y. Wang, and S. Li, “Multi-scale spatiotemporal graph convolution network for air quality prediction,” *Int. J. Speech Technol.*, vol. 51, no. 6, pp. 3491–3505, Jun. 2021.
- [14] C. Wang, Y. Zhu, T. Zang, H. Liu, and J. Yu, “Modeling inter-station relationships with attentive temporal graph convolutional network for air quality prediction,” in *Proc. 14th ACM Int. Conf. Web Search Data Mining*, Mar. 2021, pp. 616–634.
- [15] L. Chen, J. Xu, B. Wu, Y. Qian, Z. Du, Y. Li, and Y. Zhang, “Group-aware graph neural network for nationwide city air quality forecasting,” 2021, arXiv:2108.12238.
- [16] Y. Liu, J. Ma, P. Dhillon, and Q. Mei, “A new benchmark of graph learning for PM_{2.5} forecasting under distribution shift,” in *Proc. Workshop Graph Learn. Benchmarks Web Conf. (GLB)*. New York, NY, USA: ACM, 2021, p. 6.