FENI UNIVERSITY JOURNAL, 2025, *4*(1), PP. (25-52)
DOI: https://doi.org/10.3329/fuj.v4i1.89656
ISSN: 2518-3869

A Smart Approach to Environmental Toxicology and Monitoring: Artificial Intelligence for Air Pollution Detection in the Coastal Regions of Bangladesh

Rayhanuzzaman*

Abstract

Globally, the contamination of the air by damaging agents presents a critical environmental and public health concern. These pollutants, arising from a spectrum of human and natural origins, impair air quality and create severe hazards for human health, ecological balance, and climatic conditions. Recognizing the intricate nature of air pollution, characterized by various gases and particulate matter, accurate and timely identification is indispensable for devising and enacting effective solutions to protect our planet and its inhabitants. Monitoring air pollution in dynamic places is much critical due to various emissions such as industrial, automotive, biogenic and seasonal fluctuations as well. Traditional air quality sensors often lack the real-time data needed for effective intervention and pollution control. To address this, this research introduces an AI-driven solution for real-time air pollution monitoring, particularly for the coastal region of Bangladesh. The approach utilizes a comprehensive dataset of PM2.5, PM10, and NOx from governmental and regional monitoring stations. Employing supervised learning techniques with machine learning and deep learning techniques, the proposed system can accurately (97%) predict pollution levels in real-time and identify pollution trends. The findings highlight the capability of Artificial Intelligence in enhancing air quality governance and supporting public health decisions in densely populated areas.

Keywords: *Public Health; Air Quality Governance; Pollution Trends; Artificial Intelligence; Real-Time Monitoring; Coastal region; Machine Learning.*

*Lecturer, Department of CSE, International Islamic University Chittagong, Chattogram, Bangladesh, Email: rayhanuzzamanr@gmail.com, Phone: 01722-600566

Introduction

In relation to environmental governance, technological innovations are being used more often to solve critical issues like air pollution. One useful technique is to apply data-analysis techniques for monitoring and forecasting an environmental threat; the other method would focus on characterizing pollution sources for timely intervention and planning. These methods hinge on recent advancements in artificial intelligence allowing them to have practical applications in public health management and strategic relevance.

Although much progress is being made worldwide to ensure the applicability of real-time air quality monitoring systems, coastal areas of Chattogram, a major urban and industrial center in Bangladesh, still face huge limitations. With quick industrialization set against increasing vehicular density and a growing population, air quality continues to degrade in this city. The infrastructure for robust pollution monitoring, along with appropriate public alerts, remains further down the priority list. Lack of structured data collection, real-time feedback mechanisms, and predictive tools for pollution mapping has weak repercussions on timely decision-making and environmental safety.

While pollution monitoring in megacities worldwide has received a considerable amount of attention, local data-driven solutions have not been explored in the city of Chattogram. Addressing this very gap, we propose an air pollution-trend-prediction framework based on machine learning. The aim is to initiate a system that is capable of real-time analysis and prediction of pollution levels by utilizing sensor data integrated with intelligent algorithms for forecasting pollutants like PM2.5, PM10, and NOx.

In summary, pollution monitoring has been an area of vast advancement; unfortunately, the previous attempts only realized near-real-time localized predictive information for events that may happen in cities such as Chattogram in fits and starts. Our study aims to address this gap in pollution monitoring by using advanced ML and DL models in harmony with the unique characteristics of the pollution environment in certain regions, thus yielding a robust, responsive air quality forecasting solution.

Literature Review

Air pollution prediction and monitoring have long remained burning issues in the eyes of environmental researchers and urban planners alike. In the earlier days of environmental data analysis, most activities were focused on static or periodic monitoring methodologies owing to technological constraints. Sensor technologies, together with advances in ML and DL, ushered in a new era of precision and responsiveness, given that such improvements have made a great deal of the burden a little lighter for researchers in recent days. These methodologies have enabled researchers to move from observation into prediction-consideration, into real-time feedback and early warning consideration, especially relevant in rapidly urbanizing and industrial-growth cities like Chattogram.

The effects of air pollution on land-surface temperature were studied entailing the multivariate geo-statistical framework application. This study shows how environmental datasets may complement some respects to give a better overview of pollution dynamics, which is a methodology that would guide our spatial modeling [10].

Seasonality-sensitive forecasting systems were clearly affected by the findings, tracking the air quality variations in post-monsoon studies conducted across Greater Dhaka. Again, these findings strengthen the argument for developing air quality monitoring systems in response to seasonal and meteorological changes; the relevance of which would be even more pronounced in conjunction with our approach [11].

On top of that, this research introduced a model through the incorporation of ML algorithms SVM and decision trees into an IoT infrastructure focusing on devising a consistent model of pollution monitoring in Bangladesh. Their framework is intended to collect the environmental data from sensor nodes, followed by modeling it for predictive insights [5]. Likewise, a health hazard monitoring system was introduced which will combine route mapping and real-time alerts, which is quite similar to our health-directed public alert system [9].

This study was performed with air quality and AQI measures over time in Chattogram. It brought about data on topographical as well as seasonal effects upon air pollution figures. This called for a dynamic and real-time monitoring system [3]. Further contributions include such work led by using thousands of Low-Cost Sensors for rural Bangladesh. Demonstrated how, even in highly resource-constrained settings, one can collect live pollution data [8]. So, such effort can pave the way toward affordable environmental monitoring. The study went further by making this system IoT-integrated for prediction that will use sensor real-time data for predicting future air quality; the research is quite much in our stretch [4].

Perhaps recent studies place even greater emphasis on the integration of all these multi-dimensional data for air quality forecasting. Satellite images were used with Google Earth Engine to study air pollution geospatially during COVID-19, and they contributed to filling in the gaps where limited ground stations are concerned. Through this use of spatiotemporal data, these studies contribute to broader algorithms like ours that seek to integrate ground-level with aerial sources of data [1].

Many studies have revealed the significance of setting up suitable frameworks of air quality monitoring in a populated region. For example, A study was performed on average PM and black carbon levels locally related to urban cities of Bangladesh and put forward very strong indications towards the requirement for continuous air quality monitoring in major cities like Chattogram and Dhaka. This initial work was based on conventional methods of air collection and chemical analysis, and provided very preliminary information for the established spread and intensity of pollution [2].

This literature review section highlights the use of advanced technology like IoT and machine learning for real-time monitoring, emphasizes the need for systems that account for seasonal and geographical changes, and argues for the importance of integrating different data sources (like ground sensors and satellite images) to create a more comprehensive and accurate picture of pollution. This review establishes a clear need for your study, which will build on these findings to create a new model for monitoring air quality.

Proposed Methodology

The air quality data has been sourced from various sources that include national environmental agencies, meteorological stations, and satellites for Chattogram. The data set comprises all the parameters including PM2.5, PM10, NOx, CO, temperature, humidity, and wind speed of various sites and time periods.

Data preprocessing included missing value imputation, Z-score normalization, feature selection, and four different pollution categories, such as Secure, Moderate, Unhealthy, and Extremely Unhealthy. Seasonal trends were captured using time-attributed features such as the time of day and the month.

Following the data pre-processing stage, the dataset was partitioned into two distinct subsets: a training set, comprising 80% of the total data, and a testing set, containing the remaining 20%. This division is a standard practice in machine learning to train the model on a substantial portion of the data and subsequently evaluate its performance on unseen data to assess its generalization capabilities. Machine learning models like KNN, Naïve Bayes, SVM and deep learning models like MobileNetV2, Feed Forward Neural Network (FNN), CNN and ResNet50 were trained using this data. The comparison of these models will be the last stage in identifying the best model for air pollution forecasting.

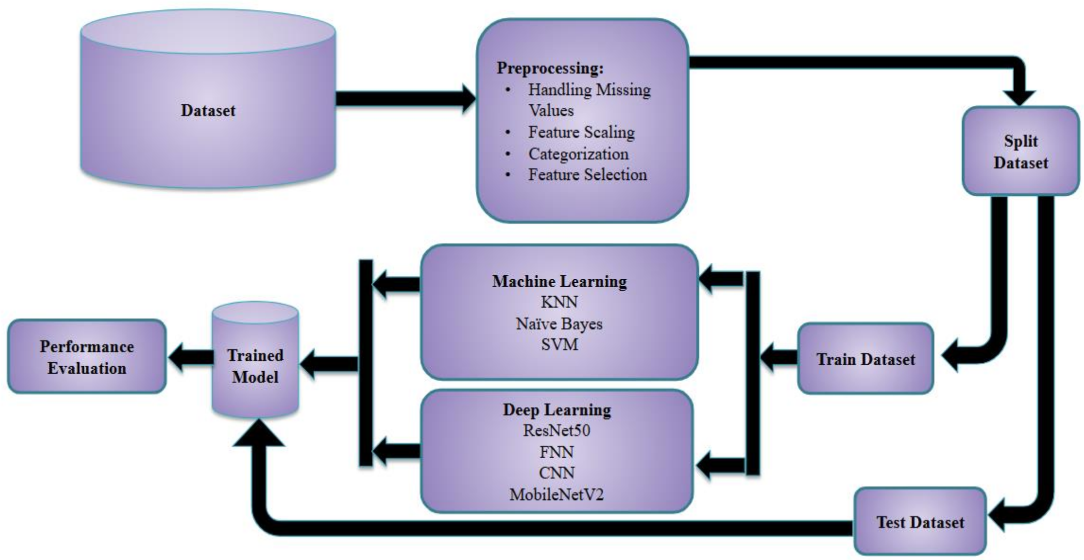

Figure 1: Proposed Methodology

Data Collection

We utilized a public dataset, known as Chattogram Air Quality Index (AQI), that has pollutant data in it like PM2.5, PM10, NOx, CO, and climatological parameters such as temperature, humidity, and wind speed. The dataset contains data of 1000+ readings in various areas of Chittagong. These data points would be divided into both training and testing datasets so that models can really generalize well. Based on pollution readings throughout the day and in various weather conditions, the system can learn different patterns in the air quality. Data is split into training and testing sets, with the latter being held back to test how the model performs on unseen data.

Table1: Labels of certain classes in the dataset

Label	Class
0	CAUTION
1	EXTREMELY UNHEALTHY
2	GOOD
3	MODERATE
4	UNHEALTHY
5	Very UNHEALTHY

Data Preprocessing

In order to ensure precision and uniformity, the dataset undergoes a series of intricate preprocessing steps before the application of machine learning models aimed at predicting the air pollution levels in Chittagong. The dataset encompasses a range of key air pollutants such as PM2.5, PM10, NOx, and CO, in addition to various meteorological factors including wind speed, temperature, and humidity. Through the reduction of noise and the elevation of data quality, appropriate preprocessing intricately enhances the performance of models.

- **Addressing the Complication of Absent Information**

The issue of absent data is tackled through mean imputation, a method that involves substituting any missing values with the average from the pertinent column associated with the missing data. This safeguards the integrity of the data in its entirety and guarantees that models remain unbiased despite any gaps in information.

- **Standardization of Attributes**

Due to the diverse scales of numerous attributes, it is essential that either standardization or normalization must be utilized. We employ Z-score normalization, commonly referred to as standardization, to ensure that all features, such as PM2.5, PM10, temperature, and humidity, are assessed on a uniform scale. Techniques in machine learning, including support vector machines (SVMs) and k-nearest neighbor (KNN) algorithms, possess the capability to manage data with greater efficiency and mitigate biases arising from varying magnitudes due to transformation, thereby enhancing their overall performance.

- **Classifications of Pollution Severity**

The variable representing pollution levels is transformed into various category groupings based on established standards. The system encompasses a range of tiers: Caution, Moderate, Good, Unhealthy, Very Unhealthy, and Extremely Unhealthy.

- **Determining the Features to Incorporate**

Should there exist traits within the dataset that are superfluous or redundant, it stands to reason that the efficacy of the model may suffer as a result. We employ various techniques for selecting features to ensure that we retain only the most significant characteristics that influence air quality. The essential traits encompass the subsequent aspects: Signs of environmental contamination encompass particulate matter 2.5 (PM2.5), particulate matter 10 (PM10), nitrogen oxides (NOx), and carbon monoxide

(CO). The various factors existing in the atmosphere, including wind velocity, humidity levels, and air temperature.

Data Splitting

The dataset is segmented into two distinct subsets that operate independently, facilitating a comprehensive evaluation of the performance of the machine learning model. The model undergoes training with a set that constitutes 80% of the available data, enabling it to grasp the intricate patterns and relationships that are embedded within the dataset. The final 20 percent is designated as the test set, functioning as a stand-in for data that has not been encountered, akin to real-world scenarios. We can obtain an impartial assessment of the trained model's ability to generalize specifically, its skill in producing accurate predictions based on unfamiliar data not encountered during training by evaluating its performance on this reserved test set. Utilizing this partitioning approach on innovative, real-world data yields a more reliable reflection of the model's actual usefulness.

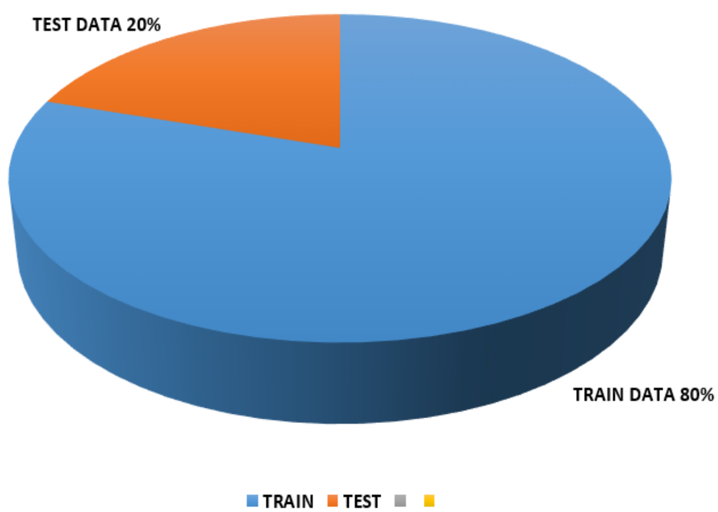

Figure 2: Dataset Splitting

Model Description

Convolutional Neural Network

Convolutional neural networks, often referred to as ConvNets, embody a specific category of deep neural networks that are commonly employed in the evaluation of visual imagery. Rather than depending solely on the conventional method of matrix multiplications typical of standard neural networks, the CNN architecture adopts a distinctive strategy referred to as convolution. Convolutional Neural Networks (CNNs) utilize a mathematical operation known as convolution, which fundamentally merges two functions to examine how one influences the structural properties of the other. An essential benefit of CNNs compared to traditional machine learning methods is their intrinsic capacity to autonomously acquire pertinent features from large datasets. This automated feature extraction intricately simplifies the process by eliminating the need for manual feature engineering, thus improving overall efficiency [6].

Furthermore, the intricate layered characteristic of Convolutional Neural Networks (CNNs) endow them with the essential attribute of translation invariance. This trait enables the networks to reliably discern and retrieve pertinent patterns and features from input data, irrespective of their spatial positioning, orientation, scale, or any translational discrepancies present within the input. A typical architecture of a Convolutional Neural Network is generally made up of four essential components.

- **Convolution Layers**

The primary component of a Convolutional Neural Network is the convolutional layer. The fundamental function, as indicated by its designation, revolves around the intricate mathematical procedure known as convolution. This entails the utilization of a sliding window function, frequently known as a kernel or filter (these designations are often interchangeably employed), traversing the pixel matrix that forms the input image. A layer characterized by convolution generally consists of numerous filters that share the same dimensions, with each filter tailored to identify particular visual patterns present in the image. The patterns in question can encompass a spectrum that includes fundamental elements such as the orientations of edges and the intricacies of curves, extending to more complex attributes like the overarching shapes of objects, including the nuanced bends found in digits.

- **Activation Function**

Following each convolutional operation, a Rectified Linear Unit (ReLU) activation function is applied. This non-linear function is crucial for enabling the network to learn complex, non-linear relationships between the features extracted from the image, thereby enhancing the CNN's ability to recognize a wider range of patterns. Furthermore, the use of ReLU helps mitigate the vanishing gradient problem, which can hinder the training of deep neural networks.

- **Pooling Layer**

The pooling layer in a Convolutional Neural Network functions to extract the most significant features from the preceding convolutional layers' output feature maps. This is accomplished by applying aggregation functions that down sample the spatial dimensions of these feature maps. This reduction in dimensionality not only decreases the memory requirements for training the network but also aids in mitigating overfitting. Common pooling operations include max pooling, which outputs the maximum value within a defined local receptive field; sum pooling, which computes the sum of all values within that field; and average pooling, which calculates the mean value. Importantly, the application of a pooling layer leads to a progressive decrease in the spatial size of the feature maps. The final pooling layer often flattens the resulting multi-dimensional feature map into a one-dimensional vector, making it suitable as input for the subsequent fully connected layers.

- **Fully Connected Layer**

Situated in the concluding stage of a Convolutional Neural Network are the fully connected layers. The input to these layers is the flattened, one-dimensional vector generated by the preceding final pooling layer. Non-linear transformations are introduced through the application of ReLU activation functions. Ultimately, a softmax prediction layer is employed to produce a probability distribution across all potential output classes. The final classification is then determined by selecting the label associated with the highest probability score.

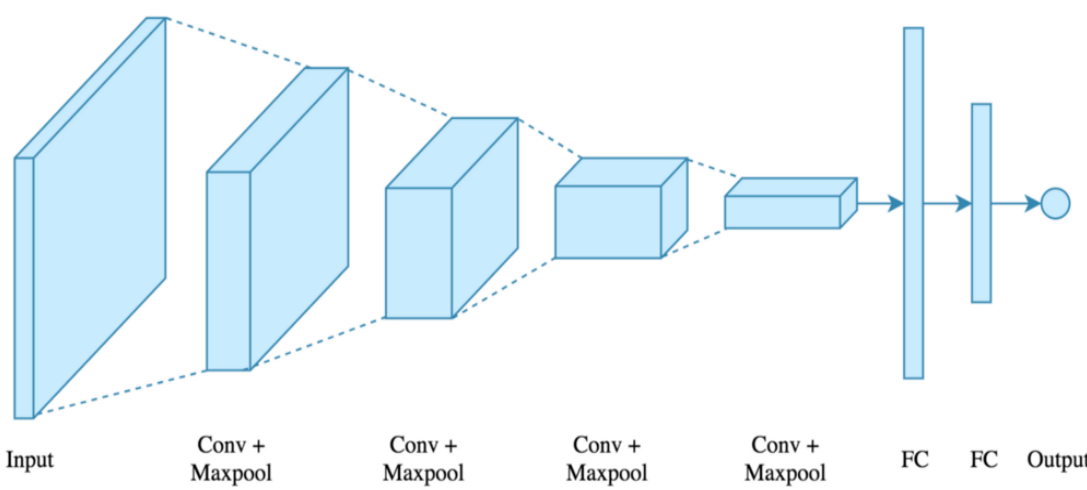

Figure 3: Convolutional Neural Network [12]

MobileNetV2

This architecture delineates a more complex version of a Convolutional Neural Network (CNN) that has been carefully designed for computer vision applications, specifically suited for mobile and embedded platforms. A fundamental aspect of the new MobileNet is its intricate calibration, designed to attain a precise balance between the model's size, quantified in parameters, and its effectiveness in image classification. This makes it especially beneficial for devices that have constrained processing power. The design showcases a complex array of elaborate architectural improvements that enhance its efficiency and effectiveness in classification. The elements include depth-wise separable convolutions, inverted residual connections, a bottleneck layer architecture, linear bottlenecks, and Squeeze-and-Excitation (SE) blocks. Every single one of these elements contributes significantly to keeping computational expenses minimal for the model, while simultaneously guaranteeing that the precision of classification stays remarkably elevated.

- **Depthwise Separable Convolution**

The depth-wise separable convolution, an essential optimization method employed in the MobileNetV2 architecture, aims to markedly decrease the computational expense in contrast to conventional convolution operations. This is accomplished by deconstructing a conventional convolution into two separate sequential processes: initially, a depth-wise convolution, and subsequently, a pointwise (1x1) convolution. The outcome of this factorization leads to a significant decrease in the quantity of floating-point operations necessary, thereby enhancing the model's efficiency, a crucial aspect for implementation on devices with limited resources that are prevalent in our scenario.

- **Residuals that are inverted**

The MobileNetV2 architecture incorporates inverted residual connections as a significant design feature, which is essential for improving the model's accuracy. This method utilizes a bottleneck architecture, deliberately augmenting the number of feature channels prior to the implementation of the computationally efficient depth-wise separable convolutions. The broadening of the feature space allows the model to grasp and embody a wider array of intricate and subtle characteristics from the input data, thus enhancing its overall capacity for representation and ultimately resulting in superior classification performance, even with the kinds of datasets we may face in our local applications.

- **Bottleneck Design**

MobileNetV2 employs a bottleneck architecture that cleverly incorporates 1x1 convolutional layers to reduce dimensionality (the number of feature channels) prior to the application of depth-wise separable convolutions. The decrease in the number of channels leads to a further reduction in the computational resources needed by the network, thereby enhancing its suitability for deployment on devices that possess constrained processing capabilities, a crucial factor for numerous applications in our area. This design decision is crucial for attaining a workable equilibrium between the model's memory usage and its capacity to deliver precise predictions across the diverse categories of visual data we may encounter.

- **Linear Bottlenecks**

MobileNetV2 employs a design strategy featuring linear bottlenecks to address the potential loss of information that may arise during the dimensionality reduction process in the bottleneck layers. Utilizing linear activation functions in these particular bottleneck layers, rather than opting for non-linear alternatives, allows the model to more effectively maintain the fundamental traits of the data. This preservation amplifies the network's ability to hold onto and assimilate even the most nuanced and intricate characteristics found within the input, which could result in enhanced efficacy in tasks pertinent to the kinds of data we examine in this context.

- **Squeeze-and-Excitation (SE) Blocks**

The MobileNetV2 architecture integrates Squeeze-and-Excitation (SE) blocks to enhance the network's ability to represent and leverage features effectively. This set of blocks presents a mechanism that enables the network to flexibly modify the significance of various feature channels. This intricate process of recalibration allows the model to prioritize the features that hold the most significance for the specific task, simultaneously diminishing the influence of features that are of lesser utility. The mechanism of selective attention intricately contributes to a notable enhancement in performance across a spectrum of computer vision tasks, which may, in turn, elevate the accuracy of the models we create for applications pertinent to our local context.

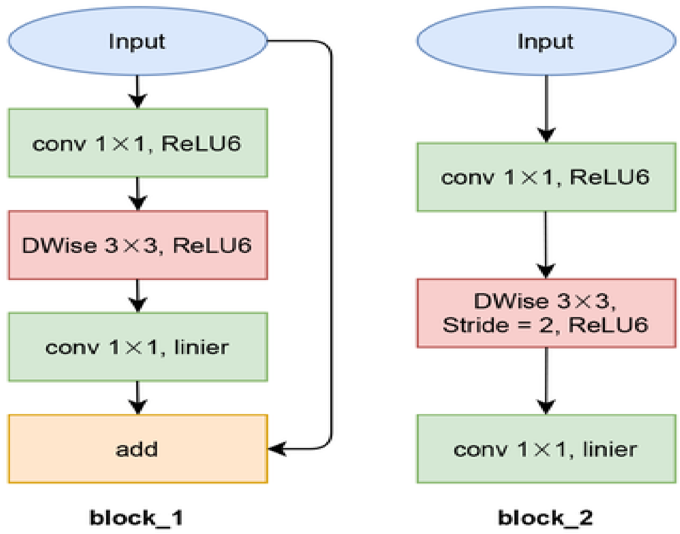

Figure 4: MobileNet V2 [13]

Feed-Forward Neural Network

A Feed-Forward Neural Network (FFNN) is a fundamental type of artificial neural network distinguished by the absence of cyclic connections between its processing nodes. Unlike Recurrent Neural Networks (RNNs), which incorporate feedback mechanisms, data in an FFNN flows in a single direction, progressing sequentially from the input layer, through any intermediate (hidden) layers, to the final output layer. This straightforward, unidirectional flow makes the feed-forward architecture the most basic and foundational structure in neural networks. The historical development of neural networks originates from the perceptron, a crucial early algorithm in machine learning. The perceptron can be viewed as essentially a single-layer FFNN. In this model, a set of input values is fed into the layer, and each input is multiplied by an associated weight. These weighted inputs are then summed to produce a total. If this sum exceeds a predefined threshold (often zero), the output is typically set to one; otherwise, if the sum is below the threshold, the output is usually the negative one. The single-layer perceptron remains a significant feed-forward neural network model widely applied in classification tasks and can also incorporate machine learning capabilities.

A key learning mechanism in these networks is the delta rule, which allows the network to compare its output at each node with the desired target value. Based on this comparison, the network can adjust the weights of its connections during training to produce more accurate outputs. This training process effectively implements gradient descent. For multi-layered FFNNs, the weight update mechanism is similar but is termed back-propagation. In this process, the error at the output layer is propagated back through the network to adjust the weights of connections in each of the preceding hidden layers.

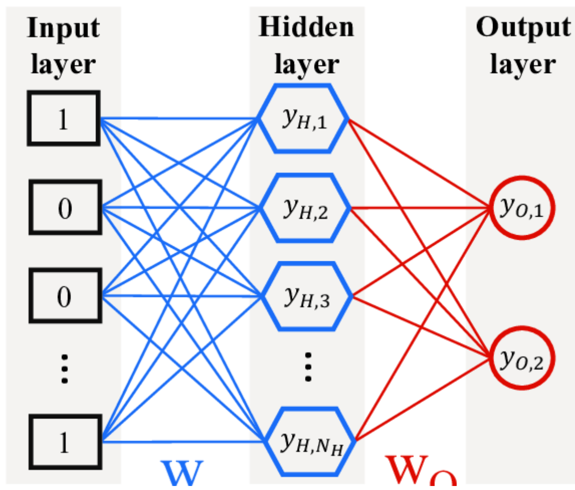

Figure 5: Feed-Forward Neural Network [14]

Deep Residual Network - 50

ResNet-50 stands out as a notable architecture in the realm of Convolutional Neural Networks (CNNs), belonging to the broader category of Residual Networks (ResNet). This collection of models was crafted to tackle the challenges faced during the training of profoundly deep neural networks. Developed by scholars at Microsoft Research Asia, ResNet-50 is noted for its significant complexity and proficiency in the realm of image classification. The ResNet family comprises a range of models characterized by varying layer counts, including ResNet-18 and ResNet-32, while ResNet-50 occupies a middle ground in terms of depth within this collection. Grasping the intricate design and functionalities of models such as ResNet-50 is essential for propelling our efforts in computer vision within Bangladesh.

The central advancement of ResNet is found in its ability to address the challenges posed by the degradation issue that arises in deep neural networks. As the complexity of neural networks escalates, their performance generally reaches a peak and subsequently starts to diminish, not as a result of overfitting, but rather because of the intrinsic challenges associated with optimizing these profoundly layered architectures. ResNet tackles this issue through the incorporation of Residual Blocks. The design of these blocks includes skip connections that create a direct route for information to travel, which significantly alleviates the issue of vanishing gradients and supports the training of much deeper networks.

- **ReLu Activation Function**

Within the framework of the ResNet-50 architecture, a Rectified Linear Unit (ReLU) activation function is employed after each convolutional layer and batch normalization layer. The ReLU function functions in a manner where it permits only positive values to remain intact, while it relegates any negative values to a state of nullity. The incorporation of non-linearity is crucial for allowing the neural network to grasp and represent the intricate, non-linear connections found within the input image data, thus improving its capacity to identify elaborate patterns pertinent to the image classification endeavors we may pursue in our research here.

- **Convolution Layers with Bottlenecks**

In the ResNet-50 architecture, a conventional convolutional block is generally made up of three consecutive convolutional layers, with each layer succeeded by a batch normalization layer and a ReLU activation function. The first layer in this particular arrangement frequently utilizes a filter measuring 1x1. This 1x1 convolution functions to diminish the quantity of feature channels present in the input, thereby compressing the data and enhancing computational efficiency, all while striving to preserve the most significant information. The following layer, characterized by its convolutional nature, typically employs a 3x3 filter, meticulously crafted to seize local spatial relationships and discern patterns embedded within the feature maps. Ultimately, the third layer within the block employs an additional 1x1 filter to restore the channel count to its original dimensionality prior to the summation of the block's output with that of the shortcut connection. This design facilitates a complex interplay of feature processing, enabling the acquisition of both channel-wise and spatial attributes of the input data, which is essential for the nuanced image analysis required in our research and applications here in Bangladesh.

- **Skip Connection**

The fundamental aspect of a typical residual block lies in the shortcut connection, facilitating the immediate addition of the block's input to the output generated by its convolutional layers. This alternative route is essential for maintaining and transmitting crucial information from earlier layers as it moves through the more profound sections of the network. This holds significant relevance in scenarios where the layers of convolution within a designated residual block may encounter difficulties in acquiring novel, distinguishing characteristics. ResNet-50 employs intricate bottleneck residual blocks that intricately merge convolutional layers crafted for feature extraction with shortcut connections, ensuring the preservation of information flow. Moreover, within these bottleneck blocks, there exists a layer designed to diminish the dimensionality of the feature maps. This architectural design facilitates ResNet-50 in addressing the vanishing gradient issue, thereby permitting the effective training of exceedingly deep networks and attaining elevated accuracy levels in image classification tasks pertinent to our research and applications within our local context.

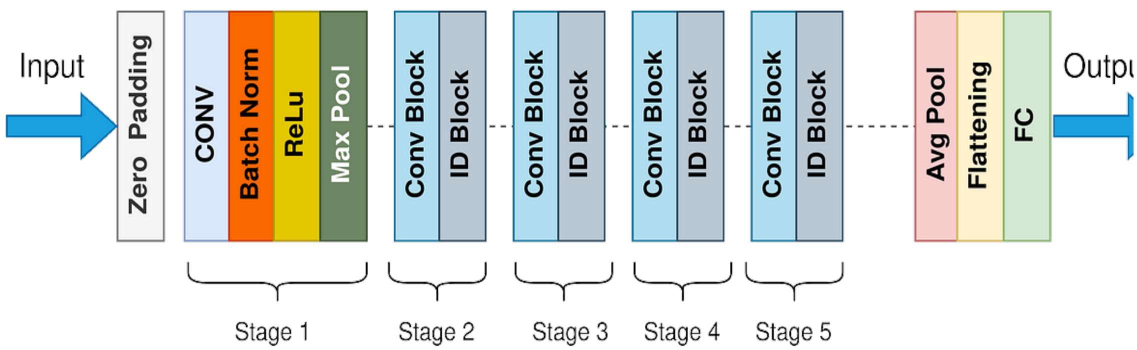

Figure 6: Deep Residual Network – 50 [15]

Naïve Bayes

The Naïve Bayes classifier serves as a supervised machine learning method frequently utilized for the intricate tasks of text classification. The approach it employs for classification is intricately linked to the underlying principles of probabilistic theories. Within the realm of generative learning models, Naïve Bayes operates by intricately capturing the probability distribution of input features that relate to each distinct class. In contrast to discriminative classifiers such as logistic regression, it does not successfully acquire the ability to distinctly assess the importance of each feature. Often referred to as a probabilistic classifier, Naïve Bayes is fundamentally rooted in the principles of Bayes' Theorem. Understanding this technique necessitates a grasp of basic Bayesian statistics, as the core of Naïve Bayes hinges on this theorem, commonly known as Bayes' Rule, which facilitates the inversion of conditional probabilities. Keep in mind that conditional probability assesses the likelihood of an event transpiring, contingent upon the prior occurrence of another event.

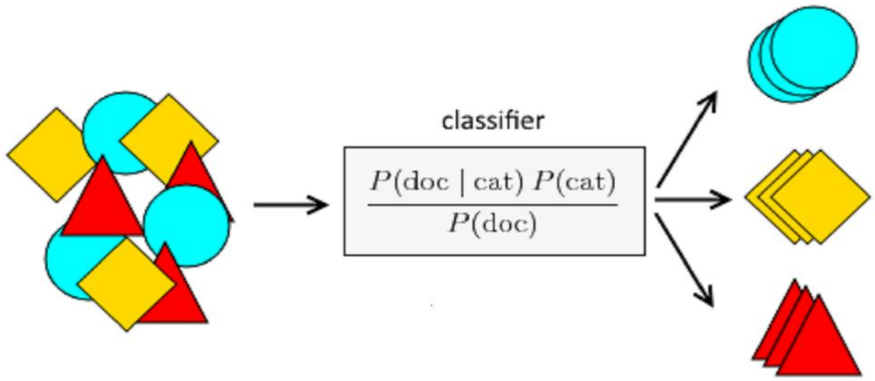

Figure 7: Naïve Bayes [16]

K-Nearest Neighbour

The k-nearest neighbors (KNN) algorithm represents a non-parametric, supervised learning approach that finds application in both classification and regression challenges. It functions by employing the notion of closeness, which suggests that data points situated near one another tend to exhibit greater similarity. This method stands as a cornerstone, frequently employed for both classification and regression tasks within the realm of contemporary machine learning. KNN is primarily employed for classification challenges, although it can also be applied to address both categories of issues. This is due to the premise that data points exhibiting analogous characteristics tend to congregate within the feature space. The value of 'k' in KNN stands as a pivotal parameter, dictating the count of neighboring data points that are factored in when generating a prediction for a specific instance. The fundamental traits of the input dataset play a crucial role in ascertaining the optimal value of "k." In scenarios where the data is rife with considerable noise or when numerous outliers are present, opting for a higher value of 'k' may serve as a beneficial approach to refine prediction boundaries and mitigate the impact of those errant data points that introduce noise. Conversely, opting for a 'k' that is excessively large may lead to underfitting. Underfitting occurs when the model adopts an excessively broad approach, leading to a diminished ability to detect nuances in localized data patterns.

Statistical Methods for Selecting ‘k’:

- **Cross-Validation:** K-fold cross-validation is a reliable method for determining the optimal k. This entails dividing the data into k subgroups, training the model on some of them, testing it on the rest, and repeating the process for each subset. The value of k that yields the highest average validation accuracy is typically the best choice.

- **Elbow approach:** The elbow approach involves plotting the model's error rate or accuracy for various values of k. When we raise k, the error usually decreases initially. However, beyond a certain threshold, the mistake rate begins to decline more gradually. The point where the curve forms an "elbow" is considered the best k.

- **Odd Values for k:** It is also recommended to use an odd number for k, particularly in classification jobs, to avoid ties when determining the majority class.

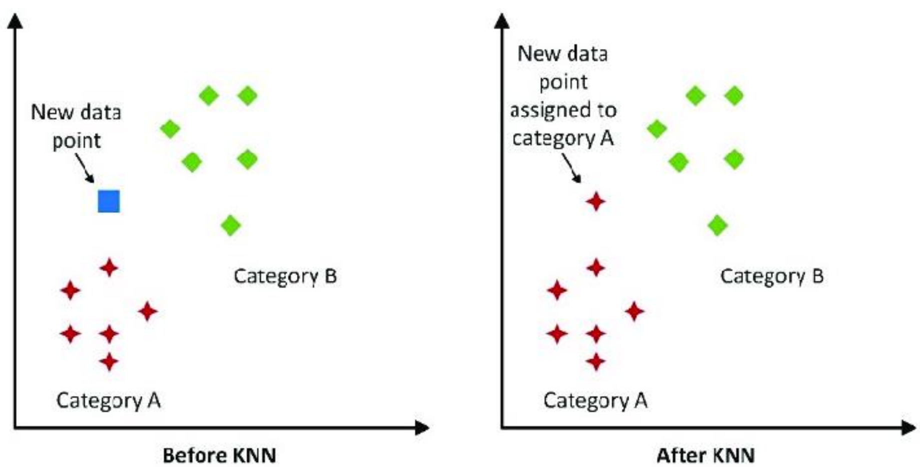

Figure 8: K-Nearest Neighbour [17]

Support Vector Machine

A Support Vector Machine (SVM) is a supervised machine learning algorithm widely used for classification problems. Its core principle involves finding the best decision boundary – which can be a line, a plane, or a more complex hyperplane in a high-dimensional space – that optimally separates data points belonging to different classes. This optimal boundary is determined by maximizing the margin, which is the distance between the hyperplane and the nearest data points from each class. The dimensionality of this separating hyperplane is directly related to the number of features in the input data. Since multiple hyperplanes might be able to separate the classes, SVMs focus on finding the one that provides the largest margin. This maximization strategy leads to a more robust model that is better at generalizing to new, unseen data and making accurate classifications. The data points that are closest to this optimal hyperplane and essentially define the margin are known as support vectors. A key strength of SVMs lies in their ability to handle both linearly and non-linearly separable data. When the data cannot be separated by a straight line or plane, SVMs employ kernel functions. These functions project the data into a higher-dimensional space where a linear separating hyperplane can be found. This technique is known as the "kernel trick," and the specific kernel function used (such as linear, polynomial, Radial Basis Function (RBF), or sigmoid) is selected based on the characteristics of the dataset and the specific requirements of the classification task, which is an important consideration in our research and applications here in Bangladesh.

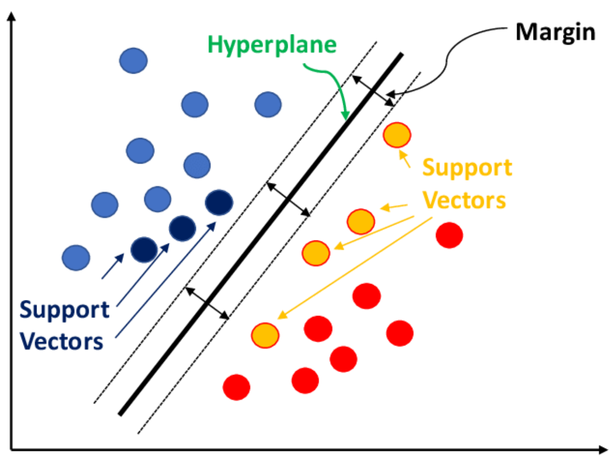

Figure 9: Support Vector Machine [18]

Result Analysis

Following the rigorous groundwork laid in the prior sections, we have arrived at the most important point of result analysis. This portion acts as a furnace for transforming painstakingly obtained and meticulously sorted raw data into relevant insights and evidence-based conclusions. Here, we go beyond basic description to explore the subtle patterns, correlations, and trends buried in our findings.

The process of research analysis is more than just using statistical tools or computer approaches; it necessitates a deliberate and methodical strategy led by the research questions and hypotheses that have defined this investigation. This section will therefore investigate the individual approaches used to deconstruct the acquired data, highlighting the logic behind their selection as well as their inherent strengths and limitations. The insights gained from this analysis will serve as the foundation for future discussions and conclusions, directly addressing the study's primary objectives and contributing to a better knowledge.

Training machine learning and deep learning models entails giving them labeled data, which allows them to discover the underlying patterns and correlations between features and target variables. This learning process is aided by an optimization algorithm that iteratively adjusts the model's internal parameters to minimize a defined loss function, which quantifies the difference between the model's predictions and actual values. During training, it is critical to track the model's performance on both the training data and a separate validation dataset. This aids in detecting overfitting, in which the model learns the training data too well and performs badly on unknown data. Obtaining graphs is an important aspect of the monitoring and assessment process. Key graphs include:

- **Loss Curves:** Plotting the loss function value across training epochs (iterations) for both the training and validation sets. These curves allow to visualize learning progress and indicate when the model begins to overfit (validation loss increases while training loss decreases).

- **Accuracy/Metric Curves:** Similar to loss curves, they show the chosen evaluation metric (e.g., accuracy, precision, recall, F1-score) over epochs for both training and validation sets. They demonstrate how well the model generalizes to previously unexplored data.

- **Confusion Matrix Visualization:** Representing the confusion matrix as a heatmap provides a clear visual summary of the model's classification performance, displaying the numbers of true positives, true negatives, false positives, and false negatives.

- **ROC Curve (Receiver Operating Characteristic):** For binary classification, this graph compares the True Positive Rate (Recall) to the False Positive Rate at various threshold levels. The area under the ROC curve (AUC) 1 represents the model's ability to differentiate between two classes. We may adapt it to evaluate performance in the case of multiclass classification as well.

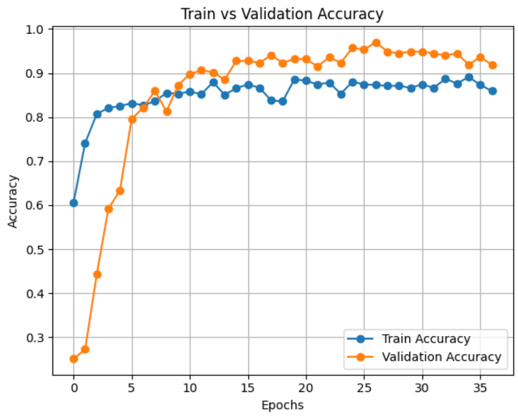

Figure 10: Train-validation Accuracy (MobileNetV2)

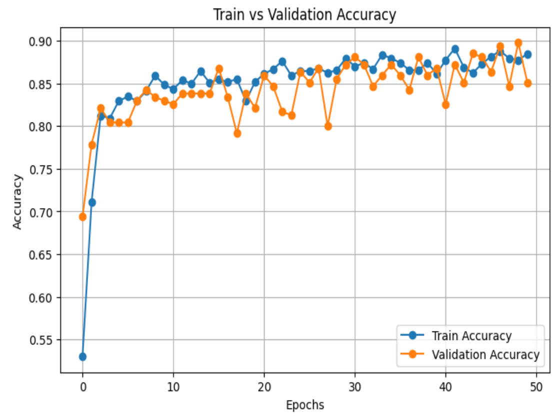

Figure 11: Train-validation Accuracy (ResNet50)

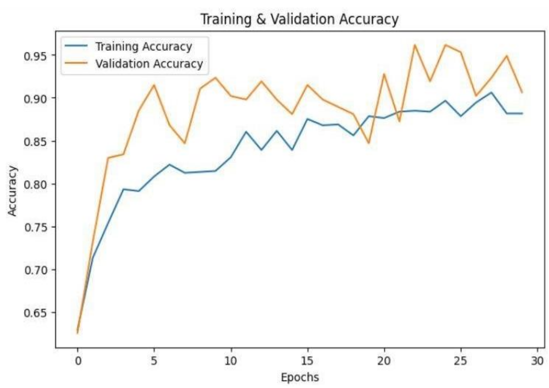

Figure 12: Train-validation Accuracy (CNN)

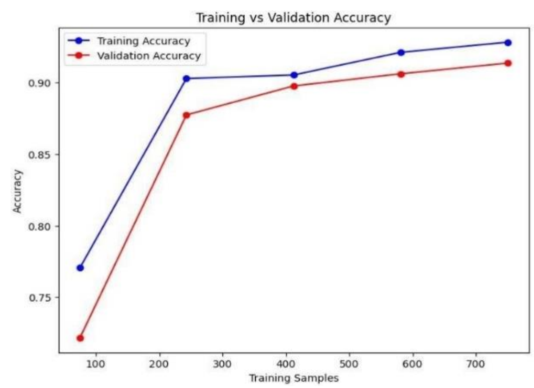

Figure 13: Train-validation Accuracy (SVM)

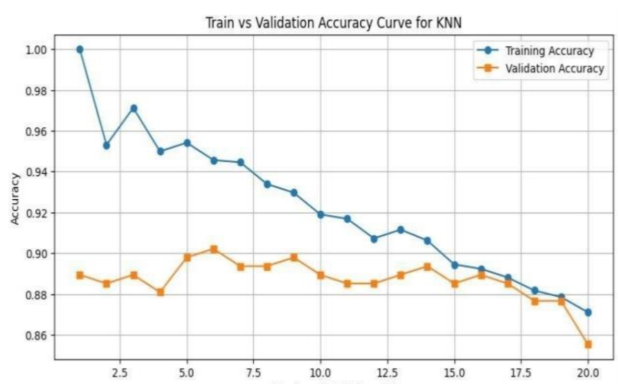

Figure 14: Train-validation Accuracy (KNN)

The plot exhibits the amount of accuracy in training and validation of a model established over several epochs, highlighting the progress of learning. Accuracy underwent a considerable hike at the initial stage. At some points, validation accuracy hit the ceiling and grew to exceed that of training, a strong symbol of generalization. The following portion of the epochs (MobileNetV2) exhibited a plateau on both curves with validation accuracy fluctuating and reaching and maintaining an average of around 97%, while training accuracy leveled off at slightly lower average values of around 90%. The two closely sitting and consistent curves testify well against overfitting and good performance on unseen data.

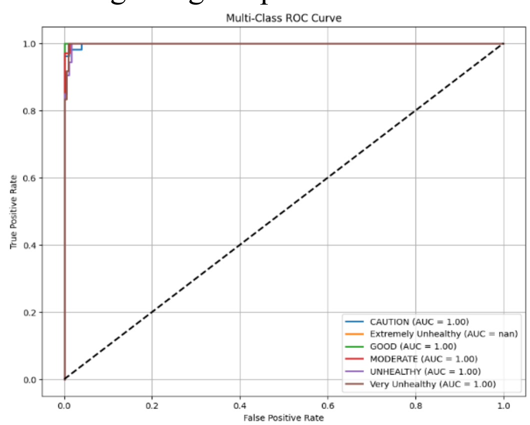

Figure 15: Multi-class ROC (MobileNetV2)

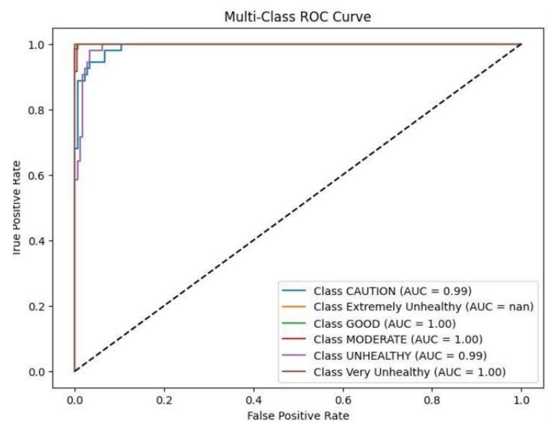

Figure 16: Multi-class ROC (CNN)

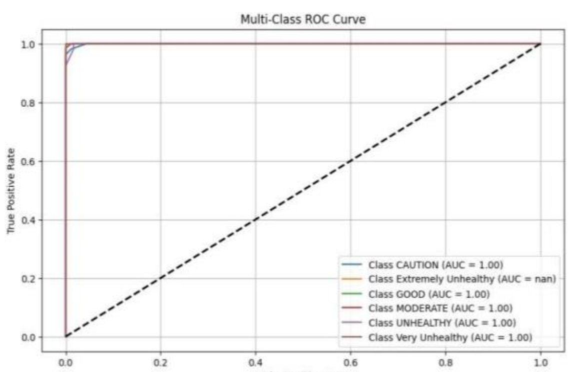

Figure 17: Multi-class ROC (KNN)

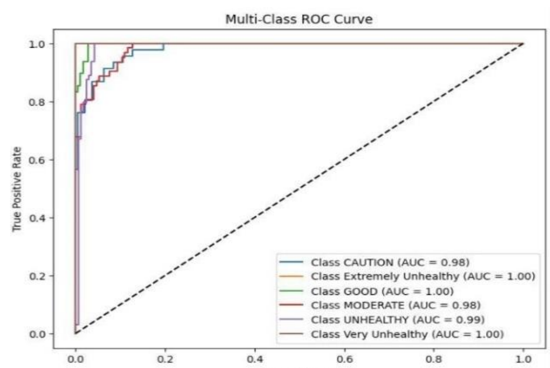

Figure 18: Multi-class ROC (SVM)

The multi-class ROC curve illustrates the proficiency of a classification model, covering the six air quality categories: CAUTION, GOOD, MODERATE, UNHEALTHY, Very Unhealthy, and Extremely Unhealthy. Each curve represents the trade-off between the true positive rate and false positive rate for one class, being all curves visible hugging the top-left corner for near-perfect classification. The AUC (Area Under Curve) values for all classes, with the exception of Extremely Unhealthy, are 1.00, showing a very good performance of the model; however, Extremely Unhealthy shows an AUC of NaN, which is indicative of the probably missing instances for that class in the test set. This shows a need for class balancing to obtain a comprehensive evaluation of the model.

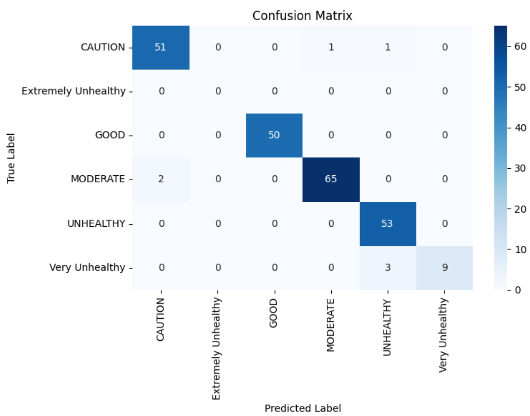

Figure 19: Confusion Matrix (MobileNetV2)

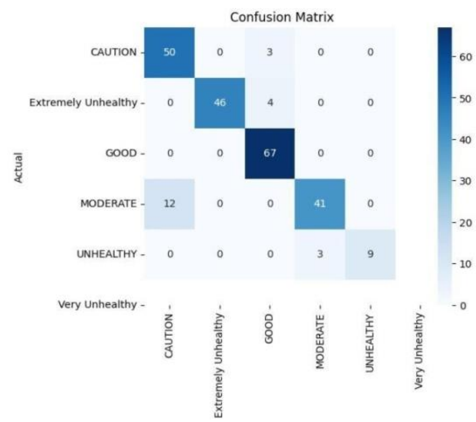

Figure 20: Confusion Matrix (CNN)

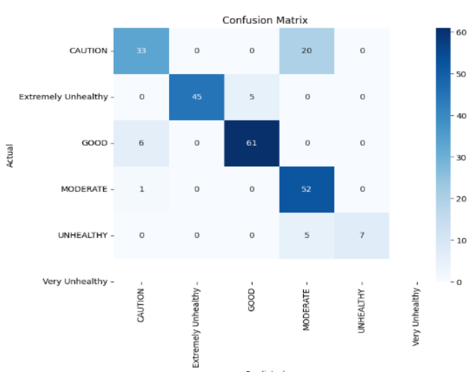

Figure 21: Confusion Matrix (KNN)

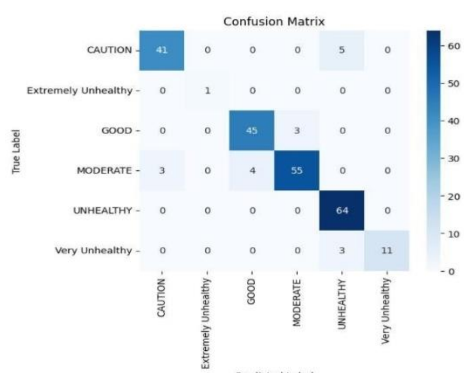

Figure 22: Confusion Matrix (SVM)

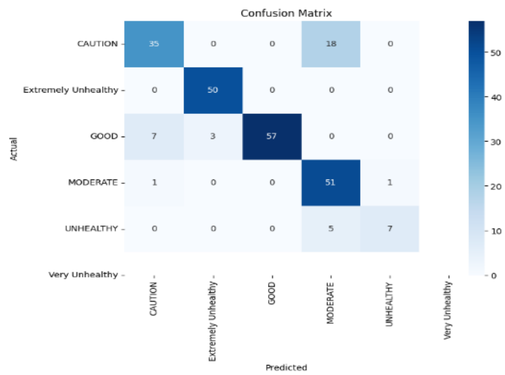

Figure 23: Confusion Matrix
(ResNet50)

Evaluation Metric

The confusion matrix is a fundamental tool for evaluating the performance of classification models [7]. It consists of four key terms:

- **True Positives (TP):** The model demonstrated accurate prediction of the positive class.
- **True Negatives (TN):** The model accurately classified the instance as negative.
- **False Positives (FP):** This instance represents a Type I error, where the model incorrectly predicted a positive outcome.
- **False Negatives (FN):** A Type II error, or false negative, occurred; the model incorrectly classified the instance as belonging to the negative class. From these values, several crucial evaluation metrics can be calculated:
- **Accuracy:** The total proportion of correct predictions made by the model.
  Accuracy = (TP+TN)/(TP+TN+FP+FN)
- **Precision:** Determine the proportion of true positive instances among all instances classified as positive by the model.
  Precision=TP/(TP+FP)
- **Recall (Sensitivity or True Positive Rate):** Calculate the ratio of true positives to the total number of actual positive instances.
  Recall= TP/(TP+FN)
- **F1-Score:** To obtain a balanced performance evaluation, particularly for imbalanced datasets, we use the F1-score, which is the harmonic mean of precision and recall.
  F1-Score=2×{(Precision×Recall)/(Precision+Recall)}

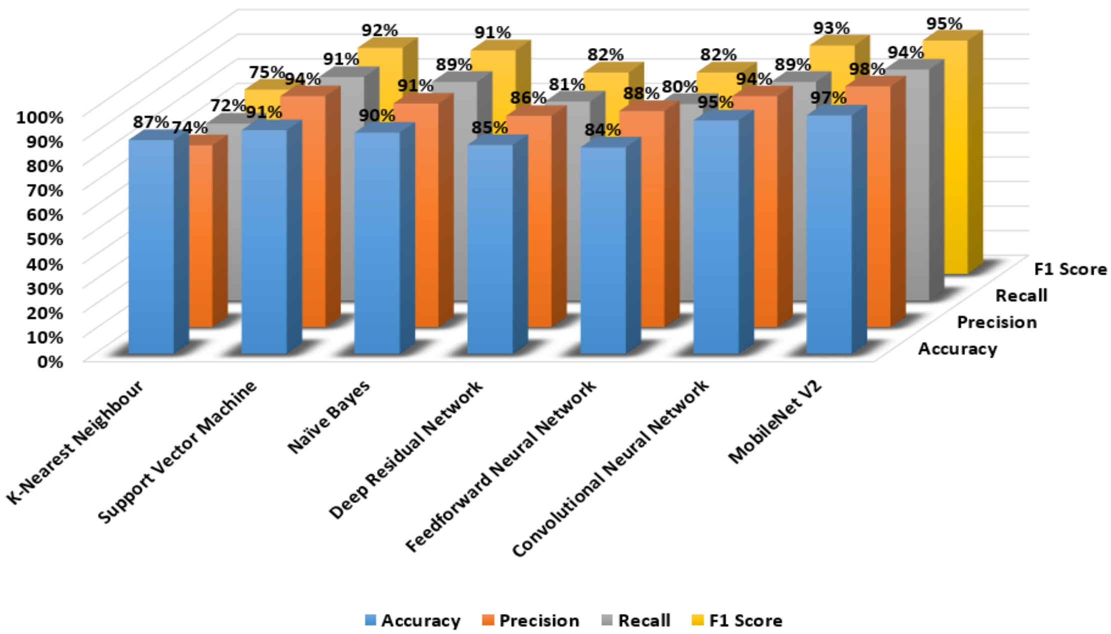

Figure 24: Bar diagram depicting the Accuracy, Precision, Recall, F1 Score of the utilized models.

Among all the used machine learning and deep learning models it was observed that MobileNet V2 having an accuracy of 97%, precision 98%, recall 94% and F1 Score 95% outperformed all the rest of the models.

Future Work

This work aims to improve air quality prediction and monitoring systems using machine learning approaches. The next phase is to improve generalization for extreme air quality classifications such as "Extremely Unhealthy" or "Very Unhealthy". In the coming times, we will concentrate on improving the models' performance to attain greater accuracy and computational efficiency. We will integrate sophisticated techniques, such as hybrid models and data augmentation, to improve model reliability and effectively handle diverse ambient data. The results emphasize deep learning's promises in environmental analytics while indicating that model parameter tuning, increasing dataset size, or investigation of ensemble methods might further improve the accuracy of predictions. Incorporating temporal and spatial features, along with alternative activation functions, would probably yield richer insights. In the future, model selection and architectural customization will remain fundamental to effective, scalable, and context-relevant systems for pollution forecasting.

Conclusions

A diligent artificial intelligence approach has been proposed to predict air pollution trends in the coastal region of Bangladesh. Most traditional studies were conducted using classical statistical models or local-level assessments, but this research incorporates deep-learning techniques to capture the complex nonlinear dynamics of environmental data. Different models, including MobileNetV2, ResNet50, and Feedforward Neural Networks, were deployed, and each model introduced its new perspective towards prediction. These architectures have proven to be successful and popular in image and pattern recognition applications, brought considerable success when adapted to pollution data patterns, proving their usability in applications relatively different from their core domain. The obtained results tend to be relevant to previous investigations and can strengthen the arguments for these models in the area of environmental trend prediction. However, varying performance of models as per network depth, choice of features, and configurations in hyper-parameters indicated the importance of model tuning to achieve superior results. The findings thus confirm deep learning as a valid methodology for handling multivariate datasets typically associated with air quality monitoring systems. They also pave the way for real-time integration of advanced sensors to increase data richness and responsiveness at future stages.

References

1. Haq, S. S. Hasan, M. Z. Rahman, D. Das and A. Ullah, "IoT Based Air Quality and Weather Monitoring System with Android Application," *2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh*, **2022**, pp. 24-28, doi: 10.1109/ICISET54810.2022.9775870.

2. B.A. Begum, G. Saroar, M. Nasiruddin, S. Randal, B. Sivertsen and P.K. Hopke. “Particulate Matter and Black Carbon Monitoring at Urban Environment in Bangladesh.” *Nuclear Science and Applications*, Vol. 23. No. 1&2, **2014**.

3. Mir Md. Mozammal Hoque, Md. Moshiuzzaman Khan, Md. Eusuf Sarker, Md. Nuralam Hossain, Md. Sirajul Islam, Md.Mehedi Hasan Khan, Manik Shil, Md. Nazirul Islam Sarker. “Assessment of Seasonal Variations of Air Quality and AQI Status: Evidence from Chittagong, Bangladesh.” *Indonesian Journal of Environmental Management and Sustainability*, Vol. 6 No. 3, **2022**.

4. M. Rakib, S. Haq, M. I. Hossain and T. Rahman, "IoT Based Air Pollution Monitoring & Prediction System," *2022 International Conference on Innovations in Science, Engineering and Technology (ICISET), Chittagong, Bangladesh*, **2022**, pp. 184-189, doi: 10.1109/ICISET54810.2022.9775871.

5. Md Monirul Islam, Shalah Uddin Perbhez Shakil, Nasim Mahmud Nayan, Mohammod Abul Kashem, and Jia Uddin. "Air Pollution Monitoring Using IoT and Machine Learning in the Perspective of Bangladesh." *Annals of Emerging Technologies in Computing*, **2024**.

6. M. M. Hasan, H. J. Poly, Rayhanuzzaman and A. Y. Srizon, "Enhanced DNA Sequence Classification Through Hyperparameter Optimized Convolutional Neural Networks," *2025 International Conference on Electrical, Computer and Communication Engineering (ECCE), Chittagong, Bangladesh*, **2025**, pp. 1-6, doi: 10.1109/ECCE64574.2025.11013923.

7. Rayhanuzzaman, T. Mahmud, U. K. Das, S. R. Naher, M. S. Hossain and K. Andersson, "Investigating the Effectiveness of Deep Learning and Machine Learning for Bangla Poems Genre Classification," *2024 4th International Conference on Intelligent Technologies (CONIT), Bangalore, India*, **2024**, pp. 1-6, doi: 10.1109/CONIT61985.2024.10627555.

8. Sanjeev Delwar. "Characterizing Particulate Air Pollution in Rural Areas of Bangladesh Using a Network of Low-Cost Sensors." *Department of Civil Engineering Bangladesh University of Engineering and Technology Dhaka, Bangladesh*, **2023**.

9. Sayem, A., Adit, M., Shadman Shahriar, Y., Shams, R., & Iqbal, M. A. (**2023**). Design a monitoring system to measure health hazards through route mapping using air and weather quality. Brac University. https://hdl.handle.net/10361/23853

10. Sk Nafiz Rahaman, Jake Nelson, Al Artat Bin Ali, Nishat Shermin, Narcisa G. Pricope, Abdulla Al Kafy, Md Shahaduzzaman Sabuj, and Sharmin Sultana Toa."A Multivariate Geostatistical Framework to Assess the Spatio-Temporal Dynamics of Air Pollution and Land Surface Temperature in Bangladesh." *Earth Systems and Environment*, **2024**, https://doi.org/10.1007/s41748-024-00418-9.

11. S. M. Sium, Afrin Sharabony, and Kazi Md. Fazlul Haq. **2024**. "Comparative Evaluation of Post-Monsoon Crossroads Air Quality Variations in Major Cities of

the Greater Dhaka Region." *Journal of Environmental Impact and Management Policy* 4, no. 2: 1-18. https://doi.org/10.55529/jeimp.42.1.18.

12. Malaria Cell Detection Model - Scientific Figure on ResearchGate. Available from: *https://www.researchgate.net/figure/Flowchart-of-Sequential-Model_fig2_353092069* [accessed **19 May 2025**]

13. DCNet: Noise-Robust Convolutional Neural Networks for Degradation Classification on Ancient Documents - Scientific Figure on ResearchGate. Available from: *https://www.researchgate.net/figure/The-basic-architecture-of-MobileNetV2-and-ShuffleNet_fig4_353233191* [accessed **19 May 2025**]

14. The principle of learning sign rules by neural networks in qubit lattice models - Scientific Figure on ResearchGate. Available from: *https://www.researchgate.net/figure/The-feed-forward-neural-network-FNN-is-designed-to-learn-the-sign-rules-for-a-quantum_fig1_368304372* [accessed **21 May 2025]**

15. RWHO: A hybrid of CNN architecture and optimization algorithm to predict basal cell carcinoma skin cancer in dermoscopic images - Scientific Figure on ResearchGate. Available from: *https://www.researchgate.net/figure/ResNet-50-architecture_fig1_381505174* [accessed **21 May 2025**]

16. Comparative Study of Machine Learning Algorithms for Phishing Website Detection - Scientific Figure on ResearchGate. Available from: *https://www.researchgate.net/figure/Naive-Bayes-algorithm-21_fig4_374387841* [accessed **21 May 2025**]

17. On Hyperparameter Optimization of Machine Learning Methods Using a Bayesian Optimization Algorithm to Predict Work Travel Mode Choice - Scientific Figure on ResearchGate. Available from: *https://www.researchgate.net/figure/The-Schematic-system-of-the-KNN-model_fig3_368968650* [accessed **22 May 2025**]

18. Identification of flow regime in a bubble column reactor with a combination of optical probe data and machine learning technique - Scientific Figure on ResearchGate. Available from: *https://www.researchgate.net/figure/Support-Vector-Machine-visualization_fig5_332248436* [accessed **22 May 2025**]