Republic of Iraq
Ministry of Higher Education and Scientific Research
University of Babylon
College of Information Technology
Department of Information Network

Intelligent air quality classification and monitoring system using machine learning techniques

A Thesis

Submitted to the Council of the College of Information Technology for Postgraduate Studies of the University of Babylon in Partial Fulfillment of the Requirements for the Degree of Master in Information Technology-Information Networks

By

HUDA KADEM ALWAN HASSON

Supervised by

Prof. Dr. Ghidaa A. Al-Sultany

Lect. Dr. Saba Mohammed Hussein

2023 A.D. **1445 A.H.**

(وَيَسْأَلُونَكَ عَنِ الرُّوحِ ۖقُلِ الرُّوحُ مِنْ أَمْرِ رَبِّي
وَمَا أُوتِيتُم مِّنَ الْعِلْمِ إِلَّا قَلِيلًا)

صدق الله العظيم

سورة الاسراء ٨٥

Supervisor Certification

I certify that the thesis entitled (**intelligent air quality classification and monitoring system using machine learning techniques)** Was prepared under our supervision at the Department of Information Networks/ College of Information Technology/ University of Babylon as partial fulfillment of the requirements of the degree of Master in Information Technology-Information Networks.

Signature:
Supervisor Name: Prof. Dr. Ghidaa A. Al-Sultany
Date: / /2023
Signature:
Co-Supervisor Name: Dr. Saba Mohammed Hussein
Date: / /2023

The Head of the Department Certification

In view of the available recommendations, I forward the thesis entitled "**intelligent air quality classification and monitoring system using machine learning techniques)** Was prepared under our supervision at the Department of Information" for debate by the examination committee.

Signature:
Assist.Prof. Dr. Alharith A. Abdullah
Head of Information Networks Department
Date: / /2023

Certification of the Examination Committee

We hereby certify that we have studied the thesis entitled (**Intelligent air quality classification and monitoring system using machine learning techniques)** was prepared under our supervision at the Department of Information) presented by the student (**Huda Kadem Alwan**) and examined her in its content and what is related to it, and that, in our opinion, it is adequate with standing as a thesis for the degree of Master in Information Technology-Information Networks.

Signature: Name: **Dr. Mahdi Nsaif Jasim** Title: **Professor** Date: / / 2023 **(Chairman)**	Signature: Name: **Dr. Alharith A. Abdullah** Title: Assist. Professor Date: / / 2023 **(Member)**
Signature: Name: **Dr. Suad A. Alasadi** Title: Assist. Professor Date: / / 2023 **(Member)**	Signature: Name: **Dr.Ghidaa A. Al-Sultany** Title: Professor Date: / / 2023 **(Member and Supervisor)**
Signature: Name: **Dr. Saba Mohammed Hussein** Title: Lecturer Date: / / 2023 **(Member and Co-Supervisor)**	

Approved by the Dean of the College of Information Technology, University of Babylon.

Signature: **Dr. Wesam S. Bhaya**
Name: Title: Professor
Date: / / 2023
(**Dean of Collage of Information Technology**)

Dedication

I dedicate this thesis to my dear father, who provides me with unlimited supporter until my research was fully finished, my beloved mother, my husband who has been a constant source of support and encouragement, my dear son Ali who was my strength, and my lovely sisters and brothers.

Acknowledgement

All praise is to ALLAH Almighty, who successfully enabled me to complete this task.

My deepest gratitude is to **my advisors** Prof. Dr. Ghidaa A. Al-Sultany and Dr. Saba Mohammed Hussein, for invaluable guidance, supervision, and untiring efforts during this work.

I would like to thank all **my dear College of Information Technology teachers** for their remarkable education and guidance during the study period.

I would like to thank all my **dear colleagues** in the MSc. course and the College of Information Technology staff.

Finally, I would like to thank all the helpful and lovely people who helped me directly or indirectly to complete this work.

Abstract

Air pollution is a mounting global concern due to its adverse effects on human well-being and the environment. Despite the substantial attention given to air pollution as a significant environmental issue, there is a research gap in terms of comprehensive measurements of pollutant concentrations, including CO, NO2, SO2, O3, PM2.5 and PM10. This gap limits the ability to accurately predict air pollution in advance, making it challenging for individuals to manage their health and for governments to make informed policies.

This thesis introduces intelligent air quality monitoring system that synergizes Internet of Things (IoT) technology with Multilayer Perceptron (MLP) deep learning models to forecast air pollution levels. This system built by using min-max normalized and rectifies missing values in preprocessing, employing the Pearson correlation coefficient, a judicious selection of the most influential features, including pm2.5, Pm10, NOx, NO2, NO, O3, and CO2, is undertaken. In the classification stage, this model prefers to choose adam optimizer that considers more suitable in deep learning. ReLU and softmax as activation functions. cross-entropy as loss function, two datasets have been implemented to evaluate the suggested model, the standard Central Pollution Control Board (CPCB) dataset and India air quality station dataset. Integral to the system's operation are IoT components such as sensors and Arduino devices, instrumental in real-time collection of air quality data. This data undergoes processing before being fed into the MLP model, which has been meticulously trained on the CPCB dataset, thereby engendering precise air pollution predictions.

The consequences of the proposed work have shown encouraging results in terms of high accuracy. The accuracy of CPCB dataset reaches 99.115%. Moreover, upon validation with a distinct dataset from India's Air Quality Stations, the model maintains a commendable performance rate of 97.47%. This cross-validation underscores the model's robustness and its potential applicability across varied geographical contexts.

Declaration Associated with this Thesis

Some of the works presented in this thesis that has been published or accepted are listed below.

1. **Deep learning-based air quality prediction in the Internet of Things environment**

   TELKOMNIKA, 2023

2. **Air Pollution Prediction Using Machine Learning and Neural Network: A Survey**

   5th International Conference on Information Technology, Applied Mathematics and Statistics. 2023

Table of Contents

List of Tables

List of Figures

List of Abbreviations

Abbreviation	**Description**
AI	Artificial Intelligence
ANN	Artificial Neural Network
AQI	Air Quality Index
CNN	Convolutional Neural Network
DNN	Deep Neural Network
EEA	European Environment Agency
FN	False Negative
FP	False Positive
IoT	Internet of Thing
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MLP	Multilayer Perceptron
MSE	Mean Square Error
PPM	Parts Per Million
ReLU	Rectified Linear Unit
TP	True Positive

Chapter One

General Introduction

Chapter One

General Introduction

1.1 Overview

In recent times, air pollution has emerged as a significant source of concern due to its detrimental effects on human health and the environment. The rise in air pollution is primarily attributed to increased industrialization, urbanization, and vehicle emissions, posing significant threats to public well-being and overall quality of life. Consequently, the development of dependable and efficient air pollution forecasting systems is crucial to mitigate these adverse effects. Early and precise forecasting of pollution levels plays a pivotal role in minimizing the negative impacts of air pollution [1]. The availability of such forecast systems enables policymakers, urban planners, and citizens to make informed decisions and take proactive measures to minimize their exposure to polluted air. The key air pollutants include Pb, CO, NO2, O3, SO2, particulate matter with a diameter of 10 micrometers or less (PM10), and particulate matter with a diameter of 2.5 micrometers or less (PM2.5). However, according to the European Environment Agency (EEA) report, most air pollution-related deaths are primarily attributed to contaminants such as PM, NO2, and ozone [2].

IoT has become a significant technology that has the potential to revolutionize various industries, including environmental management and monitoring. By connecting physical devices, sensors, and data networks, IoT enables the collection, analysis, and real-time transmsission of data. This presents an opportunity for more proactive decision-making based on data-driven insights [3]. IoT offers a unique opportunity to enhance monitoring capabilities and prediction models, ultimately leading to a reduction in the adverse impacts of air pollution on public health [4].

Deep learning, a field within artificial intelligence, offers powerful tools for analyzing large datasets, identifying patterns, and making accurate predictions. Environmental sciences have recently make benefits from the application of machine learning techniques. By leveraging historical data on air pollution, meteorological data, and other relevant factors, machine learning algorithms can uncover complex relationships and provide valuable insights into patterns and trends in air pollution. This forecasting capability empowers authorities, decision-makers, and individuals to take measures in mitigating air pollution and protecting public health [5].

1.2 The Related Works

In recent years, numerous researchers have focused on using deep learning techniques to predict air pollution. This section of the thesis provides a summary of some relevant studies conducted in this domain.

In (Zhang et al. 2018) The Wavelet neural network was introduced as a dependable method for predicting the levels of air pollutants. Two groups of climatic factors, namely temperature, and humidity, along with significant air pollutants such as O3, CO, NO2, SO2, PM10, and PM2.5, were considered influential variables. The findings from the experiment indicate that the detection outcomes based on the Wavelet neural network are more accurate and precise, and the network possesses a strong capability for self-learning [6].

In (X. Zhao et al. 2018) A highly accurate prediction model is developed using Recurrent Neural Network (RNN), a deep learning technique. The model is trained on a dataset comprising six features, namely Pm2.5, pm10, CO, NO2, SO2, and O3. When compared to alternative machine learning approaches such as support vector machines and random forests, the RNN model outperforms them, yielding the most favorable results [7].

(Srivastava et al. 2018), In their study, the authors made predictions on the Air Quality Index (AQI) for important pollutants such as PM2.5, PM10, CO, NO2, SO2, and O3. They employed several techniques including Gradient Boosting, Linear Regression, SDG Regression, Random Forest Regression, Decision Tree Regression, Support Vector Regression, Artificial Neural Networks, and Adaptive Boosting Regression. The performance of these methods was evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE). The findings indicate that Support Vector Regression and Artificial Neural Networks outperformed the other models in forecasting air quality in New Delhi. These models achieved higher accuracy and lower error rates compared to the rest[8].

In (Tsai et al. 2018), The authors introduced a technique for predicting PM2.5 levels by utilizing a combination of Recurrent Neural Networks (RNN) and Long Short-Term Memory (LSTM). They obtained the training data for the network from the Taiwan Environmental Protection Agency (EPA) spanning the years 2012 to 2016. The findings demonstrated that the proposed method effectively predicts PM2.5 concentrations with high accuracy[9].

In (Akhtar et al. 2018) Air quality is predicted by a multilayer perceptron algorithm, which is a multilayer perceptron with two hidden layers and a sigmoid activation function, Naïve Bayes algorithm, and Support Vector Machine algorithm. Results are then compared for different algorithms, which show MLP as the best with an overall accuracy of 98%[10].

In (Gnana Soundari et al. 2019) the authors developed a model to predict the air quality index based on historical data by utilizing a Gradient decent boosted multivariable regression using India air quality station dataset. The model has 96% accuracy on predicting the current available dataset on predicting the air quality index of whole India [11].

(Maleki et al. 2019) comparing different techniques for predicting PM2.5 and PM10 levels. Based on their evaluation, the autoregressive nonlinear neural network method was determined to be the most precise approach among the tested techniques, outperforming RSVM, GWR, and ANN [12].

(Wang et al. 2019), the authors of the study developed an optimized neural network by utilizing backpropagation and genetic algorithms to enhance its performance. They proposed an improved neural network structure and assessed its effectiveness in predicting PM2.5 mass concentration. The findings of the study demonstrate that the genetic algorithm-optimized neural network achieves superior prediction accuracy and reduces error rates in comparison to previous models [13].

(Mahalingam et al. 2019) In this research work, a Machine Learning model for Air Quality Index prediction for smart cities is proposed. The model is tested with the Delhi Air Quality data By using the Neural Networks (with one hidden layer and sigmoid function) and SVM, AQI is predicted successfully with 91.62% of accuracy for Neural Networks and 97.3% for Support Vector Machines[14].

(Ivanova and Elenkov 2019), the authors of the study developed an optimized neural network by utilizing backpropagation and genetic algorithms to enhance its performance. They proposed an improved neural network structure and assessed its effectiveness in predicting PM2.5 mass concentration. The findings of the study demonstrate that the genetic algorithm-optimized neural network achieves superior prediction accuracy and reduces error rates in comparison to previous models [4].

In (Aljanabi et al. 2020), A predictive model was developed to forecast ozone concentration by incorporating seasonal and meteorological factors such as ozone levels, humidity, and temperature. The researchers utilized the forward feature wrapper selection method to select the most relevant features, which helped reduce the prediction time. To evaluate the performance of different algorithms, namely multi-layer perceptron (MLP), support vector regression (SVR), decision tree, and extreme gradient boosting (XGBoost), the researchers compared their results. The MLP algorithm demonstrated superior predictive performance with a mean absolute error (MAE) value of 4.717 compared to the other algorithms [15].

(Hamami et al. 2020) performed a classification system for air pollution levels using IoT data. The data included measurements of various pollutants such as PM2.5, PM10, CO, SO2, and O3. The research employed neural network methods to classify the data into three air pollution levels. The neural network architecture consisted of multiple hidden layers and neurons. By training the neural network model, the researchers achieved a high accuracy rate of up to 96.61%. This indicates that the proposed neural network approach was successful in accurately classifying air pollution levels based on the provided IoT data [16].

In (Shwet Ketu et al. 2021) the authors introduced a technique for predicting Air quality performed on sensor-based Indian Central Pollution Control Board (CPCB) dataset. The proposed algorithm with accuracy of Multi-Layer Perceptron (95.71%), NB (80.87%), and SVM (96.92). [17]

(Kumar et al. 2022) conducted a study on AQI prediction for India, employing SVM and XGBoost models. They utilized the CPCB dataset and it was properly preprocessed, and important features were identified through correlation analysis. The evaluation of the models revealed that the XGBoost model achieved the highest accuracy among the two models with an accuracy of 96%, indicating its effectiveness in predicting air quality in this context [18].

Table 1-1 : A compilation of recent related works

No.	Authors	Applied Algorithm	Features	Dataset	Results
1	(Zhang *et al.*, 2018) [6]	Wavelet neural network	O3, CO, NO2, PM10 and PM2.5	data is collected from the air quality sensor network monitor (XHAQSN-808) located on the campus of the Beijing	Accuracy: 98.92%
2	(Zhao *et al.*, 2018) [7].	Recurrent Neural Network (RNN)	Pm2.5, pm10, CO, NO2, SO2, O3	the air quality data is collected from U.S. EPA database	Accuracy: 80.27
3	(Srivastava, Singh and Singh, 2018) [8]	Gboosting, DT,SVR,MLP	NEW Delhi AQ dataset	NEW Delhi AQ dataset	SVR MAE: 0.3192 MLP MAE: 0.3976
4	(Tsai, Zeng and Chang, 2018) [9]	combined (RNN) and (LSTM).	PM2.5	retrieved from The Taiwan EPA from 2012 to 2016	MAE: 1.644
5	(Akhtar *et al.*, 2018) [10]	MLP', NB	SO2, NO2, O3, CO, pm2.5, pm10	CPCB dataset	MLP accuracy:98%
6	(Ivanova and Elenkov, 2019) [11]	Multilayer Perceptron	NO, NO2, O3 and PM10	Data acquired from Executive Environment Agency	R^2: 0.65

No.	Authors	Applied Algorithm	Features	Dataset	Results
7	(Wang and Wang, 2019) [13]	1- Back propagation neural network	CO, NO2, SO2, O3, pm10	real-time air quality data released by the Ministry of Environmental Protection of China	Accuracy: 98.32%
8	(Mahalingam *et al.*, 2019) [14]	SVM ANN	PM2.5, PM10, CO, NO2, SO2, and O3	Delhi AQ dataset	SVM accuracy: 97.3 ANN accuracy :91.62
9	(Hamami and Fithriyah, 2020) [15]	neural network	O3, CO, NO2, SO2, PM10 and PM2.5	data is collected from the air quality sensor network monitor (XHAQSN-808)	Accuracy: 96.61%.
10	(Aljanabi, Shkoukani and Hijjawi, 2020) [16]	1-MLp 2-support vector regression 3-DT 4-XGB	Temperature Wind speed Wind direction humidity O3	The dataset was obtained from King Hussein Public Park station	MAE: 4.717.
11	(Shwet Ketu et al. 2021) [17]	MLP	CO, NO2, SO2,pm10 and PM2.5	CPCB dataset	Accuracy ; 95 %
12	(Kumar and Pande, 2022), [18]	X-GBoost	NOx, NO2, CO, SO2, O3, pm10, PM2.5	CPCB dataset	Accuracy: 96%

1.3 The Problem Statement

Despite the substantial attention given to air pollution as a significant environmental issue, there is a research gap in terms of comprehensive measurements of pollutant concentrations, including CO, NO2, SO2, O3, PM2.5 and PM10. This gap limits the ability to accurately predict air pollution in advance, making it challenging for individuals to manage their health and for governments to make informed policies. Additionally, there is a need to develop a more precise monitoring system that uses IoT technology to improve air quality predictions in specific regions of Iraq.

1.4 Aim of the thesis

The aim of this thesis is to develop an advanced air pollution prediction system that harnesses the power of deep learning techniques within IoT environment. The system will be designed to enhance air quality monitoring and prediction, with the primary objective of improving air quality and accurately forecasting pollution levels. By leveraging comprehensive measurements of pollutant concentrations, including CO, NO2, SO2, O3, PM2.5 and PM10, the developed system will provide robust and precise predictions. Through this research, the thesis aims to make significant contributions to the field of air quality management by utilizing deep learning algorithms in an IoT framework, ultimately contributing to the mitigation of adverse effects caused by air pollution.

1.5 The Thesis Objectives

The Objectives of this thesis can be divided into the following:

a. Develop an integrated system that combines deep learning using Multilayer Perceptron (MLP) and Internet of Things (IoT) technologies to create an advanced air quality monitoring system.

b. Utilize the Central Pollution Control Board (CPCB) dataset as a valuable source of historical air quality data to enhance the accuracy and reliability of the proposed system, ensuring comprehensive coverage of pollutant concentrations.

c. Enhance air quality modeling and prediction by leveraging the power of MLP in identifying complex relationships and patterns between pollutant concentrations, enabling the system to generate reliable decisions and predictions.

d. Implement real-time data collection capabilities from diverse IoT devices, including air quality sensors.

e. Evaluate and validate the performance of the developed system by comparing the predicted air pollution levels with actual measurements, assessing its effectiveness in accurately forecasting pollution levels and supporting air quality management decisions.

1.6 The Outlines

This thesis contains five chapters organized as follows:

Chapter Two: The Theoretical Background

The chapter provides a thorough explanation of the concept of the system and its various types. It then delves into the techniques of machine learning and deep learning that are utilized to construct the model. The chapter concludes by referring to the performance metrics and drawing conclusions based on them.

Chapter Three: The Proposed System Design

In this chapter, the proposed system details have been mentioned, for include, the proposed system explains the main steps of designing a modern prediction model for air pollution using a deep neural.

Chapter Four: The Results Discussion and Analysis

This section presents the discoveries and offers an assessment of the methodology employed in the system.

Chapter Five: The Conclusions and Suggested Future Works.

The conclusion of the findings was reported in this chapter. It also included a description of future works.

Chapter Two
The Theoretical Background

Chapter Two
The Theoretical Background

2.1 Introduction

This chapter reviews the preliminary concepts of air pollution and the main pollutants that affects the air. The role of machine learning and neural network techniques in building the system will be surveyed by describing each technique's benefits and determining the best to use.

2.2 The Air pollution

Air pollution is a global environmental issue that has significant impacts on human health, ecosystems, and the overall quality of life. It refers to the contamination of the Earth's atmosphere by harmful substances, both natural and human-made, that can cause adverse effects when present in excessive amounts. Air pollution is a complex problem that arises from various sources and can take different forms, including gases, particulate matter, and volatile compounds[19].

Breathing in polluted air can lead to a range of health problems, from respiratory issues like asthma and bronchitis to more severe conditions like heart disease and even lung cancer. It's particularly harmful to vulnerable groups like children and the elderly. Beyond its impact on human health, air pollution also harms ecosystems and wildlife. It can damage crops, forests, and bodies of water, disrupting delicate ecosystems and affecting biodiversity [20].

Given the increasing global death toll caused by air pollution, it is imperative that governments, universities, individuals, and anyone capable of contributing take collective responsibility to combat this issue. Together, they must actively work towards reducing pollution levels and preventing the rapid proliferation of harmful substances. The idea of globalization has introduced fresh perspectives to the world. Today, the world is like a shared dwelling for humanity, where both positive and negative actions impact everyone. As a result, we must come together and develop collective strategies to address these global challenges. Unity and cooperation are necessary to ensure mutual benefits and minimize losses in the face of these universal issues [21].

Air pollution is caused by various chemical elements, and the specific sources of pollution can differ between regions and countries. To effectively combat air pollution, it is important to address these elements individually, taking into account their specific properties and characteristics. The following provides a brief overview of these elements [22].

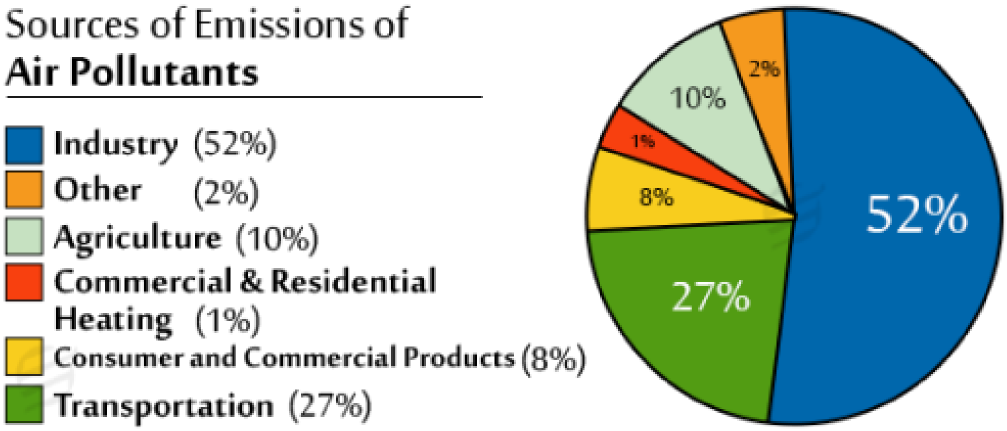


Figure 2.1: Air pollution contribution percentages [23]

2.2.1 The Air pollutants

An air pollutant refers to a substance present in the air or atmosphere that can cause detrimental effects on human health and disrupt the ecosystem. They exist in different forms, including gases, particles, and aerosols. Some common air pollutants include particulate matter (tiny particles), ground-level ozone (a component of smog), nitrogen dioxide (a reddish-brown gas), sulfur dioxide (from burning fossil fuels), carbon monoxide (produced by incomplete combustion), volatile organic compounds (emitted from various sources), heavy metals (toxic elements like lead and mercury), and ammonia (released from agricultural and industrial activities). [24]. These pollutants can have negative effects on human health, causing respiratory problems, cardiovascular issues, and other health risks. They also harm the environment, contributing to acid rain, damaging ecosystems, and affecting wildlife. Addressing air pollution is important for maintaining clean air and a healthy environment [25]. The pollutant can be categorized as primary and secondary pollutants.

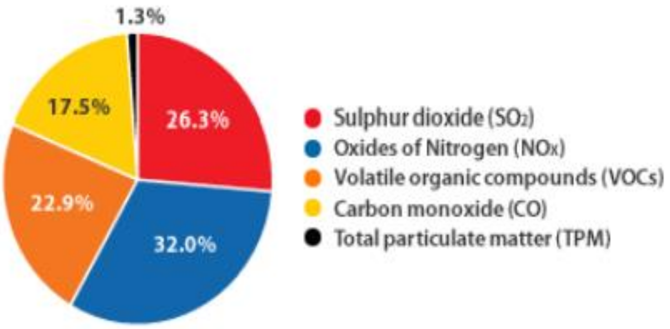


Figure 2.2: Most common pollutants [5]

A. Primary pollutants

Primary pollutants are substances released directly into the atmosphere as a result of human activities or natural processes. These pollutants are typically emitted in their original form and can have harmful effects on the environment and human health. They play a significant role in air pollution and contribute to the formation of secondary pollutants [26].

1. **Carbon monoxide (CO2):** This particular pollutant is one of the most notable pollutants compared to others that are being rapidly generated due to various factors, and it poses a greater level of harm. Carbon dioxide, which is naturally present in excessive amounts in the atmosphere and necessary for plant life, is now at approximately 410 parts per million (ppm) in the Earth's atmosphere, whereas it was at 280 ppm during pre-industrial times. It is estimated that billions of metric tons of CO2 are emitted each year as a result of burning fossil fuels [27].
2. **Sulfur dioxide (SO2**): This pollutant and the most popular form of it which is SO2 is likely produced in many industrial processes by volcanoes. It is found in coal and petroleum. Therefore, using these fuels, especially for the power system is much of environmental concern [28].

3. **Carbon monoxide (CO**): This pollutant is a colorless, odorless toxic gas. The production and spreading of carbon monoxide are from the combustion of natural gas, coal, or wood. Although, the exhaust of vehicles produces a lot of carbon monoxide into the atmosphere [28].
4. **Particulates Matter (PM):** PM is a significant air pollutant and can originate from both natural and anthropogenic sources. Natural sources include dust, pollen, and volcanic eruptions, while anthropogenic sources include industrial emissions, vehicle exhaust, and burning of fossil fuels. The size of PM is often categorized into different fractions. The most commonly measured fractions are PM10, PM2.5, and PM1. These refer to particles with diameters of 10 micrometers or less, 2.5 micrometers or less, and 1 micrometer or less, respectively. The smaller the particle size, the longer it can remain suspended in the air and potentially cause adverse health effects when inhaled [29]. PM pollution has been linked to various health problems, including respiratory, cardiovascular diseases, and premature death. Monitoring and controlling PM levels in the air is crucial for protecting human health and maintaining air quality standards [30].
5. **Nitrogen dioxide (NO2):** is a reddish-brown gas with a pungent odor. It is formed when nitrogen oxide (NO) reacts with oxygen in the atmosphere. NO2 is a significant air pollutant and a key component of smog. It contributes to the formation of ground-level ozone and can have harmful effects on human health and the environment [31].
6. **Ammonia (NH3):** Released from agricultural activities and industrial processes [31].

B. Secondary pollutants

These pollutants are formed in the air as the consequence of primary pollutants reacting or we can say interacting. These pollutants are not released directly. For saying, ground-level ozone is one of these pollutants. Not to be forgotten some of the pollutants can fit into both primary and secondary categories [32]. Secondary pollutants are divided into three categories.

Ground-level ozone (O3) shaped from NOx and VOCs. Ozone (O3) is a key constituent of the troposphere. It is additionally an essential constituent of specific areas of the stratosphere generally known as the Ozone layer [33].

Nitrogen oxide or NOx is a family of poisonous, highly reactive gases that form when fuel is burned at high temperatures. It is brown in color and emits from vehicles as well as industrial sources such as power plants, industrial boilers, cement kilns, and turbines. Nitrogen oxides have problematic chemical reactions in the atmosphere with volatile organic compounds. These reactions produce smog on hot summer days [34].

Acid Deposition It is a type of precipitation – rain, snow, sleet, hail, or fog – that has a lower pH (and is, therefore, more acidic) than normal. This higher acidity causes problems in ecosystems and the environment and remains one of the major environmental concerns. Acid rain forms when water in the air combines with nitrogen oxides and sulfur dioxide (two types of pollutants) and then falls down the surface of the Earth. It has many damaging effects on vegetation, lakes, fish, buildings, and other structures. It also causes respiratory diseases in humans, especially those that have bad health [35].

2.2.2 Air Pollution Sources

Air pollution arises from various sources, both natural and human-made, that release pollutants into the atmosphere. These pollutants can have adverse effects on air quality, human health, and the environment. Understanding the sources of air pollution is crucial for developing effective strategies to mitigate its impact [36].

A. Human-made sources:

- Stationary sources such as fossil fuel power, factories, waste incinerators, wood, crop waste, and dung.
- Mobile sources like vehicles, marines, and aircraft.

- Controlled Burn in agriculture
- Fumes taken from varnish, aerosol sprays, paint, hairspray, etc.
- Waste deposition in landfills, consequences to methane.
- germ warfare, rocketry, nuclear weapons, and toxic gases which are used in the Military.
- Fertilized farmland which produces NOx

B. Natural Sources:

- Dust from natural earth sources
- Methane, which is from animals’ food and sewage processing plants
- Radon gas, coming from radioactive decay
- CO and Smoke from wildfires
- Vegetation

2.3 Data Preprocessing

Data preprocessing considers the general first step in data mining processes handling missing data, normalizing data to decrease the complexity of the process, discovering the smoothen relationships between attributes, reducing outliers' values, and selecting the most important or extracting features [37].

The section 2.3.1 will explain preprocessing techniques for data mining. These techniques will be explained in sequence. The first techniques is data cleaning to handling the missing value. The next is handling big values to small values by using normalized techniques, determine the benefit of those transformations, and identify the used methods in this thesis.

2.3.1 Data cleaning

Data in the real world is incomplete (contains missing values), noisy (has an error or outrange values) and inconsistency (has asymmetric data for the same attributes), etc. described as follows:

1- **Incomplete**: These are the attributes that contain null values or missing values. For example, data of service’ types were null. In this thesis, the used dataset contains missing data. Deal with it by filling it in estimated data.

2- **Noisy**: when the dataset contains invalid values or outliers, data will be considered a noisy dataset. For example, age =256.

3- **Inconsistent**: the inconsistency of data correctly in some features of the dataset that may contain contradictory data, whether names, numbers, or otherwise. For example, Birthday="16/10/1994" in the 2020 year has Age= "41" [37].

2.3.2 Normalization process

Usually, a dataset's original attributes are Not very appropriate to get accurate predictive models of deep learning. And the big numerical values have negatively affected the time of mathematical operations and complicated the implementation process. Therefore, we implement a Normalization method on the original dataset to generate a new set of values with desired properties to help the model's predictive power [37]. The common normalized methods will explain in the following:

1- **Z-Score normalization**: (also known as a standard score) provides an estimation of how far a data point is from the mean. Although more practically, it's a calculation of how many standard deviations indicate a raw score is below or above the population [38].

2- **Min–Max Normalization:** (range transformation) is one of the most common ways to normalize data. It is a simple technique to fit the data in a pre-defined boundary with a pre-defined boundary. To use a Min-max normalization, it must first detect the new min value and the new max value and define the old min and max values [38]. The formula for Min-max normalization is below:

$$x\ scaled = \frac{x - x\ min}{x\ \max - x\ min} \ldots. (2.1)$$

Where, x scaled= the result of min-max normalization, x min = old minimum value, max_x = old maximum value.

2.3.3 Feature Selection-based Parson Correlation Coefficient

The Pearson Correlation Coefficient, often denoted as "r," is a statistical measure that quantifies the strength and direction of the linear relationship between two continuous variables. It assesses how well the relationship between these variables can be described by a straight line. The value of the Pearson correlation coefficient ranges from -1 to 1:

1. If "r" is close to 1, it indicates a strong positive linear correlation, meaning that as one variable increases, the other tends to increase as well.
2. If "r" is close to -1, it indicates a strong negative linear correlation, meaning that as one variable increases, the other tends to decrease.
3. If "r" is close to 0, it suggests a weak or no linear correlation, meaning that there ~~is~~ little to no relationship between the variables.
4. The formula for calculating the Pearson correlation coefficient between two variables X and Y with n data points is:

$$R = \frac{\sum(x_i - \grave{x}) - (y - \grave{y})}{\sqrt{\sum(x_i - \grave{x})^2}\ \sum(y_i - \grave{y})^2} \quad \dots . (2.2)$$

Where:

r is the Pearson correlation coefficient.

x_i and y_i are the data points for variables X and Y, respectively.

x_i ¯ and x_i¯ are the means (averages) of the X and Y data, respectively.

2.4 Air Pollutions Prediction Techniques

Air pollution prediction techniques involve the use of various models and algorithms to estimate the levels of pollutants in the air. These techniques rely on data from monitoring stations, satellite imagery, and other sources to make predictions about future air pollution levels [39]. Here are some commonly used air pollution prediction techniques:

2.4.1 Machine Learning

Machine learning is a first-class ticket to the most thrilling career in data prediction and analysis. It is an idea to learn from examples and specifications or experience data without being frankly programmed, without writing any code. We build a logic depending on the data given and we feed it into that genic algorithm[40]. Machine learning also can be referred to as the alteration in network systems that implement tasks related to and linked with artificial intelligence systems. Some of are recognition, diagnosis, prediction planning, and robot control system. We can say that machine learning is training the computer for sure with different algorithms to test the machine in automatic intelligent data processing [41].

For example, in one kind of classification algorithm, can put the data in different groups. it can detect the handwriting of the alphabet, or identify faces in the image for example. machine learning is a field raised from artificial intelligence (AI). It seeks to build better smart and intelligent machines. And the only way to achieve this task is to let the machine learns from its environ and this sound similar to a little child learning from himself in human childhood. Machine learning developed as a new ability. Now, machine learning exists in many divisions of technology that we didn’t even realize while we are using it. Machine learning is correlating with the study of the algorithms to increase the effectiveness of the machine spontaneously through testing and training of that machine using the algorithm with different data. ML improves and evolves rules that help machine learning to process similar conditions ever relation efficaciously in the hybrid model. Understanding input variables, and how it's moving into vectors is such an important thing [42].

It is necessary to predict the air quality accurately. Various traditional methods are there to measure it but results are not accurate and it involves a lot of mathematical calculations. Machine Learning a subset of Artificial Intelligence has an important role in predicting air quality. Various researches are being done on measuring Air quality Index by using Machine Learning algorithms such as decision tree and random forest. So, to control Air Pollution first necessary step is to measure accurately the Air Index Quality. Machine Learning algorithms plays an important role in measuring air quality index accurately [28].

2.4.2 Neural Network and Deep Learning Techniques

Neural networks are a form of computational network (a collection of nodes and their connections) inspired by biological neural networks, which are the dynamic networks of neurons found in human brain. Artificial neurons are nodes formed in neural networks that are supposedly designed to function like real neurons. The artificial neural network is the simplest and most popular kind of neural network. The network of nodes (artificial neurons) that make up the neural network (or artificial neural network) is shown in Figure 2.3 with the biological neuron [43].

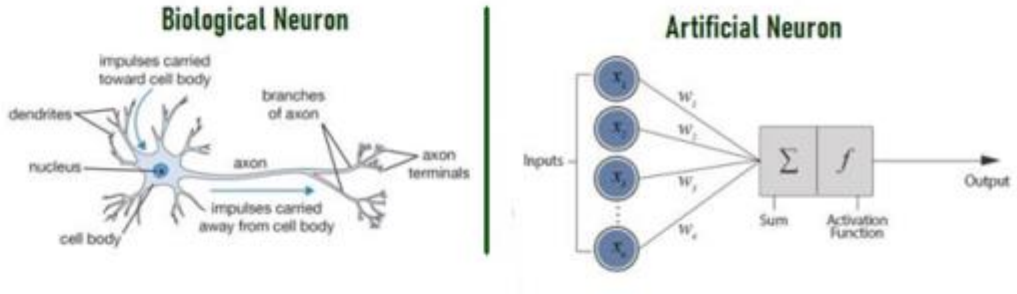


Figure 2.3:Biological Neuron and Artificial Neuron [43]

The artificial neural network structure consisted of several layers, and each layer consisted of several nodes (artificial neurons). The initial weights (which reflect the interconnection between these nodes) and bias are not strong enough for decision-making (classification, etc.). It's similar to the brain of a newborn. To be a successful decision-maker, a kid learns about his previous experience. The experiences (labeled data) assist the brain's neural network in tuning the (neural) weights and biases. The artificial neural network follows a similar route. To build a strong classifier, the weights are fine-tuned iteratively. Since manually tuning and obtaining the right weights for thousands of neurons is time-consuming, we use algorithms to accomplish these tasks[43].

A Neural Network is constructed from three layers: An input layer, hidden layers, and an output layer. The input layer represents initializing input data. In the model training phase, the input data is represented by the dataset of the previous experience. In the evaluation phase, the input data is represented by the testing data or new unlabeled data. The Hidden layers represent the intermediate layer between the input and output layer and the place where all the computation (i.e., feature extraction and classification) is executed. At last, the Output layer produces the results for given data inputs [43]. Figure 2.4 shows the layers of the Neural Network.

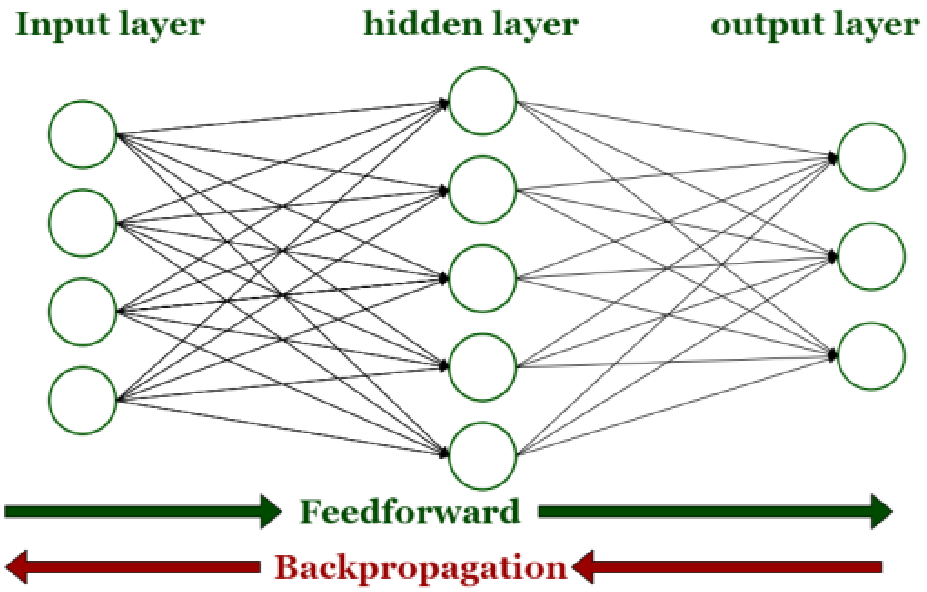


Figure 2.4: neural network [43]

A neural network has two propagations: Feed Forward Propagation and Back Propagation. Feed Forward Propagation represents the Forwarding from the input layer to the hidden layers and then to the output layers. This Forwarding aims to calculate output results from input data (predict results) by using a perceptron classifier, as shown in Equation (2.2):

$$y = \sum_{i=1}^{n} x_i \, w_i + b \qquad \text{…... (2.3)}$$

Where n the number of neurons in the current layer, x represents the data of neurons, w represents the weights, and b is the bias. The results will then be inserted into Activation Functions (more details will explain in the following subsection). Back Propagation reverse Feed Forward Propagation, its propagation from the output layer to the input layer. The main aim of Back Propagation is learning the model by modification weights and bias. This modification is done by using optimizer and loss function methods (more details will explain in the following subsection) [44][45].

A. Activation Functions

An activation function is developed from a study of the biological neuron of the human brain. The neuron becomes active at a certain level, identified as the activation potential. In most cases, it tries to keep the output in a limited range. Activation functions are beneficial since they insert nonlinear into neural networks that enable the neural network models to learn complex operations. The most popular activation functions are Rectified Linear Unit (ReLU) Function, Hyperbolic Tangent Function (Tanh), Sigmoid Function, and Softmax Function [46] As shown in Figure(2.5).

1. **Sigmoid function**: is a nonlinear activation function that is used in neural network feedforward. It is a differentiable real function with an output range of 0 to 1,[47], and its Equation is:

$$f(x) = \frac{1}{1 + exp^{-x}} \qquad \text{…… (2.4)}$$

x represents the input data of the active function.

2. **Tanh Function**: This is an activation function commonly used with recurrent neural networks for speech recognition and natural language processing tasks. The range of its output is -1 to 1[47]. its equation is:

$$(x_i) = \frac{e(x_i) - e(-x_i)}{e(-x_i) + e(x_i)} \quad \text{..... (2.5)}$$

3. **ReLU Function**: this is a linear activation function. Consider the most common activation function used with deep learning. It is fast learning and easy to optimize[47]. it is represented as

$$f(x) = \max(0, x) \begin{cases} x_{i,} & x_i \geq 0 \\ 0, & x < 0 \end{cases} \quad \text{....... (2.6)}$$

4. **SoftMax Function**: is an activation function usually used with an output layer of neural networks. Its output Represent as probability distribution values from a vector of real numbers. The output range of this activation function is between 0 and 1, and the sum of its output probabilities values equal 1.[48] The equation:

$$f(x) = \frac{e(x_i)}{\sum_j e(x_i\,)} \quad \text{...... (2.7)}$$

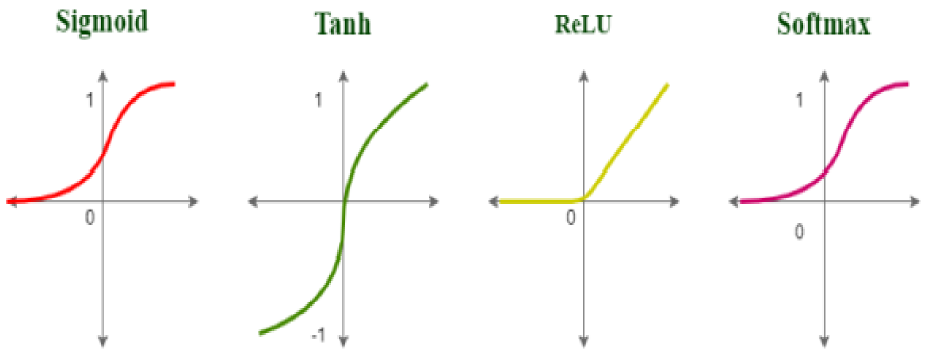


Figure 2.5: Activation Functions [48]

B. The Loss Functions

The loss function (also known as a cost function) is the method that calculates the difference between the prediction outputs of a neural network to the actual values (training label data). There are several methods used for the Loss function. The most common ones are:

1. **Mean Squared Error Loss (MSE):** The sum of the squared differences between the predicted and actual values, divided by the count of the actual values, is the regression loss function[49]. It is mathematically expressed as:

$$E_{total=}\frac{\sum_{i=1}^{n}(yi-y')^2}{n} \qquad (2.8)$$

Yi is the actual values, $yi\ '$ is the predicted values, and n represents those values.

2. **Mean Absolute Error Loss (MAE):** is another method to calculate loss functions for regression algorithms like MSE that does not square the values of the differences; instead, it uses the absolute sum of the differences between the predicted and actual values, divided by the number of the actual values[50]. It is mathematically expressed as:

$$E_{total=}\frac{\sum_{i=1}^{n}|yi-yi'|}{n} \qquad (2.9)$$

3. **Binary Cross-Entropy:** The cross-entropy loss function, also known as log entropy, is utilized to evaluate the effectiveness of binary classification models, where the output is a probability value ranging from 0 to 1 [51]. It can be mathematically defined as follows:

$$E_{total} = -(y \log (y\ ') + (1-y) \log (1-y\ ')) \quad \ldots\ldots \ (2.10)$$

Where, y is actual value (label) for that class, y' corresponds to the predicted value (probability) for that class.

4. **Multi-class Cross-Entropy Loss:** The cross-entropy loss function, which is also known as log entropy loss, is used to evaluate the performance of multi-class classification models. It quantifies how much the predicted probabilities deviate from the actual labels. The equation for cross-entropy loss is as follows:

$$E_{total} = -\sum_{C=1}^{M} y_c \log\left(y'_c\right) + \left(1 - y_c\right) \log\left(1 - y'_c\right) \quad \dots\dots (2.11)$$

Where, M represents the number of classes in the classification problem.

The c denotes a specific class.

yc represents the actual value (label) for that class.

yc' corresponds to the predicted value (probability) for that class.

The summation (Σ) calculates the sum of the losses for all classes [52].

C. Optimizers

As explained earlier, the loss function calculates the difference between the predictor results and the actual values. To obtain an ideal model (high accuracy), this rate of variation must be minimal. The optimizers' main purpose is to minimize the loss function values by modifying the deep learning algorithm parameters (weights and biases) by many iterations during Backpropagation to reach to best values. The initial values of weight and bias suggest zero (or one or any number). After that, the optimizer will assume whether every parameter will increase or decrease in the next iteration[53] There are several optimizers in deep learning. The following subsections will mention some common optimizers:

1- **Gradient Descent (GD):** is the most common optimizer used with neural networks. As it is considered the basis of most of the neural network's optimizers, considering gradient descent as one of the types of optimization methods, it aims the same as that of the optimizers, which they are minimize the cost function values, Basic math equation for gradient descent is:

$$\theta' = \theta - \eta \frac{\partial L}{\partial b} \qquad (2.12)$$

Where θ' is the new parameter that is optimized, θ is the old parameter, η is the learning rate, and $\frac{\partial L}{\partial b}$ is the gradient descent[54], in the neural network, the gradient descent equations are:

$$w_t{}' = w_t - \eta(\frac{\partial E_{total}}{\partial w_t}) \qquad .....(2.13)$$

$$b_t{}' = b_t - \eta\,(\frac{\partial E_{total}}{\partial b_t}) \qquad ......(2.14)$$

Where w is weight, b is bias, $\partial Etotal\ \partial w$ is the gradient of error rate in the loss function. w′ represent the new wight, b′ represent the new bais.

According to the amount of data, gradient descent has three types:

1- Batch gradient descent: uses the whole training data to update weight and bias for each iteration.

2- Stochastic gradient descent SGD: uses only single records to update parameters for each iteration.

3- - Mini-batch gradient descent: use a batch of records[55] .

The Figure (2.6) represent the loss function with gradient descent.

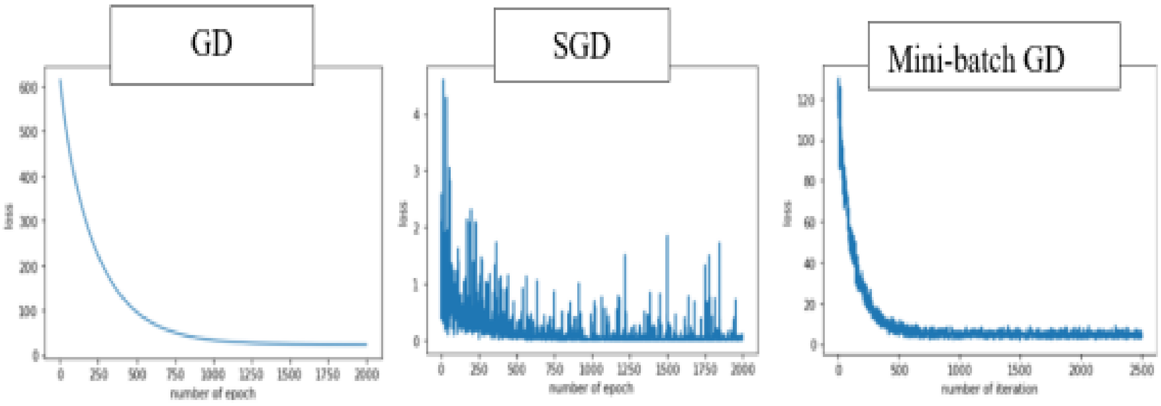


Figure 2.6: The loss function with gradient descent types[55]

2- **Adam Optimizer**: this is an optimization algorithm commonly used in machine learning and deep learning models to update the weights of the neural network during the training process. It is an extension of the stochastic gradient descent (SGD) algorithm that incorporates adaptive learning rates. [56], Adam's steps equations are:

$$w_t{}' = w_t - \frac{\eta}{\sqrt{S_{dw_t}{}'-\varepsilon}} * V_{dw_t}{}' \qquad \text{…. (2.15)}$$

$$S'_{dw_t} = \beta S_{dw_t} + (1-\beta)\,(\frac{\partial E_{total}}{\partial w_t})^2 \qquad \text{…. (2.16)}$$

$$where\ V_{dw_t}{}' = \beta V_{dw_t} + (1\text{-}\ \beta)(\frac{\partial E_{total}}{\partial w_t} \qquad \text{…. (2.17)}$$

Where , $S_{dw_t}{}'$ is an exponentially weighted average representation on a diagonal matrix, and β is the exponential decay rate The same equations are used with bias [56].The same equations are used with bias.

2.5 The Deep Learning Algorithms

Deep learning is a branch of machine learning that enables computers to learn from experience and understand the world in terms of a hierarchy of concepts. Where it is based on artificial neural networks, which are algorithms generated by the structure and processes of the brain of humans. Wherefore there is no deep learning without neural networks. In general, deep learning is more complex and takes more training time than a traditional neural network[57].

There are several algorithms for deep learning. In the next sub suction explain the most common ones:

2.5.1 The Convolutional Neural Networks (CNN)

It is a deep, feed-forward artificial neural network that retains hierarchical structure in image problems such as object recognition and other computer vision problems through studying internal feature representations and generalizing features. The most beneficial aspect of CNNs is reducing the number of parameters in ANN. In general, A CNN consists of multiple layers: the convolutional layers, pooling layers, and fully-connected (FC) layers[58].

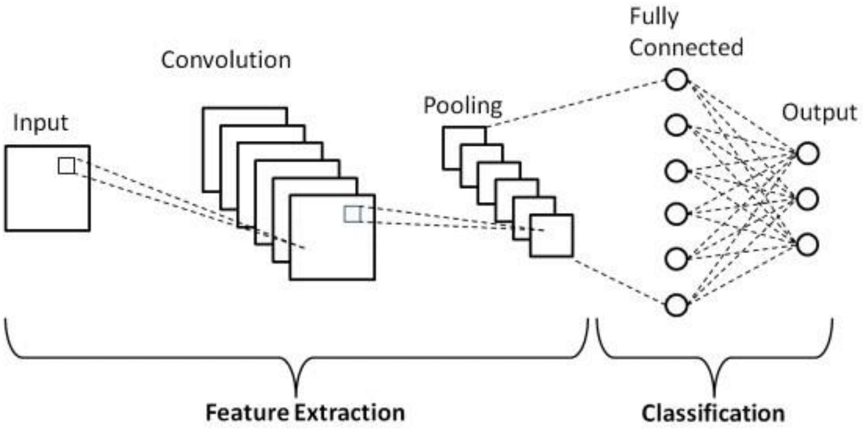


Figure 2.7: Convolutional Neural Network Architecture [58]

2.5.2 The Multilayer Perceptron

The Multilayer Perceptron (MLP) algorithm is a type of artificial neural network (ANN) that consists of multiple layers of interconnected nodes (neurons). MLPs are known for their ability to learn complex patterns and relationships in data. They can handle both numerical and categorical inputs, making them suitable for a wide range of applications. However, they may be prone to overfitting if not properly regularized or if the training data is limited. Regularization techniques, such as dropout or weight decay, can be employed to mitigate overfitting.

MLPs are a fundamental algorithm in deep learning and have been successfully applied to various real-world problems, including image classification, speech recognition, sentiment analysis, and financial forecasting, among others[59].

An MLP consists of an input layer, one or more hidden layers, and an output layer. Each layer is composed of multiple artificial neurons (also called nodes or units) that perform computations and transformations on the input data. Neurons in an MLP receive inputs, apply a transformation or activation function to the inputs, and produce an output. Activation functions introduce non-linearity to the network, enabling it to learn complex relationships. Common activation functions used in MLPs include sigmoid, hyperbolic tangent (tanh), and rectified linear unit (ReLU). Connections between neurons in different layers are assigned weights, which determine the strength or importance of the connection. Each neuron typically has a bias term that allows for adjusting the output based on a predetermined threshold. The training of an MLP involves adjusting the weights and biases to minimize the difference between predicted outputs and target outputs. This is typically done using an optimization algorithm, such as gradient descent, and a technique called backpropagation, which calculates the gradient of the loss function concerning the weights and biases. The gradients are then used to update the parameters iteratively. In forward propagation, input data is fed into the network, and the activations of the neurons are calculated layer by layer until the output layer produces the final predictions or outputs [60].

Backpropagation is the key algorithm used to train MLPs. It involves propagating the errors from the output layer back through the network, and calculating the gradients of the loss function concerning the weights and biases in each layer. These gradients are then used to update the parameters values in the opposite direction of the gradient, iteratively improving the network's performance [61]. Figure 2.11 illustrates the DNN's structure.

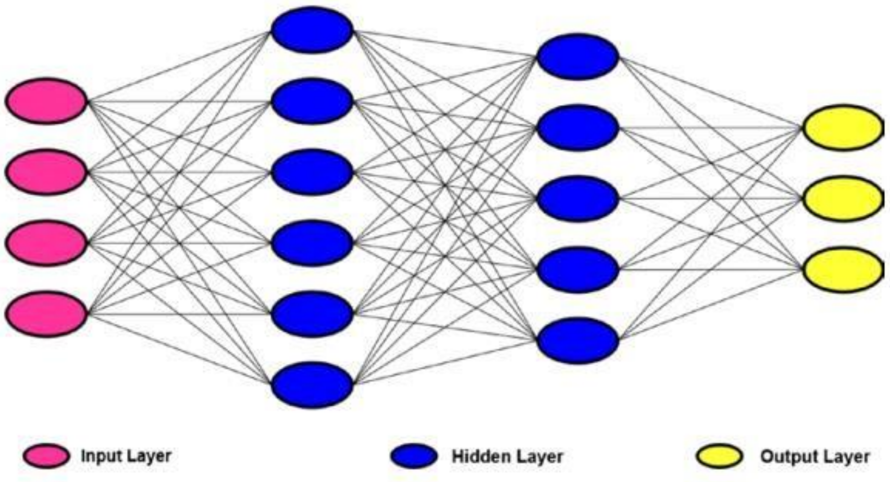


Figure 2.10: Deep neural network architecture

2.6 The Internet of Things

IoT refers to a various device, such as smart appliances and sensors, are interconnected and able to communicate with each other via the Internet. This network of connected devices finds applications in diverse areas, including smart cities, smart homes, and more [62]. The idea behind IoT is to create a seamlessly connected environment where objects can gather and share information in real-time, leading to improved efficiency, convenience, and automation in various aspects of our lives. These objects, often referred to as "smart" devices, can range from household appliances like refrigerators and thermostats to industrial machinery, healthcare devices, and even entire cities or infrastructure systems[63]. The key components of an IoT system typically include sensors or actuators, which collect and transmit data, a network infrastructure that enables communication between devices, and cloud-based platforms or applications that process and analyze the collected data. Additionally, artificial intelligence and machine learning technologies play a crucial role in extracting valuable insights from the vast amounts of data generated by IoT devices [64].

IoT has the potential to revolutionize numerous industries and domains. In healthcare, it can enable remote patient monitoring, smart pill dispensers, and personalized healthcare solutions. In agriculture, IoT can optimize irrigation systems, monitor soil conditions, and enhance crop yield. In transportation, it can improve traffic management, enable autonomous vehicles, and enhance logistics operations. These are just a few examples, as the possibilities of IoT applications are vast and continue to expand [65].

Overall, the Internet of Things has the potential to transform the way we live, work, and interact with our environment. It opens up a world of possibilities for innovation, automation, and efficiency, shaping a future where everyday objects are interconnected and intelligent, ultimately leading to a more connected and intelligent world [66].

2.6.1 The IoT Architecture

There are many architectures of IoT devices, each of which will be explained separately:

A. Three-layer architecture

The three-layer of IoT architecture typically begins with the most fundamental structures [67] as depicted in Figure (2.11). The subsequent framework embodies the core concept of IoT architecture.

1. **Perception layer:** This layer includes the sensors, which are believed to be the physical layer through which the environment may be perceived or even the devices near it can be detected or recognized [68].
2. **Network Layer:** This layer manages the connection between Internet of Things devices, servers, and network devices. It allows data to be transported for processing and storage between devices and between servers and network devices [68].
3. **The application layer**: This is near to the user since it offers him a variety of application services such as health care, smart cities, and other similar services [68].

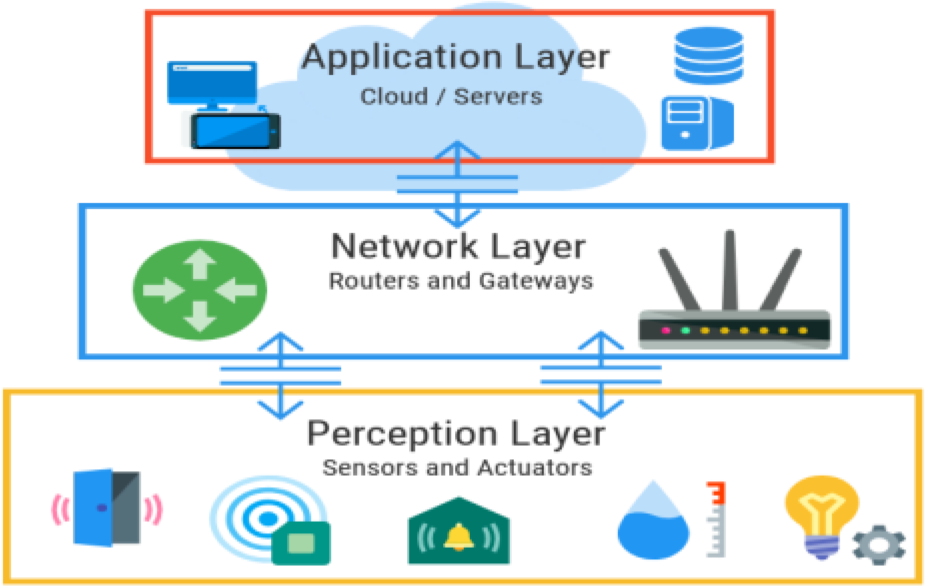


Figure 2.2: : Three-layer Internet of Things architecture[67]

2.6.2 Air quality sensors

air quality sensors play a vital role in improving our understanding of ambient air quality, which has a direct impact on human health and the environment. As the awareness of air pollution and its effects on public health increases, the demand for high-performance and portable online gas sensor detectors is indeed on the rise [69].

1. **MP503:** The MP503 is a specific type of gas sensor designed to detect and measure concentrations of various gases in the air. It is based on the Metal Oxide Semiconductor (MOS) principle, similar to many other gas sensors. The MP503 sensor is often used to detect carbon monoxide (CO) and methane (CH4) gases, making it suitable for applications such as indoor air quality monitoring, gas leak detection, and safety systems [70].
2. **MQ7:** The MQ-7 is a popular gas sensor module that is commonly used to detect and measure carbon monoxide (CO) gas concentration in the air. The MQ-7 sensor is widely used in various applications, including air quality monitoring, indoor safety systems, industrial environments, and automotive

applications. The MQ-7 sensor is sensitive to detecting carbon monoxide (CO) concentrations in the air in the range of 20 to 200 ppm (parts per million). he MQ-7 sensor provides an analog output voltage that varies with the concentration of carbon monoxide. This analog output can be interfaced with microcontrollers or analog-to-digital converters for further processing[71].

3. **MQ135:** The MQ-135 is another popular gas sensor module in the MQ series, Similar to the MQ-7, the MQ-135 gas sensor is widely used for air quality monitoring and is commonly employed in various indoor and outdoor applications to detect and measure the presence of specific gases. The MQ-135 sensor is sensitive to a variety of gases, including ammonia (NH3), nitrogen oxides (NOx), benzene (C6H6), smoke, and various other volatile organic compounds (VOCs). The MQ-135 sensor provides an analog output voltage that varies with the concentration of the target gases. The analog output can be interfaced with microcontrollers or analog-to-digital converters for data processing. The sensor is designed to operate with low power consumption, making it suitable for portable and battery-powered applications. It's important to note that the MQ-135 sensor may have cross-sensitivity to different gases, which means it could respond to gases other than its target gases. Calibration and understanding the specific environmental conditions are essential for accurate measurements [72].
4. **WSP2110:** The is a type of gas sensor module designed to detect the presence of carbon monoxide (CO) gas in the air. The WSP2110 sensor is sensitive to detecting carbon monoxide (CO) concentrations in the air in the range of 10 to 1000 ppm (parts per million). the sensor provides an analog output voltage that varies with the concentration of carbon monoxide. This analog output can be interfaced with microcontrollers or analog-to-digital converters for further processing and data acquisition. The sensor typically has a fast response time to changes in carbon monoxide concentration, enabling real-time monitoring.

The sensor is known for its stability and reliability, providing consistent readings over time [73].

2.7 The Performance Evaluation

The performance and effectiveness of the proposed system are evaluated by several metrics. A confusion matrix is a table that is often used to describe the performance of the proposed system [74]. A binary confusion matrix is described in Table (2.1).

Table 2-1: Confusion Matrix

		Predicted Classes	
		Positives	Negatives
Actual Classes	Positives	True Positive (TP)	False Negative (FN)
	Negatives	False Positive (FP)	True Negative (TN)

• TP: the number of positive records correctly classified.

• TN: the number of negative records correctly classified.

• FP: the number of positive records incorrectly classified.

• FN: the number of negative records incorrectly classified [74].

For the evaluation purpose, Accuracy, Precision, Recall, and F1-score are applied. These metrics are calculated as follows:

accuracy is a performance metric that measures the overall correctness of a classification model. It provides an estimate of how well the model predicts the true class labels.

Accuracy is calculated by dividing the number of correctly classified instances (true positives and true negatives) by the total number of instances in the dataset [75] (see Equation 2.17).

$$Accuracy\ = \frac{number\ of\ true\ classifications}{total\ number\ of\ classifications}$$

$$= \frac{TP + TN}{TP + FP + TN + FN} \quad ...\ (2.18)$$

A recall is the percentage of predicted positive class which are correctly classified by the classifier. [75] While the recall referred to as (R) is counted like the following:

$$\text{Recall } = \frac{TP}{TP+FN} \quad ....\ (2.18)$$

Precision is the percentage of predicted positive class, which are actually positive class. The precision referred to as (P), [75] is counted like the following equation:

$$\text{Precision} = \frac{TP}{TP+FP} \quad ....\ (2.19)$$

The F1-score is the harmonic mean of precision and recall, giving equal weight to both measures [75].

$$\text{F Score} = 2\ \frac{\text{Precision} * \text{Recall}}{\text{Precision}+ \text{Recall}} \ ....\ (2.20)$$

The efficiency of the proposed system has been evaluated through the number of measures based on the confusion matrix.

2.8 The Hardwar and Software Used to Execute the Proposed System

The tools used to execute a proposed system can vary based on the nature of the project, its requirements, and the technologies involved. Below are some types of tools that might be utilized in different stages of executing a system

2.8.1 The IoT Environments

IoT environments require a variety of tools these tools help create a seamless and efficient IoT ecosystem. Here are some categories of Hardwar part commonly used in IoT environments:

A. Arduino Uno

It is a popular microcontroller board based on the ATmega328P microcontroller. It is a member of the Arduino family and is widely used for various electronics and prototyping projects[76]. Here's a brief overview of Arduino Uno:

1) **Microcontroller**: Arduino Uno is built around the ATmega328P microcontroller from Atmel (now Microchip). It operates at a clock speed of 16 MHz and has 32KB of Flash memory for storing the program, 2KB of SRAM for data storage, and 1KB of EEPROM for non-volatile storage [76].
2) **Digital and Analog I/O**: Arduino Uno has 14 digital input/output (I/O) pins, among which 6 can be used for pulse width modulation (PWM) output. It also has 6 analog input pins for reading analog voltage levels [76].
3) **Communication Interfaces:** Arduino Uno features a USB interface for connecting to a computer for programming and serial communication. It also has a dedicated ICSP (In-Circuit Serial Programming) header for advanced programming and communication [76].
4) **Power Supply**: Arduino Uno can be powered through a USB connection or an external power source (7-12V DC). It has a built-in voltage regulator that provides a stable 5V supply to the microcontroller and other components [76].
5) **Programming:** Arduino Uno can be programmed using the Arduino Integrated Development Environment (IDE), which is a user-friendly software environment for writing, compiling, and uploading code to the board. The programming language is based on C/C++, with simplified libraries and functions provided by the Arduino framework [76].
6) **Extensibility:** Arduino Uno is designed to be highly extensible and can be easily connected to various sensors, actuators, and modules through its I/O pins. There is a wide range of Arduino-compatible shields and modules

available that can be stacked on top of the board for additional functionalities [76].

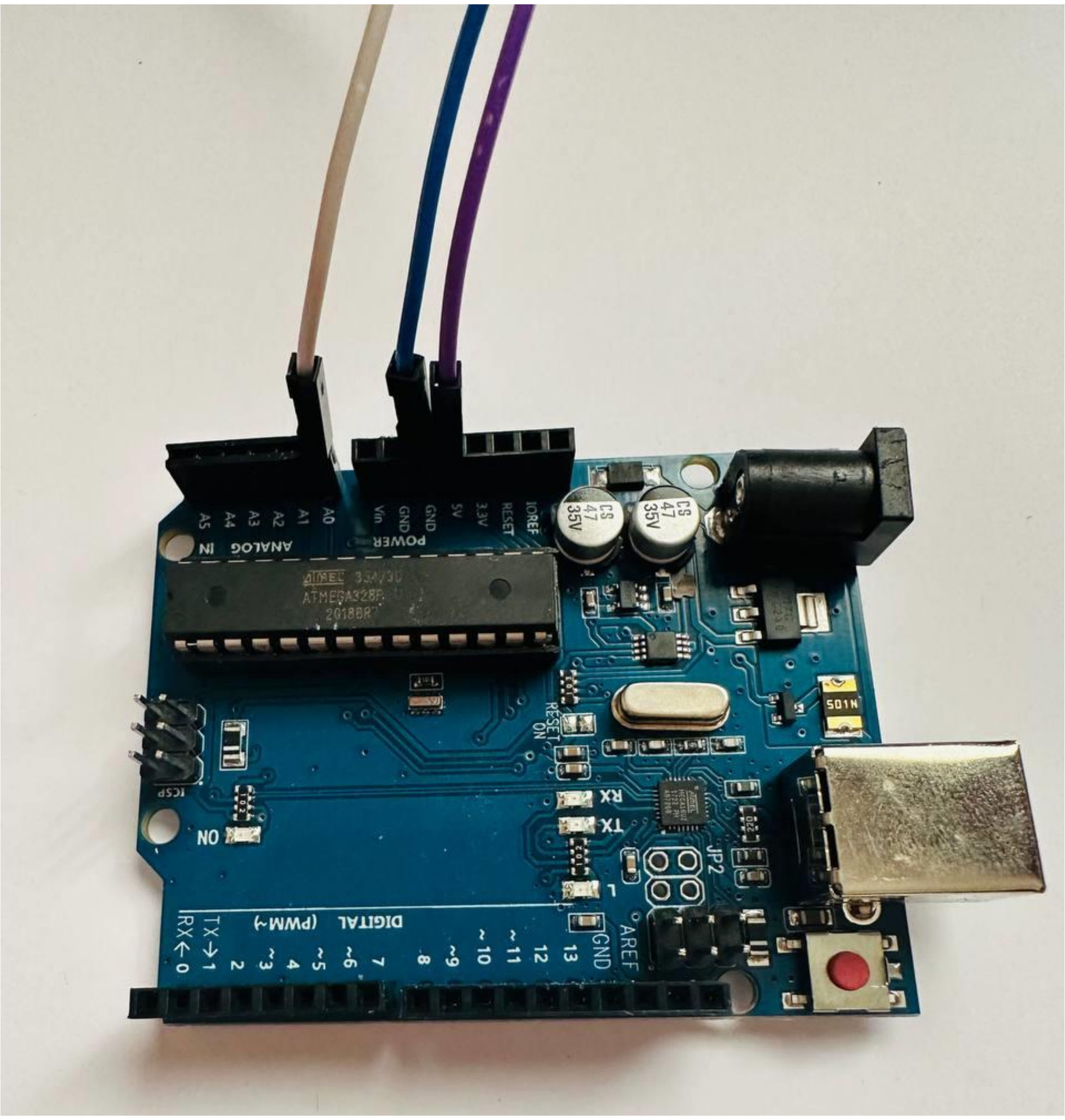

ANALOG IN
A5 A4 A3 A2 A1 A0
POWER
Vin GND GND 5V 3.3V RESET IOREF
CS 47 35V
ATMEGA328P
ICSP
ON
RESET
RX
TX
L
JP2
DIGITAL (PWM~)
RX←0 TX→1 2 ~3 4 ~5 ~6 7 8 ~9 ~10 ~11 12 13 GND AREF

Figure 2.3: Arduino Uno

Arduino Uno is beginner-friendly, versatile, and widely supported, making it a popular choice for hobbyists, students, and professionals in the field of electronics and programming. It provides a convenient platform (see Figure 2.13) for experimenting, prototyping, and creating interactive projects [77].

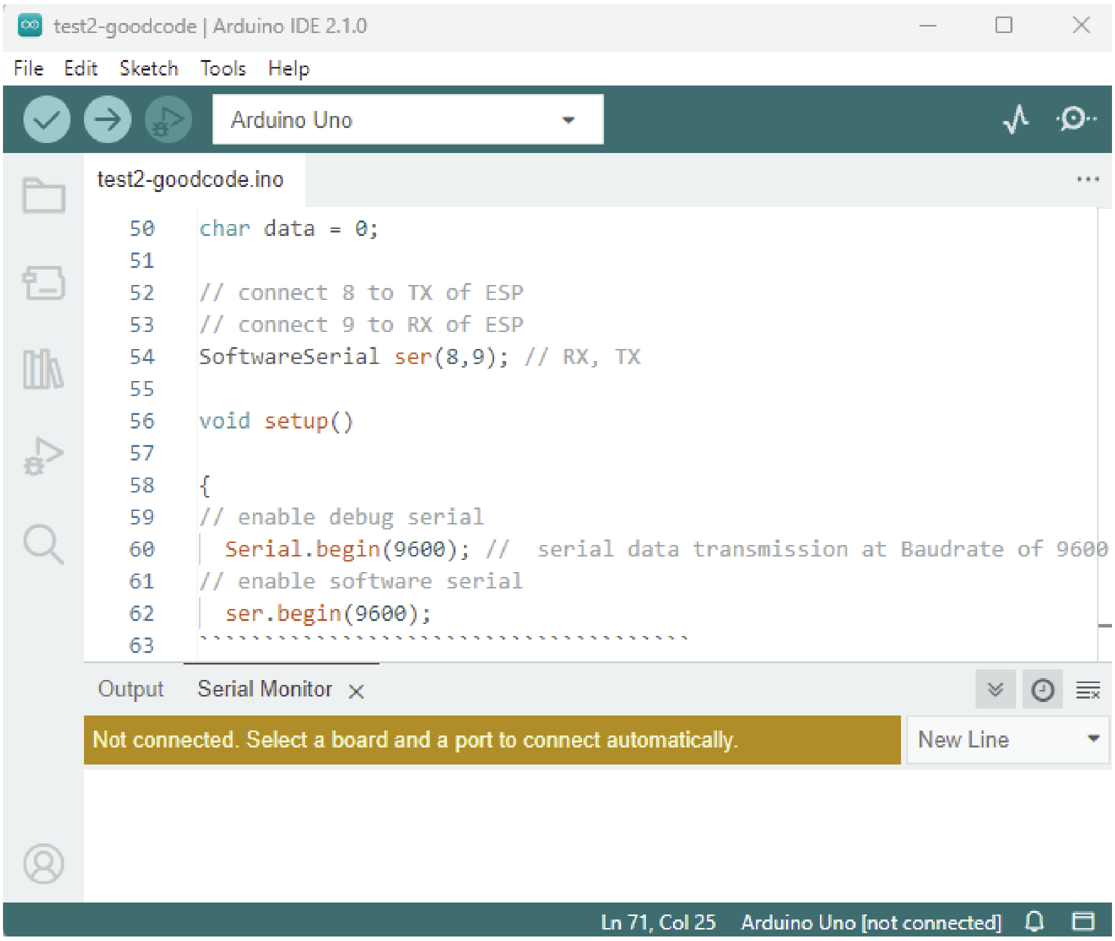
B. Arduino IDE

The Arduino IDE (Integrated Development Environment) is a software application used to write, compile, and upload code to Arduino boards. It provides a user-friendly interface for programming Arduino microcontrollers and simplifies the process of developing projects [78].

Here are some key features and functionalities of the Arduino IDE:

1) Code Editor: The Arduino IDE offers a text editor where we can write the code. It supports syntax highlighting, auto-indentation, and code completion, which helps in writing and organizing the code efficiently [79].
2) Library Manager: The IDE includes a library manager that allow to easily search, install, and manage libraries. Libraries provide pre-written code that can be used to interact with various components and modules, saving time and effort in writing low-level code [79].
3) Sketches: In the Arduino IDE, projects are referred to as "sketches." A sketch typically consists of two main functions: setup() and loop(). The setup() function is executed once when the Arduino board is powered on or reset, while the loop() function is executed repeatedly as long as the board is powered on [79].
4) Board Manager: The Arduino IDE supports a wide range of Arduino boards. The board manager allows to select the appropriate board from a list and install the necessary board-specific configurations and libraries [79].
5) Serial Monitor: The IDE provides a built-in serial monitor that allows to send and receive data between the Arduino board and the computer via the USB connection. It is useful for debugging and monitoring the output of the code during runtime [79].
6) Upload: With the Arduino IDE, can easily upload the code to the Arduino board. It handles the compilation process and communicates with the board to transfer the compiled code and execute it [79].

7) Examples and Tutorials: The IDE includes a collection of example sketches that demonstrate various functionalities and concepts. It also provides access to online tutorials and resources to get started and learn Arduino programming [79] as showed in Figure 2.4) .



```
char data = 0;

// connect 8 to TX of ESP
// connect 9 to RX of ESP
SoftwareSerial ser(8,9); // RX, TX

void setup()

{
// enable debug serial
  Serial.begin(9600); //  serial data transmission at Baudrate of 9600
// enable software serial
  ser.begin(9600);
```

Not connected. Select a board and a port to connect automatically.

Figure 2.5: Arduino IDE

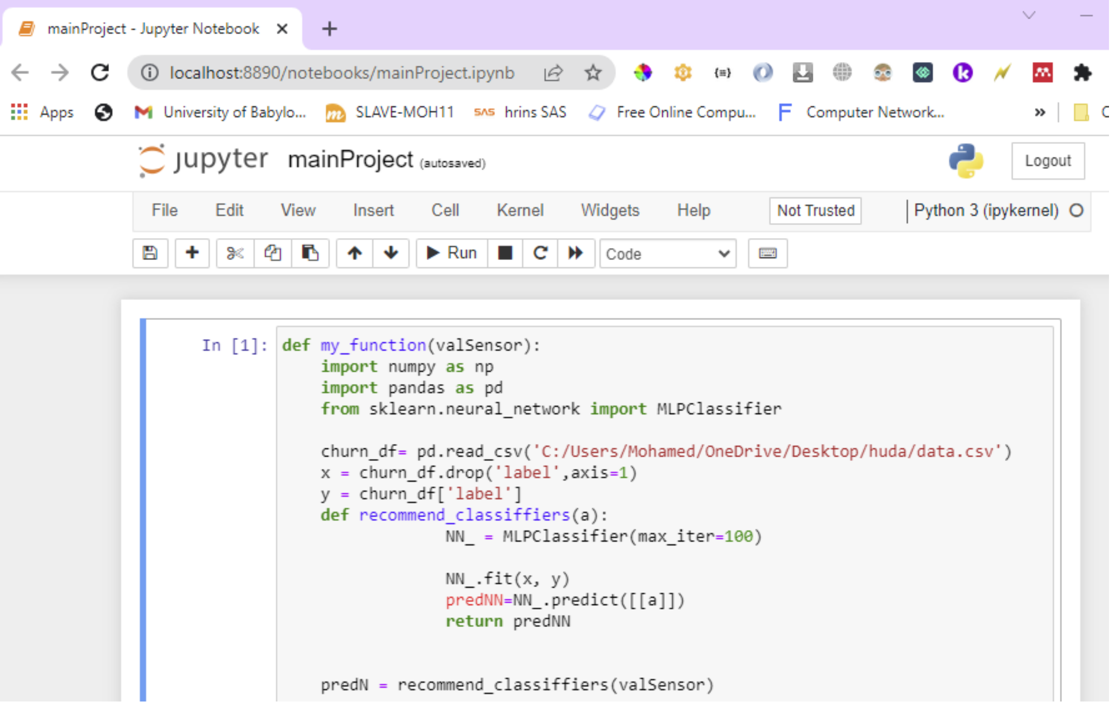
The Arduino IDE is open-source and available for Windows, macOS, and Linux platforms. It provides a beginner-friendly environment for programming Arduino boards and is widely used by hobbyists, students, and professionals for prototyping and creating interactive projects [79].

2.8.2 The ML Environment

A. Jupyter Notebook

Jupyter Notebook is an open-source web application that allows to create and share documents containing live code, equations, visualizations, and narrative text. It is a popular choice for data analysis, data visualization, and machine learning tasks, among others.

Jupyter Notebook supports multiple programming languages, including Python, R, Julia, and more. However, Python is the most widely used language in Jupyter Notebook [80].



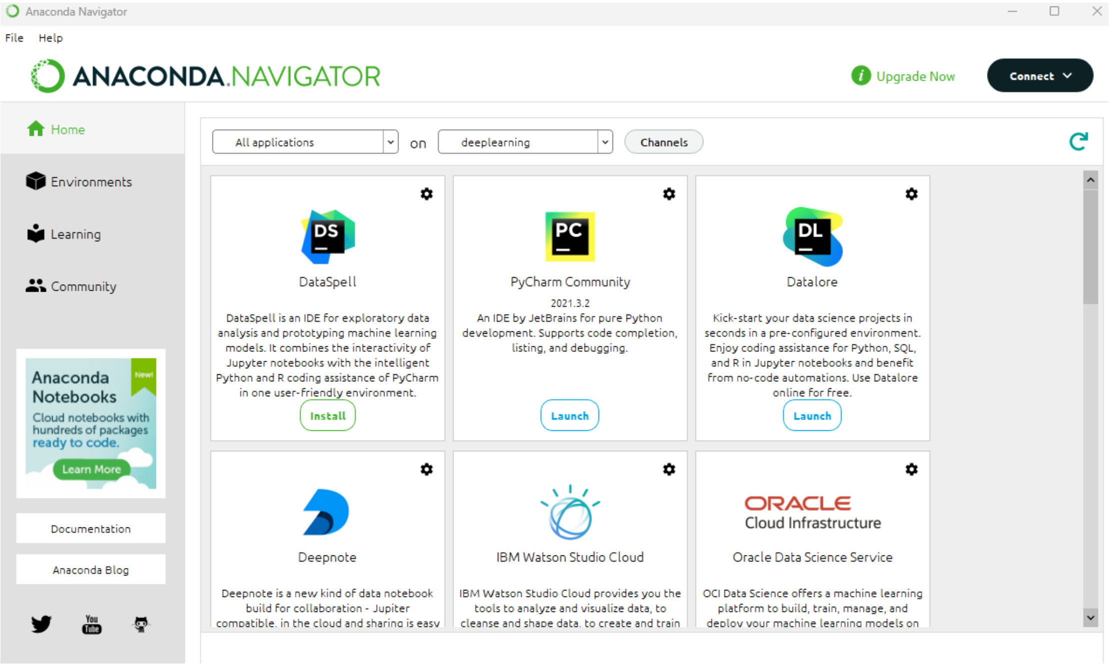
```
In [1]: def my_function(valSensor):
            import numpy as np
            import pandas as pd
            from sklearn.neural_network import MLPClassifier

            churn_df= pd.read_csv('C:/Users/Mohamed/OneDrive/Desktop/huda/data.csv')
            x = churn_df.drop('label',axis=1)
            y = churn_df['label']
            def recommend_classiffiers(a):
                         NN_ = MLPClassifier(max_iter=100)

                         NN_.fit(x, y)
                         predNN=NN_.predict([[a]])
                         return predNN


            predN = recommend_classiffiers(valSensor)
```

Figure 2.6: Jupyter Notebook

B. ANACONDA

Anaconda is an open-source distribution of Python programming languages for scientific computing, data science, and machine learning. It is a popular choice among data scientists and developers due to its vast collection of pre-installed libraries and tools for data analysis, visualization, and model development.

Anaconda comes with its package management system called Conda, which allows to easily install, manage, and update packages and dependencies. It also provides a virtual environment manager, allowing to create isolated environments with specific package versions for different projects [81].



DataSpell is an IDE for exploratory data analysis and prototyping machine learning models. It combines the interactivity of Jupyter notebooks with the intelligent Python and R coding assistance of PyCharm in one user-friendly environment.

An IDE by JetBrains for pure Python development. Supports code completion, listing, and debugging.

Kick-start your data science projects in seconds in a pre-configured environment. Enjoy coding assistance for Python, SQL, and R in Jupyter notebooks and benefit from no-code automations. Use Datalore online for free.

Deepnote is a new kind of data notebook build for collaboration - Jupiter compatible, in the cloud and sharing is easy

IBM Watson Studio Cloud provides you the tools to analyze and visualize data, to cleanse and shape data, to create and train

OCI Data Science offers a machine learning platform to build, train, manage, and deploy your machine learning models on

Figure 2.7: Anaconda platform

C. PySerial

PySerial is a Python library that provides a way to communicate with serial ports. It allows to read from and write to serial devices, such as microcontrollers, Arduino boards, and other hardware devices that communicate via serial communication.

To establish communication between Jupyter Notebook (Python) and an Arduino Uno (C++), we can use PySerial on the Python side to send and receive data via the serial port. On the Arduino side, we can use the Serial library to read and write data to the serial port [82].

Chapter Three

The Proposed System Design

CHAPTER THREE

The Proposed System Design

3.1 Overview

This chapter explains the methodology of the proposed system in detail. The proposed system represents an air quality monitoring system using deep neural network techniques and applied in IoT. In this research work, datasets (CPCB) were applied to assess this work.

3.2 The System Architecture

The layout of the proposed system consists of four major stages, each stage is playing a crucial role in the development and implementation of the system. The following are proposed system phases.

3.2.1 The Dataset Description

In this stage, a comprehensive overview of the datasets used in the system is provided. The goal is to describe the characteristics, structure, and properties of the datasets, giving a clear understanding of the data that will be used in subsequent stages.

3.2.2 The Preprocessing Stage

This stage involves preparing the input data for further preprocessing and analysis. Tasks such as data cleaning, data normalization, feature selection, and other necessary data transformations are performed to ensure the data is in a suitable format for the next stages.

3.2.3 The Learning Stage (Model Formation):

Once the preprocessing stage is complete, the next step is to construct a classifier model using deep learning techniques. Deep learning focuses on training artificial neural networks with multiple layers to extract patterns from data. The classifier model aims to predict or classify data into different categories based on the patterns learned during learning stage.

3.2.4 The Evaluation Stage:

After building the classifier model, it needs to be evaluated to assess its performance and effectiveness. This stage involves measuring various metrics such as accuracy, precision, recall, and F1 score to determine how well the model performs on the given dataset. Evaluation helps identify the strengths and weaknesses of the classifier model and may lead to further refinement or optimization.

3.2.5 The Integration into an IoT Environment

The fourth stage involves integrating the developed classifier model into an Internet of Things (IoT) environment. IoT refers to a network of interconnected devices that can communicate and exchange data. In this stage, the classifier model can be deployed on IoT devices or systems, enabling real-time or remote classification of data collected from IoT sensors or devices. This integration enables various applications and scenarios where the classifier model can make predictions or decisions based on incoming IoT data.

Overall, these four stages aim to preprocess the data, build a deep learning classifier model, evaluate its performance, and integrate it into an IoT environment for real-time or remote classification.

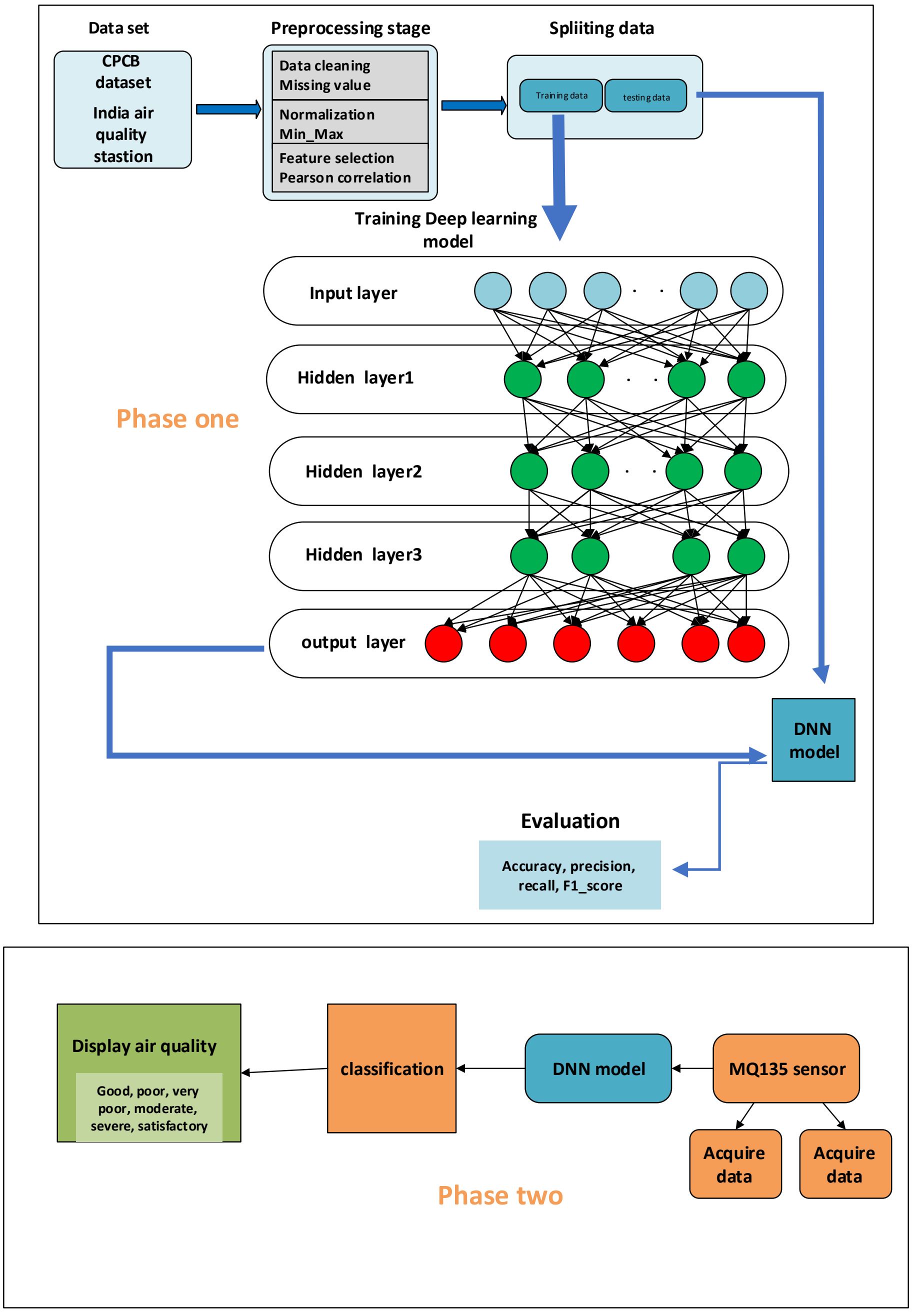


Figure 3.1: The Main Block Diagram of the System Architecture

3.3 The datasets description

The proposed system will employ the CPCB dataset for the purpose of constructing and evaluating the air quality monitoring system. This dataset encompasses air quality measurements and Air Quality Index (AQI) readings obtained at the daily level across multiple cities in India. With a total of 29,531 records, the CPCB dataset is comprised of 12 attributes that provide valuable information regarding air quality parameters. Notably, the dataset encompasses six distinct classes, namely Good, Poor, Very Poor, Severe, Moderate, and Satisfactory, which serve as indicators for the categorization of air quality levels. The detailed characteristics of the dataset features can be found in Table 3.1.

Table 3-1: Details of CPCB Dataset

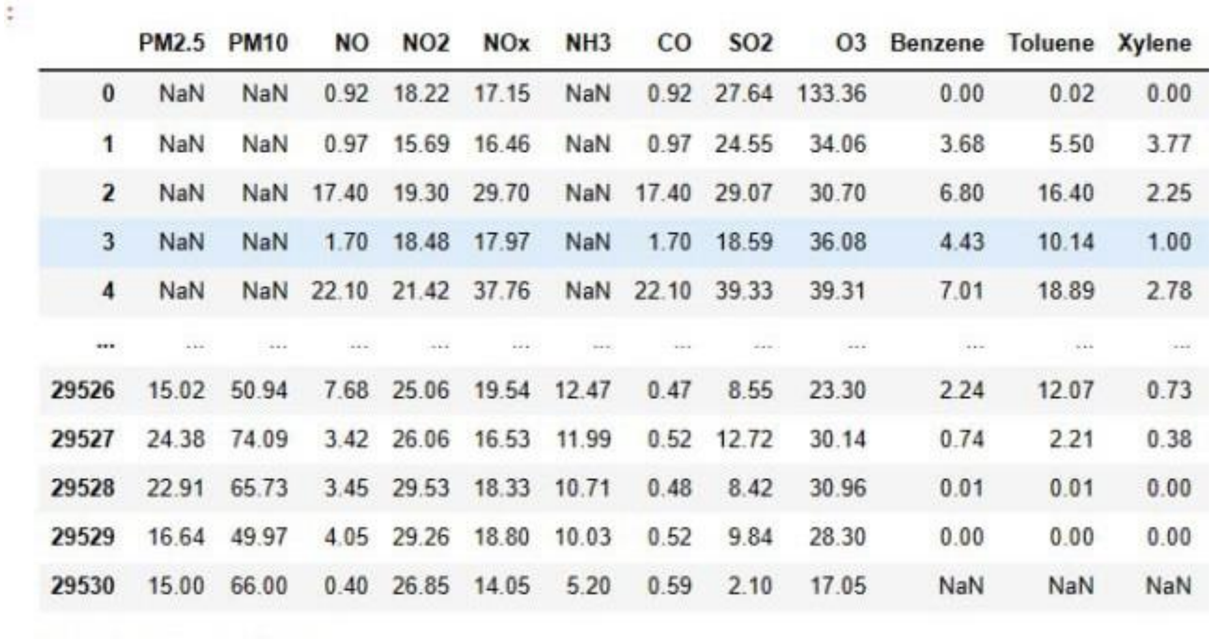
Dataset Key	Description
Record Count	29,531
Number of Attributes	12
Air Quality Data	Daily level across multiple cities in India
Air Quality Index (AQI)	Provided for each record
Number of Classes	6 (Good, Poor, Very Poor, Severe, Moderate, Satisfactory)

3.4 The Preprocessing Stage

Prior to applying the deep learning classifier to the CPCB dataset, it is essential to perform suitable preprocessing and reformatting procedures. In the proposed system, as depicted in Figure 3.1, this preprocessing stage can be described as consisting of two primary steps: data cleaning to handle missing values, and normalization of normal numerical values.

3.4.1 The Data Cleaning

The quality of data plays a crucial role in achieving effective visualization and developing efficient Deep learning models. In the CBCP dataset, there are instances of not-a-number (NaN) values of all the features in the dataset. It is important to note that among all the features, Xylene has the highest number of missing values, while CO has the least. To address this issue, data cleaning procedures are necessary. To handle the missing values, a common approach is to replace them with the mean values specific to each feature. By calculating the mean value for each feature and filling the missing values accordingly, the second part of data cleaning is to remove the personal columns (date, city) which, they are unneeded in the proposed model. the dataset can be cleansed of these inconsistencies. This step ensures that the dataset becomes suitable for further analysis and modeling. Figure (3.2) provides a visual representation of the dataset prior to addressing the missing values, illustrating the presence of NaN values within the dataset. By resolving these missing value problems through data cleaning, the dataset is transformed into a clean and complete form, laying the foundation for subsequent analysis, visualization, and the development of accurate Deep learning models.



	PM2.5	PM10	NO	NO2	NOx	NH3	CO	SO2	O3	Benzene	Toluene	Xylene
0	NaN	NaN	0.92	18.22	17.15	NaN	0.92	27.64	133.36	0.00	0.02	0.00
1	NaN	NaN	0.97	15.69	16.46	NaN	0.97	24.55	34.06	3.68	5.50	3.77
2	NaN	NaN	17.40	19.30	29.70	NaN	17.40	29.07	30.70	6.80	16.40	2.25
3	NaN	NaN	1.70	18.48	17.97	NaN	1.70	18.59	36.08	4.43	10.14	1.00
4	NaN	NaN	22.10	21.42	37.76	NaN	22.10	39.33	39.31	7.01	18.89	2.78
...	...	...	...	...	...	...	...	...	...	...	...	...
29526	15.02	50.94	7.68	25.06	19.54	12.47	0.47	8.55	23.30	2.24	12.07	0.73
29527	24.38	74.09	3.42	26.06	16.53	11.99	0.52	12.72	30.14	0.74	2.21	0.38
29528	22.91	65.73	3.45	29.53	18.33	10.71	0.48	8.42	30.96	0.01	0.01	0.00
29529	16.64	49.97	4.05	29.26	18.80	10.03	0.52	9.84	28.30	0.00	0.00	0.00
29530	15.00	66.00	0.40	26.85	14.05	5.20	0.59	2.10	17.05	NaN	NaN	NaN

Figure 3.2: The sample of features with their missing values in the dataset

3.4.2 The normalization processes

In this thesis, the Min-Max normalization method has been utilized to normalize the dataset as mention in equation (2.1). Min-Max normalization rescales the values to a specified range, typically between 0 and 1, by subtracting the minimum value from each data point and dividing it by the range (the difference between the maximum and minimum values). By applying Min-Max normalization, the dataset's large numerical values are effectively minimized, enabling more efficient mathematical operations and facilitating the implementation process of the system while preserving the relative importance of the data.

3.4.3 The feature selection

The correlation feature selection method is employed to select features based on their correlation coefficients. By calculating the correlation coefficient between the AQI and other pollutant variables, the strength and direction of their relationship can determine by analyzing the correlation coefficients, the pollutants can be identified that have a stronger correlation with the AQI as mention in equation (2-2). In this case the pollutants PM10, PM2.5, CO, NO2, SO2, NOX, and NO these pollutants generally exhibit a positive correlation with the AQI, indicating that higher concentrations of these pollutants contribute to higher AQI values. Algorithm (3.1) represent the Correlation coefficient.

Algorithm (3.1): Correlation Coefficient

```
# Initialize empty lists for AQI and other pollutant variables
AQI_ data = []
pollutant_ data = []

for i in range(n):
    AQI_data.append(AQI)
    pollutant_data.append(pollutant_value)

# Calculate the mean of AQI and the pollutant variable
mean_AQI = sum (AQI_data) / n
mean_pollutant = sum(pollutant_data) / n

# Calculate the numerator and denominators for Pearson's r
numerator = 0
denom_AQI = 0
denom_pollutant = 0

for i in range(n):
    numerator += (AQI_data[i] - mean_AQI) * (pollutant_data[i] - mean_pollutant)
    denom_AQI += (AQI_data[i] - mean_AQI) ** 2
    denom_pollutant += (pollutant_data[i] - mean_pollutant) ** 2

# Calculate Pearson's correlation coefficient
Pearson = numerator / sqrt(denom_AQI * denom_pollutant)
```

3.5 The deep neural network model

Deep neural networks are widely recognized as one of the most prominent types of deep learning models. In this thesis, a deep neural network is employed, which constitutes an artificial neural network comprising multiple hidden layers. This section aims to elucidate the algorithm of the deep neural network in four steps: the model's architecture, forward propagation with the perceptron classifier and activation function, total error with the loss function or cost function, and backpropagation with the optimizer.

3.5.1 The Model Architecture

The model architecture of the Multilayer Perceptron (MLP) comprises interconnected layers of artificial neurons, also referred to as nodes or units. The MLP model consists of an input layer, Three hidden layers, and an output layer.

1. **Input Layer:** This layer represents the initial layer that receives input data. In the proposed system, the input layer consists of 7 nodes corresponding to the number of features in the preprocessed CPCB dataset (Pm2.5, Pm10, NOx, NO2, NO, O3, CO)
2. **Hidden Layers:** The hidden layers are situated between the input and output layers and perform the computation necessary for the model. In the multiclass classification model employed in this work, three hidden layers are utilized in the first hidden layer which composed of 30 nodes, in the two hidden layer 20 nodes and in the last hidden layer composed of 10 nodes
3. **Output Layer:** The output layer produces prediction results, specifically in multiclass classification, where it identifies the type of air pollution. there are 6 nodes in output layer which are (Good, Poor, very poor, Moderate, Sever, Satisfactory).

All nodes in the input layer are fully connected to all nodes in the subsequent hidden layer and subsequent layers, forming a connected graph. Figure 3.3 illustrates this deep neural network architecture.

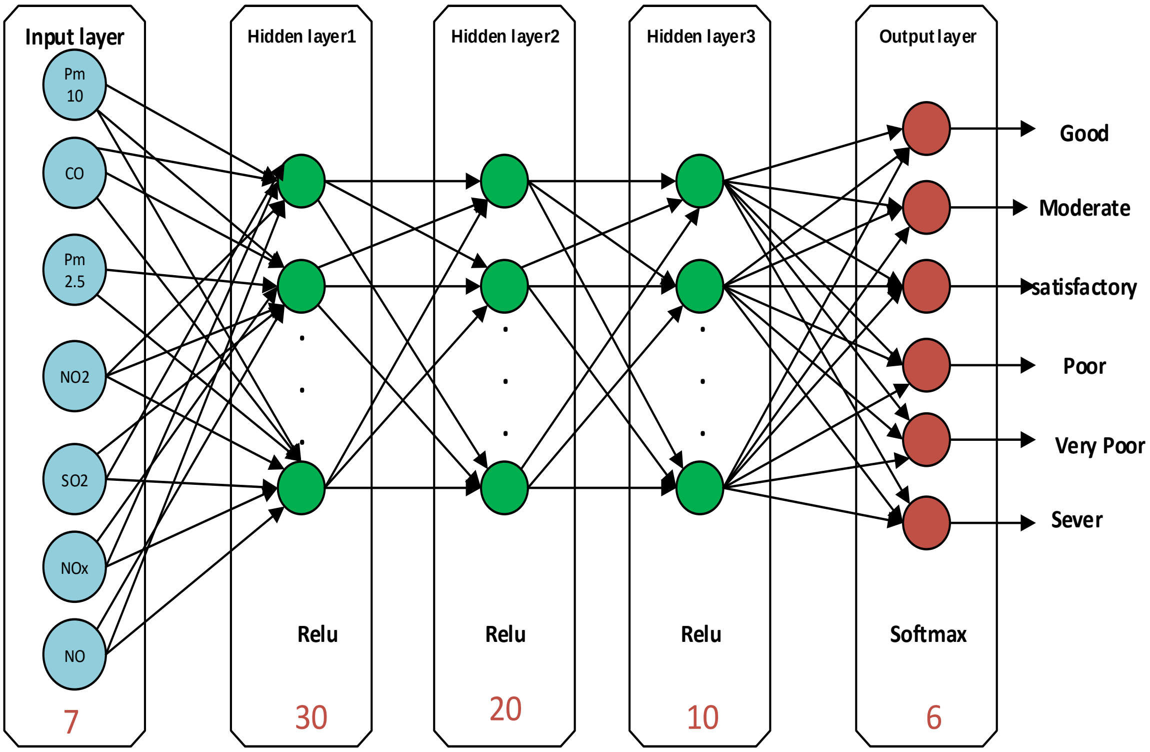


Figure 3.3: Deep neural network archoticture

3.5.2 The Feedforward Propagation

The objective of forward propagation is to predict results by employing a perceptron classifier and activation functions. The perceptron equation, as mentioned in equation (2.3), requires data (X), the number of nodes (n), weights (w), and bias (b). In the input layer, data (X) represents an array of values from the preprocessed CPCB datasets. In other layers, data (X) represents the results of the perceptron equation and activation function from the previous layer. The number of nodes (n) is determined by the model architecture. The weights (w) and bias (b) initially have an initial value (set to 1) for all layers. Subsequently, the weights (w) and bias (b) takes on values obtained from the optimizer results. The results of the perceptron equation serve as the input values for the activation system, as illustrated in Figure (3.4).

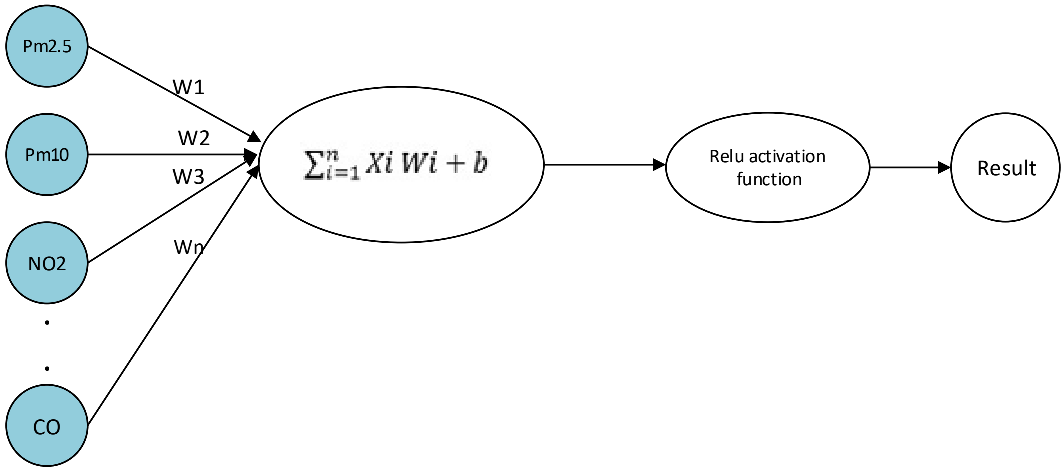


Figure 3.4 : Feedforward propogation baseline

The activation function plays a crucial role in determining how a neuron behaves, akin to the activation potential observed in neurons within the human brain. It defines the threshold for neuron activation and restricts the output within a defined range.

Several commonly used activation functions include Sigmoid, ReLU, Softmax, and Tanh. In this model, ReLU activation functions are utilized in the hidden layers, while softmax activation functions are employed in the output layer. The ReLU activation function acts as a gate between the current and subsequent layers, eliminating negative outputs. softmax activation functions transform the output predictions into categorical probability variables, providing insights into the type of pollution. Refer to Algorithm 3.2 for a representation of the forward propagation process within the deep learning classifier.

Algorithm (3.2): The forward pass

```
Input: preprocessed CPCB dataset
Output: prediction results (multiclass)
FUNCTION ReLU (x)
  Return  max(0,x)  //  as mentioned in equation (2.7)
```

```
END FUNCTION
//returns a value between 0 and 1
2-FUNCTION  softmax (x)
      Return   $f(x) = \frac{e(x_i)}{\sum_j e(x_i)}$   //  as mentioned in equation (2.8)
END FUNCTION
3-PROCEDURE forwardPass()
 // HIDDEN LAYERS
    FOR each hiddenLayers // 3 hidden
      FOR each neurons in hiddenLayers
         Initialize  weightedSum  to zero
         Initialize  bias  to zero
         Initialize  sum  to zero
            FOR each neuron's links
             Set  weightedSum to Multiply link's weight with  associated previous Layer's
              neuron's value
            END FOR
        Set sum to  weightedSum with adding to bias of layer
                 //as mentioned in equation (2.4)
    Call  ReLU (sum) //using activation function to the results
    Set neuron's value to result //  results after activation function
    END FOR // neurons of hiddenLayers
   END FOR // hiddenLayers
 // MULTICLASS OUTPUT LAYER
// 6  neurons in multiclass prediction
 FOR each outputLayer's neurons
     Initialize  weightedSum  to zero
     Initialize  bias  to zero
     Initialize  sum  to zero
    FOR each AQI_Outputs neuron's links
         Set weightedSum to ruselt of Multiply link's weight  with associated previous
         Layer's neuron's value
    END FOR
    Call softmax(sum) //using activation function (AV) to the results
    Set neuron's value to result //the results after AV function   END FOR
END PROCEDURE
```

3.5.3 The Total Error

The Total Error is a crucial aspect in assessing the accuracy of a neural network by measuring the disparity between its output and the actual solution. It is determined by comparing each prediction result with its corresponding real output value using mean square error (MSE) and summing up these differences. The concept and details of mean square error as mentioned in equation (2.8). However, for classification algorithms, including the one utilized in this thesis, cross-entropy is considered more suitable than mean square error. Cross-entropy helps mitigate the issue of large total errors in initial predictions and results in lower loss. Specifically, in this thesis, multiclass cross-entropy can be employed since our classification problem involves multiple classes.

In Algorithm 3.3, the predicted values are derived from the output values of the Forward Pass algorithm (3.2), while the actual values are obtained from the class labels of the training data.

Algorithm (3.3): Total Error using Cross entropy

```
Input: expectedValue(predict Value ) and realvalue(from the training dataset)
Output: result to totalError
PROCEDURE calculateTotalError()
            Initialize totalError to zero
    FOR each outputLayer's neurons
        IF neuron == expected_output neuron THEN
            Set neuron's expectedValue to 1 // true expected
        ELSE
            Set neuron's expectedValue to 0 // false expected
        END IF
        FOR each class neurons
            Do neuron's value of class * log (neuron's expectedValue of class) //as mentioned  in equation (2.12)
            Add result to totalError
        END FOR
    END FOR
END PROCEDURE
```

3.5.4 The Backpropagation Process

The primary objective of the backpropagation process is to train the deep neural network by adjusting its weights and biases, as depicted in Figure 3.5. Backpropagation utilizes the calculated errors from the previous forward propagation stage, incorporating the loss function and optimizer. The selection of optimization methods and loss functions is crucial for achieving optimal and efficient performance. In this thesis, the cross-entropy loss function is employed, and the Adaptive Moment Estimation (Adam) optimizer is utilized.

Gradient descent is a widely used optimization algorithm, particularly in neural networks. It represents the most common optimizer in the context of deep learning. The Adam optimizer is an enhanced version of gradient descent, combining Root Mean Square propagation (RMSprop) and Stochastic Gradient Descent with momentum. It employs a multi-learning rate approach for each weight and incorporates the momentum method during each training iteration. These features make Adam optimizer superior to traditional gradient descent. In essence, Adam optimizer is specifically designed for training deep neural networks and is considered the best optimizer to the best of our knowledge.

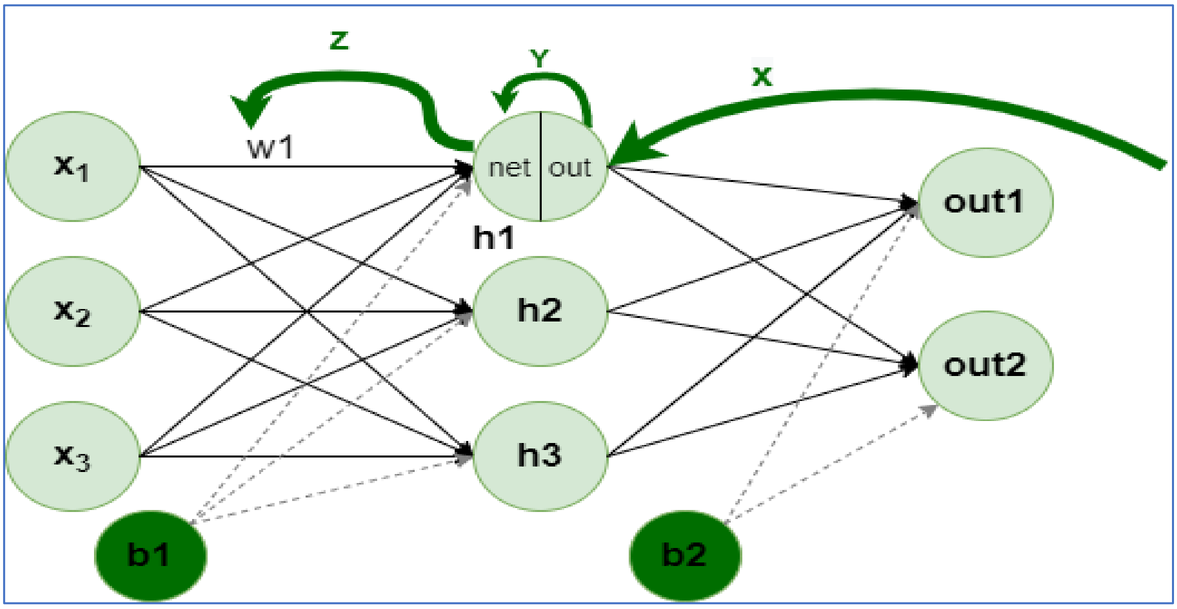


Figure 3.5 : The backpropogation process

3.6 The Evaluation Metrics

When evaluating air pollution using a Multilayer Perceptron (MLP), the choice of evaluation metrics depends on the specific task or objective at hand. Here are a few commonly employed metrics for assessing the performance of an MLP in air pollution-related tasks:

1. **Accuracy:** Accuracy is a frequently used metric for classification tasks. If the MLP is employed to classify air quality into different categories (e.g., Good, Moderate, Poor), accuracy can be calculated by comparing the predicted labels with the actual labels.
2. **Precision, Recall, and F1-Score:** These metrics are commonly utilized in classification tasks, particularly when dealing with imbalanced datasets.

Precision measures the proportion of true positive predictions out of all positive predictions, while recall measures the proportion of true positive predictions out of all actual positives. The F1-score combines precision and recall, providing a single metric that balances both measures.

3. **Confusion Matrix:** A confusion matrix provides a detailed breakdown of the MLP predictions and their alignment with the actual values. It enables analysis of true positives, true negatives, false positives, and false negatives, facilitating the identification of specific areas for improvement or potential issues.

These metrics serve as a starting point for evaluating the performance of an MLP in air quality-related tasks, as will be demonstrated in Chapter 4. However, it is crucial to select metrics that align with the specific objectives and consider domain-specific factors when evaluating air pollution models.

3.7 The IoT Environment

In this section, the IoT aspect of the proposed system and its application in predicting air quality will discussion. The air quality monitoring system encompasses several components, including the Arduino Uno, MQ135 sensor, and a main controller or PC.

3.7.1 The IoT Components

The integration of IoT components is vital for implementing the air quality prediction system. The following components are utilized:

A. Arduino Uno: The Arduino Uno serves as a microcontroller board that functions as the central controller of the air quality monitoring system. It provides the necessary processing power and interfaces to connect and communicate with other components.

B. MQ135 Sensor: The MQ135 sensor is a gas sensor commonly employed for detecting and measuring air quality parameters such as ammonia, nitrogen oxide, carbon monoxide, and other harmful gases. It plays a crucial role in collecting real-time air quality data. Algorithm (3.4) explaining assumes the use of an Arduino board and proper connection of the MQ135 sensor to the designated analog pin (mq135Pin). Additionally, the algorithm can be customized by adjusting the delay duration according to specific requirements.

C. The main controller/PC: The main controller or PC is responsible for coordinating the operation of the Arduino Uno and the MQ135 sensor. It collects the sensor readings, processes the data, and communicates with the trained MLP model for air quality prediction.

Algorithm (3.4): Read Analog Values from MQ135 Sensor

```
Inputs: - MQ135PIN: Analog pin connected to the MQ135 sensor data
Outputs: Sensor Values
Algorithm Steps:
    1. Initialize the necessary variables and pins:
        - Initialize MQ135PIN to the analog pin connected to the MQ135 sensor
        - Initialize sensorValue to 0
    2. Setup:
        - Initialize the serial communication for debugging with a baud rate of 9600
    3. Loop:
        Repeat the following steps indefinitely:
            - Read the analog value from the MQ135 sensor using the analogRead() function and store it in sensorValue
            - Print "Sensor Value: " followed by the value of sensorValue to the serial monitor using the Serial.print() and Serial.println() functions
            - Delay for a certain period (e.g., 1000 milliseconds) using the delay() function before taking the next reading
        loop
End of Algorithm
```

In this algorithm, first declare the necessary variables and assign the analog pin connected to the MQ135 sensor to mq135Pin. Inside the *setup*() function, initialize the serial communication for debugging purposes. Then, in the *loop*() function, continuously read the analog value from the MQ135 sensor using the *analogRead*() function and store it in the sensor Value variable. Then print the sensor value to the serial monitor using *Serial.print*() and *Serial.println*() functions for debugging and analysis.

In the context of connecting an Arduino Uno to an MQ135 sensor, the following pin connections are used in the work:

- **GND Pin:** Connect the GND (Ground) pin of the MQ135 sensor to the GND pin of the Arduino Uno. This establishes the common ground reference between the sensor and the Arduino.
- **5V Pin:** Connect the 5V pin of the MQ135 sensor to the 5V pin of the Arduino Uno. This provides the necessary power supply to the sensor.
- **A0 Pin:** Connect the A0 (Analog Input) pin of the MQ135 sensor to any available analog input pin on the Arduino Uno, such as A0, A1, A2, etc. This allows the Arduino to read the analog voltage output from the sensor, which corresponds to the air quality measurement as illustrated in Figure (3.7).

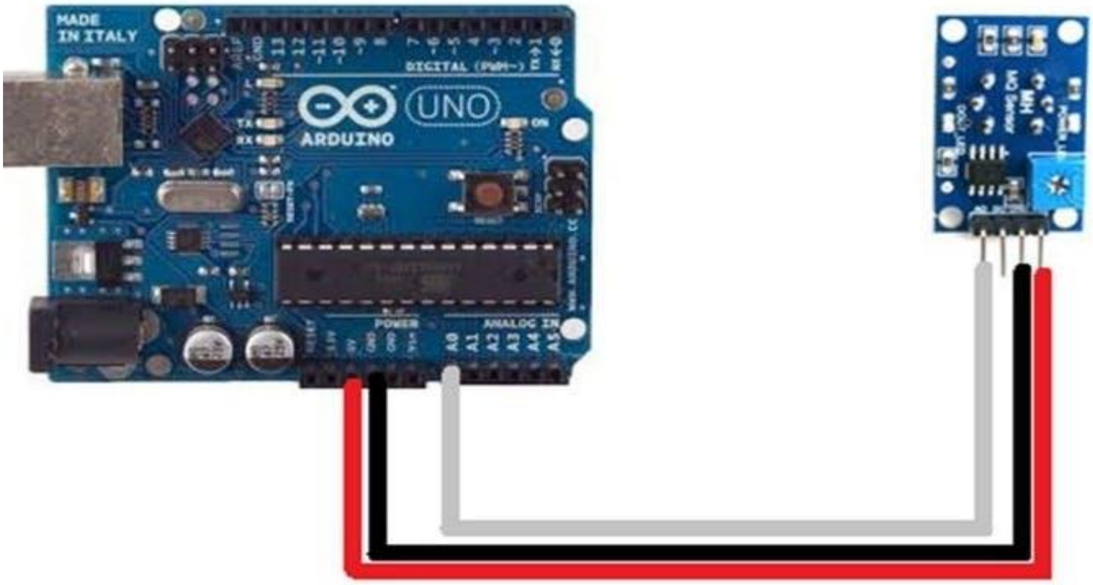


Figure 3.6: The connection between arduino and MQ135 sensor

By establishing these pin connections between the Arduino Uno and the MQ135 sensor, the Arduino enable to read the analog output from the sensor and perform further processing or analysis using the connected IoT system or the MLP model.

3.7.2 The implementation Steps

To implement the integration of IoT and MLP for air quality prediction, the following steps can be followed:

1. **Hardware Setup:** Set up the necessary hardware components for the IoT system, including the sensor nodes (MQ135), Arduino Uno, and the main controller. Connect the sensor nodes to the Arduino Uno, ensuring proper wiring and connections.
2. **Sensor Data Collection:** Develop a program or firmware for the Arduino Uno to collect air quality data from the MQ135 sensors. Read sensor values periodically and store them in variables.

3. **Data Transmission:** Establish the communication network between the sensor nodes and the central Arduino using appropriate wired or wireless connections. Configure the network settings and ensure proper connectivity.
4. **Data Transfer to the PC/Server:** Implement the mechanism to transfer the collected sensor data from the central hub/gateway to the PC or server. This can be achieved using libraries such as PySerial for serial communication.
5. **Data Processing and MLP Integration:** On the PC/server side, develop the necessary software infrastructure to receive the transmitted sensor data. Preprocess the data if required, such as normalization or feature scaling. Then, integrate the trained MLP model into the system. This involves loading the model weights and architecture and providing the necessary input data for prediction.
6. **Air Quality Prediction:** Apply the integrated MLP model to the received sensor data to predict air quality parameters, such as pollutant levels or air quality indexes. Utilize the MLP model to process the input data and generate predictions based on learned patterns and relationships.
7. **Display and Utilization:** Develop an application or user interface that retrieves the predicted air quality data from the server/PC and displays it to users. This can include visualizations, real-time monitoring, alerts, and other relevant information to enable users to make informed decisions or take appropriate actions regarding air quality.
8. **Testing and Evaluation:** Perform thorough testing and evaluation of the integrated IoT and MLP system. Validate the accuracy and performance of the air quality predictions against ground truth data or established

benchmarks. Make any necessary adjustments or optimizations to improve the system's reliability and prediction capabilities.

Algorithm (3.5) is explaining the implementation steps, the IoT and MLP integration can be realized, enabling real-time air quality monitoring and prediction based on the collected sensor data. It is important to ensure proper configuration, data processing, and validation to achieve accurate and reliable results.

Algorithm (3.5): Overall process of integration of MLP with IoT

```
Inputs:
        - IoT sensor data
Outputs:
        - Predicted air quality values
Algorithm Steps:
    1. Read the IoT sensor data:
        - Establish a connection with the IoT sensor
        - Connection build over PySerial with Continuously read the sensor data and store it
          in a variable (e.g., sensorData) until a stopping condition is met.
    2. Predict air quality using the MLP model:
        - Preprocess the sensor data by normalizing it using the same scaling factors as used
          for the training dataset
        - Pass the preprocessed sensor data (X_sensor) to the trained MLP model's "predict"
          method to obtain air quality predictions (y_pred)
    3. Output the predicted air quality values:
        - Print or store the predicted air quality values (y_pred) for further analysis or
          visualization
End of Algorithm
```

The algorithm provides an overview of the steps involved in integrating MLP with IoT for air quality prediction. The specific implementation details may vary depending on the programming language, libraries, and frameworks used.

Chapter Four

The Results Discussion and Analysis

CHAPTER FOUR
The Result Discussion and Analysis

4.1 Overview

This chapter demonstrates the results of the proposal of prediction and monitoring methods for air quality. The chapter also discusses the results proposed system and explains the methodology presented in chapter three.

4.2 The Hardware and Software Requirements

The proposed system is implemented in an IoT environment. As consisted of microcontroller, an MQ135 sensor, and a main controller (pc). The main platform for testing our detection system is a Windows 10 machine with 6th Gen Intel(R) Core (TM) i7 @ 2.40GHz processor and 16 GB RAM.

The Arduino Uno serves as the main component of the IoT setup, providing a user-friendly platform for creating interactive projects. The MQ135 sensor is specifically designed to detect and measure the concentration of various gases in the surrounding environment. It operates based on changes in resistance when exposed to target gases as shown in Table (4.1).

Table 4-1: IoT component

Device	size	voltage
Arduino uno	68.6*53.4	5V
MQ135 sensor	40.0mm * 21.0mm	2.5V ~ 5.0V

By combining the Arduino Uno microcontroller board with the MQ135 sensor, the IoT environment enables the development of projects that involve gas detection and air quality monitoring.

4.3 The Datasets

As mentioned in Chapter 3, the CPCB dataset will be used for building the MLP model and testing it. India AQ (Air quality) station day dataset will also be used to validate the proposed model.

In Figure (4.1), illustrates the relative proportions of each air quality category within the dataset. It provides insights into the frequency and distribution of different pollution levels, highlighting areas of concern and identifying patterns and trends in air quality.

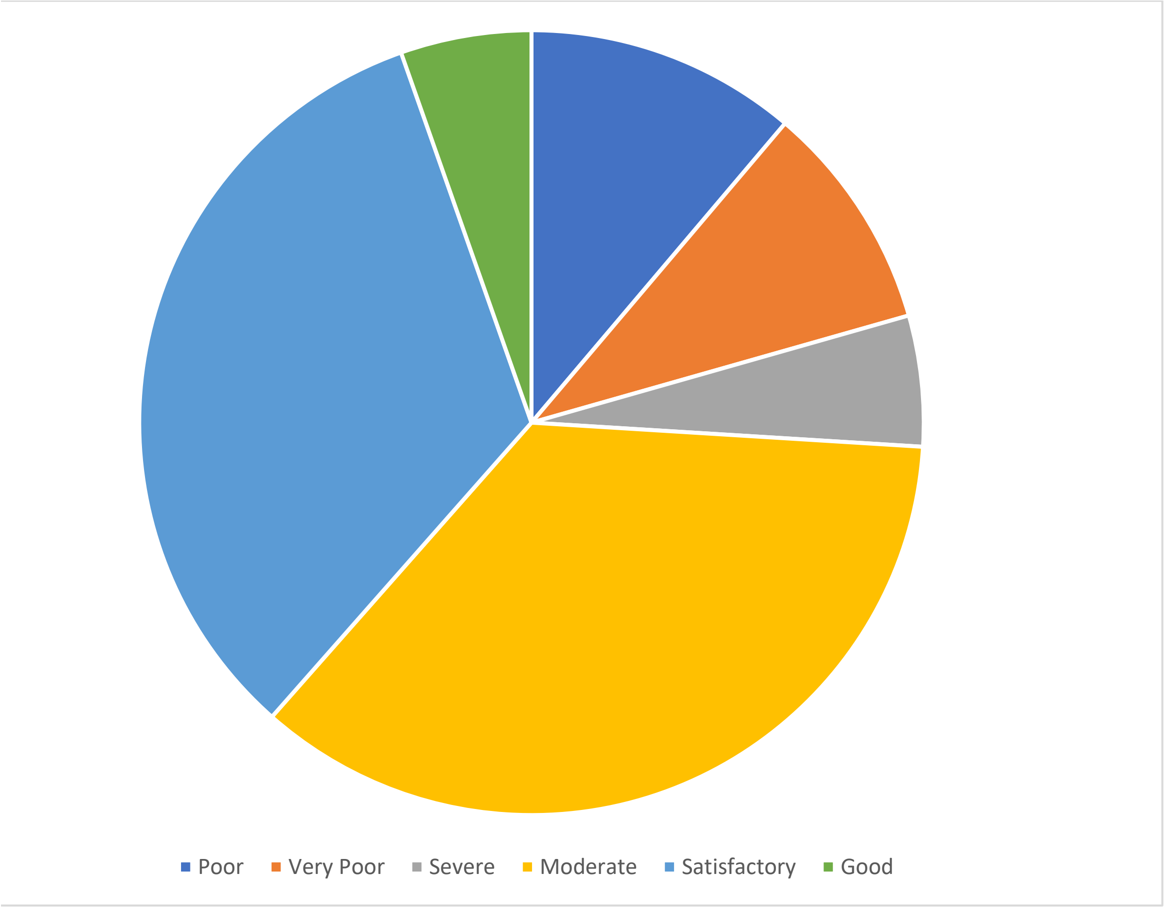


Figure 4.1 The classification of the CPCB dataset

Understanding the classification of the CPCB dataset is crucial for interpreting and analyzing the air quality data accurately. It serves as a reference for evaluating the performance of prediction models, assessing the effectiveness of pollution control measures, and guiding decision-making processes related to environmental management and public health protection.

By utilizing the classification information provided in Figure (4.1), researchers and stakeholders can comprehensively understand the air quality levels represented in the CPCB dataset, facilitating further analysis and interpretation of the data.

4.3.1 Data Cleaning Result

Table (4.2) presents the number of missing values in the CPCB dataset for various air quality features. It is important to identify and handle missing values appropriately to ensure the accuracy and reliability of data analysis and modeling.

The table provides information on the percentage of missing values for each feature in the dataset. For example, the PM2.5 feature has a missing value of 4598. Similarly, other features such as PM10, CO2, NO, SO2, NO2, O3, NOX, and Benzene also have varying of missing values.

Dealing with missing values is a crucial step in the data preprocessing phase. Depending on the extent and pattern of missing values, various techniques can be applied, such as imputation methods or deletion of rows or columns with missing values.

Table 4.2: The missing values in the dataset

Feature	NO. of missing value	Feature	NO. of missing value
PM2.5	4598	**Xylene**	18109
PM10	11140	**CO**	2059
NO	3582	**SO2**	3854
NO2	3585	**O3**	4022
NOX	4185	**Benzene**	5623
NH3	10328	**Toluene**	8041

The processes of cleaning data by filling in missing values which 12 columns containing empty cells (PM2.5, PM10, CO, NOx, NO, NO2, O3, benzene, toluene, xylene), dealing with this problem by full it using the average method.

It is important to address missing values appropriately to avoid biases or inaccurate conclusions in subsequent analyses. By identifying the missing values in the CPCB dataset and applying suitable strategies to handle them, ensuring the integrity and quality of the data, and improving the reliability of results and predictions related to air quality.

Figure (4.2) displays the features' representative and the corresponding number of missing values for each feature in the right column. As it seems clear, the feature " Xylene " has a more missing value.

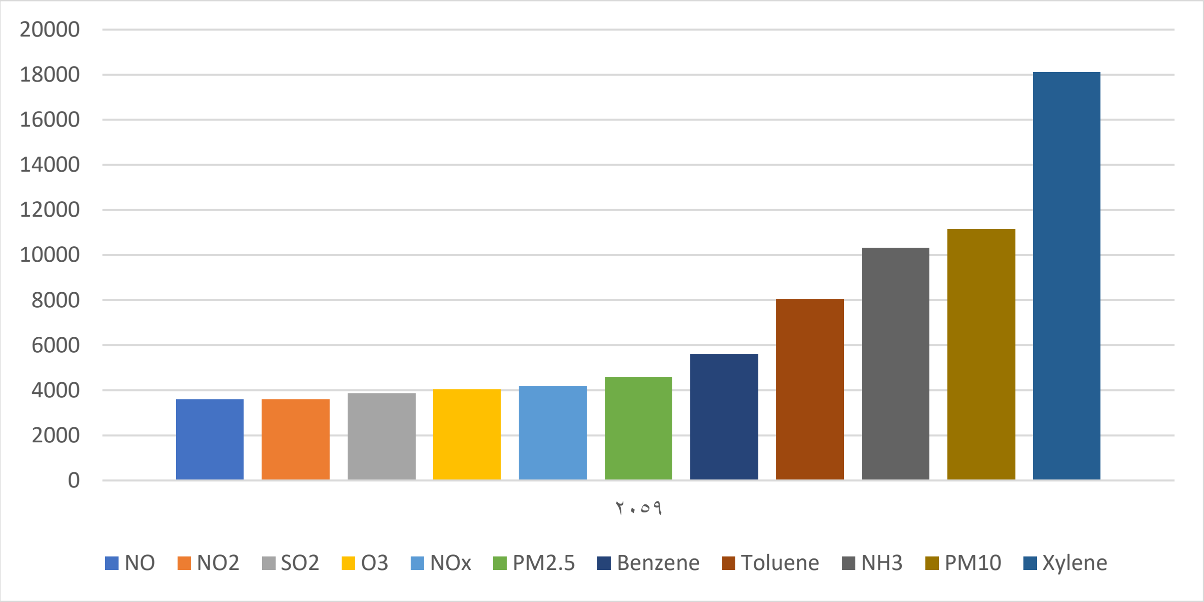
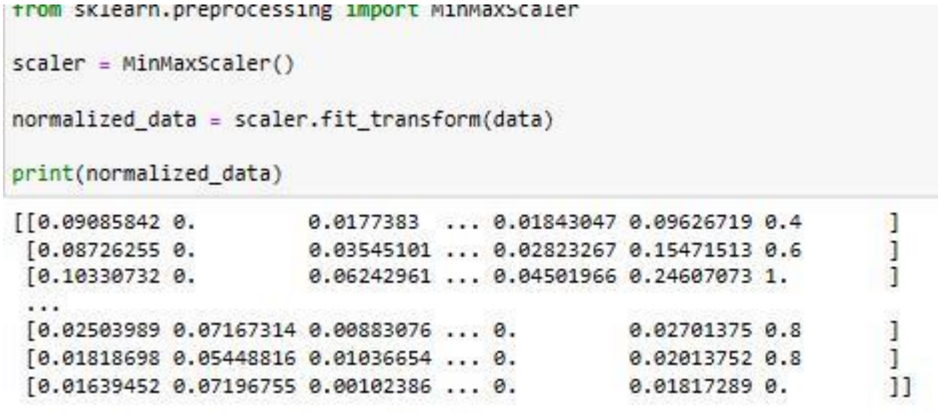


Figure 4.2: The missing values of each feature

4.3.2 Min-Max Normalized Results

This proposed system used min-max normalized with 0 as a new minimum value and 1 as a new maximum value, the range of results is between 0 and 1. Figure 4.2 show a sample from the normalized dataset.



```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

normalized_data = scaler.fit_transform(data)

print(normalized_data)
```

```
[[0.09085842 0.         0.0177383  ... 0.01843047 0.09626719 0.4       ]
 [0.08726255 0.         0.03545101 ... 0.02823267 0.15471513 0.6       ]
 [0.10330732 0.         0.06242961 ... 0.04501966 0.24607073 1.        ]
 ...
 [0.02503989 0.07167314 0.00883076 ... 0.         0.02701375 0.8       ]
 [0.01818698 0.05448816 0.01036654 ... 0.         0.02013752 0.8       ]
 [0.01639452 0.07196755 0.00102386 ... 0.         0.01817289 0.        ]]
```

Figure 4.3: Normalazation Result of CCPCB dataset

The normalized processing used with columns ('NH3',' PM2.5', 'PM10', 'NOx', 'NO2', 'NO', 'O3', 'CO', 'SO2', ' Benzene', 'xylene', 'toluene') that representation the **numerical data**.

4.3.3 The Feature Selection Method

Feature selection is essential for removing unnecessary features. This section presents one of the feature selection mechanisms as will be demonstrated in the section 3. Due to this, a term (GROUPS) will be used to describe each outcome referred to as multiple features of the previous mechanism of feature selection.

A. Pearson Correlation Based on Feature Selection

To select significant features, the correlation analysis of the AQI feature has been exercised with features of other pollutants. Figure (4.4) shown below clearly reveals that pollutants PM$_{10}$, PM$_{2.5}$, CO, NO$_2$, SO$_2$, NOX, and NO are generally responsible for the AQI to attain higher values.

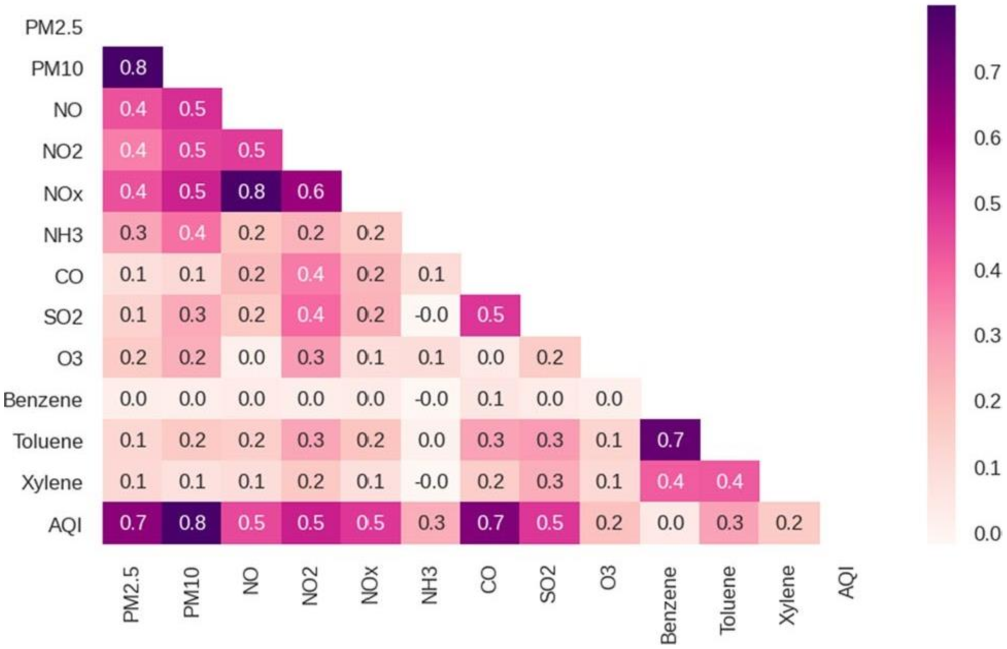


Figure 4.4: Correlation heatmap

Correlation analysis depicted in Figure (4.4) illustrates the relationship between the Air Quality Index (AQI) and various pollutants, including PM10, PM2.5, CO, NO2, SO2, NOX, and NO. The analysis aims to identify the significant features that contribute to higher AQI values.

The figure demonstrates that these pollutants exhibit positive correlations with the AQI. When the concentrations of PM10, PM2.5, CO, NO2, SO2, NOX, and NO increase, the AQI tends to reach higher values. This correlation analysis highlights the importance of these pollutants in influencing the overall air quality and its subsequent impact on the AQI.

Table (4.3) presents the features along with their corresponding Pearson correlation values. The higher the correlation value, the stronger the linear relationship between the feature and the target variable

Based on the correlation ranks, certain features exhibit stronger correlations with the air pollution level. For example, Pm10 has a high correlation of 0.81, indicating a strong positive linear relationship.

Table 4-3: The Pearson correlation ranks of features

No	Name of feature	Pearson Correlation	No	Name of feature	Pearson Correlation
1	PM10	0.81	7	NO	0.45
2	CO	0.68	8	O3	0.27
3	PM2.5	0.65	9	NH3	0.25
4	NO2	0.53	10	Toluene	0.19
5	SO2	0.52	11	Xylene	0.16
6	NOx	0.48	12	Benzene	0.04

B. Multiple Tests based Features Groups

In this study, multiple tests based on feature groups can be combined with Pearson correlation-based feature selection to enhance the effectiveness of the feature selection process. Pearson correlation is a statistical measure that quantifies the linear relationship between two variables.

Features were selected based on Pearson correlation values greater than 0.4 after then implementing multiple tests to obtain the best features group. By applying this criterion, features with the best performance were chosen to build and test the proposed model.

Table (4.4) present the multiple tests based on feature groups involving conducting experiments using different combinations of features from the feature groups. This approach allows of how different feature sets impact the performance or results of an air quality monitoring system.

Table 4-4 : The feature groups

No	Number of Features Group	No. Features	List of Features
1	Feature group 1	3	PM10, CO, PM2.5
2	Feature group 2	6	PM10, CO, PM2.5, NO2, SO2, NOx
3	Feature group 3	7	PM10, CO, PM2.5, NO2, SO2, NOx, NO
4	Feature group 4	9	PM10, CO, PM2.5, NO2, SO2, NOx, NO, O3, NH3
5	Feature group 5	10	PM10, CO, PM2.5, NO2, SO2, NOx, NO, O3, NH3, Toluene
6	Feature group 6	12	PM10, CO, PM2.5, NO2, SO2, NOx, NO, O3, NH3, Toluene, Xylene, benzene

The four major performance measures criteria were used to evaluate the performance of the MLP model which are Accuracy, Recall, Precision, and F1 scores.

The tables below show how well the MLP model performs in terms of group concern. The performance of the MLP model is shown in Table (4.5), the 70:30 ratio mode of split data was used to train and test the algorithm.

Table 4-5: The performance of classifiers on feature groups

Feature Group	Accuracy%	Recall%	Precision%	F1 score%
Feature group 1	99.026	0.99	0.99	0.99
Feature group 2	99.011	0.99	0.98	0.98
Feature group 3	99.115	0.99	0.99	0.99
Feature group 4	97.331	0.97	0.97	0.97
Feature group 5	97.261	0.97	0.97	0.97
Feature group 6	98.441	0.98	0.98	0.98

Figures (4.5–4.8) provide a comparison through accuracy, precision, recall, and F score.

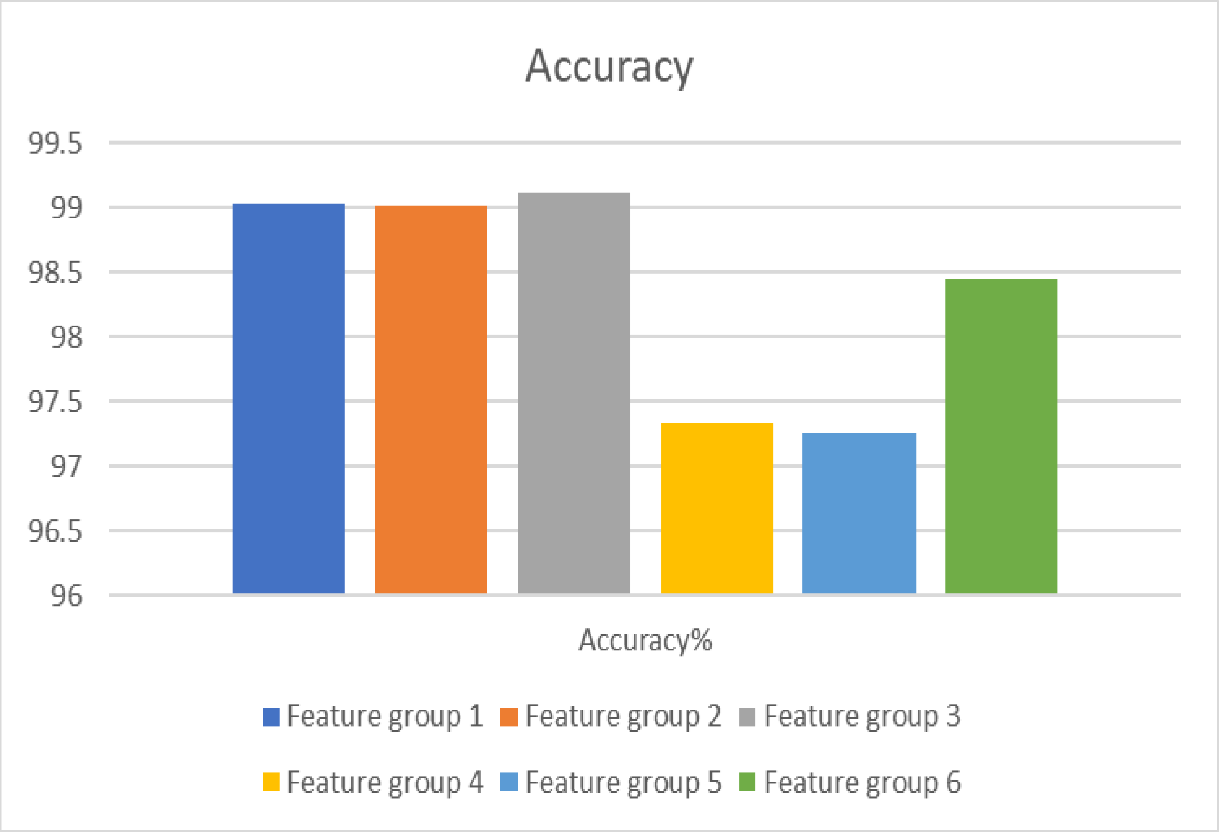


Figure 4.5: Classifier's accuracy for six Feature Groups

Figure (4.5) depicts the evaluation of the MLP model for the six-feature groups. The figure presents the evaluation results of an MLP (Multilayer Perceptron) model for six different feature groups. The accuracy of the MLP classifier is reported for each feature group, indicating the performance of the model in classifying air quality based on the selected features.

The results indicate that Feature Group 3, which includes all seven features, achieves the highest overall accuracy of 99.115%. This suggests that the combination of features in Feature Group 3 provides the most discriminative information for accurately classifying air quality levels. However, it's worth noting that Feature Group 1 and Feature Group 2 also achieve high accuracies, close to the performance of Feature Group 3.

These findings emphasize the importance of feature selection and highlight the effectiveness of certain feature combinations in improving the accuracy of the MLP classifier for air quality classification. Feature Group 3, with all seven features, is identified as the best-performing feature group based on the reported accuracy results.

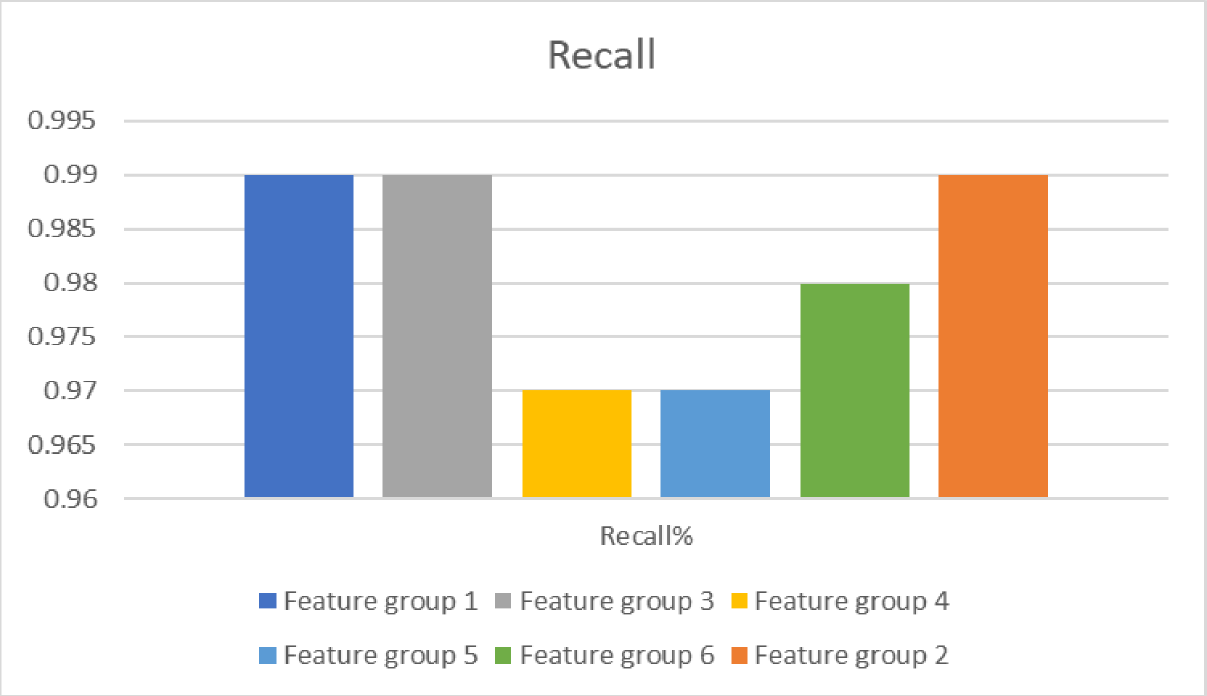


Figure 4.6: Classifier Recall for Six Feature Groups

Figure (4.6) depicts model recall for the six feature groupings. The recall of the MLP model in feature groups 1, 2, and 3 remains the same.

Figure (4.7) depicts the precision of the MLP model for the six feature groups. With feature groups 1 and 3, the MLP model acquires a higher precision. The average precision for other features group classifiers for feature group 2, feature group 4, feature group 5, and feature group 6 are 0.98%, 0.97%, 0.97%, and 0.98% respectively. Feature group 1,3 has the greatest overall classifier F-measure score, which is 0.99%.

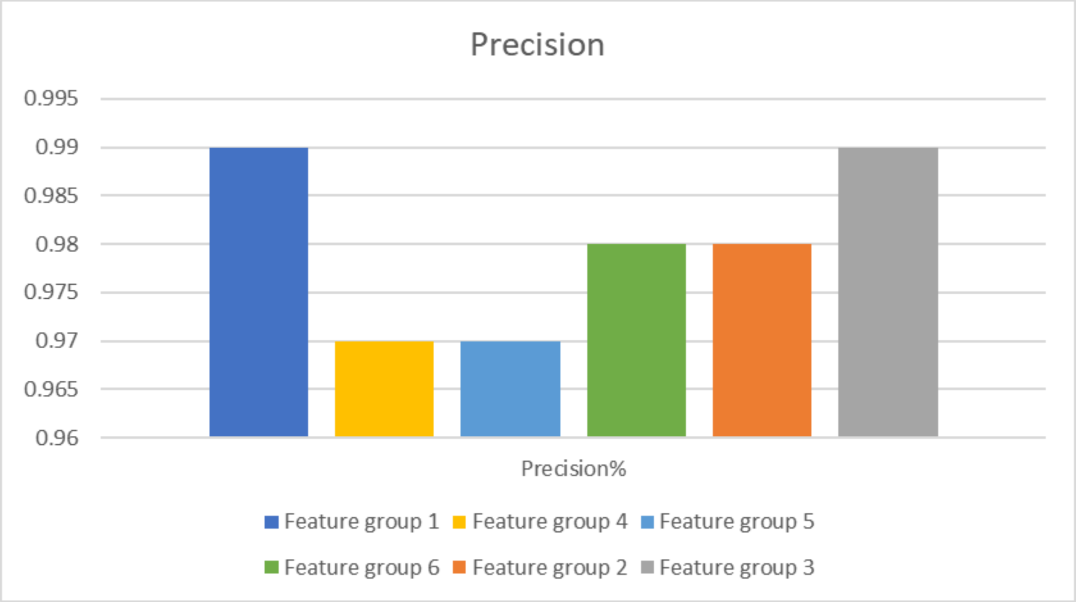


Figure 4.7: Precision of Classifiers for six feature Groups

Figure (4.8) depicts the F1 scores of the MLP model for the six feature groups. With feature groups 1 and 3, the MLP model acquires a higher F1 score. Feature groups have an impact on these algorithms. The average F-measure for other features group classifiers for feature group 2, feature group 4, feature group 5, and feature Group 6 is 0.98%, 0.97%, 0.97%, and 0.98% respectively. Feature group 1,3 has the greatest overall classifier F-measure score, which is 0.99%.

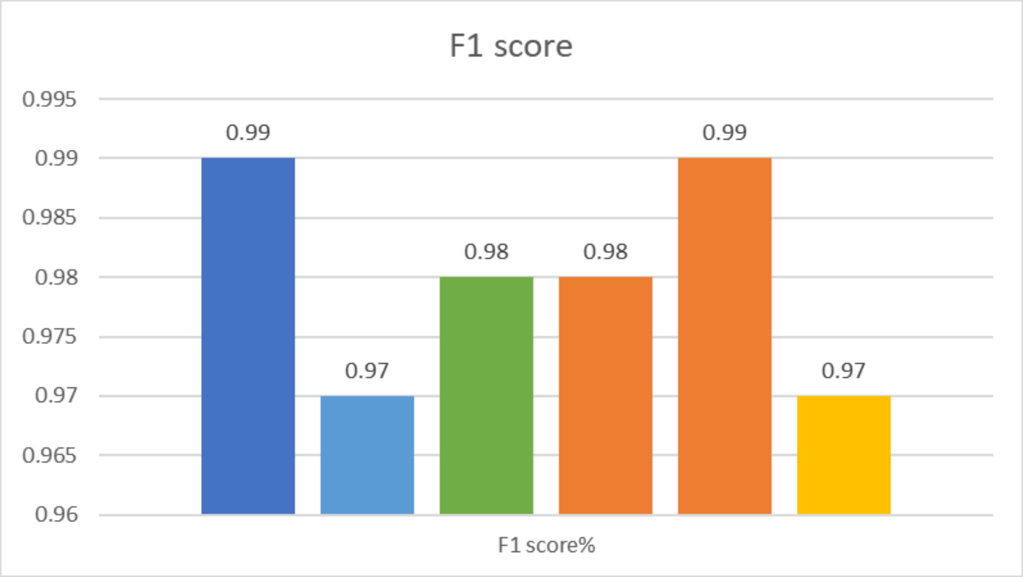


Figure 4.8: The F1-score for six different feature groups

After examining the four performance measures obtained by the MLP model for the six feature groups, it is discovered that feature group 3 with all features achieves the highest overall classifier performance. According to the experimental results, the CPCB dataset characteristics chosen and employed in this work are capable of identifying air pollution with an average accuracy of 99%, a recall rate of 0.99% on average, the average accuracy was 0.99%, and the average F1 score was 0.99%.

4.4 Evaluation metrics results

The accuracy was 99.115, and the result of the confusion matrix for testing data is shown in Table 4.6.

Table 4-6: Confusion matrix for CPCB dataset

		Predicted					
		Good	**moderate**	**poor**	**Very poor**	**satisfactory**	**sever**
Actual	**Good**	724	0	0	0	2	0
	moderate	0	3874	6	0	0	0
	poor	0	3	1064	0	0	0
	Very poor	0	0	9	720	0	8
	satisfactory	0	0	0	0	4153	0
	sever	0	0	0	1	0	326

Figure (4.9) display the result of the accuracy, precision, recall, and F1 score for testing data

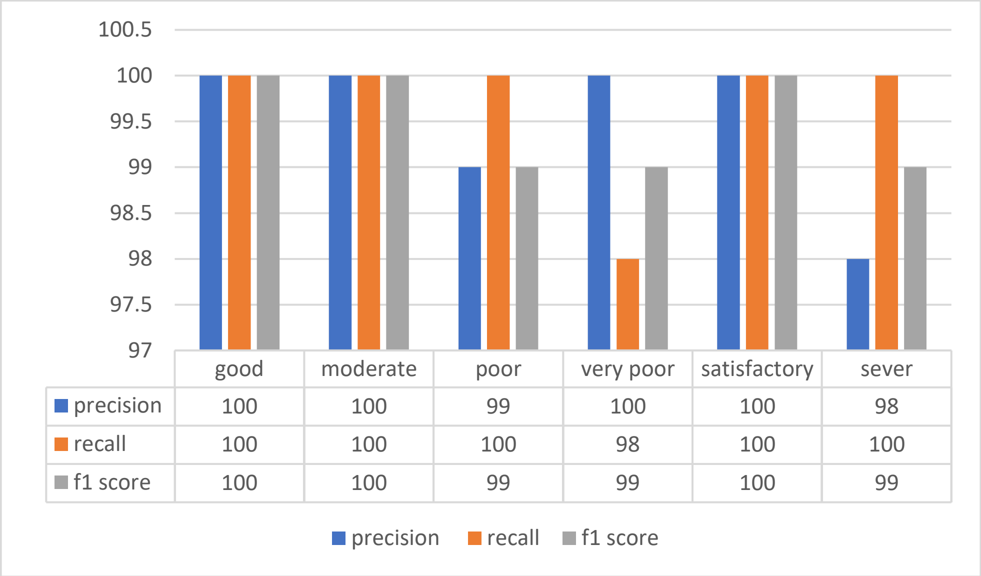


Figure 4.9: Evaluation result for testing data

In the India air quality station day dataset, the accuracy was 97.47%, and the result of the confusion matrix for testing data is shown in Table 4.7

Table 4-7:Confusion matrix of India air quality station dataset

		Predicted					
		Good	**moderate**	**poor**	**Very poor**	**satisfactory**	**sever**
Actual	**Good**	8866	0	0	0	0	0
	moderate	227	3163	31	0	0	0
	poor	0	6	3519	0	0	2
	Very poor	157	0	9	6925	0	8
	satisfactory	0	0	0	203	1417	0
	sever	0	0	80	0	0	1484

The model achieved high accuracy indicating its effectiveness in predicting air pollution levels. The confusion matrix demonstrates that the model performs well across different pollution levels, with a majority of instances being correctly classified in their respective categories.

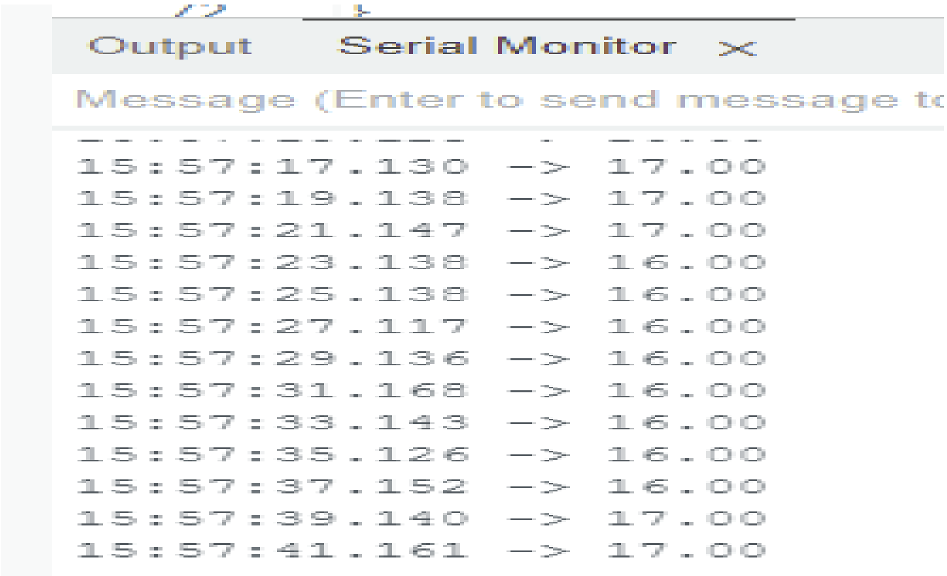
4.5 Implement MLP in Proposed IoT Environment

Implementing MLP in a proposed IoT environment based on air quality aims to leverage the power of artificial neural networks to monitor and analyze air pollution levels. With IoT devices equipped with sensors deployed in various locations, MLPs provide a means to process the collected air quality data. In this scenario, the problem is monitoring and predicting air pollution levels in the IoT environment. The fellow steps of implementing the proposed system in IoT.

4.5.1 Read Data by Sensor

To read data from the MQ135 sensor and display numeric values using Arduino IDE, follow these steps:

1. Connect the MQ135 sensor to the Arduino board.
2. Make sure to connect the necessary pins (e.g., VCC, GND, and analog output) based on the sensor's specifications.
3. Open the Arduino IDE and create a new sketch and put the code that read the data of the sensor when senses the pullulation. The monitor side of the Arduino IDE would display the values outputted by the Arduino board. Figure 4.10 represents the values of the sensor displayed in the Arduino IDE monitor.

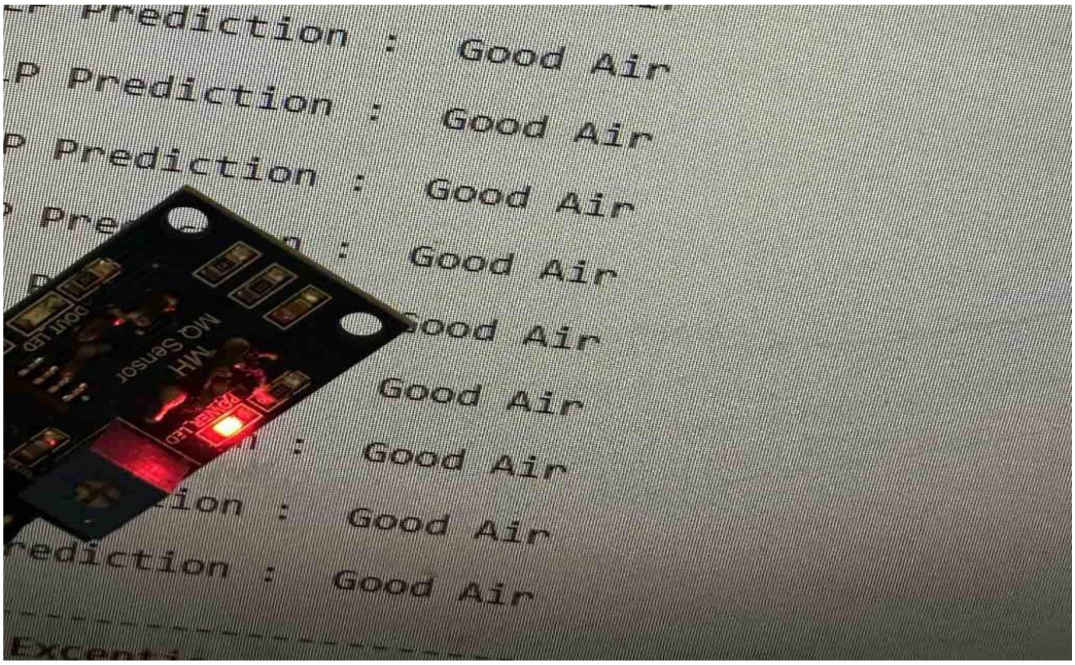


```
15:57:17.130 -> 17.00
15:57:19.138 -> 17.00
15:57:21.147 -> 17.00
15:57:23.138 -> 16.00
15:57:25.138 -> 16.00
15:57:27.117 -> 16.00
15:57:29.136 -> 16.00
15:57:31.168 -> 16.00
15:57:33.143 -> 16.00
15:57:35.126 -> 16.00
15:57:37.152 -> 16.00
15:57:39.140 -> 17.00
15:57:41.161 -> 17.00
```

Figure 4.10: The values sensor in Arduino IDE monitor side

The values in the Arduino IDE monitor side might include numeric readings of the sensor values at different time intervals. These values could be continuously updated and displayed in real time as the sensor readings change. When there is no pollution present, the readings from the MQ135 sensor are expected to indicate good air quality. The MQ135 sensor primarily measures the concentration of gases, and in the absence of pollutants or harmful gases, the sensor readings should reflect a clean and healthy environment.

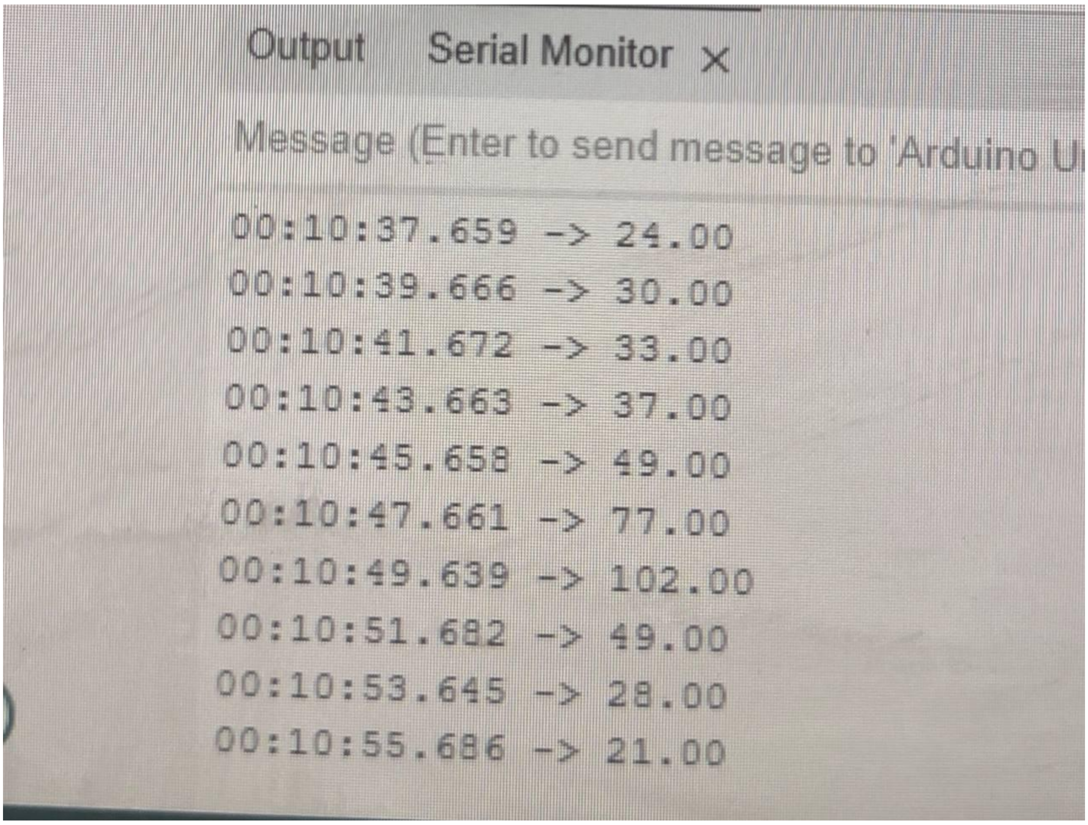
Typically, the MQ135 sensor operates on the principle of changes in resistance due to the presence of gases. When no pollution or harmful gases are present, the resistance measured by the sensor remains relatively stable. This stability in readings indicates that the air quality is within acceptable limits. Figure (4-11) shows the prediction without pollution.



```
r Prediction :  Good Air
.P Prediction :  Good Air
P Prediction :  Good Air
 Pre             :  Good Air
                    ood Air
                    Good Air
             :  Good Air
    ion :  Good Air
rediction :  Good Air
-------------------------
Except
```

Figure 4.11:Result with no pollution

When expose a gas to the MQ135 sensor, the sensor will detect and measure the concentration of that particular gas in the air. Figure (4-12) show the result value when expose a NOX gas to the sensor



```
00:10:37.659 -> 24.00
00:10:39.666 -> 30.00
00:10:41.672 -> 33.00
00:10:43.663 -> 37.00
00:10:45.658 -> 49.00
00:10:47.661 -> 77.00
00:10:49.639 -> 102.00
00:10:51.682 -> 49.00
00:10:53.645 -> 28.00
00:10:55.686 -> 21.00
```

Figure 4.12: Result of Mq135 sensor with NOx Gas

4.5.2 The PySerial Setup

To setup PySerial and take data from a sensor, follow these steps:

1. Import the `serial` module from the `PySerial` library in the Python script.
2. Set up the serial connection by creating an instance of the `Serial` class. Specify the port to match the sensor's configuration. The port refers to the communication port through which the sensor is connected (e.g., 'COM1' for Windows).
3. Use the appropriate functions provided by PySerial to read data from the sensor. One commonly used function is `readline()`, which reads a line of data from the serial buffer.
4. Display or utilize the sensor data, print the data to the console, store it in a file, and send it to a database.

When implementing these steps, it's essential to ensure that the sensor is actively sending data through the serial port and that the port settings match the sensor's configuration.

4.5.3 The Data Transfer to Main Controller (PC)

The **analogRead()** function reads the analog value from the MQ135 sensor connected to the MQ135_PIN (A0 in this case). The sensor value is then sent to the computer via serial communication using **Serial. println().**

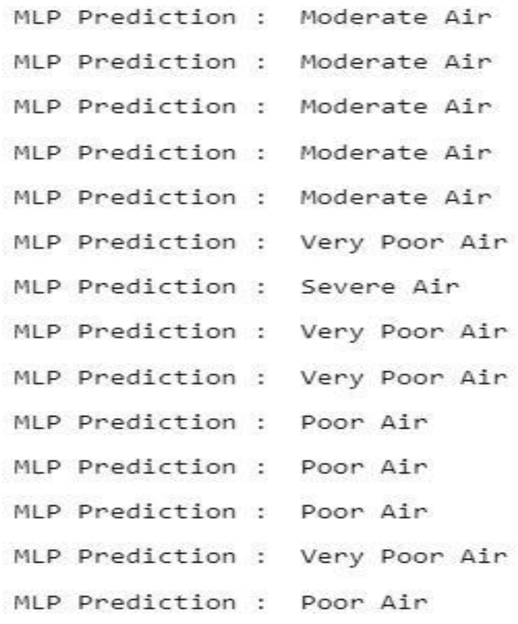
1. Upload the code to the Arduino board by clicking on the "Upload" button in the Arduino IDE.
2. Open the Serial Monitor in the Arduino IDE.
3. Once the code is uploaded and the Serial Monitor is open, should start seeing the sensor readings displayed in the Serial Monitor. The readings from the MQ135 sensor are sent to the computer through the USB connection, to view and analyze the data on the computer.

By following these steps, successfully pass data from the MQ135 sensor to the computer using the Arduino board and the serial communication feature.

4.5.4 Air Quality Prediction in Real-Time

In the proposed IoT environment, the MQ135 sensor and Arduino play essential roles in capturing air quality data and integrating it into the overall system.

The Arduino reads the digital values from the MQ135 sensor at regular intervals, enabling real-time monitoring of air quality. The Arduino can interface with a user interface, such as a computer (see Figure 4-13), to display the air quality data.



```
MLP Prediction :  Moderate Air
MLP Prediction :  Moderate Air
MLP Prediction :  Moderate Air
MLP Prediction :  Moderate Air
MLP Prediction :  Moderate Air
MLP Prediction :  Very Poor Air
MLP Prediction :  Severe Air
MLP Prediction :  Very Poor Air
MLP Prediction :  Very Poor Air
MLP Prediction :  Poor Air
MLP Prediction :  Poor Air
MLP Prediction :  Poor Air
MLP Prediction :  Very Poor Air
MLP Prediction :  Poor Air
```

Figure 4.13 : Predction values in realtime

4.6 The comparisons of the proposed system against related works

When evaluating the proposed system and comparing its results with the other related work results, it was found that the proposed system achieved better results and higher accuracy in both data sets (CPCB and India air quality station dataset). The reason for this is due to the efficiency of the methodology as it was built step by step in a deliberate and accurate manner. , the results of first dataset is higher than the result of India air quality station dataset In addition to taking advantage of the features selection when building the model, also using three hidden layer of the deep learning model and the number of nodes achieve better result than the author [10] that used only one hidden layer .

The Relu activation function is consider the best activation function that help to accurate model than sigmoid function.

Table 4.8 is illustrated a comparison between proposed system work of CPCB and India air quality station dataset with multi classification and other related works.

Table 4-6: The proposed system is compared with recent works

No	Author	Dataset	Algorithm	Accuracy
1	[10]	CPCB dataset	Multilayer Perceptron	98.1%
2	[14]	CPCB dataset	SVM	97.3%
3	[83]	OFEPP	Neural Network	92.3%
4	[11]	India air quality station day	GRADIENT BOOST	95%
5	[17]	CPCB	SVM	96.92%
6	[16]	Open Data Jakarta	neural network	96.61%
7	[18]	CPCB	XGboost	91%
#	**Our proposed system**	CPCB	MLP	99.115%
		India air quality station day		97.47%

Chapter Five
The Conclusions and The Suggested Future Works

CHAPTER FIVE

The Conclusions and The Future Works

5.1 The Conclusion

This study explores the synergy between Internet of Things (IoT) devices and deep learning techniques to enhance air quality monitoring. Recognizing the critical importance of accurate air quality assessment for both human health and the environment, this research demonstrates the potential of real-time data collection and analysis through IoT integration. Utilizing deep learning, the study showcases the ability to predict air quality parameters, enabling proactive measures to mitigate risks associated with poor air quality. The impact of feature selection on prediction accuracy is examined, and the effectiveness of deep learning models in forecasting air quality parameters is illustrated through analysis of a comprehensive dataset.

1- **Accurate Air Quality Prediction**

Deep learning techniques accurately predict air quality parameters based on historical data and real-time sensor readings. These predictions empower proactive measures to address potential risks linked with poor air quality.

2- **Impact of Feature Selection**

The study reveals that feature selection enhances prediction accuracy. Choosing the right data features is crucial in refining predictive models and boosting their precision.

5.2 The Suggested Future Works

We recommend that, as part of future development:

1. **Integration of Additional Data Sources**

Combine meteorological data to air quality prediction. Meteorological data affects air quality. Focus on preprocessing, feature extraction, and interdisciplinary collaboration to effectively merge diverse data.

2. **Model Refinement and Optimization**

   Improve deep learning models by tuning hyperparameters and exploring ensemble techniques. Enhance interpretability for better insights. Collaborate between experts to align refined models with air quality intricacies.

References

[1] I. Manisalidis, E. Stavropoulou, A. Stavropoulos, and E. Bezirtzoglou, “Environmental and health impacts of air pollution: a review,” *Front. public Heal.*, vol. 8, p. 14, 2020.

[2] A. Özkara, D. Akyıl, and M. Konuk, “Pesticides, environmental pollution, and health,” in *Environmental health risk-hazardous factors to living species*, IntechOpen, 2016.

[3] A. Sadeghi-Niaraki, “Internet of Thing (IoT) review of review: Bibliometric overview since its foundation,” *Futur. Gener. Comput. Syst.*, 2023.

[4] D. Ivanova and A. Elenkov, “Intelligent System for Air Quality Monitoring Assessment using the Raspberry Pi Platform,” *2019 Int. Conf. Inf. Technol. InfoTech 2019 - Proc.*, no. September, pp. 1–4, 2019, doi: 10.1109/InfoTech.2019.8860883.

[5] Y. C. Liang, Y. Maimury, A. H. L. Chen, and J. R. C. Juarez, “Machine learning-based prediction of air quality,” *Appl. Sci.*, vol. 10, no. 24, pp. 1–17, 2020, doi: 10.3390/app10249151.

[6] S. Zhang, X. Li, Y. Li, and J. Mei, “Prediction of urban pm 2.5 concentration based on wavelet neural network,” in *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 5514–5519.

[7] X. Zhao, R. Zhang, J. L. Wu, and P. C. Chang, “A deep recurrent neural network for air quality classification,” *J. Inf. Hiding Multimed. Signal Process.*, vol. 9, no. 2, pp. 346–354, 2018.

[8] C. Srivastava, S. Singh, and A. P. Singh, “Estimation of air pollution in Delhi using machine learning techniques,” in *2018 International Conference on Computing, Power and Communication Technologies (GUCON)*, 2018, pp. 304–309.

[9] Y.-T. Tsai, Y.-R. Zeng, and Y.-S. Chang, “Air pollution forecasting using RNN with LSTM,” in *2018 IEEE 16th Intl Conf on Dependable, Autonomic and Secure Computing, 16th Intl Conf on Pervasive Intelligence and Computing, 4th Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech*, 2018, pp. 1074–1079.

[10] A. Akhtar, S. Masood, C. Gupta, and A. Masood, “Prediction and analysis of pollution levels in delhi using multilayer perceptron,” *Adv. Intell. Syst. Comput.*, vol. 542, pp. 563–572, 2008, doi:

10.1007/978-981-10-3223-3_54.

[11] S. Bali, “Indian Air Quality Prediction and,” vol. 14, no. 11, pp. 181–186, 2019.

[12] H. Maleki, A. Sorooshian, G. Goudarzi, Z. Baboli, Y. Tahmasebi Birgani, and M. Rahmati, “Air pollution prediction by using an artificial neural network model,” *Clean Technol. Environ. Policy*, vol. 21, no. 6, pp. 1341–1352, 2019, doi: 10.1007/s10098-019-01709-w.

[13] X. Wang and B. Wang, “Research on prediction of environmental aerosol and PM2.5 based on artificial neural network,” *Neural Comput. Appl.*, vol. 31, no. 12, pp. 8217–8227, 2019, doi: 10.1007/s00521-018-3861-y.

[14] U. Mahalingam, K. Elangovan, H. Dobhal, C. Valliappa, S. Shrestha, and G. Kedam, “A machine learning model for air quality prediction for smart cities,” *2019 Int. Conf. Wirel. Commun. Signal Process. Networking, WiSPNET 2019*, pp. 452–457, 2019, doi: 10.1109/WiSPNET45539.2019.9032734.

[15] M. Aljanabi, M. Shkoukani, and M. Hijjawi, “Ground-level ozone prediction using machine learning techniques: A case study in Amman, Jordan,” *Int. J. Autom. Comput.*, vol. 17, no. 5, pp. 667–677, 2020.

[16] F. Hamami and I. Fithriyah, “Classification of air pollution levels using artificial neural network,” *2020 Int. Conf. Inf. Technol. Syst. Innov. ICITSI 2020 - Proc.*, pp. 217–220, 2020, doi: 10.1109/ICITSI50517.2020.9264910.

[17] S. Ketu and P. K. Mishra, “Scalable kernel-based SVM classification algorithm on imbalance air quality data for proficient healthcare,” *Complex Intell. Syst.*, vol. 7, no. 5, pp. 2597–2615, 2021, doi: 10.1007/s40747-021-00435-5.

[18] K. Kumar and B. P. Pande, “Air pollution prediction with machine learning: a case study of Indian cities,” *Int. J. Environ. Sci. Technol.*, no. 0123456789, 2022, doi: 10.1007/s13762-022-04241-5.

[19] M. Mannan and S. G. Al-Ghamdi, “Indoor air quality in buildings: a comprehensive review on the factors influencing air pollution in residential and commercial structure,” *Int. J. Environ. Res. Public Health*, vol. 18, no. 6, p. 3276, 2021.

[20] H. Xu *et al.*, “Environmental pollution, a hidden culprit for health issues,” *Eco-Environment Heal.*, vol. 1, no. 1, pp. 31–45, 2022.

[21] J. Spiteri and P. von Brockdorff, “Transboundary air pollution and respiratory disease mortality: Evidence from European countries,” *J. Econ. Stud.*, vol. 48, no. 7, pp. 1371–1387, 2021.

[22] J. Meng *et al.*, “The slowdown in global air-pollutant emission growth and driving factors,” *One Earth*, vol. 1, no. 1, pp. 138–148, 2019.

[23] A. M. Graham *et al.*, “Impact of weather types on UK ambient particulate matter concentrations,” *Atmos. Environ. X*, vol. 5, p. 100061, 2020.

[24] R. Sivarethinamohan, S. Sujatha, S. Priya, A. Gafoor, and Z. Rahman, “Impact of air pollution in health and socio-economic aspects: review on future approach,” *Mater. Today Proc.*, vol. 37, pp. 2725–2729, 2021.

[25] Y. Feng, M. Ning, Y. Lei, Y. Sun, W. Liu, and J. Wang, “Defending blue sky in China: Effectiveness of the ‘Air Pollution Prevention and Control Action Plan’ on air quality improvements from 2013 to 2017,” *J. Environ. Manage.*, vol. 252, p. 109603, 2019.

[26] P. Saxena, S. Sonwani, P. Saxena, and S. Sonwani, “Primary criteria air pollutants: environmental health effects,” *Criteria air Pollut. their impact Environ. Heal.*, pp. 49–82, 2019.

[27] A. Eldering *et al.*, “The Orbiting Carbon Observatory-2: First 18 months of science data products,” *Atmos. Meas. Tech.*, vol. 10, no. 2, pp. 549–563, 2017.

[28] T. Madan, S. Sagar, and D. Virmani, “Air quality prediction using machine learning algorithms–a review,” in *2020 2nd International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, 2020, pp. 140–145.

[29] M. Fadel *et al.*, “Identification and apportionment of local and long-range sources of PM2. 5 in two East-Mediterranean sites,” *Atmos. Pollut. Res.*, vol. 14, no. 1, p. 101622, 2023.

[30] P. Kumar, A. B. Singh, T. Arora, S. Singh, and R. Singh, “Critical review on emerging health effects associated with the indoor air quality and its sustainable management,” *Sci. Total Environ.*, vol. 872, p. 162163, 2023.

[31] N. R. Kapoor, A. Kumar, A. Kumar, A. Kumar, and H. C. Arora, "Prediction of Indoor Air Quality Using Artificial Intelligence," *Mach. Intell. Big Data Anal. IoT Image Process. Pract. Appl.*, pp. 447–469, 2023.

[32] B. C. Singer *et al.*, "Indoor secondary pollutants from cleaning product and air freshener use in the presence of ozone," *Atmos. Environ.*, vol. 40, no. 35, pp. 6696–6710, 2006.

[33] T. Bhattacharya and J. Garg, "Causes and Prevention of air pollution in modern age," *J. AMITY Bus. Sch. AMITY Univ. NOIDA, INDIA*, p. 1, 2018.

[34] J. Lin, W. Ho, X. Qin, C. Leung, V. K. Au, and S. Lee, "Metal–organic frameworks for NOx adsorption and their applications in separation, sensing, catalysis, and biology," *Small*, vol. 18, no. 13, p. 2105484, 2022.

[35] W. B. GRANT, "Acid Rain And Deposition-Hydrology-Climate Policy Watcher (2023)".

[36] P. O. Ukaogo, U. Ewuzie, and C. V Onwuka, "Environmental pollution: causes, effects, and the remedies," in *Microorganisms for sustainable environment and health*, Elsevier, 2020, pp. 419–429.

[37] T. A. Alghamdi and N. Javaid, "A survey of preprocessing methods used for analysis of big data originated from smart grids," *IEEE Access*, vol. 10, pp. 29149–29171, 2022.

[38] T. Tanaka, I. Nambu, Y. Maruyama, and Y. Wada, "Sliding-Window Normalization to Improve the Performance of Machine-Learning Models for Real-Time Motion Prediction Using Electromyography," *Sensors*, vol. 22, no. 13, p. 5005, 2022.

[39] G. K. Kang, J. Z. Gao, S. Chiao, S. Lu, and G. Xie, "Air quality prediction: Big data and machine learning approaches," *Int. J. Environ. Sci. Dev*, vol. 9, no. 1, pp. 8–16, 2018.

[40] M. F. KANJO, "INTELIGENT SYSTEM FOR AIR POLLUTION PREDICTION." NEAR EAST UNIVERSITY, 2019.

[41] J. H. Thrall *et al.*, "Artificial intelligence and machine learning in radiology: opportunities, challenges, pitfalls, and criteria for success," *J. Am. Coll. Radiol.*, vol. 15, no. 3, pp. 504–508, 2018.

[42] A. Smola and S. V. N. Vishwanathan, "Introduction to machine

learning," *Cambridge Univ. UK*, vol. 32, no. 34, p. 2008, 2008.

[43] K.-L. Du and M. N. S. Swamy, *Neural networks and statistical learning*. Springer Science & Business Media, 2013.

[44] G. Ososkov and P. Goncharov, "Shallow and deep learning for image classification," *Opt. Mem. Neural Networks*, vol. 26, pp. 221–248, 2017.

[45] N. Gupta, P. Bedi, and V. Jindal, "Effect of activation functions on the performance of deep learning algorithms for network intrusion detection systems," in *Proceedings of ICETIT 2019: Emerging Trends in Information Technology*, 2020, pp. 949–960.

[46] Y. Yang, K. Zheng, C. Wu, and Y. Yang, "Improving the classification effectiveness of intrusion detection by using improved conditional variational autoencoder and deep neural network," *Sensors*, vol. 19, no. 11, p. 2528, 2019.

[47] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv Prepr. arXiv1811.03378*, 2018.

[48] B. Wang, X. Luo, Z. Li, W. Zhu, Z. Shi, and S. Osher, "Deep neural nets with interpolating function as output activation," *Adv. Neural Inf. Process. Syst.*, vol. 31, 2018.

[49] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "On mean absolute error for deep neural network based vector-to-vector regression," *IEEE Signal Process. Lett.*, vol. 27, pp. 1485–1489, 2020.

[50] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geosci. Model Dev.*, vol. 15, no. 14, pp. 5481–5487, 2022.

[51] Y. Zhou, X. Wang, M. Zhang, J. Zhu, R. Zheng, and Q. Wu, "MPCE: a maximum probability based cross entropy loss function for neural network classification," *IEEE Access*, vol. 7, pp. 146331–146341, 2019.

[52] C.-H. Chen, P.-H. Lin, J.-G. Hsieh, S.-L. Cheng, and J.-H. Jeng, "Robust multi-class classification using linearly scored categorical cross-entropy," in *2020 3rd IEEE International Conference on Knowledge Innovation and Invention (ICKII)*, 2020, pp. 200–203.

[53] N. K. Manaswi, N. K. Manaswi, and S. John, *Deep learning with*

applications using python. Springer, 2018.

[54] J. Duchi, E. Hazan, and Y. Singer, “Adaptive subgradient methods for online learning and stochastic optimization.,” *J. Mach. Learn. Res.*, vol. 12, no. 7, 2011.

[55] J. Lee *et al.*, “Wide neural networks of any depth evolve as linear models under gradient descent,” *Adv. Neural Inf. Process. Syst.*, vol. 32, 2019.

[56] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv Prepr. arXiv1412.6980*, 2014.

[57] B. Zohuri and F. M. Rahmani, “Artificial intelligence driven resiliency with machine learning and deep learning components,” *Japan J. Res.*, vol. 1, no. 1, 2023.

[58] A. Khan, A. Sohail, U. Zahoora, and A. S. Qureshi, “A survey of the recent architectures of deep convolutional neural networks,” *Artif. Intell. Rev.*, vol. 53, pp. 5455–5516, 2020.

[59] H. Tercan and T. Meisen, “Machine learning and deep learning based predictive quality in manufacturing: a systematic review,” *J. Intell. Manuf.*, vol. 33, no. 7, pp. 1879–1905, 2022.

[60] Z. Khoshraftar and A. Ghaemi, “Prediction of CO2 solubility in water at high pressure and temperature via deep learning and response surface methodology,” *Case Stud. Chem. Environ. Eng.*, vol. 7, p. 100338, 2023.

[61] K. Rahmani *et al.*, “Early prediction of central line associated bloodstream infection using machine learning,” *Am. J. Infect. Control*, vol. 50, no. 4, pp. 440–445, 2022.

[62] S. Kumar, P. Tiwari, and M. Zymbler, “Internet of Things is a revolutionary approach for future technology enhancement: a review,” *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0268-2.

[63] Y. Harbi, Z. Aliouat, S. Harous, A. Bentaleb, and A. Refoufi, “A review of security in internet of things,” *Wirel. Pers. Commun.*, vol. 108, pp. 325–344, 2019.

[64] A. Rayes and S. Salam, “The things in iot: Sensors and actuators,” in *Internet of Things From Hype to Reality: The Road to Digitization*, Springer, 2022, pp. 63–82.

[65] S. M. R. Islam, D. Kwak, M. D. H. Kabir, M. Hossain, and K.-S. Kwak, “The internet of things for health care: a comprehensive

survey," *IEEE access*, vol. 3, pp. 678–708, 2015.

[66] S. Munirathinam, "Industry 4.0: Industrial internet of things (IIOT)," in *Advances in computers*, vol. 117, no. 1, Elsevier, 2020, pp. 129–164.

[67] D. Darwish, "Improved layered architecture for Internet of Things," *Int. J. Comput. Acad. Res.(IJCAR)*, vol. 4, no. 4, pp. 214–223, 2015.

[68] M. Kalmeshwar and N. Prasad, "Internet of Things: architecture, issues and applications," *Int. J. Eng. Res. Appl*, vol. 7, no. 06, pp. 85–88, 2017.

[69] J. Pramanik, A. K. Samal, S. K. Pani, and C. Chakraborty, "Elementary framework for an IoT based diverse ambient air quality monitoring system," *Multimed. Tools Appl.*, vol. 81, no. 26, pp. 36983–37005, 2022.

[70] W. Zhang, L. Wang, J. Chen, W. Xiao, and X. Bi, "A novel gas recognition and concentration detection algorithm for artificial olfaction," *IEEE Trans. Instrum. Meas.*, vol. 70, pp. 1–14, 2021.

[71] F. I. Adhim *et al.*, "Carbon monoxide and methane gas identification system," in *2019 International Conference on Advanced Mechatronics, Intelligent Manufacture and Industrial Automation (ICAMIMIA)*, 2019, pp. 263–267.

[72] A. S. Hadi, M. Alsaker, A. Eshoom, M. Elmnifi, M. A. Alhmode, and L. J. Habeeb, "Development of Low-Cost and Multi-Material Sensing Approach for MQ 135 Sensor," *ECS Trans.*, vol. 107, no. 1, p. 17309, 2022.

[73] J. Saini, "Indoor Environmental Sensing Techniques for Occupant Health and Comfort," *Indoor Air Qual. Assess. Smart Environ.*, vol. 30, p. 17, 2022.

[74] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Inf. Process. Manag.*, vol. 45, no. 4, pp. 427–437, 2009.

[75] M. Hossin and M. N. Sulaiman, "A review on evaluation metrics for data classification evaluations," *Int. J. data Min. Knowl. Manag. Process*, vol. 5, no. 2, p. 1, 2015.

[76] J. M. Hughes, *Arduino: a technical reference: a handbook for technicians, engineers, and makers*. " O'Reilly Media, Inc.," 2016.

[77] M. I. Ahamed, M. N. Srinivasan, N. H. Venkatesh, R. Ganesh, R. U. Kumar, and R. A. Ameerudeen, “Free Space Laser Communication Using Modern Laser Diodes”.

[78] M. Fezari and A. Al Dahoud, “Integrated development environment ‘IDE’ for Arduino,” *WSN Appl.*, pp. 1–12, 2018.

[79] L. Louis, “working principle of Arduino and u sing it,” *Int. J. Control. Autom. Commun. Syst.*, vol. 1, no. 2, pp. 21–29, 2016.

[80] T. Kluyver *et al.*, “Jupyter Notebooks-a publishing format for reproducible computational workflows.,” *Elpub*, vol. 2016, pp. 87–90, 2016.

[81] D. Rolon-Mérette, M. Ross, T. Rolon-Mérette, and K. Church, “Introduction to Anaconda and Python: Installation and setup,” *Quant. Methods Psychol*, vol. 16, no. 5, pp. S3–S11, 2016.

[82] W. Donat and W. Donat, “The Raspberry Pi and the Arduino,” *Learn Raspberry Pi Program. with Python Learn to Progr. World’s Most Pop. Tiny Comput.*, pp. 349–364, 2018.

[83] D. S. Shanthi* and M. Pyingkodi, “Air Quality Index Prediction using Machine Learning Algorithms,” *Int. J. Recent Technol. Eng.*, vol. 8, no. 4, pp. 7489–7492, 2019, doi: 10.35940/ijrte.d5326.118419.

الخلاصة

يعد تلوث الهواء مصدر قلق عالمي متزايد بسبب آثاره الضارة على صحة الإنسان والبيئة. على الرغم من الاهتمام الكبير الممنوح لتلوث الهواء باعتباره قضية بيئية مهمة، إلا أن هناك فجوة بحثية من حيث القياسات الشاملة لتركيزات الملوثات، بما في ذلك ثاني أكسيد الكربون، وثاني أكسيد النيتروجين، وثاني أكسيد الكبريت، والاوزون ، وPM2.5، وPM10. وتحد هذه الفجوة من القدرة على التنبؤ بدقة بتلوث الهواء مقدما، مما يجعل من الصعب على الأفراد إدارة صحتهم وعلى الحكومات وضع سياسات مستنيرة.

تقدم هذه الرسالة نظامًا ذكيًا لمراقبة جودة الهواء يعمل على دمج تقنية إنترنت الأشياء (IoT) مع نماذج التعلم العميق متعددة الطبقات Perceptron (MLP) للتنبؤ بمستويات تلوث الهواء. تم إنشاء هذا النظام باستخدام min_max normalization وتصحيح القيم المفقودة في المعالجة المسبقة، وتم استخدام Pearson correlation coefficient وهو اختيار حكيم للميزات (Features) الأكثر تأثيرًا، بما في ذلك PM2.5، وPm10، وNOx، وNO2، وNO، وO3، وCO2. في مرحلة التصنيف، يفضل هذا النموذج اختيار محسن adam الذي يعتبره أكثر ملاءمة للتعلم العميق. ReLU وsoftmax كوظائف التنشيط. تم تنفيذ مجموعتي بيانات لتقييم النموذج المقترح، ومجموعة بيانات المجلس المركزي لمكافحة التلوث (CPCB) القياسي ومجموعة بيانات محطة جودة الهواء الهندية. تعد مكونات إنترنت الأشياء، مثل أجهزة الاستشعار (sensors) وأجهزة Arduino، جزءًا لا يتجزأ من مكونات النظام المقترح، والتي تلعب دورًا أساسيًا في جمع بيانات جودة الهواء في الوقت الفعلي. ثم تتم معالجة هذه البيانات وإدخالها في نموذج MLP ، الذي تم تدريبه على مجموعة البيانات للتنبؤ بدقة عالية بمستويات تلوث الهواء.الذي تم تدريبه بدقة على مجموعة بيانات CPCB، وبالتالي إنشاء تنبؤات دقيقة بتلوث الهواء.

وقد أظهرت نتائج العمل المقترح نتائج مشجعة من حيث الدقة العالية. تصل دقة مجموعة بيانات CPCB إلى ٩٩,١١٥٪. علاوة على ذلك، بعد التحقق من صحته باستخدام مجموعة بيانات متميزة من محطات جودة الهواء في الهند، يحافظ النموذج على معدل أداء جدير بالثناء يبلغ ٩٧,٤٧٪. يؤكد هذا التحقق المتبادل على قوة النموذج وإمكانية تطبيقه عبر سياقات جغرافية متنوعة.

جمهورية العراق

وزارة التعليم العالي والبحث العلمي

جامعة بابل

كلية تكنولوجيا المعلومات

نظام ذكي لتصنيف ومراقبة جودة الهواء باستخدام تقنيات تعلم الالة

رسالة

مقدمة الى مجلس كلية تكنولوجيا المعلومات للدراسات العليا بجامعة بابل
في استيفاء جزئي لمتطلبات درجة الماجستير في كلية تكنولوجيا المعلومات
/ شبكات المعلومات

من قبل الطالبة

هدى كاظم علوان حسون

باشراف

ا.د غيداء عبد الحسين بلال

م.د صبا محمد حسين

٢٠٢٣ م　　　　**1445 هـ**