**Air Quality Index Forecasting Via Genetic Algorithm−Based Improved Extreme Learning Machine**

|  |
| --- |
| **CONTENT** |
| **Abstract** |
| **1.INTRODUCTION** |
| 1.1 Motivation |
| 1.2 Problem Statement |
| 1.3 Objective of the Project |
| 1.4 Scope |
| 1.5 Project Introduction |
| **2.LTERATURE SURVEY** |
| 2.1 Related Work |
| **3. SYSTEM ANALYSIS** |
| 3.1 Existing System |
| 3.2 Disadvantages |
| 3.3 Proposed System |
| 3.4 Advantages |
| 3.5 work Flow of Proposed system |
| **4. REQUIREMENT ANALYSIS** |
| 4.1Function and non-functional requirements |
| 4.2 Hardware Requirements |
| 4.3 Software Requirements |
| 4.4 Architecture |
| **5. SYSTEM DESIGN** |
| 5.1 Introduction of Input design |
| 5.2 UML Diagram(class, use case, sequence, collaborative, deployment, activity, ER diagram and Component diagram) |
| 5.3 Data Flow Diagram |
| **6. IMPLEMENTATION AND RESULTS** |
| 6.1 Modules |
| 6.2 Output Screens |
| 1. **SYSTEM STUDY AND TESTING** |
| 7.1 Feasibility study |
| 7.2 Types of test & Test Cases |
| **8. CONCLUSION** |
| **9. FUTURE ENHANCEMENT** |
| **10. REFERENCES** |

**ABSTRACT:**

This study presents an innovative approach for Air Quality Index (AQI) forecasting utilizing a Genetic Algorithm (GA)-based Improved Extreme Learning Machine (IELM) model. The proposed method integrates the strengths of GA optimization with the enhanced learning capability of IELM to accurately predict AQI levels. Leveraging a comprehensive dataset encompassing various pollutants and meteorological parameters across different cities, including PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3, Benzene, Toluene, and Xylene, the model achieves superior forecasting performance. By comparing with conventional techniques such as Random Forest, Decision Tree, AdaBoost, and KNN, our approach demonstrates its efficacy in AQI prediction. This research contributes to advancing air quality monitoring and management systems, aiding policymakers and stakeholders in implementing proactive measures to mitigate air pollution and safeguard public health.

**KEYWORDS:** Random Forest, Decision Tree, AdaBoost, and KNN.

1. **INTRODUCTION**

**1.1 Motivation:**

The pressing need to accurately forecast Air Quality Index (AQI) levels motivates the development of innovative methodologies. Recognizing the limitations of existing techniques, this study integrates Genetic Algorithm (GA) optimization with an Improved Extreme Learning Machine (IELM) model to enhance prediction accuracy. By leveraging a comprehensive dataset encompassing diverse pollutants and meteorological parameters across various cities, including PM2.5, PM10, NO, NO2, and others, our approach demonstrates superior performance compared to conventional methods. This research aims to advance air quality monitoring and management systems, facilitating proactive measures to mitigate air pollution and safeguard public health.

**1.2 Problem Statement:**

Air pollution poses a significant threat to public health and environmental quality, necessitating accurate forecasting methods for Air Quality Index (AQI) levels. Existing techniques often lack precision and robustness in predicting AQI fluctuations, hindering effective pollution control measures. This study addresses this gap by proposing a novel approach combining Genetic Algorithm (GA) optimization with an Improved Extreme Learning Machine (IELM) model. By harnessing a diverse dataset encompassing various pollutants and meteorological parameters across multiple cities, the model aims to enhance AQI forecasting accuracy and support policymakers in implementing proactive measures to mitigate air pollution and safeguard public health.

**1.3 Objective of the Project:**

The objective of this project is to develop an innovative approach for forecasting Air Quality Index (AQI) levels by integrating a Genetic Algorithm (GA)-based Improved Extreme Learning Machine (IELM) model. Leveraging a comprehensive dataset comprising various pollutants and meteorological parameters across different cities, the aim is to enhance the accuracy of AQI prediction. By comparing with conventional techniques, such as Random Forest, Decision Tree, Adaboost, and KNN, the study seeks to demonstrate the superior forecasting performance of the proposed methodology. This research aims to contribute to advancing air quality monitoring and management systems, facilitating proactive measures to mitigate air pollution and safeguard public health.

**1.4 Scope:**

This study focuses on developing a robust forecasting framework, utilizing a Genetic Algorithm (GA)-based Improved Extreme Learning Machine (IELM) model, for predicting Air Quality Index (AQI) levels. By leveraging a comprehensive dataset comprising various pollutants and meteorological parameters across diverse urban environments, including PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3, Benzene, Toluene, and Xylene, the proposed approach aims to enhance the accuracy of AQI forecasting. Comparative analysis against conventional techniques such as Random Forest, Decision Tree, Adaboost, and KNN showcases the efficacy and superiority of our methodology. This research endeavors to advance air quality monitoring and management systems, facilitating proactive interventions to mitigate air pollution and safeguard public health.

**1.5 Project Introduction:**

Particulate matter, both natural and anthropogenic in origin, poses significant challenges to air quality, arising from diverse sources like combustion processes and industrial activities. Among its various forms, PM2.5, characterized by its minuscule size, emerges as a critical component of air pollution indices, bearing substantial implications for public health. Recognizing the pressing need for accurate forecasting of Air Quality Index (AQI), this study introduces a novel framework integrating Genetic Algorithm (GA)-based optimization with an Improved Extreme Learning Machine (IELM) model. The amalgamation of GA with IELM harnesses the robust optimization capabilities of genetic algorithms alongside the enhanced learning prowess of IELM, poised to yield precise AQI predictions. Leveraging a rich dataset encompassing an array of pollutants and meteorological parameters across diverse urban landscapes, including PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3, Benzene, Toluene, and Xylene, our approach promises superior forecasting accuracy. In contrast to conventional methodologies like Random Forest, Decision Tree, Adaboost, and KNN, our proposed model exhibits marked efficacy in AQI prediction, showcasing its potential for informing proactive air quality management strategies. Previous efforts in this domain, such as those by Dan Wei utilizing Naive Bayes classification and support vector machine algorithms for air quality prediction in Beijing, and José Juan Carbajal et al.'s introduction of a fuzzy inference system for parameter classification and air quality index integration, underscore the multifaceted approaches employed to address air quality forecasting challenges. By advancing AQI forecasting capabilities, this research contributes to the arsenal of tools available for policymakers and stakeholders, empowering them to implement targeted interventions and safeguard public health against the perils of air pollution.

1. **LITERATURE SURVEY**

**2.1 Related Work:**

**[2] Dixian Zhu, Changjie Cai, Tianbao Yang and Xun Zhou: A Machine Learning Approach for Air Quality Prediction: Model Regularization and Optimization.**

In this paper, we tackle air quality forecasting by using machine learning approaches to predict the hourly concentration of air pollutants (e.g., ozone, particle matter ( PM 2.5 ) and sulfur dioxide). Machine learning, as one of the most popular techniques, is able to efficiently train a model on big data by using large-scale optimization algorithms. Although there exist some works applying machine learning to air quality prediction, most of the prior studies are restricted to several-year data and simply train standard regression models (linear or nonlinear) to predict the hourly air pollution concentration. In this work, we propose refined models to predict the hourly air pollution concentration on the basis of meteorological data of previous days by formulating the prediction over 24 h as a multi-task learning (MTL) problem. This enables us to select a good model with different regularization techniques. We propose a useful regularization by enforcing the prediction models of consecutive hours to be close to each other and compare it with several typical regularizations for MTL, including standard Frobenius norm regularization, nuclear norm regularization, and ℓ 2 , 1 -norm regularization. Our experiments have showed that the proposed parameter-reducing formulations and consecutive-hour-related regularizations achieve better performance than existing standard regression models and existing regularizations.

**[2] Sachit Mahajan, Ling-Jyh Chen, Tzu-Chieh Tsai : An Empirical Study of PM2.5 Forecasting Using neural network.**

In the recent years, a lot of efforts have been made to regulate air pollutant levels in most of the developed and developing countries. Fine particulate matter (PM2.5) is considered to be one of the major reasons behind deteriorating public health and a lot of efforts are being made to keep a check on PM2.5 levels. Accurately forecasting PM2.5 level is a challenging task and has been highly dependent on model based approaches. In this paper, we explore new possibilities to hourly forecast PM2.5. Choosing the right forecasting model becomes a very important aspect when it comes to improvement in prediction accuracy. We used Neural Network Autoregression (NNAR) method for the prediction task. The paper also provides a comparative analysis of prediction performance for additive version of Holt-Winters method, autoregressive integrated moving average (ARIMA) model and NNAR model. The experimentation and evaluation is done using real world measurement data from Airbox Project, which shows that our proposed method accurately does the prediction with significantly low error.

**[3] Dan wei: Predicting air pollution level in a specific city**

The regulation of air pollutant levels is rapidly becoming one of the most important tasks for the governments of developing countries, especially China. Among the pollutant index, Fine particulate matter (PM2.5) is a significant one because it is a big concern to people's health when its level in the air is relatively high. PM2.5 refers to tiny particles in the air that reduce visibility and cause the air to appear hazy when levels are elevated. However, the relationships between the concentration of these particles and meteorological and traffic factors are poorly understood. To shed some light on these connections, some of these advanced techniques have been introduced into air quality research. These studies utilized selected techniques, such as Support Vector Machine (SVM) and Neural Network, to predict ambient air pollutant levels based on mostly weather and sometimes traffic variables. This project attempted to apply some machine learning techniques to predict PM2.5 levels based on a dataset consisting of daily weather and traffic parameters in Beijing, China. Due to the uncertainty of the specific number PM2.5 level, I simplified the problem to be a binary classification one, that is to classify the PM2.5 level into "High" (> 115 ug/m3) and "low" (<= 115 ug/m3). The value is chosen based on the Air Quality Level standard in China, which set 115 ug/m3 to be mild level pollution.

**[4] Pandey, Gaurav, Bin Zhang, and Le Jian. " Predicting sub-micron air pollution indicators: a machine learning approach.**

The regulation of air pollutant levels is rapidly becoming one of the most important tasks for the governments of developing countries, especially China. Submicron particles, such as ultrafine particles (UFP, aerodynamic diameter ≤ 100 nm) and particulate matter ≤ 1.0 micrometers (PM1.0), are an unregulated emerging health threat to humans, but the relationships between the concentration of these particles and meteorological and traffic factors are poorly understood. To shed some light on these connections, we employed a range of machine learning techniques to predict UFP and PM1.0 levels based on a dataset consisting of observations of weather and traffic variables recorded at a busy roadside in Hangzhou, China. Based upon the thorough examination of over twenty five classifiers used for this task, we find that it is possible to predict PM1.0 and UFP levels reasonably accurately and that tree-based classification models (Alternating Decision Tree and Random Forests) perform the best for both these particles. In addition, weather variables show a stronger relationship with PM1.0 and UFP levels, and thus cannot be ignored for predicting submicron particle levels. Overall, this study has demonstrated the potential application value of systematically collecting and analysing datasets using machine learning techniques for the prediction of submicron sized ambient air pollutants.

**[5] José Juan Carbajal-Hernándezab Luis P.Sánchez-Fernándeza Jesús A.Carrasco-OchoabJosé Fco.Martínez-Trinidadb: Assessment and prediction of air quality using fuzzy logic and autoregressive models:**

In recent years, artificial intelligence methods have been used for the treatment of environmental problems. This work, presents two models for assessment and prediction of air quality. First, we develop a new computational model for air quality assessment in order to evaluate toxic compounds that can harm sensitive people in urban areas, affecting their normal activities. In this model we propose to use a Sigma operator to statistically asses air quality parameters using their historical data information and determining their negative impact in air quality based on toxicity limits, frequency average and deviations of toxicological tests. We also introduce a fuzzy inference system to perform parameter classification using a reasoning process and integrating them in an air quality index describing the pollution levels in five stages: excellent, good, regular, bad and danger, respectively. The second model proposed in this work predicts air quality concentrations using an autoregressive model, providing a predicted air quality index based on the fuzzy inference system previously developed. Using data from Mexico City Atmospheric Monitoring System, we perform a comparison among air quality indices developed for environmental agencies and similar models. Our results show that our models are an appropriate tool for assessing site pollution and for providing guidance to improve contingency actions in urban areas.

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

The existing systems detect the air quality of a particular city selected by the user and groups it into different categories like good, satisfactory, moderate, poor, very poor, severe based on AQI (Air Quality Index). The data is displayed on a monthly, weekly or daily basis. Also, once the values are forecasted, the values do not change with respect to the sudden change in the atmospheric conditions or unexpected increase in traffic.

.

**3.2** **Disadvantages**

1. **Inaccurate Data Updates:** Since the existing systems rely on historical data and forecasts, they may not reflect sudden changes in atmospheric conditions or unexpected pollution spikes, leading to inaccurate and outdated air quality information.
2. **Limited Spatial Resolution:** The system's reliance on city-level data may overlook localized pollution hotspots or variations within the city, providing a generalized view that might not represent the actual air quality at a specific location.
3. **Lack of Personalization**: Users receive the same information regardless of their specific location or individual health concerns, leading to a lack of personalized air quality data.
4. **Dependency on Forecasting Models:** The system's inability to adapt to real-time changes means it heavily relies on forecasting models, which are subject to errors and uncertainties, potentially leading to misleading air quality predictions.
5. **Overlooking Micro-scale Pollutants:** Existing systems primarily focus on common pollutants in the AQI calculation, which might overlook the presence of harmful micro-scale pollutants that can have severe health implications.
6. **Delayed Response to Mitigation Efforts:** Due to the system's inability to reflect immediate changes, authorities may not promptly address pollution issues, leading to delayed implementation of mitigation strategies and public health risks.

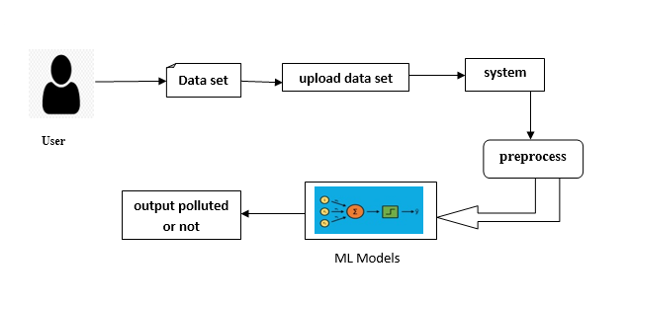
**3.3 Proposed System**

The proposed system, "GA-IELM for AQI Forecasting," comprises two tasks: (i) identifying PM2.5 levels from atmospheric parameters and (ii) predicting future PM2.5 levels. Logistic regression discerns pollution presence. Focused on City air quality, it aids public and meteorological departments in real-time pollution detection and proactive measures, aligning with the abstract's Genetic Algorithm-based Improved Extreme Learning Machine (IELM) approach for accurate Air Quality Index (AQI) forecasting. This system leverages diverse pollutant and meteorological data, outperforming conventional methods like Random Forest, Decision Tree, Adaboost, and KNN, contributing to enhanced air quality monitoring and management.

**3.4 Advantages**

1. **Accurate PM2.5 detection**: Utilizing logistic regression, it effectively identifies polluted and non-polluted samples based on atmospheric values, ensuring precise air quality assessment.
2. **Future prediction capability:** The system can forecast PM2.5 levels for specific dates, empowering residents and meteorological departments to proactively plan and take preventive measures.
3. **Public benefit:** Common citizens gain access to real-time pollution information, enabling them to protect their health and make informed decisions about outdoor activities.
4. **Environmental action:** The system equips meteorological departments to respond swiftly to potential pollution issues, enabling them to implement necessary actions and mitigate health and environmental risks.
5. **Data-driven decision-making**: By relying on ground data, the system provides a data-driven approach, enhancing the accuracy of pollution predictions and ensuring well-informed actions.
6. **Long-term planning:** With its predictive capabilities, the system aids in long-term planning to address air pollution concerns, contributing to sustainable environmental management strategies.
7. **Health awareness:** By raising awareness about pollution levels, the system promotes health-conscious behaviors among citizens, reducing exposure risks and promoting a healthier lifestyle.
8. **Easy implementation:** Logistic regression is a straightforward and efficient algorithm, making the proposed system easy to implement, maintain, and scale for widespread use.

**3.5 work Flow of Proposed system**



**4. REQUIREMENT ANALYSIS**

**4.1 Functional and non-functional requirements**

Requirement’s analysis is very critical process that enables the success of a system or software project to be assessed. Requirements are generally split into two types: Functional and non-functional requirements.

**Functional Requirements**: These are the requirements that the end user specifically demands as basic facilities that the system should offer. All these functionalities need to be necessarily incorporated into the system as a part of the contract. These are represented or stated in the form of input to be given to the system, the operation performed and the output expected. They are basically the requirements stated by the user which one can see directly in the final product, unlike the non-functional requirements.

Examples of functional requirements:

1. Authentication of user whenever he/she logs into the system
2. A verification email is sent to user whenever he/she register for the first time on some software system.

**Non-functional requirements**: These are basically the quality constraints that the system must satisfy according to the project contract. The priority or extent to which these factors are implemented varies from one project to other. They are also called non-behavioral requirements.  
They basically deal with issues like:

* Portability
* Security
* Maintainability
* Reliability
* Scalability
* Performance
* Reusability
* Flexibility

Examples of non-functional requirements:

1. Emails should be sent with a latency of no greater than 12 hours from such an activity.
2. The processing of each request should be done within 10 seconds
3. The site should load in 3 seconds whenever of simultaneous users are > 10000
   1. **Hardware Requirements**

# Processor - I7/Intel Processor

Hard Disk - 160GB

Key Board - Standard Windows Keyboard

Mouse - Two or Three Button Mouse

Monitor - SVGA

RAM - 8GB

* 1. **Software Requirements:**

Operating System : Windows 11

Server side Script : HTML, CSS, Bootstrap & JS

Programming Language : Python

Libraries : Flask, Pandas, Mysql.connector, Os, Smtplib, Numpy

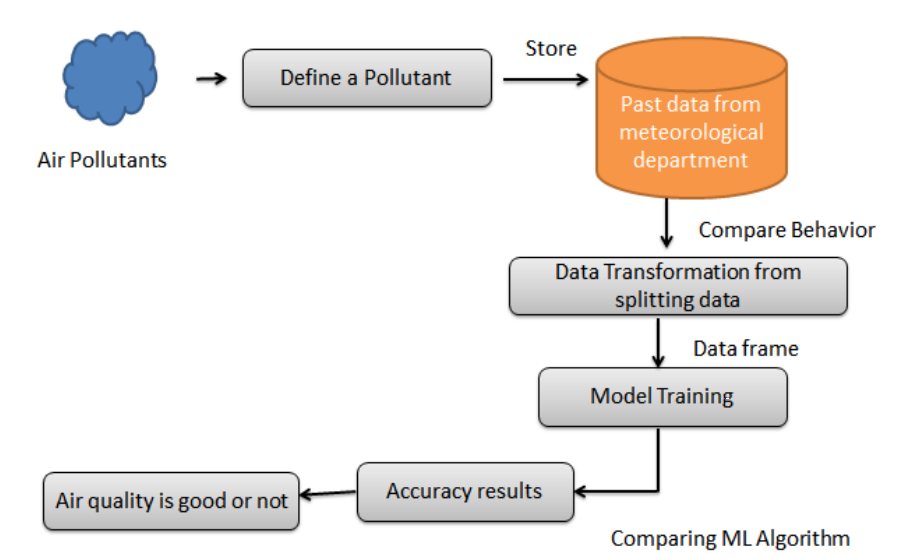
IDE/Workbench : PyCharm

Technology : Python 3.6+

Server Deployment : Xampp Server

Database : MySQL

* 1. **Architecture:**



**5. SYSTEM DESIGN**

**5.1 Introduction of Input Design:**

In an information system, input is the raw data that is processed to produce output. During the input design, the developers must consider the input devices such as PC, MICR, OMR, etc.

Therefore, the quality of system input determines the quality of system output. Well-designed input forms and screens have following properties −

* It should serve specific purpose effectively such as storing, recording, and retrieving the information.
* It ensures proper completion with accuracy.
* It should be easy to fill and straightforward.
* It should focus on user’s attention, consistency, and simplicity.
* All these objectives are obtained using the knowledge of basic design principles regarding −
  + What are the inputs needed for the system?
  + How end users respond to different elements of forms and screens.

### **Objectives for Input Design:**

The objectives of input design are −

* To design data entry and input procedures
* To reduce input volume
* To design source documents for data capture or devise other data capture methods
* To design input data records, data entry screens, user interface screens, etc.
* To use validation checks and develop effective input controls.

**Output Design:**

The design of output is the most important task of any system. During output design, developers identify the type of outputs needed, and consider the necessary output controls and prototype report layouts.

### Objectives of Output Design:

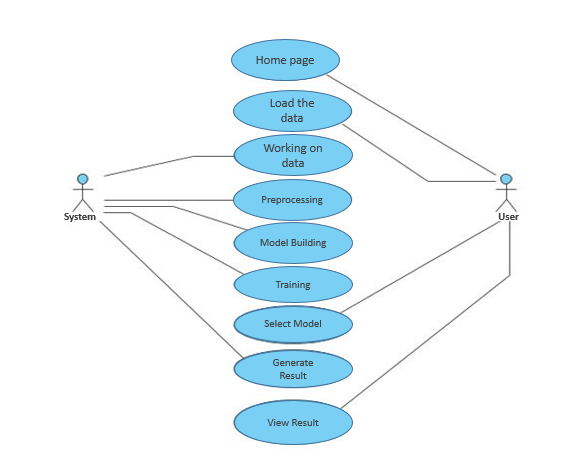
The objectives of input design are:

* To develop output design that serves the intended purpose and eliminates the production of unwanted output.
* To develop the output design that meets the end user’s requirements.
* To deliver the appropriate quantity of output.
* To form the output in appropriate format and direct it to the right person.
* To make the output available on time for making good decisions.

**5.2 UML Diagrams:**

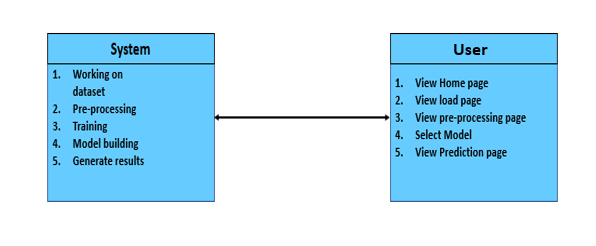
**5.2.1 Use Case Diagram:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



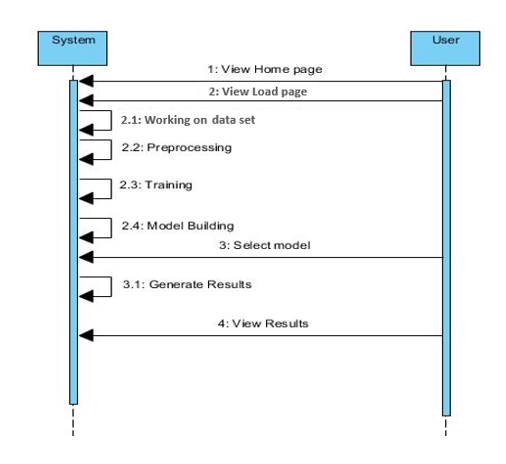
**5.2.2 Class Diagram:**

In software engineering, a class diagram in the Unified Modelling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



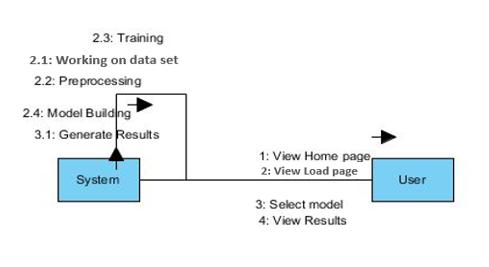
**5.2.3 Sequence Diagram:**

A sequence diagram in Unified Modelling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



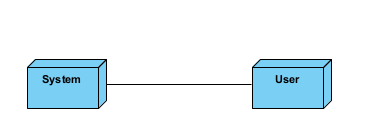
**5.2.4 Collaboration Diagram:**

In collaboration diagram the method call sequence is indicated by some numbering technique as shown below. The number indicates how the methods are called one after another. We have taken the same order management system to describe the collaboration diagram. The method calls are similar to that of a sequence diagram. But the difference is that the sequence diagram does not describe the object organization whereas the collaboration diagram shows the object organization.



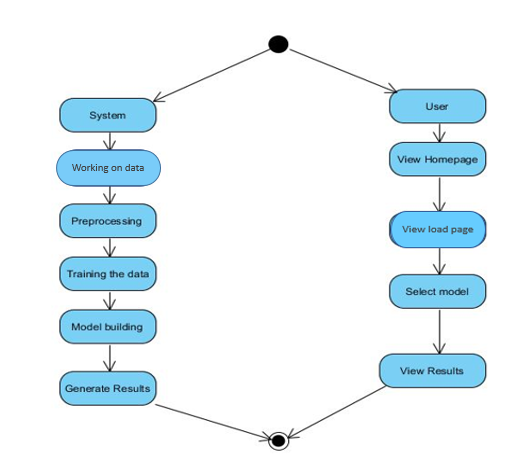
**5.2.5 Deployment Diagram**

Deployment diagram represents the deployment view of a system. It is related to the component diagram. Because the components are deployed using the deployment diagrams. A deployment diagram consists of nodes. Nodes are nothing but physical hardware’s used to deploy the application.



**5.2.6 Activity Diagram:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modelling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**5.2.7 Component Diagram**:

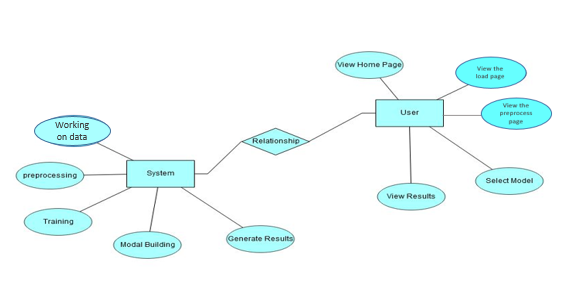
A component diagram, also known as a UML component diagram, describes the organization and wiring of the physical **c**omponents in a system. Component diagrams are often drawn to help model implementation details and double-check that every aspect of the system's required functions is covered by planned development.



**5.2.8 ER Diagram:**

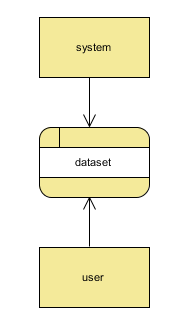
An Entity–relationship model (ER model) describes the structure of a database with the help of a diagram, which is known as Entity Relationship Diagram (ER Diagram). An ER model is a design or blueprint of a database that can later be implemented as a database. The main components of E-R model are: entity set and relationship set.

An ER diagram shows the relationship among entity sets. An entity set is a group of similar entities and these entities can have attributes. In terms of DBMS, an entity is a table or attribute of a table in database, so by showing relationship among tables and their attributes, ER diagram shows the complete logical structure of a database. Let’s have a look at a simple ER diagram to understand this concept.

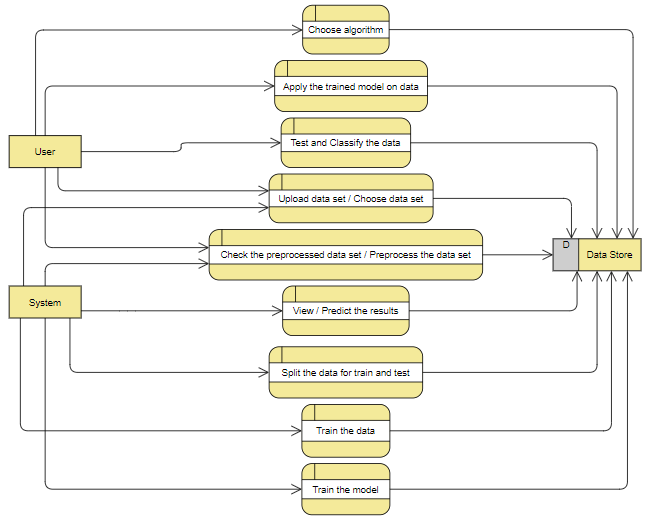


**5.3 DFD Diagram:**

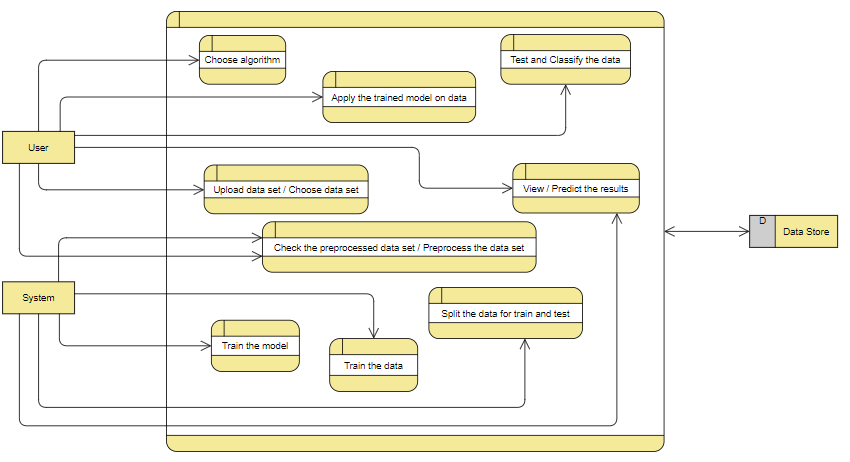
A Data Flow Diagram (DFD) is a traditional way to visualize the information flows within a system. A neat and clear DFD can depict a good amount of the system requirements graphically. It can be manual, automated, or a combination of both. It shows how information enters and leaves the system, what changes the information and where information is stored. The purpose of a DFD is to show the scope and boundaries of a system as a whole. It may be used as a communications tool between a systems analyst and any person who plays a part in the system that acts as the starting point for redesigning a system.

****

**Level 1 Diagram**:



**Level 2 Diagram**:



**IMPLEMENTATION AND RESULTS**

**6.1 Modules:**

1. **User**:
   1. **View Home page:**

Here user view the home page of the Air pollution web application.

* 1. **View about page:**

In the about page, users can learn more about the poverty classification.

* 1. **Input Model:**

The user must provide input values for the certain fields in order to get results.

* 1. **View Results:**

User view’s the generated results from the model.

* 1. **View score:**

Here user have ability to view the score in %

1. **System**
   1. **Working on dataset:**

System checks for data whether it is available or not and load the data in csv files.

* 1. **Pre-processing:**

Data need to be pre-processed according the models it helps to increase the accuracy of the model and better information about the data.

* 1. **Training the data:**

After pre-processing the data will split into two parts as train and test data before training with the given algorithms.

* 1. **Model Building**

To create a model that predicts the personality with better accuracy, this module will help user.

* 1. **Generated Score:**

Here user view the score in %

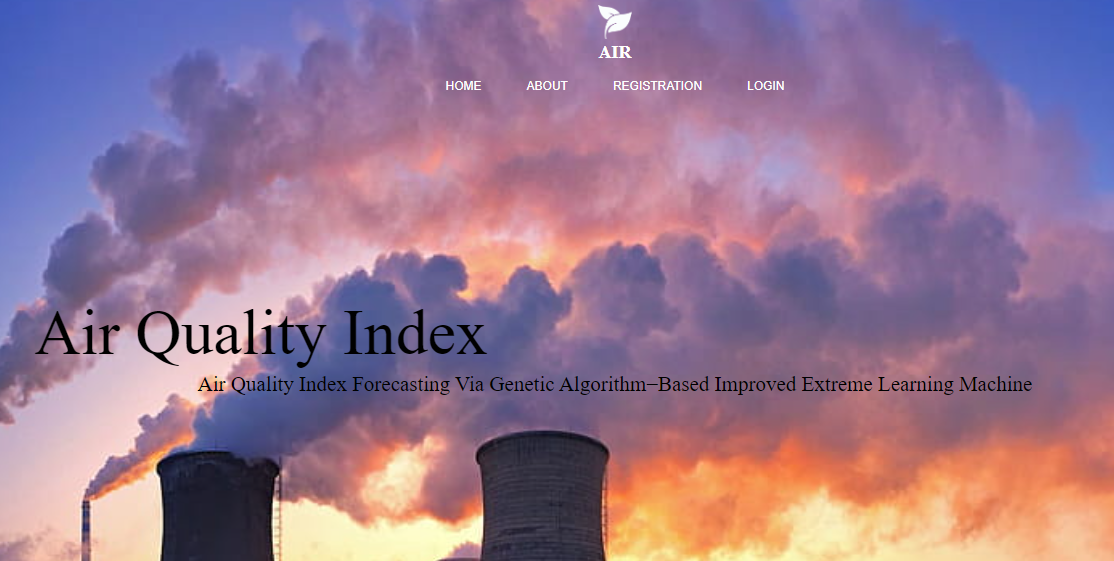
* 1. **Generate Results:**

We train the machine learning algorithm and predict the Air Pollution.

**6.2 Results:**

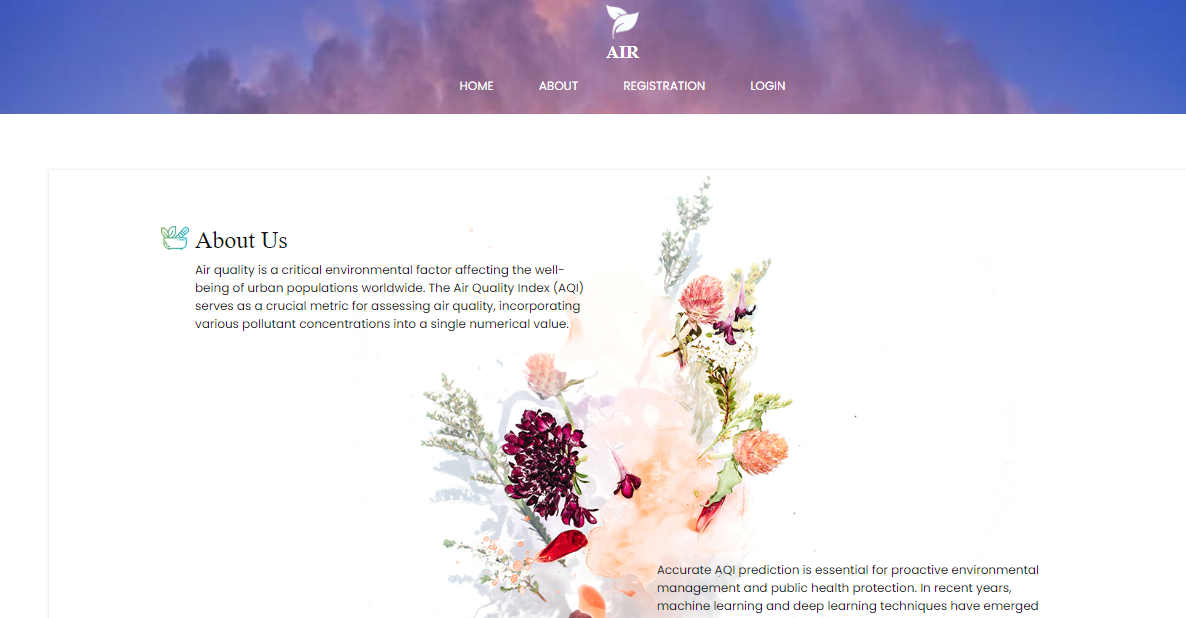
**Home Page:**

Here user view the home page of Air pollution web application..

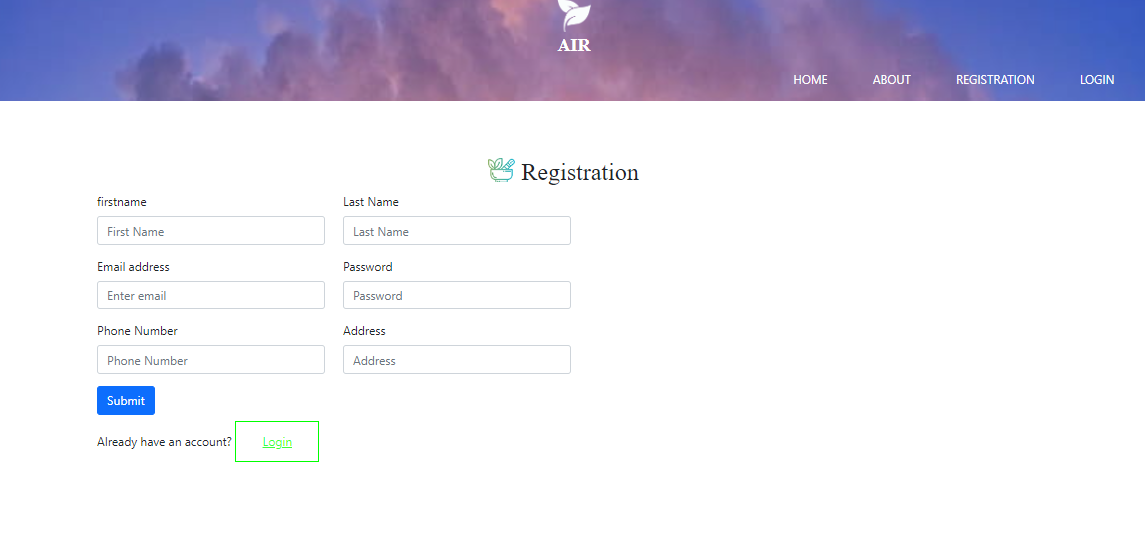


**About:**

Here user view the About page of Air Quality Index.

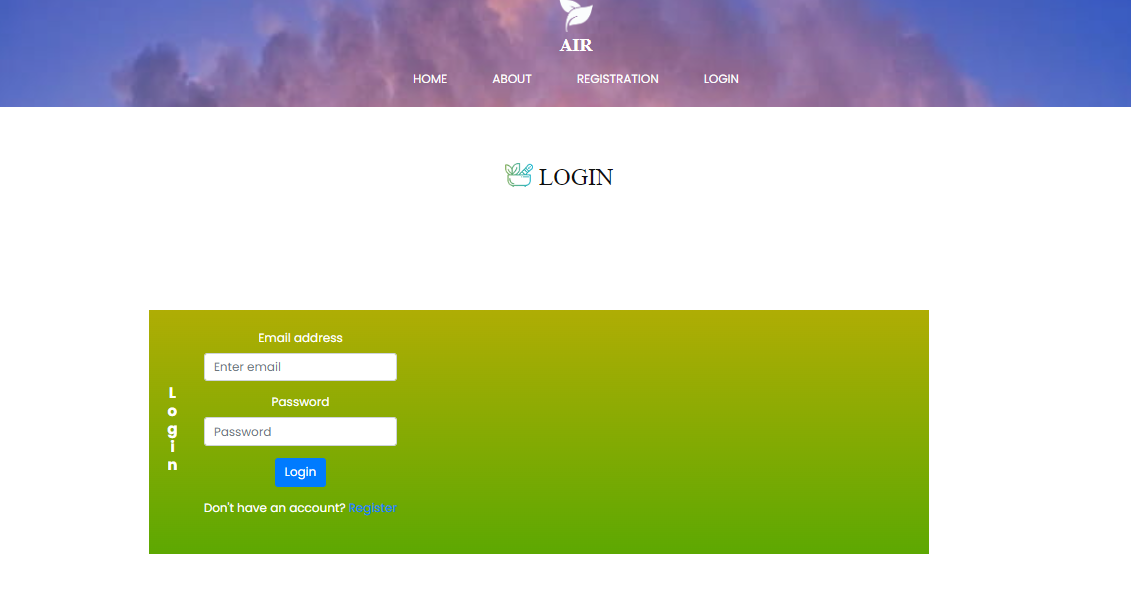


**Registration:**



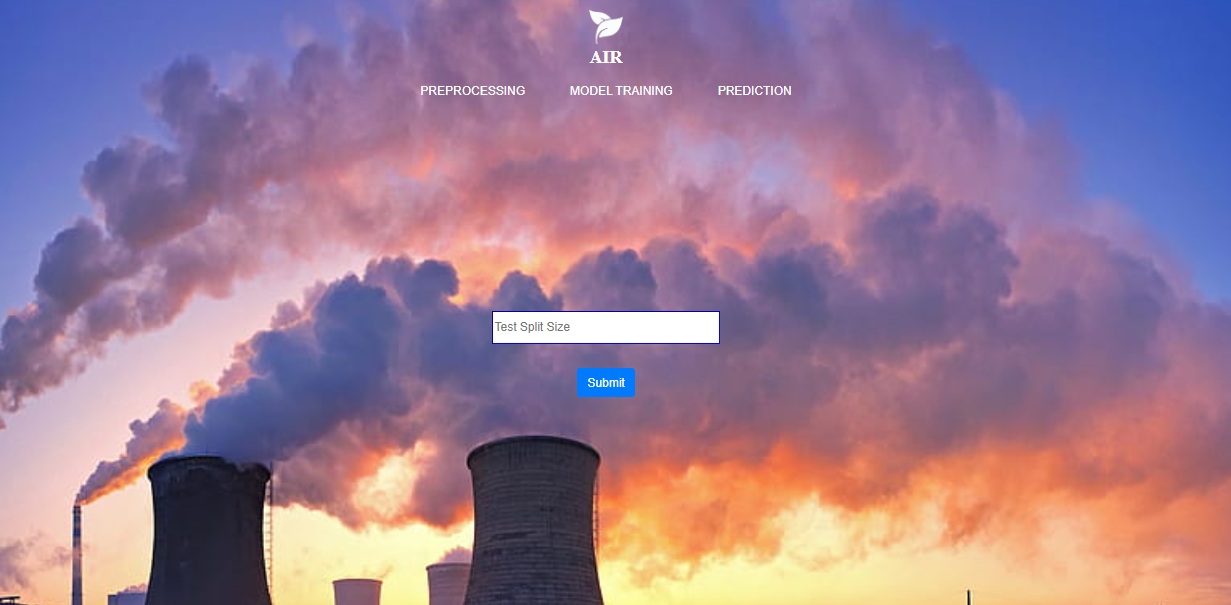
**Login:**

Here user login by provide their information.



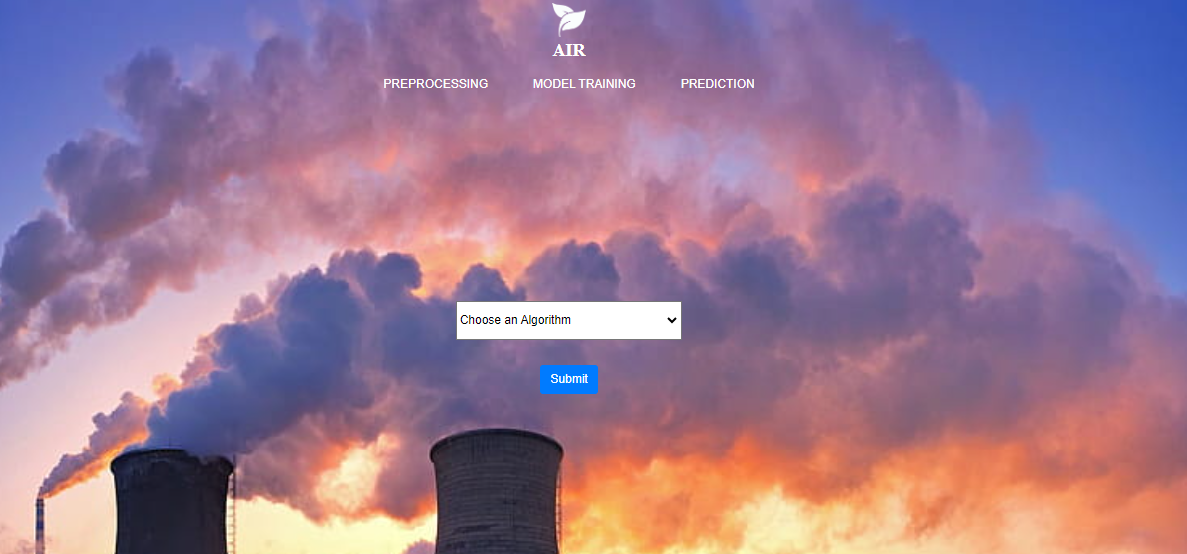
**Pre-process:**

Here we can pre-process and split our data into train and test.

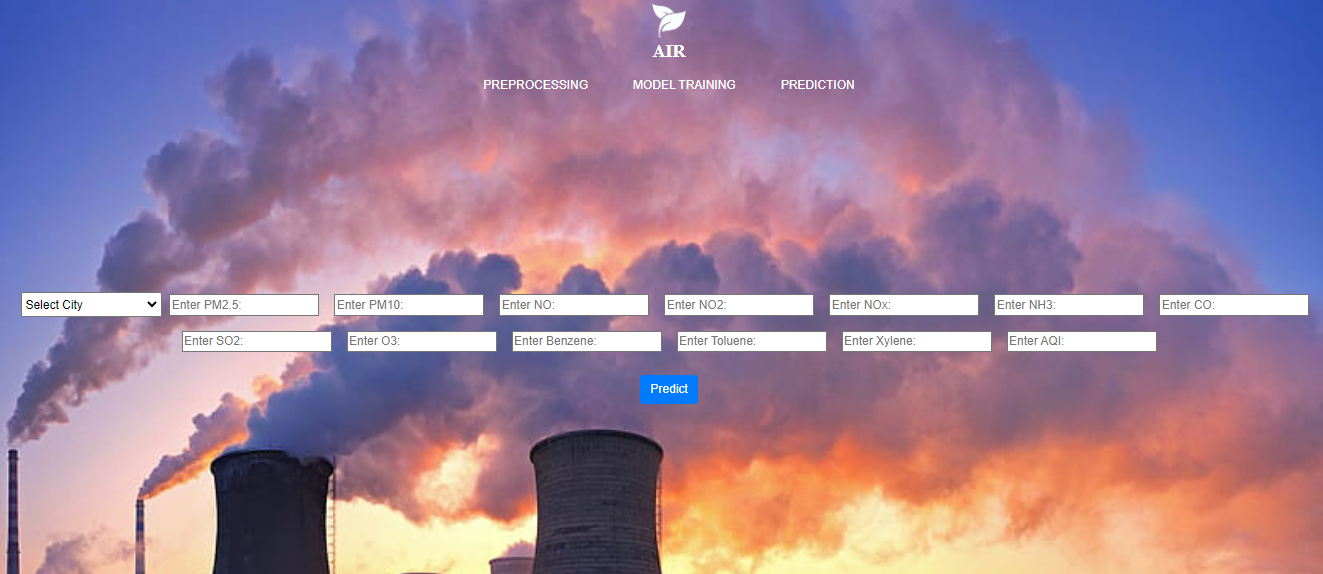
` 

**Model:**

Here we train our data with different ML algorithms.



**Prediction:**



**7. SYSTEM STUDY AND TESTING**

**7.1 Feasibility Study**

The feasibility of the project is analysed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are

* Economical feasibility
* Technical feasibility
* Social feasibility

**Economical Feasibility**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system as well within the budget and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### **Technical Feasibility**

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

**Social Feasibility**

The aspect of study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system, instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

**System Testing**

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product It is the process of exercising software with the intent of ensuring that the

Software system meets its requirements and user expectations and does not fail in an unacceptable manner. There are various types of test. Each test type addresses a specific testing requirement.

**7.2 Types of Tests**

**7.2.1 Unit testing**

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

**7.2.2 Integration testing**

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**Acceptance Testing**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

**Test Results:** All the test cases mentioned above passed successfully. No defects encountered.

**7.2.3 Functional testing**

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised.

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

**7.2.4 White Box Testing**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is purpose. It is used to test areas that cannot be reached from a black box level.

**7.2.5 Black Box Testing**

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

**Test objectives**

* All field entries must work properly.
* Pages must be activated from the identified link.
* The entry screen, messages and responses must not be delayed.

**Features to be tested**

* Verify that the entries are of the correct format
* No duplicate entries should be allowed
* All links should take the user to the correct page.
* **TEST CASES:**

|  |  |  |
| --- | --- | --- |
| **Input** | **Output** | **Result** |
| Input Data | A smart air pollution detector using machine learning | Success |

**Test cases Model building:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **S.NO** | **Test cases** | **I/O** | **Expected O/T** | **Actual O/T** | **P/F** |
| 1 | Read the datasets. | Dataset’s path. | Datasets need to read successfully. | Datasets fetched successfully. | It produced P. If this not F will come |
| 2 | Verifying the A smart air pollution using machine learning | Input for A smart air pollution using machine learning | Output as either in the form Air pollution | Output is accuracy as Air pollution | It produced P. If this is not, it will undergo F |
| 3 | Verifying the A smart air pollution using machine learning | Input for A smart air pollution using machine learning | Output as either in the form Air pollution | Output is accuracy as Air pollution | It produced P. If this is not, it will undergo F |
| 4 | Verifying the | Input the result | Need to predict the best accuracy | Model successfully predicted accuracy | It produced P. If this is not, it will undergo F |

**8.CONCLUSION:**

In conclusion, the integration of Genetic Algorithm-based Improved Extreme Learning Machine (IELM) proves highly effective in forecasting Air Quality Index (AQI), outperforming conventional methods like Random Forest, Decision Tree, Adaboost, and KNN. Leveraging diverse pollutant and meteorological data, including PM2.5, PM10, NO, NO2, NOx, NH3, CO, SO2, O3, Benzene, Toluene, and Xylene, our model demonstrates superior predictive accuracy. This research significantly advances air quality monitoring and management systems, providing valuable insights for policymakers and stakeholders to implement proactive measures against air pollution, thereby safeguarding public health and promoting environmental sustainability.

**9. FUTURE ENHANCEMENT**

To further enhance the proposed methodology, future research could explore the integration of real-time data streams from IoT sensors and satellite observations. Incorporating additional environmental parameters, such as wind speed, humidity, and geographical features, could refine predictive accuracy. Moreover, the utilization of advanced machine learning techniques like deep learning networks and ensemble models could offer deeper insights into complex AQI dynamics. Implementing a user-friendly interface for stakeholders to visualize forecasted AQI levels and associated uncertainties in real-time would enhance decision-making processes. Additionally, investigating the scalability of the model for application in diverse geographical regions and expanding the dataset for long-term forecasting could strengthen its utility in comprehensive air quality management systems.

**10. REFERENCES:**

[1] Pandey, Gaurav, Bin Zhang, and Le Jian. " Predicting sub-micron air pollution indicators: a machine learning approach." ; Environmental Science: Processes & amp; Impacts 15.5 (2013): 996-1005.

[2] Dan wei: Predicting air pollution level in a specific city [2014]

[3] Dixian Zhu, Changjie Cai, Tianbao Yang and Xun Zhou: A Machine Learning Approach for Air Quality Prediction: Model Regularization and Optimization. Big data and cognitive computing [2018].

[4] José Juan Carbajal-Hernándezab Luis P.Sánchez-Fernándeza Jesús A.Carrasco-OchoabJosé Fco.Martínez-Trinidadb: Assessment and prediction of air quality using fuzzy logic and autoregressive models: Center of Computer Research – National Polytechnic Institute, Av. Juan de Dios Bátiz S/N, Gustavo A. Madero, Col. Nueva. Industrial Vallejo, 07738 México D.F., Mexico1. (2012) Doi :https://doi.org/10.1016/j.atmosenv.2012.06.004

[5] Sachit Mahajan, Ling-Jyh Chen, Tzu-Chieh Tsai : An Empirical Study of PM2.5 Forecasting Using neural network. IEEE Smart World Congress, At San Francisco, USA [2017]

[6] Athanasiadis, Ioannis N., et al. "Applying machine learning techniques on air quality data for real-time decision support." First international NAISO symposium on information technologies in environmental engineering (ITEE'2003), Gdansk, Poland. 2003.

[7] Ioannis N. Athanasiadis, Kostas D. Karatzas and Pericles A. Mitkas. "Classification techniques for air quality forecasting." Fifth ECAI Workshop on Binding Environmental Sciences and Artificial Intelligence, 17th European Conference on Artificial Intelligence, Riva del Garda, Italy, August 2006.

[8] M. Caselli & L. Trizio & G. de Gennaro & P. Ielpo. "A Simple Feedforward Neural Network for the PM10 Forecasting: Comparison with a Radial Basis Function Network and a Multivariate Linear Regression Model." Water Air Soil Pollut (2009) 201:365–377