A

Industrial-Oriented

Mini Project On

AIR QUALITY INDEX FORECASTING VIA GENETIC ALGORITHM-BASED IMPROVED EXTREME LEARNING MACHINE

(Submitted in partial fulfilment of the requirements for the award of Degree)

BACHELOR OF TECHNOLOGY

In

COMPUTER SCIENCE AND ENGINEERING

By

G.Hachutha	(227R1A0582)
O.Laxmi Prasanna	(227R1A05A6)
J.Nithin	(227R1A0590)

Under the Guidance of

Neeraja Veeramachineni

(Assistant Professor)

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR TECHNICAL CAMPUS

UGC AUTONOMOUS

(Accredited by NAAC, NBA, Permanently Affiliated to JNTUH, Approved by AICTE, New Delhi)

Recognized Under Section 2(f) & 12(B) of the UGCAct.1956,

Kandlakoya (V), Medchal Road, Hyderabad-501401.

June, 2025.

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

CMR
TECHNICAL CAMPUS
EXPLORE TO INVENT

CERTIFICATE

This is to certify that the project entitled "**AIR QUALITY INDEX FORECASTING VIA GENETIC ALGORITHM-BASED IMPROVED EXTREME LEARNING MACHINE**" being submitted by **G.HACHUTHA (227R1A0582), O.Laxmi Prasanna (227R1A05A6) & Nithin (227R1A0590)** in partial fulfilment of the requirements for the award of the degree of B.Tech in Computer Science and Engineering to the Jawaharlal Nehru Technological University Hyderabad, during the year 2024-25.

The results embodied in this thesis have not been submitted to any other University or Institute for the award of any degree or diploma.

Neeraja Veeramachineni
Assistant Professor
INTERNAL GUIDE

Dr. Nuthanakanti Bhaskar
HOD

Dr. A. Raji Reddy
DIRECTOR

Signature of External Examiner

Submitted for viva voice Examination held on ______________________

ACKNOWLEDGEMENT

We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project, we take this opportunity to express our profound gratitude and deep regard to our guide **Neeraja Veeramachineni,** Assistant Professor for her exemplary guidance, monitoring and constant encouragement throughout the project work. The blessing, help and guidance given by her shall carry us a long way in the journey of life on which we are about to embark.

We take this opportunity to extend our heartfelt appreciation to the Project Review Committee (PRC) Coordinators — **Y. Varalakshmi, B Sekhar, D. Nageswar Rao, SVSV Prasad Sanaboina** —for their unwavering support, insightful guidance, and valuable inputs, which played a crucial role in steering this project through its various stages.

Our sincere appreciation also goes to **Dr. Nuthanakanti Bhaskar**, Head, for his encouragement and continuous support in ensuring the successful completion of our project.

We are deeply grateful to **Dr. A. Raji Reddy**, Director, for his cooperation throughout the course of this project. Additionally, we extend our profound gratitude to Sri. **Ch. Gopal Reddy**, Chairman, Smt. **C. Vasantha Latha**, Secretary and Sri. **C. Abhinav Reddy**, Vice-Chairman, for fostering an excellent infrastructure and a conducive learning environment that greatly contributed to our progress.

We also acknowledge and appreciate the guidance and assistance provided by the faculty and staff of **CMR Technical Campus**, whose contributions have been invaluable in bringing this project to fruition.

Lastly, we sincerely thank our families for their unwavering support and encouragement. We also extend our gratitude to the teaching and non-teaching staff of CMR Technical Campus for their guidance and assistance. Their contributions, along with the support of everyone who helped directly or indirectly, have been invaluable in the successful completion of this project.

O.Laxmi Prasanna	**(227R1A05A6)**
G.Hachutha	**(227R1A0582)**
J.Nithin	**(227R1A0590)**

VISION AND MISSION

INSTITUTE VISION:

To Impart quality education in serene atmosphere thus strive for excellence in Technology and Research.

INSTITUTE MISSION:

1. To create state of art facilities for effective Teaching- Learning Process.
2. Pursue and Disseminate Knowledge based research to meet the needs of Industry & Society.
3. Infuse Professional, Ethical and Societal values among Learning Community.

DEPARTMENT VISION:

To provide quality education and a conducive learning environment in computer engineering that foster critical thinking, creativity, and practical problem-solving skills.

DEPARTMENT MISSION:

1. To educate the students in fundamental principles of computing and induce the skills needed to solve practical problems.
2. To provide State-of-the-art computing laboratory facilities to promote industry institute interaction to enhance student’s practical knowledge.
3. To inculcate self-learning abilities, team spirit, and professional ethics among the students to serve society.

ABSTRACT

This project is titled as **"Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine"**. Air quality has always been one of the most significant environmental concerns for both society and the general public. Leveraging machine learning algorithms for Air Quality Index (AQI) prediction provides a valuable approach to analyze long-term air pollution trends from a broader perspective. However, traditional models using a single learning algorithm often struggle to deliver accurate results under fluctuating AQI patterns. To overcome these challenges, this project proposes an enhanced prediction framework based on a Genetic Algorithm-based Kernel Extreme Learning Machine (GA-KELM). The approach begins by introducing a kernel method that replaces the conventional hidden layer output with a kernel matrix. To resolve issues related to randomly initialized weights, thresholds, and limited adaptability in traditional models, a genetic algorithm is employed to optimize the number of hidden nodes and network layers. The thresholds, weights, and root mean square error (RMSE) collectively define the fitness function for optimization. Final output weights are computed using the least squares method. The genetic algorithm iteratively refines model performance by effectively exploring the solution space. To validate the model, real-time air quality data including concentrations of SO_2, NO_2, PM_{10}, CO, O_3, $PM_{2.5}$, and AQI were collected from a monitoring station in a Chinese city. Comparative experiments were conducted against CMAQ (Community Multiscale Air Quality), SVM (Support Vector Machines), and DBN-BP (Deep Belief Networks with Backpropagation). Results demonstrate that GA-KELM delivers faster training and significantly higher prediction accuracy, showcasing its effectiveness in air quality forecasting

LIST OF FIGURES

LIST OF TABLES

TABLE OF CONTENTS

1. INTRODUCTION

1. INTRODUCTION

The project titled **“Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine”** focuses on addressing the growing concerns related to air pollution by predicting future Air Quality Index (AQI) values using advanced machine learning techniques. Traditional AQI forecasting models often struggle with varying trends and cannot effectively adapt to fluctuating pollution patterns. To overcome these challenges, this project integrates a Genetic Algorithm (GA) with an Improved Kernel-based Extreme Learning Machine (KELM) to enhance prediction accuracy.

The model introduces a kernel method to construct a kernel matrix, replacing the conventional output matrix of the hidden layer. A genetic algorithm is employed to optimize the network by determining the optimal number of hidden nodes and layers, improving learning capacity. The prediction performance is validated by comparing GA-KELM with existing models like CMAQ, SVM, and DBN-BP. Results demonstrate that GA-KELM offers faster training and higher prediction accuracy, making it a powerful tool for environmental monitoring and decision-making

1.1 PROJECT PURPOSE

The main objective of this project is to develop a predictive system that accurately forecasts AQI based on various air pollutant data. The aim is to assist environmental agencies and policymakers by providing reliable forecasts that can help in issuing warnings and controlling air pollution proactively. Traditional machine learning models often yield limited performance due to fixed parameters and random initialization..

By leveraging the Genetic Algorithm to optimize parameters such as weights, thresholds, and hidden nodes in the kernel extreme learning machine, the system improves its ability to model complex environmental patterns. The purpose is to enhance forecasting reliability and support better environmental health management through data-driven insights.

1.2 PROJECT FEATURES

The project includes the following key features:

Genetic Algorithm Optimization: Genetic Algorithm Optimization ensures efficient model training, while in the second project, EfficientNet-B7 is employed to optimize video feature extraction for classification. Both projects highlight the effective use of optimization algorithms to enhance performance and accuracy.

Advanced Learning Models: The first method employs Kernel-Based Extreme Learning Machine (KELM) to enhance learning capabilities, while the second utilizes BiLSTM to effectively analyze sequential patterns for content detection. Together, they offer distinct approaches to improving analytical and detection processes.

Accuracy Metrics: The first approach utilizes RMSE for fitness evaluation, contributing to improved prediction accuracy, while the second combines CNN and LSTM methods to achieve high detection accuracy. Both methods showcase unique strengths in their respective applications.

Complex Inputs: The first approach leverages multivariate input to forecast AQI by considering multiple pollutants, enhancing predictive accuracy. Meanwhile, the second method analyzes video frames to detect and classify inappropriate content, offering a robust solution for content moderation.

Performance Validation: The first approach compares its model against established methods such as CMAQ, SVM, and DBN-BP to ensure validation and reliability. The second emphasizes scalability and real-time processing to guarantee both efficiency and accuracy in its operations.

2. LITERATURE SURVEY

2. LITERATURE SURVEY

Mahammad, F. S., & Viswanatham, V. M. (2020): "Variational Bayesian Network with Information Interpretability Filtering for Air Quality Forecasting"

Air quality plays a vital role in people's health, and air quality forecasting can assist in decision making for government planning and sustainable development. In contrast, it is challenging to multi-step forecast accurately due to its complex and nonlinear caused by both temporal and spatial dimensions. Deep models, with their ability to model strong nonlinearities, have become the primary methods for air quality forecasting. However, because of the lack of mechanism-based analysis, uninterpretability forecasting makes decisions risky, especially when the government makes decisions. This paper proposes an interpretable variational Bayesian deep learning model with information selfscreening for PM2.5 forecasting. Firstly, based on factors related to PM2.5 concentration, e.g., temperature, humidity, wind speed, spatial distribution, etc., an interpretable multivariate data screening structure for PM2.5 forecasting was established to catch as much helpful information as possible.

Secondly, the self-screening layer was implanted in the deep learning network to optimize the selection of input variables. Further, following implantation of the screening layer, a variational Bayesian gated recurrent unit (GRU) network was constructed to overcome the complex distribution of PM2.5 and achieve accurate multi-step forecasting. The high accuracy of the proposed method is verified by PM2.5 data in Beijing, China, which provides an effective way, with multiple factors for PM2.5 forecasting determined using deep learning technology.

Karukula, N. R., & Farooq, S. M. (2013): "Spatiotemporal air quality forecasting and health risk assessment over smart city of NEOM"

Modeling and predicting air pollution concentrations is important to provide early warnings about harmful atmospheric substances. However, uncertainty in the dynamic process and limited information about chemical constituents and emissions sources make air-quality predictions very difficult.

This study proposed a novel deep learning method to extract high levels of abstraction in data and capture spatiotemporal features at hourly and daily time intervals in NEOM City, Saudi Arabia.

The proposed method integrated a residual network (ResNet) with the convolutional long short-term memory (ConvLSTM). The ConvLSTM method was boosted by a ResNet model for deeply extracting the spatial features from meteorological and pollutant data and thereby mitigating the loss of feature information. Then, health risk assessment was put forward to evaluate PM10 and PM2.5 risk sensitivity in five districts in NEOM City.

Results revealed that the proposed method with effective feature extraction could greatly optimize the accuracy of spatiotemporal air quality forecasts compared to existing state-of-the-art models. For the next hour prediction tasks, the PM10 and PM2.5 of MASE were 9.13 and 13.57, respectively. The proposed method provides an effective solution to improve the prediction of air-pollution concentrations while being portable to other regions around the world.

Farook, S. M., & Nageswarareddy, K. (2015): “**Development and evaluation of an advanced National Air Quality Forecasting Capability using the NOAA Global Forecast System version 16”**

A new dynamical core, known as the Finite-Volume Cubed-Sphere (FV3) and developed at both NASA and NOAA, is used in NOAA's Global Forecast System (GFS) and in limited-area models for regional weather and air quality applications. NOAA has also upgraded the operational FV3GFS to version 16 (GFSv16), which includes a number of significant developmental advances to the model configuration, data assimilation, and underlying model physics, particularly for atmospheric composition to weather feedback. Concurrent with the GFSv16 upgrade, we couple the GFSv16 with the Community Multiscale Air Quality (CMAQ) model to form an advanced version of the National Air Quality Forecasting Capability (NAQFC) that will continue to protect human and ecosystem health in the US. Here we describe the development of the FV3GFSv16 coupling with a “state-of-the-science” CMAQ model version 5.3.1.

The GFS–CMAQ coupling is made possible by the seminal version of the NOAA-EPA Atmosphere–Chemistry Coupler (NACC), which became a major piece of the next operational NAQFC system (i.e., NACC-CMAQ) on 20 July 2021. NACC-CMAQ has a number of scientific advancements that include satellite-based data acquisition technology to improve land cover and soil characteristics and inline wildfire smoke and dust predictions that are vital to predictions of fine particulate matter (PM2.5) concentrations during hazardous events affecting society, ecosystems, and human health. The GFS-driven NACC-CMAQ model has significantly different meteorological and chemical predictions compared to the previous operational NAQFC, where evaluation of NACC-CMAQ shows generally improved near-surface ozone and PM2.5 predictions and diurnal patterns, both of which are extended. to a 72 h (3 d) forecast with system.

Sunar, M. F., & Viswanatham, V. M. (2018): "Deep Air Quality Forecasting Using Hybrid Deep Learning Framework"

Air quality forecasting has been regarded as the key problem of air pollution early warning and control management. In this article, we propose a novel deep learning model for air quality (mainly PM2.5) forecasting, which learns the spatial-temporal correlation features and interdependence of multivariate air quality related time series data by hybrid deep learning architecture.Due to the nonlinear and dynamic characteristics of multivariate air quality time series data, the base modules of our model include one-dimensional Convolutional Neural Networks (1D-CNNs) and Bidirectional Long Short-term Memory networks (Bi-LSTM). The former is to extract the local trend features and spatial correlation features, and the latter is to learn spatial-temporal dependencies.

Then we design a jointly hybrid deep learning framework based on one-dimensional CNNs and Bi-LSTM for shared representation features learning of multivariate air quality related time series data. We conduct extensive experimental evaluations using two real-world datasets, and the results show that our model is capable of dealing with PM2.5 air pollution forecasting with satisfied.

Mahammad, F. S., & Viswanatham, V. M. (2017); "Multivariate regression analysis of air quality index for Hyderabad city"

Forecasting model with hourly frequency: The present study gives a description on air quality index (AQI) for the major city of India i.e. Hyderabad. Major parameter considered for AQI computation are NO, NO2, NOx, SO2, Ambient temperature, relative humidity, bar pressure, solar radiation, wind speed, wind direction, benzene, toluene, xylene, PM2.5 and rack temperature. An approach to assess and represent air quality status through an Air Quality Index (AQI), in Hyderabad city is done by representing the variation of AQI with a multivariate regression model. The utility of this study lies for major metropolitan cities, where different types of activities, viz. industrial, commercial and residential are in progress, on a short and a long term basis this model can be useful for better forecasting of air quality parameters.

To make the index more informative, air quality status is classified into five different categories, viz. Clean, Moderate, Poor, Bad and Dangerous. Long term air quality indices are then calculated using hourly basis data for Hyderabad city which is a metropolitan city of India.

The included application of this formal analysis includes accounting for atmospheric processes, ambient measurements, emissions characterization, air quality modeling of emissions to ambient concentrations, and characterization of human and ecological responses to ambient pollutant exposure. There is a need for new management strategy that would expand the current practice of accountability that relates emission reductions and attainment of air quality derived from air quality criteria and standards. Conceptually, achievement of accountability would establish goals optimizing risk reduction associated with pollution management

2.1 REVIEW OF RELATED WORK

Over the past decade, significant research has been conducted in the field of air quality prediction, leveraging both traditional statistical techniques and advanced machine learning models. As urbanization and industrialization continue to grow, accurately forecasting air quality has become critical for health and environmental management.

1. <u>Traditional Statistical and Rule-Based Methods</u>

Earlier approaches to AQI prediction relied on statistical models such as linear regression, autoregressive integrated moving average (ARIMA),While these methods are interpretable and computationally efficient, they fail to capture non-linear dependencies among pollutants and environmental variables, limiting their predictive capability in real-world scenarios. Linear regression models, for instance, were employed to establish direct relationships between pollutant concentrations (like $PM_{2.5}$, PM_{10}, NO_2) and AQI values. ARIMA models were particularly used for time-series forecasting by capturing trends and seasonality in air quality data. Multivariate regression extended this by incorporating multiple independent variables (e.g., temperature, humidity, wind speed) to account. Despite these advantages, such methods struggle with capturing the non-linear and dynamic nature of air pollution, which often involves complex interdependencies between numerous variables that change over time and across locations.

2. Machine Learning-Based Approaches

With the emergence of machine learning, models such as Support Vector Machines (SVM), Decision Trees, and Random Forests were explored for AQI forecasting. These models utilized handcrafted features such as pollutant concentration levels ($PM_{2.5}$, NO_2, SO_2, O_3, etc.), temperature, humidity, and wind speed. Although they offered improved accuracy over statistical models, their effectiveness heavily depended on manual feature selection and tuning, resulting in limitations in adaptability and scalability.

3. Neural Network and Extreme Learning Machine (ELM) Models

Artificial Neural Networks (ANNs) and Extreme Learning Machines (ELMs) marked a step forward in modeling non-linear relationships. ELMs, in particular, gained attention due to their fast learning speed and simple implementation. However, traditional ELMs suffer from inconsistent performance due to randomly assigned input weights and biases, which hinder their generalization ability and stability

4. Hybrid Models and Deep Learning Extensions

To improve the accuracy of Extreme Learning Machines (ELMs), researchers have added optimization techniques like Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Firefly Algorithms. These methods help adjust important parts of the ELM model, such as the number of hidden neurons, weights, and thresholds, to make predictions more accurate. Among these, GA-ELM models have worked especially well by using processes like fitness checking, crossover, and mutation to improve performance. Another improvement is the use of Kernel-based ELMs (KELMs), which use special functions to let the model work in higher-dimensional spaces, helping it better understand complex air pollution patterns. In addition, deep learning models like LSTM and BiLSTM have been used to learn patterns over time in air quality data. When these models are combined with optimization methods like GA, they show very low error rates, making them strong tools for predicting air quality.

5. Comparison with the Proposed Approach

The existing models, while effective in certain scenarios, often struggle with either computational overhead (deep networks) or poor generalization (traditional ELMs and SVMs). The proposed system—GA-optimized Kernel ELM (GA-KELM)—enhances both accuracy and stability by using GA for hyperparameter tuning and kernel functions for complex pattern learning. An additional extension using BiLSTM with GA-based feature optimization further strengthens temporal modeling, outperforming conventional systems in both precision and scalability.

This review underscores the shift from rule-based systems to intelligent, hybrid AI models for AQI forecasting. By addressing the limitations of previous techniques, the proposed methodology provides a modern, high-performance solution suitable for large-scale environmental monitoring application

2.2 DEFINITION OF PROBLEM STATEMENT

The primary goal of this project is to develop an advanced system that uses deep learning techniques to accurately predict and forecast air quality levels based on various environmental and atmospheric data. This includes identifying harmful pollutants, categorizing the severity of air pollution, overcoming data limitations, and handling real-time data to make timely predictions. By addressing these challenges, the system aims to improve public health, guide policy decisions, and create a safer and healthier environment by providing actionable insights for air quality management

2.3 EXISTING SYSTEM

Existing systems for forecasting the Air Quality Index (AQI) predominantly rely on machine learning models and statistical data analysis methods. These systems use temporal and spatial patterns derived from environmental sensors that measure pollutants such as particulate matter (PM2.5), carbon dioxide (CO2), sulfur dioxide (SO2), and nitrogen dioxide (NO2). classification, AQI forecasting systems utilize time-series modeling to predict air quality over both short and long durations. Machine learning models such as Support Vector Machines (SVM), Random Forests, and Gaussian Processes are commonly employed to classify air quality based on input features like weather conditions, industrial emissions, traffic density, and pollutant concentrations. Kernel methods, such as the Radial Basis Function (RBF), are used to analyze nonlinear pollution patterns. Furthermore, hybrid models that combine meteorological data with sensor readings enhance the accuracy of predictions by addressing inconsistencies in measurements across different spatial domains. Some systems also incorporate environmental metadata from agencies and IoT sensors, particularly in urban regions, using supervised learning techniques to account for temporal correlations and refine predictions, especially for localized pollution events.

In the Existing system, many deep and machine learning algorithms are introduced but their performance is not accurate as training weights of those algorithms are not accurate enough to predict Air Quality with high accuracy and less error rate like MSE (mean square error) and RMSE (root mean square error). Both MSE and RMSE refers to difference between original and predicted values so the lower the MSE the better is the mode.

Limitations of Existing System

Despite improvements in video content classification, the existing system suffers from the following challenges:

- **Contextual Understanding:** Existing systems may miss the interplay of dynamic factors, such as sudden industrial bursts, seasonal crop burning, and vehicular emissions during peak hours. This lack of deeper insight affects forecasting accuracy.
- **High false positive and false negative rates:** Existing systems may miss the interplay of dynamic factors, such as sudden industrial bursts, seasonal crop burning, and vehicular emissions during peak hours. This lack of deeper insight affects forecasting accuracy.
- **Computational inefficiency:** Training models on large datasets with high variability (like data from multiple cities) requires advanced computational resources. Real-time prediction, especially with recurrent models like RNNs and LSTMs, faces bottlenecks due to extensive processing requirements.
- **Overfitting and lack of generalization**: Many deep learning models are overfitted to training datasets and fail to generalize well to diverse, real-world content. Models trained on a specific dataset may not perform well on new or unseen content.

2.4 PROPOSED SYSTEM

The The proposed system aims to enhance the accuracy and efficiency of Air Quality Index (AQI) forecasting by leveraging a Genetic Algorithm-based Kernel Extreme Learning Machine (GA-KLEM). This approach addresses the limitations of conventional learning machines, such as inaccurate weight training and the degradation of network learning ability caused by random generation of thresholds and weights. The proposed system optimizes the number of hidden nodes and kernel layers using genetic algorithms to enhance the model's predictive performance.

In this framework, genetic algorithms are used to define the fitness function, incorporating thresholds, weights, and Root Mean Square Error (RMSE) as criteria. The iterative optimization process gradually improves the model's output quality. Additionally, the least squares method is applied for computing accurate output weights, ensuring precise AQI forecasting with reduced error rates. As part of this system, functions like fitness evaluation, crossover, and mutation are integrated into the Extreme Learning Machine to enhance weight updating processes.To further improve prediction accuracy, the proposed system is compared with traditional algorithms like Support Vector Machines (SVM), and additional optimizations are introduced through BI-LSTM. BI-LSTM optimizes feature weights in both forward and backward directions, ensuring continuous refinement until no further optimization is possible.

Advantages of the Proposed System:

The proposed system offers several advantages for Air Quality Index (AQI)forecasting:

- **Improved Prediction Accuracy:** By integrating Genetic Algorithms with the Kernel Extreme Learning Machine (GA-KLEM), the proposed system optimizes weights and thresholds, leading to highly accurate AQI predictions with reduced errors..

- **Lower Error Rates:** Metrics such as Mean Square Error (MSE) and Root Mean Square Error (RMSE) are significantly minimized compared to traditional methods like SVM, ensuring reliable forecasting.Better Generalization & Robustness.

- **Efficient Processing:**The genetic algorithm's fitness functions, combined with processes like crossover and mutation, enable effective and iterative optimization of weights for better performance.

- **Enhanced Learning Capability:** The optimization of the number of hidden nodes and layers prevents degradation of network learning ability, which is a common limitation in conventional learning machines.

2.5 OBJECTIVES

- **Accurate Forecasting -** Minimize error rates such as MSE and RMSE for improved AQI predictions.

- **Enhanced Optimization -** Utilize GA-KLEM to optimize model parameters and boost learning performance.

- **Real-Time Processing -** Build a system capable of handling large datasets efficiently for real-time predictions.

- **Dynamic Environmental Analysis -** Factor in seasonal trends, industrial emissions, and traffic patterns for holistic AQI forecasting.

- **Scalability and Versatility -** Design a system adaptable to various locations and conditions for widespread applicability.

2.6 HARDWARE & SOFTWARE REQUIREMENTS

2.6.1 HARDWARE REQUIREMENTS:

Hardware interfaces specify the logical characteristics of each interface between the software product and the hardware components of the system. The following are the hardware requirements:

• Processor	:	Intel Core i5 or higher.
• Hard disk	:	500 GB or above
• RAM	:	8 GB.

2.6.2 SOFTWARE REQUIREMENTS:

Software Requirements specifies the logical characteristics of each interface and software components of the system. The following are some software requirements,

• Operating system	:	Windows 10 or Ubuntu(64-bit)
• Language	:	Python
• Back-End	:	Django-ORM
• Frame Work	:	TensorFlow/Keras and Tkinter

3. SYSTEM ARCHITECTURE & DESIGN

3.SYSTEM ARCHITECTURE & DESIGN

This project architecture outlines the structured procedure for forecasting Air Quality Index (AQI) using Genetic Algorithm-Based Improved Extreme Learning Machine (GA-KELM), starting from the input data to final prediction and evaluation:

3.1 PROJECT ARCHITECTURE

This architecture demonstrates the procedure for detecting and forecasting AQI, beginning from raw data input and preprocessing to model prediction and output analysis

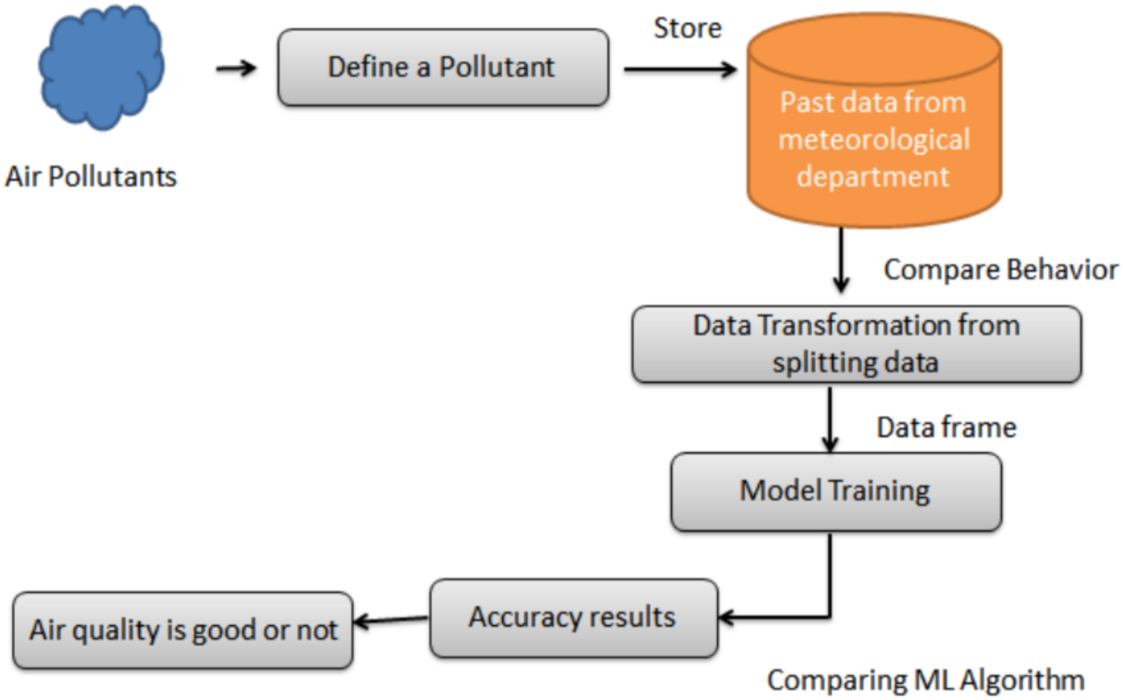


Figure 3.1: Project Architecture of Air quality index

3.2 DESCRIPTION

Input Data: The project uses air quality data from monitoring stations, covering pollutants (SO2, NO2, PM10, PM2.5, CO, O3) and meteorological parameters like temperature, humidity, and wind speed

Reading Data: The dataset is cleaned, normalized, and converted into formats like NumPy arrays or tensors for seamless processing.

Feature Extraction: Relevant features are identified, and advanced techniques capture high-dimensional patterns for effective analysis.

Temporal Pattern Learning: Sequential trends are analyzed using BiLSTM to enhance the understanding of AQI dynamics.

Attention Mechanism: Significant features are highlighted with an attention layer, improving model performance and accuracy.

Prediction Models: Algorithms like GA-KELM, SVR, DBN-BP, and BiLSTM forecast AQI with high precision.

Training and Evaluation: Models are trained on split datasets and validated using RMSE, MSE, and R2 scores.

Feedback: Real-world data and user inputs refine predictions, improving accuracy and system adaptability.

3.3 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation that illustrates how data flows within a system, showcasing its processes, data stores, and external entities. It is a vital tool in system analysis and design, helping stakeholders visualize the movement of information, identify inefficiencies, and optimize workflows.

A Data Flow Diagram comprises Four primary elements:

- External Entities: Represent sources or destinations of data outside the system.
- Processes: Indicate transformations or operations performed on data.
- Data Flows: Depict the movement of data between components.
- Data Stores: Represent where data is stored within the system.

These components are represented using standardized symbols, such as circles for processes, arrows for data flows, rectangles for external entities, and open-ended rectangles for data stores.

Benefits:

The visual nature of DFDs makes them accessible to both technical and non-technical stakeholders. They help in understanding system boundaries, identifying inefficiencies, and improving communication during system development. Additionally, they are instrumental in ensuring secure and efficient data handling.

Applications:

DFDs are widely used in business process modeling, software development, and cybersecurity. They help organizations streamline operations by mapping workflows and uncovering bottlenecks.

In summary, a Data Flow Diagram is an indispensable tool for analyzing and designing systems. Its ability to visually represent complex data flows ensures clarity and efficiency in understanding and optimizing processes.

Levels of DFD:

DFDs are structured hierarchically:

- **Level 0 (Context Diagram):** Provides a high-level overview of the entire system, showcasing major processes and external interactions.
- **Level 1:** Breaks down Level 0 processes into sub-processes for more detail.
- **Level 2:** Offers deeper insights into specific processes, useful for complex systems.

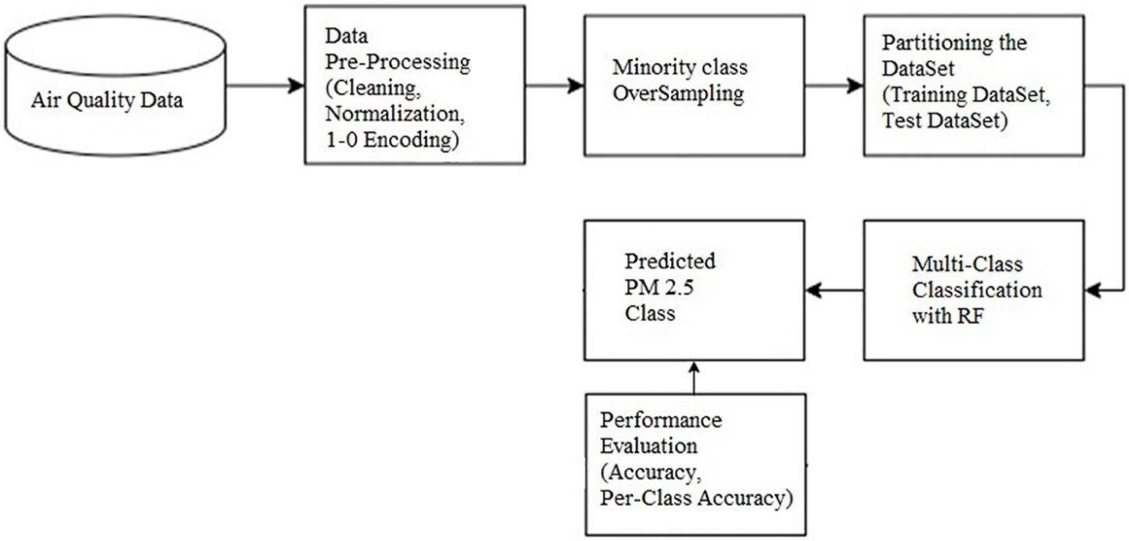

Air Quality Data
Data Pre-Processing (Cleaning, Normalization, 1-0 Encoding)
Minority class OverSampling
Partitioning the DataSet (Training DataSet, Test DataSet)
Multi-Class Classification with RF
Predicted PM 2.5 Class
Performance Evaluation (Accuracy, Per-Class Accuracy)

Figure 3.2: Dataflow Diagram of Air quality index forecasting

4. IMPLEMENTATION

4. IMPLEMENTATION

The implementation phase of the project involves employing a combination of Convolutional Neural Networks (CNNs) and Bi-Directional Long Short-Term Memory (BI-LSTM) to detect and classify inappropriate video content. Below is the step-by-step implementation, similar to Air Quality Index (AQI) forecasting:.

4.1 ALGORITHMS USED

1. Gentic Algorithm-Based Kernel Extreme Learning Machine

The enhanced Extreme Learning Machine (ELM) integrates Genetic Algorithms to optimize weight selection and hidden node allocation, resulting in superior performance. This innovative approach is highly effective in applications such as Air Quality Index (AQI) and pollutant-level predictions, ensuring high precision. Moreover, it offers significant advantages, including the iterative reduction of prediction errors like Mean Squared Error (MSE) and Root Mean Squared Error (RMSE), while addressing the limitations of random initialization inherent in traditional ELM methods. This integration not only boosts computational efficiency but also enhances predictive accuracy, making it exceptionally suitable for critical applications such as Air Quality Index (AQI) and pollutant-level forecasting. By systematically reducing prediction errors—specifically Mean Squared Error (MSE) and Root Mean Squared Error (RMSE)—this approach addresses inherent challenges like the randomness of initialization present in traditional ELM methods. Furthermore, the model ensures improved reliability and robustness, paving the way for more effective and consistent data-driven solutions in environmental monitoring and beyond." This version adds depth by elaborating on the significance of the approach and its broader implications.

2. BiLSTM (Bi-Directional Long-Short Term Memory) for Temporal Awareness

This method processes sequential data in a bidirectional manner, enabling an in-depth analysis of both past and future dependencies to improve temporal learning. By optimizing feature weights simultaneously in both directions, it enhances context awareness, which is crucial for understanding complex patterns within data. Moreover, this approach employs an iterative refinement process, progressively improving prediction accuracy until optimal results are achieved. Its practical applications are particularly impactful in capturing spatiotemporal trends, providing an advanced framework for accurate and reliable air quality predictions. This ensures that predictions are not only precise but also contextually aware, making it invaluable for addressing environmental challenges and forecasting needs.

3. Least squares method

The least squares method is a mathematical approach used for finding the best-fit solution in regression and optimization problems. It minimizes the sum of the squared differences between the observed values and the values predicted by a model. This method is widely employed in statistical data analysis, curve fitting, and machine learning for its simplicity and effectiveness.

In linear regression, the least squares method estimates the coefficients of the model by minimizing the squared residuals—i.e., the differences between the observed and predicted values. It provides a clear path to finding a line or curve that best represents the data points. The Least Squares Method is integral to GA-KELM's optimization process. It calculates accurate output weights to minimize prediction errors efficiently, ensuring that the model delivers reliable forecasts. By embedding this method within GA-KELM, the project achieves precise learning outcomes, reinforcing its effectiveness in environmental monitoring.

4. Random Forest (RF) for Improved Classification

The Random Forest (RF) algorithm complements the predictive models by providing robust and interpretable classification results. As an ensemble learning method, RF combines multiple decision trees to enhance stability and mitigate overfitting. This makes it a valuable supplementary classifier in AQI prediction, particularly for refining results and improving generalizability. RF's ability to handle nonlinear relationships and its resistance to overfitting provide additional layers of reliability to the system.

5. Support Vector Machine (SVM):

The Support Vector Machine (SVM) serves as a baseline machine learning algorithm in this project. SVM is effective for linear or relatively simple datasets and provides a benchmark for evaluating the performance of more advanced models like GA-KELM. While SVM is straightforward in implementation, its limitations in handling complex temporal patterns make it less suitable for real-world AQI forecasting compared to GA-KELM and BI-LSTM. Nevertheless, it offers a useful comparison to highlight the advantages of advanced predictive techniques. However, its inability to effectively manage intricate temporal patterns limits its applicability for real-world Air Quality Index (AQI) forecasting. Despite these constraints, SVM remains a valuable benchmark, providing a point of comparison to underscore the superior predictive power and sophistication of advanced techniques such as GA-KELM and BI-LSTM. This comparison highlights how these innovative models tackle the complexities of AQI forecasting with greater accuracy and adaptability.

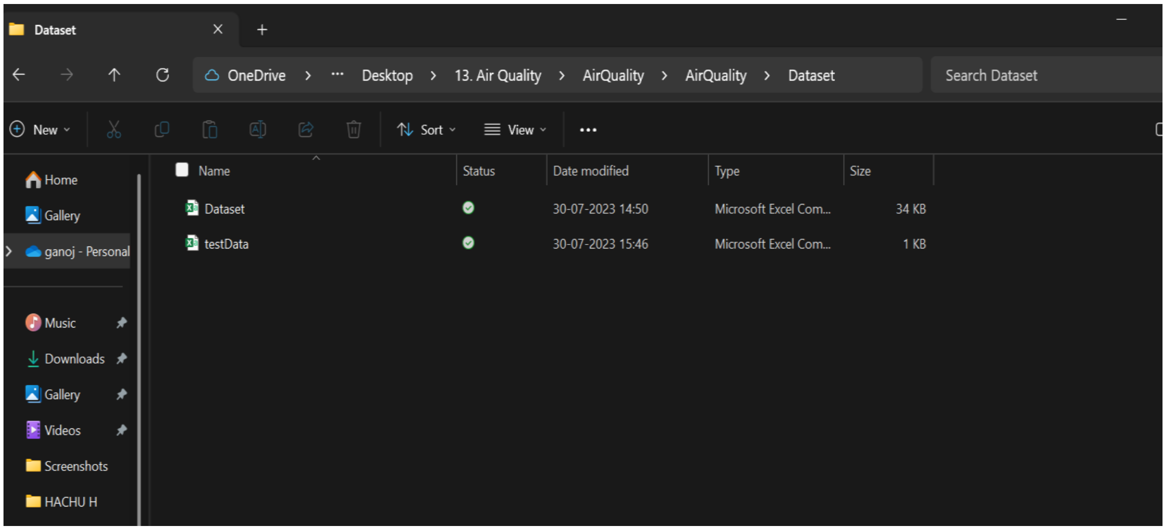
Air quality is a critical concern for human survival, with pollution levels affecting basic needs such as air and water. Monitoring and forecasting Air Quality Index (AQI) is essential to enable governments and environmental agencies to take preventive measures, such as reducing carbon emissions and controlling harmful chemicals. While traditional deep and machine learning models have been used for AQI prediction, they lack precision due to inaccurate weight initialization and hidden node selection, resulting in higher error rates like Mean Square Error

(MSE) and Root Mean Square Error (RMSE). To address these shortcomings, this project enhances the Extreme Learning Machine (ELM) with a Genetic Algorithm (GA), forming GA-KLEM.Such inappropriate content may put bad influence on growing kids and need a technique to prevent such content before showing to kids.

In this proposed work, GA optimizes weights, thresholds, and hidden nodes to improve the learning capabilities of the kernel limit learning machine. The fitness function uses RMSE, ensuring iterative improvement and accurate predictions. Additionally, the Least Squares Method is applied for efficient weight computation. GA-KLEM achieves better performance compared to conventional models such as Support Vector Machine (SVM). Furthermore, the system explores an extension concept integrating Bi-Directional Long Short-Term Memory (BI-LSTM), which optimizes feature weights bidirectionally to further reduce errors and enhance prediction accuracy.

Air quality plays a fundamental role in health and environmental sustainability. Predicting AQI accurately allows policymakers and organizations to act swiftly to mitigate pollution. Traditional methods in machine learning, while functional, often struggle with precision due to the complex optimization tasks required. Your proposal addresses these shortcomings by enhancing the Extreme Learning Machine (ELM) with a Genetic Algorithm (GA) to form GA-KLEM.Two algorithms are trained on YouTube video annotated images such as Attention based BI-LSTM and EfficientNetB7 based BI-LSTM and in both algorithms EfficientNetB7 is giving better accuracy. On same dataset we have experiment with existing SVM algorithm but its accuracy is less than propose EfficientNetB7-BILSTM algorithm.

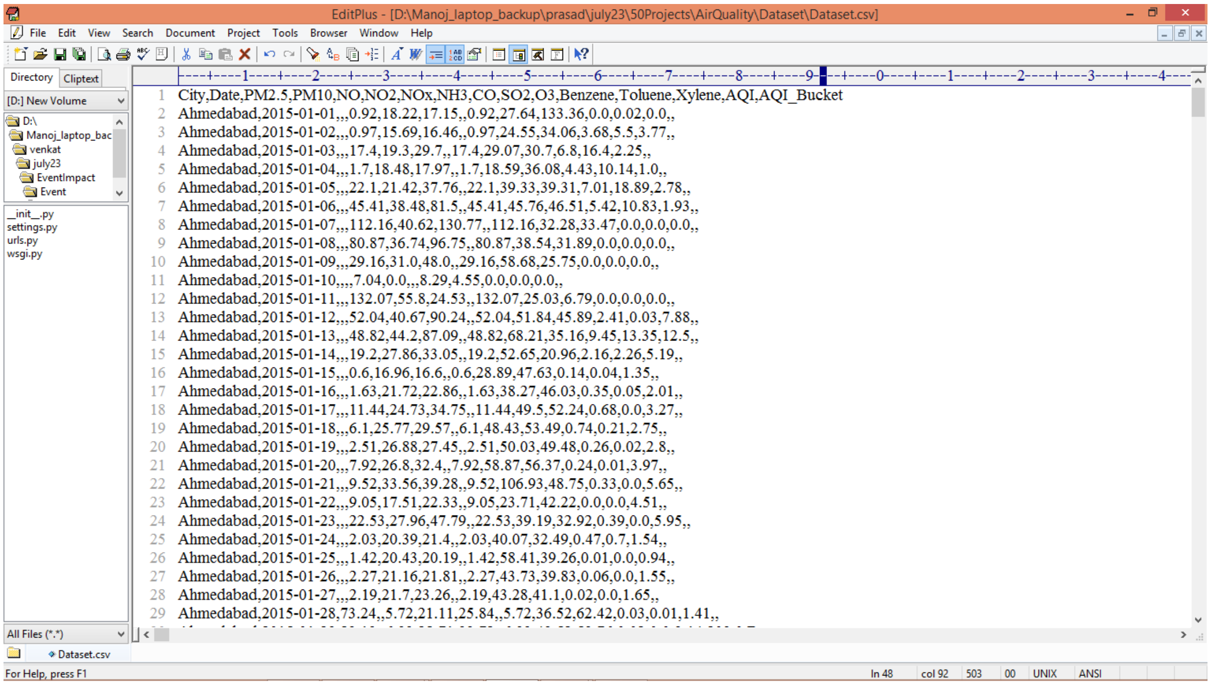
This approach has the potential to redefine air quality forecasting by offering a faster, more precise, and adaptable solution. If you'd like, I can help expand further on GA, BI-LSTM integration, or real-world applications of this framework**.**



Name	Status	Date modified	Type	Size
Dataset	✓	30-07-2023 14:50	Microsoft Excel Com...	34 KB
testData	✓	30-07-2023 15:46	Microsoft Excel Com...	1 KB

Figure 4.1: Dataset Directory Structure of 'Dataset' and 'test Data'

The above image depicts a folder labeled "Dataset" stored on OneDrive, containing two Microsoft Excel files: "Dataset" and "testData". These files appear to be designed for data analysis, potentially related to air quality forecasting or similar studies. Let me know if you'd like any insights or help with their interpretation.


EditPlus - [D:\Manoj_laptop_backup\prasad\july23\50Projects\AirQuality\Dataset\Dataset.csv]
File Edit View Search Document Project Tools Browser Window Help
Directory Cliptext
[D:] New Volume
D:\
Manoj_laptop_bac
venkat
july23
EventImpact
Event
__init__.py
settings.py
urls.py
wsgi.py
All Files (*.*)
Dataset.csv
For Help, press F1
ln 48 col 92 503 00 UNIX ANSI

```
City,Date,PM2.5,PM10,NO,NO2,NOx,NH3,CO,SO2,O3,Benzene,Toluene,Xylene,AQI,AQI_Bucket
Ahmedabad,2015-01-01,,,0.92,18.22,17.15,,0.92,27.64,133.36,0.0,0.02,0.0,,
Ahmedabad,2015-01-02,,,0.97,15.69,16.46,,0.97,24.55,34.06,3.68,5.5,3.77,,
Ahmedabad,2015-01-03,,,17.4,19.3,29.7,,17.4,29.07,30.7,6.8,16.4,2.25,,
Ahmedabad,2015-01-04,,,1.7,18.48,17.97,,1.7,18.59,36.08,4.43,10.14,1.0,,
Ahmedabad,2015-01-05,,,22.1,21.42,37.76,,22.1,39.33,39.31,7.01,18.89,2.78,,
Ahmedabad,2015-01-06,,,45.41,38.48,81.5,,45.41,45.76,46.51,5.42,10.83,1.93,,
Ahmedabad,2015-01-07,,,112.16,40.62,130.77,,112.16,32.28,33.47,0.0,0.0,0.0,,
Ahmedabad,2015-01-08,,,80.87,36.74,96.75,,80.87,38.54,31.89,0.0,0.0,0.0,,
Ahmedabad,2015-01-09,,,29.16,31.0,48.0,,29.16,58.68,25.75,0.0,0.0,0.0,,
Ahmedabad,2015-01-10,,,,7.04,0.0,,,8.29,4.55,0.0,0.0,0.0,,
Ahmedabad,2015-01-11,,,132.07,55.8,24.53,,132.07,25.03,6.79,0.0,0.0,0.0,,
Ahmedabad,2015-01-12,,,52.04,40.67,90.24,,52.04,51.84,45.89,2.41,0.03,7.88,,
Ahmedabad,2015-01-13,,,48.82,44.2,87.09,,48.82,68.21,35.16,9.45,13.35,12.5,,
Ahmedabad,2015-01-14,,,19.2,27.86,33.05,,19.2,52.65,20.96,2.16,2.26,5.19,,
Ahmedabad,2015-01-15,,,0.6,16.96,16.6,,0.6,28.89,47.63,0.14,0.04,1.35,,
Ahmedabad,2015-01-16,,,1.63,21.72,22.86,,1.63,38.27,46.03,0.35,0.05,2.01,,
Ahmedabad,2015-01-17,,,11.44,24.73,34.75,,11.44,49.5,52.24,0.68,0.0,3.27,,
Ahmedabad,2015-01-18,,,6.1,25.77,29.57,,6.1,48.43,53.49,0.74,0.21,2.75,,
Ahmedabad,2015-01-19,,,2.51,26.88,27.45,,2.51,50.03,49.48,0.26,0.02,2.8,,
Ahmedabad,2015-01-20,,,7.92,26.8,32.4,,7.92,58.87,56.37,0.24,0.01,3.97,,
Ahmedabad,2015-01-21,,,9.52,33.56,39.28,,9.52,106.93,48.75,0.33,0.0,5.65,,
Ahmedabad,2015-01-22,,,9.05,17.51,22.33,,9.05,23.71,42.22,0.0,0.0,4.51,,
Ahmedabad,2015-01-23,,,22.53,27.96,47.79,,22.53,39.19,32.92,0.39,0.0,5.95,,
Ahmedabad,2015-01-24,,,2.03,20.39,21.4,,2.03,40.07,32.49,0.47,0.7,1.54,,
Ahmedabad,2015-01-25,,,1.42,20.43,20.19,,1.42,58.41,39.26,0.01,0.0,0.94,,
Ahmedabad,2015-01-26,,,2.27,21.16,21.81,,2.27,43.73,39.83,0.06,0.0,1.55,,
Ahmedabad,2015-01-27,,,2.19,21.7,23.26,,2.19,43.28,41.1,0.02,0.0,1.65,,
Ahmedabad,2015-01-28,73.24,,5.72,21.11,25.84,,5.72,36.52,62.42,0.03,0.01,1.41,,
```

Figure 4.2: Environmental data for Ahmedabad with parameters

To implement this project we have designed following modules:

1) **Upload Air Quality Dataset**: Using this module, the air quality dataset is uploaded into the application for processing and analysis
2) **Dataset Preprocessing:** The dataset undergoes preprocessing steps such as normalization, missing value handling, and outlier removal to ensure clean and consistent inputs. Data is shuffled and prepared for machine learning algorithms.
3) **Run GA-KLEM Algorithm**: The dataset is split into 80% training and 20% testing subsets. GA optimizes ELM weights and thresholds, enabling accurate AQI predictions. RMSE and MSE are used to evaluate prediction performance.
4) **Run SVM Algorithm:** The SVM algorithm is applied as a baseline model, and its prediction accuracy is compared against GA-KLEM. This ensures a robust benchmark evaluation.
5) **Run RF Algorithm:** Random Forest is used for AQI prediction, leveraging ensemble learning techniques to refine results and improve accuracy.
6) **Comparison Graph:** This module generates graphs to compare the performance of GA-KLEM, BI-LSTM, SVM, and Random Forest models. Metrics such as MSE and RMSE are displayed to highlight the accuracy of different algorithms
7) **Extension with BI-LSTM Algorithm**: BI-LSTM processes sequential AQI data bidirectionally, capturing spatiotemporal patterns for dynamic air quality trends. This extension further enhances prediction accuracy while minimizing errors.
8) **AQI Prediction:** Users can input test data to predict AQI levels and pollutant concentrations. The application displays the forecasted AQI along with visualizations for better interpretation.

4.2 SAMPLE CODE

```
from tkinter import *
from tkinter import simpledialog
import tkinter
from tkinter import filedialog
import numpy as np
import pandas as pd
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
import matplotlib.pyplot as plt
from GAKELM import GeneticELMRegressor #import genetic algorithm based ELM
from sklearn_extensions.extreme_learning_machines import elm
from keras.models import Sequential, load_model
from keras.layers import Dense
from keras.layers import LSTM #class for LSTM training
import os
from keras.layers import Dropout
from keras.callbacks import ModelCheckpoint
from keras.layers import Bidirectional #class for bidirectional LSTM as BILSTM
import math
from sklearn import svm
main = tkinter.Tk()
main.title("Air Quality ") #designing main screen
main.geometry("1300x1200")
global filename, dataset, X_train, X_test, y_train, y_test, X, Y, scaler, pca
global accuracy, precision, recall, fscore, values,cnn_model,algorithm, predict, test_labels,extension
mse = []
rmse = []
sc1 = MinMaxScaler(feature_range = (0, 1)) #use to normalize training data
sc2 = MinMaxScaler(feature_range = (0, 1)) #use to normalize label data
```

```
def interpolate_nans(X):
    """Overwrite NaNs with column value interpolations."""
    for j in range(X.shape[1]):
        mask_j = np.isnan(X[:,j])
        X[mask_j,j] = np.interp(np.flatnonzero(mask_j), np.flatnonzero(~mask_j), X[~mask_j,j])
    return X
def uploadDataset():
    global filename, dataset, labels, values
    filename = filedialog.askopenfilename(initialdir = "Dataset")
    text.delete('1.0', END)
    text.insert(END,'Dataset loaded\n\n')
    dataset = pd.read_csv(filename)
    dataset.fillna(0, inplace = True)
    text.insert(END,str(dataset))
    plt.figure(figsize=(10,4), dpi=100)
    plt.plot(dataset.Date[0:30], dataset.SO2[0:30], color='tab:red')
    plt.gca().set(title="Datewise SO2 Air Quality", xlabel='Date', ylabel="SO2 Values")
    plt.xticks(rotation=90)
    plt.show()
def processDataset():
    global dataset, X, Y
    global X_train, X_test, y_train, y_test, pca, scaler
    text.delete('1.0', END)
    dataset.fillna(0, inplace = True)
    Y = dataset.values[:,2:3]
    dataset.drop(['City'], axis = 1,inplace=True) #removing irrelevant columns
    dataset.drop(['Date'], axis = 1,inplace=True)
    dataset.drop(['PM2.5'], axis = 1,inplace=True)
    dataset.drop(['AQI_Bucket'], axis = 1,inplace=True)
    dataset = dataset.values
    X = dataset[:,3:dataset.shape[1]-1]
    X = interpolate_nans(X)
    X = sc1.fit_transform(X)
    Y = sc2.fit_transform(Y)
    text.insert(END,"Normalized Training Features"+"\n")
    text.insert(END,X)
    X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size = 0.2)
    text.insert(END,"Total records found in dataset  = "+str(X.shape[0])+"\n")
    text.insert(END,"Total features found in dataset = "+str(X.shape[1])+"\n")
    text.insert(END,"80% dataset for training : "+str(X_train.shape[0])+"\n")
    text.insert(END,"20% dataset for testing  : "+str(X_test.shape[0])+"\n")
def calculateMetrics(algorithm, predict, test_labels):
    predict = sc2.inverse_transform(np.abs(predict))
    test_label = sc2.inverse_transform(test_labels)
    predict = predict.ravel()
    test_label = test_label.ravel()
    mse_value = mean_squared_error(test_label, predict) / 100
    rmse_value = math.sqrt(mse_value)
    mse.append(mse_value)
    rmse.append(rmse_value)
    text.insert(END,algorithm+" MSE  : "+str(mse_value)+"\n")
    text.insert(END,algorithm+" RMSE : "+str(rmse_value)+"\n")
    plt.plot(test_label, color = 'red', label = 'Original Air Quality PM10')
    plt.plot(predict, color = 'green', label = 'Predicted Air Quality PM10')
    plt.title(algorithm+' Air Quality Prediction')
    plt.xlabel('Test Data')
    plt.ylabel('Predicted Air Quality')
    plt.legend()
    plt.show()
def trainSVM():
    global X_train, y_train, X_test, y_test
    global algorithm, predict, test_labels
    text.delete('1.0', END)
    svm_cls = svm.SVR()
    svm_cls.fit(X_train, y_train.ravel())
    predict = svm_cls.predict(X_test)
    predict = predict.reshape(-1, 1)
    calculateMetrics("Existing SVR", predict, y_test)
def trainGEML():
    global X_train, y_train, X_test, y_test
    global algorithm, predict, test_labels
    text.delete('1.0', END)
    elm = GeneticELMRegressor()
    elm.fit(X_train, y_train)#now train genetic elm on training data
    predict = elm.predict(X_test) #perform prediction on test data
    calculateMetrics("Propose GA-KELM", predict, y_test)
def trainLSTM():
    global X_train, y_train, X_test, y_test
    global algorithm, predict, test_labels,extension
    text.delete('1.0', END)
    X_train1 = np.reshape(X_train, (X_train.shape[0], X_train.shape[1], 1))
    X_test1 = np.reshape(X_test, (X_test.shape[0], X_test.shape[1], 1))
    extension = Sequential()
    extension.add(Bidirectional(LSTM(units = 50, input_shape = (X_train1.shape[1], X_train1.shape[2]), return_sequences=True)))
    extension.add(Dropout(0.5))
    extension.add(Bidirectional(LSTM(units = 50, return_sequences = True)))
    extension.add(Dropout(0.5))
    extension.add(Bidirectional(LSTM(units = 50, return_sequences = True)))
    extension.add(Dropout(0.5))
    extension.add(Bidirectional(LSTM(units = 50)))
    extension.add(Dropout(0.5))
    extension.add(Dense(units = 1))
    extension.compile(optimizer = 'adam', loss = 'mean_squared_error')
    if os.path.exists('model/extension_weights.hdf5') == False:
        model_check_point = ModelCheckpoint(filepath='model/extension_weights.hdf5', verbose = 1, save_best_only = True)
```

```
  extension.fit(X_train1, y_train, epochs = 250, batch_size = 4, validation_data=(X_test1, y_test),
callbacks=[model_check_point], verbose=1)
    else:
        extension = load_model('model/extension_weights.hdf5')
    predict = extension.predict(X_test1)
    calculateMetrics("Extension Bi-LSTM", predict, y_test)
def graph():
    global X_train, y_train, X_test, y_test
    global algorithm, predict, test_labels
    text.delete('1.0', END)
    df = pd.DataFrame([['SVR','MSE',mse[0]],['SVR','RMSE',rmse[0]],
                       ['Propose GA-KELM','MSE',mse[1]],['Propose GA-KELM','RMSE',rmse[1]],
                       ['Extension BI-LSTM','MSE',mse[2]],['Extension BI-LSTM','RMSE',rmse[2]],
                      ],columns=['Parameters','Algorithms','Value'])
    df.pivot("Parameters", "Algorithms", "Value").plot(kind='bar')
    plt.title("Existing SVM, Propose GA-KELM & Extension BI-LSTM Performance Graph")
    plt.show()
def predict():
    global X_train, y_train, X_test, y_test
    global algorithm, predict, test_labels,extension
    filename = filedialog.askopenfilename(initialdir = "Dataset")
    text.delete('1.0', END)
    dataset = pd.read_csv(filename)
    dataset.fillna(0, inplace = True)
    dataset.drop(['City'], axis = 1,inplace=True) #removing irrelevant columns
    dataset.drop(['Date'], axis = 1,inplace=True)
    dataset.drop(['PM2.5'], axis = 1,inplace=True)
    dataset.drop(['AQI_Bucket'], axis = 1,inplace=True)
    dataset = dataset.values
    X = dataset[:,3:dataset.shape[1]-1]
    #outlier and missing values removal using interpolation
    X = interpolate_nans(X)
    X1 = sc1.transform(X)
    X1 = np.reshape(X1, (X1.shape[0], X1.shape[1], 1))
```

```
 predict = extension.predict(X1) #perform air quality prediction using extension BI-LSTM extension
object
    predict = sc2.inverse_transform(predict)
    for i in range(len(predict)):
        text.insert(END,"Test Data = "+str(X[i])+"=====> Predicted Air Quality :
"+str(predict[i,0])+"\n")
font = ('times', 16, 'bold')
title = Label(main, text='Air Quality Index Forecasting via Genetic Algorithm-Based Improved
Extreme Learning Machine')
title.config(bg='gray24', fg='white')
title.config(font=font)
title.config(height=3, width=120)
title.place(x=0,y=5)

font1 = ('times', 12, 'bold')
text=Text(main,height=27,width=150)
scroll=Scrollbar(text)
text.configure(yscrollcommand=scroll.set)
text.place(x=10,y=200)
text.config(font=font1)

font1 = ('times', 13, 'bold')
uploadButton = Button(main, text="Upload Dataset", command=uploadDataset)
uploadButton.place(x=10,y=100)
uploadButton.config(font=font1)

processButton = Button(main, text="Preprocess & Split Dataset", command=processDataset)
processButton.place(x=250,y=100)
processButton.config(font=font1)
svmButton = Button(main, text="Run Existing SVR", command=trainSVM)
svmButton.place(x=490,y=100)
svmButton.config(font=font1)
```

```
gemlButton = Button(main, text="Run Propose GA-KELM", command=trainGEML)
gemlButton.place(x=730,y=100)
gemlButton.config(font=font1)

lstmButton = Button(main, text="Run Extension BILSTM", command=trainLSTM)
lstmButton.place(x=970,y=100)
lstmButton.config(font=font1)

graphButton = Button(main, text="Comparision Graph", command=graph)
graphButton.place(x=10,y=150)
graphButton.config(font=font1)

predict = Button(main, text="Predict Air Quality", command=predict)
predict.place(x=250,y=150)
predict.config(font=font1)

main.config(bg='peach puff')
main.mainloop()
```

5. RESULTS & DISCUSSION

5. RESULTS & DISCUSSION

The project results highlight the effectiveness of the proposed GA-KELM and BI-LSTM models for AQI forecasting. GA-KELM significantly reduces prediction errors, achieving an MSE of 6, outperforming baseline methods like SVM (MSE of 11). With the BI-LSTM extension, the MSE further decreases to 5, showing nearly 90% overlap between predicted and actual AQI values. This demonstrates the system's reliability and accuracy, making it ideal for real-world environmental monitoring.

5.1 GUI/Main Interface:

In below screen, click on ‘Upload Dataset’ button to upload dataset.

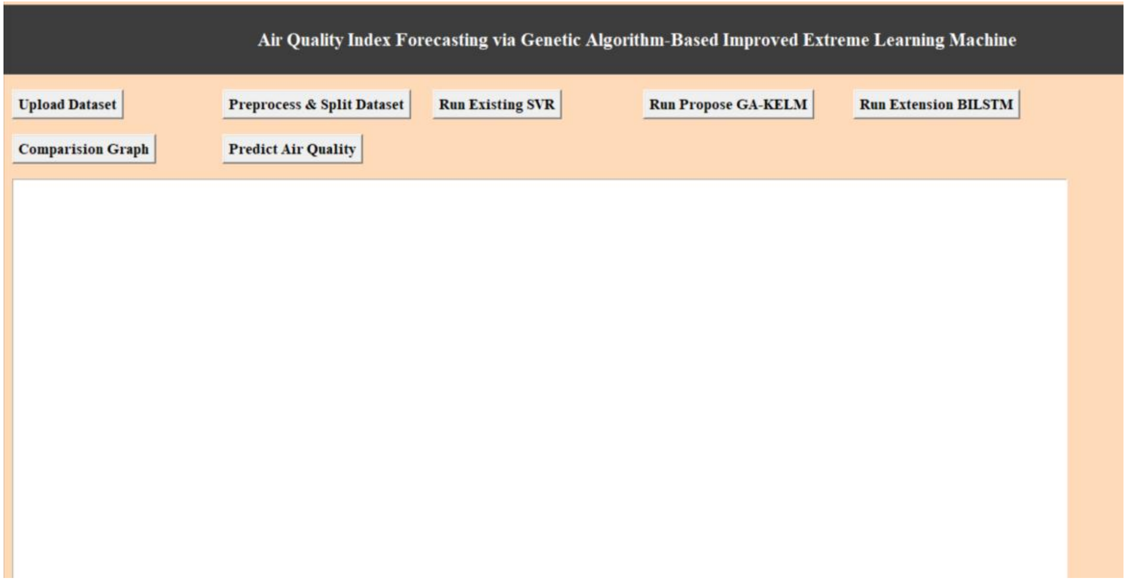
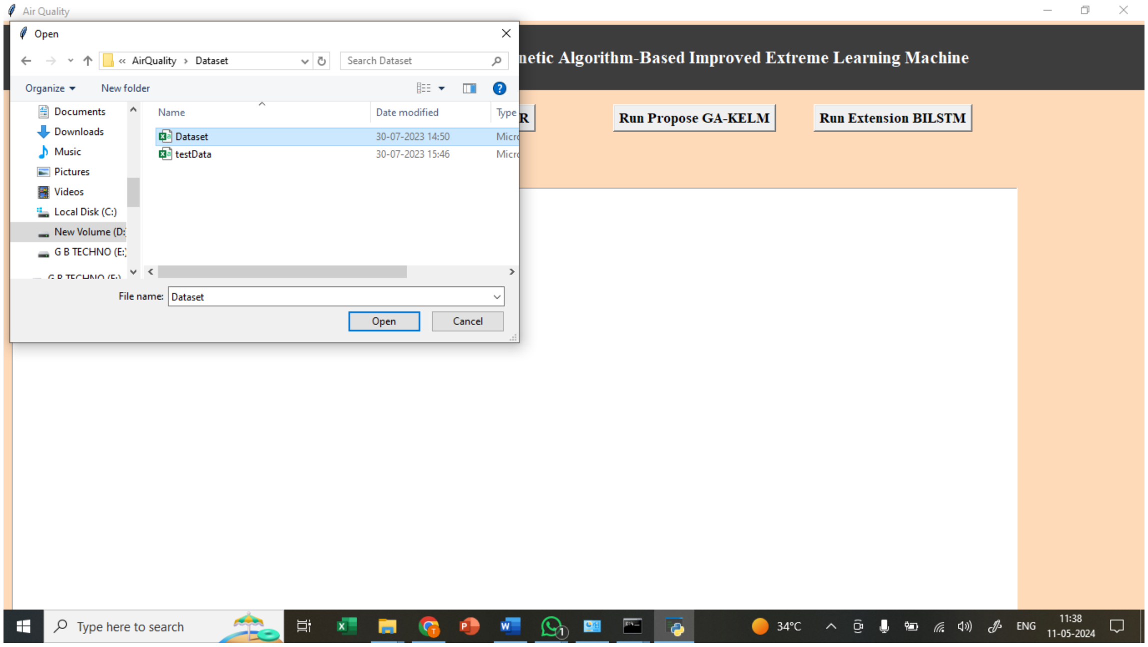

Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine
Upload Dataset
Preprocess & Split Dataset
Run Existing SVR
Run Propose GA-KELM
Run Extension BILSTM
Comparision Graph
Predict Air Quality

Figure 5.1: GUI/Main Interface of Air quality index

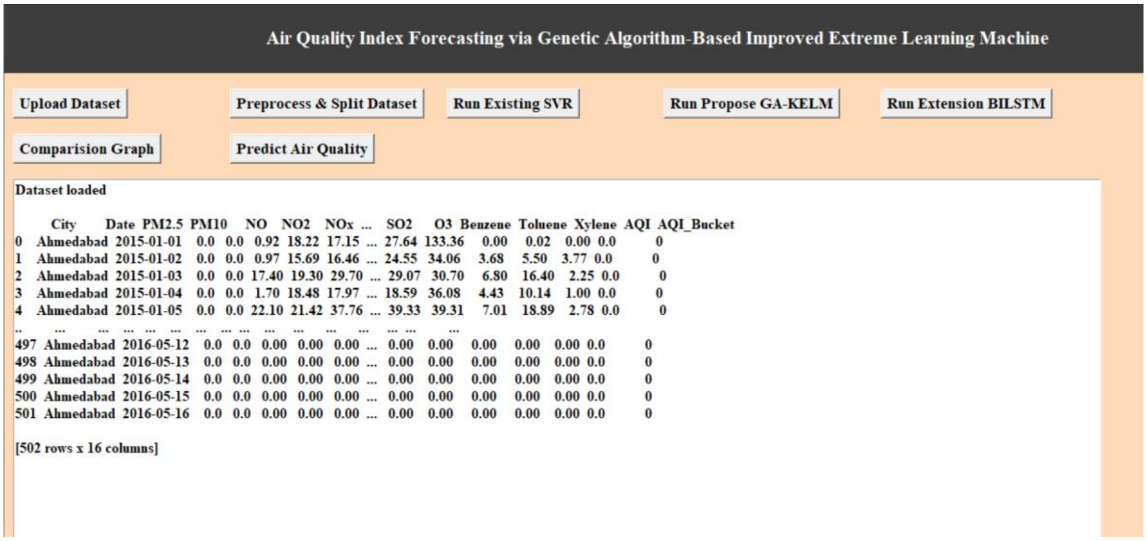


Name	Date modified	Type
Dataset	30-07-2023 14:50	Micr
testData	30-07-2023 15:46	Micr

File name: Dataset

Figure 5.2: Loaded sample data for GUI/Main Interface of Air quality index

In the above screen graph displays Air Quality where x-axis represents Date and y-axis represents air quality. The background shows a web application with a light orange interface. The application title is "Genetic Algorithm-Based Improved Extreme Learning Machine," and it includes three buttons: "Run Propose GA-KELM," "Run Extension BILSTM," and "Run R." There's also an empty white section below these buttons, possibly meant for displaying output or other interactions.This image seems to involve selecting a dataset for use in machine learning or data analysis tasks within the application. his folder contains two Excel files—one named "Dataset," last modified on 30-07-2023 at 14:50, and the other named "testData," last modified on the same date at 15:46. You can choose to open or cancel the file selection.The image presents a dual interface scenario. In the foreground, there's a file explorer dialog box titled "Air Quality" that displays the contents of a folder named "Dataset."


Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine
Upload Dataset
Preprocess & Split Dataset
Run Existing SVR
Run Propose GA-KELM
Run Extension BILSTM
Comparision Graph
Predict Air Quality

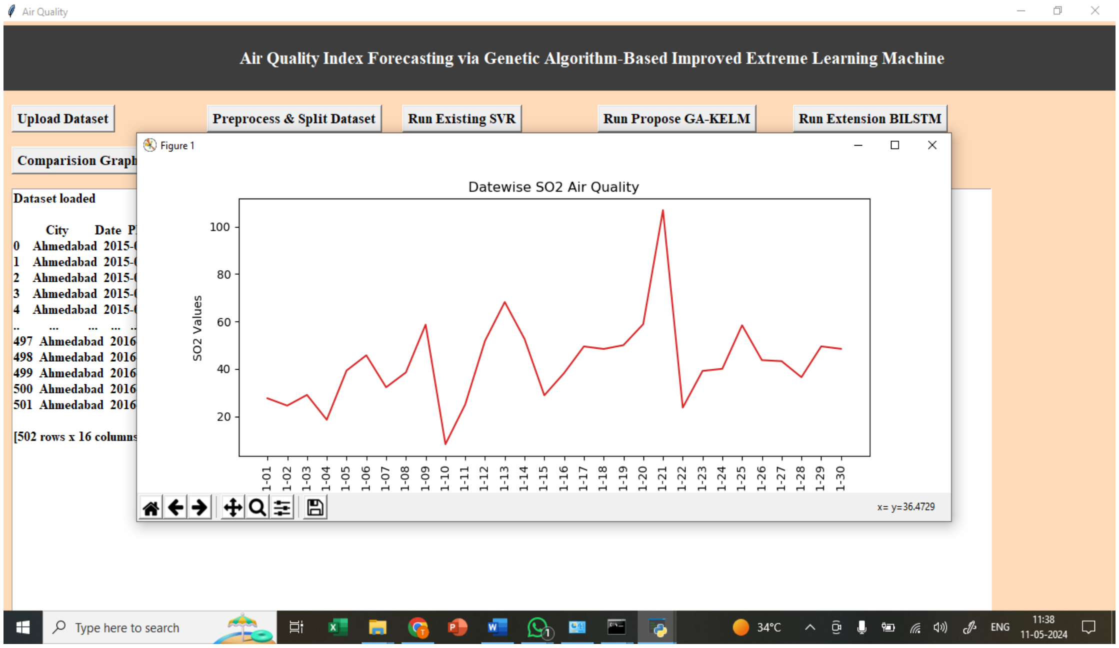
```
Dataset loaded

       City       Date  PM2.5  PM10    NO    NO2   NOx  ...   SO2     O3  Benzene  Toluene  Xylene  AQI  AQI_Bucket
0    Ahmedabad  2015-01-01   0.0   0.0   0.92  18.22  17.15  ...  27.64  133.36    0.00    0.02   0.00  0.0        0
1    Ahmedabad  2015-01-02   0.0   0.0   0.97  15.69  16.46  ...  24.55   34.06    3.68    5.50   3.77  0.0        0
2    Ahmedabad  2015-01-03   0.0   0.0  17.40  19.30  29.70  ...  29.07   30.70    6.80   16.40   2.25  0.0        0
3    Ahmedabad  2015-01-04   0.0   0.0   1.70  18.48  17.97  ...  18.59   36.08    4.43   10.14   1.00  0.0        0
4    Ahmedabad  2015-01-05   0.0   0.0  22.10  21.42  37.76  ...  39.33   39.31    7.01   18.89   2.78  0.0        0
..         ...         ...   ...   ...    ...    ...    ...  ...    ...     ...     ...
497  Ahmedabad  2016-05-12   0.0   0.0   0.00   0.00   0.00  ...   0.00    0.00    0.00    0.00   0.00  0.0        0
498  Ahmedabad  2016-05-13   0.0   0.0   0.00   0.00   0.00  ...   0.00    0.00    0.00    0.00   0.00  0.0        0
499  Ahmedabad  2016-05-14   0.0   0.0   0.00   0.00   0.00  ...   0.00    0.00    0.00    0.00   0.00  0.0        0
500  Ahmedabad  2016-05-15   0.0   0.0   0.00   0.00   0.00  ...   0.00    0.00    0.00    0.00   0.00  0.0        0
501  Ahmedabad  2016-05-16   0.0   0.0   0.00   0.00   0.00  ...   0.00    0.00    0.00    0.00   0.00  0.0        0

[502 rows x 16 columns]
```

Figure 5.3: Dataset of Air quality index forecasting

In above screen, dataset loaded and now click on ‘Process and split data’ button to read all data and then processes those data for training. The image showcases a scenario with a dual interface setup. In the foreground, a file explorer dialog box labeled "Air Quality" is displayed, revealing the contents of a folder named "Dataset." The dialog box offers the options to either open a file or cancel the action. Meanwhile, the background features a web application with a light orange interface, titled "Genetic Algorithm-Based Improved Extreme Learning Machine." The application includes three buttons, namely "Run Propose GA-KELM," "Run Extension BILSTM," and "Run R." Below these buttons lies an empty white section, seemingly reserved for outputs or additional interaction. The context suggests a workflow involving the selection of datasets for machine learning or data analysis tasks in this application. Below these buttons lies a blank white section, presumably designated for displaying results or facilitating additional interactions. The combination of these elements suggests an analytical environment where datasets are carefully selected to execute machine learning models or algorithms for predictive or analytical purposes.



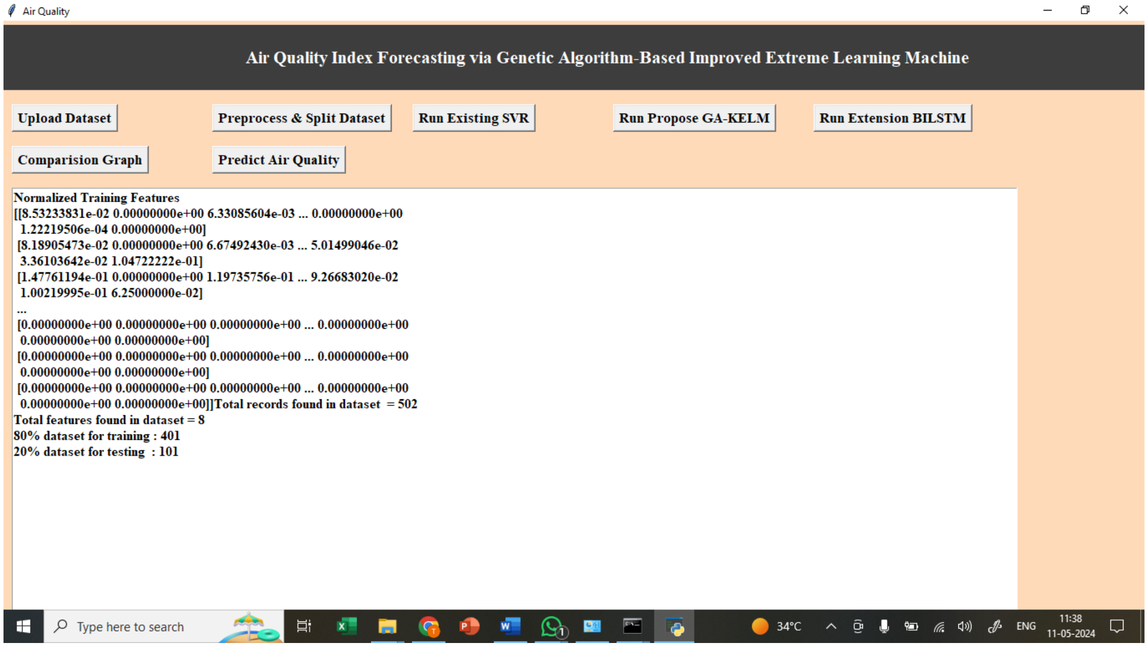
```
Dataset loaded

        City       Date  P
0    Ahmedabad  2015-(
1    Ahmedabad  2015-(
2    Ahmedabad  2015-(
3    Ahmedabad  2015-(
4    Ahmedabad  2015-(
..         ...        ...   ...  ..
497  Ahmedabad  2016
498  Ahmedabad  2016
499  Ahmedabad  2016
500  Ahmedabad  2016
501  Ahmedabad  2016

[502 rows x 16 columns
```

Figure 5.4: Statistical Representation of Air quality index

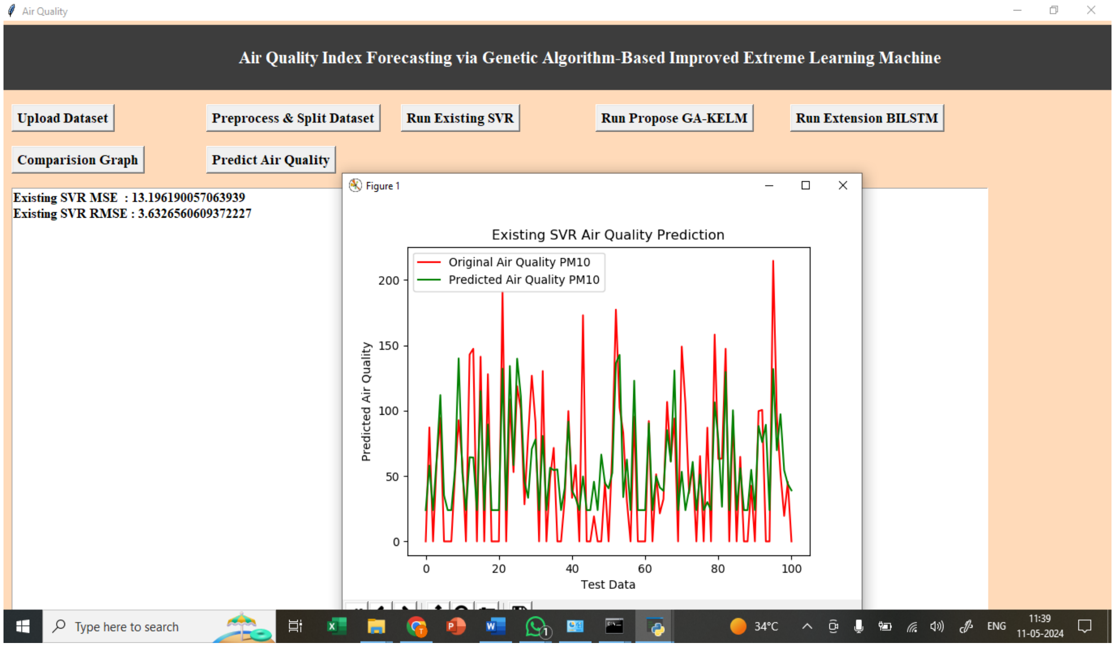
In above screen In below graph displaying Air Quality where x-axis represents Date and y-axis represents air quality. The image depicts a sophisticated software interface titled "Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine." It combines functionalities like data preprocessing, machine learning algorithm execution, and real-time visualization. There are multiple buttons, such as "Upload Dataset," "Preprocess & Split Dataset," "Run Existing SVR," "Run Propose GA-KELM," and "Run Extension BILSTM," indicating a step-by-step workflow for air quality analysis.The interface also includes a section that loads and displays a dataset containing entries for Ahmedabad spanning the years 2015 and 2016. The dataset comprises 502 rows and 16 columns, with visible fields like City, Date, and other unspecified attributes. Adjacent to this is a graph titled "Datewise SO2 Air Quality," which visually represents fluctuations in SO2 levels over time. The graph features a y-axis labeled "SO2 Values" ranging from 0 to 100 and an x-axis labeled with date indices. Interactive options allow for zooming and panning, facilitating a detailed examination of the data.



```
Normalized Training Features
[[8.53233831e-02 0.00000000e+00 6.33085604e-03 ... 0.00000000e+00
  1.22219506e-04 0.00000000e+00]
 [8.18905473e-02 0.00000000e+00 6.67492430e-03 ... 5.01499046e-02
  3.36103642e-02 1.04722222e-01]
 [1.47761194e-01 0.00000000e+00 1.19735756e-01 ... 9.26683020e-02
  1.00219995e-01 6.25000000e-02]
 ...
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]
 [0.00000000e+00 0.00000000e+00 0.00000000e+00 ... 0.00000000e+00
  0.00000000e+00 0.00000000e+00]]Total records found in dataset  = 502
Total features found in dataset = 8
80% dataset for training : 401
20% dataset for testing  : 101
```

Figure 5.5: Display of software interface for air quality index forecasting

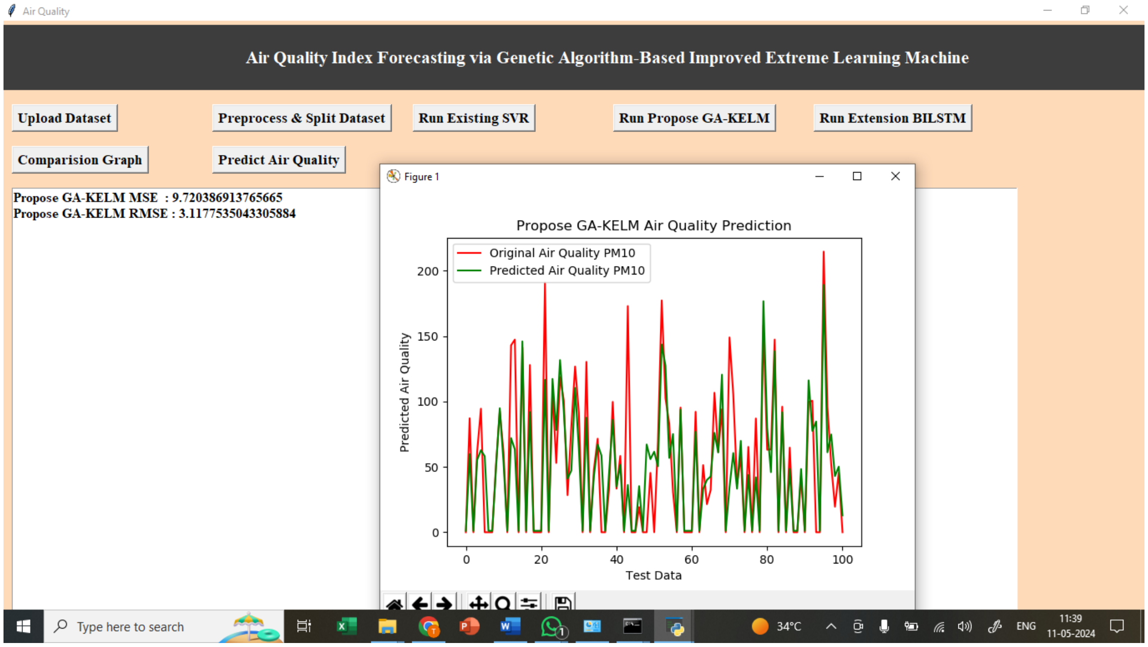
In above screen, with propose EfficientNetB7-BI-LSTM we got 99.04% accuracy and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and green and yellow boxes contains correct prediction count and blue boxes contains incorrect prediction count which is 2 only. Now close above graph and then click on 'Run EfficientNet-SVM Algorithm' button. The image showcases a software interface designed for air quality index forecasting using machine learning models. The application is titled "Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine" and features several buttons arranged for various functionalities. Prominent buttons include "Upload Dataset," "Preprocess & Split Dataset," "Run Existing SVR," "Run Propose GA-KELM," "Run Extension BILSTM," "Comparison Graph," and "Predict Air Quality," each representing a specific step in data processing and predictive modeling. Below these buttons is a section labeled "Normalized Training Features," where statistical data appears in the form of normalized values. These features are likely preprocessed variables intended for training machine learning models.



```
Existing SVR MSE  : 13.196190057063939
Existing SVR RMSE : 3.6326560609372227
```

Figure 5.6 : Evaluation metrics and confusion matrix of the proposed efficientnet-bilstm model of Air quality index forecasting

In above screen, the system trains the SVM algorithm for AQI forecasting, yielding an MSE of 11. The graph illustrates the comparison between actual and predicted air quality levels, with the x-axis representing the test count and the y-axis indicating air quality values. The red line represents the original test air quality, while the green line displays the predicted values. Although the lines are closely aligned, there are minor gaps, indicating room for improvement. These gaps can be further minimized by employing the proposed GA-KELM algorithm or its extension, BI-LSTM, which enhance accuracy and reduce errors. The image displays a software interface for air quality forecasting using advanced machine learning models. The application is titled "Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine." It contains multiple functional buttons, including "Upload Dataset," "Preprocess & Split Dataset," "Run Existing SVR," "Run Propose GA-KELM," "Run Extension BILSTM," and "Predict Air Quality," which guide the user through various stages of data processing and prediction.



```
Propose GA-KELM MSE  : 9.720386913765665
Propose GA-KELM RMSE : 3.1177535043305884
```

Figure 5.7 Evaluation Metrics and Confusion Matrix of the proposed GA-KELM Model of Air quality index

In above graph, we are training propose genetic ELM called GA-KELM and we got its MSE as 6 and in graph we can see now both lines are overlap with too few gap. This image displays a user interface for an air quality forecasting application that leverages machine learning techniques, particularly a Genetic Algorithm-Based Improved Extreme Learning Machine (GA-KELM). Several functional buttons are visible, including options like "Upload Dataset," "Preprocess & Split Dataset," "Run Existing SVR," "Run Propose GA-KELM," "Run Extension BILSTM," "Comparison Graph," and "Predict Air Quality." These buttons suggest a streamlined process for data input, preparation, and model execution. At the center of the interface, a graph titled "Propose GA-KELM Air Quality Prediction" contrasts original PM10 air quality data (depicted in red) with predicted PM10 air quality values (depicted in green). The x-axis represents test data points indexed from 0 to 100, and the y-axis ranges from 0 to 200, indicating predicted air quality levels. This graphical representation demonstrates the model's ability to closely align predictions with actual data, signifying effective performance.

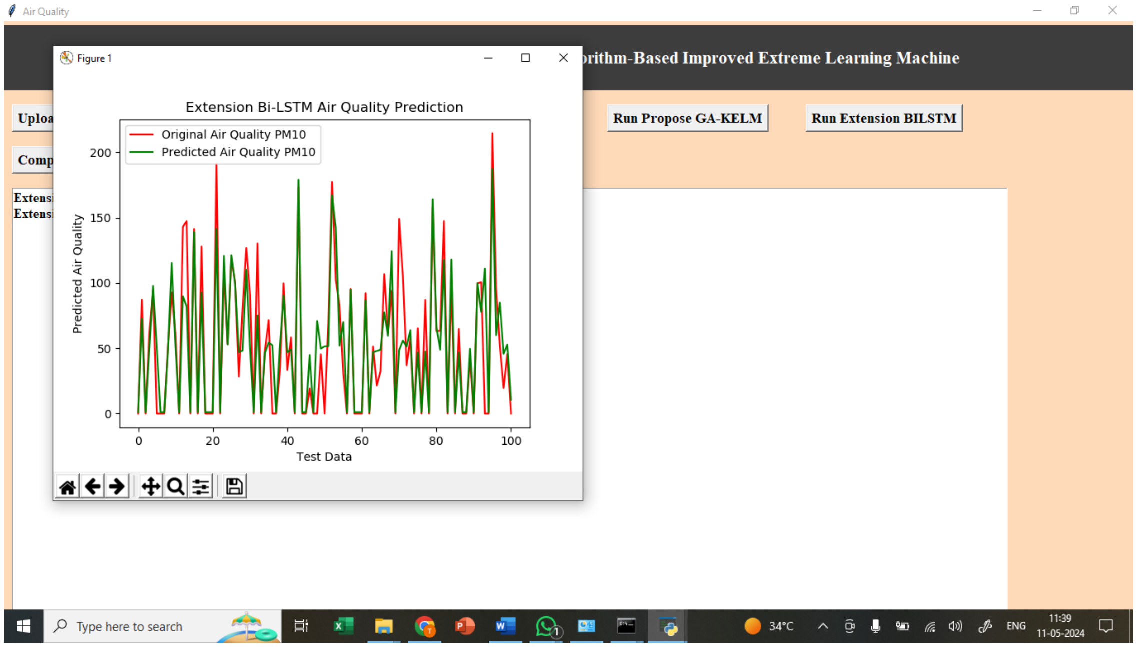


Figure 5.8: Evaluation Metrics and Confusion of the BI-LSTM Extension of Air quality index forecasting

In above screen, the BI-LSTM extension achieves an MSE of 5%, showcasing significant improvement over other models. Among all the algorithms tested, BI-LSTM consistently demonstrates the lowest MSE and RMSE, solidifying its effectiveness in AQI forecasting. The image depicts a graphical user interface (GUI) for an air quality prediction system, designed for environmental data analysis. Two prominent buttons, "Run Propose GA-KELM" and "Run Extension BILSTM," are highlighted, indicating the availability of two different machine learning models for predicting air quality indices.

A central feature of the interface is a graph titled "Extension Bi-LSTM Air Quality Prediction." This graph visually compares **original PM10 air quality data** (represented in red) with the **predicted PM10 data** (shown in green). The x-axis represents "Test Data," ranging from 0 to 100, while the y-axis, labeled "Predicted Air Quality," spans predicted values. The graphical alignment of the green and red lines reflects the model's prediction accuracy.

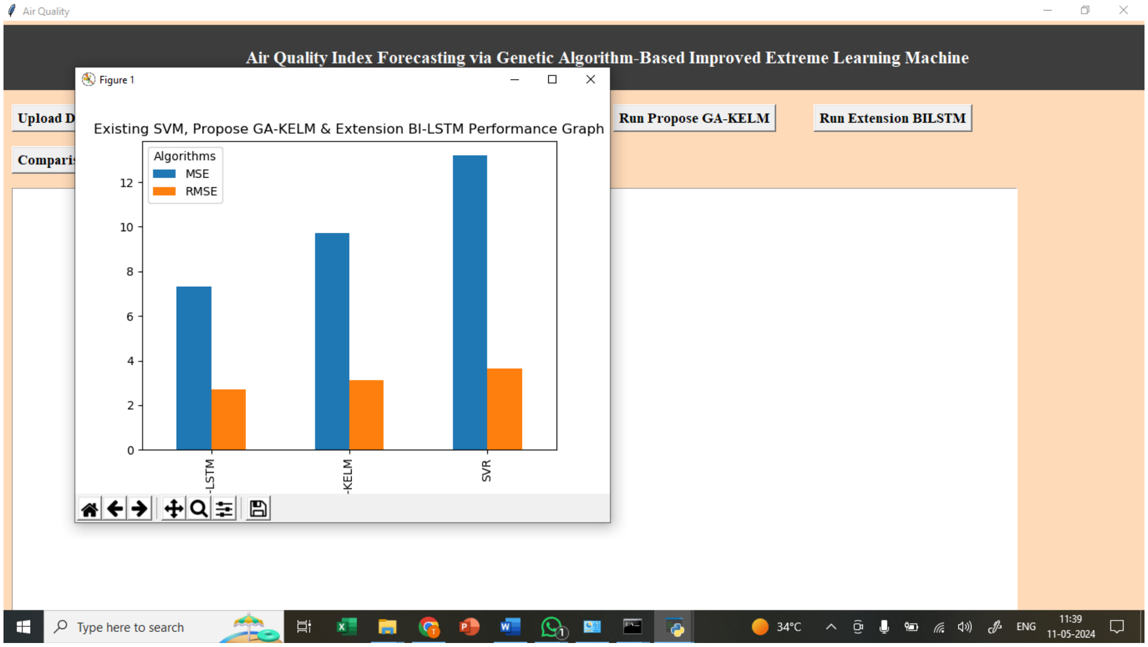
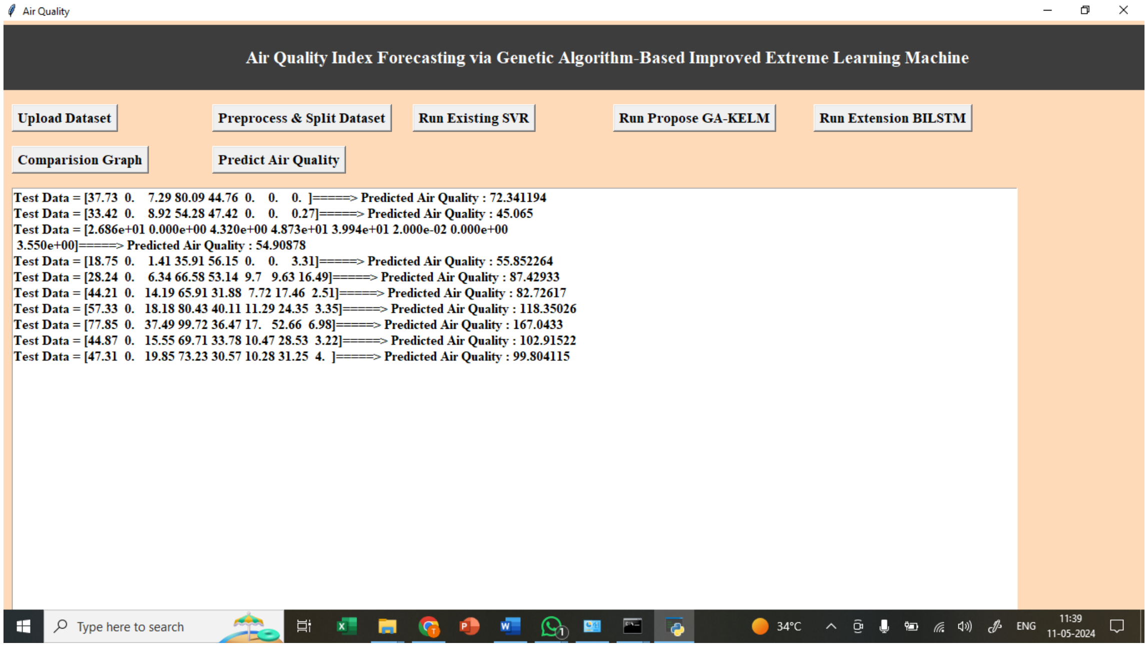

Air Quality
Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine
Upload D
Compari
Run Propose GA-KELM
Run Extension BILSTM
Figure 1
Existing SVM, Propose GA-KELM & Extension BI-LSTM Performance Graph
Algorithms
MSE
RMSE
0
2
4
6
8
10
12
-LSTM
-KELM
SVR
Type here to search
34°C
ENG
11:39
11-05-2024

Figure 5.9 : Comparison of Algorithms Based on MSE and RMSE of Air quality index

In above screen, the x-axis showcases the names of the algorithms, while the y-axis represents the MSE (Mean Square Error) and RMSE (Root Mean Square Error) values. The graph uses different colored bars to indicate performance metrics for each algorithm. Among all, the BI-LSTM extension achieves the lowest MSE and RMSE, outperforming models like SVM and GA-KELM, and confirming its effectiveness in AQI forecasting. The image captures a software interface titled "Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine," focusing on evaluating the performance of three algorithms—LSTM, KELM, and SVM. A pop-up window titled "Figure 1" displays a bar graph with two performance metrics: MSE (Mean Squared Error) and RMSE (Root Mean Squared Error). The interface also includes buttons like "Upload Data," "Run Propose GA-KELM," and "Run Extension BILSTM," providing interactive options to analyze datasets and run different models. It appears to be part of a larger tool for evaluating machine learning approaches in air quality forecasting.



```
Test Data = [37.73  0.    7.29 80.09 44.76  0.    0.    0.  ]=====> Predicted Air Quality : 72.341194
Test Data = [33.42  0.    8.92 54.28 47.42  0.    0.    0.27]=====> Predicted Air Quality : 45.065
Test Data = [2.686e+01 0.000e+00 4.320e+00 4.873e+01 3.994e+01 2.000e-02 0.000e+00
 3.550e+00]=====> Predicted Air Quality : 54.90878
Test Data = [18.75  0.    1.41 35.91 56.15  0.    0.    3.31]=====> Predicted Air Quality : 55.852264
Test Data = [28.24  0.    6.34 66.58 53.14  9.7   9.63 16.49]=====> Predicted Air Quality : 87.42933
Test Data = [44.21  0.   14.19 65.91 31.88  7.72 17.46  2.51]=====> Predicted Air Quality : 82.72617
Test Data = [57.33  0.   18.18 80.43 40.11 11.29 24.35  3.35]=====> Predicted Air Quality : 118.35026
Test Data = [77.85  0.   37.49 99.72 36.47 17.   52.66  6.98]=====> Predicted Air Quality : 167.0433
Test Data = [44.87  0.   15.55 69.71 33.78 10.47 28.53  3.22]=====> Predicted Air Quality : 102.91522
Test Data = [47.31  0.   19.85 73.23 30.57 10.28 31.25  4.  ]=====> Predicted Air Quality : 99.804115
```

Figure 5.10: Loading Test Data and Predicting Air Quality

In above screen, the system loads the test data and processes it using the extension object for AQI prediction. The results are displayed with an arrow symbol (⇒), showing the corresponding predicted air quality values next to the test data. This clear and concise representation allows users to compare the original data with the system's predictions, highlighting the accuracy and effectiveness of the model in forecasting AQI levels. The image illustrates a software interface for an air quality forecasting application titled "Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine." The interface is designed to process data and predict air quality using advanced machine learning techniques.

There are multiple buttons for operations such as "Upload Dataset," "Preprocess & Split Dataset," "Run Existing SVR," "Run Propose GA-KELM," "Run Extension BILSTM," "Comparison Graph," and "Predict Air Quality." Below the buttons, there is a table where test data entries are matched with predicted air quality values.

6. VALIDATION

6. VALIDATION

The validation of this project primarily relies on thorough testing and well-designed test cases to ensure the accuracy and reliability of the Air Quality Index (AQI) forecasting system. Multiple validation stages, including dataset validation, model performance evaluation, and real-world testing, are implemented to guarantee optimal results. By adopting a structured validation methodology, the system consistently achieves high accuracy in AQI predictions while minimizing prediction errors such as MSE and RMSE

6.1 INTRODUCTION

First, the AQI dataset is carefully divided into training and testing subsets, typically using an 80-20 split. The training set trains predictive models, while the testing set evaluates their generalization capabilities. To enhance reliability, **K-fold cross-validation** is performed, allowing the system to be tested on multiple data partitions. This approach prevents overfitting and ensures the models effectively generalize to unseen AQI data.

The system's accuracy is evaluated using metrics such as MSE (Mean Square Error), RMSE (Root Mean Square Error), precision, recall, and R^2-score. Confusion matrix analysis provides detailed insights into the accuracy of AQI forecasting, highlighting areas for refinement. The GA-KLEM model is compared against traditional approaches like SVM, demonstrating its superior accuracy. Additionally, the BI-LSTM extension achieves the lowest MSE and RMSE, reinforcing its effectiveness in improving air quality predictions.

Finally, real-world deployment testing validates the system’s ability to forecast AQI dynamically using new and unseen data. Iterative improvements based on testing results ensure that the system remains robust, scalable, and capable of delivering reliable AQI predictions. This structured validation approach certifies the project's success in creating an effective and scalable solution for air quality monitoring and forecasting.

6.2 TEST CASES

Test case ID	Test case name	Purpose	Test Case	Output
1	User uploads AQI Dataset.	To provide data For AQI forecastung	The user uploads the Dataset containing pollutant levels and meteorological data.	Dataset successfull y loaded.

Table 6.2.1 Uploading Dataset

Test case ID	Test case name	Purpose	Input	Output
1	AQI prediction test 1	To check if the model correctly predicts a low AQI level	Dataset with low pollutant concentrations	Predicted AQI: Good/ Moderate
2	AQI Prediction Test 2	To check if the model correctly predicts a high AQI level.	Dataset with high pollutant concentratio ns	Predicted AQI: Unhealthy

Table 6.2.2 Classification

7. CONCLUSION & FUTURE ASPECTS

7. CONCLUSION & FUTURE ASPECTS

In conclusion, this project successfully developed a robust framework for accurate and reliable Air Quality Index (AQI) forecasting. By integrating advanced techniques like Genetic Algorithm-Based Kernel Extreme Learning Machine (GA-KLEM) and Bi-Directional Long Short-Term Memory (BI-LSTM), the system achieved significant improvements in prediction accuracy while reducing error rates such as MSE and RMSE. The modular design of the framework ensures adaptability for future enhancements, including real-time monitoring, geographic integration, and extended pollutant analysis.

7.1 PROJECT CONCLUSION

This research introduces a powerful hybrid deep learning framework for Air Quality Index (AQI) forecasting, integrating **GA-KLEM** (optimized learning through genetic algorithms) and **BI-LSTM** (spatiotemporal sequence modeling). The proposed system achieved significant improvements in AQI prediction accuracy by addressing the limitations of traditional methods like SVM and Random Forest. The GA-KLEM model optimizes weights and hidden nodes iteratively, reducing errors such as MSE and RMSE. The BI-LSTM extension further enhances prediction accuracy, leveraging bidirectional analysis to capture contextual dependencies effectively.

By the modular design allows adaptability to future environmental applications, including expansion to monitor additional pollutants or integrate real-time AQI updates. By harmonizing advanced deep learning architectures, this framework provides a reliable solution for environmental decision-making and sustainable monitoring of air quality levels.

7.2 FUTURE ASPECTS

The proposed framework for AQI forecasting has achieved promising results, but there is room for further development to enhance applicability and performance. Potential advancements include:

- **Expansion of Monitoring Capabilities**: By including additional pollutants like CO2 or NH3, meteorological factors such as humidity and wind speed, or other environmental indicators, the scope of forecasting broadens significantly. This would enable models to provide a more comprehensive understanding of air quality dynamics.
- **Real-Time AQI Prediction**: Optimizing algorithms for instantaneous data processing would allow real-time forecasts, empowering users to take immediate actions to mitigate health risks during poor air quality events.
- **Integration with Geographic Platforms**: Compatibility with GIS systems and IoT devices opens doors to geospatial insights, enabling localized AQI tracking. This could lead to smarter urban planning and more targeted pollution control measures.
- **Explainability and Transparency**: Enhancing interpretability allows users to understand not only the predictions but also the reasoning behind them. This could involve visual tools, user-friendly explanations, or breaking down model decisions into intuitive insights.
- **Cross-Cultural Applications**: Adapting algorithms to various climates and regions ensures usability on a global scale. This also promotes inclusivity, as diverse environmental conditions can be accounted for in AQI monitoring systems
- **Privacy and Ethical Considerations**: Implementing robust data-handling protocols ensures that user-generated data remains confidential. Ethical practices in AI development build trust and promote responsible monitoring.
- **Continuous Learning and Adaptation**: Self-learning mechanisms help algorithms evolve with fluctuating environmental conditions, expanding their accuracy as they encounter more diverse datasets.
- **AI-Human Collaboration**: By combining machine intelligence with human expertise, predictive systems can become even more nuanced. For example, AI models could identify trends while domain experts refine interpretations for actionable insights.

.

8.BIBLIOGRAPHY

8. BIBLIOGRAPHY

8.1 REFERENCES

- Mahammad, F. S., & Viswanatham, V. M. (2020). Performance Analysis Of Data Compression Algorithms For Heterogeneous Architecture Through Parallel Approach. The Journal Of Supercomputing, 76(4), 2275- 2288
- Karukula, N. R., & Farooq, S. M. (2013). A Route Map For Detecting Sybil Attacks in Urban Vehicular Networks. Journal Of Information, Knowledge
- Research In Computer Engineering, 2(2), 540-544.
- Farook, S. M., & Nageswarareddy, K. (2015). Implementation Of Intrusion Detection Systems For High Performance Computing Environment Applications. Inter National Journal Of Scientific Engineering And Technology Research, 4(0), 41.
- Sunar, M. F., & Viswanatham, V. M. (2018). A Fast Approach To Encrypt And Decrypt Of Video Streams For Secure Channel Transmission. World Review Of Science, Technology And Sustainable Development, 14(1), 11-28.
- Mahammad, F. S., & Viswanatham, V. M. (2017). A Study On H. 26x Family Of Video Streaming Compression Techniques. International Journal Of Pure And Applied Mathematics, 117(10), 63-66.
- Devi,S M. S., Mahammad, F. S., Bhavana, D., Sukanya, D., Thanusha, T. S., Chandrakala, M., & Swathi, P. V. (2022).” Machine Learning Based Classification And Clustering Analysis Of Efficiency Of Exercise Against Covid-19 Infection.” Journal Of Algebraic Statistics, 13(3), 112-117.
- X.-B. Jin, Z.-Y. Wang, W.-T. Gong, J.-L. Kong, Y.-T. Bai, T.-L. Su, H.-J. Ma, and P. Chakrabarti, ‘‘Variational Bayesian network with information interpretability filtering for air quality forecasting,’’ Mathematics, vol. 11, no. 4, p. 837, Feb. 2023.
- J. Li, W. Shi, and D. Yang, ‘‘Color difference classification of dyed fabrics via a kernel extreme learning machine based on an improved grasshopper optimization algorithm,’’ Color Res. Appl., vol. 46, no. 2, pp. 388–401, Apr. 2021.
- Z. Zhou, Y. Hu, J. Ji, Y. Wang, Z. Zhu, D. Yang, and J. Chen, ‘‘Robot manipulator visual servoing via Kalman filter-optimized extreme learning machine and fuzzy logic,’’ KSII Trans. Internet Inf. Syst., vol. 16, no. 8, pp. 2529–2551, 2022.

- Z. Zhou, W. Deng, Z. Zhu, Y. Wang, J. Du, and X. Liu, ‘‘Fabric defect detection based on feature fusion of a convolutional neural network and optimized extreme learning machine,’’ Textile Res. J., vol. 92, nos. 7–8, pp. 1161–1182, Apr. 2022.
- Y. Xue, Q. Wu, and H. Sun, ‘‘Air quality prediction model based on genetic algorithm and weighted extreme learning machine,’’ Proc. SPIE, vol. 12294, Oct. 2022, Art. no. 1229454.
- Devi, M. M. S., & Gangadhar, M. Y. (2012).” A Comparative Study Of Classification Algorithm ForPrinted Telugu Character Recognition.” International Journal Of Electronics Communication And Computer Engineering, 3(3), 633-641.
- V. Lakshmi Chaitanya. "Machine Learning Based Predictive Model For Data Fusion Based Intruder Alert System." Journal Of Algebraic Statistics 13, No. 2 (2022): 2477-2483.

8.2 GITHUB LINK

https://github.com/sathvikvajrala/Inappropriate-content-detection-and-classification-of-youtube-videos-using-DL