

# **SALINITY INTRUSION PREDICTION USING MACHINE LEARNING**

*A main project report submitted in partial fulfillment of the  
Requirements for the award of the Degree of*  
**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

*By*

<b>R.HIMAVANT NAATH</b>	<b>196M1A0566</b>
<b>D.AKHILA DEVI</b>	<b>196M1A0514</b>
<b>G.NAMO SAI KRISHNA MURTHY</b>	<b>196M1A0522</b>
<b>M.KARUNA</b>	<b>196M1A0543</b>

**Under the esteemed Guidance of**

**Mr.R.N.V.VISHNU MURTHY** M.Tech., MISTE.

**Assistant Professor**

**Department of CSE**



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**  
**B. V. C. COLLEGE OF ENGINEERING**

**(Accredited by NAAC with “A” Grade)**

**RAJAHMAHENDRAVARAM,**

**ANDHRA PRADESH**

**(2019-2023)**

JNTUK Code : **6M**  
SBTET Code : **347**

Counseling Code : **BVCR**



# **B V C COLLEGE OF ENGINEERING**

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada & SBTET, Vijayawada)

**PALACHARLA, RAJAHMAHENDRAVARAM** - 533 102. E G Dt. (AP). Cell : 97045 78666, 99519 49356.


email : [bvcr@bvcgroup.in](mailto:bvcr@bvcgroup.in), [poly347@bvcgroup.in](mailto:poly347@bvcgroup.in), website : [bvcce.org](http://bvcce.org)

## **Vision**

To become a model abode of Learning with time trusted Academic values for serving the Nation and the World.

## **Mission**

- To build lively ambience and provide learning etiquette for all round growth of students.
- To augment Industry – Institute Interaction through training and skill development activities.
- To inculcate Social service and Human values amongst budding professionals.
- To promote Innovation, Entrepreneurship, Research and Consultancy.

  
PRINCIPAL  
PRINCIPAL  
BVC COLLEGE OF ENGINEERING  
PALACHARLA, RAJAMAHENDRAVARAM

JNTUK Code : 6M  
SBTET Code : 347

Counseling Code : BVCR



# B V C COLLEGE OF ENGINEERING

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada & SBTET, Vijayawada)

PALACHARLA, RAJAHMAHENDRAVARAM - 533 102. E G Dt. (AP). Cell : 97045 78666, 99519 49356.

email : bvcrc@bvcgroup.in, poly347@bvcgroup.in, website : bvcce.org

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### DEPARTMENT VISION

To be a center of excellence in Computer Science and Engineering to meet the growing needs of the industry and society with good ethical practices.

### DEPARTMENT MISSION

- M1:** To improve high quality education in Computer Science and Engineering which enables students globally competent.
- M2:** To collaborate with industry and institutes of higher learning for National and International repute.
- M3:** To foster Civic Minded Leadership with values and ethics among students.

Head of the Department

Head of the Department  
Computer Science & Engineering  
BVC COLLEGE OF ENGINEERING  
PALACHARLA - 533 102.



# B V C COLLEGE OF ENGINEERING

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada & SBTET, Vijayawada)

PALACHARLA, RAJAHMAHENDRAVARAM - 533 102. E G Dt. (AP). Cell : 97045 78666, 99519 49356.


email : bvcr@bvccgroup.in, poly347@bvccgroup.in, website : bvccce.org

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### Program Outcomes

Engineering Graduates will be able to:

- 1.Engineering knowledge:** Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.
- 2.Problem analysis:** Identify, formulate, review research literature, and analyse complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.
- 3.Design/development of solutions:** Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.
- 4.Conduct investigations of complex problems:** Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.
- 5.Modern tool usage:** Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modelling to complex engineering activities with an understanding of the limitations.
- 6.The engineer and society:** Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.
- 7.Environment and sustainability:** Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.
- 8.Ethics:** Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.
- 9.Individual and team work:** Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.
- 10.Communication:** Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.
- 11.Project management and finance:** Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.
- 12.Life-long learning:** Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

  
Head of the Department

Head of the Department  
Computer Science & Engineering  
BVC COLLEGE OF ENGINEERING  
PALACHARLA - 533 102.



JNTUK Code : 6M  
SBTET Code : 347

Counseling Code : BVCR



# B V C COLLEGE OF ENGINEERING

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada & SBTET, Vijayawada)

PALACHARLA, RAJAHMAHENDRAVARAM - 533 102. E G Dt. (AP). Cell : 97045 78666, 99519 49356.

email : bvcrc@bvcgroup.in, poly347@bvcgroup.in, website : bvcce.org

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### PROGRAM SPECIFIC OUTCOMES( PSOs):

**PSO 1 :** Ability to apply Mathematical Methodologies to solve computational tasks and build real time problem using appropriate Data Structure and suitable algorithms for the different domains in Computer Science and Engineering .

**PSO 2:** To build models and provide solutions on societal, ethical and environmental issues.

Head of the Department.

Head of the Department  
Computer Science & Engineering  
BVC COLLEGE OF ENGINEERING  
PALACHARLA - 533 102.

JNTUK Code : 6M  
SBTET Code : 347

Counseling Code : BVCR



# B V C COLLEGE OF ENGINEERING

(Approved by AICTE, New Delhi, Affiliated to JNTUK, Kakinada & SBTET, Vijayawada)

PALACHARLA, RAJAHMAHENDRAVARAM - 533 102. E G Dt. (AP). Cell : 97045 78666, 99519 49356.

email : bvcrc@bvcgroup.in, poly347@bvcgroup.in, website : bvcce.org

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

### PROGRAM EDUCATIONAL OUTCOME (PEOs)

**PEO1:** Graduates will have knowledge of Mathematics, Science, Engineering fundamentals and in-depth studies in Computer Science and Engineering, and will be able to apply them for formulating, analyzing and solving real-world problems.

**PEO2:** Graduates will demonstrate creativity in their Engineering practices including Entrepreneurial and Collaborative ventures with strategic thinking, planning and execution.

**PEO3:** Graduates will communicate effectively, recognize and incorporate societal needs and constraints in their professional endeavors, and practice them in profession with legal and ethical responsibilities.

**PEO4:** Graduates will succeed in earning coveted entry level positions in leading computer software and hardware firms in India and Abroad.

Head of the Department.

Head of the Department  
Computer Science & Engineering  
BVC COLLEGE OF ENGINEERING  
PALACHARLA - 533 102.

# **B.V.C COLLEGE OF ENGINEERING**

**PALACHARLA- 533102**

## **DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING**

### **CERTIFICATE**



*This is to certify that project entitled “**SALINITY INTRUSION PREDICTION USING MACHINE LEARNING**” that is being submitted by **R.HIMAVANT NAATH (196M1A0566)**, **D.AKHILA DEVI (196M1A0514)**, **G.NAMO SAI KRISHNA MURTHY (196M1A0522)**, **M.KARUNA (196M1A0543)** in partial fulfillment of the requirements for the award of *Bachelor of Technology* in **COMPUTER SCIENCE AND ENGINEERING** of **B.V.C COLLEGE OF ENGINEERING** is a bonafide work carried out by them during the academic year 2022- 2023.*

#### **INTERNAL GUIDE**

**Mr.R.N.V.VISHNU MURTHY**, M.Tech, MISTE.

Assistant Professor,

Department of CSE,

**B.V.C COLLEGE OF ENGINEERING,**

**Rajamahendravaram.**

#### **HEAD OF THE DEPARTMENT**

**Dr .Y. Venkateswarlu**, Ph. D

Professor,

HOD, Department of CSE,

**B.V.C COLLEGE OF ENGINEERING,**

**Rajamahendravaram.**

**External Examiner**

## **ACKNOWLEDGEMENT**

First and foremost, we sincerely salute our esteemed institution **B.V.C COLLEGE OF ENGINEERING** for giving the support in completion of the project work.

We would like to express our sincere gratitude to our project guide, **Mr.R.N.V.VISHNU MURTHY** M.Tech., MISTE **Assistant Professor** for his guidance, encouragement and continuing support throughout the course of this work.

We are highly obliged to our Head of the Department, **Dr .Y. Venkateswarlu,** Ph. D **Professor** for his constant inspiration, extensive help and valuable support in our every step.

We owe a great deal to our principal **Dr. G. RAVI KANTH,** Ph.D., **Professor** for extending a helping hand at every juncture of need.

Finally, we are pleased to acknowledge our indebtedness to all those who devoted themselves directly or indirectly to make this project work a total success.

### **PROJECTEES**

<b>R.HIMAVANT NAATH</b>	<b>196M1A0566</b>
<b>D.AKHILA DEVI</b>	<b>196M1A0514</b>
<b>G.NAMO SAI KRISHNA MURTHY</b>	<b>196M1A0522</b>
<b>M.KARUNA</b>	<b>196M1A0543</b>



## **DECLARATION BY THE CANDIDATE**

We, **R.HIMAVANT NAATH, D.AKHILA DEVI, G.NAMO SAI KRISHNA MURTHY, and M.KARUNA**, bearing register numbers **196M1A0566, 196M1A0514, 196M1A0522, and 196M1A0543** hereby declare that the project reported titled **“SALINITY INTRUSION PREDICTION USING MACHINE LEARNING”** under the guidance of **Mr.R.N.V.VISHNU MURTHY** M.Tech., MISTE. is submitted in partial fulfillment of the requirements for the award of the degree of Bachelor of Technology in Computer Science And Engineering.

This is a record of bonafide work carried out by us, the result Embodied in this project report have not been reproduced or copied from any source and have not been submitted to any other university or institute for the award of any other degree.

### **PROJECTEES**

<b>R.HIMAVANT NAATH</b>	<b>196M1A0566</b>
<b>D.AKHILA DEVI</b>	<b>196M1A0514</b>
<b>G.NAMO SAI KRISHNA MURTHY</b>	<b>196M1A0522</b>
<b>M.KARUNA</b>	<b>196M1A0543</b>

## ABSTRACT

Seawater intrusion is the movement of seawater into freshwater aquifers due to natural processes or human activities. Seawater intrusion is caused by decreases in ground water levels or by rises in seawater levels. When you pump out fresh water rapidly, you lower the height of the freshwater in the aquifer forming a cone of depression.

Intrusion can affect the quality of water not only at the pumping well sites, but also at other well sites, and undeveloped portions of the aquifer.

The objectives of this study were to predict the water quality index using a linear regression model and to identify the most important attributes affecting the variability of the water quality index in Indian coastal basins. Water samples at each site have been collected yearly. At each station, water samples were collected from inside the middle of the river by means of a plastic bucket and were transported to the laboratory.

Water quality parameters were measured, calculated and classified according to the standard methods. Prediction of the linear regression models in the study area resulted in determination coefficient and root mean square error of 0.87 and 0.061 for the water quality index respectively. Nitrate was identified as the most important attribute influencing the water quality index.

Overall, our results indicated that the linear regression models could explain 87% of the total variability in the water quality index. Besides, the predictability of the water quality index could be improved by other statistical and intelligent models. These predictions help us to improve river management regarding water quality

# CONTENTS

INDEX	PAGE NO
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 OVERVIEW	1
1.2 BACKGROUND	1
1.3 ENVIRONMENT ISSUES	2
1.4 WATER QUALITY ANALYSIS	3
<b>2. LITERATURE SURVEY</b>	<b>4</b>
2.1 WHAT IS MACHINE LEARNING	6
2.2 TYPES OF MACHINE LEARNING	7
2.2.1 SUPERVISED LEARNING	7
2.2.2 UNSUPERVISED LEARNING	8
2.2.3 REINFORCEMENT LEARNING	9
2.3 PYTHON PROGRAMMING LANGUAGE	9
2.4 TEACHING MACHINES TO LEARN	10
2.5 FOR WHAT REASON IS PYTHON THE BEST PROGRAMMING LANGUAGE FOR AI?	11
2.6 PAST SUCCESSFUL PYTHON AI PROJECTS	11
2.7 HOW TO MAKE A BOT WITH PYTHON?	12
2.8 PYTHON LOVES WEB DEVELOPMENT	13
2.9 INFORMATION MINING AND PYTHON	13
2.10 GUI-BASED DESKTOP PROGRAMS	14
2.11 MAKE GAMES AND 3D GRAPHICS WITH PYTHON	15
2.12 TERMINATIONS	15
2.13 PYTHON LIBRARIES	15
2.14 PYTHON LIBRARIES THAT USED IN MACHINE LEARNING	16
2.15 SYSTEM STUDY	17
2.15.1 FEASIBILITY STUDY	17
2.15.2 TECHNICAL FEASIBILITY	17

2.15.3 SOCIAL FEASIBILITY	18
2.16 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS	18
2.16.1 FUNCTIONAL REQUIREMENTS	18
2.16.2 USER REQUIREMENTS	18
2.16.3 NON FUNCTIONAL REQUIREMENTS	19
<b>3. SYSTEM SPECIFICATION</b>	<b>20</b>
3.1 HARDWARE REQUIREMENTS	20
3.2 SOFTWARE REQUIREMENTS	20
<b>4. SYSTEM ANALYSIS</b>	<b>21</b>
4.1 EXISTING SYSTEM	21
4.2 PROPOSED SYSTEM	22
4.3 SYSTEM ARCHITECTURE	22
<b>5. SYSTEM DESIGN</b>	<b>30</b>
5.1 UML DIAGRAM	30
5.1.1 USE CASE DIAGRAM	31
5.1.2 DATA FLOW DIAGRAM	32
5.1.3 SEQUENCE DIAGRAM	33
<b>6. IMPLEMENTATION AND CODING</b>	<b>34</b>
<b>7. SYSTEM TESTING</b>	<b>41</b>
7.1 TYPES OF TESTS	41
7.1.1 UNIT TESTING	41
7.1.2 INTEGRATION TESTING	41
7.1.3 FUNCTIONAL TESTING	42
7.1.4 SYSTEM TESTING	42
7.1.5 WHITEBOX TESTING	42
7.1.6 BLACK BOX TESTING	42
7.1.7 ACCEPTANCE TESTING	44
7.2 TEST CASES	44

<b>8. OUTPUT SCREENS</b>	<b>45</b>
8.1 COMPARING THE ACTUAL AND PREDICTED VALUES:	45
8.2 THE SCATTERED PLOT OF DATA POINTS YEAR WISE:	46
8.3 VISUALIZING THE FILTERED DATA:	47
8.4 PLOTTING THE COST FUNCTION:	48
8.5 PLOTTING THE ACTUAL AND PREDICTED RESULTS:	48
<b>9. CONCLUSION</b>	<b>50</b>
<b>10. REFERENCES</b>	<b>51</b>





## LIST OF DIAGRAMS

FIGURE NO	NAME OF THE FIGURE	PAGE NO
1	Salinity Intrusion in Coastal Area	3
2	Location of the meakong Delta of Vietnam	6
4	Machine Learning	7
5	BOT 1	10
6	BOT 2	12
7	System Architecture	23
8	Collection of Data Set	24
9	Data Pre-Processing	25
10	salinity prediction	26
11	Linear Regression Sample Graph	26
12	Linear Regression Flow Chart	29
13	Use Case Diagram	31
14	Data Flow Diagram	32
15	Sequence Diagram	33
16	Test case 1	44
17	Test case 2	44
18	Prediction of January (2014-2022)	45
19	Plot	46
20	Scatter Plot	46
22	Plotting the Actual and Predicted Results	48

# 1. INTRODUCTION

## 1.1 OVERVIEW

With population growth, the demand for land coffer is anticipated to increase significantly in the coming decades. Maintaining the integrity of soil distribution requires a remarkable quantum of work to deal with agrarian extension.

Saltiness intrusion monitoring is a pivotal process, which directly affects sustainable development, especially in areas affected by global warming and in littoral zones. In recent times, colourful studies have used the soil-water saltiness data to estimate the spatiotemporal increase in saltiness intrusion. This study aims to establish a new frame for covering saltiness intrusion.

Using remote seeing and machine literacy. It focuses on the salt intrusion in water, which affects water vacuity, food security, mortal health, etc.

An aggregate of 1700 samples collected from 2014 to 2022 at few dimension stations were divided into two sets 70 training and 30 testings. Thirty-One independent variables were used to develop the model.

The results show that the vaticination model was erected successfully by applying data from the enforced saltiness dimension stations, Vaticination of the direct retrogression models in the study area was redounded in determination measure and root mean square error of 0.87 and 0.061 for the water quality indicator, independently. Nitrate was linked as the most important trait impacting the water quality indicator.

## 1.2 BACKGROUND

Water management is essential to ensure sustainable agricultural development and food production, with high-quality and environmental protection. According to the most recent estimates of the UN Food and Agriculture Organization, more than 20% of the cultivated land on Earth is variably degraded by soil salinization, and this is projected to reach ~50% by 2050 . Soil salinization affects approximately 10% of the world's food production. It is particularly high in many coastal countries and is said to increase in the future due to climate change. The major part

of affected land is in Asia (65%), followed by Africa (19%), and Europe (5%). The situation is Environment, 38% of its land could be submerged by 2100, affecting 55% of the population in the area, which would threaten the national food security if effective preventive measures are not taken. Soil salinization leads to serious consequences, especially in the context of population growth, which requires natural resources, especially food, which is directly related to the requirement of more cultivable land. Therefore, it is necessary to develop a global strategy to reduce the negative effects of Water salinity. Water salinity is the result of a range of phenomena, including irregular rainfall, evaporation, salinity of groundwater, flooding from storm surges, flooding of saline rivers, and the presence of a soluble salt source. Salinization can occur during soil formation by release of soluble salts either during weathering or by external natural inputs. Inappropriate agricultural practices, such as salt water irrigation in the absence of proper drainage, cause soil salinization.

### 1.3 ENVIRONMENT ISSUES

In general, shrimp culture needs regular minerals, chemicals, and antibiotics for shrimp productivity. But currently, this rate of operation has been drastically increased.

Further, high attention to conk and chemicals operation leads to the conformation of poisonous substances and also possible for declination of submarine species and impact on mortal health. The intensity of monoculture practice discharges large amounts of undressed waste water.

❖ Unmanaged orun-engineered culture practices retain ecological impacts that occurs

1. Oxygen insufficiency due to dissolved organic matter.
2. conformation of algae bloom due to accumulation of organic nutrients like nitrogen and phosphorus, further creates high biomass in the face water,
3. Unmanaged running of remainders from monoculture causes serious problems for mortal health, foliage and fauna, ecosystem and profitable development.

## 1.4 WATER QUALITY ANALYSIS

1. The water quality parameters were proposed to measure from the influent and effluent monoculture installations. The parcels which are going to measure were pH, total dissolved solids( TDS), electrical conductivity( EC), turbidity, saltiness, dissolved oxygen( DO), chlorides, sulphates, alkalinity, ammonia, nitrites, nitrates, natural oxygen demand( Duck) and Chemical oxygen demand (COD). And also, a water quality assessment was planned for monsoon and post-monsoon, for a better understanding of the contaminant rate of the monoculture ponds.
2. Essence traces can be linked and anatomized using high-pressure liquid chromatography (HPLC). High-performance liquid chromatography or high-pressure liquid chromatography (HPLC) is a chromatographic system that's used to separate a mixture of composites in logical chemistry and biochemistry so as to identify, quantify or purify the individual factors of the admixture.

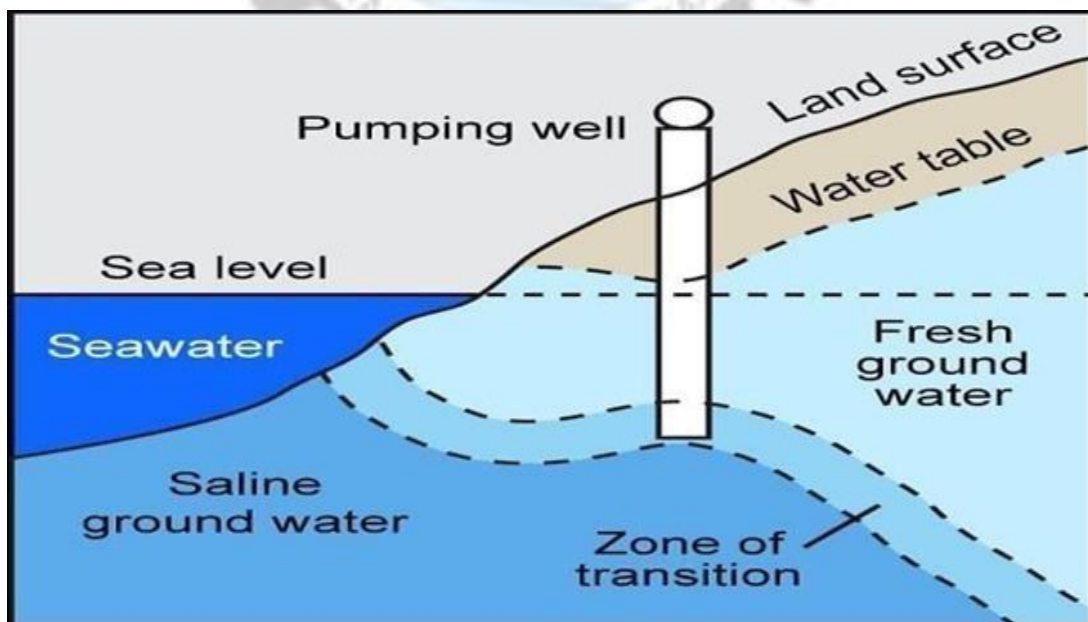


Figure 1 Salinity Intrusion in Coastal Area

## 2. LITERATURE SURVEY

The Vietnamese Mekong Delta is situated downstream of the Mekong River and covers an area of 39,734 km<sup>2</sup>. The average altitude in the region is 1–2 m above sea level. The region experiences a tropical climate, with rainy (from May to October) and dry seasons (from November to April). The average rainfall is 1400–2200 mm, of which 90–95% falls during the rainy season. The rivers, streams, and canals are dense, leading to abundant surface water in the Mekong Delta. The average annual flow volume is approximately 500 km<sup>3</sup>. The tides in the delta are mixed, diurnal and semi-diurnal, whose magnitude can reach up to 3 m. Usually, there are two troughs and two peaks during the day; however, the relative tide heights change every two weeks (Duy et al., 2021), i.e., when the first troughs decrease day by day, the other troughs increase, and vice versa. Apart from the tidal influence, this region is also influenced by river flooding. The flood season occurs from July to November, which inundates approximately 35– 50% (up to 4 m) of the delta's surface (Wassmann et al., 2019). The delta is home to nearly 20 million people (with a density of 500 people/km<sup>2</sup>). Four out of five people live in rural areas, and the workforce is mainly involved in agricultural practices. However, this region has been identified as a global “hot spot” that is susceptible to the effects of climate change, particularly rising sea levels, leading to increased salinity in the river systems. Studies have predicted that by 2050, the sea level will increase by 33 cm, and by 2100, it will increase by 1 m, which will lead to the submergence of at least 25% of agricultural land in the Mekong Delta, and ~ 75% of the current cultivated area will be affected by salinity in the dry season. This will cause ~40– 50% of the agricultural area to be affected by salt water even during the rainy season, seriously affecting rice crops. Therefore, water salinity needs to be monitored to form a sustainable agricultural development strategy.

A total of 39 automatic measurement stations were set up to measure the salinity in the study site. All of these stations were placed in the transitional area between water and land along the estuaries, rivers, and canals of the Mekong Delta to determine the electrical conductivity (EC), which is directly proportional to salinity. Twelve of the salinity measurement stations were established along



estuaries, including Co Chien, Cua Dai, Cua Tieu, Cua Soai Rap, and Ham Luong, while 25 stations were placed along rivers, including Cai Be, Cai Lon, Dong Dien, Dong Nai, Ganh Hao, Hau, Maspero, My Tho, Nhu Gia, Doc, Kien, Tien, Vam Co, Vam Co Dong, and Vam Co Tay; two stations were constructed along canals, including Rach Gia and Phung Hiep. At the location of automatic measurements, the soil had an average pH of 4.9–6.5, average  $\text{Cl}^-$  index of 0.05–0.25, and average  $\text{P2O5}$  values of 0.04–0.05. The data were collected daily during 2016–2020, and the noisy and missing data were filtered in the gathering process to avoid bias of machine learning models. All samples were collected in dry weather conditions, without cloud cover at the location of samples, so that satellite imagery data could be incorporated to estimate the operation of the models. These meteorological conditions were chosen for sampling to gather more acute data when freshwater flows from rivers would be narrowed (Habiba et al., 2015). In addition, the Mekong Delta is a large region and all areas cannot be imaged in one day; therefore, the measured salinity values were averaged over four months per year to synchronize the sample dataset and ensure that the entire study area was covered by the satellite orbit. Finally, 70% of the data were used for training, while 30% were applied to validate the models (Fig. 2).

**2.2.2. Satellite imagery and preprocessing**

Satellite imagery is one of the most common data sources used in many salinity intrusion studies (Tran et al., 2019), where the EC is used to identify potential salinity. Various studies have shown how the relationship between spectral bands and EC value can aid in the prediction of the level of salinity intrusion (Tran et al., 2019). Many studies have also used salinity indices, calculated from spectral band values, to predict the EC (Seifi et al., 2020). In this study, Landsat 8 OLI/TIRS, provided by USGS, was collected and atmospherically corrected by LaSRC using the CFMask algorithm (Vermote et al., 2018). Landsat 8 orbits the Earth in a sun-synchronous manner, close to a polar orbit, with a 16-day repeat cycle for temporal resolution. The Landsat 8 OLI/TIRS image has 11 spectral bands, including eight bands at 30 m spatial resolution, one panchromatic band at 15 m spatial resolution, and two TIRS bands at 100 m spatial resolution, which is sufficient for regional studies (Markham et al., 2018). More than 50 scenes were acquired with a cloud cover of less than 30%, during the period when the water samples were collected, to ensure consistency of the dataset. All sample data without the corresponding

satellite imagery were removed. 2.2.3. Salinity intrusion geodatabase a salinity intrusion database was established to improve the training dataset using indices that have been previously verified in numerous studies. Each index demonstrated a constraint on the EC. For example, (Nguyen et al., 2020b) built a regression model from 11 bands of Landsat 8 (B1 to B11) to estimate the EC. (Matinfar et al., 2020) used salinity indices (SI1 to SI11), normalized difference vegetation index, normalized difference salinity index, soil adjusted vegetation index, and vegetation soil salinity index to predict the EC. (Bouaziz et al., 2011) applied intensity indices and enhanced vegetation index to obtain the EC by multiple linear regression and other algorithms. Several specific indices have also been implemented to estimate the EC, such as ND23 and ND47 (Wu, 2019). DEM data were also added to this database. Finally, 31 indices were included in the analysis.

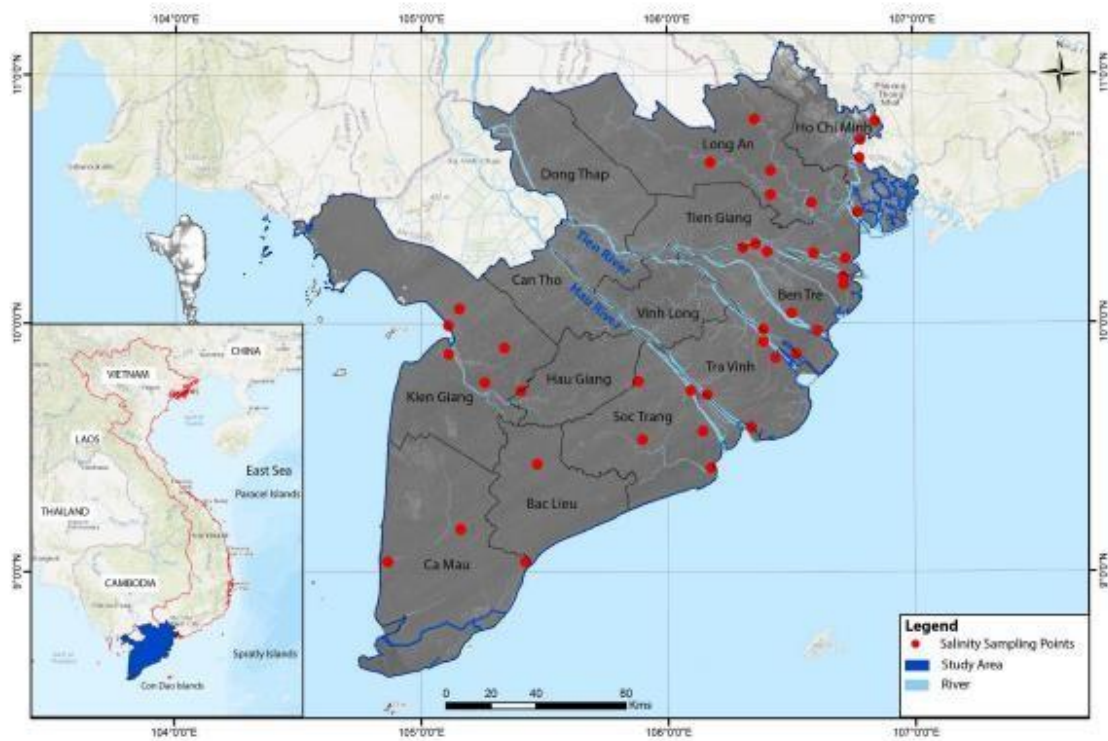


Figure 2 Location of the Mekong Delta of Vietnam.

## 2.1 WHAT IS MACHINE LEARNING

Machine learning is a type of artificial intelligence (AI) that allows computer system to automatically improve their performance on a specific task by learning from data, without being, explicitly programmed. In other words, is the science of

getting computers to learn and make decisions or predictions based on data, without being explicitly programmed. Machine learning algorithms are designed to analyze large datasets and identify patterns and relationships within the data. These algorithms learn from the patterns they identify and use this knowledge to make predictions or decisions on new data.

There are various types of machine learning algorithms, such as supervised learning, unsupervised learning, and reinforcement learning. Supervised learning involves training an algorithm on labeled dataset, while unsupervised learning involves training on an unlabeled dataset. Reinforcement learning involves training an algorithm to make decisions based on rewards or punishments.

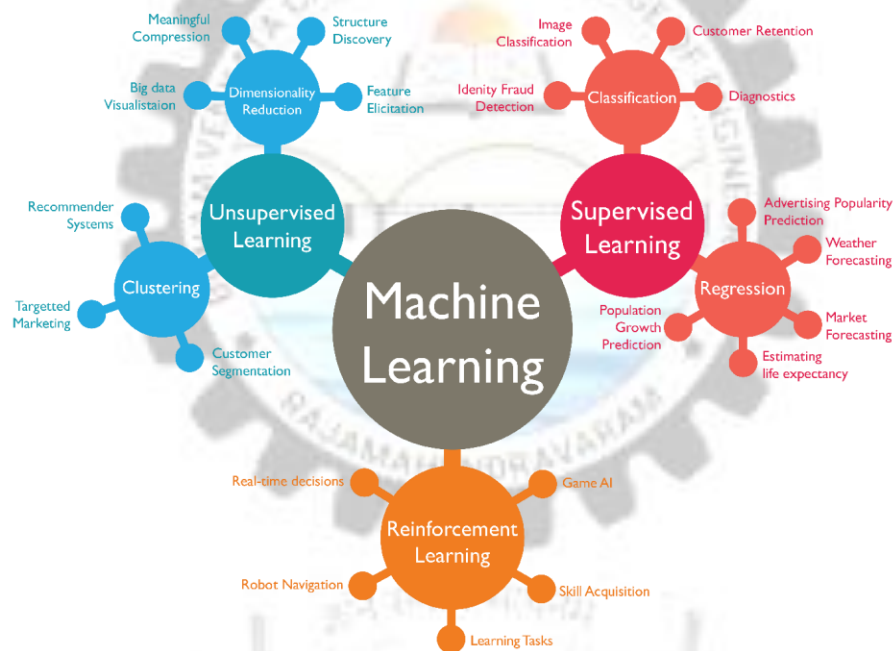


Figure 3 Machinelearning

## 2.2 TYPES OF MACHINE LEARNING

### 2.2.1 SUPERVISED LEARNING

Supervised learning is the type of machine learning in which machines are trained using well "labelled" training data, and on the basis of that data, machines predict the output. The labelled data means some input data is already tagged with the correct output.

In supervised learning, the training data provided to the machines work as the supervisor that teaches the machines to predict the output correctly. It applies the same concept as a student learns in the supervision of the teacher. Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable( $x$ ) with the output variable( $y$ ).

- **CLASSIFICATION:** When inputs are divided into two or more classes, and the learner must produce a model that assigns unseen inputs to one or more (multi-label classification) of these classes. This is typically tackled in a supervised way. Spam filtering is an example of classification, where the inputs are email (or other) messages and the classes are "spam" and "not spam".
- **REGRESSION:** Which is also a supervised problem. A case when the outputs are continuous rather than discrete.

### 2.2.2 UNSUPERVISED LEARNING

Unsupervised learning cannot be directly applied to a regression or classification problem because, unlike supervised learning, we have the input data but no corresponding output data. The goal of unsupervised learning is to find the underlying structure of the dataset, group that data according to similarities, and represent that dataset in a compressed format.

- Unsupervised learning is helpful for finding useful insights from the data.
- Unsupervised learning is much similar to how a human learns to think by their own experiences, which makes it closer to real AI.
- Unsupervised learning works on unlabeled and uncategorized data which makes unsupervised learning more important.
- In the real-world, we do not always have input data with the corresponding output so to solve such cases, we need unsupervised learning.

**CLUSTERING:** When a set of inputs is to be divided into groups. Unlike in classification, the group are not known beforehand, making this typically an unsupervised task.

**ASSOCIATION:** Association rules allows you to establish associations amongst data objects inside large databases, this is unsupervised technique is about discovering interesting relationships between variables in large datasets

For example, people that buy a new home most likely to buy new furniture. Market basket analysis is an example for association, it uses the previous history to predict the data.

### 2.2.3 REINFORCEMENT LEARNING

Reinforcement learning is an area of Machine Learning. It is about taking suitable action to maximize reward in a particular situation. It is employed by various software and machines to find the best possible behaviour or path it should take in a specific situation. Reinforcement learning differs from supervised learning in a way that in supervised learning the training data has the answer key with it so the model is trained with the correct answer itself whereas in reinforcement learning, there is no answer but the reinforcement agent decides what to do to perform the given task. In the absence of a training dataset, it is bound to learn from its experience.

The reinforcement learning system is comprised of four main components:

- Agent
- Environment
- Task
- Rewards

## 2.3 PYTHON PROGRAMMING LANGUAGE

Python is incredibly simple to inspect. As a deciphered language, it doesn't change code to get PC significant. Python is moreover, a raised level, all-around supportive programming language. Planners sorted out it to change into a chameleon of the programming scene.

In like manner, Python plans to pass on an even clearer and progressively genuine code for little expansions that connect similarly concerning more prominent ones, too.

You can offset Python with a Rubik's 3D shape: it has different sides to it so you



can bend and play around. The language is set up for finishing enormous proportions of PC structures to pass on advancement that can stagger you.

A couple of enchanting genuine elements, addressing the certifiable effect of this language, and what is Python utilized for:

- The watched BitTorrent began as a Python Program.
- The NSA uses Python for information appraisal and cryptography.
- Developers framed Youtube utilizing Python (among different dialects).
- Google isn't any progressively irregular to Python in like manner: the affiliation based its praised web search framework on it.

### 2.4 TEACHING MACHINES TO LEARN

PC based insight is a unique idea. It improves personalization and future inclination gauges. In the most recent decade, man-made reasoning has changed particular industry fields. It gave an open entryway for new, incomprehensible improvements to ascend from nothing. Without a doubt, not anything: Python.



*Figure 4 Bot*

Making modernized reasoning empowered programming sounds tangled. PC-based knowledge with Python instructs PCs to get from express models and review them, in the same way, individuals educate kids. In addition, Python AI is set up for

making figures, evaluating potential answers, and thus amazingly more!

Man-made reasoning is driven by the improvement of neural structures, one of the musings that answers an issue of what is Python utilized for. In the least complex terms, Python neural system is a structure including estimations subject to the human mind. With Python, engineers make induced structures and use them to cause machines to learn by taking a gander at models.

## **2.5 FOR WHAT REASON IS PYTHON THE BEST PROGRAMMING LANGUAGE FOR AI?**

The natural course of action of Python unequivocally underpins the formation of AI and ML. There are some particularly kept up assets and instructional exercises. They give bits of information concerning which Python libraries to use for modernized reasoning and critical learning.

Another enormous issue for what is Python utilized for is information on the board. Appropriately directing information in the present time of bleeding-edge improvement is major.

Individuals are obliged in this significant, man-made insight set up to arrange huge extents of complex information with high productivity and lower creation costs.

Since the emphasis of Python looks like English, it is sensibly, progressively direct to learning. Additionally, this language licenses preparing and overseeing complex frameworks.

## **2.6 PAST SUCCESSFUL PYTHON AI PROJECTS**

Making Python AI has as of late been shown to be altogether profitable. The voyaging business was refreshed when Skyscanner applied a free Python AI calculation. Expecting essentially no effort and high sufficiency, it evaluated the lead of new plane courses and wrapped up potential targets for adventurers.



*Figure 5 Bot2*

Another model, indicating that Python is the best programming language for AI, is its relationship to human organizations. Python AI undertakings are upsetting infection want and injury territory, making it less hard to follow patients' success and deal with it.

Likewise, Python urges thriving related applications to make. AiCure is one of the open advantageous applications that ensures patients recognize their prescriptions as grasped. This model is really what Python is utilized for: to improve advancement and our lives.

On this chance that you are essentially beginning to find a few solutions concerning AI in Python, it is impeccable to begin investigating the potential outcomes with the Keras library. It gives an improved chance of making Python neural structures. Beginning there, you should start looking into TensorFlow, PyTorch, or Theano.

## **2.7 HOW TO MAKE A BOT WITH PYTHON?**

Bots are programs for performing unequivocal assignments over the Internet. Such applications execute dreary activities a lot snappier than people. For example, Twitter is from time to time the objective of bots, sending the equivalent or relative messages a hundred times each day. By the by, bots can also be noteworthy for explicit or any help as they can make reactions to clients' data. Thus, client help winds up being dynamically beneficial.

Bots are one of the musings concerning what is Python utilized for. It is one of the essential tongues to use for making bots.

As an issue of first, noteworthiness, we should consider the potential open-source bot models:

- The python-rt bot is an outstanding bot structure for making Slack bots with Real-Time Messaging API over WebSockets.
- GitHub gives unfathomable points of interest for making bots, including code pieces and significant clues.
- ErrorFind is a chatbot for creating bots for Slack, Discord, and Hipchat. The basic objective of Err-bot is to permit individuals to convey their endeavours by controlling the given Python source code.

## 2.8 PYTHON LOVES WEB DEVELOPMENT

Web progress is an expansive idea. It combines all exercises performed to achieve goals. The multifaceted thought of this approach relies on the kind of thing made of.

What is Python utilized for web progress? It is a device for making back-end web applications. Django, Flask, and Falcon are the most praised structures that draftsmen use for motivation and recovering various things of code for their web experiences.

Web programs don't execute Python: they run JavaScript. You can utilize the PJs undertaking to assemble from Python to JavaScript. Considering that, most web apps include both JScript and Python. These codes execute on the server side.

## 2.9 INFORMATION MINING AND PYTHON

Information mining is a procedure of isolating gigantic databases to make propensity wants. This framework is abnormal. Researchers take a gander at huge amounts of data and base certain questions on them. Information mining combines assessment of social affiliations, awful conduct imaging, and so forth.

Something other than what's expected of what Python is utilized for is to filter through and clean information. It is considered an excellent appreciation for making differences in other programming vernaculars to do it. Also, AI with Python improves information evaluation with the utilization of estimations.

Python is famous for the full degree of structures, giving titanic measures of pre-shaped code bits that permit draftsmen to improve their undertakings. The corresponding applies to information mining. Here is a synopsis of most standard structures for driving information assessment:

- Numpy is the essential structure made available for numerical estimations in Python.
- Scikit-Learn is a Python AI system for beneficial information mining, permitting to play out the apostatize, bunching, model choice, preprocessing, and demand structures.

### 2.10 GUI-BASED DESKTOP PROGRAMS

The graphical UI is in like way, what is Python utilized for. GUI lets individuals partner with PCs utilizing visual parts, for example, pictures or pictures, rather than content-based solicitations. There are different modules open for making a GUI with Python. Consequently, we quickly show the most routinely utilized ones:

- Tkinter is a worked in Python interface. This GUI toolbox runs on the entirety of the most prominent stages like Microsoft, Linux, and Mac OS X.
- PyGTK is a free toolbox that assists with making graphical interfaces. WxPython is a lock for the cross-stage GUI toolbox and wxWidgets. From the earliest starting point, fashioners made Python utilizing C++. Regardless, Python uprooted C++.
- Kivy is a Python library for conveying versatile applications and multi-contact application programming. It is a remarkable decision for depicting UI and joint endeavours.



## 2.11 MAKE GAMES AND 3D GRAPHICS WITH PYTHON

One, the outline concerning what is Python utilized for notice that it is moreover an appropriate open door for a game's unexpected turn of events. In a little while, there are various structures and instruments for the game and reasonable creation:

PyGame is no uncertain the crucial decision for specific, engineers utilizing Python. The eminent library offers modules to pass on completely, including games and instinctive media programs. Moreover, understudies ought to consider this system as the given models help to acknowledge game movement more. Take the necessary steps not to anticipate that it should clarify each framework a smidgen at once. Yet the library is more predominant than the typical beginning stage. PyOpenGL is a framework for OpenGL apps. It contains different events on how to make 3D models.

Panda3D is an open-source structure for 3D rendering and game-unanticipated turn of events. Blender is a multi-sided instrument for making 3D models. The gadgets apply an installed Python translator for conveying 3D games. Arcade is a Python library for conveying 2D games into the world.

## 2.12 TERMINATIONS

Clarifying what is Python utilized for isn't for every circumstance essential. There is a tremendous measure of layers to take off, to give signs of progress, and take a gander at the limits of Python.

## 2.13 PYTHON LIBRARIES

Python library is a collection of functions and methods that allows you to perform many actions without writing your code. For example, if you are working with data, numpy, pandas, etc. are the libraries you must know. They have very convenient data transformation function that will save you life to do small tricks.

## 2.14 PYTHON LIBRARIES THAT USED IN MACHINE LEARNING

- Numpy
- Pandas
- Matplotlib
- Scikit-learn

### ***NUMPY***

Numpy is a very popular python library for large multi-dimensional array and matrix processing, with the help of a large collection of high-level mathematical Functions. It is very useful for fundamental scientific computations in Machine learning. It is particularly useful for linear algebra. Fourier transform and random number capabilities, high-end libraries like tensorflow uses numpy internally for manipulation of tensors.

### ***PANDAS***

Pandas is a popular library for data analysis. It is not directly related to machine learning. As we know that the database must be prepared before training in this case, pandas comes handy as it was developed, specifically for data extraction and preparation. It provides high-level data structures and wide variety tools for data analysis. It provides many inbuilt methods for grouping, combining and filtering data.

### ***MATPLOTLIB***

Matplotlib is a very popular python library for data visualization. Like pandas, it is not directly related to Machine Learning. It particularly comes in handy when a programmer wants to visualize the patterns in the data. It is a 2D plotting library used for creating 2D graphs and plots. A module named pyplot makes it easy for programmers for plotting as it provides features to control line styles, font properties, formatting axes, etc. it provides various kinds of graphs and plots for

data visualization, viz., histogram, error charts, bar charts, etc...

### ***SCIKIT-LEARN***

Scikit-learn provides range of supervised and unsupervised learning algorithms via a consistent interface in python.

## **2.15 SYSTEM STUDY**

### ***2.15.1 FEASIBILITY STUDY***

The feasibility of the project is analyzed in this phase and a business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis, the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden on the company. For feasibility analysis, some understanding of the major requirements for the system is essential.

Three key considerations involved in the feasibility analysis are,

- ECONOMICAL FEASIBILITY
- TECHNICAL FEASIBILITY
- SOCIAL FEASIBILITY

### ***2.15.2 ECONOMICAL FEASIBILITY***

This study was carried out to check the economic impact that the system will have on the organization. The amount of funds that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus, the developed system was developed as well within the budget, and this was achieved because most of the technologies used are freely available. Only the customized products had to be purchased.

### ***2.15.3 TECHNICAL FEASIBILITY***

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand for available technical resources. This will lead to high demands on available technical resources. This will lead to high demands being placed on the client. The

developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

### ***2.15.4 SOCIAL FEASIBILITY***

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. The user must not feel threatened by the system. Instead must accept it as a necessity. The level of acceptance by the users solely depends on the methods that are employed to educate the user about the system and to make him familiar with it. His level of confidence must be raised so that he is also able to make some constructive criticism, which is welcomed, as he is the final user of the system.

## **2.16 FUNCTIONAL AND NON-FUNCTIONAL REQUIREMENTS**

### ***2.16.1 FUNCTIONAL REQUIREMENTS***

Functional requirements explain what has to be done by identifying the necessary task, action or activity that must be accomplished. Functional requirements analysis will be used as the top-level function for functional analysis.

- Customer Data
- Action Against Fraudulent Graph Analysis

### ***2.16.2 USER REQUIREMENTS***

User Requirements Analysis is the process of determining user expectations for a new or modified product. These features must be quantifiable, relevant and detailed. The main user requirements of our project are as follows:

Internet Facility/ LAN Connection

- Sending file
- Mode selection
- IP address of a particular system
- Destination Path

### ***2.16.3 NON-FUNCTIONAL REQUIREMENTS***

Non-functional requirements describe the general characteristics of a system. They are also known as quality attributes. Some typical non-functional requirements are Performance, Response Time, Throughput, Utilization, and Scalability.

Performance:

The performance of a device is essentially estimated in terms of efficiency, effectiveness and speed.

1. Short response time for a given piece of work.
2. High throughput (rate of processing work)



### 3 SYSTEM SPECIFICATION

#### 3.1 HARDWARE REQUIREMENTS

- ❖ Processor : Intel Core i3 2.4 GHz.
- ❖ Hard Disk : 150 GB.
- ❖ Monitor : 14” Colour Monitor.
- ❖ Ram : 4 GB

#### 3.2 SOFTWARE REQUIREMENTS

- ❖ Operating System: Windows 7 Ultimate.
- ❖ Coding Language: Python.
- ❖ IDE : Jupyter notebook.
- ❖ Packages : Numpy, Pandas, OS, Matplotlib, Sklearn



## 4. SYSTEM ANALYSIS

### 4.1 EXISTING SYSTEM

Salinity intrusion prediction using remote sensing and machine learning in data-limited regions: A case study in Vietnam's Mekong Delta, This study aims to establish a novel framework for monitoring salinity intrusion using remote sensing and machine learning. It focuses on the salinity intrusion in soil, which affects water availability, food security, human health, etc. Numerous algorithms have been implemented to find the best solution for this issue, including Xgboost (XGR), Gaussian processes, support vector regression, deep neural networks, and the grasshopper optimization algorithm (GOA). A total of 143 samples collected from 2016 to 2020 at 39 measurement stations were divided into two sets: 70% training and 30% testing. Thirty- one independent variables were used to develop the model. Vietnam's Mekong Delta, where the salinity intrusion problem is becoming increasingly serious due to global warming and demographics was selected as the study area of the proposed models was compared and evaluated by applying various statistical indices such as the root mean square error, coefficient of determination ( $R^2$ ), and mean absolute error. The results show that the prediction model was built successfully by wielding data from the implemented salinity measurement stations, and the XGR-GOA model was better than the other models ( $R^2 = 0.86$ ,  $RMSE = 0.076$ , and  $MAE = 0.065$ ). This finding demonstrates the feasibility of estimating and monitoring salinity intrusion in data-limited regions by integrating optical satellite images and machine learning, which are easily and cost-effectively obtainable. The proposed conceptual methodology in our study is novel and provides additional useful information for the monitoring and management of salinity intrusion not only in Vietnam's Mekong Delta but also in other sites that have similar natural conditions.

### DISADVANTAGES OF EXISTING SYSTEM

- Limited interpretability.
- Gaussian processes can be computationally expensive, especially for large datasets.

- SVR requires tuning of hyperparameters and Sensitivity to Hyperparameters.
- Vulnerability to adversarial attacks.
- GOA may not be robust to changes in problem formulation or changes in the environment, as it relies on a fixed set of rules and control parameters.

### 4.2 PROPOSED SYSTEM

Salinity intrusion prediction using machine learning the aim of this model is to identify the increase of salinity year by year and predict for future years. We collect water samples data from the past 7-8 years where saltwater is intruding into fresh aquifers so that we can identify

How rapidly the intrusion is done year wise and we can predict how much salt concentration will be increasing in future years. Using a linear regression algorithm, we can obtain the output for the given data sets in a simple and efficient way.

### 4.3 SYSTEM ARCHITECTURE

The system architecture gives an overview of the working of the system. The working of this system is described as follows:

Dataset collection is collecting data which contains samples (Ph, B.O.D, D.O, Temp etc.). The attributes selection process selects the useful attributes for the prediction of salinity intrusion. After identifying the available data resources, they are further selected, cleaned, and made into the desired form. Different classification techniques as stated will be applied to pre-processed data to predict the accuracy of salinity intrusion.

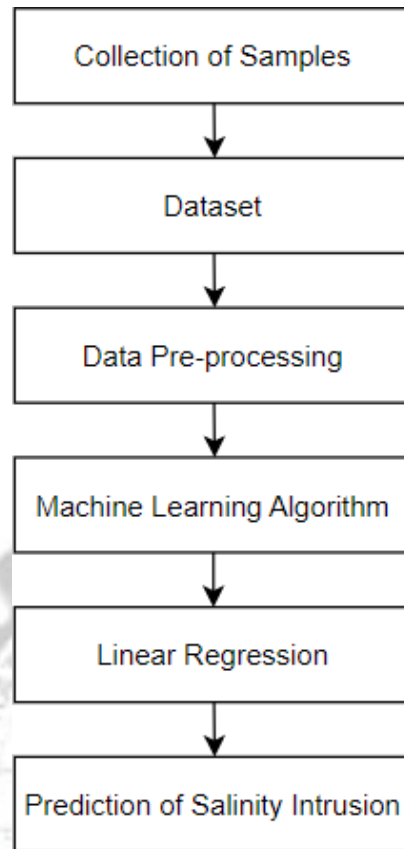


Figure 6 System Architecture

This system is implemented using the following modules.

- 1.) Collection of the Dataset
- 2.) Selection of attributes
- 3.) Data Pre-Processing
- 4.) Balancing of Data
- 5.) Salinity Prediction

### ***Collection of the dataset***

Originally, we collect a dataset for our saltwater prediction system. After the collection of the dataset, we resolve the dataset into training data and testing data. The training dataset is used for prediction model literacy and testing data is used for assessing the prediction model. For this design, 70 of training data is used and 30 of data is used for testing. The dataset used for this design is saltwater prediction. The dataset consists of 12 attributes, out of which, 8 attributes are used for the system.

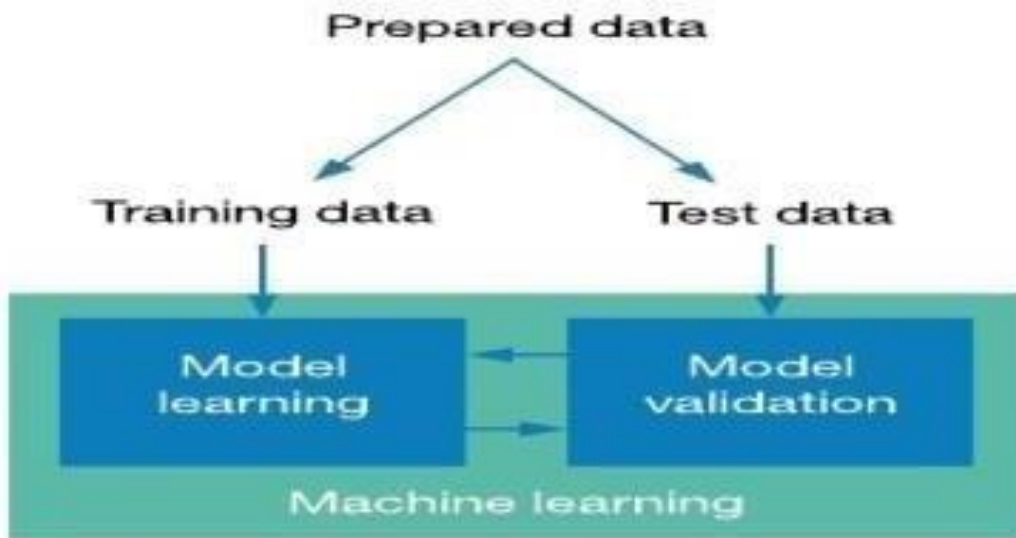


Figure 7 Collection of the Dataset

### ***Selection of Attributes***

Attribute or Feature selection includes the selection of appropriate attributes for the prediction system. This is used to increase the efficiency of the system. Various attributes of the station code, Location, State, D.O (mg/l), PH, Conductivity, B.O.D (mg/l), etc are selected for the prediction.

### ***Data Pre-processing***

Data processing is an important step in the creation of a machine literacy model. Originally, data may not be clean or in the needed format for the model which can beget deceiving issues. In the processing of data, we transfigure data into our needed format. It's used to deal with noises, duplicates, and missing values of the dataset. Data-processing has conditioning like importing datasets, unyoking datasets, trait scaling, etc. Pre-processing of data is needed for perfecting the delicacy of the model.



*Figure 8 Data Pre-processing*

### **BALANCING OF DATA**

Imbalanced datasets can be balanced in two ways. They are Under Sampling and Over Sampling

(a) Under Sampling:

In Under Sampling, dataset balance is done by the reduction of the size of the sample class. This process is considered when the amount of data is adequate.

(b) OverSampling:

In Over Sampling, dataset balance is done by increasing the size of the scarce samples. This process is considered when the amount of data is inadequate.

### **4.4 SALINITY PREDICTION**

Various machine learning algorithms like Linear Regression are used for classification. Comparative analysis is performed among algorithms and the algorithm that gives the highest accuracy is used for salinity prediction.

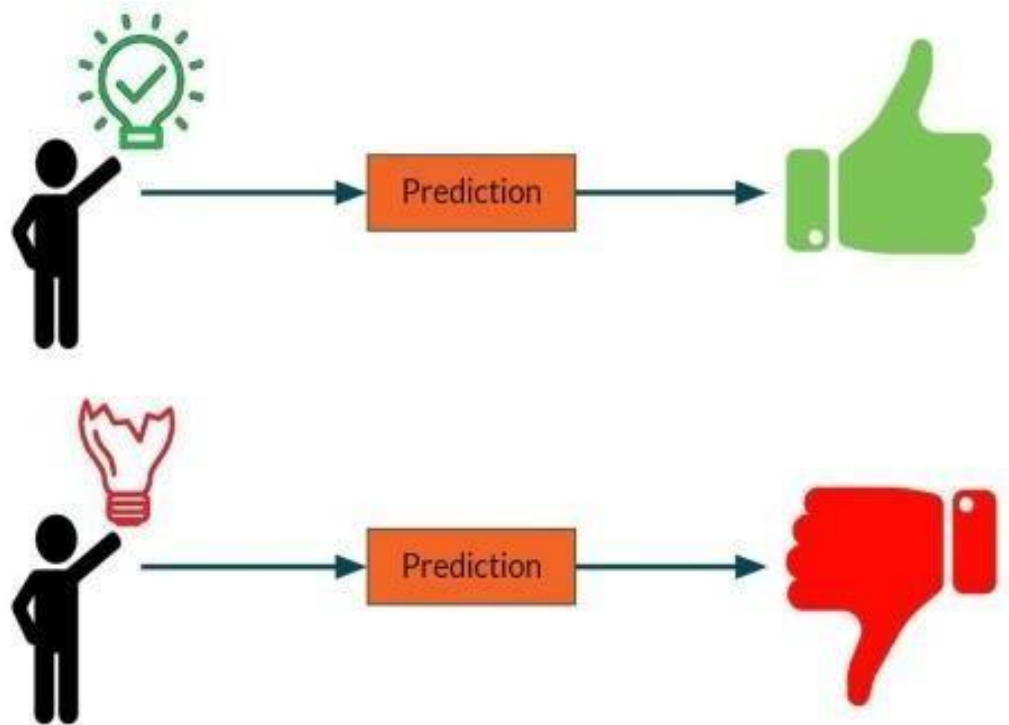


Figure 9 Salinity Prediction

#### 4.4.1 ALGORITHM

Linear Regression is a machine learning algorithm based on supervised learning. It performs a regression task. Regression models a target prediction value based on independent variables. It is mostly used for finding out the relationship between variables and forecasting. Different regression models differ based on – the kind of relationship between dependent and independent variables they are considering, and the number of independent variables getting used.

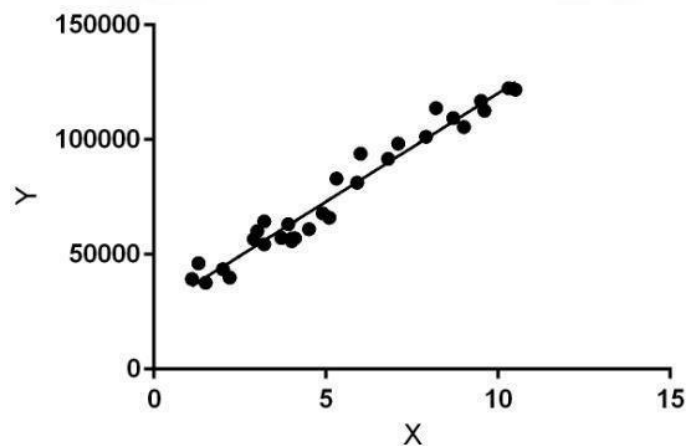


Figure10 Linear Regression Sample Graph



Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x). So, this regression technique finds out a linear relationship between x (input) and y (output). Hence, the name is Linear Regression.

In the figure above, X (input) is the work experience and Y (output) is the salary of a person. The regression line is the best fit line for our model.

Hypothesis function for Linear Regression:

$$y = \theta_1 + \theta_2 \cdot x$$

While training the model we are given:

X: input training data (univariate one input variable (parameter))

Y: labels to data (supervised learning)

When training the model – it fits the best line to predict the value of y for a given value of x. The model gets the best regression fit line by finding the best  $\theta_1$  and  $\theta_2$  values.

$\theta_1$ : intercept

$\theta_2$ : coefficient of x

Once we find the best  $\theta_1$  and  $\theta_2$  values, we get the best fit line. So when we are finally using our model for prediction, it will predict the value of y for the input value of x.

How to update  $\theta_1$  and  $\theta_2$  values to get the best fit line? Cost Function (J):

By achieving the best-fit regression line, the model aims to predict the y value such that the error difference between the predicted value and the true value is minimum. So, it is very important to update the  $\theta_1$  and  $\theta_2$  values, to reach the best value that minimizes the error between the predicted y value (pred) and true y value (y).

$$J = \frac{1}{n} \sum_{i=1}^n (pred_i - y_i)^2$$

Cost function (J) of Linear Regression is the Root Mean Squared Error (RMSE) between predicted y value (pred) and true y value (y).

Gradient Descent:

To update  $\theta_1$  and  $\theta_2$  values in order to reduce the Cost function (minimizing RMSE value) and achieve the best fit line the model uses Gradient Descent. The idea is to start with random  $\theta_1$  and  $\theta_2$  values and then iteratively update the values, reaching minimum cost.

### **Implementing Linear Regression**

The process takes place in the following steps:

Loading the Data Exploring the Data Slicing the Data Train and Split Data  
Generate the Model Evaluate the accuracy

- **Loading the Data:**

We can start with the basic diabetes data set that is already present in the sklearn (Scikit-learn) data sets module to begin our journey with linear regression.

- **Exploring The Data:**

After we are done loading the data, we can start exploring by simply checking the labels by using the following code.

- **Splitting The Data:**

We will split the data into train and test sets.

- **Generating the model:**

The next part involves generating the model, which will include importing the linear model from the evaluation to evaluate the accuracy of the model, we will use the mean squared error from the Scikit-learn.

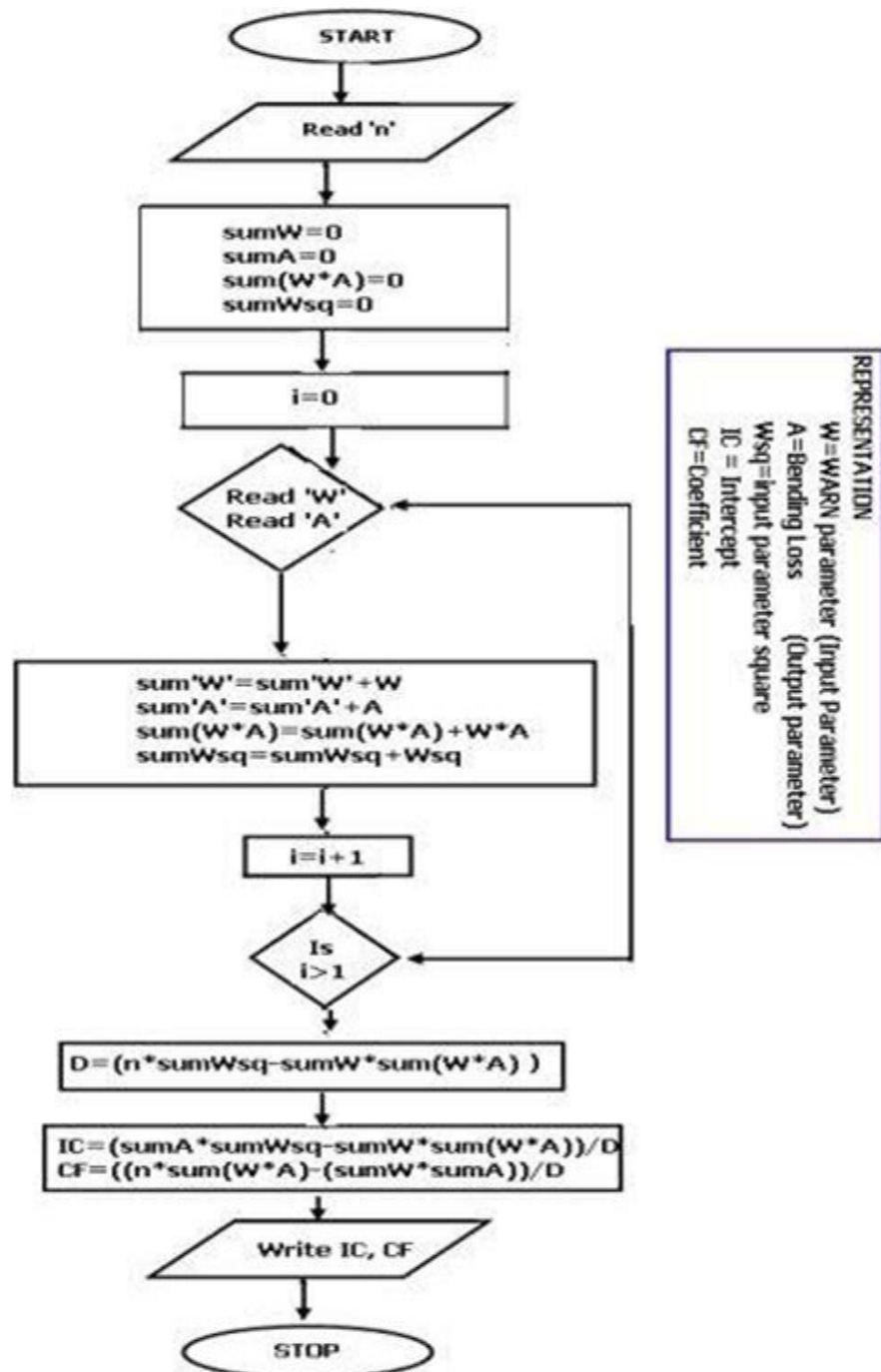


Figure 11 Linear Regression Flow Chart

## 5. SYSTEM DESIGN

### 5.1 UML DIAGRAM

UML stands for Unified Modeling Language. UML is a standardized general-purpose modelling language in the field of object-oriented software engineering. The standard is managed and was created by, the Object Management Group.

The goal is for UML to become a common language for creating models of object-oriented computer software. In its current form, UML is comprised of two major components: a Meta- model and a notation. In the future, some form of method or process may also be added to; or associated with, UML.

The Unified Modeling Language is a standard language for specifying, visualizing, constructing and documenting the artifacts of software systems, as well as for business modelling and other non-software systems.

The UML represents a collection of best engineering practices that have proven successful in the modelling of large and complex systems.

UML is a very important part of developing object-oriented software and the software development process. The UML uses mostly graphical notations to express the design of software projects.

#### GOALS:

The primary goals in the design of the UML are as follows:

1. Provide users with a ready-to-use, expressive visual modelling Language so that they can develop and exchange meaningful models.
2. Provide extensibility and specialization mechanisms to extend the core concepts.
3. Be independent of particular programming languages and development processes.
4. Provide a formal basis for understanding the modelling language.
5. Encourage the growth of the OO tools market.

6. Support higher-level development concepts such as collaborations, frameworks, patterns and components.

### 5.1.1 USE CASE DIAGRAM

A use case diagram in the Unified Modeling Language (UML) is a type of behavioural diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. The roles of the actors in the system can be depicted.

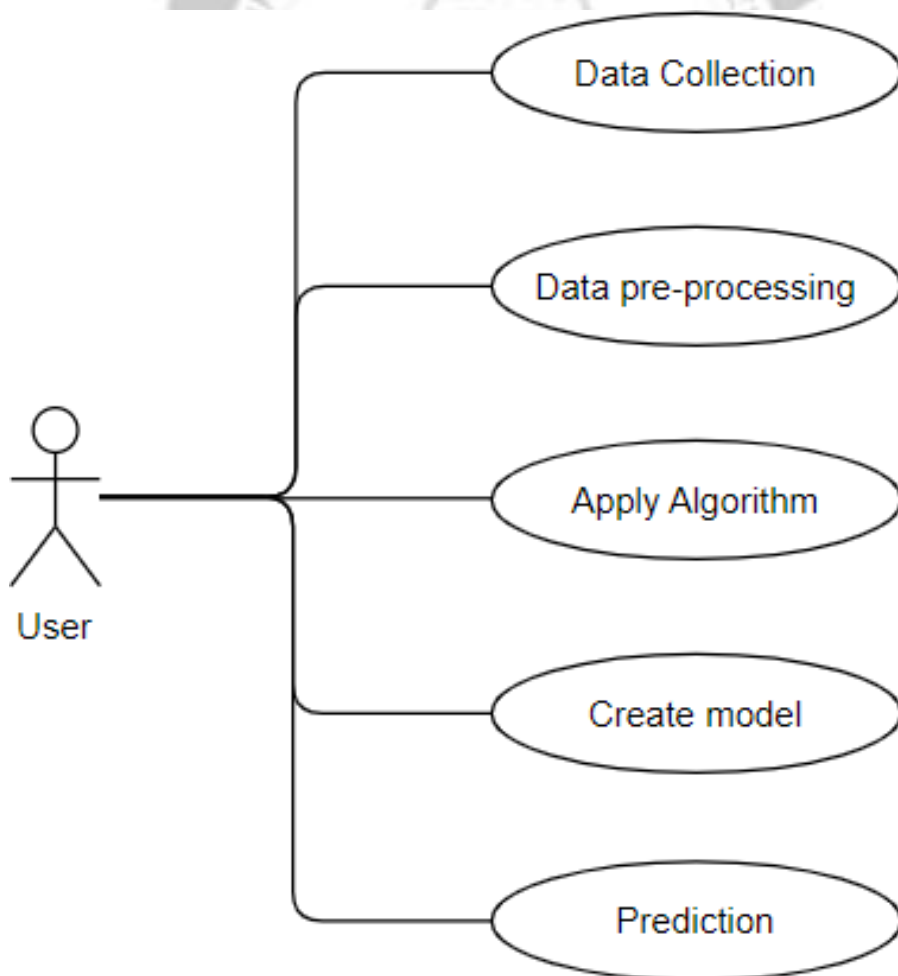


Figure 12 Use Case Diagram

### 5.1.2 DATA FLOW DIAGRAM

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one.

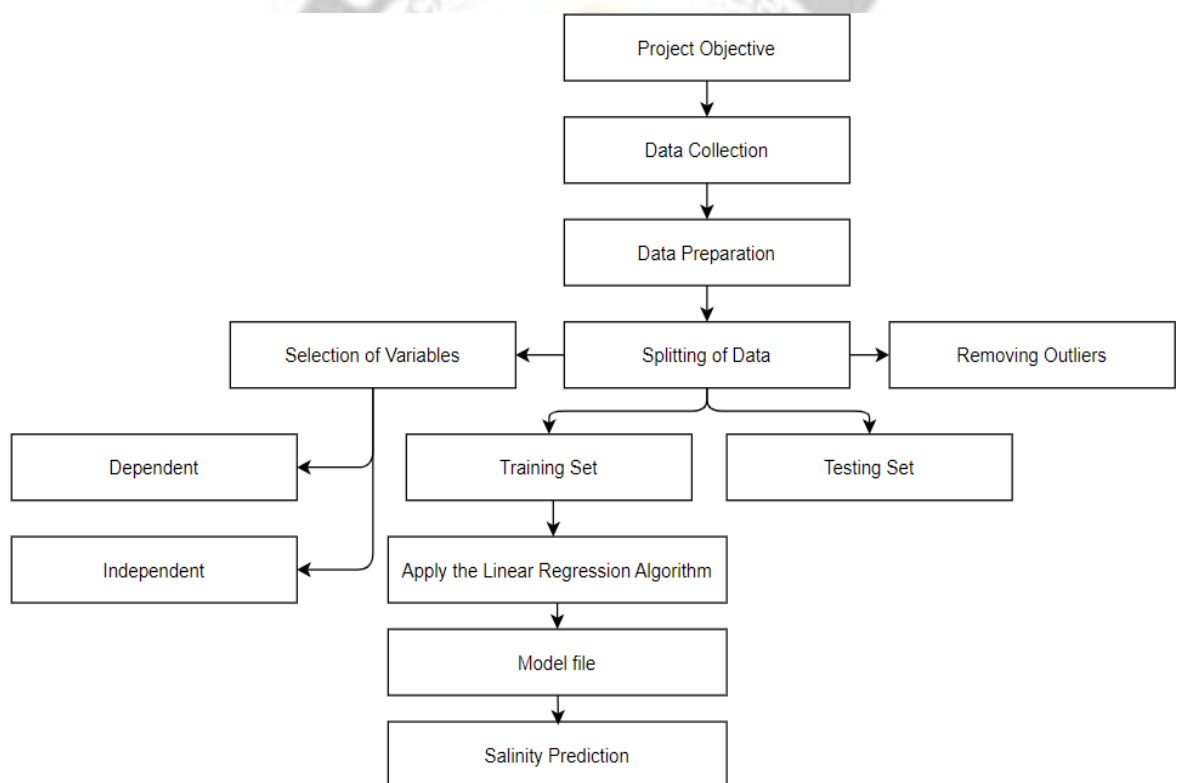


Figure 13 Data Flow Diagram



### 5.1.3 SEQUENCE DIAGRAM

A sequence diagram in Unified Modeling Language (UML) is a kind of interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.

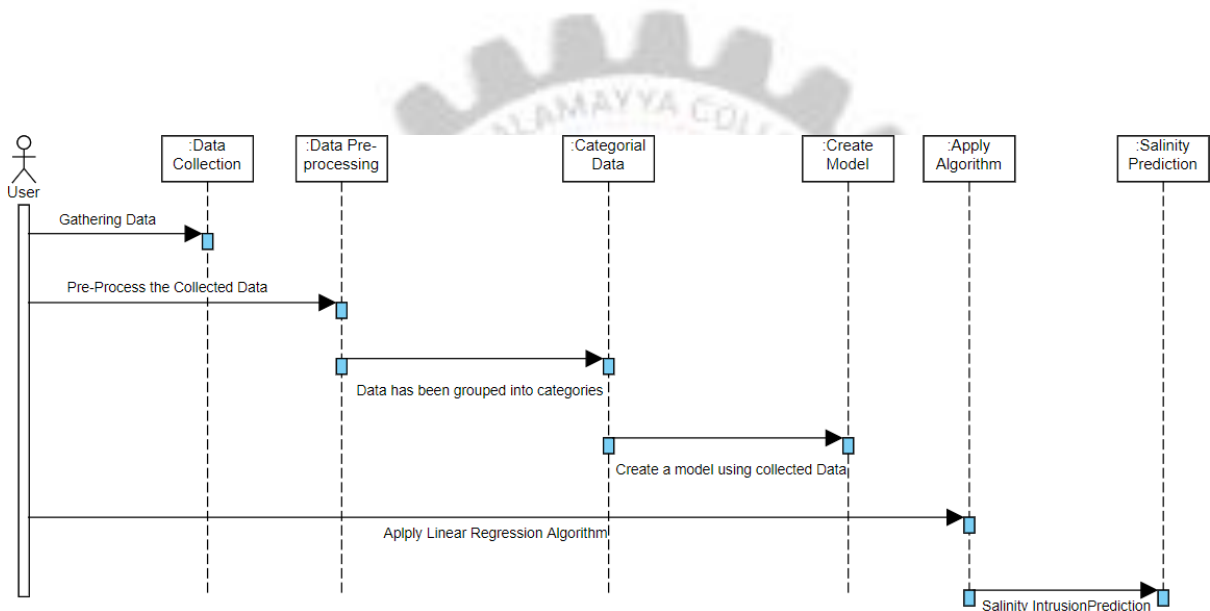


Figure 14 Sequence Diagram

## 6. IMPLEMENTATION AND CODING

```

import numpy as np # linear algebra
import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)#
Input data files are available in the "../input/" directory.
# For example, running this (by clicking run or pressing Shift+Enter) will list the files in the
input directory

import os

print(os.listdir("C:\\Users\\HP\\Downloads\\Salinity Intusion \\Salinity Intrusion "))

data=pd.read_csv("C:\\Users\\HP\\Downloads\\Salini
Ty Intusion \\Salinity
Intrusion\\water_dataX.csv",encoding="ISO-8859-1")

data.fillna(0, inplace=True)
data.head()
data.dtypes
#conversions
data['Temp']=pd.to_numeric(data['Temp'],errors='coerce')

data['D.O. (mg/l)']=pd.to_numeric(data['D.O. (mg/l)'],errors='coerce')
data['PH']=pd.to_numeric(data['PH'],errors='coerce')
data['B.O.D. (mg/l)']=pd.to_numeric(data['B.O.D. (mg/l)'],errors='coerce')

data['CONDUCTIVITY(μmhos/cm)']=pd.to_numeric(data['CONDUCTIVITY
(μmhos/cm)'],errors='coerce')

data['NITRATENAN N+ NITRITENANN (mg/l)']=pd.to_numeric(data['NITRATENAN N+
NITRITENANN (mg/l)'],errors='coerce')

data['TOTAL COLIFORM (MPN/100ml)Mean']=pd.to_numeric(data['TOTAL COLIFORM
(MPN/100ml)Mean'],errors='coerce')

data.dtypes

```

```

#initialization
start=2
end=1779
station=data.iloc [start:end ,0]

location=data.iloc [start:end ,1]

state=data.iloc [start:end ,2]

do= data.iloc [start:end ,4].astype(np.float64)
value=0
ph = data.iloc[ start:end,5]

co = data.iloc [start:end ,6].astype(np.float64)
year=data.iloc[start:end,11]
tc=data.iloc [2:end ,10].astype(np.float64)

bod = data.iloc [start:end ,7].astype(np.float64)
na= data.iloc [start:end ,8].astype(np.float64)
na.dtype
data.head()
data=pd.concat([station,location,state,do,ph,co,bod,na,tc,year],axis=1) data.
columns = ['station','location','state','do','ph','co','bod','na','tc','year']
#calulation of Ph
data['npH']=data.ph.apply(lambda x: (100 if (8.5>=x>=7)

else(80 if (8.6>=x>=8.5) or (6.9>=x>=6.8)
else(60 if (8.8>=x>=8.6) or (6.8>=x>=6.7)
else(40 if (9>=x>=8.8) or (6.7>=x>=6.5)

else 0))))))

#calulation of dissolved oxygen

```

```
data['ndo']=data.do.apply(lambda x:(100 if (x>=6)
                                else(80 if (6>=x>=5.1)
                                else(60 if (5>=x>=4.1)
                                else(40 if (4>=x>=3)
                                else 0))))))
```

#calculation of total coliform

```
data['nco']=data.tc.apply(lambda x:(100 if (5>=x>=0)
                                else(80 if (50>=x>=5)
                                else(60 if (500>=x>=50)
                                else(40 if (10000>=x>=500)
                                else 0))))))
```

#calc of B.D.O

```
data['nbdo']=data.bod.apply(lambda x:(100 if (3>=x>=0)
                                else(80 if (6>=x>=3)
                                else(60 if (80>=x>=6)
                                else(40 if (125>=x>=80)
                                else 0))))))
```

#calculation of electrical conductivity

```
data['nec']=data.co.apply(lambda x:(100 if (75>=x>=0)
                                else(80 if (150>=x>=75)
                                else(60 if (225>=x>=150)
                                else(40 if (300>=x>=225)
                                else 0))))))
```

#Calulation of nitrate

```
data['nna']=data.na.apply(lambda x:(100 if (20>=x>=0)
                                else(80 if (50>=x>=20)
                                else(60 if (100>=x>=50)
                                else(40 if (200>=x>=100)
                                else 0))))))
```

```
data.head()
data.dtypes

data['wph']=data.npH * 0.165

data['wdo']=data.ndo * 0.281

data['wbdo']=data.nbdo * 0.234

data['wec']=data.nec* 0.009

data['wna']=data.nna * 0.028

data['wco']=data.nco * 0.281
data['wqi']=data.wph+data.wdo+data.wbdo+data.wec+data.wna+data.wco
data
#calculation overall wqi for each year
ag=data.groupby('year')['wqi'].mean()
ag.head()
data=ag.reset_index(level=0,inplace=False)
data
#visualizing the filtered data
year=data['year'].values
AQI=data['wqi'].values
data['wqi']=pd.to_numeric(data['wqi'],errors='coerce')

data['year']=pd.to_numeric(data['year'],errors='coerce')
import matplotlib.pyplot as plt
plt.rcParams['figure.figsize'] = (20.0, 10.0)
from mpl_toolkits.mplot3d import Axes3D
fig = plt.figure()
ax = Axes3D(fig)
ax.scatter(year,AQI, color='red')
plt.show()
```



```
data
data = data[np.isfinite(data['wqi'])]
data.head()
#scatter plot of data points
cols =['year']
y = data['wqi']
x=data[cols]
plt.scatter(x,y)
plt.show()
import matplotlib.pyplot as plt
data=data.set_index('year')
data.plot(figsize=(15,6))
plt.show()
from sklearn import neighbors,datasets
data=data.reset_index(level=0,inplace=False)
data
#using linear regression to predict

from sklearn import linear_model

from sklearn.model_selection import train_test_split
cols =['year']
y = data['wqi']
x=data[cols]
reg=linear_model.LinearRegression()
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=4)
reg.fit(x_train,y_train)
a=reg.predict(x_test)a
y_test

from sklearn.metrics import mean_squared_error
print('mse:%.2f'%mean_squared_error(y_test,a))
dt=pd.DataFrame({'Actual': y_test, 'Predicted': a})
#using gradient descent to optimize it further
```

```

x = (x - x.mean()) / x.std()
x = np.c_[np.ones(x.shape[0]), x]
x_alpha = 0.1 #Step size
iterations = 3000 #No. of iterations
m = y.size #No. of data points
np.random.seed(4) #Setting the seed
theta = np.random.rand(2) #Picking some random values to start with
def gradient_descent(x, y, theta, iterations, alpha):
    past_costs = []
    past_thetas = [theta]
    for i in range(iterations):
        prediction = np.dot(x, theta)
        error = prediction - y
        cost = 1/(2*m) * np.dot(error.T, error)
        past_costs.append(cost)
        theta = theta - (alpha * (1/m) * np.dot(x.T, error))
        past_thetas.append(theta)
    return past_thetas, past_costs
past_thetas, past_costs = gradient_descent(x, y, theta, iterations, alpha)
theta = past_thetas[-1]
#Print the results...
print("Gradient Descent: {:.2f}, {:.2f}".format(theta[0], theta[1]))
plt.title('Cost Function J')
plt.xlabel('No. of iterations')
plt.ylabel('Cost')
plt.plot(past_costs)
plt.show()#prediction of
january(2013-2015) across
indiaimport numpy as np
newB=[74.76, 2.13]
def rmse(y,y_pred):
    rmse= np.sqrt(sum(y-y_pred))
    return rmse

```

```
y_pred=x.dot(newB)
dt = pd.DataFrame({'Actual': y,
'Predicted': y_pred})
dt=pd.concat([data, dt], axis=1)
dt
#testing the accuracy of the model
from sklearn import metrics
print(np.sqrt(metrics.mean_squared_error(y,y_pred)))
#plotting the actual and predicted results
x_axis=dt.yeary_axis=dt.Actual y1_axis=dt.Predicted
plt.scatter(x_axis,y_axis)
plt.plot(x_axis,y1_axis,color='r')
plt.title("linear regression")
plt.show()
```



## 7. SYSTEM TESTING

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, sub-assemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the software system meets its requirements and user expectations and does not fail unacceptably. There are various types of tests. Each test type addresses a specific testing requirement.

### ❖ TYPES OF TESTS

- ***UNIT TESTING***

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application. It is done after the completion of an individual unit, before integration. This is structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at the component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

- ***INTEGRATION TESTING***

Integration tests are designed to test integrated software components to determine if they run as one program. Testing is event-driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfied, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

- ***FUNCTIONAL TESTING***

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centred on the following items:

**Valid Input:** identified classes of valid input accepted.

**Invalid Input:** identified classes of invalid input must be rejected. Functions: identified functions must be exercised

**Output:** identified classes of application outputs must be executed.

**Systems/Procedures:** interfacing systems or procedures must be invoked.

Organization and preparation of functional tests focused on requirements, key functions, or special test cases. In addition, systematic coverage about identifying business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

- ***SYSTEM TESTING***

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration-oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

- ***WHITEBOX TESTING***

White Box Testing is testing in which the software tester knows the inner workings, structure and language of the software, or at least its purpose. It is a purpose. It is used to test areas that cannot be reached from a black-box level.

- ***BLACK BOX TESTING***

Black Box Testing is testing the software without any knowledge of the inner

workings, structure or language of the module being tested. Black box tests, like most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is testing in which the software under test is treated, as a black box. You cannot "see" into it. The test provides inputs and responds to outputs without considering how the software works.

- ***UNIT TESTING***

Unit testing is usually conducted as part of a combined code and unit test phase of the software lifecycle, although it is not uncommon for coding and unit testing to be conducted as two distinct phases. Test strategy and approach Field testing will be performed manually and functional tests will be written in detail.

Test objectives

- All field entries must work properly.
- Pages must be activated from the identified link.
- The entry screen, messages and responses must not be delayed.
- Features to be tested
- Verify that the entries are of the correct format
- No duplicate entries should be allowed
- All links should take the user to the correct page.

- ***INTEGRATION TESTING***

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results: All the test cases mentioned above passed successfully. No defects were encountered.



- **ACCEPTANCE TESTING**

User Acceptance Testing is a critical phase of any project and requires significant participation by the end-user. It also ensures that the system meets the functional requirements

Test Results: All the test cases mentioned above passed successfully. No defects encounter

## 7.1 TEST CASES

```

In [101]: #using linear regression to predict
from sklearn import linear_model
from sklearn.model_selection import train_test_split

In [102]: cols = ["year"]

In [103]: y = data["sal"]
xdata[cols]

In [104]: reg = linear_model.LinearRegression()
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.4, random_state=5)

In [105]: reg.fit(x_train, y_train)

Out[105]: LinearRegression()

In [106]: a = reg.predict(x_test)
a
Out[106]: array([73.30963425, 75.63399415, 80.12271394, 79.00053997])

In [107]: y_test
Out[107]: 2    72.570943
4    74.648723
6    76.479588
7    75.600425
Name: sal, dtype: float64

In [108]: from sklearn.metrics import mean_squared_error
print("mse: %.2f" % mean_squared_error(y_test, a))
mse: 7.62

In [109]: dt = pd.DataFrame({'actual': y_test, 'predicted': a})

```

Figure 15 Test Case 1 Using Water Data Input

```

In [139]: cols = ["year"]

In [140]: y = data["sal"]
xdata[cols]

In [141]: reg = linear_model.LinearRegression()
x_train, x_test, y_train, y_test = train_test_split(x, y, test_size=0.2, random_state=2)

In [142]: reg.fit(x_train, y_train)

Out[142]: LinearRegression()

In [143]: a = reg.predict(x_test)
a
Out[143]: array([73.86206240, 74.59948715])

In [144]: y_test
Out[144]: 3    74.055193
4    74.648723
Name: sal, dtype: float64

In [145]: from sklearn.metrics import mean_squared_error
print("mse: %.2f" % mean_squared_error(y_test, a))
mse: 0.03

In [146]: dt = pd.DataFrame({'actual': y_test, 'predicted': a})

In [147]: #using gradient descent to optimize it further
x = (x - x.mean()) / x.std()
x = np.c_[np.ones(x.shape[0]), x]
a
Out[147]: array([[ 1., -1.46851409],
[ 1., -1.09544512],

```

Figure 16 Test Case 2 Using Random Data Input

## 8. OUTPUT SCREENS

Comparing the actual and predicted values:

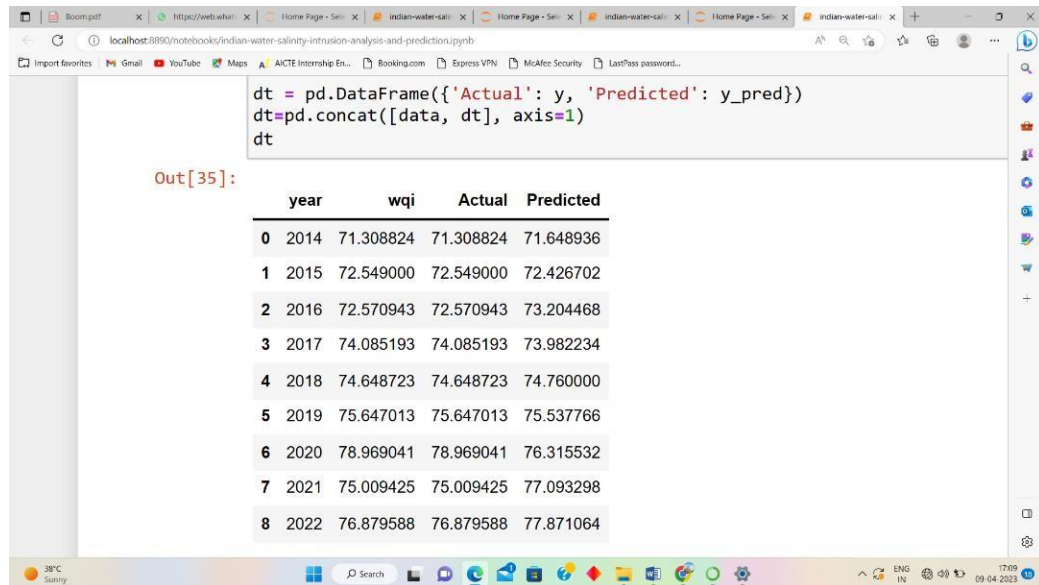


Figure 17 prediction of January (2014-2022)

We calculate wqi by using PH, DO, Nitrates and other parameters in the water. We get the predicted outputs and actual values from the taken Datasets.

By comparing the actual and predicted values we can conclude that the values are almost same.

Only in year 2021 we see much difference between actual and predicted values it is many due to covid19 pandemic as many industries are closed we see sudden down fall in pumping waters from bore wells, this leads to change in water quality index. Plot using Matplotlib:

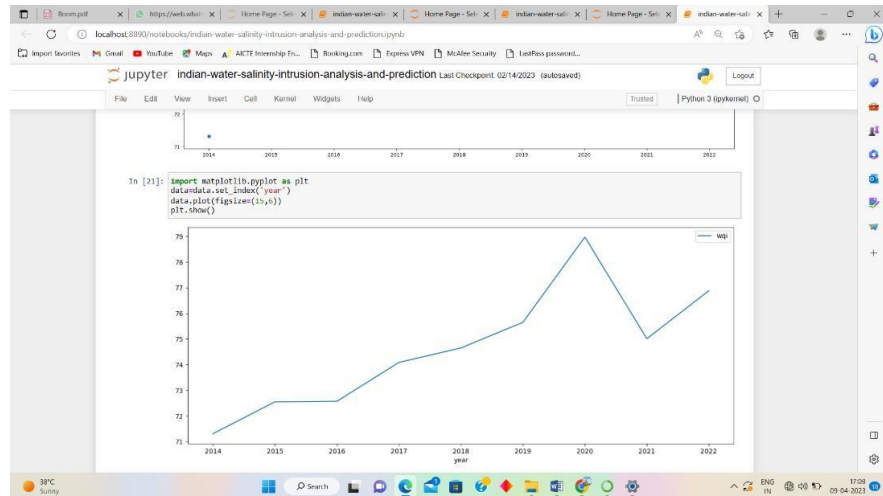


Figure 18 Plot

In matplotlib.pyplot various states are preserved across function calls, so that it keeps track of things like the current figure and plotting area, and the plotting functions are directed to the current axes.

This function accepts parameters that enables us to set axes scales and format the graphs. These parameters are mentioned below

x axis is taken

as year y axis

is taken as

WQI

We get the data plot according actual data given

**The scattered plot of data points year wise:**

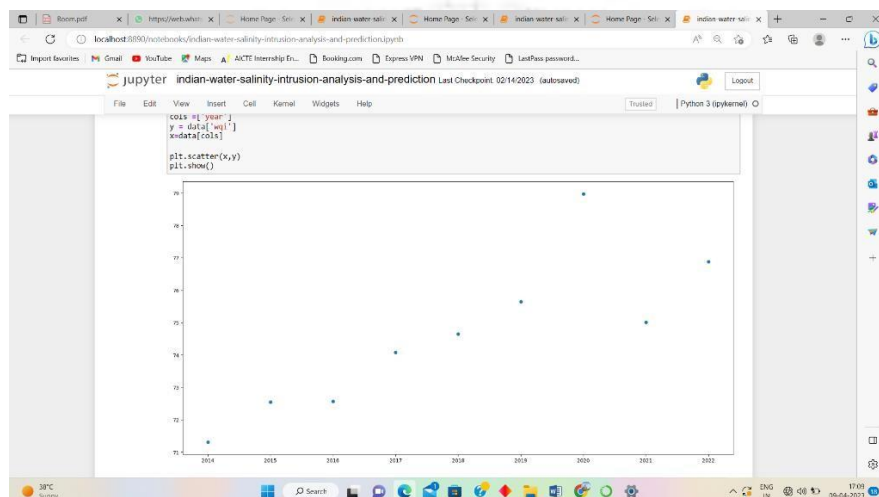


Figure 19 scatter plot

Scatter plot are use to observe relationship between WQI and year (variables) and uses dots to represent the relationship between them.

It also shows how change in one variable affects the other. We can see the change in wqi with respective to year.

Scattered plot of data points  
year wise:

X axis is taken as year

Y axis is taken as WQI

We get the data plot according to the predicted values

### Visualizing the filtered data:

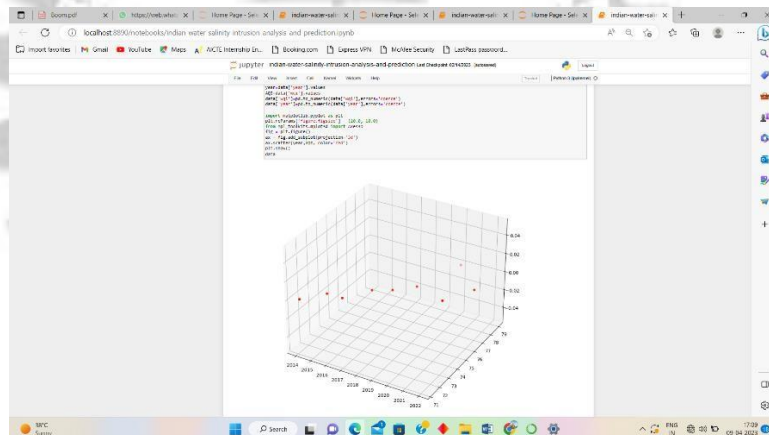
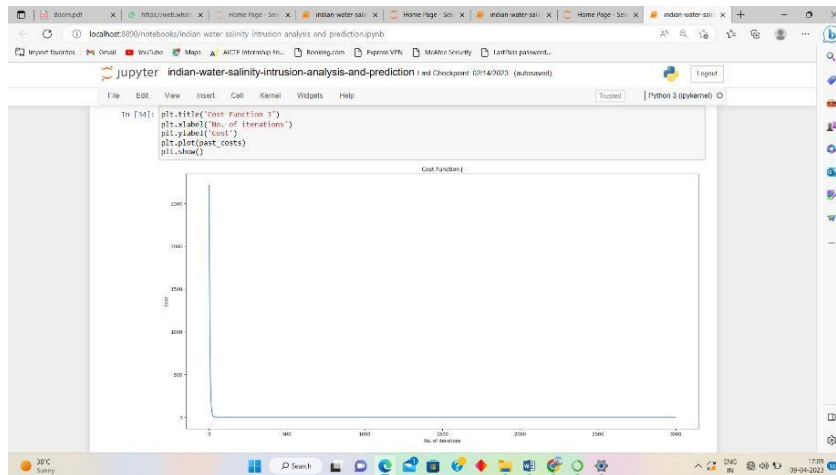


Figure 20 visual plot

With this three-dimensional axes enabled, we can now plot a variety of three-dimensional plot types. Three-dimensional plotting is one of the functionalities that benefits immensely from viewing figures interactively rather than statically. Dimensional axes are enabled and data can be plotted in 3 dimensions.

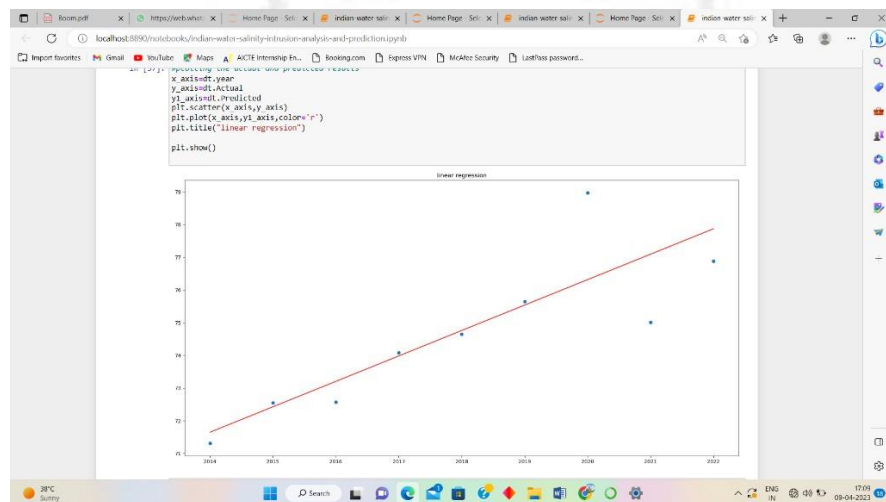
A surface plot considers the X and Y coordinates as latitude and longitude, and Z as the altitude. It represents the dataset as a surface by interpolating positions between data points.

**Plotting the cost function:***Figure 21 cost function*

It is used to chart how production expenses will change at different output levels. In other words, it estimates the total cost of production given a specific quantity produced.

It helps to analyze how well a Machine Learning model performs. A Cost function basically compares the predicted values with the actual values. Appropriate choice of the Cost function contributes to the credibility and reliability of the model.

X axis is taken as no of iterations Y axis is taken as cost

**Plotting the actual and predicted results:***Figure 22 plotting the actual and predicted results*

In linear regression, each observation consists of two values.

One value is for the dependent variable and one value is for the independent variable. Straight line approximates the relationship between the dependent variable and the independent variable.

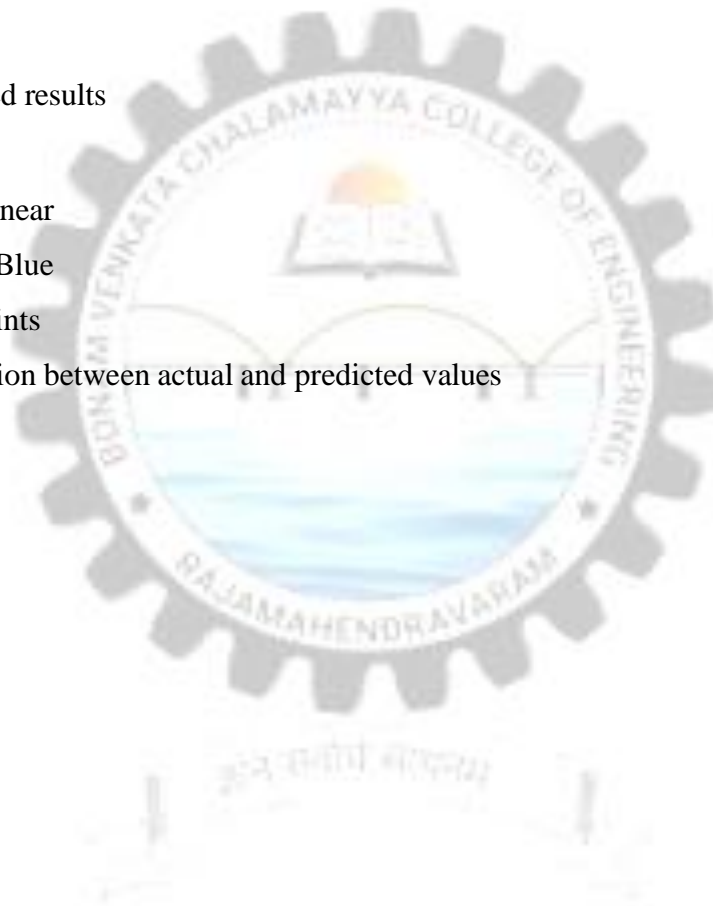
X axis year

Y axis actual input

Y1 axis predicted results

The red line linear  
regression      Blue  
dots are data points

We get the relation between actual and predicted values





## **9. CONCLUSION**

This study aimed to examine the predictability of salinity intrusion in the soil based on the value of water salinity using machine learning. The findings are very important and may support decision making for land-use planning, not only for Vietnam, but also for other countries where water salinity is common. We tested the initial hypothesis that water salinity and water salinity are strongly related. The second hypothesis was that the hybrid model is more efficient than the individual model for predicting water salinity. The results show that the water salinity values can significantly classify saline soils by applying machine learning techniques. Salinity intrusion is causing saltiness in water which indirectly affects the environment. We overcome this by predicting and giving knowledge on how to fresh water resources. So we conclude that usage of fresh water levels to its limit can reduce the salt intrusion in water and also by aware people with prediction of salinity we can reduce the risk of decreasing fresh water levels in the future. Salinity intrusion prediction is a crucial task in managing water resources in coastal areas. Predicting salinity intrusion can help us to make informed decisions about water use and allocation, agricultural planning, and infrastructure development. Despite the challenges, salinity intrusion prediction remains an essential task in managing coastal water resources. With advances in technology and increased data availability, we can expect to see continued improvements in our ability to predict and manage salinity intrusion in the future.

### **9.1 FUTURE SCOPE**

With the ongoing changes in climate, predicting salinity intrusion becomes crucial as it has a significant impact on the coastal regions. As sea levels rise and rainfall patterns change, the amount of saltwater intrusion can change, and models that predict these changes could be helpful in preparing for and mitigating the effects of climate change. Salinity intrusion affects the availability and quality of freshwater, which has a significant impact on agriculture in coastal areas. Predicting salinity intrusion can help farmers make decisions about crop selection and irrigation strategies.

## 10. REFERENCES

1. Abril, J.M. and Abdel-Aal, M.M., A modeling study on hydrodynamics and pollutant dispersion in the Suez Canal, *Ecol. Modell.*, 2000, vol. 128, pp. 1–17.
2. Anagnostopoulos, P., Mpimpas, H., and Ganoulis, J., Numerical simulation of coastal pollution in the Thermaikos Gulf, in *Computer Modeling of Seas and Coastal Regions III*, Acinas, J.R. and Brebbia, C.A., Eds., Southampton: Computational Mechanics Publications, 1997, pp. 173–182.
3. APHA-standard methods for the examination of water and waste water, Washington, DC: APHA, AWWA, WPCF, 1985, 16th Ed.
4. Brown, R.M., McClellan, N.I., Deininger, R.A., and Tozer, R.G., A water quality index—do we dare? *Water Sewage Works*, 1970, vol. 117, pp. 343–339.
5. Chang, N., Chen, H.W., and King, S.K., Identification of river water quality using the fuzzy synthetic evaluation approach, *J. Environ. Manage*, 2001, vol. 63, pp. 293–305.
6. Consulting Engineers of Sefidrud Guilan, Update of water resources in the Atlantic basin Sefidrud, Talesh and Anzali Wetland, Volume III: Statistics and analysis of water, Rasht, Iran: Guilan Regional Water Corporation, 2006.
7. Cooley, R., Classification of News Stories Using Support Vector Machines, in *IJCAI99 Workshop on Text Mining*, Minnesota, 1999.
8. Dojlido, J.R., Raniszewski, J.R., and Woyciechowska, J., Water quality index applied to rivers in the Vis-Tula River basin in Poland, *Environ. Monit. Assess*, 1994, vol. 33, pp. 33–42.
9. Horton, R.K., An index number system for rating water quality, *J. Water Pollut. Control Fed.*, 1965, vol. 37, pp. 300–305.
- 10.

11. Icaga, Y., Fuzzy evaluation of water quality classification, *Ecol. Indic.*, 2007, vol. 7, pp.710–718.
12. Kannel, P.R., Lee, S. Lee, Y.S., Kanel, S.R., and Khan, S.P., Application of water quality indices and dissolved oxygen as indicators of river water classification and urban impact assessment, *Environ. Monit. Assess*, 2007, vol. 132, no. 2, pp. 93–110.
13. Liou, S.M., Lo, S.L., and Wang, S.H., A generalized water quality index for Taiwan, *Environ. Monit. Assess*, 2004, vol. 96, pp. 35–52. Manahan, S.E., *Environmental Chemistry*, Boca Raton: Lewis Publishers, 2000, 7th edition.
14. Miller, W.W., Joung, H.M., Mahannah, C.N., and Garrett, J.R., Identification of water quality differences in Nevada through index application, *J. Environ. Qual.*, 1986, vol. 15, pp. 265–272.
15. Nagel, J.W., Colley, D., and Smith, D.J., A water quality index for contact recreation in New Zealand, *WaterSci. Technol.*, 2001, vol. 43, no. 5, pp. 285–292.

