THESIS

AIR POLLUTANT SOURCE ESTIMATION FROM SENSOR NETWORKS

Submitted by

Tanmay Thakur

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Master of Science

Colorado State University

Fort Collins, Colorado

Spring 2024

Master's Committee:

Advisor: Kevin Lear

Ali Pezeshki
Ellison Carter

ABSTRACT

AIR POLLUTANT SOURCE ESTIMATION FROM SENSOR NETWORKS

A computationally efficient model for the estimation of unknown source parameters using the Gaussian plume model, linear least square optimization, and gradient descent is presented in this work. This thesis discusses results for simulations of a two-dimensional field using advection-diffusion equations underlining the benefits of plume solutions when compared to other methods. The Gaussian plume spread for pollutant concentrations has been studied in this work and modeled in Matlab to estimate the pollutant concentration at various wireless sensor locations. To set up the model simulations, we created a field in Matlab with several pollutant-measuring sensors and one or two pollutant-emitting sources. The forward model estimated the concentration measured at the sensors when the sources emit the pollutants. These pollutants were programmed in Matlab to follow Gaussian plume equations while spreading. The initial work estimated the concentration of the pollutants with varying sensor noise, wind speed, and wind angles. The varying noise affects the sensors' readings whereas the wind speed and wind angle affect the plume shape.

The forward results are then applied to solving the inverse problem to determine the possible sources and pollutant emission rates in the presence of additive white Gaussian noise (AWGN). A vector of possible sources within a region of interest is minimized using L2 minimization and gradient descent methods. Initially, the input to the inverse model is random a guess for the source location coordinates. Then, initial values for the source emission rates are calculated using the linear least squares method since the sensor readings are proportional to the source emission rates. The accuracy of this model is calculated by comparing the predicted source locations with the true locations of the sources. The cost function reaches a minimum value when

the predicted sensor concentrations are close to the true concentration values. The model continues to minimize the cost function until it remains fairly constant. The inverse model is initially developed for a single source and later developed for two sources. Different configurations for the number of sources and locations of the sensors are considered in the inverse model to evaluate the accuracy. After verifying the inverse algorithm with synthetic data, we then used the algorithm to estimate the source of pollution with real air pollution sensor data collected by Purple Air sensors. For this problem, we extracted data from Purpleair.com from 4 sensors around the Woolsey Forest fire area in California in 2018 and used its data as input to the inverse model. The predictions suggested the source was located close to the true high-intensity forest fire in that area.

Later, we apply a neural network method to estimate the source parameters and compare estimates of the neural network with the results from the inverse problem using the physical model for the synthetic data. The neural model uses sequential neural network techniques for training, testing, and predicting the source parameters. The model was trained with sensor concentration readings, source locations, wind speeds, wind angles, and corresponding source emission rates. The model was tested using the testing data set to compare the predictions with the true source locations and emission rates. The training and testing data were subjected to feature engineering practices to improve the model's accuracy. To improve the accuracy of the model different configurations of activation functions, batch size, and epoch size were used. The neural network model was able to obtain an accuracy above 90% in predicting the source emission rates and source locations. This accuracy varied depending upon the type of configuration used such as single source, multiple sources, number of sensors, noise levels, wind speed, and wind angle used. In the presence of sensor noise, the neural network model was more accurate than the physical inverse model in predicting the source location based on a comparison of R^2 scores for fitting the predicted

source location to the true source location. Further work on this model's accuracy will help the development of a real-time air quality wireless sensor network application with automatic pollutant source detection.

ACKNOWLEDGEMENTS

I would first like to thank my advisor Dr. Kevin Lear, who has looked over many versions of this document patiently and has constantly provided me with invaluable feedback and suggestions that have helped in shaping thing thesis. He has taught me a lot about wireless sensor networks, signal processing, computer modeling, research, and academic writing which I believe are important skills and will help me throughout my life. I would not have anyone else as my advisor.

I would like to thank my committee members, Dr.Ali Pezeshki and Dr. Ellison Carter for their constant support throughout my thesis.

Finally, I would like to thank my family for their constant support throughout my difficult times during the academic career.

DEDICATION

I would like to dedicate this thesis to my late mother, Rohini Madan Thakur. She worked very hard to support my education. Her motivation and love have made me self-sufficient and educated. Without her my life is incomplete but fulfilling her dreams makes me happy.

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 PROBLEM STATEMENT AND MOTIVATION

Since the growth of industrialization, poor air quality has been a major concern across several regions of the world. Thus, it is necessary to develop technologies that quantify air pollution and help minimize its effects. Air pollution has a variety of negative effects on climate, human health, and nature. The adverse effects of air pollution such as respiratory infections and heart diseases across the population including children have been increasing over the past years [3]. Several long-term effects such as loss of biodiversity and human health have been increasing substantially.

1.1.1 Poor air quality

Unacceptable air quality is caused due to the emission of hazardous toxic gas, smoke, dust, and other particulates emitted by industries, vehicles, and forest fires [4]. Over the past decade, monitoring the number of pollutants emitted by various human and natural activities has become necessary. Many pollutants are now measured with high accuracy. Pollutants that were difficult to measure some years ago are now being measured routinely using the latest sensors and communication technologies. The ever-increasing presence of sensors and computing power has made it possible to develop models with high temporal and spatial accuracy [13]. Recently, there has been high demand to regulate the number of pollutants emitted into the atmosphere by various industrial and residential activities. Such regulations require the use of sensor technologies for the collection and monitoring of air pollutants as well as notifying industries and households to stop any activities that emit more pollutants than are permitted or have potentially harmful effects on

the environment. Various models have been implemented to predict the amount of pollutant concentration that might be present at a particular location due to the emission of toxic pollutants by an industry, vehicle, or forest fire. Such models accompanied by wireless sensor networks have accomplished tremendous success over the past decade in monitoring and reducing pollution levels. Recently, Wireless Sensor Networks (WSNs) have achieved great potential for widespread applicability in the fields of monitoring, surveillance, data collection, and medical telemetry [17]. By combining the data collected from wireless sensor networks with advanced modeling techniques, our research aims to develop a robust methodology for source identification and quantification. Traditional inverse methods involve solving mathematical models that consider the dispersion of pollutants in the atmosphere and the measurements obtained from the sensor network. These methods help estimate the emission rates and source locations based on the observed pollution levels.

However, this thesis also explores the potential of neural networks in improving the accuracy and efficiency of source identification. Neural networks can learn complex patterns and relationships from large datasets, making them suitable for analyzing the vast amount of sensor data collected by the wireless network. By training neural networks on historical data and known emission sources, they can be used to predict the emission rates and locations of unknown sources with higher precision.

1.2 THESIS ORGANIZATION

This thesis is organized as follows: Chapter 2 provides background information on Gaussian plume model also referred to as the forward model, the inverse model, and the neural network model. In Chapter 3 the analysis conducted for air quality sensors data is presented. This chapter explains the data regression and correlation conducted on deployed air quality sensors and

highlights the results. Chapter 4 provides an in-depth explanation of the forward Gaussian plume dispersion model. This chapter primarily explains the physical parameters derived from the general equations of the model. The later sections focus on derivation for the Gaussian plume model and its implementation. Chapter 5 discusses the contour plot results from the Matlab program for the forward Gaussian plume model. Chapter 6 explains the need for source parameters estimation. This chapter mentions any assumptions considered for the reverse Gaussian plume model. Sections of this chapter explain the derivation and implementation of the inverse Gaussian plume model. Chapter 7 discusses the inverse source parameter results obtained using Matlab simulation. Chapter 8 explains the solving inverse problem with neural networks. Finally, Chapter 9 is a summary of the findings discussed in this thesis, and an outline for future work is presented.

1.3 INTRODUCTION TO FORWARD MODEL AND INVERSE MODEL

The terminology and physics for the transport of particulates in the atmosphere is complex thus, to predict the unknown pollutant source we divide the model into two sections.

In this model, we first needed to create the sensor/sensor concentrations using the pollutant source information. The generated sensor concentration is synthetic data used as input for creating further inverse calculations for source parameters. Section 1.3 determines the pollutant concentration using several parameters involving diffusion and advection caused by wind. We use source information such as the source emission rate, source location, and wind parameters to determine the pollutant concentrations at the desired sensor locations. This calculation of the pollutant concentrations at different sensor locations using the Gaussian plume solution is called the "**forward model**". But, in the thesis, we focused on inverse engineering the Gaussian plume model. In this inverse engineering method, the focus is to predict the source emission rates and source locations using the sensor's pollutant concentration readings. This inverse Gaussian plume

method of predicting the source parameters is referred to as the "**inverse model**" in this work. The data input for the inverse model is the synthetic data generated for the sensor concentrations in the forward model.

The Gaussian plume model has proven to be highly accurate in determining the plume dispersion behavior and approximation of the pollutant concentration at a particular location [1].

1.4 BACKGROUND INFORMATION ABOUT GAUSSIAN PLUME SOLUTION

The Gaussian plume solution was originally derived by Sutton [2]. His work initially considered only a single point source but gradually was implemented for more general cases by including ground-level deposition, line sources, area sources, and height-dependent parameters. The Gaussian plume model requires knowledge about the location of the source and source emission rates to determine the emitted pollutant concentration at a particular location. But, in a practical situation such as a forest fire, unknown industrial emission, or vehicle emission it is often difficult to determine the source location and the source emission rates. We in this work have developed an inverse Gaussian plume model in which the pollutant concentration at several sensors, also known as receptors, locations in a region are known whereas, the source emission rates, and source locations are estimated using mathematical calculations through the model. We believe this model will have significance in situations where it is difficult to access fire-prone areas or scenarios where it is not economical to deploy wireless sensors all over the area of interest, or where access to the source location of the pollution is otherwise restricted for geographic, legal, or political reasons. This model would provide the ability to extract more information regarding the source pollutant from a limited number of sensor nodes in a wireless sensor network with the sensors located at some distance from the source.

Once the inverse model started predicting the source parameters accurately using the Gaussian plume solution for the synthetic data for the sensor concentrations we moved to use actual sensor concentrations data. The actual sensor concentration data for air pollution was gathered using PurpleAir.com sensors which are available all over the US and other countries.

1.5 BACKGROUND INFORMATION ABOUT SOURCE APPORTIONMENT (RECEPTOR) MODELS

Source apportionment (receptor) models are sophisticated mathematical procedures used to analyze and quantify the different sources of ambient air pollutants at a specific location, known as the receptor site. These models primarily rely on concentration measurements of various pollutant species at the receptor site and do not necessarily require detailed information about emissions and meteorological conditions. The U.S. Environmental Protection Agency's National Exposure Research Laboratory (NERL) has developed a comprehensive PM (particulate matter) source apportionment program that encompasses both measurement and modeling aspects [23].

The NERL PM source apportionment program includes several key components:

1. **Chemical and physical characterization of PM samples:** This involves collecting samples of particulate matter from the air at the receptor site and subjecting them to extensive laboratory analysis. The goal is to identify and quantify the different chemical components and physical properties of the PM, which can help distinguish between various pollution sources.

2. **Radiocarbon (14C) determination of biogenic content:** Radiocarbon analysis is used to estimate the contribution of biogenic (naturally occurring) sources to the total PM composition. By measuring the levels of radiocarbon in the PM samples, scientists can assess the extent of organic material derived from sources such as vegetation or biomass burning.

3. **Receptor model development:** NERL researchers work on developing receptor models, which are mathematical algorithms that interpret concentration measurements and estimate the contributions of different pollution sources. These models employ statistical techniques and data analysis methods to infer the source types, such as industrial emissions, vehicle exhaust, or residential combustion.

4. **Receptor model applications:** Once the receptor models are developed, they are applied to real-world data to determine the contributions of specific sources to the observed pollution levels. This information can be crucial for designing effective pollution control strategies and policies.

5. **Source profile measurement and collection:** To improve the accuracy of the receptor models, NERL conducts measurements and collects data on source profiles. These profiles describe the chemical composition and characteristics of emissions from various pollution sources. By incorporating these profiles into the models, researchers can enhance their ability to identify and quantify different source contributions accurately.

Overall, NERL's PM source apportionment program combines measurements, radiocarbon analysis, model development, and source profile collection to comprehensively assess the sources of ambient air pollutants at receptor sites. This integrated approach helps in better understanding the composition and origin of particulate matter, which is vital for mitigating the adverse effects of air pollution and developing effective air quality management strategies.

1.5.1 Comparing Source apportionment (receptor) models and Gaussian plume approach

The Source apportionment method and Gaussian plume methods aim is to find the pollutant source information but, they differ in several ways as explained below:

Principle: The receptor model approach, as described earlier, focuses on analyzing concentration measurements at a receptor site to estimate the contributions of different pollution sources. It does not heavily rely on emissions and meteorological data.[24] In contrast, the inverse Gaussian plume model approach utilizes a dispersion model based on emissions data, meteorological conditions, and source characteristics to simulate the transport and dispersion of pollutants in the atmosphere. The model calculates the expected concentrations at receptor locations based on these inputs.

Input Requirements: The receptor model approach primarily relies on concentration measurements at the receptor site. It does not require extensive information on emissions or meteorology, although some additional data may be useful for validation purposes. On the other hand, the inverse Gaussian plume model approach requires detailed input data, including emission rates, source characteristics (e.g., stack height, exit velocity), meteorological data (e.g., wind speed, stability class), and receptor locations [23].

Spatial Considerations: Receptor models are generally applied at specific receptor sites, where pollutant concentrations are measured. They provide information on the contributions of different sources to the pollution levels at that specific location. In contrast, inverse Gaussian plume models can simulate pollutant dispersion over a larger area, considering the transport and diffusion of pollutants over distances from the source. This allows for a broader spatial understanding of pollution patterns and impacts.

Source Identification: Receptor models excel in identifying and quantifying specific source types contributing to pollution at a receptor site. By analyzing the chemical and physical characteristics of the measured pollutants, receptor models can attribute contributions to different source categories (e.g., industrial, vehicular, biomass burning). In contrast, inverse Gaussian plume

models focus more on understanding the overall dispersion patterns and the impact of sources collectively, rather than providing a detailed breakdown of individual sources.

Validation: Receptor models can be validated using additional data, such as emissions inventories or source profiles, to assess their accuracy and reliability. Inverse Gaussian plume models can also be validated using measurements at various locations, comparing the model predictions with observed concentrations. However, these models are more reliant on the accuracy of the input data, such as emissions and meteorological information.

In summary, while both the receptor model approach and the inverse Gaussian plume model approach aim to assess the sources of ambient air pollutants, they differ in their principles, input requirements, spatial considerations, and focus on source identification. The receptor model approach emphasizes the analysis of concentration measurements at receptor sites, while the inverse Gaussian plume model approach utilizes a dispersion model based on emissions and meteorology data to simulate pollutant transport and dispersion over a wider area. Each approach has its own strengths and can provide valuable insights into understanding and managing air pollution.

CHAPTER 2

BACKGROUND

2.1 AIR QUALITY MONITORING

Air pollution, which is one of the most crucial factors impacting the quality of life and health of the growing urban population in industrial societies, has led to the development of numerous technologies for real-time air quality monitoring. Traditionally, air quality monitoring stations are usually of large sizes and have high costs for installation and maintenance, which limits their potential for deployment in dense cities. Alternatively, there are lower-cost and reliable solutions available using wireless sensor network technology. Wireless sensor networks also help reduce the labor-intensive operating cost of monitoring parameters or asset conditions. This concept is called distributed sensor networks which are explained in the subsection below.

2.1.1 Distributed sensor networks

Distributed sensor networks are spatially distributed autonomous sensors to monitor physical or environmental conditions and pass their data through the network to a central location for analysis. A distributed sensor network (DSN) consists of a set of nodes collecting data from sensors. When the nodes communicate, each node uses the information received from other nodes with the local information to update its estimate of the state of the parameters measured [15]. Distributed sensor networks may be deployed in hostile areas where communication is monitored, and nodes are subject to capture and surreptitious use by an adversary.

2.2 COMPUTER MODELING

Computational modeling is the use of computers to simulate and study complex systems using mathematics, physics, and computer science. A computational model contains numerous variables that characterize the system being studied. Simulation is done by adjusting the variables alone or in combination and observing the outcomes. Computer modeling allows scientists to conduct thousands of simulated experiments by computer. Computer modeling consists of writing a computer program version of a mathematical model for a physical or biological system. Computer simulations that are run according to such programs can produce knowledge out of reach of mathematical analysis or natural experimentation [18].

2.2.1 Physical forward model

2.2.1.1 What is physical modeling?

Physical modeling is the process to create a computer program that replicates the physical parameters in the system and runs simulations to understand the response or results of the system. This physical model can be created using computer programming languages or any computer modeling software package. A physical model needs to know the physical characteristics of each parameter in the systems so it can be accurately replicated in the model.

2.2.1.2 Gaussian plume for physical modeling

In this thesis, we are modeling the physical dispersion, i.e., spread, of pollutants, using the Gaussian plume equation to predict different parameters used in the equation. Thus, it is important to consider each parameter in the model and represent it accurately to implement a physical digital twin system that can predict results very close to the true values. We studied the Gaussian plume equation in detail so it could be effectively implemented in the physical model.

This Gaussian plume physical model is developed as a forward physical model to calculate the wireless sensor measurements and an inverse model to predict the source of the pollutant. The forward physical model and the inverse physical model are explained in Chapters 4 and 6 respectively of this thesis.

2.3 NEED TO PREDICT SOURCE TERMS

Urban air quality is a major issue owing to urbanization, increasing population, and industrialization. Consequently, there has been very high interest in finding the source of pollutants and solutions to minimize emissions. To reduce the pollutants being transported into the air it is very important to know the source of the pollutant. Two major pollutant source parameters are the location of the source and the rate at which the source is emitting the pollutants into the air, called source emission rate. The dynamics of air pollutant transportation have been extensively studied. Turner [1] mentions a multitude of possible sources of air-borne particles, including those of anthropogenic origins such as industrial complexes and automobiles, as well as natural sources such as dust storms and volcanic eruptions. Several studies have been done to determine the downwind contaminant concentrations given the source emission rates and meteorological conditions. The work was originally done by Sutton [2] and extended to higher dimensions involving ground-level deposition [6], multiple sources [7], height-dependent wind speed, and diffusion coefficients. Using this work there were several research conducted on source term estimations such as Ermark [6], Calder [6], and Chang [14]. Since this work proved to be highly reliable it was used in this research to implement an inverse Gaussian plume model to predict the source locations and the emission rates.

2.4 INTRODUCTION TO NEURAL NETWORK MODEL

Later, when the physical inverse model was predicting the source parameters accurately using the sensor concentration synthetic data and the real-world purple air data, we tried to solve the same problem using neural network methods. Neural networks are comprised of node layers, containing an input layer, one or more hidden layers, and an output layer. Each node, or artificial neuron, connects to another and has an associated weight and threshold. In this model, the training data had information about the sensor readings and source parameters. Once the model was trained with a high level of accuracy, we started the prediction of source parameters in different settings which will be discussed in depth in Chapter 8.

CHAPTER 3

SENSOR ANALYSIS OVERVIEW

3.1 INTRODUCTION

Several wireless sensor networks are used for applications such as air quality monitoring, water pressure monitoring, tank levels, and other utility monitoring applications. In such a situation, gathering accurate data from the sensors becomes very important. Working through several sensor-related projects we needed to understand each aspect of wireless sensors' susceptibility to failure. A sensor analysis study helped us to understand how wireless sensors could be made reliable for crucial data acquisition applications. Some of the deployed air quality sensors available at purpleair.com were selected for analysis. Purple Air sensors are developed using an ESP8266 chip and provide functionalities such as Analog to digital conversion, Digital inputs, and much more including connecting to a WIFI network and uploading data to the cloud.

Sensor's data ($PM_{2.5}$ readings) available in fire-prone areas were downloaded. It was observed that all the locations had two sensors deployed to read the same air quality data at the same location. After analysis of two sensor readings at the same location, it was observed that the second sensor might be used if the first sensor has abnormal readings, or it fails. Work was done to identify some outliers and obtain results for statistical methods such as correlation and data regression.

3.1.1 Sensor Correlation Overview

We did the correlation analysis for 11 air quality sensors available at purpleair.com. The analysis was specifically based on the correlation between two sensors available at the same

location. The motivation behind this analysis was to learn about any contradiction that might exist between the data collected at the same location by the two sensors of the same type. The data from the two sensors were individually separated into night and daytime data sets. The objective behind such separation was to consider any anomaly in data between the two sensors during daytime and nighttime. There were some anomalies observed in data between two sensors at the same locations. In Figure 2.1 we have indicated some of the instances in the graph where there are several contradictions in the data between the two sensors which are circled in red.

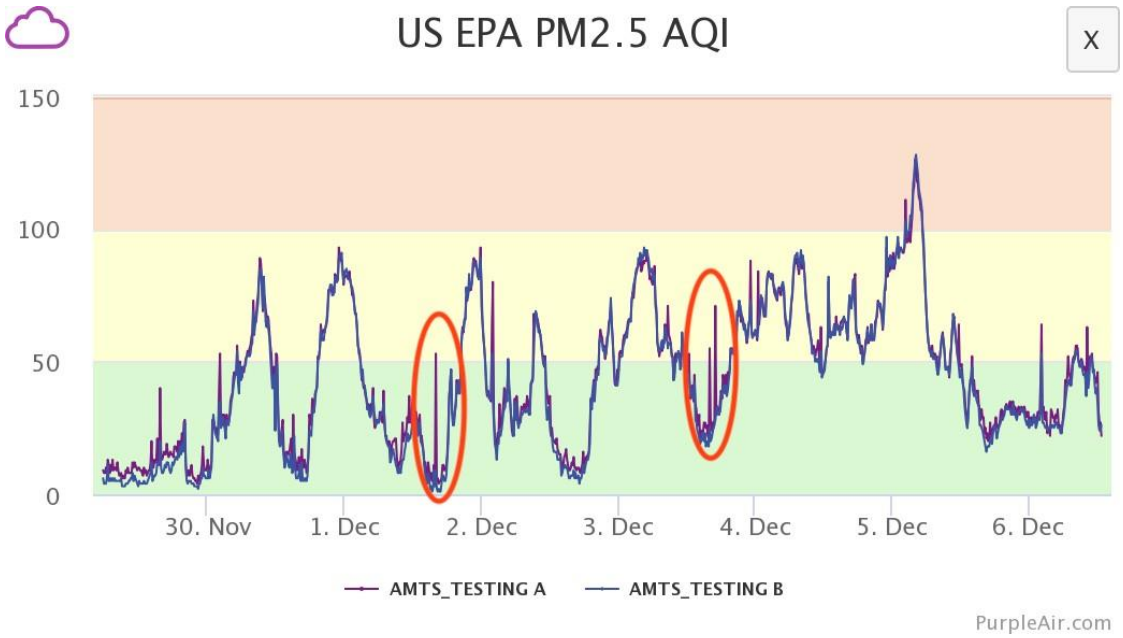

Figure 3.1: Purple Air data reading for air quality sensors installed in West Sacramento

Below in Figure 3.2 is a chart for one of the Purple Air sensors used in the analysis. The location is Palisade, CO. There was a brush fire in the nearby area during August 2018. We analyzed the data during August 2018 for this location.

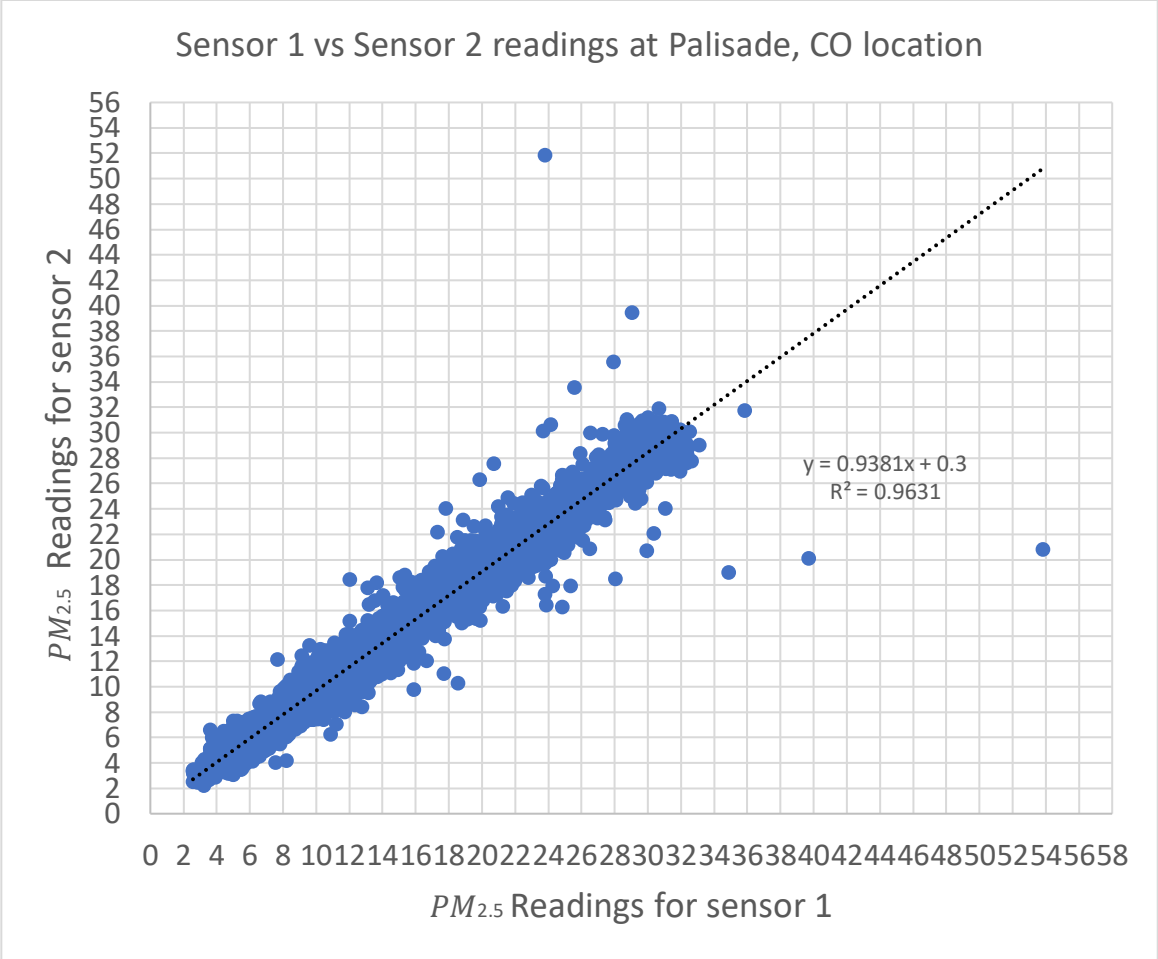

Figure 3.2: Purple Air data reading comparison between sensor 1 and sensor 2 location at Palisade, CO.

It was observed that the correlation between sensor 1 and sensor 2 was 0.981390121. The average value for sensor 1 was 14.21898981 whereas, the average value for sensor 2 was 13.63851074.

We also observed sensors across some more forest fires such as the Logan fire in Logan County, Colorado, and the Woolsey fire in Ventura County, CA. The correlation values for sensor 1 and sensor 2 for these fire locations were above 0.96.

3.2 CONCLUSION

After this study, it was concluded that the air quality data from Purple Air sensors was very reliable and could be used in this model. It was also observed that at locations with bad air quality during a particular time/day there was a spike in the reading from the sensor. These events included forest fires in cities. Also, in the later part of the study, the correlation between two sensors at the same location was studied. It was concluded from this study that two sensors at the same location in some cases had different readings at the same timestamps. But this anomaly was only for a small fraction, about 2 % of the overall studied sensors.

CHAPTER 4

FORWARD GAUSSIAN PLUME MODEL

4.1 INTRODUCTION

The Gaussian plume model is the most commonly used air pollution model which describes the plume distribution from a localized pollutant source. The concentration of air pollutants at a particular location is a function of several physical parameters such as emission rate, distance from the source, and atmospheric conditions. The atmospheric conditions having the most impact on the particle concentration include wind speed, wind direction, and vertical structure of the temperature [11]. It is important to consider the atmospheric stability conditions and various temperature parameters in a dispersion model. The stability conditions help to determine the vertical motions of the particulates in the atmosphere. The Gaussian dispersion model assumes that atmospheric turbulence is both stationary and homogenous [11].

The Gaussian dispersion model is useful for environmental engineers and scientists who study atmospheric pollutant transportation. The forward Gaussian plume model has been used in several industrial software packages for monitoring pollutant emissions and regulatory purposes. There has been a higher interest in applications related to nuclear and biological contaminant release [6, 10], for which the importance of analytical approaches is nicely summed up in a review article by Settles: “plume dispersion modeling is central to homeland security” [14]. The forward Gaussian plume work explained in the following section is based on the work in [7]. We considered certain assumptions for air quality monitoring for circumstances such as forest fires, industrial pollutant emissions, or vehicle emissions. These assumptions included constant source emission

rate for point sources, constant wind speed, and direction. We also assumed negligible topography. These assumptions or boundary conditions are explained in more detail in Section 4.2

4.1.1 Pollutant source and sensor relation

To estimate the concentration of pollutants measured by a sensor at a location, the Gaussian plume model is used. These pollutants are transported from a source of pollutants such as a chimney or forest fire. In the Gaussian plume model, the source position, source emission rates, wind speed, and wind direction are used to calculate particulate concentrations. The model is also important when solving the inverse problem to predict source location and source emission rates which will be discussed in Chapter 6. These pollutants are emitted by a source at a certain emission rate. Thus, it is important to understand the relationship between the particulate concentration at the sensor and the sensor location, source location, and source emission rates.

4.1.2 Plume Spreading

To implement a Gaussian plume model, it is necessary to understand the behavior of the plume spread in various directions. The plume primarily spreads in the downwind direction from the emission source. However, the plume also spreads vertically and in the horizontal direction perpendicular to the wind. The plume spread in the perpendicular directions is driven by diffusion as well as atmospheric conditions causing eddy currents. To understand this plume spread mathematically we need to define a coordinate system. The x direction is the wind direction, and the y direction is the horizontal direction perpendicular to the wind whereas the z direction is the vertical. The x and y directions are on the horizontal plane with reference to the source whereas the z direction is on the vertical plane. The x-axis has the highest concentration close to the source, and the concentration decreases as we move away from the source along the positive x-axis. The

particulate concentration also reduces at locations farther away in the direction perpendicular to the x-axis, i.e. along both the negative and positive y-axis.

4.1.3 Stability

The stability class determines the atmospheric condition across the plume which is very important in determining the plume shape and the pollutant concentration profile in the x,y, and z directions. The stability conditions are often determined using Pasquill-Turner stability classes [2]. The plume dispersion coefficients depend on the stability conditions, and this does have an effect on how far the plume spreads both horizontally and vertically from the center line, i.e. the x-axis. Stability conditions and their effects on the plume shape are explained in more detail in Section 4.3.4.

In the Gaussian plume dispersion model, the concentration of pollution downwind from a source is treated as spreading outward from the centerline of the plume following a normal statistical distribution [18].

4.2 PHYSICAL PARAMETER MODELING FOR AIR QUALITY DATA

The Gaussian plume forward model is based on a general equation where we consider a single contaminant having a mass concentration of C(x,t) [mg/m^3] at the location $\vec{X}$ = (x, y, z) [m] with time t > 0[s]. The differential form for the law of conservation of mass [7] can be expressed as:

$$\frac{\partial C}{\partial t} + \nabla . \vec{J} = S \tag{4.1}$$

S, is the source (positive) or sink (negative) term, and $\vec{J}$ represents a mass flux function of a contaminant on account of effects such as advection and diffusion.

The transport of contaminants in the atmosphere due to the primary air movement such as wind is referred to as advection, which follows the concept of mean fluid flow [2]. For example, if the wind is blowing toward the west, then the advection will transport all the contaminants in a westerly direction. In contrast, any transport of contaminants due to random motions of the air within the atmosphere is called diffusion. Diffusion eliminates sharp discontinuities in concentration and results in smoother, flatter concentration profiles. In this work, we have considered advection and diffusion separately. The atmospheric diffusion follows Fick's law where the diffusive flux is proportional to the concentration gradient or $J_d = -K\nabla C$. The negative sign signifies the flow of contaminants from regions of high concentration to regions of lower concentration. $K_{(\vec{x})} = diag(K_{(x)}, K_{(y)}, K_{(z)})$ $[m^2/s]$ is a diagonal matrix whose entries are the turbulent eddy diffusivities that in general are functions of position [4]. Advection can be signified as $\vec{J}a = C\vec{\mu}$. where μ is the wind velocity. The addition of these two contributions yields total flux $\vec{J} = -K\ \nabla\ C + C\vec{\mu}$. We substitute this parameter into Equation 4.1 which results in a three-dimensional advection-diffusion equation as below:

$$\frac{\partial C}{\partial t} + \nabla.(C\,\vec{\mu}) = S + \nabla.(K\nabla C) \tag{4.2}$$

The above equation is considered as the general advection-diffusion equation which we use in this work with substitutions suitable for practical situations such as the emission of pollutants from a source such as an industrial factory, forest fires, or vehicles.

An analytical solution to Equation 4.2 for the case of a point source was developed in [2] by Sutton. The solution accounts for advection in the downwind (x) direction and diffusion in the perpendicular horizontal (y) and vertical (z) directions. Very far from the centerline (x-axis), we expect the concentration to approach zero. A well-known solution to the diffusion equation with these boundary conditions is the Gaussian function. Accordingly, the full 3D solution to Equation (4.2) is called a Gaussian plume. An illustration of a Gaussian plume is shown in the figure below:

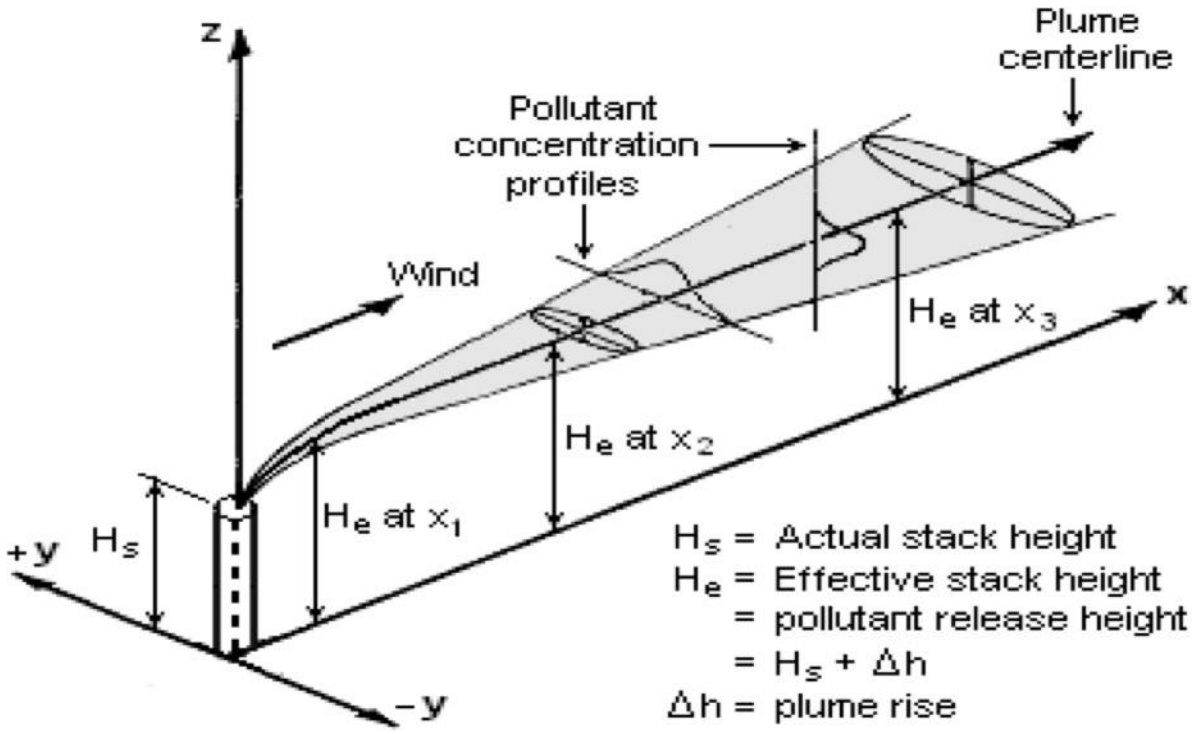

Figure 4.1 Gaussian Plume. Reproduced from [3]

H_s is the actual height of the emission source. H_e is the effective stack height at which pollutants start to disperse in the y and z directions. We have modified the Gaussian plume dispersion figure considering the application where the source is located just above ground level, or H_e is negligible compared to the vertical spread of the plume. We have approximated the actual

height to be zero for situations such as vehicle emissions and forest fires. We have considered the source emission rate Q [kg/s] and the wind speed μ is constant everywhere in the plume.

A few approximations and consideration of boundary conditions simplify the details of the Gaussian plume solution. The wind velocity is considered to be sufficiently large such that the diffusion in the x-direction is much smaller than the advection [7]. Variations in the topography are considered to be negligible, which makes it reasonable to assume advection only occurs along the x-axis in the horizontal direction. Certain boundary conditions implemented in [7] are assumed to be the same in our work. The boundary conditions are explained below.

The concentration at x = 0 for non-zero y and z i.e. C (0,y,z) is equal to 0. This boundary condition is based on the consequence of unidirectional wind, which also implies there are no contaminants from the source for negative x-direction as shown in Figure 4.3.

The concentration at $y = \pm\infty$, finite x, and finite z i.e. $C(x, \pm\infty, z)$ is zero. This boundary condition is based on the aspect that the total concentration must remain constant.

The concentration at $z = \infty$, finite x, and finite y i.e. $C(x, y, \infty)$ is zero. This boundary condition is also based on the aspect that the total concentration must remain constant.

The contaminant does not penetrate the ground. This can be represented as:

$C(x, y, z) = 0$ for $z < 0$

The above assumptions and boundary conditions result in a steady-state concentration solution C (x,y,z) where each infinite slab in the y-z directions contains the same mass of particulates, Q $\Delta x/|\vec{\mu}|$. Within each of these slabs, diffusion due to eddy currents spreads out the particulates in the y and z directions. The mass distribution is gaussian in those directions with increasingly larger spreads further from the source. As shown in Figure 4.2 the source is located at the origin and the plume is gradually dispersing as it moves along the x-direction.

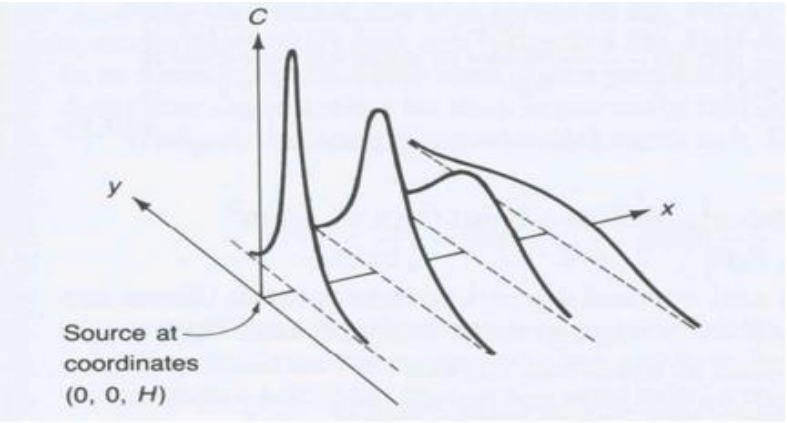

Figure 4.2: Denotes the plume dispersion increment in y and z directions as the distance along with the wind in the x-direction increases. Reproduced from [3].

4.2.1 Wind speed

For the simulations, we initially considered constant wind speed. Based on discussions in references [12][14][15] for wind speed estimation it was found that several factors are to be considered while calculating a variable wind speed with respect to time. The factors include the elevation at which the wind speed is measured, the atmospheric humidity, the topography of the area, and the method of wind measurement. Some methods of wind measurement included the least square plane [12][4], least-square quadratic [12][5], and Little's method also called the "triangular shepherd" [12][8]. But after evaluating the methods, it was observed that for an initial

model we need to assume a constant wind speed. The wind speed was considered constant to simplify the model. We have considered a wind speed of 5 [m/s] as suggested by Stockie [2]. In these simulations, we have assumed a constant wind direction as depicted in Figure 4.6. All the simulations are steady-state conditions since time-dependent simulations are complicated.

4.2.2 Wind Angle

In Gaussian plume modeling, wind direction plays a crucial role in determining the dispersion of pollutants in the atmosphere. The wind direction is parallel to the plume's axis, the pollutant travels in a long and narrow plume. The angle between the wind direction and the plume's axis also affects the plume's shape and size [4]. Therefore, accurate knowledge of wind direction is essential for effective air pollution management and regulatory compliance. We have considered different wind angles which are discussed in Table 5.1 and later sections.

4.3 FORWARD GAUSSIAN PLUME MODEL EQUATION

When we consider the mass concentration with location coordinates x, y, and z, the general advection-diffusion equation 4.2 helps to construct a simplified solution for the forward Gaussian plume model as derived in [12] and stated below:

$$C(x,y,z) = \underbrace{\frac{Q}{2\mu\pi\sigma_x\sigma_y}}_{\text{(i)}} \quad \underbrace{e^{\left(\frac{-(z-H)^2}{2\sigma_z^2}\right)}}_{\text{(ii)}} \underbrace{e^{\left(\frac{-y^2}{2\sigma_y^2}\right)}}_{\text{(iii)}} \tag{4.3}$$

C (x,y,z) [mg/m^3] denotes the pollutant concentration at location (x,y,z) . Q is the source emission rate [kg/s]. μ is the average wind speed at the effective height of the stack (H). σ_y is the horizontal dispersion coefficient, and σ_z is the vertical dispersion coefficient. This equation gives the time-averaged concentration of an air pollutant at any spatial location downwind of a source. We have

also considered constant wind speed and direction while doing the simulations. The first term to the right of the equal sign provides the concentration directly at the centerline of the plume (coordinates x,0,H); the second factor adjusts the concentration in the vertical (z) direction; the third factor adjusts the concentration as we move in the sideways (y) direction. This simplified solution will be adjusted to account for the boundary condition at the ground (z=0) in Section 4.5.

4.3.1 Effective Stack Height

The Gaussian plume model is applicable for downwind distance, x > 100 m [3] because near the source the concentration approaches infinity. H_e is defined as the effective stack height. In any consideration of the concentration in the downwind direction, it is important to estimate the effective stack height [11].

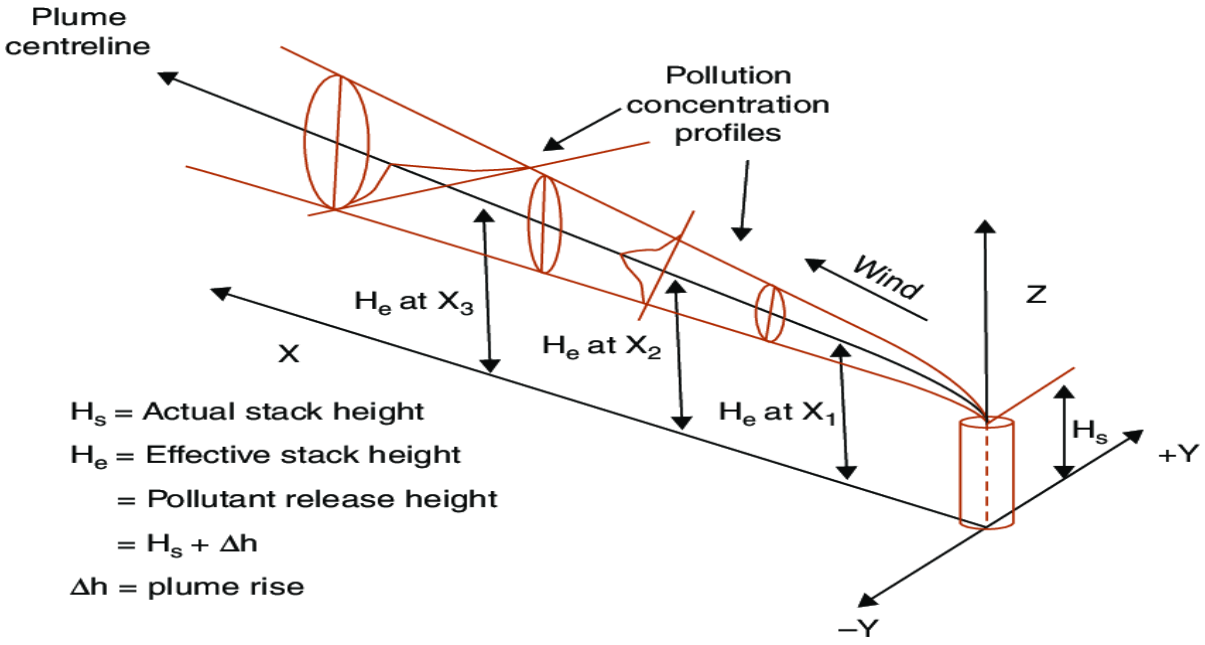

Figure 4.3: Effective Stack Height. Reproduced from [22]

At the effective stack height, the plume becomes level. The effective stack height is derived using the following equation:

$$H = h + \Delta H \tag{4.4}$$

In equation 4.4, Δ H is the plume rise [m]. The plume rise is a complex term and is derived using the following mathematical equation:

$$\Delta H = \frac{V_s d}{\mu}[1.5 + (0.0268P\left(\frac{T_s - T_a}{T_s}\right)d] \tag{4.5}$$

Vs in Equation 4.5 is the stack velocity (m/s) defined as the velocity by which the plume rises above the stack. d is the stack diameter [m], it is usually measured across the opening of the stack [13]. μ is the wind speed [m/s] of the atmosphere. The term Ts is the stack temperature [K]. Ta represents the air temperature [K] of the atmosphere. The atmospheric pressure is denoted by P [kPa]. The effective stack height calculations for a point or area such as forest fires or vehicle emissions would have some inaccuracies due to the high number of variations in topography and elevation changes.

4.3.2 Concentration Dispersion Directions for Gaussian Plume Model Equation

Part (i) of Equation 4.3 is the concentration at the centerline of the plume, Part(ii) is the concentration values as we move in the vertical direction (±z), and part (iii) denotes the concentration values as we move sideways across (±y)-direction. The concentrations are often symmetric about the y-axis and z-axis.

4.3.3 Dispersion Coefficient

Many researchers imposed a lower limit on x and σ_y or an upper limit on σ_z near the source [1]. The dispersion coefficient σ specifies the spread of the plume. This dispersion coefficient σ varies with atmospheric conditions and is explained in more detail in the following subsection 4.3.4. In most of the scenarios, 67% of the pollutant is assumed to be within ±σ from the centerline of the plume. Thus, the plume might be defined as four to six times wider than the dispersion coefficient σ [1]. Here σ is dependent on the magnitude of the turbulence in the atmosphere. Higher eddy diffusion and σ will be observed when the atmosphere is unstable. A stable atmosphere will result in lower eddies and σ. Measurements of σ used virtually in almost all models are based on Turners [1]. These values are usually taken in open and rural surroundings. Because of their origin, these values are appropriate for dispersion estimates in rural settings but less for urban areas. Higher surface roughness and higher release of heat at the surface imply that atmospheric conditions in urban areas are seldom as stable as in rural areas [8].

4.3.4 Pasquill-Turner stability classes

Pasquill-Turner stability classes describe the physical parameters of the Gaussian plume model based on the stability categories. The atmosphere is generally described as being in six stability classes, labeled A through F. Classes A through C are unstable conditions, class D is neutral, and classes E and F are stable. The most frequently observed classes are C, D, and E.

The stability classes and the values for stability parameters as explained in [1] and shown in the Table 4.1:

Table 4.1: Atmosphere stability classes under Pasquill[3]

Wind velocity on height at 10m(m/s)	Daytime incoming solar radiation			Night cloudiness	
	Strong	Moderate	Weak	Clouded	Cloud less
<2	A	A-B	B	E	F
2-3	A-B	B	C	E	F
3-5	B	B-C	C	D	E
5-6	C	C-D	D	D	D
>6	C	D	D	D	D

Table 4.2: Values of stability parameters a and b in the function for σ_y

Stability Class	a	b
A	0.28	0.9
B	0.23	0.85
C	0.22	0.8
D	0.2	0.76
E	0.15	0.73
F	0. 12	0.67

Table 4.3: Values of stability parameters a and b in the function for σ_z

Stability Class	P	10^p	q
A	-0.27819	0.527	0.865
B	-0.43063	0.371	0.866
C	-0.67985	0.209	0.897
D	-0.89279	0.128	0.905
E	-1.00877	0.098	0.902
F	-1.18709	0.065	0.902

There have been several approximations of the stability condition and parameter values [9] depending on the vegetation, topography, and the application of the model. We have selected Table 4.3 because it is very close to conditions for defining forest fire and vehicle emissions.

The information regarding the horizontal and vertical dispersion coefficients derived by Pasquill – Turner [2] can be depicted using the figures below:

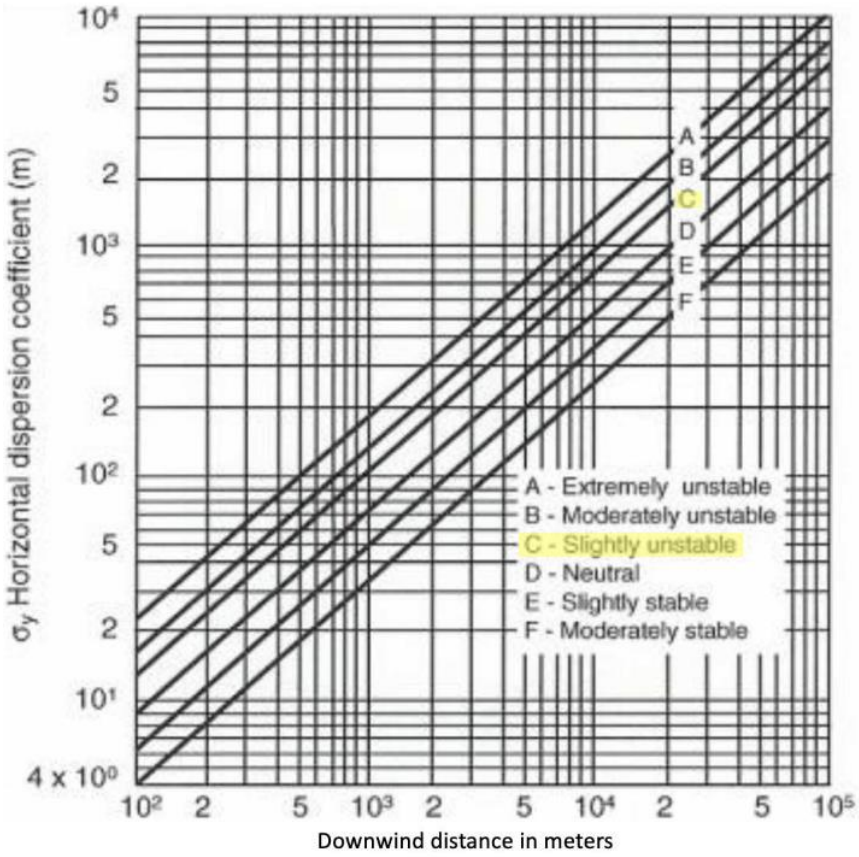

Figure 4.4: Horizontal dispersion coefficient σ_y as a function of downwind distance from the source for various stability categories. Reproduced from [5].

The Pasquill-Turner stability table is considered as a reliable resource to estimate the stability conditions or classes. This stability table is also recognized by the air resources laboratory [19].

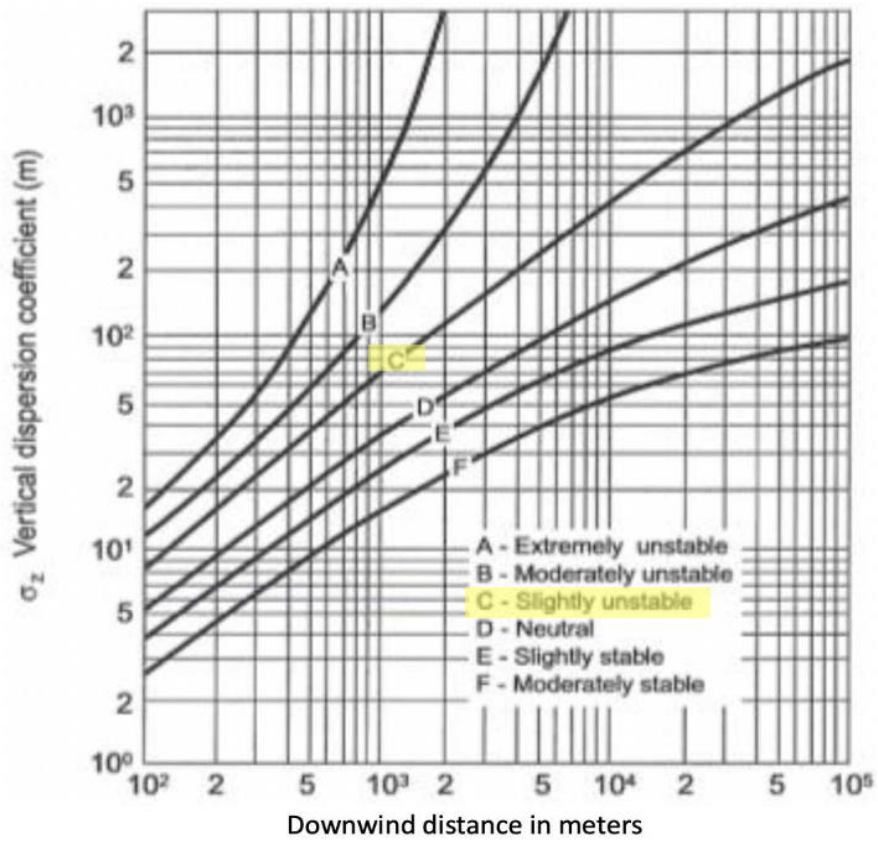

Figure 4.5: Horizontal dispersion coefficient $\sigma_z$as a function of downwind distance from the source for various stability categories. Reproduced from [5].

The parameters are calculated using empirical functions. The functions are mentioned below:

$$\sigma_z = K(z_0)ax^b \tag{4.6}$$

$$\sigma_y = K(z_0)10^p ax^q \tag{4.7}$$

The values of parameters a,b,p, and q can be defined from Tables 4.2 and 4.3 depending upon the atmospheric stability conditions. These parameters are dimensionless constants [5]. The roughness length K(z0) is calculated using the formula:

$$K(z_0) = (10z_0)^{0.53x-0.22} \tag{4.8}$$

4.4 POLLUTANT SOURCE AND SENSOR RELATION

For the practical implementation, we structured the model with the source and the sensors on a field as shown in Figure 4.6. It was assumed in this model that the contaminant is emitted from the source at a constant rate Q [kg/s]. For applications such as forest fire or vehicle emission, we consider the effective stack height as zero. For the forward model, the output is the pollutant concentration values [g/m3] at any location on the field.

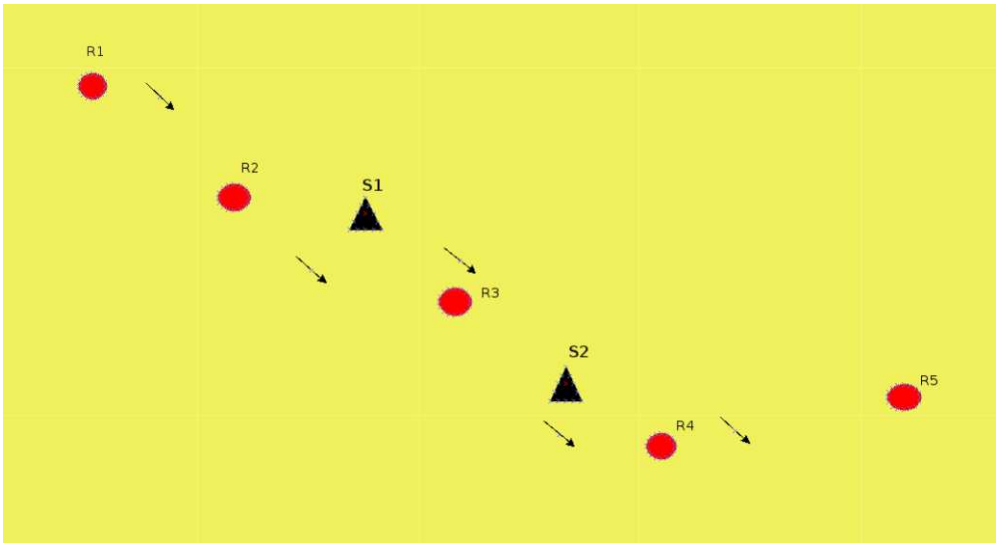

Figure 4.6: Source-Sensor field

Figure 4.6 shows a source-sensor model which has two pollutant sources and five sensors dispersed near the sources in a rectangular field. Multiple sensors can be deployed randomly across the field. The sensors are labeled R1-R5 and the source is labeled S1-S2. The arrow depicts the direction of the wind. There is research [10][11] conducted on wind speed estimation while analyzing pollutant dispersion. But, for an initial simple air pollutant monitoring system we have considered the wind speed and direction to be constant. In this work, we have considered a default wind speed of 5 m/s as suggested in [2], although this wind speed is varied for some studies presented in later sections.

A general plume model would consider the source stack height. We have designed the model in consideration of a point source similar to a forest fire. In this example, it would be appropriate to consider the stack height to be zero due to the varying elevations and topography of the forest. Thus, to simplify the model the effective height H [m] is considered zero. We make this assumption that when there is a wildfire or emission from a vehicle the smoke would start dispersing into the atmosphere very close to the ground. On the other hand, if the pollutant emission source is an industrial chimney it would be appropriate to consider the plume rise using equation 3.4. The plume rise also depends on the mass density of the contaminant. If the contaminant mass density is higher than air they settle out of the atmosphere at a higher rate [13].

As depicted in Figure 4.6 we have multiple sources $S_k$each with an emission rate Q_k[kg/s] at locations L_k, for two sources k = 1,2. Similarly, we have placed multiple sensors R_r at five locations in the field, for r = 1,2,3,4,5. To simplify the simulation, we have only considered constant wind. Since we have multiple sources in the equation, we would sum the contributions of each source by iterating over k from 1 to the total number of sources, and then calculate the total concentration based on the individual contributions. The Gaussian plume equation for the above field after defining the physical parameters and equation 3.3 can be modified as:

$$C(x,y,z) = \frac{Q_k}{2\pi\sigma_x\sigma_y}\sum_{k=1}^{+2} e^{\left(\frac{-(z)^2}{2\sigma_z^2}\right)} e^{\left(\frac{-(y)^2}{2\sigma_y^2}\right)} \qquad (4.9)$$

The pollutant concentration data for such a field setting is plotted in Section 5.2. It can be noted that forward simulation using Equation 4.9 would be extremely useful for fire-prone areas or for estimating the effect of toxic emissions in residential areas. The simulation is generated using specific emission rates.

4.5 EFFECT OF SETTLING AND DEPOSITION VELOCITY

The settling and deposition velocity has a significant effect on measuring pollutant concentrations. The problem of atmospheric pollutant transport in which gravitational settling and ground absorption cannot be ignored has been discussed by Calder [25].

4.5.1 Settling Velocity

In many applications of this model, the pollutant particles tend to be heavier than air. This makes the pollutants settle down at a well-known rate called the settling velocity W_{set} [m/s] [2]. The settling velocity – also called the fall velocity – is one of the key parameters to understand by studying sediment transport, mixing, and the exchange process. The settling velocity is usually greater than the gravitational settling velocity. The settling velocity depends on the particle radius R [m], gravitation acceleration g [m/s2], and the dynamic viscosity of air μ [kg/ms]. The settling velocity for spherical particles with uniform size can be derived using the Stokes law. The Stokes equation for settling velocity can be explained below:

$$W_{set} = 2pgR^2/(9\mu) \qquad (4.10)$$

4.5.2 Deposition Velocity

In addition to the settling velocity, we need to consider the flux situation at the ground surface. This equation considers both the gravitational settling flux and ground deposition flux. Research in [17] suggests when the particles reach the ground surface they deposit on the ground and are absorbed. The vertical flux at the surface for the contaminant particle is proportional to the surface concentration [2] and is derived as:

$$(k\frac{\sigma c}{\sigma z} + W_{set^c})|_{z=0} = W_{dep^c}|_{z=0} \quad (4.11)$$

This constant (K) is related to the diffusion coefficient of the contaminant particles in the atmosphere. It quantifies the rate at which the particles disperse vertically due to diffusion. The term W_{dep} [m/s] is the deposition velocity at the ground surface.

In the next section, we substitute the values for $W_{\boldsymbol{dep}}$ and W_{set} in the Equation 4.13 but, it becomes more complex. For simplicity and the problem requirement, we will be using just Equation 4.9 since we are setting $W_{\boldsymbol{dep}}$ and W_{set}= 0

CHAPTER 5

FORWARD MODEL SIMULATION AND RESULTS

5.1 INTRODUCTION

Our work on air pollution dispersion modeling is in agreement with several existing models, and we have further validated our results by comparing them to those from Stockie's study [4]. The are several parameters such as in the Gaussian plume model discussed in Chapter 4 which are used for solving the equation for estimating the pollutant concentration. Additionally, there are other parameters considered during simulations which are discussed in Subsection 5.1.1.

5.1.1 Other parameters considerations

The settling velocity and deposition velocity are considered to be zero. The effective stack height was also approximated as zero. We ran the simulation from a single source to two sources with varying source emission rates. For simulations, we have used Matlab programming due to the availability of useful tools such as contour() for plotting data in contour form, meshgrid() to return 2-D grid coordinates based on the coordinates contained in two different vectors, and the Gaussian plume function named gplume which returns the steady-state Gaussian function of a single continuous point source emitting at a rate of Q grams per second. The developed Matlab code for the model is mentioned in the appendix.

Multiple parameters used as inputs to the Gaussian plume model are summarized in Table 5.1:

Table 5.1: Table for different changing parameters in gaussian plume model

Parameter	**Range**	**Unit**
Wind speed	5-20	m/s
Wind angle	20-60	degrees
Source emission rate	100-1000	kg/sec
Source location	0- infinite	meters
Sensor location (distance from the source)	0 - 1500	meters
Atmospheric stability	Pasquil-Turner stability Class C	unitless

5.2 SENSOR CONCENTRATION RESULTS

5.2.1 Forward simulation: Single source

The Matlab simulation results in this section are observed using a single source and measuring the pollutant concentration across a field. The sensor locations are plotted on the field where the output concentration value is desired. For this simulation, we have modeled a field with the x-axis and y-axis. The distance on both axes is in meters. The x-axis on the field ranges from 0 [m] to 1500 [m] whereas the y-axis ranges from 0 [m] to 500 [m]. An example field is displayed in Figure 5.1. The source emission rates, source locations, wind speed, and angle are defined before generating the concentration plots in Matlab. The Gaussian plume equation is defined in Matlab which calculates the concentration at each point defined in the program.

For this specific simulation we have located the source and the sensors at the following locations on the field:

Source Location S1 :

X = 10, Y = 0, Z= 0.

Sensor Location R1 (Measurements displayed at) :

X = 250, Y = 0, Z= 0.

The source emission rate considered was 50 kg/s whereas the wind speed was 5 m/s. The σ_y value was 14.8415 at downwind distance x = 100 m. The sensor location R1 can be shifted to any location or more sensors can be added to the field to display the pollutant deposition concentration at any location.

The field and the pollutant plume concentration profile obtained across the field are depicted in Figure 5.1.

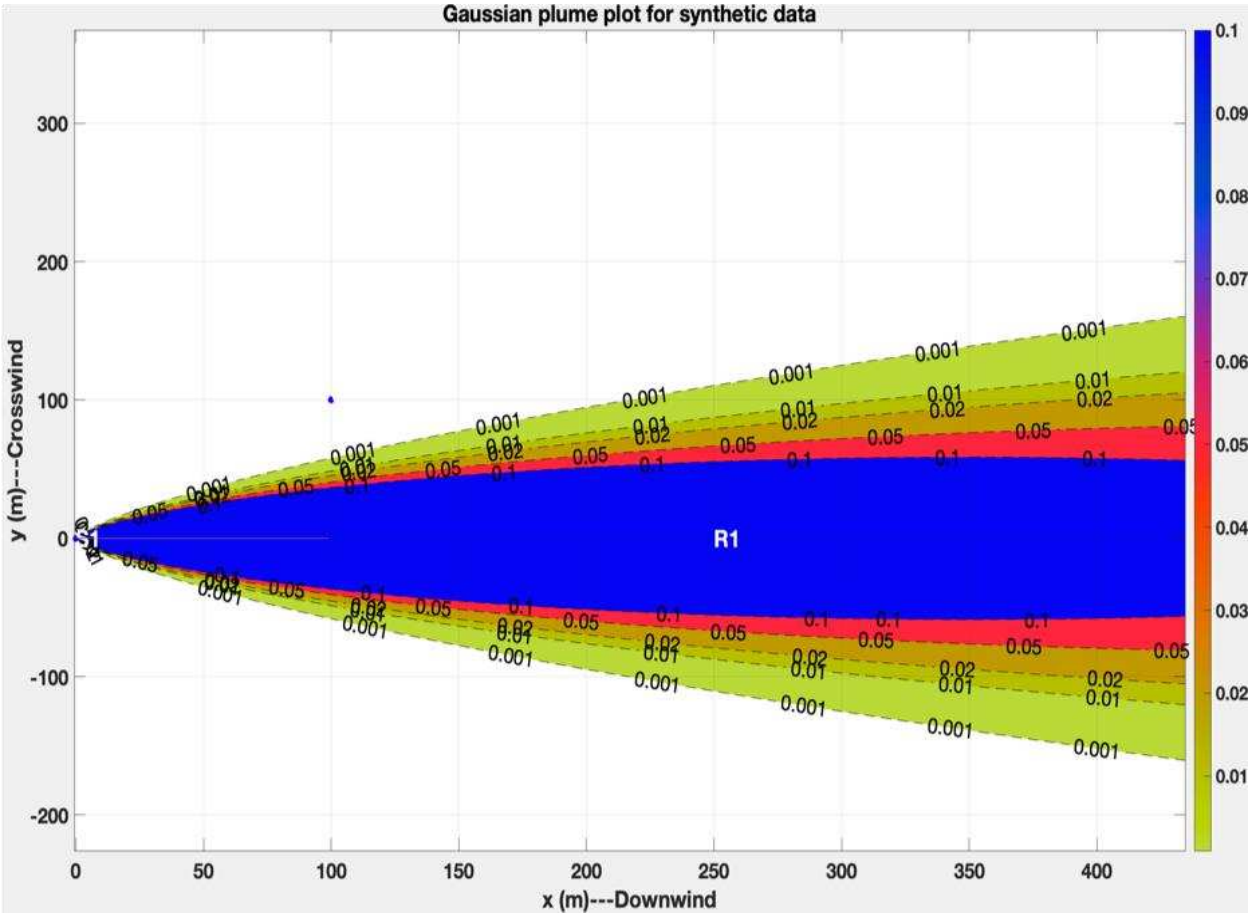

Figure 5.1: Forward simulation contour plot for plume with one source S1 located at the origin.

Figure 5.1 is a top view of the x and y planes of the plume. It can be observed that the plume becomes broader in the [$\pm y$] direction as we move further downwind in [$+x$] direction from source S1. As seen in the contour plot of Figure 5.1 the concentration varies from 0.05 [mg/m^3] close to the source S1 to 0.001 [mg/m^3] at the shaded edge of the plume. The results of the simulation for the concentration of pollutants are discussed below:

The concentration value obtained at Sensor R1 = 0.005109 mg/m^3

Theoretical value obtained using Gaussian plume solution = 0.005109 mg/m^3

Accuracy = 100 %

The accuracy of the source emission rate is calculated based on the ratio between the actual source emission rate and the model output source emission rate.

5.2.2 Forward simulation: Multiple sources

In this subsection, we have simulated the Gaussian plume solution using two pollutant-emitting sources. The concentration profile obtained is depicted in Figure 5.2.

The source location and the sensor locations in Figure 5.2 are as follows:

Source Location S1 : X = 110, Y = 0, Z= 0.

Source Location S2 : X = 350, Y = 270, Z= 0.

Sensor Location R1: X = 240, Y = 0, Z= 0.

Sensor Location R2: X = 1100, Y = 100, Z= 0

Source emission rates Q1 = 100 kg/s , **Q2** = 150 kg/s

Wind speed $\boldsymbol{\mu}$ = 5 m/s , $\boldsymbol{\sigma_y}$ for S1= 55.5436, $\boldsymbol{\sigma_y}$ for S2= 14.8415

Figure 5.2 shows example concentration results at ground height (z=0) from a top view (in the x-y plane) due to the emission from two sources S1 and S2.

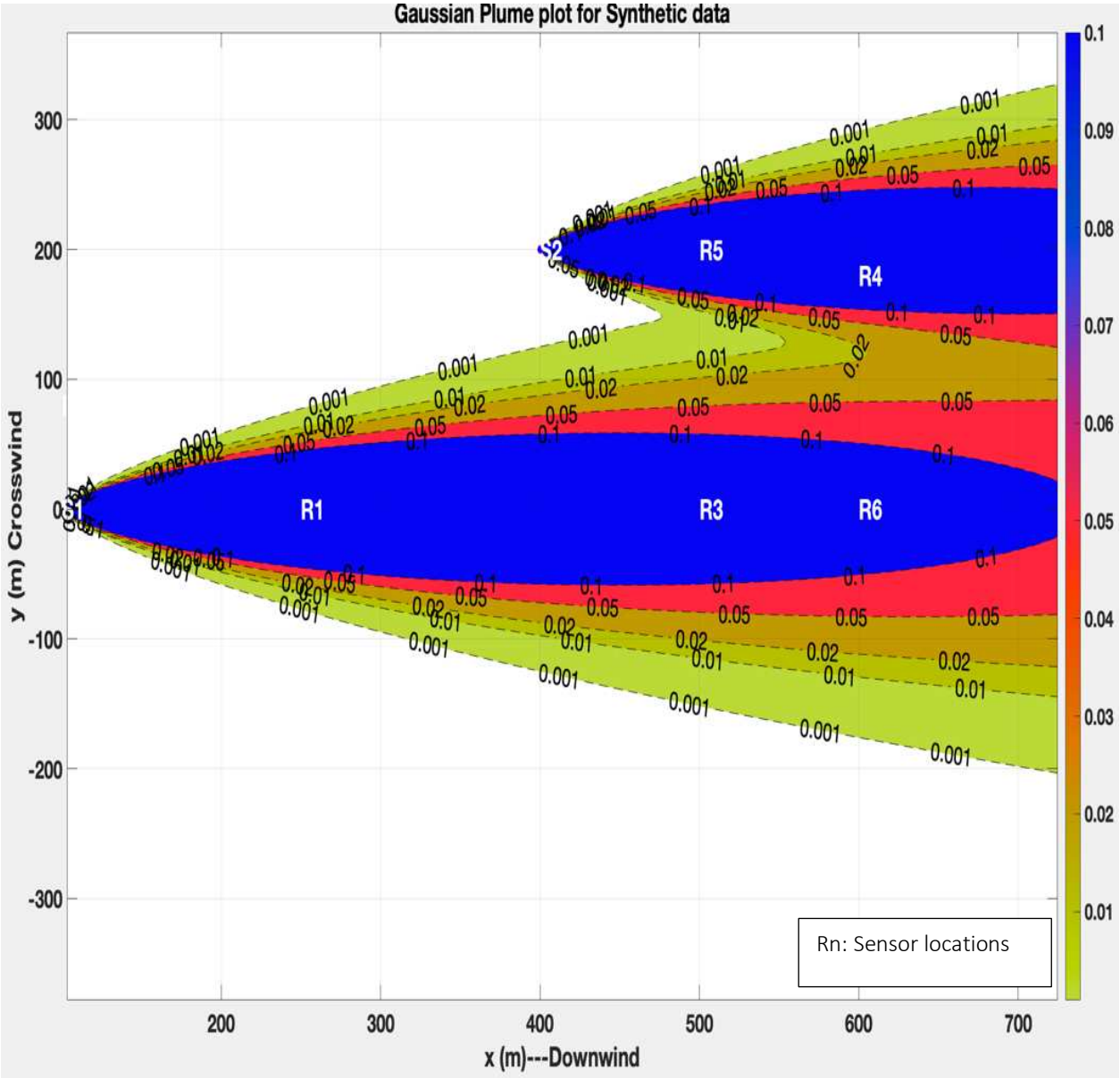

Figure 5.2: Forward simulation contour plot for plume with two sources S1 and S2

It can be observed from Figure 5.2 that the concentration profiles or the plumes overlap for the pollutants from S1 and S2. The concentration results obtained for the sensors in the fields are mentioned below:

The concentration value obtained at sensor R1 = 0.2026410 mg/m^3 Theoretical value obtained using Gaussian plume solution calculated using equation 4.13 is 0.201613 mg/m^3 Accuracy obtained = 99 % .

The concentration value obtained at sensor R2 = 0.018790802835567 mg/m^3 Theoretical value obtained using Gaussian plume solution = 0.018231 mg/m^3 Accuracy obtained = 98 %

The accuracy results obtained were calculated using the difference between the model output and the theoretical value calculated using gaussian plume equation. This accuracy is important for the reverse calculation to understand the forward model accuracy and the Matlab code validation.

5.3 CONCLUSION

From the concentration profile obtained in Figure 5.1 and Figure 5.2, it can be concluded that the highest ground level concentration is near the source along the x-axis. This is because the plume spreads out more in the y and z directions as it moves away from the source, leading to lower concentrations of pollutants in those directions. However, it's important to note that the y and z axes can also have significant effects on the dispersion of pollutants in the atmosphere. In particular, the wind direction and atmospheric stability play a crucial role in determining the extent of lateral and vertical dispersion. The concentration goes on decreasing as we move away from the centerline of the plume. The results replicate the solution we obtained for the Gaussian plume in Section 4.3. But it is very important to understand that when such simulations are used in a

practical real-time data model, several important factors are to be considered. The factors include the mode of communication between the sensor node and the central node [10], the elevation at which the sensor is deployed, and the assembly of the overall hardware systems. Thus, simulating static data might seem to be a simple solution but using a Gaussian plume solution for the simulation of real-time data might become very complex.

CHAPTER 6

INVERSE GAUSSIAN PLUME MODEL

6.1 INTRODUCTION

A method to estimate the location of the pollutant source and emission rate from sensor measurements is developed using the Gaussian plume solution. The goal of the inverse model is to estimate the unknown source emission rate and source locations given the sensor concentrations. The inverse model relies on the forward model, to evaluate how well an estimated source emission rate and location predict the corresponding sensor concentrations. This inverse method has been developed using linear least square minimization and gradient descent methods to minimize the error between the true and the predicted pollutant concentrations at the sensors.

6.2. MINIMIZATION

Minimization refers to the process of finding the values or parameters that minimize a given objective function. It is a fundamental concept in optimization, where the goal is to find the optimal solution that yields the smallest value of the objective function [18].

In the context of optimization, minimization involves searching through a space of possible solutions and adjusting the values of the parameters or variables to find the set of values that leads to the minimum value of the objective function.

The objective function represents the quantity that needs to be minimized. It can take various forms depending on the problem at hand, such as a cost function, a loss function, an error metric, or any other measure that captures the quality or performance of the solution. The minimization process typically involves an optimization algorithm that iteratively explores the

solution space and adjusts the values of the parameters to decrease the objective function value [20]. The algorithm evaluates the objective function at each iteration and updates the parameters based on a set of rules or search strategies. The goal is to converge to a solution that provides the smallest possible value of the objective function.

In this model, we needed to optimize a solution for estimating the source emission rates. To optimize the solution, we used the linear least square minimization method which is discussed in the next Subsection 6.2.1.

6.2.1 Linear Least square minimization

Linear least squares minimization is a technique used to estimate the unknown parameters of a linear model by minimizing the sum of squared differences between the observed data points and the corresponding values predicted by the linear model. In linear least squares, the goal is to find the best-fitting linear relationship between the independent variables (inputs) and the dependent variables (outputs). The linear model is defined as a linear combination of independent variables, weighted by unknown coefficients or parameters. The method involves constructing an objective function that quantifies the discrepancy between the observed data points and the predicted values from the linear model. The most common objective function is the sum of squared residuals, which measures the squared differences between the observed and predicted values.

In the Inverse model, we use linear least square minimization to solve the values for source emission rate(s). The equation we trying to minimize is as follows:

$$\vec{R} = (\text{GLSQ}) \text{ X } (\vec{Q}) \qquad (6.1)$$

In Equation 6.1, $\vec{R}$ and $\vec{Q}$ are the vectors containing the sensor concentrations and source emission rates respectively, and GLSQ is an N_r x N_s matrix whose r, s corresponding values in the matrix are calculated using Equation 4.9. N_r is the number of receptors and N_s is the number of sources. Since there are typically more receptor measurements than sources ($N_r > N_s$), this is an overdetermined system of equations; hence, we try to obtain an approximate value of the solution.

We use the linear least square solver function named lsqlin() in Matlab to solve Equation 6.1 in all the computations for estimating the value of source emission rate(s) in the inverse model.

6.3 NEED FOR SOURCE TERM ESTIMATION

The Gaussian plume model is very useful for estimating pollutant concentrations but in practical application, it is required to estimate the source of emission using the concentration measurements. In this work, we used the Gaussian plume model to find the concentrations that would be measured by a sensor network and thus, generate synthetic data for concentration measurements. Estimating the source location and emission rates using the synthetic data helps us to understand the accuracy of the method and verify the inverse problem algorithm prior to applying it to actual sensor data, such as the data from Purple Air particulate sensors. Source location is an important parameter of a pollutant source to recognize the distance of the source from the pollution-affected area and estimate the plume or pollutant spreading characteristics.

6.4 GENERAL EQUATION FOR INVERSE MODEL

As discussed in Section 5.1, the Gaussian plume equation originates from the advection-diffusion Equation 4.1 We use these equations for reverse calculations of the source emission rate and the source location when the concentration is known. A simplified equation for the advection-

diffusion for plume dispersion properties is derived by Ermak [6]. A detailed derivation of the Ermark's equation is explained in [4]. Solving Ermak's general equation for the source emission rate, Q yields the following equation:

$$Q = 2u\pi\sigma_x\sigma_y * C(x,y,z) * W_{dep} * e^{(((z-H)^2)/(2\sigma_z^2))} * e^{((y^2)/(2\sigma_y^2))} \quad (6.2)$$

In equation 6.2, x is the distance along the wind direction, with the source at x=0 (m), y is the cross-wind direction (m), z is the vertical height (m), H is the source height, Q is the contaminant emission rate (kg/s), u is wind velocity (m/s), and W_{dep} is the deposition velocity (m/s).

6.5 IMPLEMENTATION OF AN INVERSE MODEL

For implementing this inverse model, we have estimated the source locations and source emission rates through L2 minimization using gradient descent. It was important to implement the inverse model first using a single source with 3 unknown parameters (source emission rate and x and y locations) and analyze the accuracy before moving to the more complex problem with multiple sources and thus more unknown parameters. As an initial step, we generated sensor concentration data with one source and six sensors. The sensors were placed in random locations but within the spread of the Gaussian plume. This way each sensor concentration had some measurements and non-zero readings. To simplify the initial model, we considered the wind speed and the wind angle to be constant. A detailed flow chart of the inverse model is explained in the next section.

6.6 Procedure for testing inverse algorithm with synthetic data

The flow chart in Figure 6.1 Below illustrates the procedure for first generating synthetic data and then using it to test the inverse algorithm for estimating source parameters.

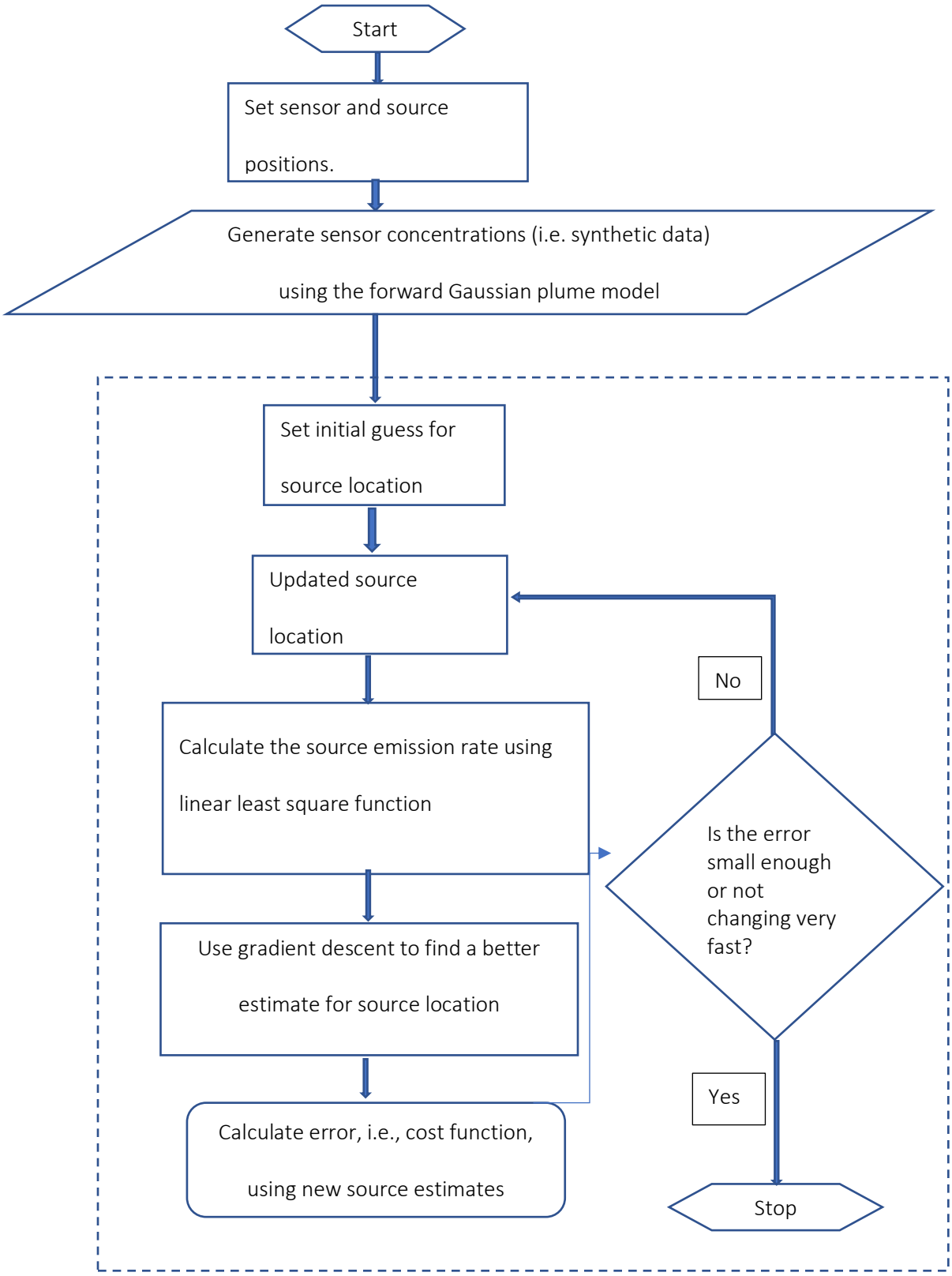

Figure 6.1 Model flow chart

Chapter 7

INVERSE MODEL SIMULATION RESULTS

7.1 INTRODUCTION

To evaluate the performance of the model, we started estimating the source parameters by setting up a simple model and then moving to more complicated scenarios by adding more sources and sensors, different wind speeds, sensor noise levels, and eventually applying the model to real particulate sensor data. When compared to the true source emission rates, the predicted source emission rates using synthetic data were observed to be 88% to 98% accurate, depending on varying parameter settings. Section 7.2 describes results with one or two sources and varying the sensor placements inside the plume. Section 7.3 presents the effect of sensor noise, and Section 7.4 addresses different wind velocities. Finally, in Section 7.5, data from a network of Purple Air sensors in the vicinity of a forest fire were used to determine the fire location and emission rate instead of relying on synthetic data with known source parameters.

7.2 MODEL RESULTS WITH CONSTANT WIND VELOCITY, AND NO SENSOR NOISE

The inverse model algorithm described in Chapter 6 was applied to synthetically generated sensor data based on one source and subsequently to two sources.

7.2.1. Single source and nine sensors

An example of Gaussian plume for one source and nine sensors setup is shown in the plot below:

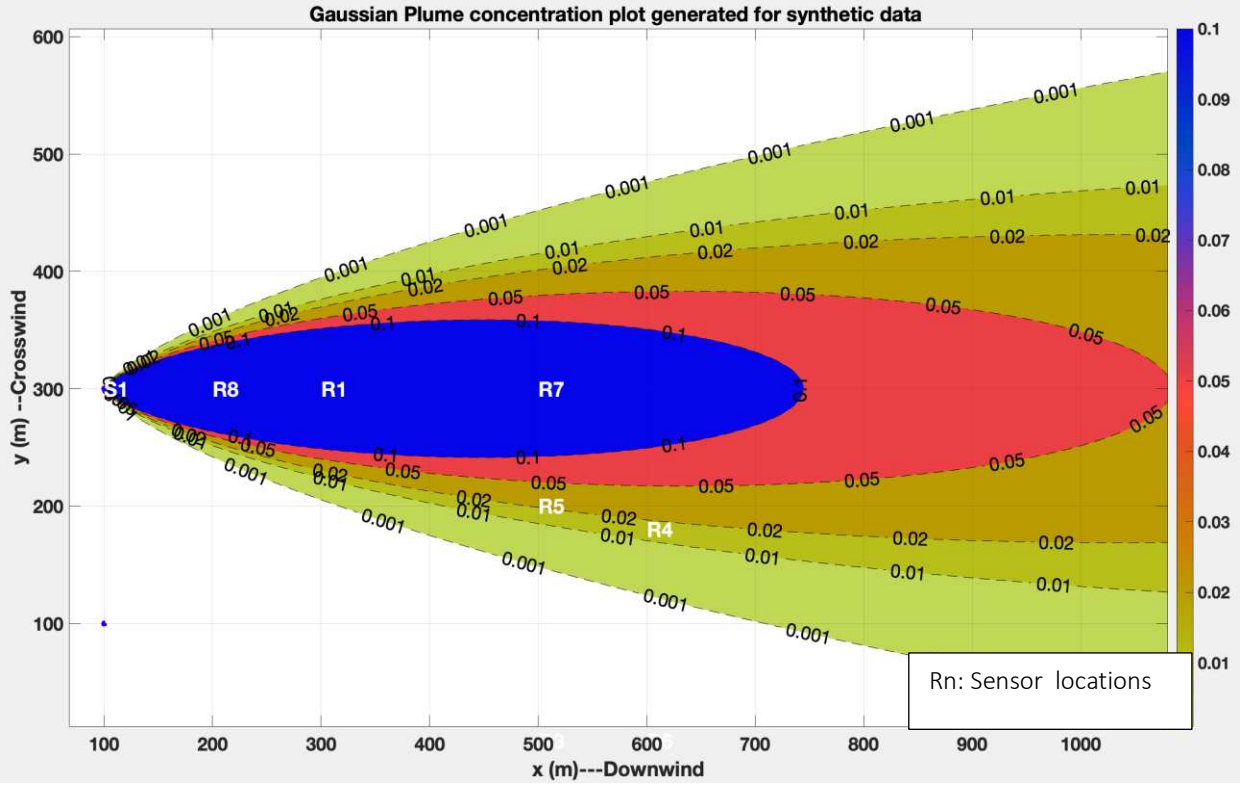

Figure 7.1 Single source and sensors set up in Matlab

For this simulation, all the sensor locations were set up to be in the Gaussian plume. Each sensor had non-zero measurements. Sensor R2 is the closest to the source in the downwind direction and has the highest measurements. Source S2 is turned off for this simulation. All the sensor locations and sensor measurements are used as the input to this inverse model simulation. For this simulation, there was no noise added to the sensor measurements with windspeed of 5 m/s and wind angle was 0 degrees.

It was observed the model performed very well with the predicted source location and source emission rate being within +/- 5% accuracy of the original values. The results for the source locations and source emission rate estimates through iterations of the gradient descent algorithm are depicted in the charts below:

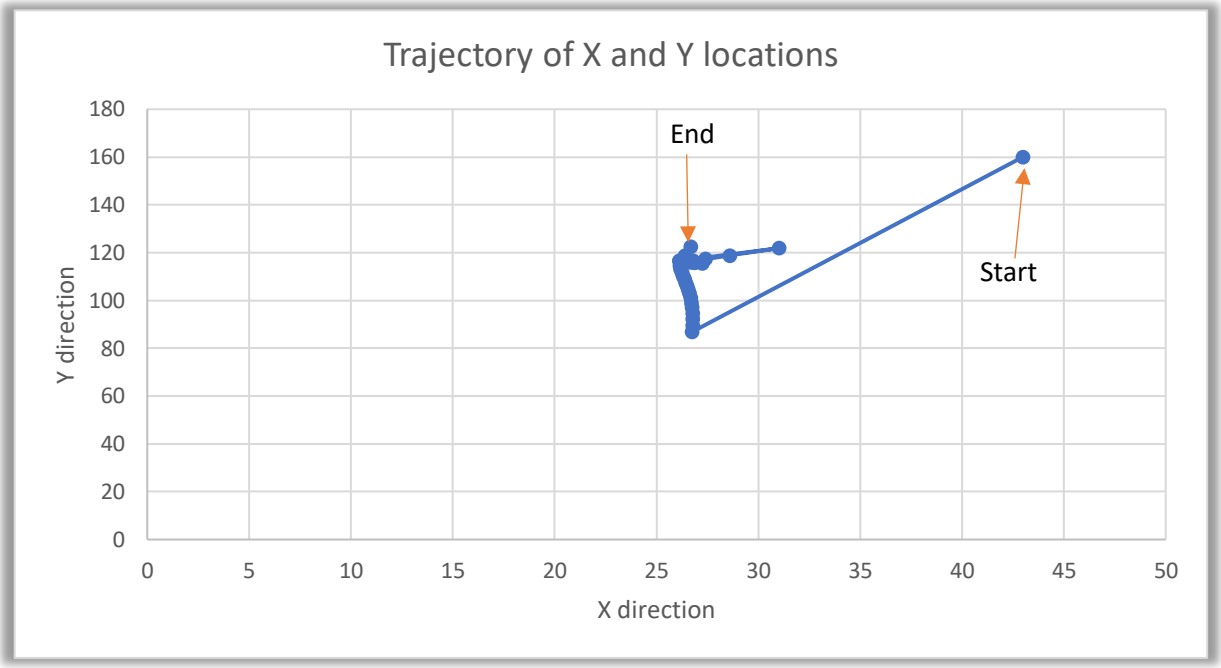

Figure 7.2 Trajectory plot for single-source X and Y locations

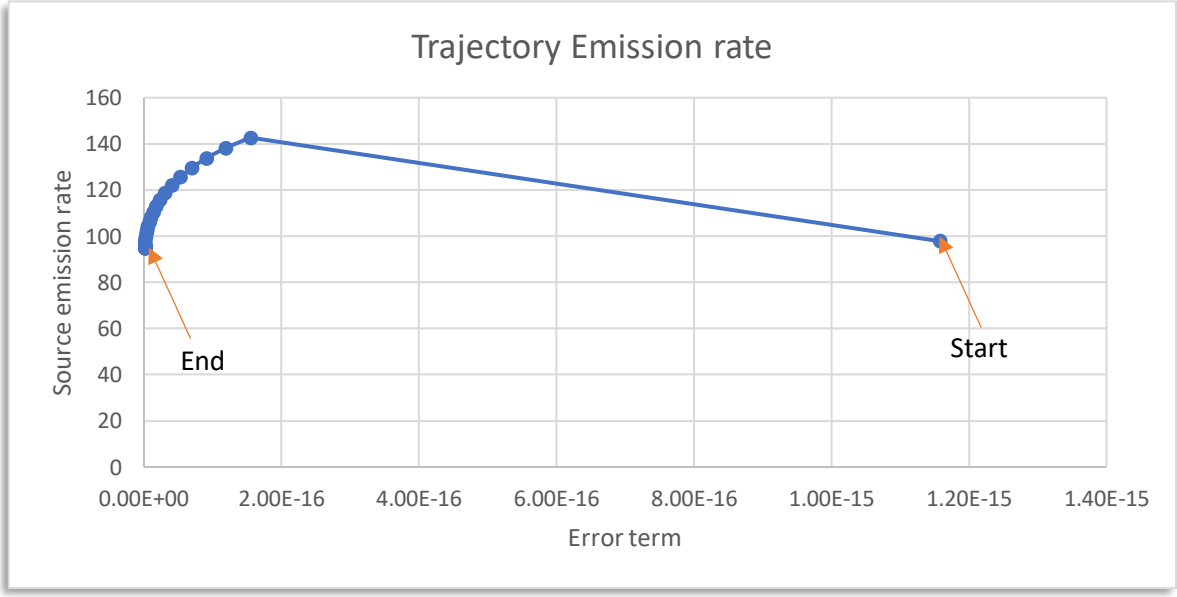

Figure 7.3 Trajectory plot for single-source emission rate

After minimizing the error term using L2 minimization via gradient descent the results summary were as below:

Table 7.1 Predicted source parameters for a single source and nine sensors

True X location: 30	True Y location: 120	True source emission rate: 90
Predicted X Location: 26.62	Predicted Y location: 116.13	Predicted source emission rate:95.01

The trajectory plots were as expected. As the error reduced the predictions seemed to be moving close to the true values of the source locations and emission rates. Also, it was observed that the model ran about 86 iterations to output the final results.

7.2.2. Two sources and nine sensors

For this simulation, synthetic data was generated using 2 sources with fixed emission rates. The measurements were taken at 9 sensor locations. Six parameters were predicted in this simulation which included the four coordinates for the two source locations (X1, Y1, X2, Y2) and the source emission rates (S1 and S2). The Gaussian plume generated for this simulation is shown below:

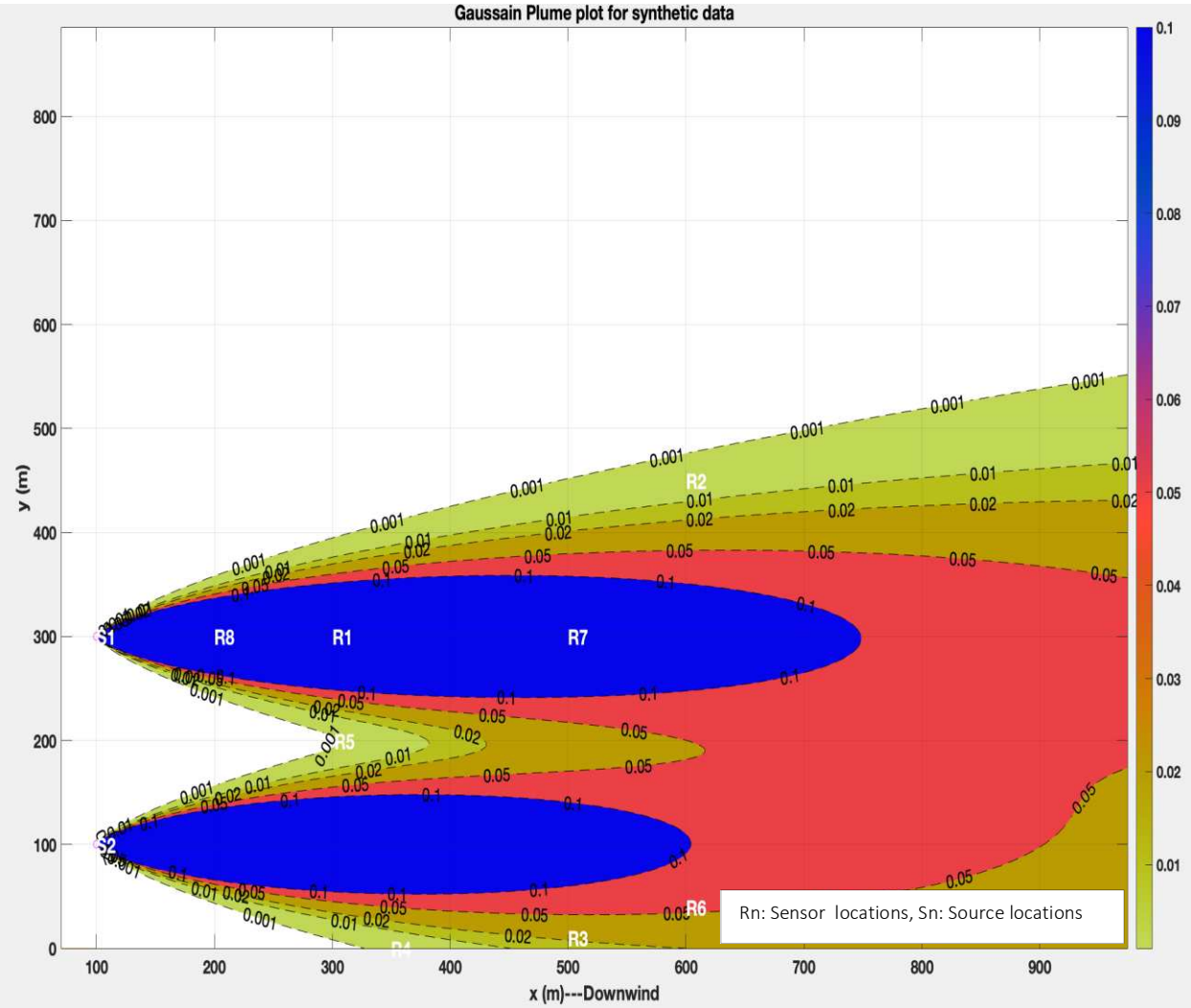

Figure 7.3 Two sources and sensors

7.2.2.1 Guess at true source locations

To check if the model is working correctly for two sources, we first entered the starting guess for both source locations at the true source locations (X1, Y1, X2, X2). The trajectory plots for this simulation with a guess at the true location are shown below:

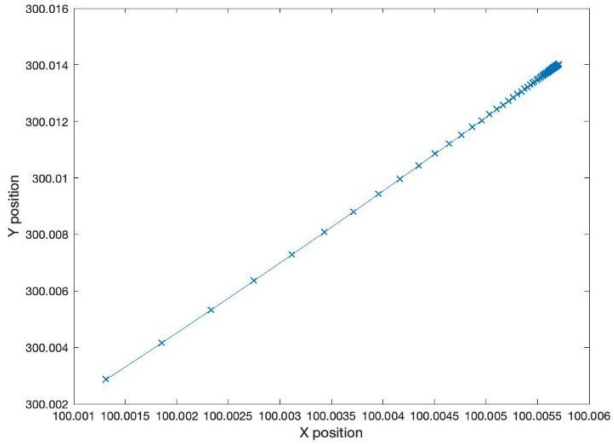

Figure 7.4 : Trajectory plot for the source location

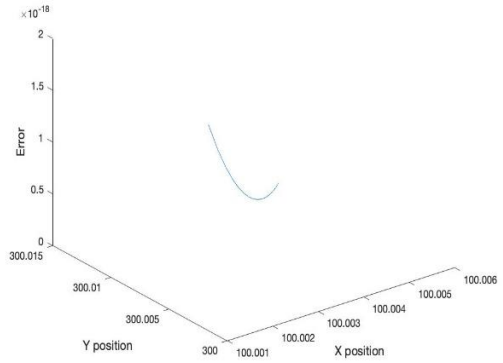

Figure7.5 Trajectory plot for the error term vs source (X,Y)

It can be observed that the model gradient trajectory does not move away from the initial guess and follows an expected path. The trajectory does not move a lot from true source locations as it learns the guess is at the true location. The error term was well minimized at the start of this simulation since the guess was at the true location.

7.2.2.2 Guess at random source locations

Now, when the model was verified to work correctly with a guess at the true location for both sources, we started simulating the model with a guess at random locations for both sources. The simulation with guess at location (10,10,10,10) for (X1, Y1, X2,Y2) is shown below:

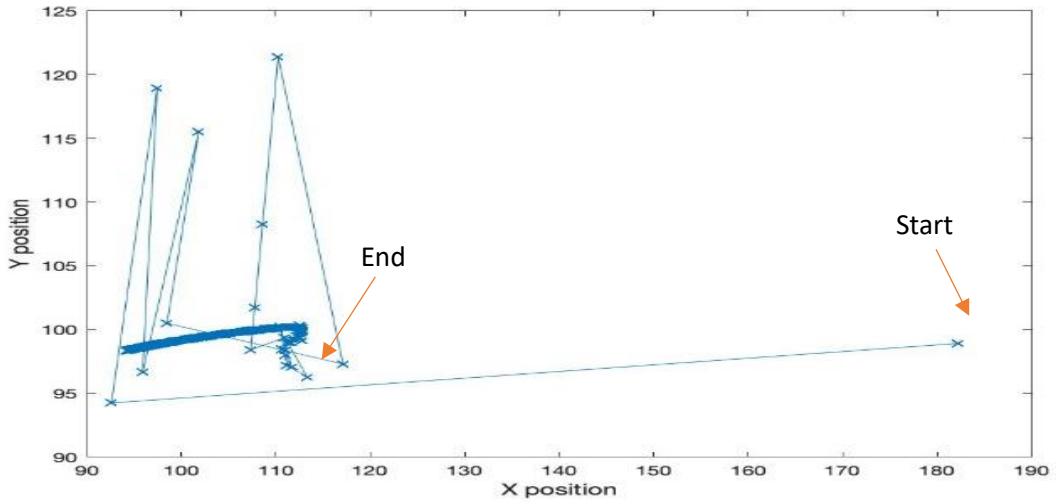

Figure 7.6: Trajectory plot for the source location

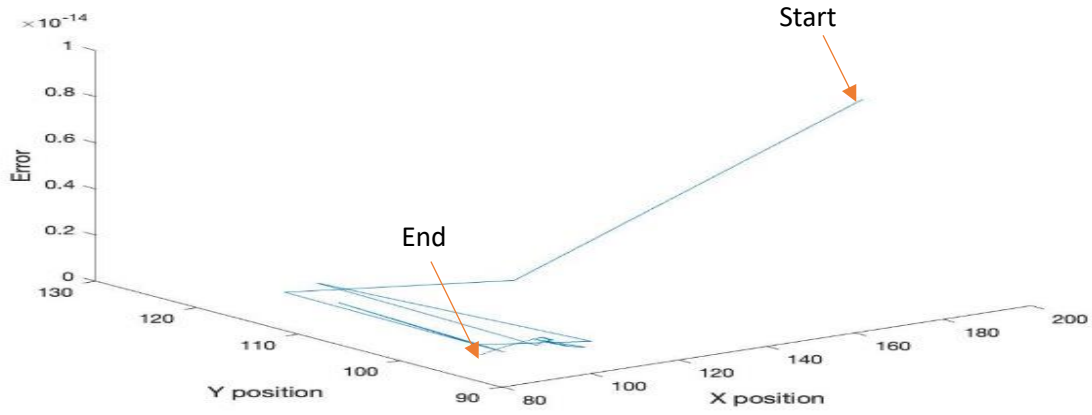

Figure 7.7: Trajectory plot for the error term vs source location

The summary of the results of both simulations is explained in Table 7.2.

Table 7.2 Predicted source parameters in inverse model for guess at random locations

Simulation	**True locations**	**Predicted locations**	**True emission Rates**	**Predicted emission Rates**
Guess at true location	X1= 100 Y1=300 X2=100 Y2=100	X1= 100.0057094 Y1= 300.0140198 X2= 99.98390003 Y2= 96.28822421	S1 = 150 S2 = 100	S1=150.38 S2 = 100.64
Guess at (10,10,10,10)	X1= 110 Y1=100 X2=100 Y2=100	X1= 114.629743 Y1=98.63294799 X2= 94.25957759 Y2= 98.35747551	S1 = 150 S2 = 100	S1 = 155.67 S2=97.56

It was observed that the accuracy of the model with one source was approximately 98% whereas, the accuracy of the model with two sources was approximately 96%. In the two sources model, the sensor measurements were subject to pollutants from two sources. Thus, in the minimization method for two sources the cost function had around 2-5% higher minimization error.

7.2.2.3 Results with varying sensor placements

To implement this model in a real-world wireless sensor network it is important to understand the effect of sensor placements in a field or area of interest. Next, we placed the sensors such that some sensors are covered by the plume whereas some sensors are not covered by the

plume. The sensors not covered by the plume had zero readings whereas the sensors covered by the plume had non-zero readings. A total of 9 sensors were set up in this simulation. The accuracy results for different configurations of the sensor locations are discussed in Table 7.3 below:

Table 7.3 Model accuracies based on the sensor placements

X inaccuracy	Y inaccuracy	Emission rate inaccuracy	Number of sensors inside the plume.
0.22%	0.11%	0.13%	9
2.46%	2.28%	2.20%	6
3.78%	3.40%	3.06%	4
6.29%	7.23%	7.89%	2

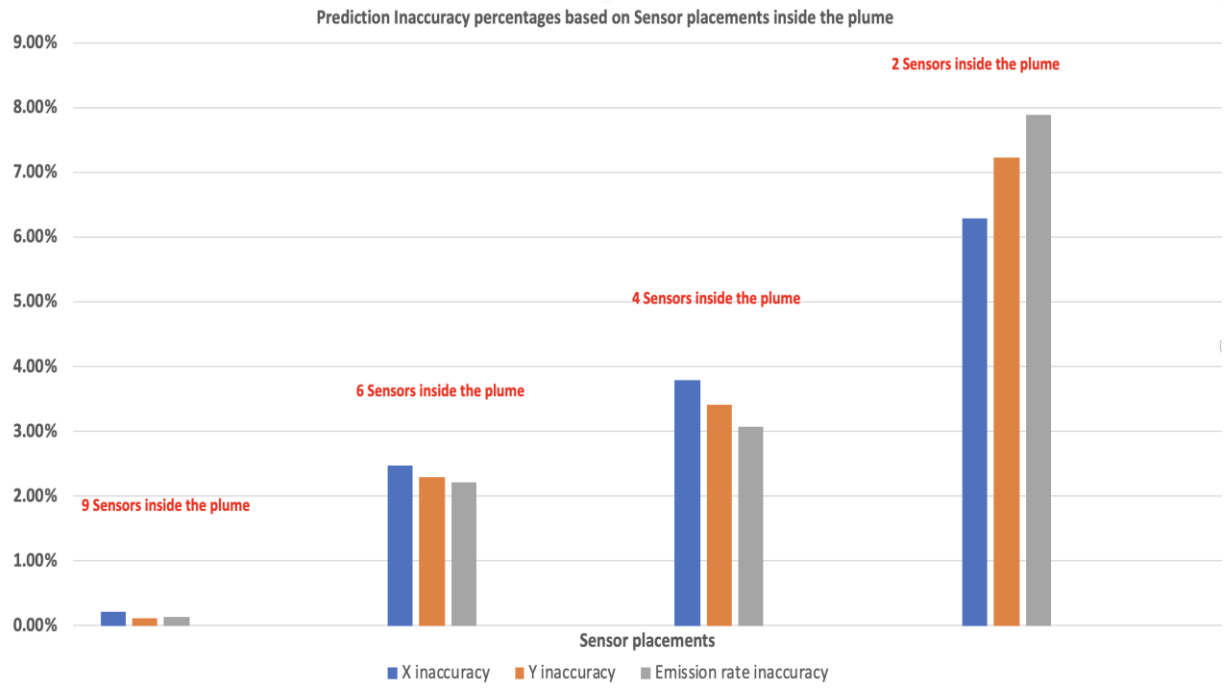

Figure 7.8: Prediction inaccuracies based on sensor placement

When all 9 sensors were placed in locations covered by the plume, the model achieved inaccuracy of 0.22% for X coordinate, inaccuracy of 0.11% for Y coordinate, and an emission rate inaccuracy of 0.13%. This indicates that the model accurately predicted the source locations and emission rate when all sensors were exposed to the plume.

The model's accuracy decreased slightly when only 6 out of 9 sensors were covered by the plume. The X inaccuracy was 2.46%, the Y inaccuracy was 2.28%, and the emission rate inaccuracy was 2.20%. Despite the decrease in accuracy, the model still performed reasonably well in estimating the source locations and emission rate.

Further reducing the number of sensors within the plume to 4 resulted in a noticeable decrease in accuracies. The X inaccuracy increased to 3.78%, the Y inaccuracy increased to 3.40%, and the emission rate inaccuracy increased to 3.06%. However, the model was still able to provide meaningful estimations with a reduced number of sensors.

The lowest number of sensors inside the plume in the experiments was 2 out of 9. In this scenario, the model's accuracy was compromised the most. The X inaccuracy was 6.29%, the Y inaccuracy was 7.23%, and the emission rate inaccuracy was 7.89%. Despite the lower accuracies, the model was able to predict the source locations and emission rate with 93.71% and 92.11% accuracy, respectively.

These results demonstrate that the placement of sensors in a wireless sensor network can significantly impact the accuracy of the model's predictions. However, even with a limited number of sensors exposed to the plume, the model can still provide meaningful estimations, although with reduced accuracy compared to scenarios with more sensors in the plume.

The physical inverse model does not exactly predict the source estimates with no noise data due to the slight inaccuracies in the linear least square estimates and the gradient descent algorithm. As discussed above in Figure 7.8 the sensor placements and the number of sources also affect the accuracy of the predictions. It was important to analyze these inaccuracies with different noise levels in data to understand the sensitivity of the model to noise which is discussed in the next Section.

7.3 INVERSE MODEL RESULTS WITH VARYING NOISE LEVELS

For a practical application of this model, the sensor concentration measurements will be susceptible to different noise levels from different contaminants and sources. It is important to evaluate the practical performance of this model by introducing noise. To evaluate the accuracy of the model with the noise we introduced Gaussian noise into the sensor readings.

Gaussian noise is a statistical noise that has a probability distribution function (PDF) equal to that of a normal distribution, which is also known as Gaussian distribution. In telecommunications and computer networking, communication channels can be affected by wideband Gaussian noise coming from many natural sources, such as the thermal vibrations of atoms in conductors (referred to as thermal noise or Johnson–Nyquist noise), shot noise, black-body radiation from the earth and other warm objects, and from celestial sources such as the Sun[13].

The Gaussian noise levels were varied for a number of simulations, and its effect on the model accuracy was studied. The noise levels refer to the Signal to Noise ratio (SNR). Signal-to-noise ratio (SNR or S/N) is a measure used in science and engineering that compares the level of the desired signal to the level of background noise. SNR is defined as the ratio of signal power to noise power, often expressed in decibels. A ratio higher than 1:1 (greater than 0 dB) indicates more signal than noise [14]. The effect of noise on the model was studied at SNR levels of -40 dB, -20 dB, 0 dB, 20 dB, and 40 dB The results of these experiments are summarized in the graph below:

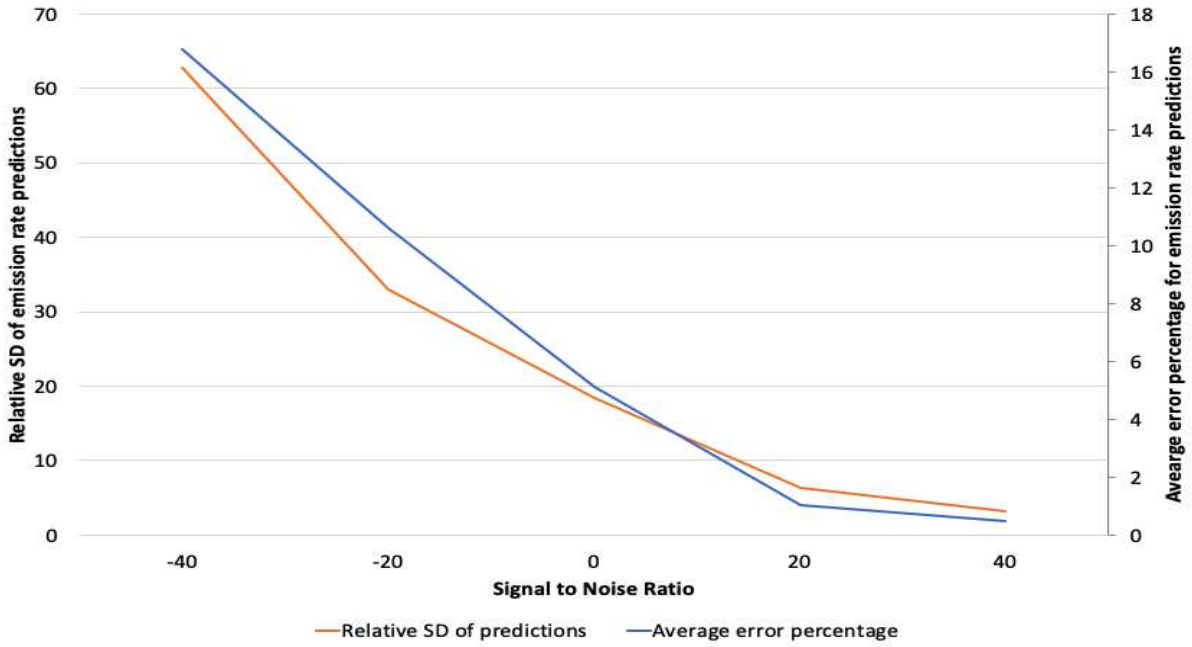

Figure 7.9: Effect of different noise levels on relative standard deviation of the predictions

In the figure above, we calculated the relative standard deviation of the predicted source emission rate. The relative standard deviation was performed on a data size of 100 readings. The formula used for relative standard deviation is as below:

Relative SD of predicted source = SD of predicted source emission rate /Average of predicted emission rate.

It can be observed that as the signal-to-noise ratio decreases the predicted source emission rate error and relative standard deviation increase. Thus, noise in the sensor readings has a substantial effect on the accuracy of the model. It can be observed that as the signal-to-noise ratio changes from 40 dB to -40 dB the error tends to increase about nine times.

7.4 INVERSE MODEL RESULTS WITH VARYING WIND SPEED, WIND ANGLE AND NOISE LEVELS

In the preceding portion of this chapter, we discussed results with fixed wind velocity and wind angle. But, in practical implementations, it is very important to consider the effect of different wind speeds and wind angles on the sensor measurements. Since the changes in wind speed and angle are in the downwind direction, changes in the wind speed will have an impact on the sensor readings since the wind carries all the particulates. The degree of effect of these parameters on the inverse model is discussed in this section.

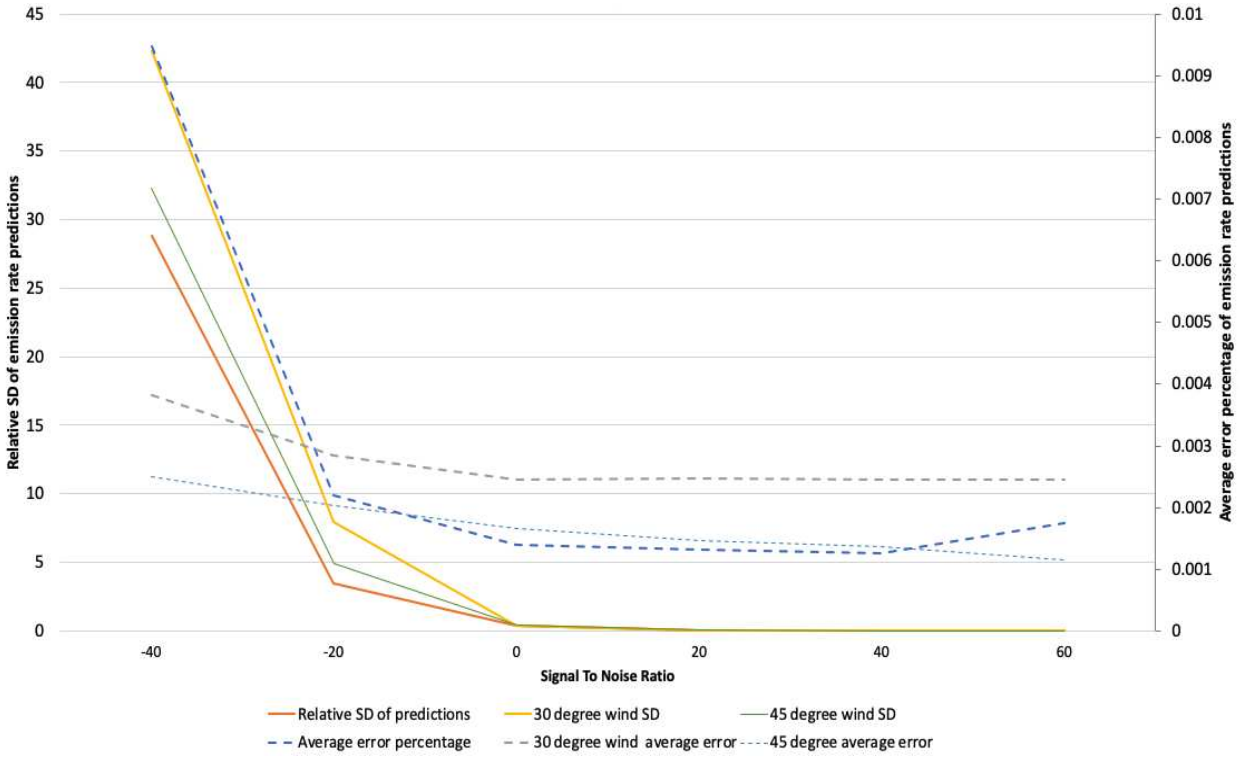

Figure 7.10: Effect of wind speed and noise levels on prediction error

In the above figure, we varied the level of noise and the wind angle. Varying the wind angle changes the number of sensor locations covered by the Gaussian plume. It can be observed that the relative standard deviation continues to increase as we increase the noise levels. The error remains fairly constant until 0 dB SNR and increases as we increase the noise.

7.5 APPLICATION OF INVERSE MODEL ON PURPLE AIR SENSORS DATA

We had access to the $PM_{2.5}$ air quality readings from the Purple Air website and performed an experiment to apply the inverse model to the purple air quality sensor data to find the source of pollutants. The experiment was performed to find the source of a forest fire in California. We chose the Woolsey fire in California which occurred from November 8^{th,} 2018 to November 15^{th,} 2018. The motive was to observe if the inverse model predicts the location of the Woolsey fire accurately or close to the actual location of the fire. We used readings from the sensors neighboring the area of the fire. The sensor locations are shown below in Figure 7.10.

Figure 7.11: Location of air quality sensors around the Woolsey Forest fire near Malibu. Reproduced from www2.purpleair.com.

While the map shown above from March 2019 contains dozens of sensors, only 4 of them had data from the period 1st November 2018 to 30th November 2018. The names of those 4 sensors are listed below, and the sensors were located at distances of 10 to 60 km from the fire.

1.Valley circle Vanowen, CA

2. CAZIER-SM-004, CA

3. Huntington Palisades, CA

4. Hacienda, CA

The sensors used in the simulation are indicated in red dotted circles on the map in Figure 7.10. The area with sensor arrangements was replicated in Matlab with a grid distribution scale.

We also researched the average wind speed during the period. It was reasonable to assume that the average wind speed was 23 m/s as suggested by climate.gov. The average wind speed was high as it was fueled by strong Santa Ana winds. The area with sensor arrangements was replicated in Matlab.

The gradient descent method was working as expected and moving the source location estimate (X, Y) in the correct direction. It was observed that after some iterations the model was adjusting the positions of both sources as the error increases above the initial guess error.

The predicted locations obtained for the forest fire are as follows:

x = [80,130];

y = [75,85];

Based on the grid distribution in Matlab and overlapping the true map co-ordinates in the simulation the location of S1 and S2 was as depicted below on the map:

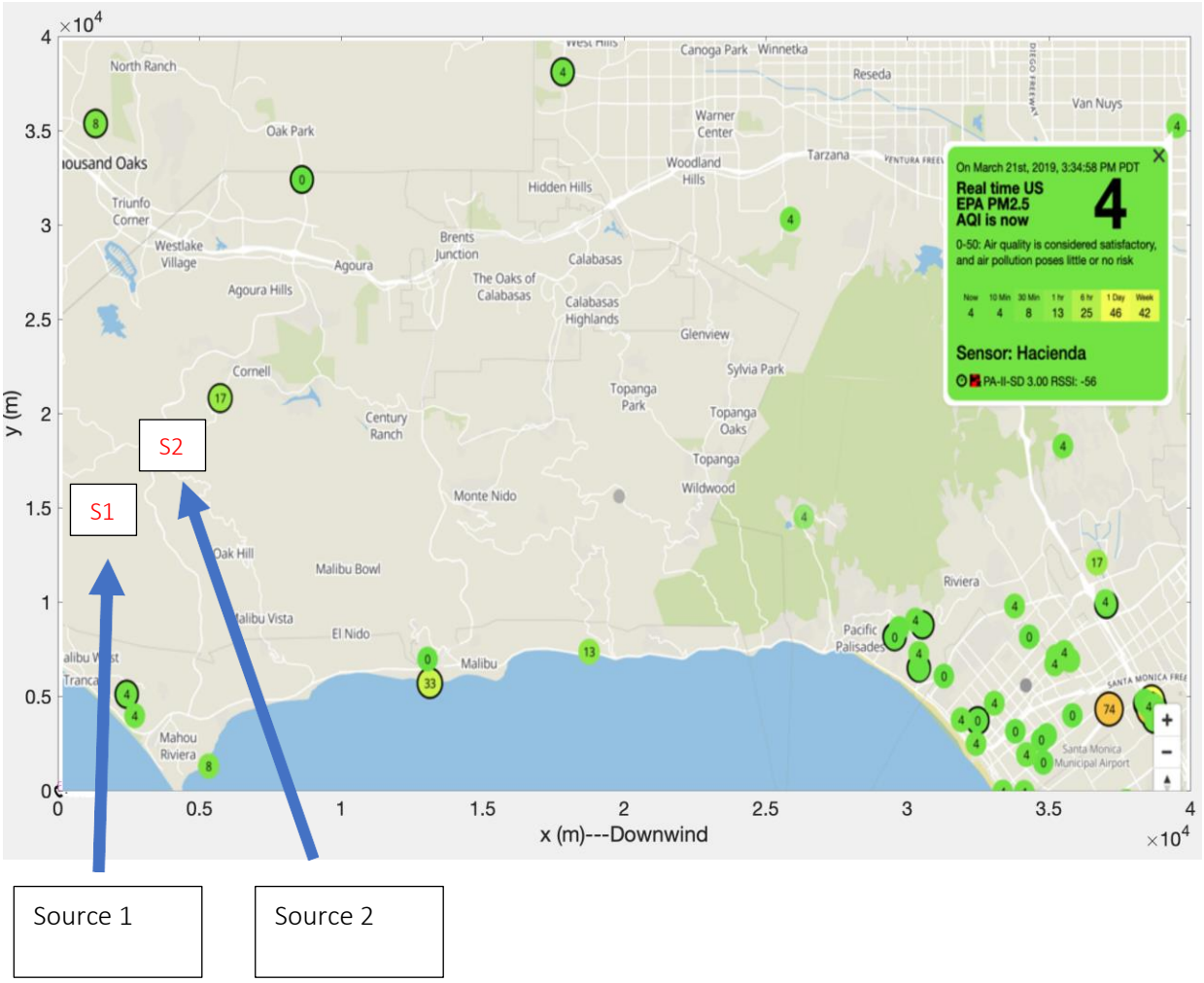

Figure 7.12 Predicted sources located on the map

We researched more information regarding the Woolsey fire on the California government fire information website.

http://cdfdata.fire.ca.gov/incidents/incidents_details_maps?incident_id=2282

The maps for the fire were as shown below:

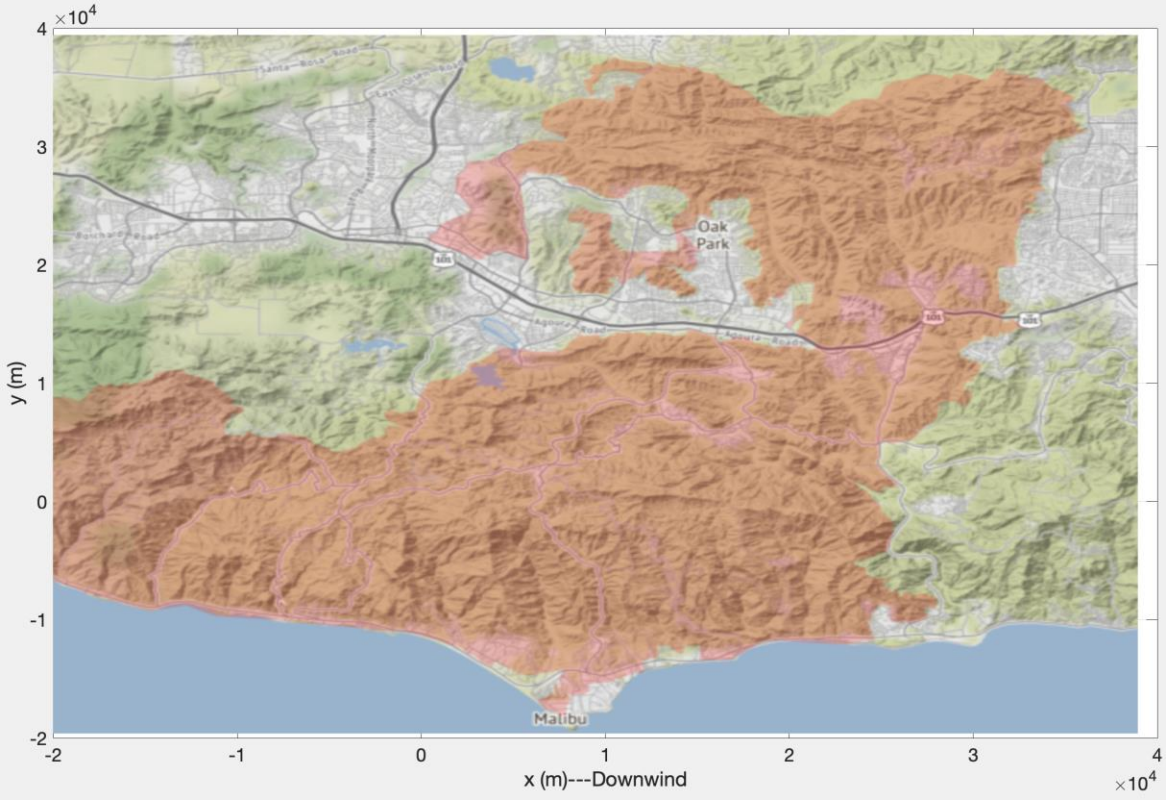

Figure 7.13 Affected Forest fire area

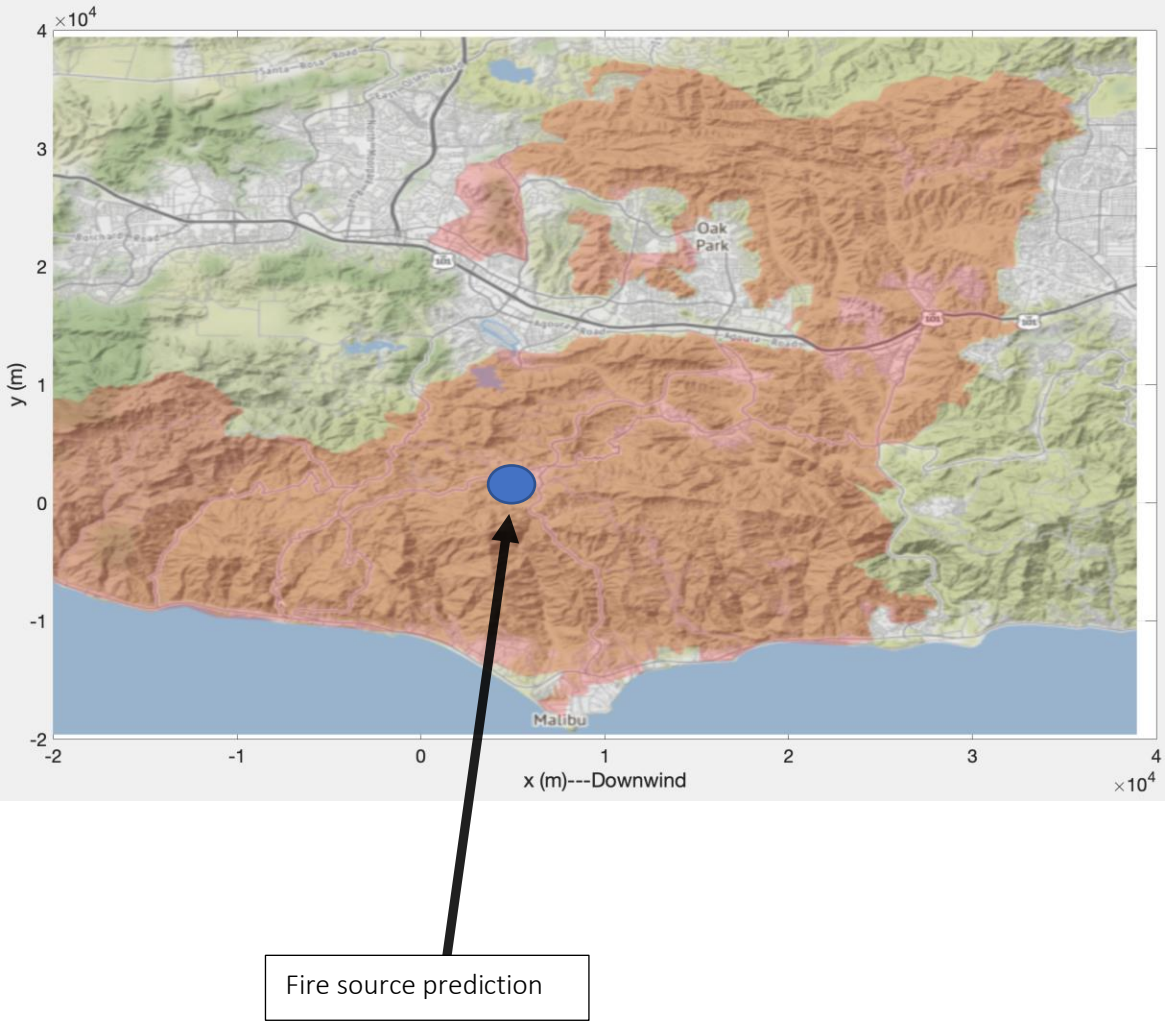

Figure 7.14 Overview of inverse models' prediction for the forest wire source

Thus, it can be observed from the above figure the model predicted the source location in the area of the fire. But it should be considered that this is not the exact location of the fire source since the area of the fire was very wide.

7.6 CONCLUSION

The inverse model predicts the source locations and emission rates with high accuracy. But it was observed that when we add noise, wind speed, wind angle, and sensor placements into the model source prediction accuracy is affected. It was observed that the accuracy of the model

decreased from 99. 97 % to around 93 % by reducing the number of sensor placements inside the plume. Moreover, the accuracy is highly affected by the noise levels which increased the error term around nine times as the signal-to-noise ratio was changed from 40 dB to – 40 dB. These parameters make the model more complex when reverse engineering to predict the source parameters. Thus, simulation in more steady conditions for wind speed and angle might be a simple solution for the theoretical model. But, in practical solutions, while implementing this model in a wireless sensor network it is important to consider the wind conditions and the noise factor to predict the source of pollutant accurately. Based on the Woolsey fire experiment we concluded that there is no optimal distance between the source and the sensor locations. There were sensors in the area of the fire as well as approximately 20 miles away from the source.

Chapter 8

INVERSE PROBLEM SOLUTION WITH NEURAL NETWORKS

In Section 2.4 we had some introduction to neural networks. In the beginning sections of this chapter, we will be learning about Feed Forward neural networks and activation functions. Later we will discuss the results of the inverse model obtained using the neural networks.

8.1 INTRODUCTION TO THE FEED FORWARD NEURAL NETWORK MODEL

A Feed Forward neural network, also known as a multilayer perceptron (MLP), is a fundamental type of artificial neural network. It consists of multiple layers of interconnected neurons [11]. The information flows through the network in a forward direction, starting from the input layer and passing through one or more hidden layers before reaching the output layer. Each neuron in a layer receives inputs from the previous layer and applies a set of weights to those inputs, followed by an activation function. The activation function introduces non-linearity to the network, allowing it to learn complex relationships between inputs and outputs. Feed Forward neural networks are widely used for various machine learning tasks, including classification, regression, and pattern recognition. Their simplicity and ability to approximate any continuous function make them a popular choice in the field of deep learning. Figure 8.1 shows an example of a Feed Forward neural network.

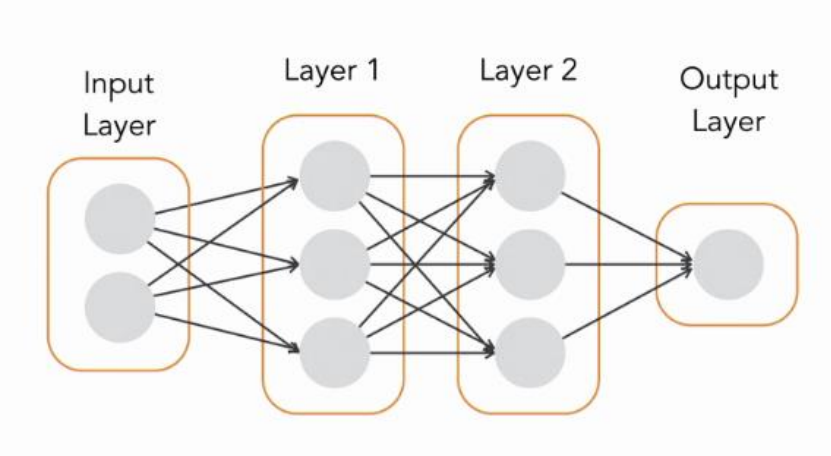

Figure 8.1 Neural network layers

The neural network in Figure 8.1 has an input and an output layer. All the layers between the input and output layers are hidden layers.

8.1.1 Introduction to neural network activation functions

The activation function performs non-linear transformation to the input enabling the neural network to learn and perform computation tasks like image processing and language translation [11]. There are many types of neural network activation functions. Three examples of types of activation functions are binary step function, linear activation function, and non-linear activation functions. In this thesis, we are interested in non-linear activation functions.

8.1.1.1 Sigmoid / Logistic Activation Function

This function takes any real value as input and outputs values in the range of 0 to 1. The larger the input (more positive), the closer the output value will be to 1.0, whereas the smaller the input (more negative), the closer the output will be to 0.0, as shown in Figure 8.2.

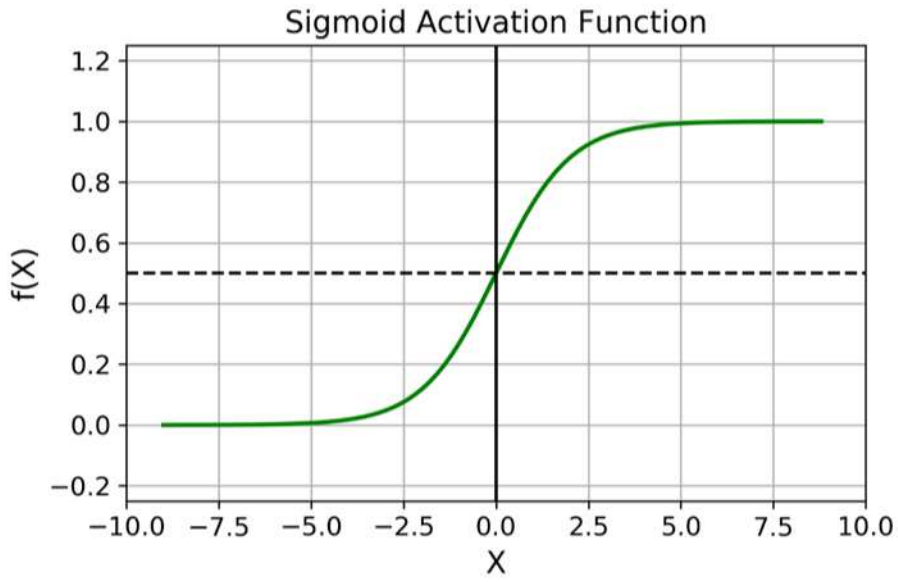

Figure 8.2: Sigmoid/Logistic function

It is commonly used for models where we have to predict the probability as an output. Since the probability of anything exists only between the range of 0 and 1, sigmoid is the right choice because of its range. The function is differentiable everywhere and provides a smooth gradient, i.e., preventing jumps in output values. This is represented by an S-shape of the sigmoid activation function.

8.1.1.2 ReLU Function

ReLU stands for Rectified Linear Unit. Although it gives an impression of a linear function, ReLU has a derivative function and allows for backpropagation while simultaneously making it computationally efficient. The neurons will only be deactivated if the input to the neuron is less than 0.

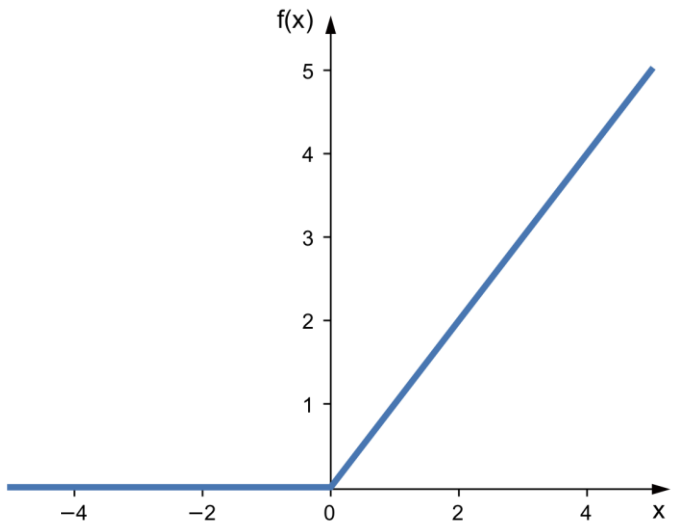

Figure 8.3: ReLU activation function

Since only a certain number of neurons are activated, the ReLU function is far more computationally efficient when compared to the sigmoid and tanh functions. ReLU accelerates the convergence of gradient descent towards the global minimum of the loss function due to its linear, non-saturating property [10].

8.1.2 Feature engineering

Feature engineering refers to the process of using domain knowledge to select and transform the most relevant variables from raw data when creating a predictive model using machine learning or statistical modeling. The goal of feature engineering and selection is to improve the performance of machine learning (ML) algorithms. Feature engineering involves steps such as feature creation, transformation, feature extraction, and feature selection. In our problem, we had a broad range of sensor readings, therefore it was difficult to analyze any regression patterns using this data. Thus, we transformed this data using a method called log transformation.

8.1.2.1 Log transformation

Data transformation is the process of taking a mathematical function and applying it to the data. Each variable x is replaced with log(x), where the base of the log is left up to the analyst. It is considered common to use base 10, base 2, and the natural log ln. For example, if we have a very large range of data, then smaller values can get overwhelmed by the larger values. Taking the log of each variable enables the visualization to be clearer. Thus, log transformation also de-emphasizes outliers and allows us to potentially obtain a bell-shaped distribution. The idea is to take the log of the data, so it restores symmetry to the data [8].

8.1.2.2 Tensor flow and Keras library

Tensor flow is an open programming framework for building machine learning applications. It works by holding data in multi-dimensional structures known as tensors. We built the model using the Keras library. This library acts as an application program interface (API) for the tensor flow platform and helps to develop neural network models faster using Python programming language.

8.2 RESULTS FOR THE NEURAL NETWORK MODEL

8.2.1 Model results for single target variable

In this chapter, we predicted the source parameters using neural networks and compared the results with the physical inverse model developed in the previous chapters. The neural network model was developed for one and two sources with nine sensor readings. This section discusses the results of the neural network model with a single target variable. The predicted

variable is the source emission rate. The neural network model was created using Python programming language. We used the tensor flow and pandas library. These libraries were very useful for enabling neural network simulations and data engineering.

The input parameters to the forward model are source rates, wind speed, wind direction, and source locations. Synthetic data for varying wind speed, and wind angle was generated using the randi() function in Matlab. The range for windspeed was 3 m/s to 30 m/s, whereas the range for wind angle was between 15 degrees to 60 degrees. The different configurations in the forward model for synthetic data and its corresponding predicted results are shown in Table 8.1.

The parameters required to generate the forward synthetic data for all NN experiments are source location, source emission rate, wind speed, wind direction, and sensor locations. For different experiments, some of these parameters were varied, as indicated in Table 8.1, while others were kept constant. The inputs to the forward model were used to calculate the concentration for each sensor location, which serve as the primary features, i.e. inputs, for the NN model to predict the source emission rate. In Table 8.1 the output predicted variable is only the source emission rate. The corresponding R^2 score for the predictions is mentioned in the performance column.

Table 8.1 Different configurations in the forward model for synthetic data and its corresponding predicted results in the neural network model

Forward model | Inverse model NN			
Experiment Name	**Varying Parameters**	**Features/Input**	**Performance**
Source emission rate predictions with varying wind speed.	Wind speed	Sensor measurements, Sensor locations, source location, wind speed, wind direction Data size: 2000	**R^2 score: 1.0** Refer Figure 8.4
Source emission rate prediction with varying	Source locations, wind speed	Sensor measurements, source locations,wind speed,wind direction	**R^2 score: 0.851** Refer Figure 8.5

source locations and wind speed.		Data size: 4000	
Source emission rate prediction with varying source locations, wind velocity, and log transformations.	Source locations, wind speed	Sensor measurements, source locations, wind speed, wind direction Data size: 4000 Feature engineering: Log transformations and reciprocals.	**R^2 score: 0.999** Refer Figure 8.6

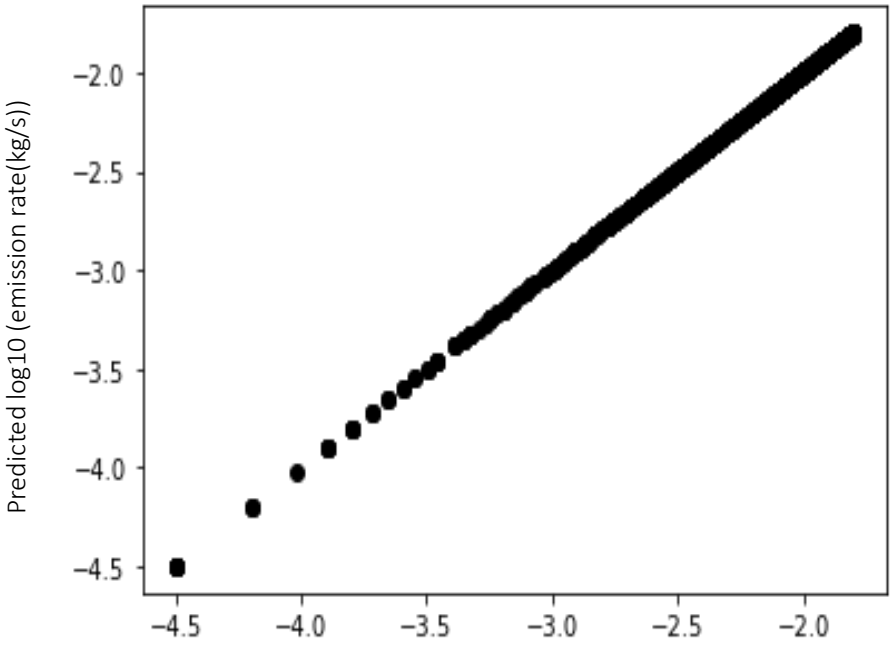

Figure 8.4 Source emission rate predictions with varying wind speed

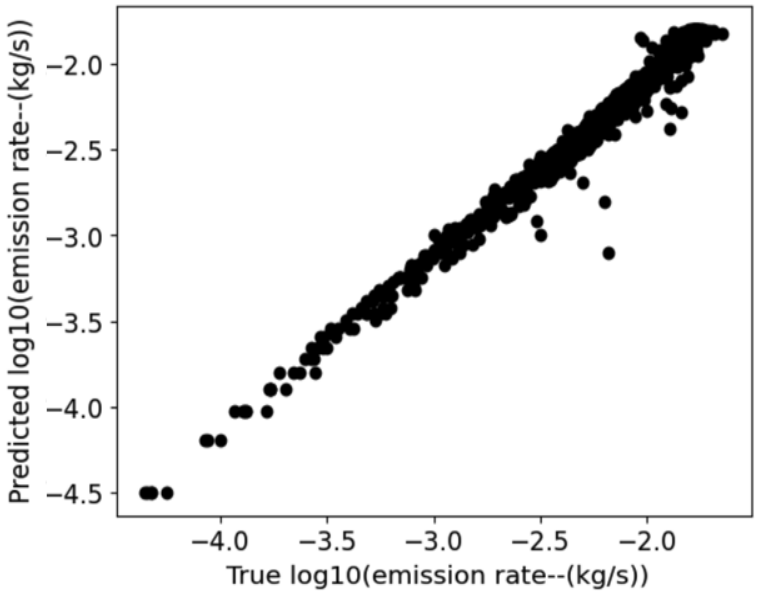

Figure 8.5 Source emission rate predictions with varying source locations and wind speed

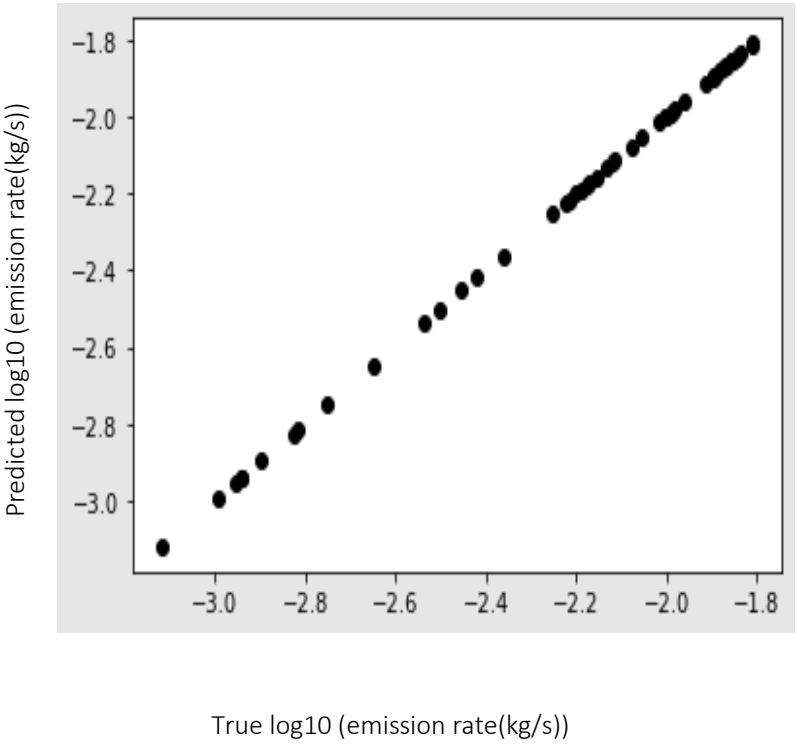

Figure 8.6 Source emission rate prediction with varying source locations, wind velocity, and log transformations.

In Figures 8.4, 8.5, and 8.6 The values are negative because of emission rate reciprocal is the target.

8.2.2 Model results for multi-target variables

In Section 8.2.1, the neural network experiments were focused on a single target variable, but in this section, we will focus on multitarget variables. The multitarget variables include source emission rates, source location X, and source location Y. For this experiment, we used the ReLU activation function. The data was separated into 80% training and 20 % testing. The batch size of the simulation was selected to be 4000 and the epoch size was selected to be 400. Table 8.1 below shows the results obtained using the multioutput neural network model.

Table 8.2 Results obtained using the multioutput neural network model

Forward model		**Inverse model NN**		
Experiment Name	**Varying Parameters**	**Features**	**Predicted variables**	**Performance**
Source rate and source location predictions with log transformations and reciprocals.	Source rates, wind speed, wind directions	Sensor measurements, wind speed, wind direction Data Size: 4000	Source rate, source locations	Refer Figures 8.7, 8.8, and 8.9. **R^2 score for X location predictions: 0.947** **R^2 score for Y location predictions: 0.935**

				R^2 score for source emission rate predictions: 0.932 **Epoch 492/500** **mean_squared error: 0.0101**

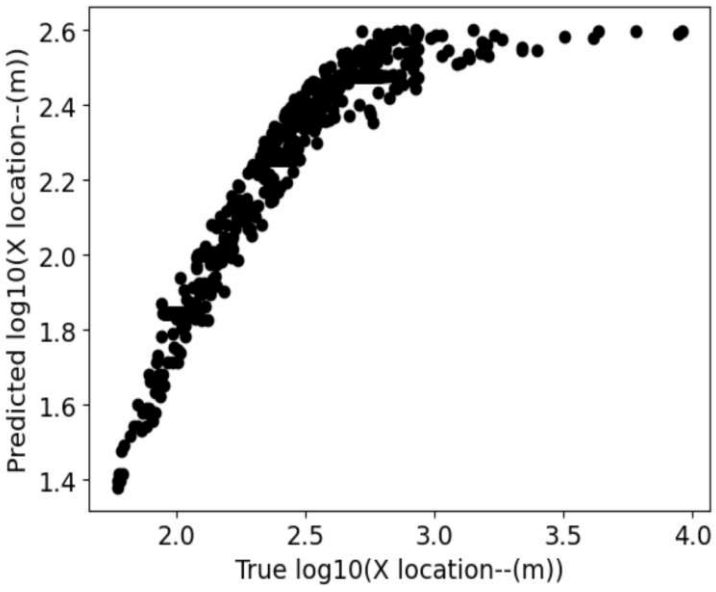

Figure 8.7 X location predictions

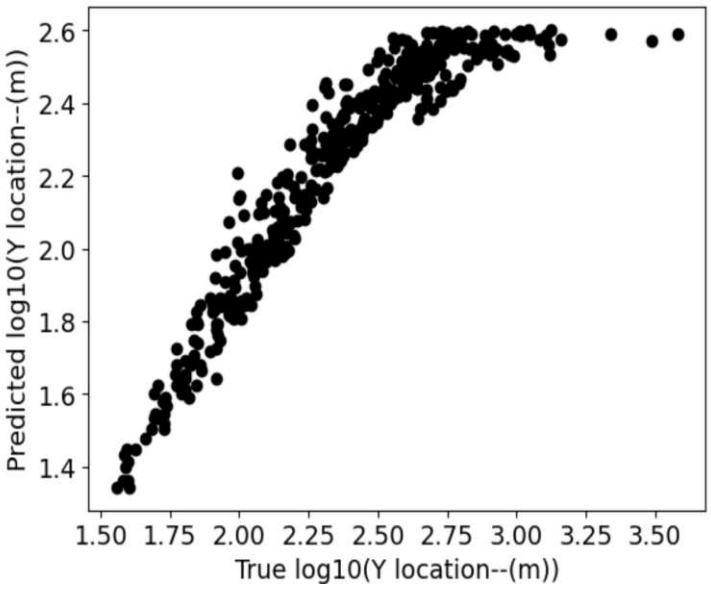

Figure 8.8 Y location predictions

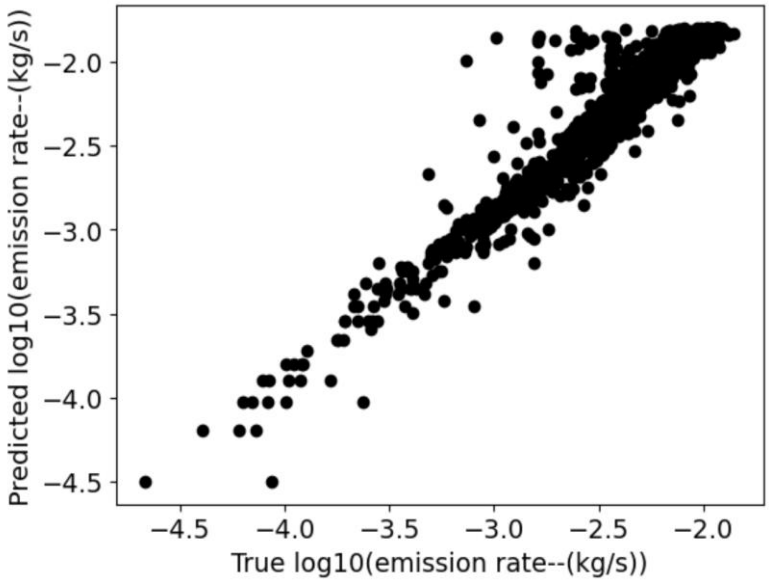

Figure 8.9 Source Emission rate predictions

8.3 COMPARISON OF NEURAL NETWORK AND PHYSICAL INVERSE MODEL RESULTS

We created synthetic data with 200 different configurations and used it to generate source estimation results in the physical inverse model and the trained neural network model.

Table 7.3 Range of configuration values for generating synthetic data

Parameter	**Range**	**Unit**
Wind speed	5-20	m/sec
Wind angle	10-60	degrees
Sensor locations (distance from the source)	0- 10000	meters
Source locations (X,Y)	(0,0) – (900, 900)	meters
Source emission rates	0.00634 - 3.1709	Kg/sec
Number of receptor measurements	9	Unitless

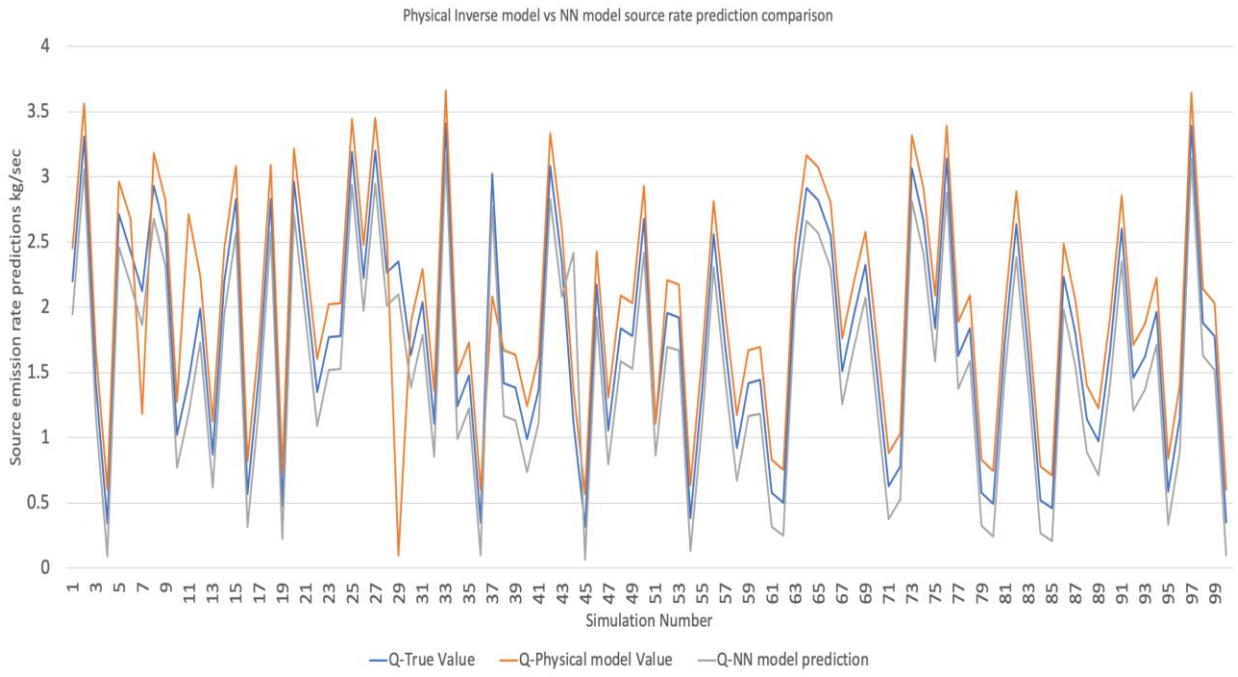

Figure 8.10 Physical inverse model vs NN model rate prediction comparison

The R^2 generated for predictions in the physical model was 0.88 whereas the R^2 generated in the neural network model was around 0.95.

8.3 CONCLUSION

The neural networks experiment suggests that source parameters prediction problem can be solved efficiently using a neural network model. The number of hidden and output nodes can be changed based on the number of output source parameters and the complexity of the input data. The regression scores and the mean square error values show promising results to implement the model for practical air quality monitoring experiments.

Chapter 9

CONCLUSION AND SUGGESTIONS FOR FUTURE WORK

9.1 CONCLUSION

The work in this thesis focused on the development of a physical model using the Gaussian plume equation for pollutant source predictions. This model was created in two parts. The first physical model part focused on the development of a forward model using the Gaussian plume equation to calculate sensor measurements based on the known source locations and source emission rates. The second part of the physical model focused on the development of an inverse model for predicting the unknown source locations and source emission rates from known measurement readings. This method used L2 minimization and gradient descent methods to predict the source emission rate and location. We also developed a neural network model to predict the source emission rates and location. It was found that the physical forward model using the Gaussian plume equation performed as expected for generating the sensor measurements. The calculated sensor values using the Gaussian plume equation and the physical model results for measurements were approximately 99% accurate for all the simulations. In the inverse physical part of the model, it was found that the source emission rate prediction accuracy varied between 85-90 % whereas the source location prediction accuracy varied between 90-95%. However, based on different settings for generating synthetic data, the accuracy of the inverse model varied for different source emission rates, source locations, noise levels, wind speeds, and wind angles. Thus, it can be concluded that while setting up this model for practical field applications the implementation process should consider all these parameters to determine the preliminary accuracy of the model. This would

also help in determining the number of wireless sensors or measurements required, the wind speed levels, and the area of application or surveillance required for desired prediction model accuracy. Further, we implemented a neural network model which could predict the source emission rate and source locations. It was found that the neural network performs well for the prediction of source emission rates and source locations. For a single target model with only the source rate as the target variable the neural network model able to achieve an R^2 score of 0.99 to 1. Whereas a multitarget sequential neural network model which had source emission rate and source locations as the target variables were able to achieve a reasonably low root mean square error with the predicted variables being very close to the true values. But it also should be noted that as we introduced the noise, wind speed, and wind angle values for the synthetic data the neural network model gradually had lower R^2 scores and higher root mean squared error values. These inaccuracies in the model were similar to the physical inverse model. Thus, it should be observed that the pollutant emission and its plume formation are highly affected by the wind parameters and noise levels of the measured reading. It can be further concluded that the inverse physical model and neural network models are promising solutions for prediction accuracies above 90% for solving inverse Gaussian plume problems, but more field testing and simulations are required to decide on the most reliable model for complete practical implementation in a wireless sensor network setup.

9.2 FUTURE WORK

Although the inverse physical model and the neural network model performed well in predicting the pollutant source parameters, there is still a very high scope for future work before this model is completely deployed for an air quality wireless sensor network project. The physical and neural network models can be trained rigorously for very huge synthetic data sets. Also, the model should be further trained with synthetic data sets generated from

more types of configurations for source locations, sensor locations, wind speed, angle, and noise. The model can be further developed using similar algorithms to the gradient descent methods such as conjugated gradient optimization, Levenberg-Marquardt algorithm (LMA), and Particle Swarm optimization to compare the accuracy. This model also can be implemented with complex neural network algorithms to test its reliability for larger synthetic data sets generated with complex configurations. It can also be trained with complex datasets such as topography information and source heights. The model can be further optimized for running the simulations faster for huge training datasets with newly optimized Matlab libraries in the future. The model needs to be tested in an air quality-based wireless sensor field setup. This model also needs to be tested with huge data sets from projects involving air quality monitoring forest fires or industrial air pollution. This model work can also facilitate implementing air quality sensors network applications which can auto-simulate based on the collected reading to flag a possible air pollutant source.

BIBLIOGRAPHY

[1] D. B. Turner, Atmospheric dispersion modeling: A critical review, Journal of the Air Pollution Control Association 29 (5) (1979) 502–519.

[2] O. G. Sutton, A theory of eddy diffusion in the atmosphere, Proceedings of the Royal Society of London, Series A 135 (826) (1932) 143–165.

[3] P. Völgyesi, A. Nádas, X. Koutsoukos and Á. Lédeczi, "Air Quality Monitoring with SensorMap," 2008 International Conference on Information Processing in Sensor Networks (ipsn 2008), 2008, pp. 529-530, doi: 10.1109/IPSN.2008.50.

[4] The Mathematics of Atmospheric Dispersion Modeling John M. Stockie SIAM Review 2011 53:2, 349-372

[5] Amiri Simkooei, A. (2003). Formulation of L1 norm minimization in Gauss-Markov models. *Journal of Surveying Engineering*, *129*(1), 37-43.

[6] D. L. Ermak, An analytical model for air pollutant transport and deposition from a point source, Atmospheric Environment 11 (3) (1977) 231–237.

[7] K. L. Calder, Multiple-source plume models of urban air pollution – Their general structure, Atmospheric Environment 11 (1977) 403–414.

[8] Leigh Metcalf, William Casey, Chapter 4 - Introduction to data analysis, Editor(s): Leigh Metcalf, William Casey, Cybersecurity and Applied Mathematics, Syngress, 2016,Pages 43-65,ISBN 9780128044520,

[9] Michael Hutchinson, Hyondong Oh, Wen-Hua Chen, Information Fusion, Volume 36, 2017, A review of source term estimation methods for atmospheric dispersion events using static or mobile sensors,

[10] Ramachandran, P., Zoph, B., & Le, Q. V. (2017). Searching for activation functions. arXiv preprint arXiv:1710.05941.

[11] Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. towards data science, 6(12), 310-316.

[12] B. Bavarian, "Introduction to neural networks for intelligent control," in IEEE Control Systems Magazine, vol. 8, no. 2, pp. 3-7, April 1988, DOI: 10.1109/37.1866.

[13] C. -Y. Chong, K. -C. Chang and S. Mori, "Distributed Tracking in Distributed Sensor Networks," 1986 American Control Conference, 1986, pp. 1863-1868, DOI: 10.23919/ACC.1986.4789229.

[14] Demir-Kavuk, O., Kamada, M., Akutsu, T., & Knapp, E. W. (2011). Prediction using step- wise L1, L2 regularization and feature selection for small data sets with large number of features. BMC bioinformatics, 12, 412.

[15] S. Mansour, N. Nasser, L. Karim and A. Ali, "Wireless Sensor Network-based air quality monitoring system," 2014 International Conference on Computing, Networking and Communications (ICNC), 2014, pp. 545-550, doi: 10.1109/ICCNC.2014.6785394.

[16] Abdel-Rahman, A. A. (2008, October). On the atmospheric dispersion and Gaussian plume model. In Proceedings of the 2nd International Conference on Waste Management, Water Pollution, Air Pollution, Indoor Climate, Corfu, Greece (Vol. 26).

[17] Kahl, Jonathan & Chapman, Hillary. (2018). Atmospheric stability characterization using the Pasquill method: A critical evaluation. Atmospheric Environment. 187. 10.1016/j.atmosenv.2018.05.058.

[18] S. Boyd and L. Vandenberghe, "Least Squares Problems," in Convex Optimization, Cambridge University Press, 2004, pp. 58-69.

[19] T. Ishigami and T. Homma, "An importance quantification technique in uncertainty analysis for computer models," [1990] Proceedings. First International Symposium on Uncertainty Modeling and Analysis, College Park, MD, USA, 1990, pp. 398-403, doi: 10.1109/ISUMA.1990.151285.

[20] K. S. Rao, “Source estimation methods for atmospheric dispersion,” Atmospheric Environment, vol. 41, pp. 6964–6973, 2007.

[21] Falla, J., Laval-Gilly, P., Henryon, M. et al. Biological Air Quality Monitoring: a Review. Environ Monit Assess 64, 627–644 (2000).

[22] Chakraborty, Monojit & Sangeeta, Bansal & Renu, M & Amit, A. (2019). Air-Pollution modelling aspects: an overview.. 10.1079/9781786393890.0079.

[23] Nguyen Thi Kim Oanh, Prapat Pongkiatkul, Nabin Upadhyay, Phillip P. Hopke,

Designing ambient particulate matter monitoring program for source apportionment study by receptor modeling,Atmospheric Environment,Volume 43, Issue 21, 2009.

[24] YuanXun Zhang, Rebecca J. Sheesley, James J. Schauer, Michael Lewandowski, Mohammed Jaoui, John H. Offenberg, Tadeusz E. Kleindienst, Edward O. Edney,

Source apportionment of primary and secondary organic aerosols using positive matrix factorization (PMF) of molecular markers, Atmospheric Environment,Volume 43, Issue 34, 2009.