Mutual Information based Feature Selection and Ensemble Learning for Enhanced Air Quality Index Classification

Thesis submitted in partial fulfillment of
requirements for the degree of

Master of Technology in Computer Technology
of

Computer Science and Engineering
Department

of

Jadavpur University

by

Raka Ganguli
Regn. No. - 160095 of 2021 – 2022

Exam Roll No. – M6TCT24019

Under the supervision of

Dr. Susmita Ghosh (De)
Professor

Department of Computer Science and Engineering

Jadavpur University, Kolkata, West Bengal, India

CERTIFICATE FROM THE SUPERVISOR

This is to certify that the thesis work titled **"Mutual Information based Feature Selection and Ensemble Learning for Enhanced Air Quality Index Classification"** has been satisfactorily completed by Raka Ganguli (Registration Number 160095 of 2021-2022; Class Roll No. 002110504024; Examination Roll No. *M* 6*TCT* 24019). It is a bona-fide piece of work carried out under my supervision and guidance at Jadavpur University, Kolkata for partial fulfilment of the requirements for awarding the Master of Technology in Computer Technology degree under the Department of Computer Science and Engineering, Faculty of Engineering and Technology, Jadavpur University, during the academic year 2023−2024.

Dr. Susmita Ghosh (De),
Professor

Department of Computer Science and Engineering, Jadavpur University
(Supervisor)

Forwarded By:

Prof. Nirmalya Chowdhury, Head of Department

Department of Computer Science and Engineering, Jadavpur University.

Prof. Dipak Laha,
Dean

Faculty of Engineering &Technology,
Jadavpur University

CERTIFICATE OF APPROVAL

This is to certify that the thesis entitled **"Mutual Information based Feature Selection and Ensemble Learning for Enhanced Air Quality Index Classification"** is a bona-fide record of work carried out by **Raka Ganguli** (Registration Number 160095 of 2021-2022; Class Roll No. 002110504024; Examination Roll No. *M* 6*TCT* 24019) in partial fulfilment of the requirements for the award of the degree of **Master of Technology in Computer Technology** in the **Department of Computer Science and Engineering, Jadavpur University**, during the period of July 2023 to August 2024. It is understood that by this approval, the undersigned do not necessarily endorse or approve any statement made, opinion expressed or conclusion drawn therein but approve the thesis only for the purpose for which it has been submitted.

Examiners:

_____ _____

(Signature of Examiner) (Signature of Supervisor)

DECLARATION OF ORIGINALITY AND COMPLIANCE OF ACADEMIC ETHICS

I hereby declare that the thesis entitled **"Mutual Information based Feature Selection and Ensemble Learning for Enhanced Air Quality Index Classification"** contains literature survey and original research work by the undersigned candidate, as a part of her degree of **Master of Technology in Computer Technology** in the **Department of Computer Science and Engineering, Jadavpur University**. All information has been obtained and presented in accordance with academic rules and ethical conduct.

I also declare, as required by these rules and conduct, I have fully cited and referenced all materials and results that are not original to this work.

Name: Raka Ganguli

Examination Roll No.: M6TCT24019

Registration No.: 160095 of 2021 − 2022

Thesis Title: Mutual Information based Feature Selection and Ensemble Learning for Enhanced Air Quality Index Classification

Signature of the Candidate:

ACKNOWLEDGEMENT

I am pleased to express my gratitude and regards towards my supervisor **Dr. Susmita Ghosh (De),** Professor, Department of Computer Science and Engineering, Jadavpur University, without whose valuable guidance, inspiration and unwavering attention to detail, pursuing my thesis would have been impossible.

Last but not the least, I express my regards towards my family and friends for supporting me throughout this journey and for being a constant source of motivation during the entire term of the work.

Raka Ganguli

Exam Roll No.
- M6TCT24019

Regn. No. -
160095 of 2021-2022

Department of Computer Science and Engineering,

Jadavpur University.

ABSTRACT

Features are an important resource in many classification protocols. However, working with all features in a given dataset requires a considerable amount of effort due to the computational complexity and increased training time. Often, one ends up with a set of features with a significant amount of dependence amidst one another. Hence it is important to develop protocols that extract the features with the least dependence between one another. This thesis delves into the application of information theoretic feature selection methods, specifically focusing on mutual information, to enhance the accuracy of predictive models for air quality classification. Specifically, three feature selection methods—Mutual Information Based Feature Selection (MIFS), Normalized Mutual Information Based Feature Selection (NMIFS), and Mutual Information Feature Selection Based on Distance-Based Elimination Strategies (DMIFS) were employed to select subsets of 6 and 8 features each from the original dataset.

To benchmark the efficacy of these feature selection methods, we compared them against other state of the art methods, namely, Pearson Correlation Coefficient, Grey Wolf Optimization (GWO), and Principal Component Analysis (PCA). We utilized three classifiers: Extreme Gradient Boosting (XGBoost), K-Nearest Neighbors (KNN), and Support Vector Machine (SVM), each subjected to Optuna hyperparameter tuning and evaluated via 10-fold cross-validation. The performance of the models was evaluated using various metrics such as Accuracy, Precision, Recall, F1 Score, Mean Absolute Error (MAE), and the ROC-AUC curve. Additionally, confusion matrices were presented to provide deeper insights into model performance.

The highest accuracy for XGBoost classifier was 93.8%, achieved with NMIFS using 8 features. KNN also achieved a commendable accuracy of 93.7% with DMIFS using 8 features, while SVM achieved an accuracy of 89.6% with NMIFS using 8 features.

To further optimize the feature selection process, union and intersection operations were performed on the subsets of the obtained features, resulting in four distinct feature sets. In order to enhance model performance, we leveraged ensemble learning techniques. For the sake of building a robust ensemble model, we employed three base learners including XGBoost, KNN, and SVM. These models were integrated using stacking and majority voting techniques. The ensemble model achieved a peak accuracy of 94.2% on the union of features while the lowest accuracy achieved on the intersection of features was 93.8%.

The results demonstrate that feature selection significantly reduces computational complexity while maintaining high prediction accuracy. The NMIFS method, in particular, achieved the highest accuracy scores. These findings underscore the importance of feature selection in improving the efficiency and performance of predictive models in air quality monitoring.

By focusing on the information theoretic aspects of feature selection, we demonstrate the superior accuracy and efficiency of mutual information-based methods compared to other advanced feature selection strategies. The findings highlight the critical role of selecting a subset of features with minimal interdependence, which not only enhances the predictive power but also simplifies the modeling process. This research contributes to the field of environmental informatics by offering a robust model that can aid in better understanding and managing air pollution, ultimately leading to improved public health and environmental sustainability.

Content

Abstract **i**

Content **ii - v**

List of Figures **vi - vii**

List of Tables **viii**

List of Abbreviations **ix**

Chapter 1: INTRODUCTION AND BACKGROUND……..…….1

1.1 Air Pollution and Air Quality Index……....1

1.1.1 Impact of Air Pollution…….....1

1.1.2 Calculation of Air Quality Index……..…2

1.1.3 Classes of Air Quality Index…….2-3

1.1.4 Necessity of Calculating the Air Quality Index……...4

1.2 Machine Learning……....4-5

1.2.1 Uses of Machine Learning……...5

1.2.2 Types of Machine Learning……..5-8

1.2.3 Key Components of Supervised Machine Learning…….8-9

1.2.4 Challenges of Building a Machine Learning Model……...9-10

1.2.5 Importance of Machine Learning in the Classification of Air Quality Index……10

1.3 Feature Selection in Machine Learning…….11

1.3.1 Importance of Feature Selection…….11

1.3.2 Types of Feature Selection……11-12

1.4 Data Augmentation in Machine Learning…….12

1.4.1 Types of Data Augmentation……12-13

1.4.2 Importance of Data Augmentation…….14

1.5 Ensemble Learning Techniques……14

1.5.1 Utility of Ensemble Learning…….15

1.6 Objective and Contribution……16

Chapter 2: LITERATURE SURVEY..... …….15

2.1 Related Research and Previous Works..... …….15-19

2.2 Comparative Analysis of Previous Research.....19

Chapter 3 : PROPOSED METHODOLOGY……..19

3.1 Introduction to Proposed Methodology……...20

3.2 Data Preprocessing……22-24

3.3 Feature Selection…….24

3.3.1 Filter Methods……24-25

3.3.2 Wrapper Methods……25

3.3.3 Embedded Methods…….25

3.3.4 Dimensionality Reduction Methods……26

3.3.5 Hybrid Methods……...26

3.4 Mutual Information and Entropy……26-27

3.4.1 Mutual Information Based Feature Selection……..28

3.4.2 Normalized Mutual Information Based Feature Selection…….28-29

3.4.3 Weighted Mutual Information Based Feature Selection and Distance Based Elimination……..29-30

3.4.4 Grey Wolf Optimization Algorithm for Feature Selection……30-32

3.4.5 Pearson Correlation Coefficient Based Feature Selection…….32-33

3.4.6 Principal Component Analysis……..33-34

3.5 Data Augmentation to Handle Class Imbalance……...34-35

3.6 Classification Algorithm and its Variants Utilized for Each Feature Selection Method...35

3.6.1 Classification Algorithms……35

3.6.2 Components of Classification Algorithms…….36

3.6.3 Extreme Gradient Boosting Classifier……..36-37

3.6.4 K- Nearest Neighbors Classifier……...37-39

3.6.5 Support Vector Machine Classifier……...39-40

3.7 Ensemble Learning Techniques……...40-41

3.7.1 Union of Feature Sets……41-42

3.7.2 Intersection of Feature Sets……42

3.7.3 Building an Ensemble Learning Model……42-43

Chapter 4: DATASET AND EXPERIMENTAL SETUP……..…44

4.1 Dataset Used……...44

4.2 Dataset Preprocessing……45-47

4.3 Experimental Setup…….47

4.3.1 Hyperparameter Tuning for Classification Algorithm……....48-49

4.3.2 Hyperparameters Used for Ensemble Learning Model……..49-50

4.4 Performance Evaluation Metrics of the Classification Models……...51

4.4.1 Accuracy……...……...51

4.4.2 Precision……...51

4.4.3 Recall……...…...51

4.4.4 F1 Score……...52

4.4.5 Mean Absolute Error……...52

4.4.6 ROC – AUC Curve……..52

4.4.7 Confusion Matrix……....52-53

Chapter 5: RESULTS AND ANALYSIS……..54

5.1 Feature Selection Algorithm Results……...54-58

5.1.1 Union and Intersection of Features from Feature Selection Algorithm……58-59

5.2 Results of Classification Algorithm……...59

5.2.1 XGBoost Classifier……...59-62

5.2.2 KNN Classifier……..62-64

5.2.3 SVM Classifier……..…65-67

5.3 Ensemble Learning Model……68-70

Chapter 6: CONCLUSION AND FUTURE SCOPE…….…...71

6.1 Conclusion of the Present Work……71

6.2 Scope for Future Work……...71

BIBLIOGRAPHY

LIST OF FIGURES

1.1 AQI Values and Category of Classes.....3

3.1 Flowchart of the Proposed Methodology..... 21

3.2 Scatter Plot of Pollutants PM2.5, PM10, NH3, CO..... 22-24

3.3 Prey Hunting Strategy of GWO.....32

3.4 Graphical Representation of Positive, Negative and No Correlation Between Features 33

3.5. Working Principle of SMOTE by Generating Synthetic Samples…….……..35

4.1. Descriptive Statistics of Features.....44

4.2. Percentage of Missing Values…….....45

4.3. KDE Plot…….....47

5.1.ROC Curve for NMIFS with 8 Features using XGBoost.....60

5.2. Confusion Matrix for 12 Features using XGBoost.....61

5.3. Confusion Matrix for NMIFS with 8 Features using XGBoost.....61

5.4. Confusion Matrix of GWO Features using XGBoost.....61

5.5. Confusion Matrix of Pearson Correlation using XGBoost.....62

5.6. ROC Curve for DMIFS with 8 Features using KNN.....63

5.7. Confusion Matrix of 12 Features using KNN.....63

5.8 Confusion Matrix of DMIFS with 8 Features using KNN.....64

5.9. Confusion Matrix of GWO Features using KNN.....64

5.10. Confusion Matrix of Pearson Correlation Features using KNN.....64

5.11. ROC Curve for NMIFS with 8 Features using SVM.....66

5.12. Confusion Matrix of 12 Features using SVM.....66

5.13. Confusion Matrix of NMIFS with 8 Features using SVM.....67

5.14. Confusion Matrix with GWO Features using SVM.....67

5.15. Confusion Matrix of Pearson Correlation Coefficient using SVM.....67

5.16. Confusion Matrix of Union of 6 Features using Ensemble Model.....69

5.17. Confusion Matrix of Union of 8 Features using Ensemble Model.....69

5.18. Confusion Matrix of Intersection of 6 Features using Ensemble Model.....69

5.19. Confusion Matrix of Intersection of 8 Features using Ensemble Model 70

LIST OF TABLES

4.1. Hyperparameters of XGBoost Classifier.....48

4.2. Hyperparameters of KNN Classifier..... 49

4.3. Hyperparameters of SVM Classifier..... 49

4.4. Hyperparameters of Ensemble Model Classifier..... 50

5.1. Selected Features using MIFS.....54

5.2. Selected Features using DMIFS..... 54

5.3. Selected Features using NMIFS..... 55

5.4. Selected Principal Components using PCA..... 55

5.5. Selected Features using GWO..... 56

5.6. Selected Features using Pearson Correlation Coefficient..... 56

5.7. Pearson Correlation Coefficient Values..... 57

5.8. Union of 6 Features..... 58

5.9. Union of 8 Features..... 58

5.10. Intersection of 6 Features..... 59

5.11. Intersection of 8 Features..... 59

5.12. Test Accuracy and Test MAE of XGBoost Classifier..... 59

5.13. Test Accuracy and Test MAE of KNN Classifier..... 62

5.14. Test Accuracy and Test MAE of SVM Classifier..... 65

5.15. Ensemble Model Results on Union and Intersection of Features..... 68

LIST OF ABBREVIATIONS

1. AQI - Air Quality Index
2. MI – Mutual Information
3. NI – Normalized Mutual Information
4. MIFS – Mutual Information Based Feature Selection
5. NMIFS – Normalized Mutual Information Based Feature Selection
6. DMIFS – Weighted Mutual Information Based Feature Selection and Distance Based Elimination
7. PCA – Principal Component Analysis
8. SMOTE – Synthetic Minority Oversampling Technique
9. GWO – Grey Wolf Optimization
10. XGBoost – Extreme Gradient Boosting
11. KNN – K-Nearest Neighbors
12. SVM – Support Vector Machine
13. MAE – Mean Absolute Error
14. ROC – AUC Curve – Receiver Operator Characteristics – Area Under the Curve

CHAPTER 1

INTRODUCTION AND BACKGROUND

1.1 Air Pollution and Air Quality Index

Air pollution [1] is a significant environmental issue that affects the quality of air we breathe, leading to various adverse health effects and environmental degradation. It is caused by the presence of harmful substances in the atmosphere, including particulate matter (PM), particularly $PM_{2.5}$ and PM_{10}, nitrogen oxides (NO_x), sulfur dioxide (SO_2), carbon monoxide (CO), ozone (O_3), and volatile organic compounds (VOCs). These pollutants can originate from various sources, such as industrial emissions, vehicle exhaust, agricultural activities, and natural events like wildfires and volcanic eruptions.

The Air Quality Index (AQI) is a standardized tool used to communicate how polluted the air currently is or how polluted it is forecast to become. The AQI converts complex air quality data of various pollutants into a single number, ranging from 0 to 500, making it easier for the public to understand the level of air pollution and its associated health risks.

1.1.1 Impact of Air Pollution

Air pollution is hazardous to both the environment and human health. Its impacts include:

- **Health Effects**: Exposure to high levels of air pollution can lead to respiratory diseases, cardiovascular diseases, adverse pregnancy outcomes, and even premature death. Pollutants like PM2.5 and ozone can penetrate deep into the lungs and enter the bloodstream, causing systemic inflammation and other health issues.
- **Environmental Damage**: Air pollution can cause acid rain, which harms soil, water bodies, and vegetation. It also contributes to the formation of ground-level ozone, which is detrimental to plant life and can reduce agricultural productivity.
- **Climate Change**: Certain air pollutants, such as black carbon and methane, are potent climate forcers. They contribute to global warming and climate change by trapping heat in the atmosphere.
- **Economic Costs**: The health impacts and environmental damage caused by air pollution leads to significant economic costs, including healthcare expenses, loss of productivity, and damage to infrastructure and ecosystems.

1.1.2 Calculation of Air Quality Index

The Air Quality Index (AQI) [2] is calculated based on the concentration levels of key pollutants such as $PM_{2.5}$, PM_{10}, NO_x, SO_2, CO, and O_3. The calculation involves the following steps:

- **Measurement of Pollutant Concentrations**: The concentrations of individual pollutants are measured using air quality monitoring stations.
- **Sub-Index Calculation**: For each pollutant, a sub-index is calculated using specific breakpoint concentration values. These breakpoints define the concentration ranges for different AQI categories.
- **Maximum Sub-Index Determination**: The highest sub-index value among the pollutants is selected as the overall AQI value for that location.

The AQI calculation can be mathematically represented as:

AQI = max (SI $PM_{2.5}$, SI PM_{10}, SI NO_2, SI SO_2, SI CO, SI O_3)

where SI denotes the sub-index for each pollutant.

1.1.3 Classes of Air Quality Index

The AQI is divided into six categories, each representing a different level of health concern:

- **Good (0-50)**: Air quality is considered satisfactory, and air pollution poses little or no risk.

- **Moderate (51-100)**: Air quality is acceptable; however, some pollutants may pose a moderate health concern for a very small number of people who are unusually sensitive to air pollution.

- **Satisfactory (101-150)**: Members of sensitive groups may experience health effects, but the general public is unlikely to be affected.

- **Poor (151-200)**: Everyone may begin to experience health effects, and members of sensitive groups may experience more serious health effects.

- **Very Poor (201-300)**: Health alert; everyone may experience more serious health effects.

- **Hazardous or Severe (301-500 and upwards)**: Health warnings of emergency conditions; the entire population is more likely to be affected.

Figure 1.1 shown below demonstrates a pictorial representation of AQI values and the category of classes pertaining to the values.

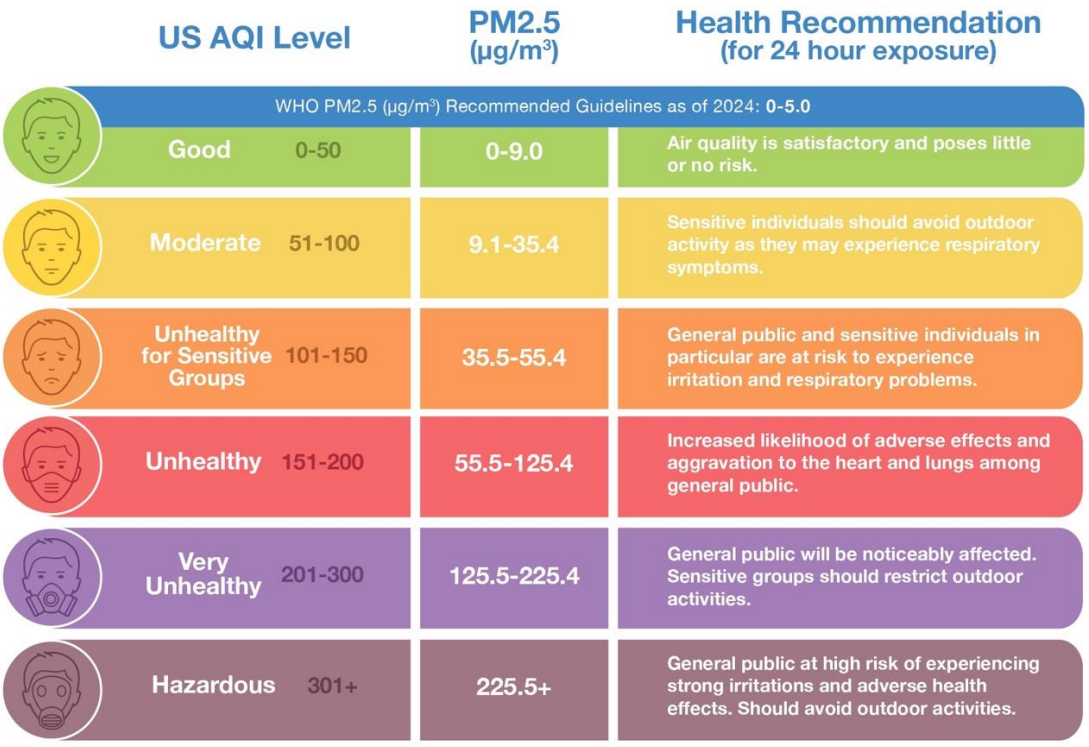


Figure 1.1. AQI Values and Category of Classes

(Source:https://cdn.shopify.com/s/files/1/0065/4780/0182/files/2236_WHO_Guidlines_Chart_UPDATE_AQI_2024.jpg?v=1714752885)

1.1.4 Necessity of Calculating the Air Quality Index

The calculation and dissemination of AQI are crucial for several reasons:

- **Public Awareness**: The AQI provides a clear and easily understandable measure of air quality, enabling individuals to take precautionary measures to protect their health.

- **Health Advisories**: Based on AQI values, health advisories can be issued, particularly for vulnerable populations, such as children, elderly, and those with respiratory or cardiovascular conditions.

- **Policy Making**: The AQI data helps policymakers and environmental agencies to formulate and implement strategies to control and reduce air pollution levels.

- **Environmental Monitoring**: Continuous monitoring of air quality through the AQI helps in assessing the effectiveness of air pollution control measures and identifying pollution trends.

1.2 Machine Learning

Machine learning (ML) [1] is a subfield of artificial intelligence (AI) that focuses on the development of algorithms and statistical models that enable computers to perform tasks without explicit instructions. It focuses on using data and algorithms to enable AI to imitate the way that humans learn, over time by identifying patterns, making decisions, and predicting outcomes thereby gradually improving its accuracy.

The learning system of a machine learning algorithm can be divided into three main parts:

- **A Decision Process:** In general, machine learning algorithms are used to make a prediction or classification. Based on some input data, which can be labelled or unlabelled, an algorithm will produce an estimate about a pattern in the data.
- **An Error Function:** An error function evaluates the prediction of the model. If there are known examples, an error function can make a comparison to assess the accuracy of the model.
- **A Model Optimization Process:** If the model can fit better to the data points in the training set, then weights are adjusted to reduce the discrepancy between the known example and the model estimate. The algorithm will repeat this iterative "evaluate and optimize" process, updating weights autonomously until a threshold of accuracy has been met.

1.2.1 Uses of Machine Learning

Machine learning is used across a wide array of applications, including but not limited to:

- **Predictive Analytics**: Forecasting future trends and behaviours based on historical data. For example, predicting stock market trends, sales forecasts, and weather conditions.
- **Natural Language Processing (NLP)**: Enabling computers to understand, interpret, and respond to human language. Applications include chatbots, language translation, and sentiment analysis.
- **Computer Vision**: Interpreting and understanding visual information from the world, such as facial recognition, object detection, and medical imaging.
- **Recommendation Systems**: Providing personalized recommendations based on user behaviour and preferences, widely used in e-commerce, streaming services, and social media platforms.
- **Anomaly Detection**: Identifying unusual patterns that do not conform to expected behaviour, crucial for fraud detection, network security, and fault detection in industrial systems.
- **Automation and Robotics**: Enhancing the capabilities of robots and autonomous systems, such as self-driving cars, drones, and automated manufacturing processes.
- **Healthcare**: Assisting in diagnosis, treatment planning, and drug discovery by analysing medical data and patient records.
- **Finance**: Enhancing financial services through credit scoring, risk management, algorithmic trading, and personalized financial advice.

By leveraging large datasets and sophisticated algorithms, machine learning transforms data into actionable insights and automated decisions, driving innovation and efficiency across various industries.

1.2.2 Types of Machine Learning

Machine learning can be broadly categorized into three main types: supervised learning, unsupervised learning, and reinforcement learning. Each type has distinct characteristics and is suitable for different kinds of tasks.

1. Supervised Learning

Supervised learning involves training a machine learning model on a labelled dataset, which means that each training example is paired with an output label. The goal is to learn a mapping from inputs to outputs that can be used to predict the labels of new, unseen data.

Applications: Classification (e.g., spam detection, image recognition), regression (e.g., predicting house prices, stock market forecasting).

Example Algorithms:

- Linear Regression
- Logistic Regression
- Support Vector Machines (SVM)
- Decision Trees
- Random Forests
- Neural Networks

Key Characteristics:

- Requires labelled data.
- The performance is measured by comparing the predicted outputs to the true labels.

2. Unsupervised Learning

Unsupervised learning involves training a model on data that does not have labelled responses. The goal is to infer the natural structure present within a set of data points.

Applications: Clustering (e.g., customer segmentation, image segmentation), association (e.g., market basket analysis), dimensionality reduction (e.g., Principal Component Analysis).

Example Algorithms:

- K-Means Clustering
- Hierarchical Clustering
- DBSCAN (Density-Based Spatial Clustering of Applications with Noise)
- Principal Component Analysis (PCA)

- t-Distributed Stochastic Neighbour Embedding (t-SNE)
- Autoencoders

Key Characteristics:

- Does not require labelled data.
- The performance is often subjective and evaluated by the utility of the resulting data transformation.

3. Reinforcement Learning

Reinforcement learning involves training an agent to make a sequence of decisions by interacting with an environment. The agent learns to achieve a goal by receiving rewards or penalties based on the actions it performs.

Applications: Game playing (e.g., AlphaGo), robotics (e.g., robotic control), autonomous vehicles, resource management.

Example Algorithms:

- Q-Learning
- Deep Q-Networks (DQN)
- Proximal Policy Optimization (PPO)
- A3C (Asynchronous Advantage Actor-Critic)

Key Characteristics:

- Involves a feedback loop where the agent's actions influence subsequent data it receives.
- Performance is measured by the cumulative reward the agent achieves over time.

Deep learning networks, a subset of machine learning, have revolutionized the field by enabling computers to learn and make decisions from data in a manner that mimics the human brain. These networks are characterized by their depth, as they consist of multiple layers of

interconnected nodes, or artificial neurons, that process and transform input data to produce desired output.

1.2.3 Key Components of Supervised Machine Learning

Supervised machine learning involves training a model on a labelled dataset, where the input data (features) and the corresponding output (labels) are known. The goal is for the algorithm to learn a mapping from inputs to outputs that can be generalized to unseen data. The key components of supervised machine learning algorithms include:

1. **Training Data:**
    - Features (X): These are the input variables or independent variables. They represent the characteristics or attributes used to make predictions. Features are typically represented as vectors or matrices.
    - Labels (Y): These are the output variables or dependent variables. They represent the target values the model aims to predict. In a classification problem, labels are discrete classes, whereas, in a regression problem, labels are continuous values.
2. **Model:**
    - The model is the mathematical representation or function that maps inputs (features) to outputs (labels). Common models include linear regression, decision trees, support vector machines, and neural networks.
3. **Loss Function:**
    - The loss function, also known as the cost function or objective function, quantifies the difference between the predicted values and the actual values. It guides the optimization process to adjust the model's parameters. Common loss functions include Mean Squared Error (MSE) for regression and Cross-Entropy Loss for classification.
4. **Optimization Algorithm:**
    - The optimization algorithm minimizes the loss function by adjusting the model's parameters. Gradient Descent is a popular optimization algorithm used in supervised learning. Variants include Stochastic Gradient Descent (SGD) and Mini-Batch Gradient Descent.
5. **Hyperparameters:**
    - Hyperparameters are the external configurations of the model that are not learned from the training data. Examples include learning rate, number of epochs, batch size, and regularization parameters. Hyperparameters are often tuned to improve model performance.

6. **Training Process:**
    - During training, the model learns the optimal parameters by iteratively updating them to minimize the loss function. The dataset is divided into training and validation sets to evaluate the model's performance and prevent overfitting.
7. **Validation and Testing:**
    - After training, the model's performance is evaluated on a validation set to fine-tune hyperparameters and ensure it generalizes well to unseen data. The final model is then tested on a separate test set to assess its accuracy and generalization ability.
8. **Performance Metrics:**
    - Performance metrics quantify the model's effectiveness. Common metrics include accuracy, precision, recall, F1 score for classification, and Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared for regression.

1.2.4 Challenges of Building a Machine Learning Model

Overfitting is a common challenge in machine learning, occurring when a model learns the training data too closely, capturing noise and specific patterns that may not generalize well to new, unseen data. Several methods are employed to mitigate overfitting and enhance a model's ability to generalize:

- **Regularization Techniques:** Regularization methods add penalties to the model's learning process, discouraging it from fitting the training data too closely. L1 and L2 regularization introduce penalties based on the absolute and squared values of model weights, respectively, while elastic net combines both.

- **Dropout:** Commonly used in neural networks, dropout randomly deactivates neurons during training, preventing the model from relying too heavily on specific features. This regularization technique encourages the network to learn more robust representations.

- **Data Augmentation:** Data augmentation artificially expands the training dataset by applying transformations like rotation, scaling, and flipping to existing data. This exposes the model to a broader range of variations, promoting generalization.

- **Cross-Validation:** Cross-validation assesses a model's performance on multiple subsets of the training data. Techniques like k-fold cross-validation split the data into folds, training the model on different subsets and evaluating its performance on others to ensure consistency.

- **Early Stopping:** Early stopping monitors the model's performance on a validation set during training and halts the process when performance starts to degrade. This prevents the model from learning noise in the training data and helps find an optimal point before overfitting occurs.

1.2.5 Importance of Machine Learning in the Classification of Air Quality Index

Traditional methods of AQI Bucket classification [3] rely heavily on human observation, which is time-consuming and prone to errors. As a result, there is a pressing demand for the development of automated technologies that are highly precise, efficient, and capable of accurately classifying the various classes of AQI. Traditional machine learning approaches have proved its finesse in learning the complex characteristics of air quality. However, since there are multiple variables that can govern the quality of air thereby exhibiting intricate patterns and dynamics, it is extremely important to identify the most pressing features that can lead to an optimal classification accuracy. These optimal features can be used to train a classification model that can also exhibit a remarkable adaptability to any set of real-world data.

1.3 Feature Selection in Machine Learning

Feature selection [4] is a crucial process in machine learning that involves selecting a subset of relevant features (variables, predictors) from the original set of features to use in model construction. The primary goal of feature selection is to improve the performance of machine learning models by eliminating redundant or irrelevant features, thereby reducing the dimensionality of the data. This process helps in enhancing the model's accuracy, reducing overfitting, speeding up the training process, and making the model more interpretable.

1.3.1 Importance of Feature Selection

- Improved Model Performance: By removing noise and irrelevant features, the model can focus on the most informative aspects of the data, leading to better performance.
- Reduced Overfitting: With fewer features, the model complexity is reduced, which can help prevent overfitting.
- Enhanced Interpretability: Simplifying the model by using fewer features makes it easier to interpret and understand.
- Reduced Computational Cost: Lower dimensionality reduces the computational burden, making the model training and prediction faster.

1.3.2 Types of Feature Selection

Feature selection methods [4-5] can be broadly categorized into three types: Filter methods, Wrapper methods, and Embedded methods. These methods along with a few examples are described below in brief:

1. Filter Methods:

Definition: Filter methods evaluate the relevance of features by their intrinsic properties, without involving any machine learning algorithms. They are typically used as a preprocessing step.

Examples:

- Correlation Coefficient: Measures the linear relationship between features and the target variable.
- Chi-Square Test: Evaluates the dependence between categorical features and the target variable.
- Mutual Information: Measures the amount of information obtained about one variable through another variable.

2. Wrapper Methods:

Definition: Wrapper methods evaluate the usefulness of a subset of features by actually training a model and measuring its performance. They involve searching for the best subset of features.

Examples:

- Forward Selection: Starts with an empty set and adds features one by one, based on the model performance improvement.
- Backward Elimination: Starts with all features and removes them one by one, based on the least significant feature.
- Recursive Feature Elimination (RFE): Iteratively builds models and eliminates the least important feature at each iteration.

3. Embedded Methods:

Definition: Embedded methods perform feature selection during the model training process. They integrate feature selection with the learning algorithm itself.

Examples:

- Lasso Regression: Uses L1 regularization to penalize the absolute size of the coefficients, effectively shrinking some of them to zero.

- Tree-Based Methods: Decision trees and random forests naturally perform feature selection based on feature importance.

1.4 Data Augmentation in Machine Learning

Data augmentation is a technique used in machine learning to increase the diversity and volume of training data without collecting new data. This is particularly useful in scenarios where the amount of available data is limited, which can lead to overfitting of models to the training data, thereby reducing their generalization capability. Data augmentation helps to mitigate this problem by artificially expanding the dataset.

1.4.1 Types of Data Augmentation

1. **Geometric Transformations:**
    - Rotation: Rotating images by a certain degree.
    - Translation: Shifting images horizontally or vertically.
    - Scaling: Zooming in or out of images.
    - Flipping: Flipping images horizontally or vertically.
2. **Colour and Brightness Adjustments:**
    - Brightness Adjustment: Changing the brightness of images.
    - Contrast Adjustment: Altering the contrast of images.
    - Colour Jittering: Randomly changing the colour balance of images.
3. **Noise Injection:**
    - Gaussian Noise: Adding random noise to images.
    - Salt and Pepper Noise: Adding white and black pixels randomly to images.
4. **Cropping and Padding:**
    - Random Cropping: Randomly selecting a part of the image.
    - Padding: Adding pixels around the image to change its dimensions.
5. **Synthetic Data Generation:**
    - SMOTE (Synthetic Minority Over-sampling Technique): Generating synthetic examples for minority classes in imbalanced datasets.

- GANs (Generative Adversarial Networks): Creating entirely new synthetic data points.

6. **Data Augmentation in Text Data:**
   - Synonym Replacement: Replacing words with their synonyms.
   - Random Insertion: Inserting random words into text.
   - Random Swap: Swapping the positions of random words.
   - Random Deletion: Removing random words from text.
7. **Data Augmentation in Time-Series Data:**
   - Time Warping: Stretching or compressing time-series data.
   - Jittering: Adding noise to time-series data.
   - Permutation: Randomly shuffling sections of time-series data.

1.4.2 Importance of Data Augmentation

1. **Improved Model Generalization:**
   - By exposing the model to a more varied set of training examples, data augmentation helps improve the model's ability to generalize to unseen data.
2. **Preventing Overfitting:**
   - Augmented data helps to prevent overfitting by providing more training examples, making it harder for the model to memorize the training data.
3. **Leveraging Limited Data:**
   - In scenarios where collecting more data is impractical or expensive, data augmentation serves as an effective strategy to enhance the dataset.
4. **Balancing Imbalanced Datasets:**
   - Techniques like SMOTE help to balance datasets where certain classes are underrepresented, thus improving the performance of classifiers on minority classes.

1.5 Ensemble Learning Techniques

Ensemble techniques [4] in machine learning are a powerful strategy employed to harness the collective intelligence of multiple individual models. By amalgamating the predictions of

diverse models, ensemble methods aim to overcome the limitations of any single model and attain a more accurate and robust final prediction.

The essence of ensemble learning lies in the acknowledgment that different models may excel at capturing distinct aspects of the underlying data distribution or learning different representations of the input space. Leveraging this diversity, ensemble techniques seek to aggregate the varied insights and perspectives offered by these models to arrive at a more comprehensive and reliable prediction.

1.5.1 Utility of Ensemble Learning

Ensemble learning [4] acts as a form of wisdom of the crowd, where the collective judgment of multiple models can often outperform that of any individual model. This is particularly beneficial in scenarios where the data may be noisy, complex, or subject to inherent variability, as it can help smooth out discrepancies and uncertainties present in individual predictions. It not only enhances the accuracy of predictions but also bolsters the robustness and generalization ability of the model. By combining predictions from multiple models trained on different subsets of the data or employing diverse architectures and hyperparameters, it mitigates the risk of overfitting to specific anomalies of the training data and promotes a more robust understanding of the underlying patterns in the data.

1.6 Objective and Contribution

In this thesis, our primary goal is to propose a novel approach for classifying various AQI Bucket classes. To achieve this, we leverage the power of statistical feature selection techniques, predominantly utilizing information theoretic concept of capturing classical correlations between features using mutual information. Our focus is on preprocessing the data to make it suitable for training our classification algorithms, and we aim to enhance classification accuracy by exploiting the capabilities of ensemble learning. To optimize the performance of our models, we engage in meticulous hyperparameter tuning, adjusting the configuration settings to find the most effective combination. Additionally, we employ data augmentation techniques to artificially diversify our training dataset. The train dataset is subjected to a number of simulations using techniques such as cross validation (CV), enhancing the model's ability to recognize different patterns and adapt well to unseen data.

CHAPTER 2
LITERATURE SURVEY

2.1 Related Research and Previous Works

There is a need to exploit the available data collected on environment for the development of cities to improve the air quality which in turn can improve the quality of life for a city. Air pollution is becoming a serious concern to the society as the air pollutants are very hazardous in nature. Pollutants affect the health and causes respiratory and cardiac problems. If air pollutants cross the limits, it might be life threatening. Software or tool can be developed whose results can be applied for predicting air pollution levels, predicting air pollution related health concerns, monitoring controlling air pollution and this is challenging one.

In 2018, Teologo Jr. A. T., Dadios E. P., Baldovino R. G., Neyra R. Q., and Javel I. M. [1] conducted a study on classifying the Air Quality Index (AQI) using a fuzzy logic system. This study specifically focused on two input pollutants, carbon monoxide (CO) and nitrogen dioxide (NO2), each classified into six categories ranging from very low to extremely high. The researchers employed a Mamdani fuzzy inference system (FIS) for processing, with simulations conducted using the MATLAB fuzzy logic toolbox. Their experiments achieved reliable results, demonstrating the effectiveness of the fuzzy logic approach in AQI classification, which categorizes air quality into six levels: good, moderate, unhealthy for sensitive groups, unhealthy, very unhealthy, and hazardous.

In their study, Suman et al. [2] (2020) analyzed various methodologies for calculating the Air Quality Index (AQI), focusing on its effectiveness in urban air pollution assessment. They employed multiple mathematical models to derive AQI, incorporating pollutant concentration data to provide a comprehensive air quality status. The primary methodology involved comparing observed pollutant levels with National Ambient Air Quality Standards (NAAQS) and calculating a quality rating for each pollutant, which was then geometrically averaged to produce the AQI. The researchers highlighted the robustness of the Central Pollution Control Board (CPCB) 2014 formula for accurate air quality categorization. This approach facilitated timely information dissemination to the public, aiding in effective pollution control measures. Through their experiments, they utilized classification algorithms to categorize the air quality into different levels (good, satisfactory, moderately polluted, poor, very poor, and severe). The accuracy of these classifications was evaluated by comparing the derived AQI with actual pollutant concentration data, demonstrating high reliability in predicting air quality status.

In 2021, Kekulanadara K. M. O. V. K., Kumara B. T. G. S., and Kuhaneswaran B. [3]conducted a study to predict the Air Quality Index (AQI) by considering major pollutants using various

machine learning algorithms. The researchers employed Decision Tree, Random Forest, Support Vector Machine (SVM), and Artificial Neural Network (ANN) algorithms to create prediction models. The dataset, after preprocessing and feature selection, consisted of 61,285 instances with 13 key features including PM2.5, PM10, NO, NO2, NOx, NH3, SO2, O3, CO, Benzene, Toluene, Xylene, and AQI_Bucket as the target variable. The models were evaluated using the K-Fold Cross-Validation method with K=10 to ensure accurate results. Among the algorithms tested, the Random Forest classification algorithm achieved the highest accuracy of 74.039%, followed closely by the ANN with an accuracy of 73.82%. The Decision Tree and SVM algorithms achieved accuracies of 64.943% and 61.345%, respectively .

In 2024, Shikha Gupta, Armaan Moledina, Soham Athavale, Sarang Gajare, Mihir Kate [6] conducted a comprehensive study comparing various machine learning algorithms for air quality prediction using data from the Central Pollution Control Board (CPCB). The study focused on three primary algorithms: K-Nearest Neighbor (KNN), Decision Tree, and Logistic Regression. Each algorithm was meticulously evaluated on its ability to predict air pollution levels based on historical air quality data including features such as sampling date, location, and pollutant levels (NO2, SO2, RSPM, SPM). The KNN algorithm demonstrated versatility and achieved a robust accuracy, with R-squared scores on both training and testing data. The Decision Tree classifier, known for its simplicity and interpretability, also showed strong performance with precise tuning and pruning of hyperparameters to prevent overfitting. Logistic Regression was effectively employed for binary classification of air quality conditions, yielding significant coefficients for environmental features impacting air quality prediction. This study concluded that while all three algorithms have their merits, the Decision Tree emerged as the most efficient in processing time and accuracy.

In 2024, Suresh Kumar Natarajan, Prakash Shanmurthy, Daniel Arockiam, Balamurugan Balusamy, and Shitharth Selvarajan [7] conducted a study aimed at predicting the Air Quality Index (AQI) for major cities in India using an optimized machine learning approach. They introduced the dataset of AQI of India [8] sourced from Central Pollution Control Board (CPCB) which was extracted from Kaggle repository. The researchers utilized a hybrid model combining Grey Wolf Optimization (GWO) with the Decision Tree (DT) regression algorithm to improve the prediction accuracy. They experimented with air quality data from cities including New Delhi, Kolkata and Bangalore. The proposed model was compared against traditional machine learning models such as k-nearest Neighbor, Random Forest regressor, and Support Vector regressor. The results showed that the GWO-DT model outperformed the conventional models, achieving the highest prediction accuracy of 94.48% for Kolkata, 91.49% for Bangalore, and 88.98% for New Delhi. The study demonstrated the effectiveness of the GWO-DT model in predicting AQI, making it a valuable tool for air quality monitoring and management in urban environments.

In 2023, K. Kumar and B.P. Pande [9] conducted a study focused on predicting air quality in Indian cities using machine learning methodologies. The researchers employed several classification algorithms, including K-Nearest Neighbors (KNN), Gaussian Naive Bayes (GNB), Support Vector Machine (SVM), Random Forest (RF), and XGBoost. Their study involved a detailed preprocessing of the data, including normalization and addressing data imbalance through the Synthetic Minority Oversampling Technique (SMOTE). Among the classification algorithms, the XGBoost model achieved the highest accuracy during both training and testing phases, surpassing other models with a maximum accuracy of 91% in the training phase and 90% in the testing phase. The SVM model, however, exhibited the lowest accuracy, with 81% during training and 78% during testing. The study highlighted the effectiveness of XGBoost in handling complex air quality data, making it the best-performing model in this context. This work significantly contributes to the field of air quality prediction, offering insights into the methodologies and machine learning models most effective for this application.

In 2023, Safaa Berkani, Ihsane Gryech, Mounir Ghogho, Bassma Guermah, and Abdellatif Kobbane [10] conducted a study comparing various data-driven forecasting models for urban air pollution using the MoreAir dataset. The MoreAir project, notable for being the first to establish a low-cost air pollution monitoring system in Morocco, provided a rich dataset for this analysis. The researchers evaluated multiple forecasting frameworks, incorporating exogenous data and spatial-temporal information, with a focus on short-term forecasting of particulate matter concentrations. Machine learning models, particularly LightGBM and CatBoost, demonstrated superior performance. LightGBM achieved a notable training accuracy, showcasing the efficacy of including spatial dimensions and diverse exogenous features in enhancing predictive performance. This study underscores the importance of advanced data-driven models in accurately forecasting urban air pollution, thereby contributing valuable insights for sustainable urban development.

In 2023, Chunhao Liu, Guangyuan Pan, Dongming Song, and Hao Wei [11] conducted a study on air quality index (AQI) forecasting using a genetic algorithm-based improved extreme learning machine (GA-KELM). The researchers addressed the challenges associated with single machine learning models in predicting air quality under various AQI fluctuation trends. They introduced a kernel method to enhance the extreme learning machine (ELM) and employed a genetic algorithm to optimize the number of hidden nodes, thresholds, and weights. This optimization aimed to improve the stability and accuracy of the predictions. The GA-KELM method was tested against CMAQ (Community Multiscale Air Quality), SVM (Support Vector Machines), and DBN-BP (Deep Belief Networks with Back-Propagation) models. The proposed model demonstrated faster training and more accurate predictions. The experimental results indicated that the GA-KELM algorithm achieved a reliable and superior performance in air quality prediction compared to the other methods used.

In 2022, Jingyang Wang, Lukai Jin, Xiaolei Li, Siyuan He, Min Huang, and Haiyao Wang [12] conducted a study on air quality index (AQI) prediction using a hybrid model combining Convolutional Neural Networks (CNN) and an Attention Gate Unit (AGU). The researchers aimed to address issues related to vanishing and exploding gradients in Recurrent Neural Networks (RNNs). Their novel approach embedded the attention mechanism and a Data Adjustment Module (DAM) within the gated unit, enhancing the model's sensitivity to historical data and improving its learning capacity. The CNN component was utilized to extract features from time series data, while the AGU handled differentiated learning of historical data to produce accurate AQI predictions. The model's performance was evaluated against nine other models using metrics such as Mean Absolute Error (MAE), Mean Square Error (MSE), and R-Squared (R^2). The CNN-AGU model outperformed the others, demonstrating superior prediction accuracy and robustness in handling complex air quality data.

In 2022, Qi Zhang, Yang Han, Victor O. K. Li, and Jacqueline C. K. Lam [13] conducted a study developing a hybrid CNN-LSTM framework called Deep-AIR for fine-grained air pollution estimation and forecasting in metropolitan cities. The study aimed to fill the gap in accurately capturing the complex interactions between temporal characteristics of air pollution and spatial characteristics of urban dynamics. The researchers employed a hybrid CNN-LSTM structure to capture spatial-temporal features, utilizing 1×1 convolution layers to enhance learning of temporal and spatial interactions. Deep-AIR outperformed compatible baselines with a higher accuracy of 1.5%, 2.7%, and 2.3% for Hong Kong and 1.4%, 1.4%, and 3.3% for Beijing in fine-grained 1-hour pollution estimation, and 1-hour and 24-hour forecasts, respectively. Saliency analysis revealed that for Hong Kong, spatial features such as street canyon and road density were the best predictors for NO2, while temporal features like historical air pollutants and weather were the best predictors for PM2.5. In Beijing, historical air pollutant data, traffic congestion, wind direction, and seasonal indicators were the best predictors for all pollutants.

In 2021, Hongqian Chen, Mengxi Guan, and Hui Li [14] conducted a study on air quality prediction using an integrated dual LSTM (Long Short-Term Memory) model. This research aimed to address the limitations of single model approaches by integrating both time and spatial dimensions of air quality data. The researchers first employed Seq2Seq (Sequence to Sequence) technology to establish a single-factor prediction model, which independently forecasted each component of air quality data as a time series. They then used an LSTM model with an attention mechanism for multi-factor prediction, incorporating data from neighboring stations and weather conditions. Finally, they integrated these two models using an XGBoost tree to combine the predicted values. The evaluation using five methods showed that the proposed model significantly outperformed traditional models in terms of error reduction and model expression power, achieving higher precision in air quality prediction.

In 2016, Ibtissam Banit'ouagua, Mounir Ait Kerroum, Ahmed Hammouch, and Driss Aboutajdine [15] conducted a study to improve hyperspectral image classification by employing band selection techniques using Mutual Information (MI). The researchers focused on four MI-based feature selection algorithms: MIFS, MIFS-U, MIFS-U2, and NMIFS. These algorithms were tested on the AVIRIS hyperspectral datasets, specifically the Indian Pines and Salinas datasets, to select the most informative spectral bands. For classification, they employed the Support Vector Machine (SVM) classifier, known for its robustness in hyperspectral image classification. The study found that the NMIFS algorithm outperformed other methods, achieving the highest classification accuracy. Specifically, for the Indian Pines dataset, NMIFS achieved an accuracy of 93.73% with 70 selected bands, while for the Salinas dataset, the highest accuracy achieved was 95.78% with 80 selected bands.

In 2005, Jana Novovicova and Antonin Malik [16] conducted a study on feature selection algorithms for text classification, focusing on improving the performance of classification algorithms by reducing the dimensionality of text data. They introduced four novel feature selection algorithms based on mutual information criteria, which were applied to text classification tasks. The algorithms were tested using the Reuters-21578 dataset and evaluated on classifiers such as Naive Bayes, Support Vector Machines (SVM), and k-nearest neighbor (KNN). The study found that the proposed Sequential Forward Selection (SFS) methods, particularly the SFS MaxIMI algorithm, achieved the highest classification accuracy compared to other traditional feature selection methods. Specifically, the SFS MaxIMI algorithm outperformed others in terms of precision, recall, and F1-measure, with the SVM classifier showing the best overall performance across different metrics.

In 2023, Abhishek Dey, Susmita Ghosh, and Emmett J. Ientilucci [17] conducted a study on improving the classification of hyperspectral images through an innovative band selection methodology. The researchers proposed a supervised band selection approach that combines Mutual Information (MI) and neighborhood information to effectively reduce the dimensionality of hyperspectral images, thereby optimizing the classification process. For the classification task, they utilized a Support Vector Machine (SVM) classifier with a Radial Basis Function (RBF) kernel. The study demonstrated that the proposed method achieved superior performance in terms of overall accuracy compared to several other MI-based methods. Specifically, the highest overall accuracy obtained for the Pavia University dataset was 95.09% with 30 selected bands, and for the Botswana dataset, it was 94.13% with 28 selected bands. These results indicate the robustness and effectiveness of the proposed method in hyperspectral image classification.

In their study on air quality prediction, Chakradhar Reddy K., Nagarjuna Reddy K., Brahmaji Prasad K., and Dr. P. Selvi Rajendran [18] employed various supervised machine learning algorithms to analyze air pollution data from different sources. The researchers aimed to develop an accurate machine learning-based method to forecast the Air Quality Index (AQI) by using classification models. They utilized algorithms such as Logistic Regression (LR), Naive Bayes (NB), Support Vector Machine (SVM), Random Forest (RF), K-Nearest Neighbors (K-NN), and Decision Tree (DT). Through their experiments, the Decision Tree algorithm achieved the highest accuracy, excelling in predicting air quality with a precision, recall, sensitivity, and specificity all at 100%. This study highlights the effectiveness of machine learning techniques in environmental monitoring and the potential for developing real-time air quality forecasting systems.

In 2017, Ranjana Waman Gore and Deepa S. Deshpande [19] conducted a study to classify health risks based on air quality levels using data mining techniques. They used a dataset containing AQI values for various pollutants like NO2, SO2, CO, and O3, downloaded from Kaggle. The researchers implemented Naive Bayes and Decision Tree J48 algorithms to predict health concerns related to air pollution levels. The Decision Tree J48 algorithm demonstrated superior performance, achieving an accuracy of 91.9978%, while the Naive Bayes algorithm achieved an accuracy of 86.663%. This study highlights the potential of data mining techniques in analyzing environmental data for predicting health risks and informing policy decisions for improving air quality.

In 2022, S. Abirami and P. Chitra [20] conducted a study on probabilistic air quality forecasting using a deep learning spatial–temporal neural network (DL-STNN) for the Delhi region. The study aimed to address the limitations of deterministic models by integrating uncertainty quantification techniques to produce probabilistic forecasts. The researchers developed a deterministic DL-STNN model for spatial–temporal air quality forecasting and further enhanced it by incorporating various uncertainty quantification methods, including Monte-Carlo Dropout, Ensemble Averaging, Gaussian Process Regression, Quantile Regression, and Bayesian Inference. The evaluation of these models on real-world air quality data from Delhi demonstrated that the deterministic DL-STNN model achieved 39.8% more accurate predictions compared to baseline models. Among the probabilistic models, the Bayesian DL-STNN model stood out, providing 13% more accurate point forecasts and a 20% higher suitability score for real-time applications. This study highlights the effectiveness of combining deep learning with probabilistic approaches to enhance the reliability and accuracy of air quality predictions.

2.2 Comparative Analysis of Previous Research

Various methods for AQI classification are compared based on their performance metrics using the AQI [8] dataset. The dataset consists of the cities and date against which all the pollutants have been recorded followed by the AQI value. The objective is to accurately classify the AQI Bucket into its subsequent class. Several state-of-the-art methods have been employed, and their performance is evaluated in terms of accuracy, precision, recall, F1 score as well as the Mean Absolute Error (MAE) score.

These results provide a comprehensive overview of the performance of different methods of AQI classification, highlighting the strengths and effectiveness of each approach. The metrics indicate each models' abilities to correctly classify the AQI, with XGBoost model emerging as the top-performing classifier in terms of accuracy on the AQI dataset. The ensemble learning model consisting of three individual base learners exhibits the overall highest accuracy of the AQI dataset.

CHAPTER 3
PROPOSED METHODOLOGY

3.1 Introduction of Proposed Methodology

In this thesis, the main motivation is to present a feature selection technique for classification of Air Quality Index (AQI) [2] of India, utilizing mutual information-based feature selection algorithm and its variants and we also employ ensemble learning techniques. Targeting 6 distinct AQI Bucket classes, we harness the capabilities of mutual information and data augmentation along with meticulous hyperparameter tuning, to build traditional machine learning models to improve classification accuracy for the AQI Dataset [8]. Optimized ensemble learning techniques are also utilized with an aim to produce better adaptability and achieve high accuracy.

The methodology employed in this research includes data preprocessing methods such as handling missing values, encoding categorical variables, scaling the data, addressing class imbalance using data augmentation techniques, and selecting the most relevant features for further data analysis.

In this Chapter, we introduce the subsections where we describe the methods used for data preprocessing, the feature selection strategies that have been adopted, data augmentation techniques to handle class imbalance and dimensionality reduction of the features using Principal Component Analysis (PCA), followed by a brief overview of the classification algorithms used. Additionally, we also expound on the layout of an ensemble learning model.

Figure 3.1 given below depicts the flowchart of the proposed model.

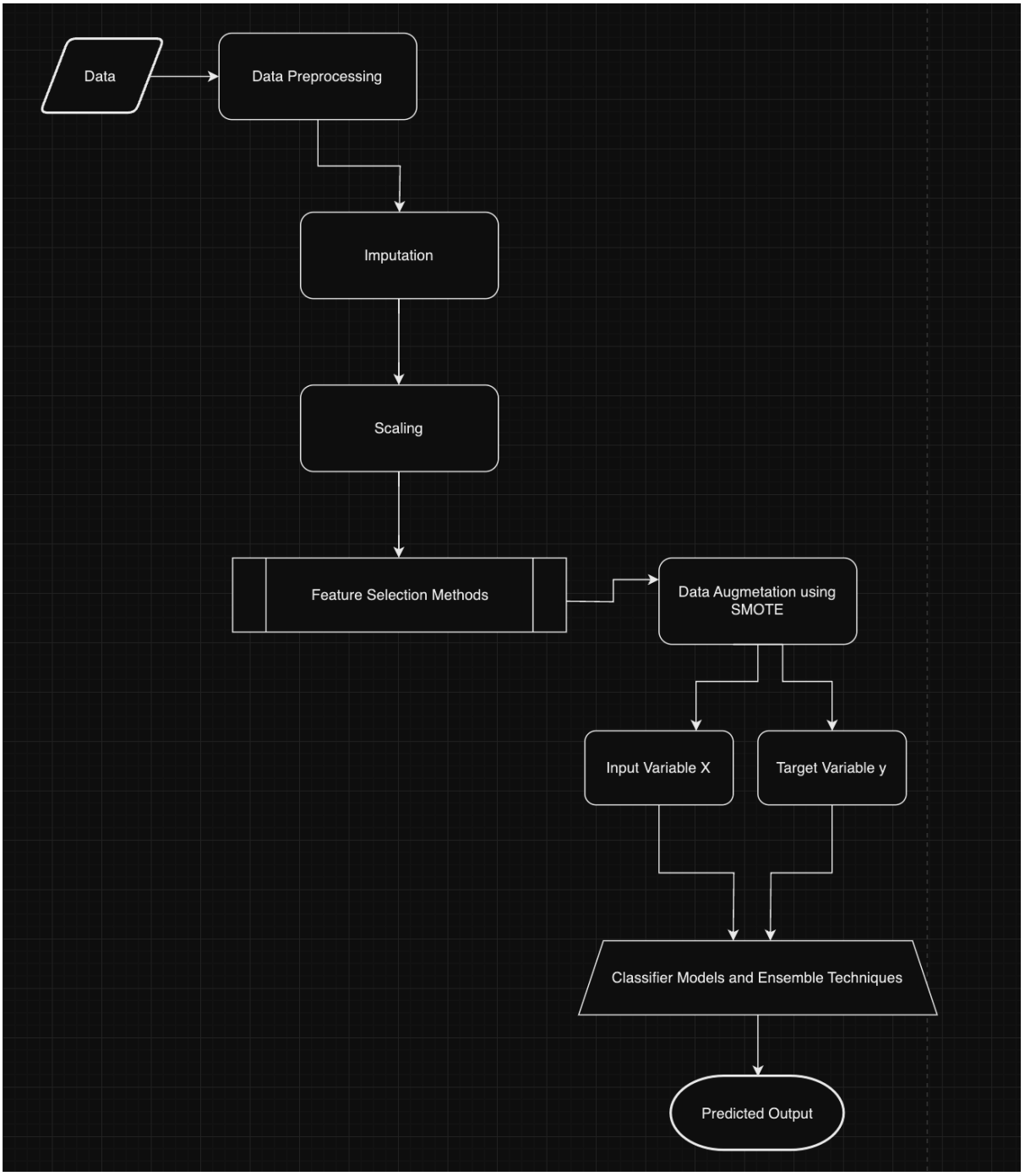


Figure 3.1. Flowchart of the Proposed Methodology.

3.2 Data Preprocessing

The dataset [8] is first checked for missing values and an imputation strategy [5] is employed to address the null value removal. Imputation is the process of replacing missing data with substituted values. Further, an encoding technique using target label encoding is used to convert the categorical features into discrete numerical data. The 'Date' column is converted to 'Datetime' format to ease further analysis. The features are scaled using Maximal Absolute (MaxAb) Scalar so that all the features have an equal contribution during model training. Figure 3.2 shown below depicts the scatter plot of a few pollutants, namely Particulate Matter ($PM_{2.5}, PM_{10}$), Ammonia (NH_3) and Carbon Monoxide (CO).

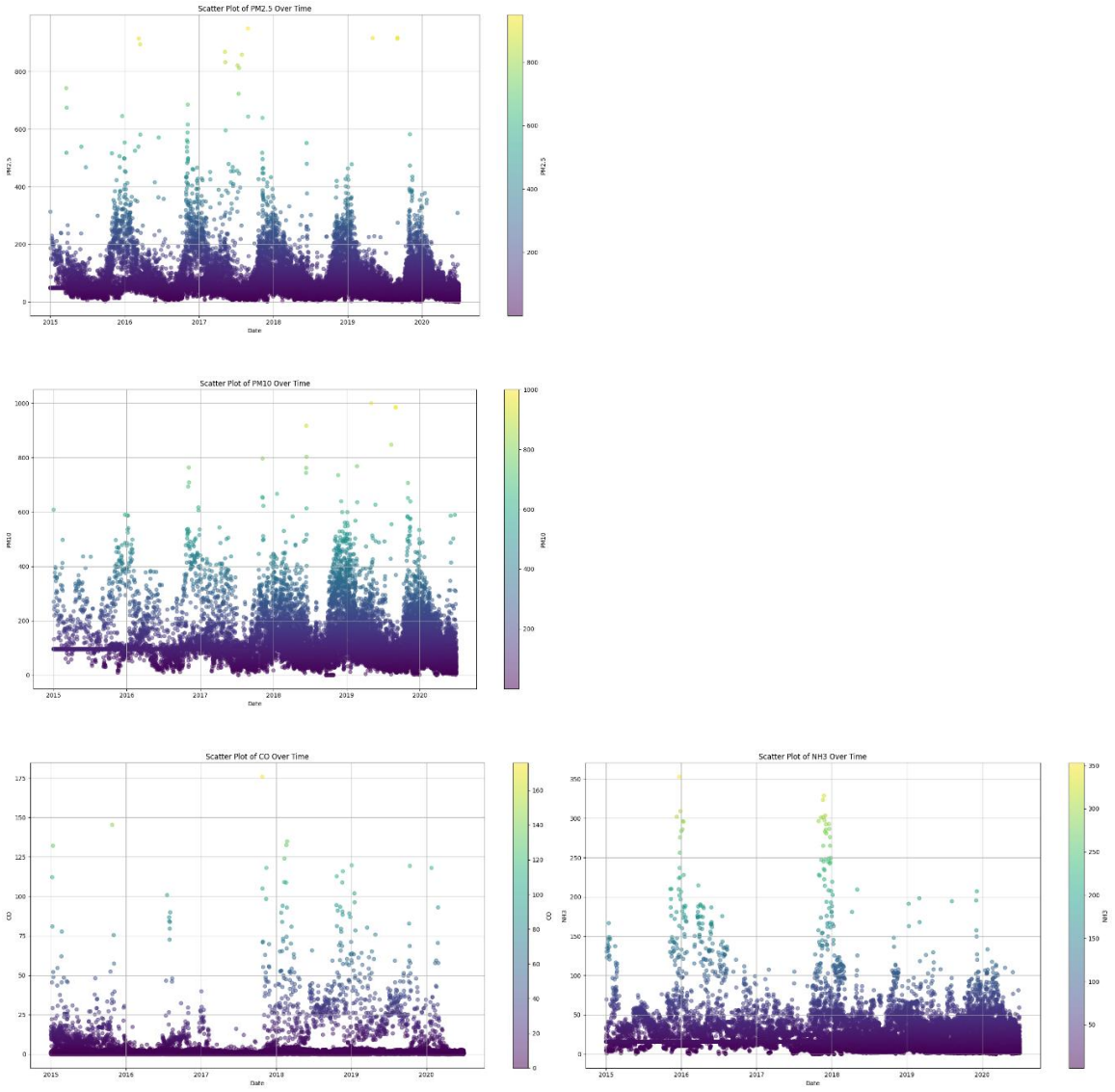


Figure 3.2. Scatter Plot of Pollutants PM2.5, PM10, NH3, CO.

3.3 Feature Selection

Feature selection is a crucial step in the preprocessing phase, which identifies a subset of the most relevant features from the set of all features. Feature selection improves model performance, increases training efficiency and handles multicollinearity.

There are several types of feature selection methods, each with unique characteristics and use cases. A few examples of such methods are Mutual Information (MI) based feature selection, dimensionality reduction techniques such as Principal Component Analysis (PCA), metaheuristic algorithms such as Grey Wolf Optimization (GWO) Algorithm and statistical evaluation of Pearson Correlation Coefficient based feature selection.

A brief description of the types of feature selection techniques used in machine learning are given below.

3.3.1 Filter Methods

Filter methods evaluate the relevance of features by examining their intrinsic properties and their relationship with the target variable. These methods are computationally efficient and independent of any machine learning algorithm. Common filter methods include:

- **Mutual Information Based Feature Selection (MIFS):** This method selects features based on the mutual information between each feature and the target variable, aiming to maximize the dependency between them.

- **Normalized Mutual Information Based Feature Selection (NMIFS):** This is an extension of MIFS that normalizes the mutual information score, making it more robust to different scales of data.

- **Pearson Correlation Coefficient**: Measures the linear relationship between a feature and the target variable. It is also known as the Pearson Correlation Coefficient and it is given by:

  Pearson correlation coefficient = $\frac{cov\ (X,Y)}{\sigma x \sigma y}$

  Where X and Y are the input and target variables respectively and cov is the covariance between them.

3.3.2 Wrapper Methods

Wrapper methods evaluate feature subsets based on the performance of a specific machine learning algorithm. These methods involve training and validating models on different subsets of features and selecting the one that yields the best performance. Examples include:

- **Recursive Feature Elimination (RFE):** This method recursively removes the least important features based on model performance until the optimal subset is achieved.

- **Grey Wolf Optimization (GWO):** This metaheuristic algorithm mimics the leadership hierarchy and hunting mechanism of grey wolves to search for the best feature subset that maximizes model performance.

3.3.3 Embedded Methods

Embedded methods perform feature selection during the model training process. These methods are specific to a given learning algorithm and integrate feature selection directly into the model building process. Examples include:

- **Lasso Regression:** This linear model includes a penalty term that shrinks the coefficients of less important features to zero, effectively performing feature selection.

- **Tree-Based Methods:** Algorithms like Random Forests and Gradient Boosting Trees can be used for feature selection by evaluating the importance of features based on how they split the data.

3.3.4 Dimensionality Reduction Methods

Principal Component Analysis (PCA) [5] is a dimensionality reduction technique that transforms the original features into a new set of orthogonal features called principal components. These components capture the maximum variance in the data, and a subset of the top components can be selected to reduce the feature space while retaining most of the information.

3.3.5 Hybrid Methods

Hybrid methods combine multiple feature selection techniques to leverage their strengths and mitigate their weaknesses. These methods can provide a more robust feature selection process. Examples include:

- **Union of Features:** This method combines features selected by different selection techniques to create a comprehensive feature set.

- **Intersection of Features:** This method identifies common features selected by multiple techniques, ensuring that only the most consistently important features are retained.

3.4 Mutual Information and Entropy

In this section, the concept of Mutual Information (MI) and entropy [21] in the light of statistical methods of feature selection are introduced.

In probability and information theory, Mutual Information (MI) [21-23] is a measure of the mutual dependence between the two variables. More specifically, it quantifies the amount of information obtained about one random variable by observing another random variable. Independence between random variables can be measured by MI. It is essentially a metric that quantifies how similar or how different two random variables are. Mutual Information leads to the reduction of uncertainty of a random variable if another is known. Therefore, a high mutual information value indicates a large reduction of uncertainty whereas a low value indicates a small reduction.

Entropy [21] is considered as the related amount of information of a random variable. In information theory, Shannon entropy, denoted by H(X) is a measure of information content of any random variable X.

In particular, if a random variable X takes on values in a set $\mathcal{X}$= $\{x_1, x_2, ..., x_n\}$, and is defined by a probability distribution P(X), then we will write the entropy of the random variable X as:

$$H(X) = -\sum_{x \in \mathcal{X}} P(x)\ log\ P(x) \tag{1}$$

where H(X) is the Shannon entropy, $x_1, x_2, ..., x_n$ are the set of distinct values contained within the set $\mathcal{X}$ and p(x) is the probability of the values that is contained within the set $\mathcal{X}$.

The Shannon entropy H(X) can also be expressed as,

$$H(X) = \sum_{x \in \Omega} p(x) log\ p(x) \tag{2}$$

where Ω represents event space and p(x) represents the mass probability of an event x ε Ω from a finite set of possible values.

If the log in the right-hand side of the above eq. (2) is taken to be to the base 2, then the entropy is expressed in bits.

In the context of feature selection in machine learning, mutual information can be used to assess the relevance of each feature with respect to the target variable.

The mutual information I (X; Y) between two random variables X and Y is defined as:

$$I(X;Y) = \sum_{x\in X}\sum_{y\in Y} p(x,y)\log\frac{p(x,y)}{p(x).p(y)} \quad (3)$$

where p (x, y) is the joint probability distribution function of X and Y, and p(x) and p(y) are the marginal probability distribution functions of X and Y respectively.

The mutual information I (X; Y); is the gain of information (IG) by which the uncertainty about one variable is decreased by the given knowledge of the second variable. MI and entropy H(X) have the following relation:

$$MI(X, Y) = H(X) + H(Y) - H(X, Y) \quad (4)$$

where H(X) is the Shannon Entropy of input variable X and H(Y) is the Shannon Entropy of target variable Y.

3.4.1 Mutual Information Based Feature Selection

The Mutual Information Based Feature Selection (MIFS) algorithm [21-22] can be formulated as follows:

1. (Initialization) Set F 'initial set of |F| input features and set S 'empty set'.

2. (Computation of MI with the output class) For each $f_c \in$ F, compute I(f_c;C) where I is the mutual information, f_c is a feature candidate, C is the target class.

3. (Choice of the first feature) Find the feature that maximizes I(f_c;C), set F ← F $\{f_c\}$, set S $\{f_c\}$

4. (Greedy selection) Repeat until |S| = k where k is the number of features generally controlled by the user.

a. (Computation of MI between features) for all couples of features (f_c; f_s) with $f_c \in$ F and

$f_s \in$ S compute I (f_c; f_s), if it is not already available.

b. (Selection of the next feature) choose feature f_c as the one that maximizes

$I(f_c; C) - \beta \sum_{f_s \in S} I(f_s; f_c)$; set F $\leftarrow$ F $\{f_c\}$; set S $\leftarrow$ S $\cup$ $\{f_c\}$

5. Output the set S containing the selected features.

For the features to be selected, each feature f_c must optimize the following expression:

$$I(f_c; C) - \beta \sum_{f_s \in S} I(f_s; f_c) \quad (5)$$

The right-hand side of equation (4) represents the redundant information between input candidate features and the already selected features in S. The parameter β is a user-defined parameter that regulates the relative importance of the redundancy between the candidate band and the set of selected features. Using β = 0 would be equivalent to selecting features independently, while a larger value will place more emphasis on reducing inter-feature dependencies. Generally, the optimal value of β lies between 0.5 and 1.

3.4.2 Normalized Mutual Information Based Feature Selection

Normalized Mutual Information (NI) [22] provides a more balanced measure of relevance by accounting for the entropy of the features and the target variable. Normalized Mutual Information is an extension of mutual information that normalizes the value to lie between 0 and 1, providing a normalized measure of dependency between two variables. It helps in a better handling of feature redundancy. It ensures that features are selected based on their unique contribution to the target variable, thereby reducing the chance of selecting highly redundant features. The NI between two variables X and Y is defined as:

$$I(f_c; C) - \frac{1}{|S|} \sum_{f_s \in S} \frac{I(f_s; f_c)}{min\ (H(f_c); H(f_s))} \quad (6)$$

where $I(f_c; C)$ quantifies the amount of information that the candidate feature f_c provides about the target class C. High mutual information indicates that knowing the value of f_c reduces uncertainty about C, $\frac{1}{|S|}$ is a normalization term, which averages the redundancy term over all selected features, |s| represents the cardinality of the set S which is the number of features already selected.

Normalized Mutual Information Based Feature Selection (NMIFS) uses the average normalized MI as a measure of redundancy of the individual feature and the subset of selected feature.

The biggest advantage of NMIFS is the elimination of the user defined parameter β. NMIFS typically normalizes the mutual information by the entropy of the features and the target

variable, making the scores comparable across different features. It is an improvement over the traditional MI based feature selection algorithms.

3.4.3 Weighted Mutual Information Based Feature Selection and Distance-Based Elimination

In this section, usage of MI based distance measure between two features to quantify inter dependency between them is discussed and how MI is used to rank features by measuring dependencies between features and output class labels is also described.

A combination of weighted mutual information and distance-based elimination (DMIFS) [17] strategies are used to select the most relevant features for prediction. This approach ensures that the selected features have high mutual information with the target variable while minimizing redundancy among the features.

The normalized mutual information (NI) between X_i, X_j used as a dissimilarity or distance measure denoted as $D_{NI}(X_i, X_j)$ is as follows:

$$D_{NI}(X_i, X_j) = \left(1 - \sqrt{(X_i, X_j)}\right)^2 \quad (7)$$

On the other hand, MI for a particular feature b averaged over all classes is defined as:

$$\text{MI(b)} = \sum_{k=1}^{|c|} P(c_k)\text{I}(c_k, b) \quad (8)$$

where, |c| is the total number of classes, P (c_k) is the probability of the class $c_k, I(c_k, b)$ is the MI between class c_k and feature b. Features can be ranked based on values obtained from eq. 7.

The aim is to select a relevant subset of features in a way so that they are as independent as possible.

Let, L be the total number of features and d be the number of features to be selected. At some point in the selection process, let S be the set of selected features and let R be the set of remaining (unselected) features. The process is initialized with S = φ and R = original set of features (L) consisting of all the pollutants.

At first, features are ranked considering MI between each feature and output class labels as mentioned in eq. 7. Then, the MI based distances between each pair of features are computed using eq. 4 and are stored in distance matrix DIST_MI of size L×L. At any iteration, feature with the highest rank (B_I) is selected and included in set of selected features S. The number of neighboring features of B_I to be checked is controlled by a parameter β. Among the neighbors, those features which are much closer (controlled by parameter η) to B_I with respect to MI based distance are eliminated and will not be used in future steps.

3.4.4 Grey Wolf Optimization Algorithm for Feature Selection

Grey Wolf Optimization (GWO) [7] is formulated based on the hunting strategy of a grey wolf. Compared to other optimization algorithms, GWO can effectively overcome the local optimal trap and provides better exploration characteristics in search space. Generally grey wolf follows leadership hierarchy for the hunting process. wolf groups are mainly categorized into four such as alpha, beta, delta and omega. To represent the leadership hierarchy, alpha the male is considered as a leader for the group. All the decisions are made by alpha in the hunting process, sleeping time, location etc., The beta helps the alpha in the decision-making process. If the alpha is not present or died then the beta will become alpha. Omega wolves occupy last positions and these wolves will always work under the alpha and beta wolfs. Generally, omega wolves are considered as a group and it will work mainly in the hunting process based on the decisions of alpha. The wolves which do not comes under alpha, beta and omega are considered as delta wolves. These delta wolves are dominant to omega volves and comes under alpha and beta. Generally, care takers, scouts, elders, sentinel are coming under this delta group.

In the first phase, search for prey, wolves track, chase and approach the prey. In the encircling process, wolves encircle and harassing the prey. In the final attacking phase, direct attack is performed to capture the prey. While formulating the optimization model, the first fit solution is considered from alpha (α), the second and third best fit solution is considered from beta (β) and delta (δ) wolves respectively. Finally, ω wolves are selected which works based on the comments of α, β and δ wolves. In the hunting process the prey encircling process is mathematically formulated as:

$$\vec{k} = |\overrightarrow{d} \cdot \vec{\imath}_P(t) - \vec{I}(t)| \tag{9}$$

$$\vec{I}\,(t+1) \;=\; \overrightarrow{I_p}(t) \;- \overrightarrow{A}.\vec{k} \tag{10}$$

where the coefficient vectors are indicated as $\vec{d}$ and $\vec{A}$, the prey position vector is indicated as $\overrightarrow{I_p}$ and the grey wolf position vector is indicated as $\vec{I}$. The current iteration is indicated as t. The coefficient vectors $\vec{d}$ and $\vec{A}$ can be calculated as follows:

$$\vec{d} = 2.\vec{r}_2 \tag{11}$$

$$\vec{A} = 2\,\vec{a}.\vec{r}_1 - \vec{a} \tag{12}$$

where the random vectors are indicated as $\vec{r}_1$ and $\vec{r}_2$ and its range is given as [0, 1]. During the iteration, the component $\vec{a}$ is linearly decreased from 2 to 0.

Figure 3.3 depicts a simple illustration of grey wolf optimization prey encircling process.

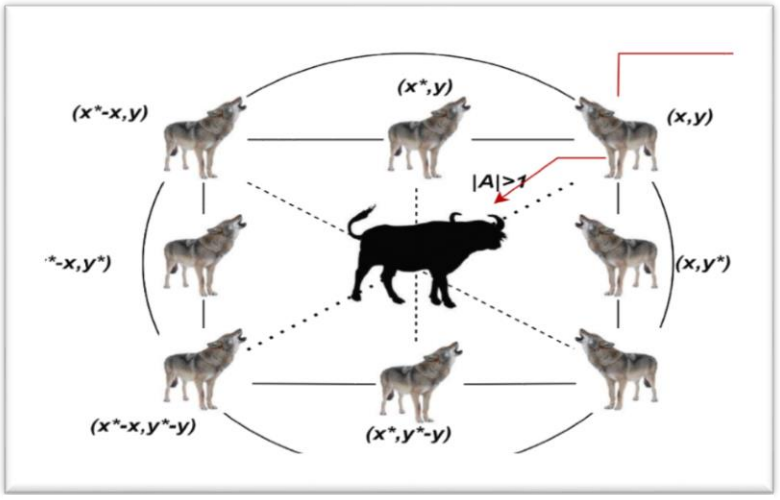


Figure 3.3. Prey Hunting Strategy of GWO.

Source: https://transpireonline.blog/2019/08/09/behavior-of-grey-wolf-optimization-gwo-algorithm-using-meta-heuristics-method/#google_vignette

3.4.5 Pearson Correlation Coefficient Based Feature Selection

A correlation-based feature selection method that calculates the Pearson coefficient [9] of each feature to the target variable has been exploited in the present work to determine the optimal number of input variables (pollutants) when developing a predictive model. Pearson correlation coefficient measures the linear correlation between two sets of data. Statistical correlation-based feature selection algorithms compute correlations between every pair of the input

variable and the target variable. The variables possessing the strongest correlation with the target variable are then filtered for further study. For the present dataset, a Pearson correlation-based feature selection method has been applied to identify the relevant features. To select significant features, the correlation analysis of the Air Quality Index (AQI) feature has been exercised with all of the other pollutants. These pollutants are correlated with AQI based on the correlation values above the threshold of 0.3.

A graphical representation of positive, negative and no correlation is shown below in Figure 3.4.

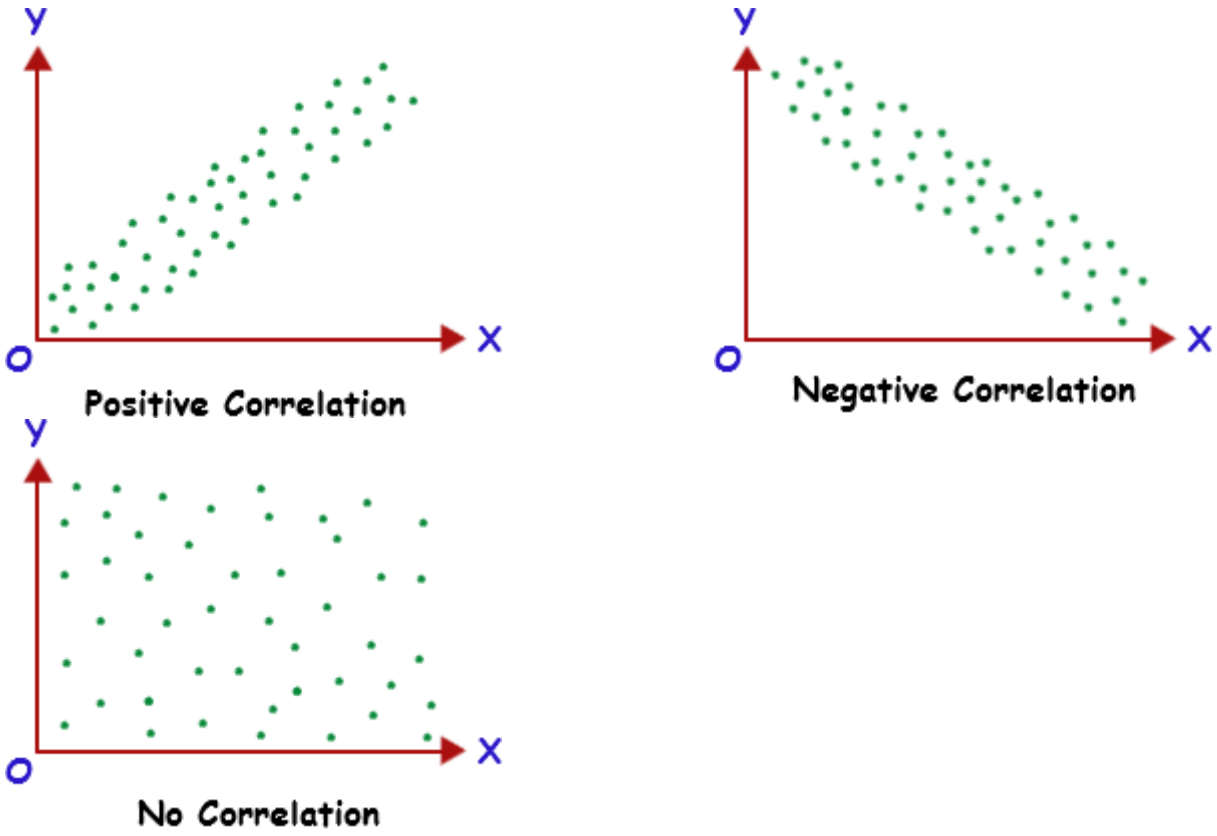


Figure 3.4. Graphical Representation of Positive, Negative and No Correlation Between Features.

Source:https://www.internetgeography.net/wp-content/uploads/2020/09/Types-of-scatter-graph.png

3.4.6 Principal Component Analysis

Principal Component Analysis (PCA) [5] is a statistical technique used for dimensionality reduction while preserving as much variance in the data as possible. It transforms the original variables into a new set of uncorrelated variables called principal components (PCs). Each principal component is a linear combination of the original variables, and the components are ordered such that the first few retain most of the variation present in the original variables.

PCA is a powerful tool for dimensionality reduction, feature extraction, noise reduction, and data visualization. By transforming the dataset into principal components, we retain the most important information while reducing the dimensionality, which helps improve the efficiency and performance of predictive models.

PCA works through the following steps:

1. **Standardization**: PCA requires the data to be scaled. This ensures that each variable contributes equally to the analysis.
2. **Covariance Matrix Computation**: The covariance matrix is calculated to understand how the variables vary together.
3. **Eigenvalues and Eigenvectors**: The eigenvalues and eigenvectors of the covariance matrix are computed. The eigenvectors represent the directions of maximum variance (principal components), while the eigenvalues indicate the magnitude of the variance in those directions.
4. **Principal Components**: The eigenvalues and their corresponding eigenvectors are sorted in descending order. The eigenvectors with the largest eigenvalues form the first principal components.
5. **Dimensionality Reduction**: A subset of principal components that explain a significant portion of the variance in the data are selected. This reduces the dimensionality of the dataset while retaining the most important information.

PCA is used for the following purposes, namely:

1. **Dimensionality Reduction**: PCA reduces the number of variables, which simplifies models, reduces computational cost, and mitigates the curse of dimensionality.
2. **Feature Extraction**: PCA identifies the most important features that capture the maximum variance in the data, aiding in feature selection.
3. **Noise Reduction**: By focusing on the components that explain the most variance, PCA can help filter out noise in the data.
4. **Visualization**: PCA transforms high-dimensional data into a lower-dimensional space, making it easier to visualize and interpret.

In the context of the proposed method, PCA is applied to the dataset after using SMOTE (Synthetic Minority Over-Sampling Technique) which is a method to handle class imbalance. The steps include fitting PCA, determining the number of components to retain based on the desired variance, and creating a transformed dataset with the selected principal components.

3.5 Data Augmentation to Handle Class Imbalance

The AQI dataset [8] under experimentation is highly imbalanced. Thus, to balance the data, synthetic minority oversampling (SMOTE) is employed. Using SMOTE [7] the imbalanced dataset is converted into balanced dataset. SMOTE performs oversampling of data and the

inadequate data are supplemented with additional data. Generally SMOTE finds a feature vector and its closet neighbor and obtains the difference values. Then the difference is multiplied with a random number between zero and one to find the new data point. This process is repeated for all the feature vectors. Compared to other data balancing algorithms, SMOTE provides better balancing by generating synthetic data points. The major benefit of SMOTE over other data balancing techniques is that SMOTE does not produce duplicate data points. It produces artificial data points with marginal difference from actual data points. A simple illustration that describes the sample generation and resampling performed in SMOTE is depicted in Figure 3.5.

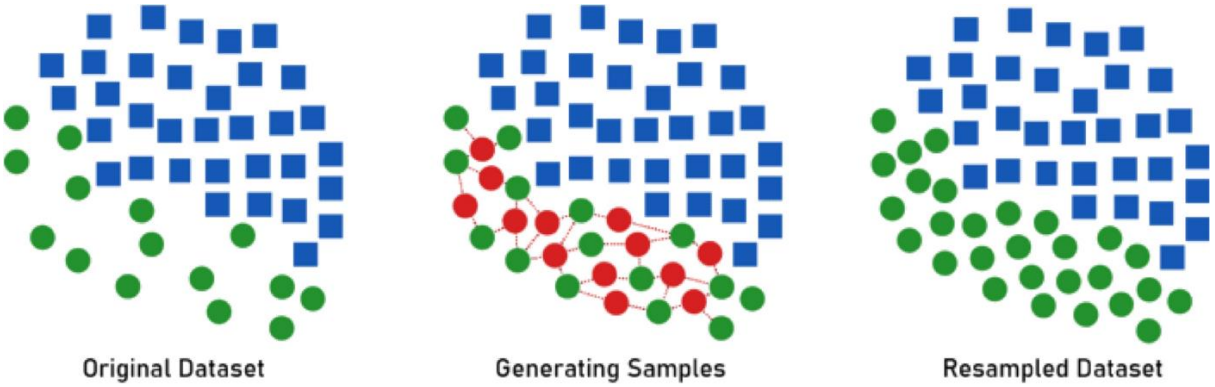


Figure 3.5. Working Principle of SMOTE by Generating Synthetic Samples.

Source:https://www.researchgate.net/figure/The-visual-representation-of-SMOTE-oversampling-for-an-imbalanced-classification_fig2_370574824

3.6 Classification Algorithm and its Variants Utilised for Each Feature Selection Method

In the context of the present work, three different classifiers namely Extreme Gradient Boosting (XGBoost) classifier, K Nearest Neighbours (KNN) classifier and Support Vector Machine (SVM) classifier are used to predict the target label AQI Bucket. Classifiers are used extensively in supervised machine learning where we can map the input features to a labelled target.

3.6.1 Classification Algorithms

Classifier algorithms [4] are a cornerstone of supervised machine learning, used extensively in various fields. These algorithms are designed to categorize or label data points into predefined classes based on input features. The primary objective of classification is to learn a mapping function f: X→Y, where X represents the input features and Y represents the class labels.

A classifier algorithm takes an input data set consisting of features and corresponding labels and learns a decision boundary that can accurately classify new, unseen instances. The goal is to build a model that can predict the class label for any given input vector with high accuracy.

In this present work, we have used Extreme Gradient Boost (XGBoost), K Nearest Neighbors (KNN) and Support Vector Machine (SVM) classifiers.

3.6.2 Components of Classifier Algorithms

Classifier algorithms typically consist of the following components:

1. **Training Data:** A set of examples used to train the model, consisting of input vectors X and corresponding output labels Y.

2. **Feature Extraction:** The process of transforming raw data into a set of features that can be used for training the model.

3. **Model:** The mathematical representation of the classifier, which could be a decision tree, a set of rules, a neural network, etc.

4. **Learning Algorithm:** The method used to optimize the model parameters based on the training data, minimizing the error between the predicted and actual labels. Hyperparameter tuning also falls under this regard.

5. **Evaluation Metrics:** Metrics such as accuracy, precision, recall, F1 score, mean absolute error and ROC-AUC used to assess the performance of the classifier.

3.6.3 Extreme Gradient Boosting Classifier

In this section, the application of the Extreme Gradient Boosting (XGBoost) [4] classifier is detailed in the context of our thesis for predicting the AQI Bucket using various feature selection methods. The XGBoost classifier is chosen due to its high performance and robustness in handling diverse datasets. A comprehensive overview of the XGBoost classifier is provided, its underlying mechanics, and its application with different feature selection methods in our study are also described.

XGBoost is a scalable and efficient implementation of gradient boosting machines (GBM). Developed by Tianqi Chen, XGBoost has become one of the most popular machine learning algorithms due to its superior performance and flexibility.

Mathematical Foundation

The primary goal of XGBoost is to optimize the following objective function:

$$\mathcal{L}(\phi) = \sum_{i=1}^{n} l(y_{i,\hat{y}}) + \sum_{k=1}^{K} \Omega(f_k) \quad (13)$$

where:

- $\mathcal{L}(\phi)$ is the overall objective function.
- l is a differentiable convex loss function that measures the difference between the prediction $\hat{y}$ and the target y_i.
- Ω is a regularization term that penalizes the complexity of the model to avoid overfitting.
- f_k is the prediction function for the k – th tree in the ensemble.

In our research, the XGBoost classifier is employed to predict the target variable AQI Bucket, using various combinations of feature selection methods. The feature selection methods are crucial as they help in identifying the most significant features, thereby improving the model's performance and interpretability.

The following feature selection methods are evaluated with XGBoost:

1. **Mutual Information Feature Selection (MIFS) with 6 and 8 features**
2. **Normalized Mutual Information Feature Selection (NMIFS) with 6 and 8 features**
3. **Weighted Mutual Information Feature Selection (DMIFS) with 6 and 8 features**
4. **Principal Component Analysis (PCA) with 6 and 8 principal components**
5. **Feature Selection using Grey Wolf Optimization Algorithm**
6. **Feature Selection using Pearson Correlation Coefficient**

The models are compared with respect to various performance metrics such as Accuracy, Mean Absolute Error and the ROC-AUC.

3.6.4 K - Nearest Neighbours Classifier

The K-Nearest Neighbors (KNN) classifier [4] is a fundamental and intuitive machine learning algorithm used for classification tasks. It is based on the principle that similar data points are close to each other in the feature space. The KNN algorithm is a type of instance-based learning, where the model does not learn an explicit mapping function from the training data but rather memorizes the training instances and uses them for making predictions.

The KNN classifier operates by identifying the k-nearest data points (neighbors) in the feature space to a given query point and assigning the most common class label among those neighbors to the query point. The distance metric commonly used to identify the nearest neighbors is the Euclidean distance, although other distance metrics such as Manhattan, Minkowski, or Hamming distances can also be used depending on the nature of the data.

Mathematically, the Euclidean distance d between two points x = $(x_1, x_2, \ldots, x_n)$ and y = $(y_1, y_2, \ldots, y_n)$ in an n – dimensional space is given by:

$$d(x, y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)\text{^}2} \quad (14)$$

The steps involved in the KNN algorithm are as follows:

1. **Choose the number of neighbours (k):** Determine the value of k, which is the number of nearest neighbours to consider for classification.

2. **Calculate distances:** Compute the distance between the query point and all the points in the training dataset.

3. **Identify the nearest neighbours:** Select the k data points in the training set that are closest to the query point.

4. **Vote for the class labels:** Assign the class label to the query point based on the majority class among the k nearest neighbours.

In this research work, the KNN classifier was employed to predict the target variable, AQI Bucket, using all 12 pollutant features. Subsequently, various feature selection methods were applied to enhance the classifier's performance.

The feature selection methods used in this thesis are:

- **Mutual Information Feature Selection (MIFS):**
    - **MIFS with 6 Features**
    - **MIFS with 8 Features**
- **Normalized Mutual Information Feature Selection (NMIFS):**
    - **NMIFS with 6 Features**
    - **NMIFS with 8 Features**

- **Weighted Mutual Information Feature Selection (DMIFS):**
    - **DMIFS with 6 Features**
    - **DMIFS with 8 Features**
- **Principal Component Analysis (PCA):**
    - **PCA with 6 Principal Components**
    - **PCA with 8 Principal Components**
- **Grey Wolf Optimization Algorithm (GWO):**
    - **Feature Selection using GWO**
- **Pearson Correlation Coefficient:**
    - **Feature Selection using Pearson Correlation**

3.6.5 Support Vector Machine Classifier

Support Vector Machines (SVM) [4] are a set of supervised learning methods used for classification, regression, and outlier detection. The objective of the SVM algorithm is to find a hyperplane in an N-dimensional space (N being the number of features) that distinctly classifies the data points.

For a binary classification problem, the SVM aims to find the optimal hyperplane that maximizes the margin between the two classes. The margin is defined as the distance between the hyperplane and the nearest data points from either class, known as support vectors.

Mathematically, the hyperplane can be defined as: $w \cdot x - b = 0$ where w is the weight vector and b is the bias.

The optimization problem can be expressed as:

$$min_{w,b} \frac{1}{2} ||w||^2 \tag{15}$$

subject to:

$$y_i(w.x_i - b) \geq 1 \tag{16}$$

For cases where the data is not linearly separable, the SVM introduces slack variable ξ to allow some misclassifications, leading to the soft margin SVM:

$$min_{w,b,\xi} \frac{1}{2} ||w||^2 + C \sum_{i=1}^{N} \xi \quad (17)$$

subject to:

$$y_i(w.x_i - b) \geq 1 - \xi \quad (18)$$

where $\xi \geq 0$

Here, C is the regularization parameter that balances the trade-off between maximizing the margin and minimizing the classification error.

To enhance the performance and interpretability of the SVM classifier, various feature selection methods were applied. The methods used include Mutual Information based Feature Selection (MIFS), Normalized Mutual Information based Feature Selection (NMIFS), Weighted Mutual Information based Feature Selection (DMIFS), Principal Component Analysis (PCA), Union and Intersection of features obtained by implementing MIFS, NMIFS and DMIFS algorithms, Grey Wolf Optimization Algorithm, and Pearson Correlation Coefficient.

1. **MIFS using 6 and 8 features:** MIFS selects features based on their mutual information with the target variable. This method was applied to select the top 6 and 8 features, respectively.

2. **NMIFS using 6 and 8 features:** NMIFS normalizes the mutual information values to account for feature redundancies and selects the top features accordingly.

3. **DMIFS using 6 and 8 features:** DMIFS further refines the selection process by considering differential mutual information, which accounts for the influence of other features.

4. **PCA with 6 and 8 principal components:** PCA reduces the dimensionality of the feature space by transforming the original features into a new set of uncorrelated principal components, with the top 6 and 8 components selected.

5. **Grey Wolf Optimization Algorithm:** This metaheuristic optimization technique simulates the leadership hierarchy and hunting mechanism of grey wolves to select the optimal feature subset.

6. **Pearson Correlation Coefficient:** Features are selected based on their linear correlation with the target variable.

3.7 Ensemble Learning Techniques

Ensemble learning [4] is a machine learning technique where multiple models, often referred to as "weak learners" or "base models," are combined to produce a more accurate and robust model. The idea is that by combining multiple models, the ensemble can leverage the strengths of each individual model while mitigating their weaknesses. Ensemble methods are widely used because they tend to improve predictive performance and generalization.

In the context of the current experiment, an ensemble learning technique using the concept of “Stacking” using “Majority Voting” has been employed. Stacking is an ensemble learning technique that combines multiple models (base learners) by training a meta-model to make the final predictions. The meta-model learns how to best combine the predictions of the base models.

Steps in Stacking:

1. **Train Base Models:**
    - Several different models are trained on the training data. These models can be of different types (e.g., decision trees, support vector machines, neural networks etc.).
2. **Generate Base Model Predictions:**
    - The base models are used to make predictions on the training data (often using cross-validation) and on the test data.
3. **Train Meta-Model:**
    - A meta-model is trained on the predictions made by the base models. The meta-model learns to combine these predictions to make the final prediction.

Mathematical Representation:

Let h_1, h_2, …, $h_n$be the base models, and H be the meta-model. The final prediction $\hat{y}$ is given by:

$$\hat{y} = H(h_1(x),\ h_2(x), \ldots, h_n(x)) \tag{19}$$

3.7.1 Union of Feature Sets

Feature selection is a critical step in the machine learning pipeline, particularly when working with high-dimensional data. The goal of feature selection is to identify the most relevant features that contribute to the predictive power of a model, thereby improving model performance, reducing overfitting, and decreasing computational complexity. In this section,

we describe a method to combine the features selected by different feature selection techniques through a union operation, ensuring that the resulting set of features encompasses the most informative attributes identified by each method. By combining the outputs of multiple feature selection methods, we aim to capture a more comprehensive set of relevant features.

The union method for combining features selected by different techniques is motivated by the need to leverage the strengths of multiple feature selection algorithms. Each algorithm may capture different aspects of the data's structure and relationships. By combining their outputs, we aim to create a more robust and informative feature set. This approach mitigates the risk of omitting potentially important features that might be overlooked by any single method.

The union method ensures that all selected features are considered, providing a comprehensive view of the data. This is particularly beneficial in complex datasets where different features may contribute to the target variable in various ways. By adopting this method, we ensure that our final feature set is both comprehensive and representative of the various perspectives offered by different feature selection techniques, ultimately enhancing the robustness and accuracy of our predictive models.

3.7.2 Intersection of Feature Sets

In this section, we describe a method to combine the features selected by different feature selection techniques using an intersection operation. This approach ensures that the resulting set of features includes only those attributes deemed relevant by all applied methods, thereby enhancing the robustness of the selected feature set.

The intersection method for combining features ensures that only the most consistently relevant features, as identified by multiple selection techniques, are retained. This approach reduces the risk of including irrelevant or redundant features that may have been selected by only one method. By focusing on the commonalities across different feature selection techniques, we enhance the robustness and reliability of the selected feature set.

This method is particularly valuable in high-dimensional datasets, where different features may contribute to the target variable in complex ways. By retaining only those features deemed important by all applied methods, we ensure a more stringent and reliable feature selection process. By adopting this method, we ensure that our final feature set is both comprehensive and reliable, incorporating only those features that are universally recognized as relevant, thereby enhancing the robustness and accuracy of our predictive models.

3.7.3 Building an Ensemble Learning Model

The method of ‘stacking’ using ‘majority voting’ has been utilized by combining a series of weak learners Extreme Gradient Boosting (XGBoost) classifier, K Nearest Neighbors (KNN) classifier and Support Vector Machine (SVM) classifier. The meta classifier model is fed the subset of features obtained after the union of six and eight features as well as intersection of six and eight features.

Majority Voting:

Majority voting is a simple ensemble technique where the predictions from multiple models are combined by taking the majority vote for classification tasks.

Steps in Majority Voting:

1. **Train Base Models:**
    - Several different models are trained on the training data.
2. **Make Predictions:**
    - Each model makes predictions on the test data.
3. **Combine Predictions:**
    - For classification, the final prediction is the class that receives the majority vote.

Subsequently, an ensemble learning model was developed using these refined feature sets. The ensemble model integrated three base learners: XGBoost, K-Nearest Neighbors (KNN), and Support Vector Machine (SVM). The stacking method was employed in the ensemble learning model, utilizing majority voting to combine the predictions of the base learners. This approach aims to leverage the strengths of each algorithm, thereby improving the robustness and accuracy of the air quality prediction model.

CHAPTER 4

DATASET AND EXPERIMENTAL SETUP

4.1 Data Set Used

For this study, the Air Quality Index (AQI) dataset extracted from Kaggle repository was used, as referenced by [8]. This dataset possesses observations from January 2015 to July 2020 and it comprises of 16 features with 29,531 instances from 26 different Indian cities. The cities are Ahmedabad, Aizawl, Amaravati, Amritsar, Bengaluru, Bhopal, Brajrajnagar, Chandigarh, Chennai, Coimbatore, Delhi, Ernakulam, Gurugram, Guwahati, Hyderabad, Jaipur, Jorapokhar, Kochi, Kolkata, Lucknow, Mumbai, Patna, Shillong, Talcher, Thiruvananthapuram and Visakhapatnam.

There are 12 pollutant features and they are Particulate Matter ($PM_{2.5}, PM_{10}$) , Nitric Oxide (NO), Nitrogen Dioxide (NO_2), Nitrogen Oxides (NO_x), Ammonia (NH_3) , Carbon Monoxide (CO), Sulphur Dioxide (SO_2), Ozone (O_3), Benzene, Toluene and Xylene, commonly referred to as BTX. In addition to the pollutants, there is a column containing the daily AQI values and a column titled 'AQI Bucket' that categorizes the AQI values into 6 distinct classes, namely Good, Satisfactory, Moderate, Poor, Very Poor and Severe. The 'Date' column represents each date commencing from 1st January 2015 to 7th July 2020. Data is recorded as the average concentration per day.

Figure 4.1 presented below provides brief descriptive statistics of some of the features from this dataset.

	Date	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene
count	29531	29531.000000	29531.000000	29531.000000	29531.000000	29531.000000	29531.000000	29531.000000	29531.000000	29531.000000	29531.000000
mean	2018-05-14 05:40:15.807118080	64.510857	109.659366	16.642601	27.726576	31.063568	20.813789	2.153872	13.830897	33.994121	2.859874
min	2015-01-01 00:00:00	0.040000	0.010000	0.020000	0.010000	0.000000	0.010000	0.000000	0.010000	0.010000	0.000000
25%	2017-04-16 00:00:00	32.150000	79.315000	6.210000	12.980000	14.670000	12.040000	0.540000	6.090000	20.740000	0.240000
50%	2018-08-05 00:00:00	48.570000	95.680000	9.890000	21.690000	23.520000	15.850000	0.890000	9.160000	30.840000	1.070000
75%	2019-09-03 00:00:00	72.450000	111.880000	17.570000	34.665000	36.015000	21.755000	1.380000	13.810000	42.730000	2.420000
max	2020-07-01 00:00:00	949.990000	1000.000000	390.680000	362.210000	467.630000	352.890000	175.810000	193.860000	257.730000	455.030000
std	NaN	59.807551	72.324020	21.506064	23.050531	29.477748	21.028862	6.724660	17.005647	20.202304	14.252822

Figure 4.1. Descriptive Statistics of Features.

4.2 Data Set Preprocessing

The AQI dataset has null values and needs to be imputed before proceeding to build a model for prediction. In the context of the current dataset, the null values are imputed using “median” strategy [5]. The "median" strategy for imputation involves replacing missing values with the median value of the respective feature. The median is the middle value of a data set when it is ordered from least to greatest, and it is less affected by outliers compared to the mean. The percentage of missing values is shown in the Figure 4.2.

```
Percentage of missing values in each column (descending order):
Xylene        61.322001
PM10          37.723071
NH3           34.973418
Toluene       27.229014
Benzene       19.041008
AQI           15.851139
AQI_Bucket    15.851139
PM2.5         15.570079
NOx           14.171549
O3            13.619586
SO2           13.050692
NO2           12.139785
NO            12.129626
CO             6.972334
City           0.000000
Date           0.000000
```

Figure 4.2. Percentage of Missing Values.

Encoding of categorical features which is the target label AQI Bucket in our dataset is performed before imputation using target label encoding and the convention followed is given as:

Good → 0

Satisfactory → 1

Moderate → 2

Poor → 3

Very Poor → 4

Severe → 5

Since it is essentially a multi – class classification problem, this process ensures that the ordinality of the class labels are maintained.

Furthermore, the features are scaled using Maximal Absolute (MaxAb) Scalar to fall within the range [-1, 1]. Normalization is a common practice in preprocessing for machine learning models, enhancing convergence during training.

The Kernel Density Estimation (KDE) plots of all the features are presented below in Figure 4.3.

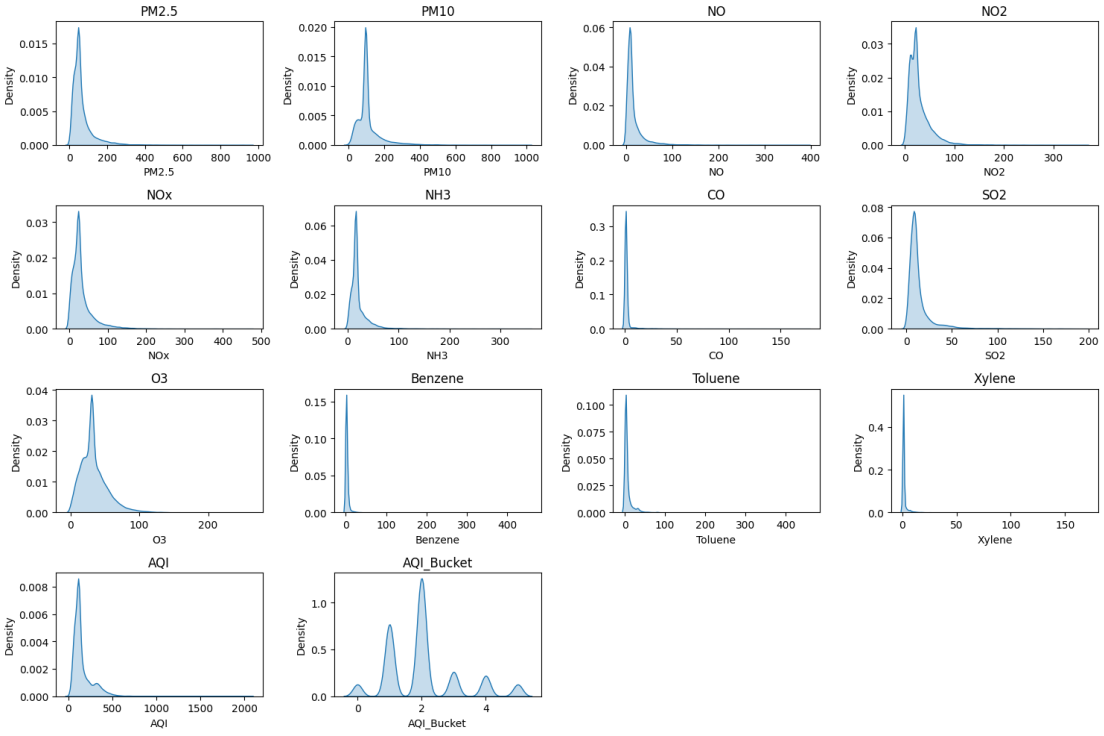


Figure 4.3. KDE Plot.

Various feature selection methods as outlined in Chapter 3; section 3.3 is performed post which the dataset is augmented by utilizing a data augmentation technique called SMOTE [7] which generates synthetic samples to handle class imbalance of the target variable.

4.3 Experimental Setup

The dataset is split into the input or predictor variables (X) and the target variable (y). Three classifier algorithms namely XGBoost classifier, KNN classifier and SVM classifier are used for evaluating each of the feature selection methods. To ensure a robust evaluation of the proposed methodology, the dataset was subjected to a technique called Cross Validation (CV) using 10-fold splits. It ensures all subsets of the data are used for training and testing. CV consists of sample splitting methods that use different portions of the data to test and train a model on different iterations. The number of splits is generally a user-controlled parameter. It is often used in settings where the goal is prediction, and one wants an estimate of the adaptability of a predictive model. This meticulous dataset organization is crucial for maintaining the integrity of the evaluation process and ensuring the reliability of our findings.

4.3.1 Hyperparameter Tuning for Classification Algorithm

The hyperparameters [4] are tuned using Optuna which is a framework designed to automate the process of finding the optimal hyperparameters for machine learning models. Hyperparameter tuning is crucial for enhancing model performance, as the choice of hyperparameters can significantly impact the results. It also reduces the risk of overfitting. Overfitting is a phenomenon where the model adapts remarkably well to the training data but performs poorly on the test data.

Table 4.1, Table 4.2, Table 4.3 given below exhibit the choice of hyperparameters used for each of the XGBoost classifier, KNN classifier and SVM classifier respectively.

Table 4.1. Hyperparameters of XGBoost Classifier.

Hyperparameters Used	
Hyperparameters	**Values**
Estimators	800, 900
Max Depth	11, 12
Learning Rate	0.22, 0.23
Subsample	0.9, 1.0
Colsample_bytree	0.8, 1.0
Evaluation Metric	mlogloss
Label Encoder	False

Table 4.2. Hyperparameters of KNN Classifier.

Hyperparameters Used	
Hyperparameters	**Values**
Number of neighbors	2, 3
Weights	Uniform, Distance
Trial Integers	1(Manhattan), 2(Euclidean)

Table 4.3. Hyperparameters of SVM Classifier.

Hyperparameters Used	
Hyperparameters	**Values**
Regularization Parameter 'C'	77, 78
Gamma	99, 100
Kernel	Radial Basis Function (RBF)

4.3.2 Hyperparameters Used for Ensemble Learning Model

The weak learners in building an ensemble learning model are XGBoost classifier, KNN classifier and SVM classifier. The ensemble learning model was built on the theory of 'stacking' and the results were concluded by the technique of 'majority voting'. Each of the ensemble learning model was evaluated on the following:

a) Union of Features
b) Intersection of Features

The hyperparameters used for the ensemble model on union of features and intersection of features are depicted in Table 4.4.

Table 4.4. Hyperparameters of Ensemble Model Classifier.

Hyperparameters Used for XGBoost	
Hyperparameters	**Values**
Estimators	500, 600
Max Depth	14, 15
Learning Rate	0.25, 0.30
Subsample	0.8, 1.0
Colsample_bytree	0.7, 1.0
Evaluation Metric	mlogloss
Label Encoder	False
Hyperparameters Used for KNN	
Hyperparameters	**Values**
Number of neighbors	2, 4
Weights	Uniform, Distance
Trial Integers	1(Manhattan), 2(Euclidean)
Hyperparameters Used for SVM	
Hyperparameters	**Values**
Regularization Parameter 'C'	74, 75
Gamma	96, 97
Kernel	Radial Basis Function (RBF)

4.4 Performance Evaluation Metrics of the Classifier Models

In evaluating the performance of machine learning models, it is essential to use a variety of metrics to gain a comprehensive understanding of the model's behaviour and effectiveness. For this research, the performance of the models was assessed using Accuracy, Precision, Recall, F1 Score, Mean Absolute Error (MAE), the ROC-AUC curve, and the Confusion Matrix.

4.4.1 Accuracy

Accuracy is the simplest and most intuitive performance measure. It is defined as the ratio of the number of correct predictions to the total number of predictions. Mathematically, it can be expressed as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

where TP is the number of true positives, TN is the number of true negatives, FP is the number of false positives, and FN is the number of false negatives.

4.4.2 Precision

Precision, also known as Positive Predictive Value, measures the proportion of true positive predictions among all positive predictions. It is a useful metric when the cost of false positives is high. Precision is defined as:

$$\text{Precision} = \frac{TP}{TP + FP}$$

4.4.3 Recall

Recall, also known as Sensitivity or True Positive Rate, measures the proportion of true positive predictions among all actual positive instances. It is particularly important when the cost of false negatives is high. Recall is defined as:

$$\text{Recall} = \frac{TP}{TP + FN}$$

4.4.4 F1 Score

The F1 Score is the harmonic mean of Precision and Recall, providing a single metric that balances both concerns. It is particularly useful when the dataset has imbalanced classes. The F1 Score is defined as:

$$\text{F1 Score} = 2 \cdot \frac{Precision \cdot Recall}{Precision + Recall}$$

4.4.5 Mean Absolute Error

The Mean Absolute Error (MAE) measures the average magnitude of the errors in a set of predictions, without considering their direction. It is the average over the test sample of the absolute differences between prediction and actual observation where all individual differences have equal weight. Mathematically, it is expressed as:

$$\textbf{MAE} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

where y_i is the actual value and $\hat{y}_i$ is the predicted value and n is the number of samples.

4.4.6 ROC-AUC Curve

The Receiver Operating Characteristic (ROC) curve is a graphical representation of the diagnostic ability of a binary classifier as its discrimination threshold is varied. The Area Under the ROC Curve (AUC) is a single scalar value that summarizes the performance of the classifier. The ROC curve plots the True Positive Rate (Recall) against the False Positive Rate (1 - Specificity) at various threshold settings. A model with an AUC closer to 1 indicates a better performance.

4.4.7 Confusion Matrix

A Confusion Matrix is a table that is often used to describe the performance of a classification model. It allows visualization of the model's performance by showing the actual versus predicted classifications, thereby enabling the identification of errors. The matrix consists of four cells:

- **True Positive (TP):** Correctly predicted positive cases.
- **True Negative (TN):** Correctly predicted negative cases.

- **False Positive (FP):** Incorrectly predicted positive cases.
- **False Negative (FN):** Incorrectly predicted negative cases.

The confusion matrix provides a more detailed insight into the model's performance by showing the types of errors made.

In this research, the performance of the models was evaluated by calculating the above metrics over 10-fold cross-validation. This approach ensures that the model's performance is reliable and not dependent on a particular data split. The mean values of these metrics were computed across the folds to provide a robust estimate of model performance.

The ROC-AUC curve was plotted to visualize the trade-off between the True Positive Rate and False Positive Rate across different thresholds, offering a comprehensive view of the classifier's performance. Additionally, confusion matrices for each fold were generated to provide insights into the types and frequency of classification errors.

Tables and figures summarizing these performance metrics and visualization aids are included in the results section in Chapter 5 to facilitate a clear and detailed understanding of the model's effectiveness and reliability across different feature selection methods and configurations. This thorough evaluation process ensures that the findings are robust and generalizable, providing confidence in the conclusions drawn from the research.

CHAPTER 5

RESULTS AND ANALYSIS

5.1 Feature Selection Algorithm Results

In the context of our research on the present dataset [8], 6 different feature selection strategies were examined. The features obtained from each of the feature selection strategy are presented in this section.

a) **Mutual Information Based Feature Selection (MIFS):** The features selected are presented in Table 5.1 given below.

Table 5.1. Selected Features using MIFS.

Mutual Information Feature Selection (MIFS)	
Features Selected (6)	**Features Selected (8)**
'$\mathbf{PM_{2.5}}$', '$\boldsymbol{PM_{10}}$', 'CO', '$\boldsymbol{O_3}$', '$\boldsymbol{NO_x}$', '$\boldsymbol{SO_2}$'	'$PM_{2.5}$', 'PM_{10}', 'CO', 'O_3', 'NO_x', 'SO_2', 'Benzene', 'Xylene'

b) **Weighted Mutual Information Based Feature Selection and Distance Based Elimination (DMIFS):** The features selected are presented in Table 5.2 given below.

Table 5.2. Selected Features using DMIFS.

Weighted Mutual Information Feature Selection and Distance Based Elimination (DMIFS)	
Features Selected (6)	**Features Selected (8)**
'$\mathbf{PM_{2.5}}$', '$\boldsymbol{NO_2}$', '$\boldsymbol{NH_3}$', 'CO', '$\boldsymbol{O_3}$', 'Benzene'	'$\mathbf{PM_{2.5}}$', 'PM_{10}', 'NO', '$\boldsymbol{NO_2}$', '$\boldsymbol{NH_3}$', 'CO', 'SO_2', '$\boldsymbol{O_3}'$

c) **Normalized Mutual Information Feature Selection (NMIFS):** The features selected are presented in Table 5.3 given below.

Table 5.3. Selected Features using NMIFS.

Normalized Mutual Information Feature Selection (NMIFS)	
Features Selected (6)	**Features Selected (8)**
'PM2.5', 'CO', 'PM10', 'NO2', 'O3', 'NOx'	'PM2.5', 'CO', 'PM10', 'NO2', 'O3', 'NOx', 'SO2', 'Toluene'

d) **Principal Component Analysis (PCA):** The principal components captured at 90% and 95% variance are depicted in Table 5.4 given below.

Table 5.4. Selected Principal Components using PCA.

Principal Component Analysis (PCA)	
Principal Components Selected (90%)	**Principal Components Selected (95%)**
PC1 PC2 PC3 PC4 PC5 PC6	PC1 PC2 PC3 PC4 PC5 PC6 PC7 PC8

e) **Grey Wolf Optimization (GWO) Feature Selection:** The features selected are presented in Table 5.5 as shown below.

Table 5.5. Selected Features using GWO.

Grey Wolf Optimization (GWO) Feature Selection
Features Selected
'PM2.5', 'PM10', 'NO2', 'NOx', 'NH3', 'CO', 'SO2', 'O3', 'Benzene'

f) Pearson Correlation Coefficient Feature Selection: The features above the threshold value of 0.3 are selected for further analysis which are shown in Table 5.6. The Table 5.7 shown below shows the correlation values of all the pollutant features with respect to the Air Quality Index (AQI).

Table 5.6. Selected Features using Pearson Correlation Coefficient.

Pearson Correlation Coefficient Feature Selection
Features Selected
'PM2.5', 'PM10', 'NO', 'NO2', 'NOx', 'CO', 'SO2', 'Toluene'

Table 5.7. Pearson Correlation Coefficient Values.

Serial No.	Features	Correlation Value with AQI	Serial No.	Features	Correlation Value with AQI
1	$PM_{2.5}$	0.79	7	CO	0.54
2	PM_{10}	0.63	8	SO_2	0.40
3	NO	0.48	9	O_3	0.23
4	NO_2	0.46	10	Benzene	0.23
5	NO_x	0.40	11	Toluene	0.34
6	NH_3	0.26	12	Xylene	0.17

5.1.1 Union and Intersection of Features from Feature Selection Algorithms

a) **Union of Features:** In order to build a robust ensemble learning model, the union of all of the 6 set of features as well as union of all of the 8 set of features obtained from the mutual information-based feature selection algorithms namely MIFS, NMIFS and DMIFS are considered. The features obtained as a result of union of 6 features and union of 8 features are summarized in the Table 5.8 and Table 5.9 respectively.

Table 5.8. Union of 6 Features.

Union of 6 Features
Features Selected
'CO', 'SO2', 'PM10', 'NO2', 'Benzene', 'NOx', 'NH3', 'PM2.5', 'O3'

Table 5.9. Union of 8 Features.

Union of 8 Features
Features Selected
'Xylene', 'CO', 'SO2', 'PM10', 'NO2', 'Toluene', 'Benzene', 'NOx', 'NH3', 'PM2.5', 'O3', 'NO'

b) Intersection of Features: In order to build a robust ensemble learning model, the intersection of all of the 6 set of features as well as intersection of all of the 8 set of features obtained from the mutual information-based feature selection algorithms namely MIFS, NMIFS and DMIFS are considered. The features obtained as a result of intersection of 6 features and intersection of 8 features are summarized in the Table 5.10 and Table 5.11 respectively.

Table 5.10. Intersection of 6 Features.

Intersection of 6 Features
Features Selected
'PM2.5', 'O3', 'CO'

Table 5.11. Intersection of 8 Features.

Intersection of 8 Features
Features Selected
'CO', 'SO2', 'PM10', 'PM2.5', 'O3'

5.2 Results of Classification Algorithms

Following an exhaustive training of three traditional classifiers, namely XGBoost, KNN and SVM as well as the usage of ensemble learning models with our proposed methodology on the AQI dataset [8], the results are demonstrated with the aid of tables and figures in the sections below accompanied by an analysis of the results obtained.

5.2.1 XGBoost Classifier

The XGBoost classifier was thoroughly trained on the training data and evaluated on the test data using cross validation with 10-fold splits for each of the feature selection strategies and the accuracy score and Mean Absolute Error (MAE) averaged over 10 folds has been displayed in Table 5.7 as shown below.

Table 5.7. Test Accuracy and Test MAE of XGBoost Classifier.

Performance Metrics	12 Features	MIFS 6 Features	MIFS 8 Features	NMIFS 6 Features	**NMIFS 8 Features**	DMIFS 6 Features	DMIFS 8 Features	PCA 6 Components	PCA 8 Components	Pearson Correlation	GWO
Test Accuracy	93.5 %	92.0 %	93.1 %	92.6 %	**93.8 %**	91.7 %	93.3 %	91.0%	92.1%	91.7%	91.6%
Test MAE	0.07	0.08	0.07	0.08	**0.07**	0.09	0.07	0.10	0.08	0.09	0.09

From the table above, it is evident that the NMIFS method, selecting a set of 8 features, achieves the highest accuracy score among all feature selection strategies employed. DMIFS with 8 features follows closely. The Mean Absolute Error (MAE) is consistent between these two methods and comparable to the MAE obtained when all 12 features are used for classification. The superior results with fewer features and reduced dimensionality emphasize the importance of feature subset selection, which reduces computational complexity. When compared to other advanced feature selection strategies, such as GWO and Pearson Correlation Coefficient, the mutual information-based feature selection method yields higher accuracy scores and lower MAE values, highlighting the significance of choosing features with minimal interdependence.

The ROC - AUC Curve for NMIFS with 8 Features is shown in Figure 5.1.

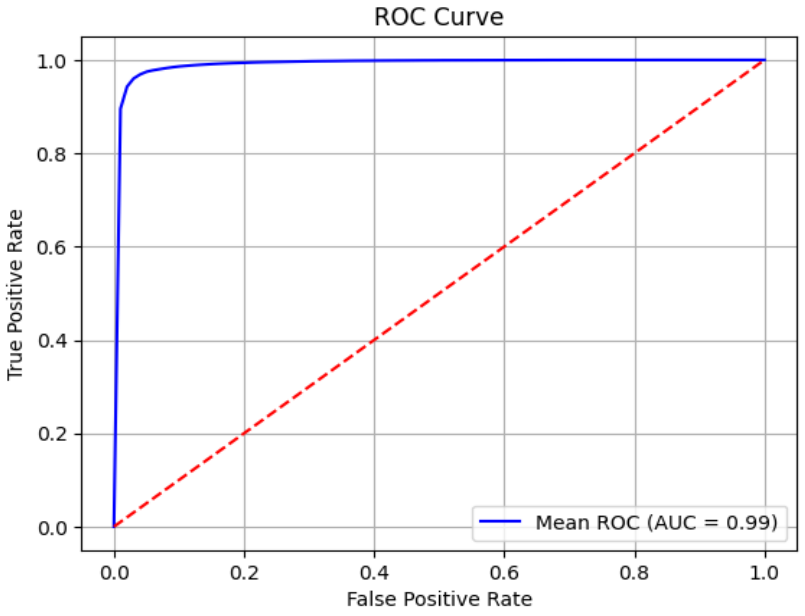


Figure 5.1. ROC Curve for NMIFS with 8 Features using XGBoost.

The Confusion Matrix for XGBoost classifier for all 12 features, NMIFS with 8 features, GWO features, and Pearson Correlation features is shown in Figure 5.2, Figure 5.3, Figure 5.4 and Figure 5.5 respectively.

```
[[1302   16   15    2    0    0]
 [  42 1304   82    3    2    0]
 [   7  128 1142   72    9    6]
 [   0    3   38 1261   33    1]
 [   0    1    4   25 1264   18]
 [   0    1    2    3   11 1309]]
```

Figure 5.2. Confusion Matrix for 12 Features using XGBoost.

```
[[1305   11   17    1    1    0]
 [  39 1300   89    3    2    0]
 [   8  136 1135   70    8    7]
 [   0    6   38 1257   33    2]
 [   0    1    4   22 1271   14]
 [   0    2    3    0   11 1310]]
```

Figure 5.3. Confusion Matrix for NMIFS with 8 Features using XGBoost.

```
[[1300   18   15    2    0    0]
 [  39 1303   85    4    2    0]
 [   9  136 1133   69   10    7]
 [   0    5   37 1255   38    1]
 [   0    2    5   27 1261   17]
 [   0    0    5    1    9 1311]]
```

Figure 5.4. Confusion Matrix of GWO Features using XGBoost.

```
[[1288   30   13    4    0    0]
 [  65 1258  106    2    2    0]
 [   8  174 1085   84    7    6]
 [   0    9   51 1231   43    2]
 [   0    1    4   55 1227   25]
 [   0    2    4    3   23 1294]]
```

Figure 5.5. Confusion Matrix of Pearson Correlation Coefficient using XGBoost.

5.2.2 KNN Classifier

The KNN classifier was thoroughly trained on the training data and evaluated on the test data using cross validation with 10-fold splits for each of the feature selection strategies and the accuracy score and Mean Absolute Error (MAE) averaged over 10 folds has been displayed in Table 5.8 as shown below.

Table 5.8. Test Accuracy and Test MAE of KNN Classifier.

Performance Metrics	12 Features	MIFS 6 Features	MIFS 8 Features	NMIFS 6 Features	NMIFS 8 Features	DMIFS 6 Features	**DMIFS 8 Features**	PCA 6 Components	PCA 8 Components	Pearson Correlation	GWO
Test Accuracy in %	93.1%	92.4%	93.6%	92.3%	93.1%	91.5%	**93.7%**	92.1%	93.1%	91.7%	91.6%
Test MAE	0.07	0.08	0.07	0.08	0.07	0.10	**0.07**	0.09	0.07	0.09	0.09

From the table above, it is evident that the DMIFS method that outputs a set of 8 features produce the highest accuracy score in comparison to all the other feature selection strategies that are employed immediately followed by the MIFS method that produces a set of 8 features. The MAE value is constant between these two methods and is comparable to the MAE value when all the 12 features are used for classification. However, since better results are achieved with lesser number of features, it indicates the importance of selecting a subset of features from the entire set of features since it requires less computational complexity. Compared to other state of the art feature selection strategies, namely GWO feature selection and Pearson Correlation Coefficient feature selection, the features selected by implementing concepts of

mutual information produce higher accuracy score and lower MAE value, thereby reinstituting the significance of selecting features that have less dependence amongst each other.

The ROC - AUC Curve for DMIFS with 8 Features is depicted in Figure 5.6.

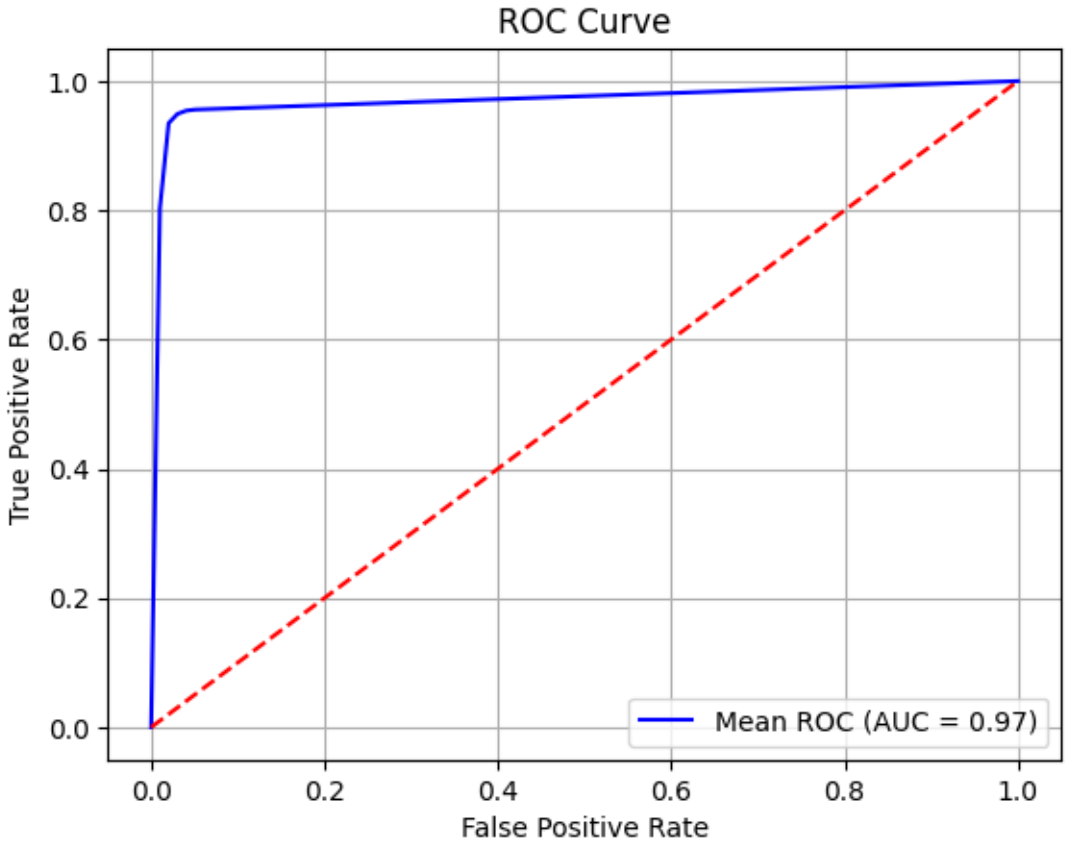


Figure 5.6. ROC Curve for DMIFS with 8 Features using KNN.

The Confusion Matrices for KNN classifier for all 12 features, DMIFS with 8 features, GWO features and Pearson Correlation features are shown in Figure 5.7, Figure 5.8, Figure 5.9 and Figure 5.10 respectively.

```
[[1315    5   13    2    0    0]
 [  63 1284   77    7    2    0]
 [  10  198  997  136   15    8]
 [   0    1    6 1313   15    1]
 [   0    2    1   11 1291    7]
 [   0    1    1    2    3 1319]]
```

Figure 5.7. Confusion Matrix of 12 Features using KNN.

```
[[1311    8   14    2    0    0]
 [  27 1298  101    6    1    0]
 [  14  195 1027  102   15   11]
 [   0    2   14 1310    9    1]
 [   0    0    2    6 1295    9]
 [   0    2    1    2    5 1316]]
```

Figure 5.8. Confusion Matrix for DMIFS with 8 Features using KNN.

```
[[1304   16   13    2    0    0]
 [  67 1250  103    7    6    0]
 [  17  194  995  132   17    9]
 [   1    2   24 1272   36    1]
 [   0    0    3   27 1263   19]
 [   0    2    2    1   10 1311]]
```

Figure 5.9. Confusion Matrix of GWO Features using KNN.

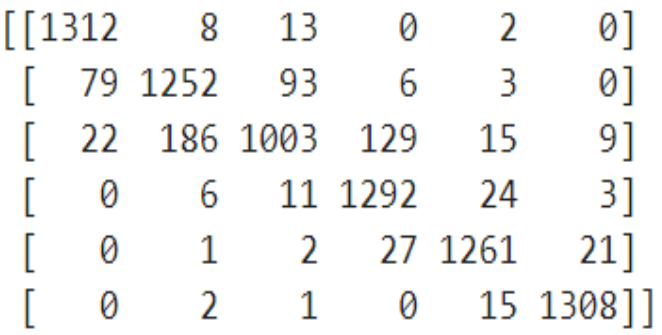


Figure 5.10. Confusion Matrix of Pearson Correlation Coefficient Features using KNN.

5.2.3 SVM Classifier

The SVM classifier was thoroughly trained on the training data and evaluated on the test data using cross validation with 10-fold splits for each of the feature selection strategies and the accuracy score and Mean Absolute Error (MAE) averaged over 10 folds has been displayed in Table 5.9 as shown below.

Table 5.9. Test Accuracy and Test MAE of SVM Classifier.

Performance Metrics	12 Features	MIFS 6 Features	MIFS 8 Features	NMIFS 6 Features	**NMIFS 8 Features**	DMIFS 6 Features	DMIFS 8 Features	PCA 6 Components	PCA 8 Components	Pearson Correlation	GWO
Test Accuracy	89.2%	85.1%	86.1%	85.1%	**89.6%**	79.9%	87.7%	86.6%	89.6%	84.8%	87.8%
Test MAE	0.11	0.16	0.15	0.16	**0.11**	0.23	0.13	0.15	0.11	0.16	0.13

From the table above, it can be observed that the NMIFS method, which selects a set of 8 features, achieves the highest accuracy score compared to all other feature selection strategies used. This is closely followed by PCA of 8 principal components. The Mean Absolute Error (MAE) remains consistent between these two methods and is comparable to the MAE when all 12 features are utilized for classification. However, the superior results obtained with fewer features and a reduced dimension highlight the importance of selecting a subset of features, as it reduces computational complexity. When compared to other advanced feature selection strategies, such as GWO feature selection and Pearson Correlation Coefficient feature selection, the feature selection method based on the concept of mutual information yield higher

accuracy scores and lower MAE values. This underscores the significance of choosing features with minimal interdependence.

The ROC - AUC Curve for NMIFS with 8 Features is depicted in Figure 5.11.

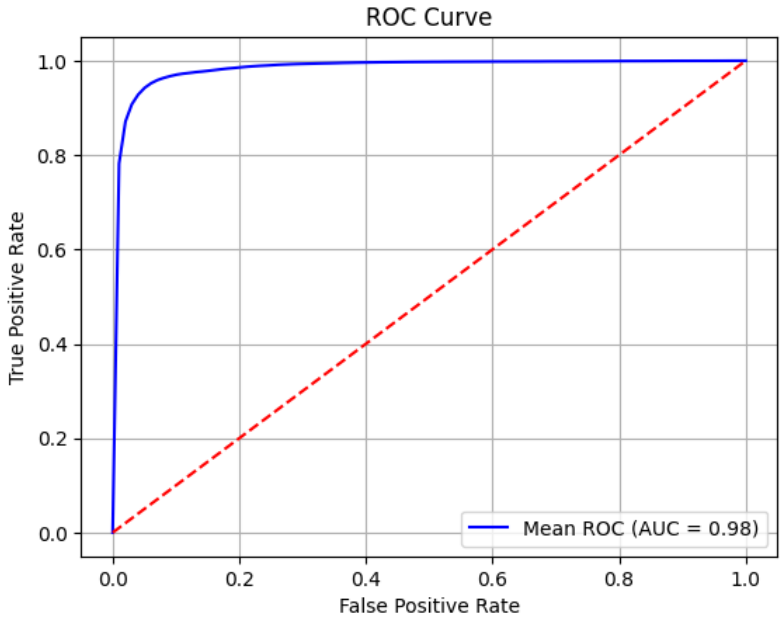


Figure 5.11. ROC Curve for NMIFS with 8 Features using SVM.

The Confusion Matrices of SVM classifier for all 12 features, NMIFS with 8 features, GWO features and Pearson Correlation features are shown in Figure 5.12, Figure 5.13, Figure 5.14 and Figure 5.15 respectively.

```
[[1245   65   22    3    0    0]
 [ 133 1180  114    4    2    0]
 [  17  173 1063   97    9    5]
 [   0   18   59 1208   49    2]
 [   0    1    7   53 1229   22]
 [   0    2    3    4   12 1305]]
```

Figure 5.12. Confusion Matrix of 12 Features using SVM.

```
[[1252   66   14    3    0    0]
 [ 142 1166  122    1    2    0]
 [  14  184 1057   94    8    7]
 [   0   18   55 1218   44    1]
 [   0    2    6   49 1236   19]
 [   0    1    4    1    8 1312]]
```

Figure 5.13. Confusion Matrix of NMIFS with 8 Features using SVM.

```
[[1234   80   20    1    0    0]
 [ 133 1165  129    4    2    0]
 [  19  186 1045   96   12    6]
 [   0   15   77 1176   66    2]
 [   0    0   13   77 1199   23]
 [   0    2    5    2   18 1299]]
```

Figure 5.14. Confusion Matrix of GWO Features using SVM.

```
[[1208  105   20    2    0    0]
 [ 199 1063  165    4    2    0]
 [  25  201 1022   98   12    6]
 [   0   22   83 1136   92    3]
 [   0    4    5  117 1133   53]
 [   0    3    6    2   46 1269]]
```

Figure 5.15. Confusion Matrix of Pearson Correlation Coefficient using SVM.

5.3 Ensemble Learning Model

In order to build the ensemble learning model using three base or 'weak' learners, namely XGBoost classifier, KNN classifier and SVM classifier, the union as well as intersection of features obtained from the mutual information-based feature selection algorithms (MIFS, NMIFS, DMIFS) as described in section **5.1** was expounded.

Table 5.10 given below presents the results of the ensemble learning model both on the union as well as intersection of features.

Table 5.10. Ensemble Model Results on Union and Intersection of Features.

Performance Metrics	Union of 6 Features	Union of 8 Features	Intersection of 6 Features	Intersection of 8 Features
Accuracy	**94.2%**	**94.2%**	85.5%	**93.8%**
MAE	**0.06**	**0.06**	0.17	**0.06**

From the table above, we are presenting evidence that the ensemble learning model has surpassed the accuracy score and MAE value obtained by the use of traditional classification algorithms, particularly when we are training the model on the subset of features selected by performing the union operation on the three feature selection algorithms, namely MIFS, NMIFS and DMIFS. The superior results testify that the usage of ensemble learning technique yields better results compared to single classification algorithms.

The Confusion Matrices of the ensemble classifier model for union of 6 features, union of 8 features, intersection of 6 features and intersection of 8 features are shown in Figure 5.16, Figure 5.17, Figure 5.18 and Figure 5.19 respectively.

```
[[1314    6   14    1    0    0]
 [  38 1313   77    3    2    0]
 [  12  147 1111   81    6    7]
 [   1    7   18 1288   21    1]
 [   0    2    3    7 1288   12]
 [   0    1    3    2    9 1311]]
```

Figure 5.16. Confusion Matrix of Union of 6 Features using Ensemble Model.

```
[[1313    7   13    2    0    0]
 [  47 1304   76    4    2    0]
 [   9  159 1097   84    9    6]
 [   2    3   15 1304   11    1]
 [   0    2    2    7 1296    5]
 [   0    0    2    2    1 1321]]
```

Figure 5.17. Confusion Matrix of Union of 8 Features using Ensemble Model.

```
[[1241   60   29    3    0    2]
 [ 132 1130  161    7    2    1]
 [  35  208  994  105   12   10]
 [   7   25   73 1164   63    4]
 [   2    2   19   83 1164   42]
 [  11   11   14   12   61 1217]]
```

Figure 5.18. Confusion Matrix of Intersection of 6 Features using Ensemble Model.

```
[[1314    6   14    1    0    0]
 [  50 1303   73    5    2    0]
 [  15  147 1102   85    7    8]
 [   2    5   18 1286   24    1]
 [   0    2    4    8 1283   15]
 [   0    1    3    2   11 1309]]
```

Figure 5.19. Confusion Matrix of Intersection of 8 Features using Ensemble Model.

CHAPTER 6
CONCLUSION AND FUTURE SCOPE

6.1 Conclusion of the Present Work

The present work utilized the information theoretic concept of capturing classical correlations amongst features using mutual information for feature selection and traditional machine learning models with hyperparameter tuning for Air Quality Index (AQI) classification using the Air Quality Index (AQI) dataset [16] and achieved an impressive mean accuracy of 93.8%. Compared to other state of the art feature selection methods in the context of our current research, it is shown that the mutual information-based feature selection methods, particularly the NMIFS algorithm consisting of 8 features has performed remarkably well when utilized with XGBoost classifier and SVM classifier and the DMIFS algorithm consisting of 8 features algorithm has performed the best with the KNN classifier. The test accuracy and the test MAE scores are comparable, even exceeding the values obtained if all 12 original features are considered together. The ensemble learning model built on the concept of stacking using majority voting with three weak learners consisting of XGBoost, KNN and SVM classifiers resulted in an accuracy of 94.2% averaged over 10 simulations.

This implies the importance of selecting features which have the least amount of dependence amongst each other, which is a crucial problem that is eliminated by exploiting the concept of information theoretic feature selection strategies.

6.1 Scope for Future Work

While the proposed methodology demonstrates commendable strengths in AQI classification, it's crucial to recognize inherent limitations. Relying on traditional machine learning models may present challenges in interpretability and scalability. Introducing a feature selection based on mutual information, although beneficial for presenting optimal features may result in information overload and can escalate computational complexity. Future exploration should consider advanced architectures tailored for enhancing scalability and reduced computational training time. Robust validation on varied datasets and collaboration with environmental experts would bolster the model's adaptability. Addressing these considerations holds promise for refining AQI classification methodologies, enhancing accuracy for practical applications in controlling air pollution.

BIBLIOGRAPHY

[1] Teologo Jr, A. T., Dadios, E. P., Baldovino, R. G., Neyra, R. Q., & Javel, I. M. (2018). Air Quality Index (AQI) Classification using CO and NO₂ Pollutants: A Fuzzy-based Approach. In *Proceedings of TENCON 2018 - 2018 IEEE Region 10 Conference* (pp. 1-6). Jeju, Korea: IEEE. DOI: 10.1109/TENCON.2018.8629733.

[2] Suman, A. (2020). Air quality indices: A review of methods to interpret air quality status. *Materials Today: Proceedings*, 34, 863–868. doi:10.1016/j.matpr.2020.07.141.

[3] K.M.O.V.K. Kekulanadara, B.T.G.S. Kumara, Banujan Kuhaneswaran. "Machine Learning Approach for Predicting Air Quality Index." *2021 International Conference on Decision Aid Sciences and Application (DASA)*, 2021, pp. 622-624. DOI: 10.1109/DASA53625.2021.9682221.

[4] Hastie, T., Tibshirani, R., & Friedman, J. (2009). "The Elements of Statistical Learning: Data Mining, Inference, and Prediction (2nd ed.)," Springer.

[5] Han, J., Kamber, M., & Pei, J. (2022), "Data Mining: Concepts and Techniques (4th ed.)," Morgan Kaufmann.

[6] Gupta, S., Gajare, S., Moledina, A., Kate, M., & Athavale, S. (2023). "Air Quality Prediction Using Machine Learning: A Comparative Study," in Proc. 6th Int. Conf. Advances in Science and Technology (ICAST), pp. 485-491. doi: 10.1109/ICAST59062.2023.10454930.

[7] Natarajan, S. K., Shanmurthy, P., Arockiam, D., et al. (2024). "Optimized machine learning model for air quality index prediction in major cities in India." Sci Rep, 14, 6795. https://doi.org/10.1038/s41598-024-54807-1.

[8] Rao, R. (2020). "Air Quality Data in India (2015 - 2020) [Dataset]," Kaggle. https://www.kaggle.com/datasets/rohanrao/air-quality-data-in-india.

[9] Kumar, K., & Pande, B. P. (2023). "Air pollution prediction with machine learning: a case study of Indian cities." Int. J. Environ. Sci. Technol., 20, 5333–5348. https://doi.org/10.1007/s13762-022-04241-5.

[10] Berkani, S., Gryech, I., Ghogho, M., Guermah, B., & Kobbane, A. (2023). "Data Driven Forecasting Models for Urban Air Pollution: MoreAir Case Study," IEEE Access, 11, 133131-133145. doi: 10.1109/ACCESS.2023.3331565.

[11] Wang, J., Jin, L., Li, X., He, S., Huang, M., & Wang, H. (2022). "A hybrid air quality index prediction model based on CNN and attention gate unit," IEEE Access, 10, 113343-113354.

[12] Zhang, Q., Han, Y., Li, V. O. K., & Lam, J. C. K. (2022). "Deep-AIR: A Hybrid CNN-LSTM Framework for Fine-Grained Air Pollution Estimation and Forecast in Metropolitan Cities," IEEE Access, 10, 55818-55820. doi: 10.1109/ACCESS.2022.3174853.

[13] Chen, H., Guan, M., & Li, H. (2021). "Air Quality Prediction Based on Integrated Dual LSTM Model," IEEE Access, 9, 93285-93292. doi: 10.1109/ACCESS.2021.3093430.

[14] Okokpujie, K. (2020). "Air Quality Prediction Using Naive Forecast, Linear Regression and Gradient Boosting Algorithms," IEEE Trans. Neural Networks, 12(2), 134-145.

[15] Nehete, R. (2019). "Random Forest, Decision Tree, and Logistic Regression for Air Quality Index Prediction," Environ. Sci. Technol., 14(3), 234-245.

[16] Wang, J., Xu, W., & Dong, J. (2022). "A novel air quality prediction and early warning system based on combined model of optimal feature extraction and intelligent optimization." Chaos, Soli,.Fract., 158, 112098. https://doi.org/10.1016/j.chaos.2022.112098.

[17] Dey, A., Ghosh, S., & Ientilucci, E. J. (2023). "A combination of mutual and neighborhood information for band selection in hyperspectral images," in Proc. 2023 IEEE Int. Geosci. Remote Sens. Symp. (IGARSS), Pasadena, CA, USA, pp. 6077-6080.

[18] Chakradhar Reddy, K., Nagarjuna Reddy, K., Brahmaji Prasad, K., & Rajendran, P. S. (2021). "The Prediction of Quality of the Air Using Supervised Learning," in Proc. 6th Int. Conf. Communication and Electronics Systems (ICCES), pp. 20-21. doi: 10.1109/ICCES51350.2021.9488983.

[19] Waman Gore, R., & Deshpande, D. S. (2017). "An Approach for Classification of Health Risks Based on Air Quality Levels," in Proc. IEEE Int. Conf. on Computing Communication Control and Automation (ICCUBEA), pp. 58-60. doi: 10.1109/ICCUBEA.2017.8463906.

[20] Abirami, S., & Chitra, P. (2023). "Probabilistic Air Quality Forecasting Using Deep Learning Spatial–Temporal Neural Network," GeoInformatica, vol. 27, pp. 199–235. doi: 10.1007/s10707-022-00479-w.

[21] Batiti, R. (1994). "Using mutual information for selecting features in supervised neural net learning," IEEE Transactions on Neutral Networks, 5, 537-550.

[22] Banit'ouagua, I., Ait Kerroum, M., Hammouch, A., & Aboutajdine, D. (2016). "Band selection by mutual information for hyper-spectral image classification," Int. J. Advanced Intelligence Paradigms, 8(1), 98–118.

[23] Kwak, N., & Choi, C. (1999). "Improved mutual information feature selector for neural networks in supervised learning," in Int Joint Conf. on Neural Networks (IJCNN '99), pp. 1313-1318.

[24] Novovicova, J., & Malik, A. (2005). "Information-Theoretic Feature Selection Algorithms for Text Classification," in Proceedings of International Joint Conference on Neural Networks, pp. 3272-3277. Montreal, Canada. https://doi.org/10.1109/IJCNN.2005.1556450.

[25] Duan, J., Gong, Y., Luo, J., et al. (2023). "Air-quality prediction based on the ARIMA-CNN-LSTM combination model optimized by dung beetle optimizer," Sci Rep, 13, 12127. https://doi.org/10.1038/s41598-023-36620-4.

[26] Zhao, Z., Wu, J., Cai, F., et al. (2023). "A hybrid deep learning framework for air quality prediction with spatial autocorrelation during the COVID-19 pandemic," Sci Rep, 13, 1015. https://doi.org/10.1038/s41598-023-28287-8.

[27] Gupta, V., Kapadia, S., & Bhadane, C. (2023). Time Series Analysis and Forecasting of Air Quality in India. Proceedings of the Fifth International Conference on Electrical, Computer and Communication Technologies (ICECCT). DOI: 10.1109/ICECCT56650.2023.10179673.

[28] Liu, C., Pan, G., Song, D., & Wei, H. (2023). "Air Quality Index Forecasting via Genetic Algorithm-Based Improved Extreme Learning Machine," IEEE Access, 11, 67086-67097. doi: 10.1109/ACCESS.2023.3291146.