**Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine**

**INTRODUCTION**

Air pollution is a prevalent environmental problem in the twenty-first century. In light of the rapid industrialization and urbanization, air pollution is getting worse, which greatly affects our living environment and health. Li et al. concluded that outdoor physical activity poses numerous health risks due to ambient air pollution in China. According to the Chinese Ambient Air Quality Standards (GB3095- 2012), there are six conventional air pollutants used to measure air quality: sulfur dioxide (SO2), nitrogen dioxide (NO2), particulate matter with a particle size less than 10 microns (PM10), particulate matter with a particle size less than 2.5 microns (PM2.5), ozone (O3), and carbon monoxide (CO). These pollutants have adverse effects on human health. The International Energy Agency estimates that air pollution causes 6.5 million premature deaths per year, while long-term exposure to pollutants, such as fine particles (e.g., PM2.5) or traffic-related pollutants, is linked to higher rates of lung cancer, coronary heart disease, and other illnesses. Therefore, studies on air quality prediction are particularly important and are considered a key factor for environmental protection. In order to more comprehensively assess the health effects of air pollution, numerous air quality monitoring stations have been set up in major cities. Air quality predictions can be made based on the data collected from these stations. Air VOLUME 4, 2016 1 This article has been accepted for publication in IEEE Access. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2023.3291146 This work is licensed under a Creative Commons Attribution-Noncommercial-No Derivatives 4.0 License. For more information, see https://creativecommons.org/licenses/by-nc-nd/4.0/ Author et al.: Preparation of Papers for IEEE TRANSACTIONS and JOURNALS quality monitoring, modeling, and accurate predictions are important for having a clear understanding of future pollution levels and their associated health risks. Recently, the inherent property of machine learning algorithms to automatically learn features at multiple levels of abstraction has become increasingly important in providing solutions to this challenging task. However, the model only forecasts PM10 and SO2 levels, and it is also challenging to obtain measurement values needed to construct the dataset. Wu Q. et al. proposed an optimal-hybrid model for daily AQI prediction considering air pollutant factors, with the model’s inputs being the six atmospheric pollutants. However, neural networks typically struggle with slow learning, a tendency to fall into local minima, and a complex network training process. Based on the generalized inverse matrix theory, Huang et al. proposed an extreme learning machine (ELM) algorithm with a feedforward neural network that includes a single hidden layer, such that the problems of conventional neural network algorithms are circumvented. The ELM algorithm used to predict the AQI outperformed neural networks in terms of parameter selection, training speed, and prediction accuracy. However, the parameters of the hidden layer nodes and the number of nodes in the test hidden layer are selected at random, which puts the prediction accuracy to a great test. In order to solve the aforementioned problems, we propose to optimize the number of ELM hidden layer nodes, thresholds, and weights, along with an improved genetic algorithm (GA) that uses root mean square error (RMSE) as the fitness function, to obtain the optimal network structure for air quality prediction. The number of hidden layer nodes is updated by continuous coding discretization, the input weights and hidden layer thresholds are updated by continuous coding, and the update thresholds and weights are selected with the number of updated layers to form a hierarchical control structure. The proposed GA-based improved extreme learning machine (GA-KELM) algorithm is applied to air quality prediction, and its performance is compared with that of community multiscale air quality modeling system (CMAQ), support vector regression (SVR), and deep belief network-back propagation (DBN-BP). The results show that the accuracy of the proposed GA-KELM algorithm is reliable for air quality prediction. In this study, an improved extreme learning machine model based on a genetic algorithm is designed and applied to AQI prediction. To verify the effectiveness of the model, we conducted tests on three real-world datasets. The results confirmed that the proposed method has superior performance and outperforms some advanced methods currently in use. The main contributions of this paper are: modifying the ELM activation function or using the kernel function to improve the prediction accuracy, optimizing the ELM using GA to improve the stability of the results and further enhance the prediction accuracy, and obtaining the correlation analysis results of atmospheric environmental quality prediction parameters by comprehensively considering each relevant factor in line with the actual situation. The remainder of this paper is organized as follows. Section 2 presents related work. Section 3 describes ELM and the proposed GA-KELM, and illustrates the improvements using the model. Section 4 discusses experimental results where GA-KELM is compared with several other methods in terms of prediction results. The last section concludes the entire work and presents directions for future research.

**1.1 Objective of the project:**

Air quality has always been one of the most important environmental concerns for the general public and society. Using machine learning algorithms for Air Quality Index (AQI) prediction is helpful for the analysis of future air quality trends from a macro perspective. When conventionally using a single machine learning model to predict air quality, it is challenging to achieve a good prediction outcome under various AQI fluctuation trends. In order to effectively address this problem, a genetic algorithm-based improved extreme learning machine (GA-KELM) prediction method is enhanced. First, a kernel method is introduced to produce the kernel matrix which replaces the output matrix of the hidden layer. To address the issue of the conventional limit learning machine where the number of hidden nodes and the random generation of thresholds and weights lead to the degradation of the network learning ability, a genetic algorithm is then used to optimize the number of hidden nodes and layers of the kernel limit learning machine. The thresholds, the weights, and the root mean square error are used to define the fitness function. Finally, the least squares method is applied to compute the output weights of the model. Genetic algorithms are able to find the optimal solution in the search space and gradually improve the performance of the model through an iterative optimization process. In order to verify the predictive ability of GA-KELM, based on the collected basic data of long-term air quality forecast at a monitoring point in a city in China, the optimized kernel extreme learning machine is applied to predict air quality ( $SO\_{2}$ , $NO\_{2}$ , $PM\_{10}$ , $CO$ , $O\_{3}$ , $PM\_{2.5}$ concentration and AQI), with comparative experiments based CMAQ (Community Multiscale Air Quality), SVM (Support Vector Machines) and DBN-BP (Deep Belief Networks with Back-Propagation). The results show that the proposed model trains faster and makes more accurate predictions.

**2. LITERATURE SURVEY:**

**A Bayesian LSTM model to evaluate the effects of air pollution control regulations in China**

Rapid socio-economic development and urbanization have resulted in serious deterioration in air-quality in many world cities, including Beijing, China. This study attempts to examine the effectiveness of air pollution control regulations implemented in Beijing during 2008–2019 through a data-driven regulatory intervention analysis. Our proposed Bayesian deep learning model utilizes proxy data including Aerosol Optical Depth (AOD) and meteorology as well as socio-economic data, while accounting for confounding effects via propensity score estimation. Our results show that air pollution control regulatory measures implemented in China and Beijing during 2008–2019 reduced PM2.5 pollution in Beijing by 11 % on average. After the introduction of Action Plan for Clean Air in China and Beijing in late 2013, as compared to the hypothetical PM2.5 concentration (without any regulatory interventions), the estimated PM2.5 reduction increased dramatically from 15 % in 2015 to 44 % in 2018. Our results suggest that Beijing’s air quality has improved gradually over the past decade, though the annual PM2.5 pollution still exceeds the WHO threshold. In this regard, the air pollution control regulations introduced in Beijing and China tend to become more effective after 2015, suggesting a 2-year time lag before the stringent air pollution control regulations starting from 2013 takes any strong positive effects. Moreover, as compared to the air pollution control regulations introduced before 2013, newly introduced policy-making governance, which couples the policy-makings of the local jurisdictions with that of the central government, and the new policy measures that tackle the vested interests of the local stakeholders in Beijing and its nearby cities, alongside with the stringent local and national air pollution control regulations and plans, should help reduce air pollution and promote healthy living in Beijing over the longer term.

**Air pollution forecasts: An overview**

Air pollution is defined as a phenomenon harmful to the ecological system and the normal conditions of human existence and development when some substances in the atmosphere exceed a certain concentration. In the face of increasingly serious environmental pollution problems, scholars have conducted a significant quantity of related research, and in those studies, the forecasting of air pollution has been of paramount importance. As a precaution, the air pollution forecast is the basis for taking effective pollution control measures, and accurate forecasting of air pollution has become an important task. Extensive research indicates that the methods of air pollution forecasting can be broadly divided into three classical categories: statistical forecasting methods, artificial intelligence methods, and numerical forecasting methods. More recently, some hybrid models have been proposed, which can improve the forecast accuracy. To provide a clear perspective on air pollution forecasting, this study reviews the theory and application of those forecasting models. In addition, based on a comparison of different forecasting methods, the advantages and disadvantages of some methods of forecasting are also provided. This study aims to provide an overview of air pollution forecasting methods for easy access and reference by researchers, which will be helpful in further studies.

**A Deep Learning Approach to Writer Identification Using Inertial Sensor Data of Air-Handwriting**

To the best of our knowledge, there are a few researches on air-handwriting character-level writer identification only employing acceleration and angular velocity data. In this paper, we propose a deep learning approach to writer identification only using inertial sensor data of air-handwriting. In particular, we separate different representations of degree of freedom (Doff) of air-handwriting to extract local dependency and interrelationship in different CNNs separately. Experiments on a public dataset achieve an average good performance without any extra hand-designed feature extractions.

**Application of computational intelligence techniques to forecast daily PM10 exceedances in Brunei Darussalam**

[Particulate matter](https://www.sciencedirect.com/topics/earth-and-planetary-sciences/atmospheric-aerosol) (PM10) is the pollutant causing exceedances of ambient air quality thresholds, and the key indicator of air quality index in Brunei Darussalam for haze related episodes caused by the recurrent biomass fires in Southeast Asia. The present study aims at providing suitable forecasts for PM10 exceedances to aid in health advisory during haze episodes at the four administrative districts of the country. A framework based on random forests (RFs), genetic algorithm (GA) and [back propagation](https://www.sciencedirect.com/topics/earth-and-planetary-sciences/back-propagation) neural networks (BPNN) computational intelligence techniques has been proposed in which the final prediction is made by the BPNN model. A hybrid combination of GA and RFs is initially applied to determine optimal set of inputs from the initial data sets of largely available meteorological, persistency of high pollution levels, short- and long-term variations of emissions rates parameters. The inputs selection procedure does not depend on the [back propagation](https://www.sciencedirect.com/topics/earth-and-planetary-sciences/back-propagation) training algorithm. The numerical results presented in this paper show that the proposed model not only produced satisfactory forecasts but also consistently performed better via several statistical performance indicators when compared with the standard BPNN and GA optimization based on back propagation training algorithm. The model also showed satisfactory threshold exceedances forecasts achieving for instance best true predicted rate of 0.800, false positive rate of 0.014, false alarm rate of 0.333 and success index of 0.786 at Brunei-Maura district monitoring station. Overall, the current study has profound implications on future studies to develop a real-time air quality forecasting system to support haze management.

**Optimization on fresh outdoor air ratio of air conditioning system with stratum venti-lotion for both targeted indoor air quality and maximal energy saving**

Stratum ventilation can energy efficiently provide good inhaled indoor air quality with a proper operation (e.g., fresh outdoor air ratio). However, the non-uniform CO2 distribution in a stratum-ventilated room challenges the provision of targeted indoor air quality. This study proposes an optimization on the fresh outdoor air ratio of stratum ventilation for both the targeted indoor air quality and maximal energy saving. A model of CO2 concentration in the breathing zone is developed by coupling CO2 removal efficiency in the breathing zone and [mass conservation laws](https://www.sciencedirect.com/topics/engineering/mass-conservation-law). With the developed model, the ventilation parameters corresponding to different fresh outdoor air ratios are quantified to achieve the targeted indoor air quality (i.e., targeted CO2 concentration in the breathing zone). Using the fresh outdoor air ratios and corresponding ventilation parameters as inputs, energy performance evaluations of the air conditioning system are conducted by building energy simulations. The fresh outdoor air ratio with the minimal energy consumption is determined as the optimal one. Experiments show that the [mean absolute error](https://www.sciencedirect.com/topics/engineering/mean-absolute-error) of the developed model of CO2 concentration in the breathing zone is 1.9%. The effectiveness of the proposed optimization is demonstrated using TRNSYS that the energy consumption of the air conditioning system with stratum ventilation is reduced by 6.4% while achieving the targeted indoor air quality. The proposed optimization is also promising for other ventilation modes for targeted indoor air quality and improved energy efficiency.

**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 Bi-directional 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 accuracy.

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

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.

**Disadvantages**

1.Less accuracy.

**PROPOSED SYSTEM**

In this proposed system, to update weights accurately author of this paper enhancing Extreme Learning Machine with Genetic Algorithm (GA-KLEM). To solve the issue of the conventional limit learning machine where the number of hidden nodes and the random generation of thresholds and weights lead to the degradation of the network learning ability, a genetic algorithm is then used to optimize the number of hidden nodes and layers of the kernel limit learning machine. The thresholds, the weights, and the root mean square error are used to define the fitness function. Finally, the least squares method is applied to compute the output weights of the model. Genetic algorithms are able to find the optimal solution in the search space and gradually improve the performance of the model through an iterative optimization process.

**Advantages**

1. High Accuracy

**Modules Information:**

To implement this project we have used same dataset given in your requirement file and to implement this project we have designed following modules

1. defining interpolate function: using this module to deal with missing values and outliers.
2. training SVM algorithm: using this module we trained svm. In graph, Red line represents Original Test Air Quality and Predicted Air Quality
3. training propose genetic ELM: using this module we trained genetic elm called GA-KELM
4. training extension BI-LSTM algorithm: using this module, we trained bi-lstm algorithm

**FUNCTIONAL REQUIREMENTS:**

**SOFTWARE REQIREMENTS:**

**System Atributes:**

1. Filename
2. dataset
3. X, Y, mse, X\_train, X\_test, y\_train, y\_test

**Prototype:**

python 3.7.0 or 3.7.4

opencv-python==4.5.1.48

keras==2.3.1

tensorflow==1.14.0

protobuf==3.16.0

h5py==2.10.0

sklearn-extensions==0.0.2

scikit-learn==0.22.2.post1

Numpy

Pandas

**NON-FUNCTIONAL REQUIREMENT:**

**Usability:**  Usability is a quality attribute that assesses how easy user interfaces are to use. The word "usability" also refers to methods for improving ease-of-use during the design process.(how it was handle entire project easy)

**Security:** the quality or state of being secure: such as. a : freedom from danger : safety. b : freedom from fear or anxiety. c : freedom from the prospect of being laid off job security.

**Readability:** Readability is the ease with which a reader can understand a written text.

**Performance**: the execution of an action. : something accomplished : deed, feat. : the fulfillment of a claim, promise, or request : implementation. 3. : the action of representing a character in a play.

**Availability**: the quality or state of being available trying to improve the availability of affordable housing. 2 : an available person or thing.

**Scalability**: Scalability is the measure of a system's ability to increase or decrease in performance and cost in response to changes in application and system processing demands.

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

Feasibility Study

TEAM FORMATION

Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artifacts will be produced. Software artifacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artifacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artifact is linked to a specific design element, and that each developed artifact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artifacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artifacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Nonfunctional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_(business)) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web-based user interface for audit workflow at NIC-CSD. Thus, it provides an easy access to. the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**SYSTEM REQUIREMENT:**

**HARDWARE REQUIREMENTS:**

# Processor - Intel i3(min)

* Speed - 1.1 GHz
* RAM - 4GB(min)
* Hard Disk - 500 GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows10(min)
* Programming Language - Python with Jupiter notebook

**4. SYSTEM DESIGN**

**CLASS DIAGRAM:**

The class diagram is the main building block of object-oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake

**USECASE DIAGRAM:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as we





**SEQUENCE DIAGRAM**

A **sequence diagram** 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. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams**, **event scenarios**, and timing diagrams.





**COLLABORATION DIAGRAM:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.





**COMPONENT DIAGRAM:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.





**DEPLOYMENT DIAGRAM:**

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

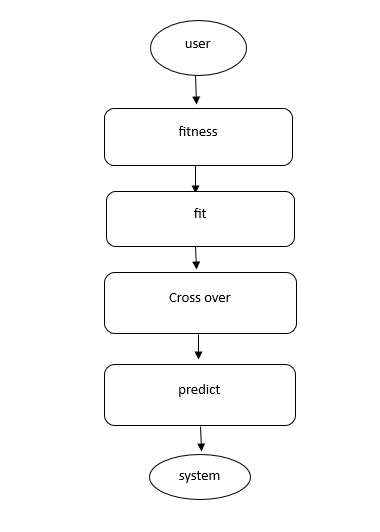
The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.

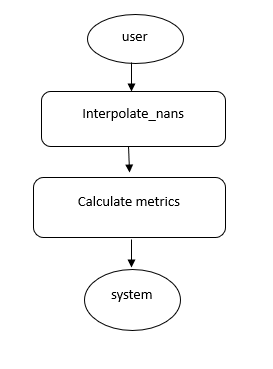




**ACTIVITY DIAGRAM:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

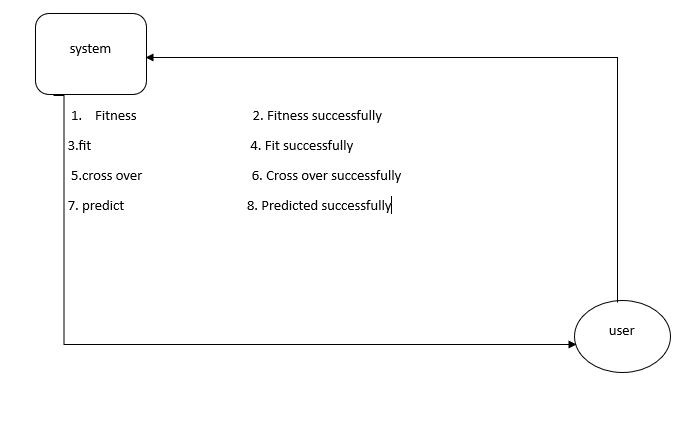


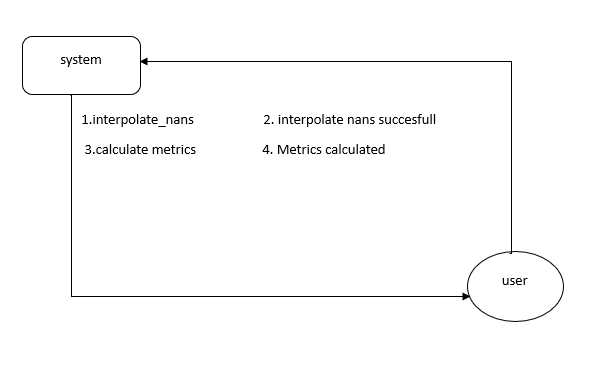


**Data flow:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.





**5. IMPLEMETATION**

**5.1 python**

One of the most popular languages is Python. Guido van Rossum released this language in 1991. Python is available on the Mac, Windows, and Raspberry Pi operating systems. The syntax of Python is simple and identical to that of English. When compared to Python, it was seen that the other language requires a few extra lines.

\*It is an interpreter-based language because code may be run line by line after it has been written. This implies that rapid prototyping is possible across all platforms. Python is a big language with a free, binary-distributed interpreter standard library.

\* It is inferior to maintenance that is conducted and is straightforward to learn. It is an object-oriented, interpreted programming language. It supports several different programming paradigms in addition to object-oriented programming, including functional and procedural programming.

\* It supports several different programming paradigms in addition to object-oriented programming, including practical and procedural programming. Python is mighty while maintaining a relatively straightforward syntax. Classes, highly dynamic data types, modules, and exceptions are covered. Python can also be utilised by programmes that require programmable interfaces as an external language.

Here are some key features and characteristics of Python:

* Readability: Python emphasizes code readability with its clean and intuitive syntax. It uses indentation and whitespace to structure code blocks, making it easy to understand and maintain.
* Easy to Learn: Python's simplicity and readability make it an excellent choice for beginners. Its straightforward syntax and extensive documentation make it accessible for newcomers to programming.
* Interpreted Language: Python is an interpreted language, meaning that it doesn't need to be compiled before running. The Python interpreter reads and executes the code directly, making the development process faster and more interactive.
* Cross-platform Compatibility: Python is available for major operating systems like Windows, macOS, and Linux. This cross-platform compatibility allows developers to write code once and run it on different platforms without modifications.
* Large Standard Library: Python comes with a vast standard library that provides ready-to-use modules and functions for various tasks. It covers areas such as file I/O, networking, regular expressions, databases, and more, saving developers time and effort.
* Extensible and Modular: Python supports modular programming, enabling developers to organize code into reusable modules and packages. Additionally, Python allows integrating modules written in other languages, such as C or C++, providing flexibility and performance optimizations.
* Wide Range of Libraries and Frameworks: Python has a vibrant ecosystem with numerous third-party libraries and frameworks. These libraries, such as NumPy, pandas, TensorFlow, and Django, extend Python's capabilities for specific domains, making it a powerful tool for diverse applications.
* Object-Oriented: Python supports object-oriented programming (OOP) principles, allowing developers to create and work with classes and objects. OOP provides a structured approach to code organization, promoting code reuse and modularity.
* Dynamic Typing: Python is dynamically typed, meaning variable types are determined at runtime. Developers do not need to declare variable types explicitly, which enhances flexibility and simplifies code writing.

**5.2 Installation**

To install Python on your computer, follow these basic steps:

* Step 1: Visit the Python website Go to the official Python website at <https://www.python.org/>.
* Step 2: Select the operating system Choose the appropriate installer for your operating system. Python supports Windows, macOS, and various Linux distributions. Make sure to select the correct version that matches your operating system.
* Step 3: Check which version of Python is installed; if the 3.7.0 version is not there, uninstall it through the control panel and
* Step 4: Install Python 3.7.0 using Cmd.
* Step 5: Install the all libraries that required to run the project
* Step 6: Run

**5.3 Python Features:**

1. **Easy:** Because Python is a more accessible and straightforward language, Python programming is easier to learn.
2. **Interpreted language:** Python is an interpreted language, therefore it can be used to examine the code line by line and provide results.
3. **Open Source:** Python is a free online programming language since it is open-source.
4. **Portable:** Python is portable because the same code may be used on several computer standard
5. **libraries:** Python offers a sizable library that we may utilize to create applications quickly.
6. **GUI:** It stands for GUI (Graphical User Interface)
7. **Dynamical typed:** Python is a dynamically typed language, therefore the type of the value will be determined at runtime.

**5.4 Python GUI (Tkinter)**

* Python provides a wide range of options for GUI development (Graphical User Interfaces).
* Tkinter, the most widely used GUI technique, is used for all of them.
* The Tk GUI toolkit offered by Python is used with the conventional Python interface.
* Tkinter is the easiest and quickest way to write Python GUI programs.
* Using Tkinter, creating a GUI is simple.
* A part of Python's built-in library is Tkinter. The GUI programs were created.
* Python and Tkinter together give a straightforward and quick way. The Tk GUI toolkit's object-oriented user interface is called Tkinter.

Making a GUI application is easy using Tkinter. Following are the steps:

1) Install the Tkinter module in place.

2) The GUI applicatioMakeske the primary window

3) Include one or more of the widgets mentioned above in the GUI application.

4) Set up the main event loop such that it reacts to each user-initiated event.

Although Tkinter is the only GUI framework included in the Python standard library, Python includes a GUI framework. The default library for Python is called Tkinter. Tk is a scripting language often used in designing, testing, and developing GUIs. Tk is a free, open-source widget toolkit that may be used to build GUI applications in a wide range of computer languages.

**5.5 Python IDLE**

* Python IDLE offers a full-fledged file editor, which gives you the ability to write and execute Python programs from within this program. The built-in file editor also includes several features, like code completion and automatic indentation, that will speed up your coding workflow.
* Guido Van Rossum named Python after the British comedy group Monty Python while the name IDLE was chosen to pay tribute to Eric Idle, who was one of the Monty Python's founding members. IDLE comes bundled with the default implementation of the Python language since the 01.5. 2b1 release
* IDLE is used to execute statements similar to Python Shell. IDLE is used to create, modify, and execute Python code. IDLE provides a fully-featured text editor to write Python scripts and provides features like syntax highlighting, auto-completion, and smart indent.
* IDLE has two modes: interactive and script. We wrote our first program, “Hello, World!” in interactive mode. Interactive mode immediately returns the results of commands you enter into the shell. In script mode, you will write a script and then run it.
* The IDE Python IDLE is a good place to start as it helps you become familiar with the way Python works and understand its syntax. This IDE is good to start programming in Python due to its great debugger, but once you are fluent and start developing projects it is necessary to jump to another, more complete IDE.
* Python IDLE (Integrated Development and Learning Environment) is an interactive development environment included with the Python programming language. It provides a convenient way to write, execute, and debug Python code.

When you install Python, IDLE is typically installed along with it. To open IDLE, you can follow these steps:

* Open the command prompt (Windows) or terminal (macOS/Linux).
* Type "idle" and press Enter. Alternatively, you can specify the version with "idle3" or "idle2" for Python 3 or Python 2, respectively.
* Once IDLE is launched, you will see the Python shell, which is an interactive environment where you can type and execute Python code directly.

Here are some features and functionalities provided by Python IDLE:

* Editor: IDLE includes a text editor where you can write your Python code. It offers syntax highlighting, automatic indentation, and code completion to enhance your coding experience.
* Interactive Shell: The Python shell in IDLE allows you to execute Python code interactively. You can type commands, statements, or function calls directly in the shell, and Python will execute them immediately.
* Debugging: IDLE provides basic debugging capabilities to help you find and fix errors in your code. You can set breakpoints, step through code, inspect variables, and track the program's execution.
* Python Help: IDLE provides access to the Python documentation and built-in help. You can access the help menu to find information about Python modules, functions, classes, and more.
* Script Execution: In addition to the interactive shell, IDLE allows you to run Python scripts stored in files. You can write your code in the editor and execute it as a script to see the output or interact with the program.
* Customization: IDLE can be customized to suit your preferences. You can modify settings related to syntax highlighting, indentation, fonts, and more.
* Python IDLE serves as a beginner-friendly development environment and learning tool. It is suitable for writing small scripts, testing code snippets, experimenting with Python features, and learning the language's basics. However, for more advanced development projects, you may consider using other code editors or integrated development environments (IDEs) that provide additional features and better project management capabilities.

**5.6 Libraries**

In Python, libraries (also referred to as modules or packages) are collections of pre-written code that provide additional functionality and tools to extend the capabilities of the Python language. Libraries contain reusable code that developers can leverage to perform specific tasks without having to write everything from scratch.

Python libraries are designed to solve common problems, such as handling data, performing mathematical operations, interacting with databases, working with files, implementing networking protocols, creating graphical user interfaces (GUIs), and much more. They provide ready-to-use functions, classes, and methods that simplify complex operations and save development time.

**Libraries in Python offer various advantages:**

* Code Reusability:
* Efficiency:
* Collaboration
* Domain-Specific Functionality
* To use a Python library, you need to install it first.

There are some libraries following:

* **Pandas:**

Pandas are a Python computer language library for data analysis and manipulation. It offers a specific operation and data format for handling time series and numerical tables. It differs significantly from the release3-clause of the BSD license. It is a well-liked open-source of opinion that is utilized in machine learning and data analysis.

Pandas are a Python package providing fast, flexible, and expressive data structures designed to make working with “relational” or “labeled” data both easy and intuitive. It aims to be the fundamental high-level building block for doing practical, real-world data analysis in Python. Pandas are a Python library used for working with data sets.

* It has functions for analysing, cleaning, exploring, and manipulating data.
* The name "Pandas" has a reference to both "Panel Data", and "Python Data Analysis" and was created by Wes McKinney in 2008.
* Pandas allow us to analyse big data and make conclusions based on statistical theories.
* Pandas can clean messy data sets, and make them readable and relevant.

Relevant data is very important in data science. Pandas are a Python library for data analysis. Started by Wes McKinney in 2008 out of a need for a powerful and flexible quantitative analysis tool, pandas have grown into one of the most popular Python libraries. It has an extremely active community of contributors. The name is derived from the term "panel data", an econometrics term for data sets that include observations over multiple time periods for the same individuals. Its name is a play on the phrase "Python data analysis" itself.

* **NumPy:**

The NumPy Python library for multi-dimensional, big-scale matrices adds a huge number of high-level mathematical functions. It is possible to modify NumPy by utilizing a Python library. Along with line, algebra, and the Fourier transform operations, it also contains several matrices-related functions.

NumPy can be used to perform a wide variety of mathematical operations on arrays. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.

* NumPy is a Python library used for working with arrays.
* It also has functions for working in domain of linear algebra, Fourier transform, and matrices.
* NumPy was created in 2005 by Travis Oliphant. It is an open source project and you can use it freely.
* NumPy stands for Numerical Python.
* In Python we have lists that serve the purpose of arrays, but they are slow to process.
* NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.
* The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.
* Arrays are very frequently used in data science, where speed and resources are very important.
* **Matplotlib:**

It is a multi-platform, array-based data visualization framework built to interact with the whole SciPy stack. MATLAB is proposed as an open-source alternative. Matplotlib is a Python extension and a cross-platform toolkit for graphical plotting and visualization.

Matplotlib is a popular Python library for creating static, animated, and interactive visualizations. It provides a flexible and comprehensive set of tools for generating plots, charts, histograms, scatter plots, and more. Matplotlib is widely used in various fields, including data analysis, scientific research, and data visualization.

Here are some key features and functionalities of the Matplotlib library:

* Plotting Functions
* Customization Options
* Multiple Interfaces
* Integration with NumPy and pandas
* Subplots and Figures:
* Saving and Exporting
* **Scikit-learn:**

The most stable and practical machine learning library for Python is scikit-learn. Regression, dimensionality reduction, classification, and clustering are just a few of the helpful tools it provides through the Python interface for statistical modeling and machine learning. It is an essential part of the Python machine learning toolbox used by JP Morgan. It is frequently used in various machine learning applications, including classification and predictive analysis.

Scikit-learn (also referred to as sklearn) is a widely used open-source machine learning library for Python. It provides a comprehensive set of tools and algorithms for various machine learning tasks, including classification, regression, clustering, dimensionality reduction, model selection, and pre-processing.

Here are some key features and functionalities of the Scikit-learn library:

* Easy-to-Use Interface:
* Broad Range of Algorithms:
* Data Pre-processing and Feature Engineering:
* Model Evaluation and Validation:
* Integration with NumPy and pandas:
* Robust Documentation and Community Support:
* **Keras:**

\* Google's Keras is a cutting-edge deep learning API for creating neural networks. It is created in Python and is designed to simplify the development of neural networks. Additionally, it enables the use of various neural networks for computation. Deep learning models are developed and tested using the free and open-source Python software known as Keras.

Keras is a high-level deep learning library for Python. It is designed to provide a user-friendly and intuitive interface for building and training deep learning models. Keras acts as a front-end API, allowing developers to define and configure neural networks while leveraging the computational backend engines, such as Tensor Flow or Theano.

Here are some key features and functionalities of the Keras library:

* User-Friendly API
* Multi-backend Support
* Wide Range of Neural Network Architectures
* Pre-trained Models and Transfer Learning:
* Easy Model Training and Evaluation:
* GPU Support:
* **h5py:**

\* The h5py Python module offers an interface for the binary HDF5 data format. Thanks to p5py, the top can quickly halt the vast amount of numerical data and alter it using the NumPy library. It employs common syntax for Python, NumPy, and dictionary arrays.

h5py is a Python library that provides a simple and efficient interface for working with datasets and files in the Hierarchical Data Format 5 (HDF5) format. HDF5 is a versatile data format commonly used for storing and managing large volumes of numerical data.

Here are some key features and functionalities of the h5py library:

* + HDF5 File Access
  + Dataset Handling:
  + Group Organization:
  + Attributes:
  + Compatibility with NumPy
  + Performance
* **Tensor flow**

TensorFlow is a Python library for fast numerical computing created and released by Google. It is a foundation library that can be used to create Deep Learning models directly or by using wrapper libraries that simplify the process built on top of TensorFlow. TensorFlow is an end-to-end open source platform for machine learning. TensorFlow is a rich system for managing all aspects of a machine learning system; however, this class focuses on using a particular TensorFlow API to develop and train machine learning models.

TensorFlow is a popular open-source library for machine learning and deep learning. It provides a comprehensive set of tools, APIs, and computational resources for building and training various types of machine learning models, especially neural networks.

Here are some key features and functionalities of TensorFlow:

* Neural Network Framework:
* Computational Graphs
* Automatic Differentiation
* GPU and TPU Support
* Distributed Computing
* Deployment Capabilities
* **Tkinter**

Tkinter is an acronym for "Tk interface". Tk was developed as a GUI extension for the Tcl scripting language by John Ousterhout. The first release was in 1991. Tkinter is the de facto way in Python to create Graphical User interfaces (GUIs) and is included in all standard Python Distributions. In fact, it's the only framework built into the Python standard library.

Tkinter is a standard Python library used for creating graphical user interfaces (GUIs). It provides a set of modules and classes that allow you to develop interactive and visually appealing desktop applications.

Here are some key features and functionalities of Tkinter:

* Cross-Platform Compatibility
* Simple and Easy-to-Use
* Widgets and Layout Management
* Event-Driven Programming
* Customization and Styling
* Integration with Other Libraries
* **NLTK**

NLTK is a toolkit build for working with NLP in Python. It provides us various text processing libraries with a lot of test datasets. A variety of tasks can be performed using NLTK such as tokenizing, parse tree visualization, etc NLTK (Natural Language Toolkit) is the go-to API for NLP (Natural Language Processing) with Python. It is a really powerful tool to pre-process text data for further analysis like with ML models for instance. It helps convert text into numbers, which the model can then easily work with.

NLTK (Natural Language Toolkit) is a Python library widely used for working with human language data and implementing natural language processing (NLP) tasks. It provides a set of tools, corpora, and resources for tasks such as tokenization, stemming, tagging, parsing, sentiment analysis, and more.

Here are some key features and functionalities of NLTK:

* Text Processing
* Part-of-Speech Tagging
* Named Entity Recognition
* Chunking and Parsing
* Sentiment Analysis:
* WordNet Integration:
* **Scipy**

SciPy is a collection of mathematical algorithms and convenience functions built on the NumPy extension of Python. It adds significant power to the interactive Python session by providing the user with high-level commands and classes for manipulating and visualizing data.

SciPy is a powerful scientific computing library for Python that provides a wide range of mathematical algorithms and functions. It builds upon NumPy, another fundamental library for numerical computing, and extends its capabilities by adding additional tools for scientific and technical computing tasks.

Here are some key features and functionalities of SciPy:

* Numerical Integration:
* Optimization and Root Finding
* Linear Algebra
* Signal and Image Processing
* Statistics

**Used Algorithms:**

**SVM:**

4. **Support Vector Machine (SVM):**

Support Vector Machine (SVM) is a powerful machine learning algorithm used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are adaptable and efficient in a variety of applications because they can manage high-dimensional data and nonlinear relationships. SVM algorithms are very effective as we try to find the maximum separating hyperplane between the different classes available in the target feature. Support Vector Machine (SVM) is a [supervised machine learning](https://www.geeksforgeeks.org/supervised-unsupervised-learning/) algorithm used for both classification and regression. Though we say regression problems as well it’s best suited for classification. The main objective of the SVM algorithm is to find the optimal [hyperplane](https://www.geeksforgeeks.org/separating-hyperplanes-in-svm/) in an N-dimensional space that can separate the data points in different classes in the feature space. The hyperplane tries that the margin between the closest points of different classes should be as maximum as possible. The dimension of the hyperplane depends upon the number of features. If the number of input features is two, then the hyperplane is just a line. If the number of input features is three, then the hyperplane becomes a 2-D plane. It becomes difficult to imagine when the number of features exceeds three.

Let’s consider two independent variables x1, x2, and one dependent variable which is either a blue circle or a red circle.



Linearly Separable Data points

From the figure above it’s very clear that there are multiple lines (our hyperplane here is a line because we are considering only two input features x1, x2) that segregate our data points or do a classification between red and blue circles. So how do we choose the best line or in general the best hyperplane that segregates our data points?

One reasonable choice as the best hyperplane is the one that represents the largest separation or margin between the two classes.



Multiple hyperplanes separate the data from two classes

So we choose the hyperplane whose distance from it to the nearest data point on each side is maximized. If such a hyperplane exists it is known as the **maximum-margin hyperplane/hard margin**. So from the above figure, we choose L2. Let’s consider a scenario like shown below



Selecting hyperplane for data with outlier

Here we have one blue ball in the boundary of the red ball. So how does SVM classify the data? It’s simple! The blue ball in the boundary of red ones is an outlier of blue balls. The SVM algorithm has the characteristics to ignore the outlier and finds the best hyperplane that maximizes the margin. SVM is robust to outliers.



Hyperplane which is the most optimized one

So in this type of data point what SVM does is, finds the maximum margin as done with previous data sets along with that it adds a penalty each time a point crosses the margin. So the margins in these types of cases are called **soft margins**. When there is a soft margin to the data set, the SVM tries to minimize *(1/margin+∧(∑penalty))*. Hinge loss is a commonly used penalty. If no violations no hinge loss.If violations hinge loss proportional to the distance of violation.

Till now, we were talking about linearly separable data(the group of blue balls and red balls are separable by a straight line/linear line). What to do if data are not linearly separable?



Original 1D dataset for classification

Say, our data is shown in the figure above. SVM solves this by creating a new variable using a **kernel**. We call a point xion the line and we create a new variable yi as a function of distance from origin o.so if we plot this we get something like as shown below



Mapping 1D data to 2D to become able to separate the two classes

In this case, the new variable y is created as a function of distance from the origin. A non-linear function that creates a new variable is referred to as a kernel.

**Support Vector Machine Terminology**

* **Hyperplane:**Hyperplane is the decision boundary that is used to separate the data points of different classes in a feature space. In the case of linear classifications, it will be a linear equation i.e. wx+b = 0.
* **Support Vectors:**Support vectors are the closest data points to the hyperplane, which makes a critical role in deciding the hyperplane and margin.
* **Margin**: Margin is the distance between the support vector and hyperplane. The main objective of the support vector machine algorithm is to maximize the margin.  The wider margin indicates better classification performance.
* **Kernel**: Kernel is the mathematical function, which is used in SVM to map the original input data points into high-dimensional feature spaces, so, that the hyperplane can be easily found out even if the data points are not linearly separable in the original input space. Some of the common kernel functions are linear, polynomial, radial basis function(RBF), and sigmoid.
* **Hard Margin:** The maximum-margin hyperplane or the hard margin hyperplane is a hyperplane that properly separates the data points of different categories without any misclassifications.
* **Soft Margin:**When the data is not perfectly separable or contains outliers, SVM permits a soft margin technique. Each data point has a slack variable introduced by the soft-margin SVM formulation, which softens the strict margin requirement and permits certain misclassifications or violations. It discovers a compromise between increasing the margin and reducing violations.
* **C:**Margin maximisation and misclassification fines are balanced by the regularisation parameter C in SVM. The penalty for going over the margin or misclassifying data items is decided by it. A stricter penalty is imposed with a greater value of C, which results in a smaller margin and perhaps fewer misclassifications.
* **Hinge Loss:** A typical loss function in SVMs is hinge loss. It punishes incorrect classifications or margin violations. The objective function in SVM is frequently formed by combining it with the regularisation term.
* **Dual Problem:** A dual Problem of the optimisation problem that requires locating the Lagrange multipliers related to the support vectors can be used to solve SVM. The dual formulation enables the use of kernel tricks and more effective computing.

**SVM implementation in Python**

Predict if cancer is Benign or malignant. Using historical data about patients diagnosed with cancer enables doctors to differentiate malignant cases and benign ones are given independent attributes.

**Steps**

* Load the breast cancer dataset from sklearn.datasets
* Separate input features and target variables.
* Buil and train the SVM classifiers using RBF kernel.
* Plot the scatter plot of the input features.
* Plot the decision boundary.
* Plot the decision boundary

Example:

# Load the important packages

from sklearn.datasets import load\_breast\_cancer

import matplotlib.pyplot as plt

from sklearn.inspection import DecisionBoundaryDisplay

from sklearn.svm import SVC

# Load the datasets

cancer = load\_breast\_cancer()

X = cancer.data[:, :2]

y = cancer.target

#Build the model

svm = SVC(kernel="rbf", gamma=0.5, C=1.0)

# Trained the model

svm.fit(X, y)

# Plot Decision Boundary

DecisionBoundaryDisplay.from\_estimator(

svm,

X,

response\_method="predict",

cmap=plt.cm.Spectral,

alpha=0.8,

xlabel=cancer.feature\_names[0],

ylabel=cancer.feature\_names[1],

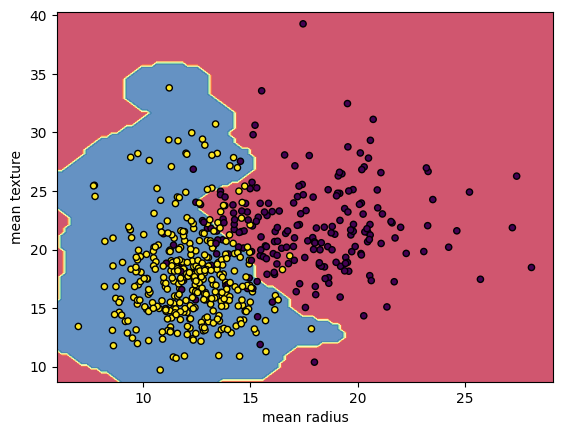
)

# Scatter plot

plt.scatter(X[:, 0], X[:, 1], c=y, s=20, edgecolors="k")

plt.show()

**Output**:



Breast Cancer Classifications with SVM RBF kernel

**BI-LSTM:**

**BERT,**

**It is an acronym** **for Bidirectional Encoder Representations from Transformers**, stands as an open-source **machine learning framework**designed for the realm of **natural language processing (NLP)**. Originating in 2018, this framework was crafted by researchers from Google AI Language. The article aims to explore the **architecture, working and applications of BERT**.

## What is BERT?

BERT (Bidirectional Encoder Representations from Transformers) leverages a transformer-based neural network to understand and generate human-like language. BERT employs an encoder-only architecture. In the original Transformer architecture, there are both encoder and decoder modules. The decision to use an encoder-only architecture in BERT suggests a primary emphasis on understanding input sequences rather than generating output sequences.

### Bidirectional Approach of BERT

Traditional language models process text sequentially, either from left to right or right to left. This method limits the model’s awareness to the immediate context preceding the target word. BERT uses a bi-directional approach considering both the left and right context of words in a sentence, instead of analyzing the text sequentially, BERT looks at all the words in a sentence simultaneously.

### Pre-training and Fine-tuning

The BERT model undergoes a two-step process:

* Pre-training on Large amounts of unlabeled text to learn contextual embeddings.
* Fine-tuning on labeled data for specific [NLP](https://www.geeksforgeeks.org/natural-language-processing-overview/) tasks.

#### Pre-Training on Large Data

* BERT is pre-trained on large amount of unlabeled text data. The model learns contextual embeddings, which are the representations of words that take into account their surrounding context in a sentence.
* BERT engages in various unsupervised pre-training tasks. For instance, it might learn to predict missing words in a sentence (Masked Language Model or MLM task), understand the relationship between two sentences, or predict the next sentence in a pair.

#### Fine-Tuning on Labeled Data

* After the pre-training phase, the BERT model, armed with its contextual embeddings, is then fine-tuned for specific natural language processing (NLP) tasks. This step tailors the model to more targeted applications by adapting its general language understanding to the nuances of the particular task.
* BERT is fine-tuned using labeled data specific to the downstream tasks of interest. These tasks could include sentiment analysis, question-answering, [named entity recognition](https://www.geeksforgeeks.org/named-entity-recognition/), or any other NLP application. The model’s parameters are adjusted to optimize its performance for the particular requirements of the task at hand.

BERT’s unified architecture allows it to adapt to various downstream tasks with minimal modifications, making it a versatile and highly effective tool in [natural language understanding](https://www.geeksforgeeks.org/nlp-vs-nlu-vs-nlg/) and processing.

**5.7 Sample Code:**

from sklearn\_extensions.extreme\_learning\_machines.random\_layer import RandomLayer, MLPRandomLayer

from sklearn\_extensions.extreme\_learning\_machines import elm

import numpy as np

from scipy.linalg import pinv2

from sklearn.utils import as\_float\_array

from sklearn.utils.extmath import safe\_sparse\_dot

#creating Genetic EML class

class GeneticELMRegressor(elm.BaseELM, elm.RegressorMixin):

def \_\_init\_\_(self, hidden\_layer=MLPRandomLayer(random\_state=0), regressor=None):

super(GeneticELMRegressor, self).\_\_init\_\_(hidden\_layer, regressor)

self.coefs\_ = None

self.fitted\_ = False

self.hidden\_activations\_ = None

#genetic algorithm fitness function using RMSE

def fitness(self, y):

if self.regressor is None:

self.coefs\_ = safe\_sparse\_dot(pinv2(self.hidden\_activations\_), y)

else:

self.regressor.fit(self.hidden\_activations\_, y)

self.fitted\_ = True

def fit(self, X, y):

# fit random hidden layer and compute the hidden layer activations

self.hidden\_activations\_ = self.hidden\_layer.fit\_transform(X)

# solve the regression from hidden activations to outputs

self.fitness(as\_float\_array(y, copy=True))

return self

#crossover to generate new population and then evaluate fitness

def crossover(self):

if self.regressor is None:

preds = safe\_sparse\_dot(self.hidden\_activations\_, self.coefs\_)

else:

preds = self.regressor.predict(self.hidden\_activations\_)

return preds

def predict(self, X):

if not self.fitted\_:

raise ValueError("ELMRegressor not fitted")

# compute hidden layer activations

self.hidden\_activations\_ = self.hidden\_layer.transform(X)

# compute output predictions for new hidden activations

predictions = self.crossover()

return predictions

**6. TESTING:**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

**Implementation**

The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifying as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## Testing

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually, testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### System Testing

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus, all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example, the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus, the mapping of jobs with resources is done correctly by the system.

**Acceptance Testing**

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | | **Actual** |
| 01 | defining interpolate function | Verify interpolate function defined or not | If interpolate function may not defined | we cannot do any further operations | we can do further operations | | High | High |
| 02 | Preprocess Dataset | Verify Preprocess Dataset or not | If Dataset may not Preprocess | we cannot do any further operations | we can do further operations | | High | High |
| 03 | Run SVM Algorithm | Verify Run SVM Algorithm or not | If Run SVM Algorithm not be | we cannot do any further operations | we can do further operations | | High | High |
| 04 | Run genetic ELM Algorithm | Verify Run ELM Algorithm or not | If ELM Algorithm not Run | We cannot run  operation | We can Run the Operation | | High | High |
| 05 | Run BI-LSTM algorithm | Verify Bi-LSTM algorithm run or not | If Bi-LSTM algorithm not run | We cannot run  operation | We can Run the Operation | | High | High |

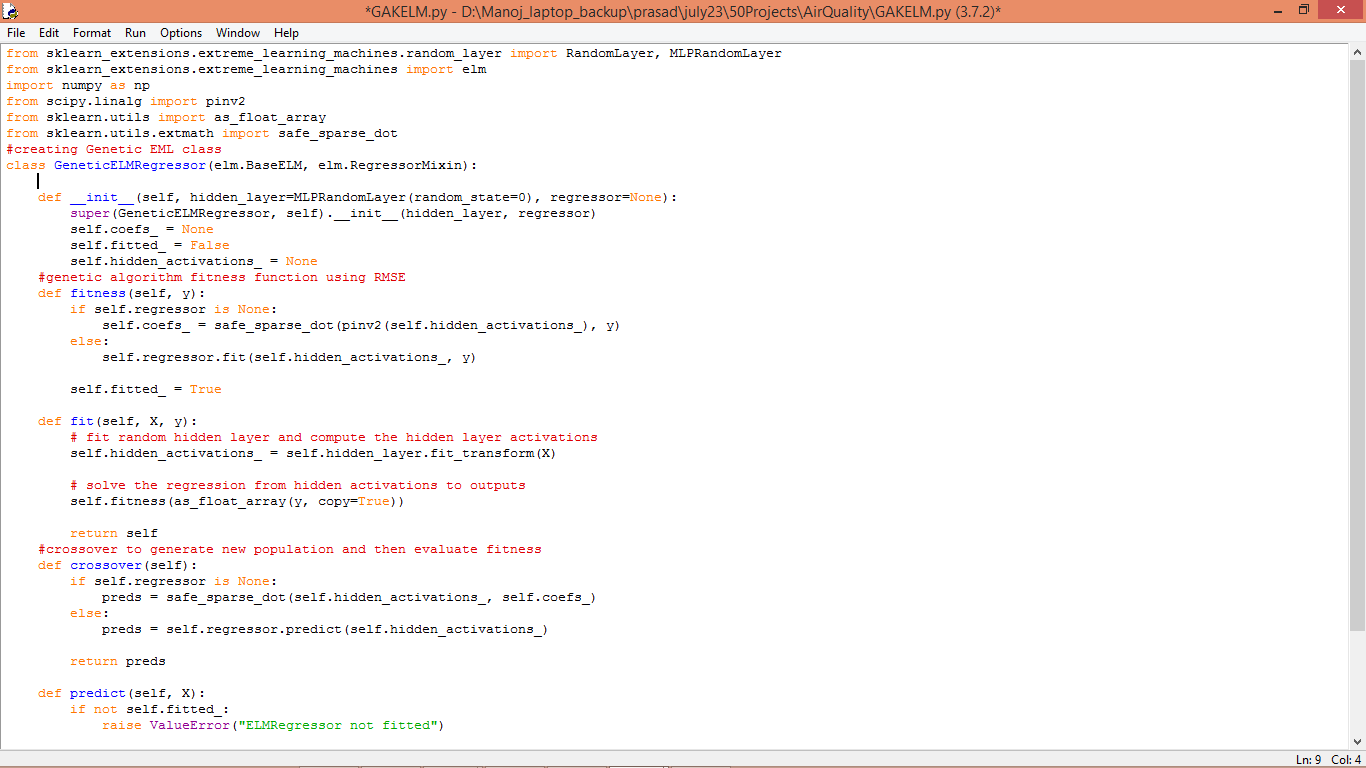
**7. SCREENSHOTS:**

For humans to live Air and Water is the basic requirement but because of pollutions both basic requirements are getting polluted and now it become mandatory to adjust pollution by monitoring Air Quality, if air quality is low then government can make necessary arrangement like reducing carbon emission and other dangerous chemicals.

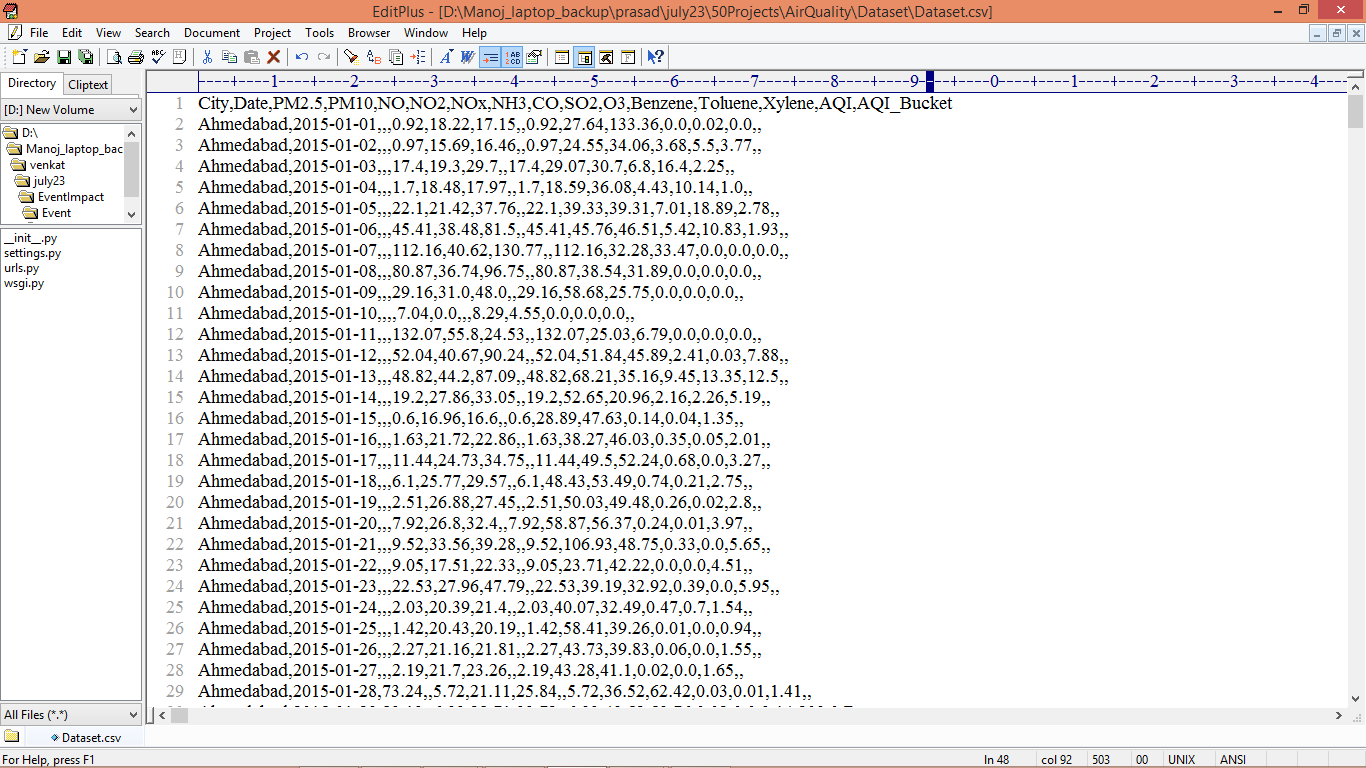
In the past many deep and machine learning algorithms are introduced but their performance is not accurate as training weights of those algorithms are not accurate enough to predict Air Quality with high accuracy and less error rate like MSE (mean square error) and RMSE (root mean square error). Both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

So to update weights accurately author of this paper enhancing Extreme Learning Machine with Genetic Algorithm (GA-KLEM). To solve the issue of the conventional limit learning machine where the number of hidden nodes and the random generation of thresholds and weights lead to the degradation of the network learning ability, a genetic algorithm is then used to optimize the number of hidden nodes and layers of the kernel limit learning machine. The thresholds, the weights, and the root mean square error are used to define the fitness function. Finally, the least squares method is applied to compute the output weights of the model. Genetic algorithms are able to find the optimal solution in the search space and gradually improve the performance of the model through an iterative optimization process.

To enhance ELM, we have added genetic algorithm functions like Fitness (weight fitness will be check using RMSE function), crossover and mutation inside ELM to update weights. In below screen showing Enhance ELM code



In above screen read red color comments to know about GA-KLEM algorithm. Propose algorithm is compare with traditional algorithm called SVM and each algorithm performance is evaluated using MSE and RMSE. To train all algorithm author has used Air Quality dataset and below screen showing dataset details.

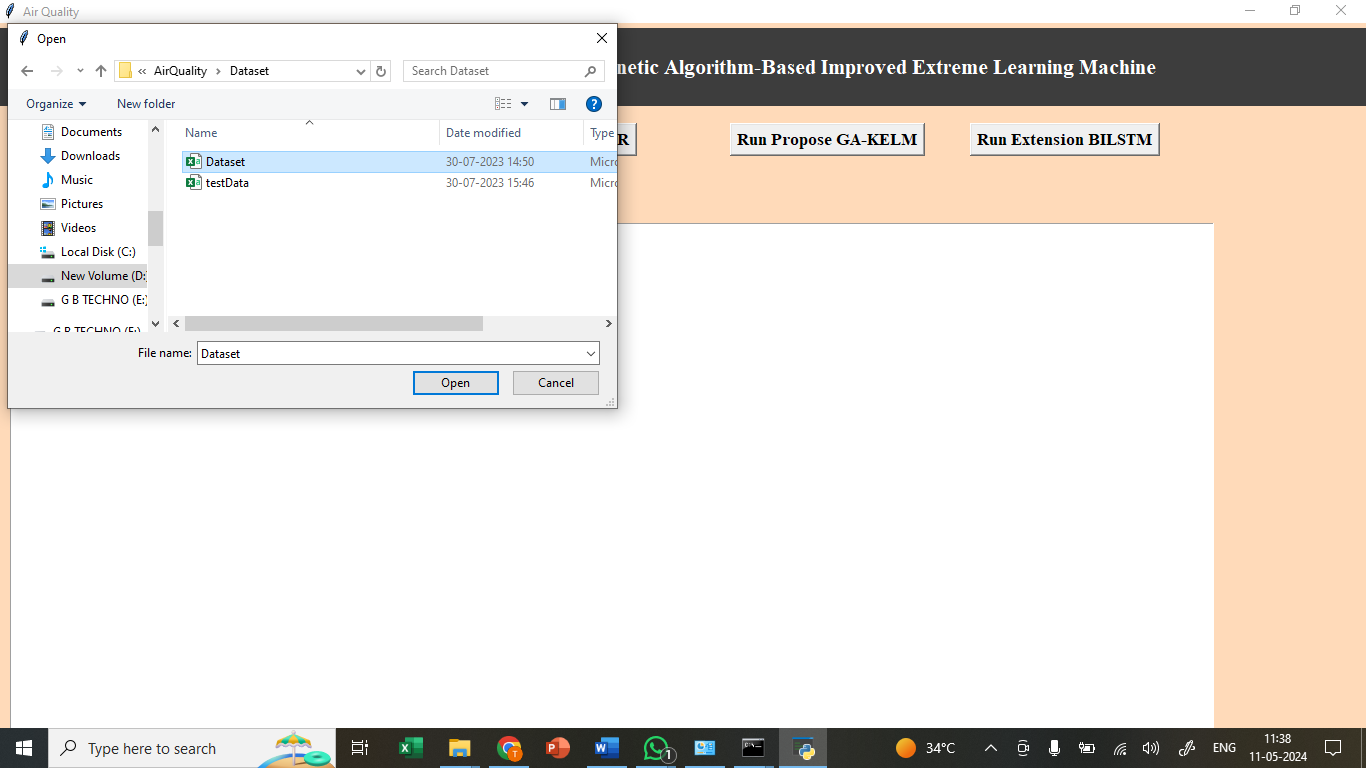


In above dataset screen first row represents dataset column names and remaining rows represents dataset values and we can test with each column for Air Quality prediction but we are using PM2.5 column for Air Quality prediction. So, by using above dataset we will train and test each algorithm performance.

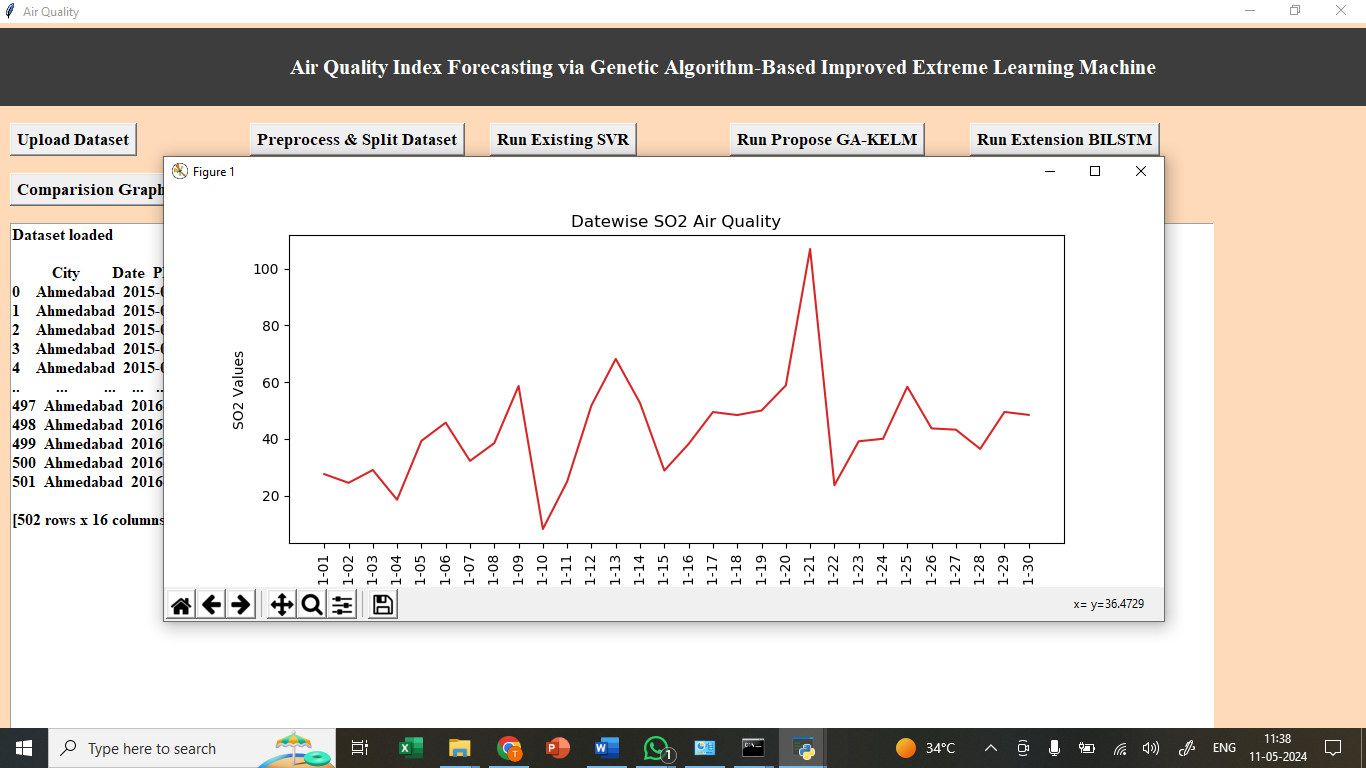
Extension Concept:

In propose paper author plan to enhance traditional ELM algorithm using GA optimization but as extension we have experimented with BI-LSTM algorithm which will optimize features weight in both forward and backward direction. BI-LSTM will be optimized features till no more optimizations are possible so prediction accuracy will automatically get increased and error MSE rate will get decrease.

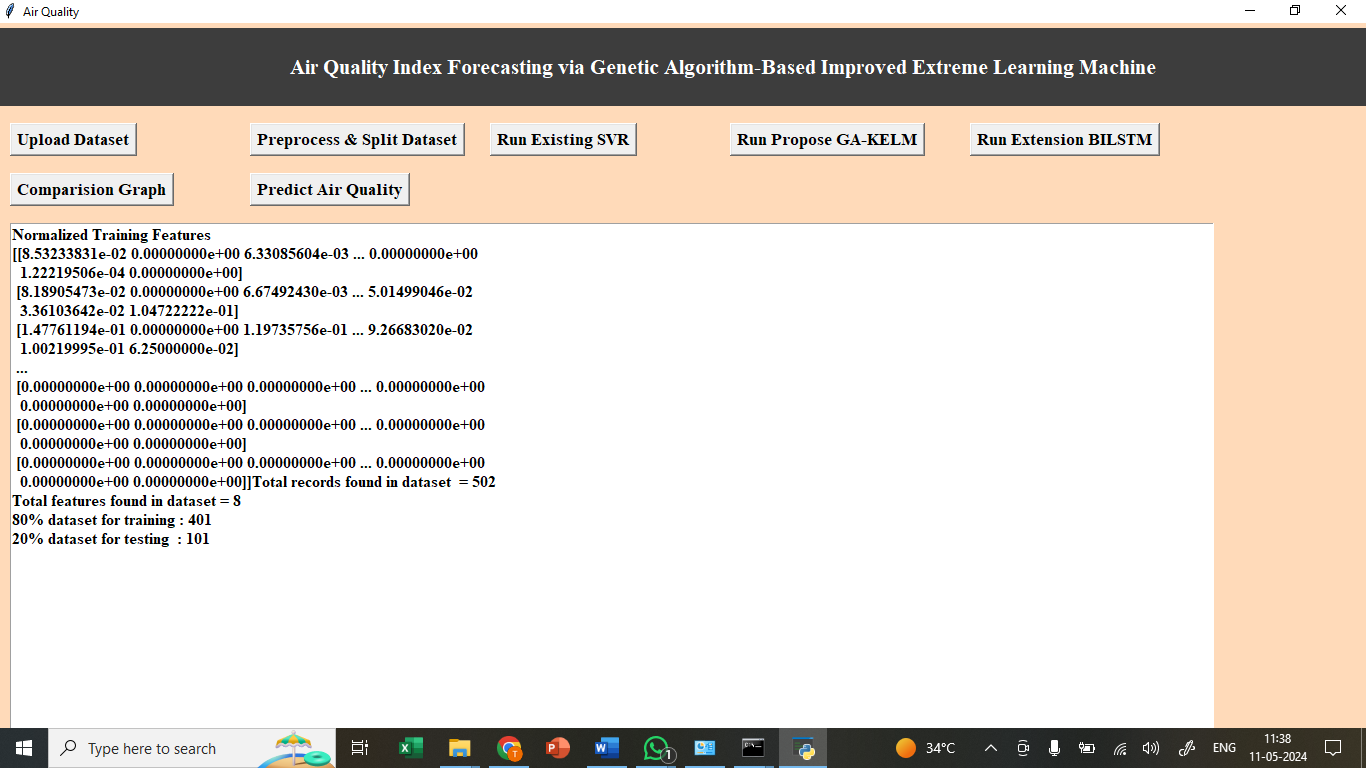
SCREEN SHOTS



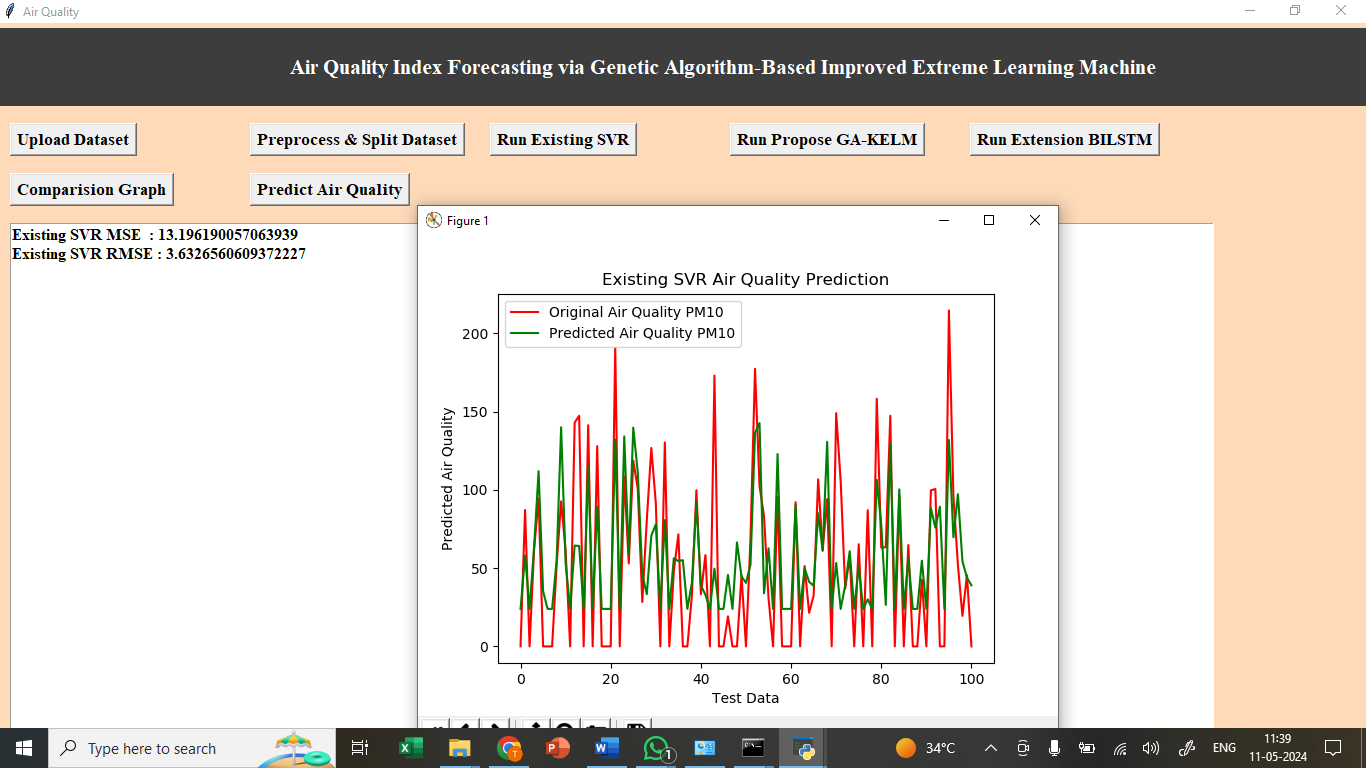
In above screen uploading dataset



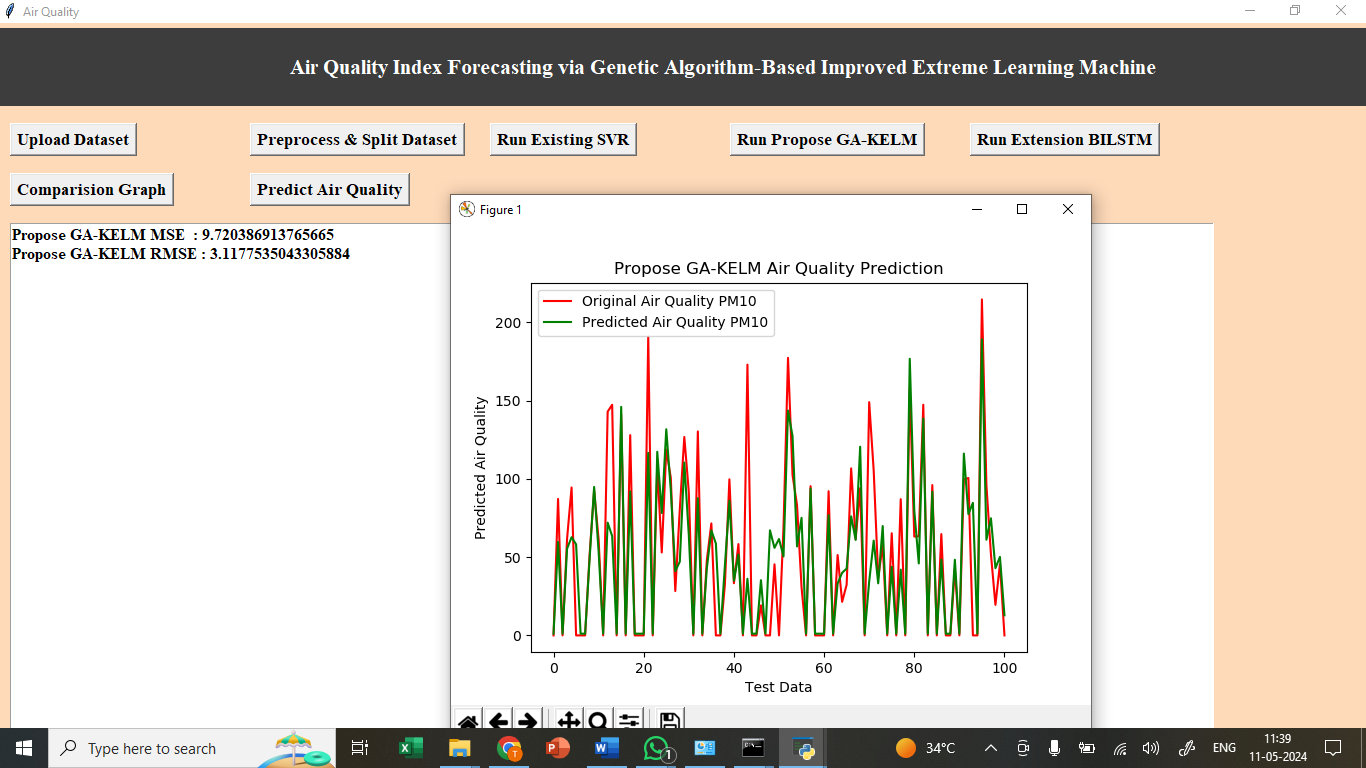
In above graph displaying Air Quality where x-axis represents Date and y-axis represents air quality



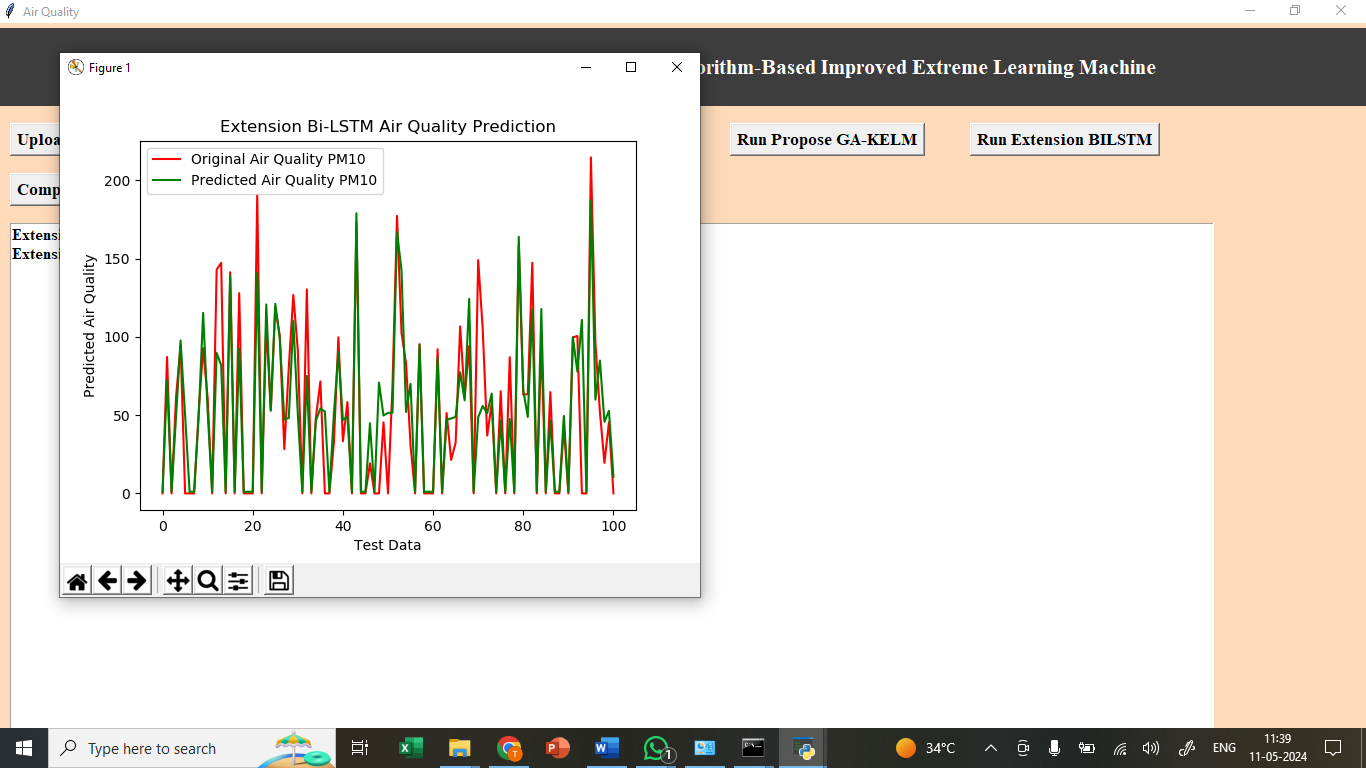
In above screen splitting dataset into train and test



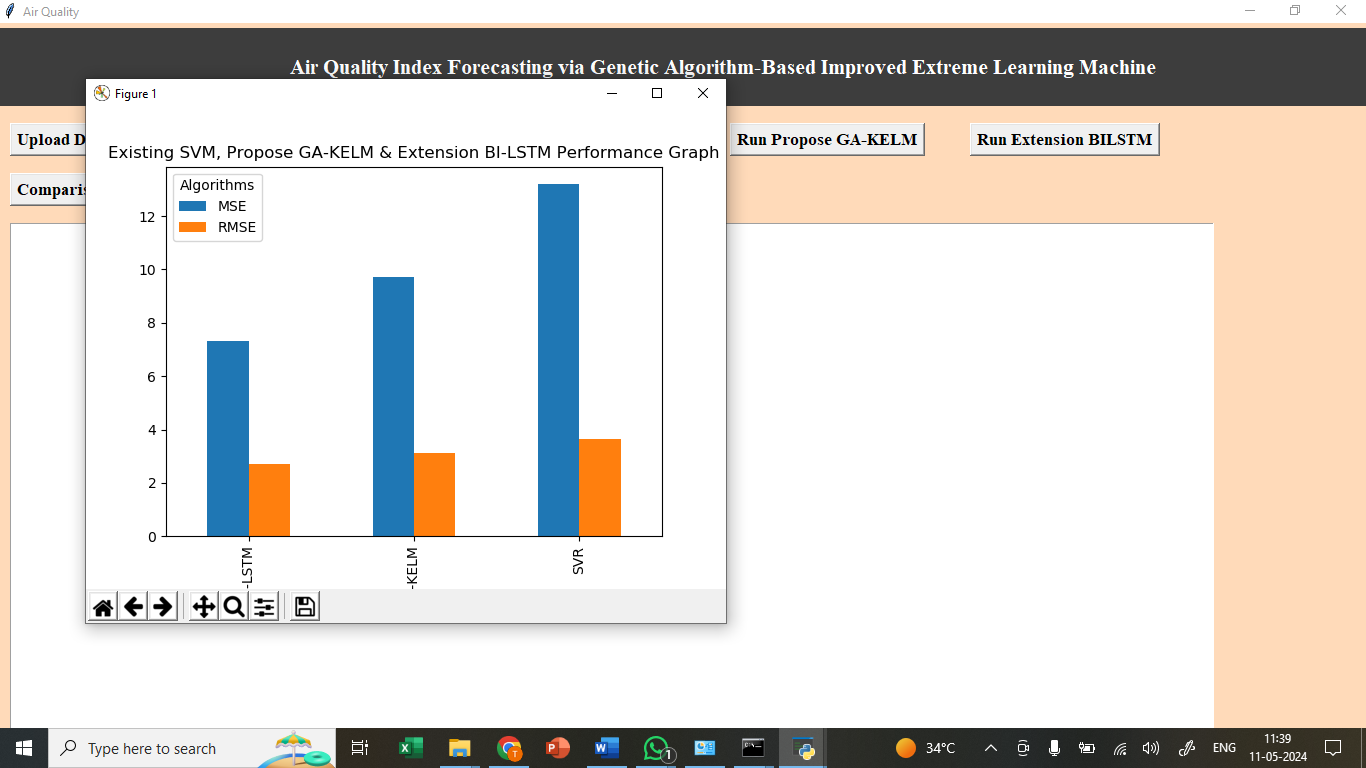
In above screen training SVM algorithm and then with SVM we got 11 as the MSE and in graph x-axis represents TEST COUNT and y-axis represents air quality. Red line represents Original Test Air Quality and green line represents Predicted Air Quality and we can see both lines are closed as they are overlapping with little gaps and this gap can reduce by applying propose algorithm



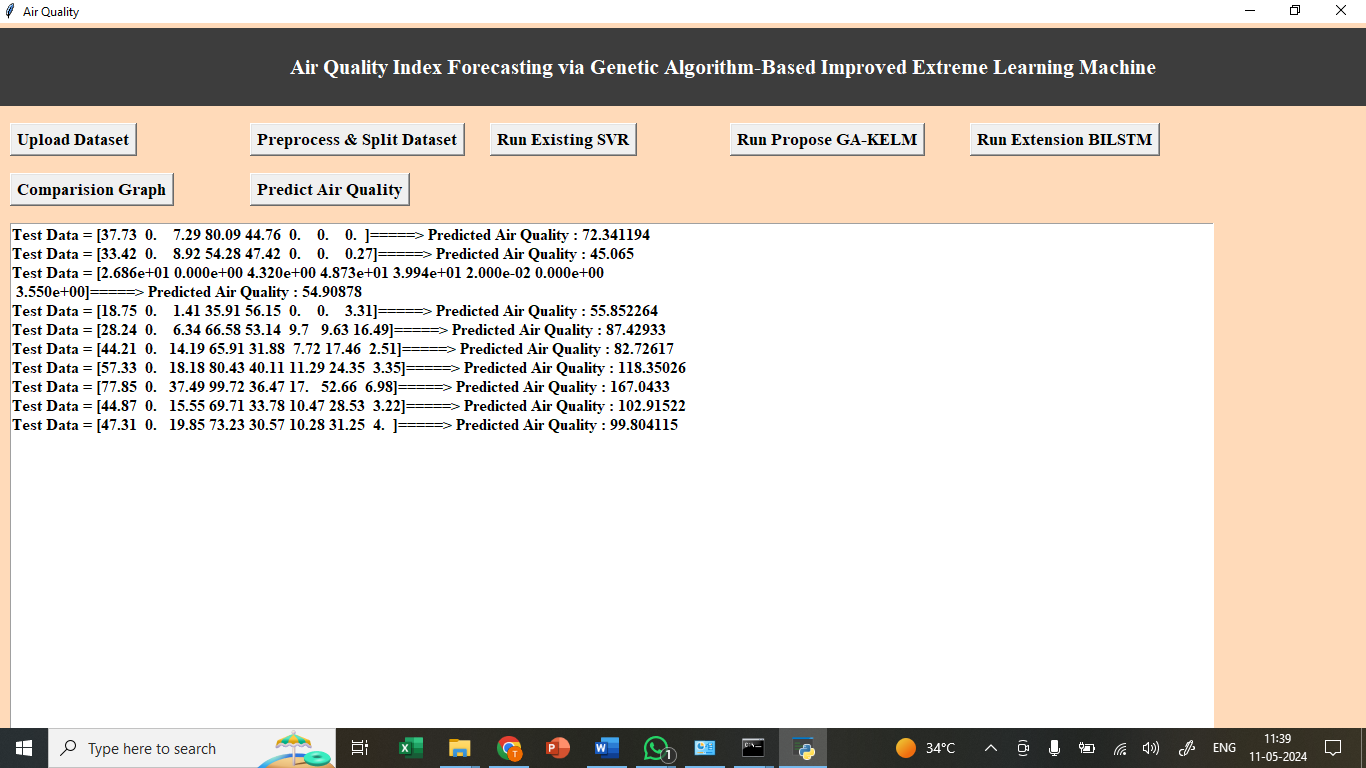
In above screen we are training propose genetic ELM called GA-KELM and we got its MSE as 6 and In graph we can see now both lines are overlap with too few gap.



In above screen with extension BI-LSTM we got MSE as 5% and in graph also nearly 90% test and predicted air quality lines are overlapping and in all algorithms extension BI-LSTM has got less MSE and RMSE



In above graph x-axis represents algorithm names and y-axis represents MSE and RMSE in different colour bars and in all algorithms extension BI-LSTM got less RMSE and MSE



In above screen loading TEST data and then predicting air quality using extension object and then we can see TEST data and predicted air quality after = arrow symbol

**8. CONCLUSION:**

The economic development achieved by the country through rapid urbanization is polluting the environment in an alarming way and putting people’s lives in danger. Therefore, a correct analysis and accurate prediction of air quality remains a primary condition to achieve the objective of sustainable development. This paper focuses on the problem of prediction model design, and investigates the problems related to the optimization of the model parameters. A GA-KELM model is designed, implemented, and tested. It is experimentally proven to be more efficient than the classical shallow learning and can effectively explore and learn the interdependence of multivariate air quality correlation time series such as temperature, humidity, wind speed, SO2, and PM10. Therefore, the GA-KELM model developed in this study can be used to provide valuable support to vulnerable groups and trigger early warning of adverse air quality events. However, there are still areas for further investigation and improvement. In recent years, numerous advanced algorithms and optimization methods based on genetic algorithms and population intelligence have emerged. Therefore, future research should explore the underlying significance and value of combinatorial intelligence optimization algorithms such as the Limit Learning Machine. Additionally, the issue of manually setting the number of hidden layer nodes in the optimal Limit Learning Machine. Although the Dynamic Extreme Learning Machine (DELM) algorithm offers adaptive determination of hidden layer nodes without human intervention, further work should be dedicated to this aspect. Moreover, to enhance the accuracy and validity of air quality measurement and assessment, it is crucial to integrate pollutant emission factors and meteorological factors into the evaluation system. This integration will enable a more precise and comprehensive evaluation of air quality. In conclusion, our study highlights the significance of the GA-KELM model in predicting air quality. We have addressed the optimization challenges and demonstrated its superiority over traditional methods. However, there is still room for improvement and further research. Future studies should delve into advanced optimization algorithms based on genetic algorithms and population intelligence, explore the potential of the Limit Learning Machine, and strive for adaptive determination of hidden layer nodes. Furthermore, the integration of pollutant emission factors and meteorological factors into the evaluation system will advance the accuracy and reliability of air quality measurement and assessment.

**9. REFERENCES:**

[1] X. Li, L. Jinn, and H. Kan, “Air pollution: A global problem needs local fixes,” Nature, vol.570, pp.437–439, 2019.

[2] Y. Han, J. C. K. Lam, and V. O. K. Li, “A Bayesian LSTM model to evaluate the effects of air pollution control regulations in China, “in Proc. IEEE Big Data Workshop Big Data AI Air Qual. EstimationForecasting Health Advice, 2018, pp. 4465–4468.

[3] L. Bai, J. Wang, X. Ma, and H. Lu, “Air pollution forecasts: An overview,” Int. J. Environ. Res. Public Health, vol. 15, no. 4, 2018, Art. no. 780.

[4] Y. Ding and Y. Xu, “A Deep Learning Approach to Writer Identification Using Inertial Sensor Data of Air-Handwriting,” IEICE Transient. & Syst., vol. E102-D, no.10, pp.2059–2063, 2019.

[5] S.-Q. Dotes, M. I. Petra, L. Dagger, and L. C. De Silva, “Application of computational intelligence techniques to forecast daily PM10 exceedances in Brunei Darussalam,” Atmos. Pollute. Res., vol. 9, no. 2, pp. 358368,2018.

[6] M. Jia, A. Comelily, Y. Wang, and R. S. Srinivasan, “Adopting Internet of Things for the development of smart buildings: A review of enabling technology gives and applications,” Automat. Construct., vol. 101, pp. 111126, May 2019.

[7] Abram, S., Chitra, P., Mahometh, R., Keserwan, S.R., 2020. Hybrid Spatio-temporal Deep Learning Framework for Particulate Matter (PM2.5) Concentration Forecasting.2020 International Conference on Innovative Trends in Information Technology (ICITIIT) 1e6.

[8] Y. Cheng, S. Zhang, C. Huan, M. O. Loudoun, and Z. Lin, “Optimization on fresh outdoor air ratio of air conditioning system with stratum venti-lotion for both targeted indoor air quality and maximal energy saving, “Building Environ., vol. 147, pp. 1122, Jan. 2019.

[9] A. C. Cosmas and R. Simcha, “Machine learning method for real-time non-invasive prediction of individual thermal preference in transient conditions,” Building Environ., vol. 148, pp. 372383, Jan. 2019.

[10] Bhowmik, M., Deb, K., Debnath, A., Asha, B., 2018. Mixed phase Fe 2 O 3/Mn 3 O 4 magnetic nanocomposite for enhanced adsorption of methyl orange dye: neural network modeling and response surface methodology optimization. Appl.Organomet. Chem. 32, e4186 <https://doi.org/10.1002/aoc.4186>.

[11] Chaudhary, V., Ditchwater, A., Kumar, V., Paul, D., 2018. Time Series Based LSTM Model to Predict Air Pollutant’s Concentration for Prominent Cities in Indium, Aug.

[12] Chen, M., Yang, J., Hu, L., Shamim Hossain, M., Muhammad, G., 2018. Urban healthcare big data system based on crowdsourced and cloud-based air quality indicators. IEEE Commune. Mag. 56, 14e20. <https://doi.org/10.1109/MCOM.2018.1700571>

[13] Chen, R., Wang, X., Zhang, W., Zhu, X., Li, A., Yang, C., 2019. A hybrid CNN-LSTM model for typhoon formation forecasting. Geoinformatics 23, 375e396. <https://doi.org/10.1007/s10707-019- 00355-0>.

[14] Du, S., Li, T., Yang, Y., Hong, S.-J., 2019. Deep air quality forecasting using hybrid deep learning framework. IEEE Trans. Knowle. Data Eng. https://doi.org/10.1109/ tkde.2019.2954510, 1-1..

[15] Feng, R., Zheng, H., Gao, H., Zhang, A., Huang, C., Zhang, J., Luo, K., Fan, J., 2019.Recurrent Neural Network and random forest for analysis and accurate forecast of atmospheric pollutants: a case study in Hangzhou, China. J. Clean. Prod.