**A Machine Learning Approach for Early Detection of Fish Diseases by Analyzing Water Quality**

**1.INTRODUCTION:**

Fishes account for approximately 15% of the animal protein intake of the human population globally. In countries like Bangladesh, fishes provide as high as 60% of the animal protein to the populace, also in economic valuation fishes contribute to approximately 3.6% to the national GDP which weighs nearly 25% of the entire agricultural GDP. Furthermore, this section employs about 11% of the total population in Bangladesh in full-time and part-time basis . Despite being the very vibrant economic sector for the country, one major threat to the fish farmers are the fish diseases which eventually puts a huge constraint on the economic progress, and severely strains the expansion of the aquaculture and fish farming. Fish culture faces severe threat from waterborne pathogens, such as bacteria and virus, responsible mainly for mass mortality and poor health. It is, therefore, imperative to monitor the purity of the water habitat to detect fish diseases timely and accurately. Fish performs all their physical activities under water; fish dependents on water for breathing, feeding, reproducing and growth. When the water quality of the habitat deteriorates it becomes unfavorable for fish to live in. Water quality depends on certain parameters, and when the parameters change the quality deteriorates. As a result, the health of fishes is threatened by the compromise of their immune system critically leading them to be vulnerable to harmful pathogens. Oxygen plays a pivotal role in maintaining life under water. When oxygen level goes below to the preferable range, the physiological and physical growth of fish species are hampered. Decreasing oxygen level under water results in increasing carbon dioxide level causing acidosis and nephron calcinosis that obstructs the development of granulomas in many internal organs of fish . Moreover, recirculated water contains high pH which turns up ammonia level in water. High level of ammonia causes harm to the gills and liver of fishes. Depending on the level of saturation and the time of exposure, Gas supersaturation of the water can result in the gas bubble disease. The main cause to the disease is the development of bubbles in the eyes, skin and gills. Degraded water quality causes pollution that creates serious problem to fish; necrotic alteration, papilloma, degenerative, and fin erosion is the result of water pollution. As a result, the fish body gets abnormal growth, and farmers do not get optimum production. Problems such as these can easily be resolved if the farmers could identify them early. Machine learning techniques and artificial intelligent algorithms in recent years have been used extensively and very successfully in classification and decision-making problems. Smart algorithms can learn from parameter space of system the correct desired classification based on real dataset, and infer very accurately any deviation from the desired sets of configuration, which may be exploited for decision making. A hierarchical architecture to the algorithm ensures higher performance from learning. Machine learning technique, such as Gradient boosting, exploits decision-tree based hierarchical in-built structure based on regression algorithm to classify complex problems and to help in the decision-making automation. Inspired by the successes of such technique in variety of complex problems, we have employed the technique in this study to predict fish diseases by solving classification problems from real dataset composed of desired parameters of water purity. Our approaches to the development of an automated fish disease detection algorithm and employing it in decision-making process may be summarized in the following sequential steps: Step 1: Taking sample of water to identify the water quality. Step 2: Making prediction of the water quality using machine learning algorithm. We have already collected and prepared dataset and trained our algorithm using it, so that machine can make prediction on probable fish diseases based on the water quality parameters. Step 3: Analyzing the disease and identifying. Step 4: Making smart decision to minimize harm to fish farm and ensuring healthy habitat. The paper is arranged as following: the review of the related literature is conducted in the second section followed by an in-depth look into the proposed machine learning algorithm and parameters of the model in the section three. The dataset is discussed in the section four along with the preparation, processing, and implementation in the model. The experimental results are discussed in the section five followed by concluding remarks.

* 1. **Objective of the project:**

Fish diseases in aquaculture constitute a significant hazard to nutriment security. Identification of infected fishes in aquaculture remains challenging to find out at the early stage due to the dearth of necessary infrastructure. The identification of infected fish timely is an obligatory step to thwart from spreading disease. In this work, we want to find out the salmon fish disease in aquaculture, as salmon aquaculture is the fastest-growing food production system globally, accounting for 70 percent (2.5 million tons) of the market. In the alliance of flawless image processing and machine learning mechanism, we identify the infected fishes caused by the various pathogen. This work divides into two portions. In the rudimentary portion, image pre-processing and segmentation have been applied to reduce noise and exaggerate the image, respectively. In the second portion, we extract the involved features to classify the diseases with the help of the Support Vector Machine (SVM) algorithm of machine learning with a kernel function. The processed images of the first portion have passed through this (SVM) model. Then we harmonize a comprehensive experiment with the proposed combination of techniques on the salmon fish image dataset used to examine the fish disease. We have conveyed this work on a novel dataset compromising with and without image augmentation. The results have bought a judgment of our applied SVM performs notably with 91.42 and 94.12 percent of accuracy, respectively, with and without augmentation.

**2. LITERATURE SURVEY:**

**“Water quality analysis of surface water”**

The fundamental of water quality management and analysis is to adopt several monitoring actions to save water bodies and protect users. Evaluation of physiochemical and biological parameters of surface water is an important aspect for water quality model formation and adopted suitable management actions. This chapter focused on evaluation, monitoring, and analysis of surface water quality. Recently, Water Quality Indices (WQIs) was introduced as new methods for water quality management. Dissolved oxygen in surface water is necessary for aquatic life. Microbial quality of surface water should be safe and should not have adverse effect on human health. In this chapter, WQIs formation, definitions, and limitations were discussed, surface water modeling, standards and indicators of chemical and microbial quality of surface water were presented and discussed.

**“Greedy Function Approximation: A Gradient Boosting Machine”**

Function estimation/approximation is viewed from the perspective of numerical optimization in function space, rather than parameter space. A connection is made between stagewise additive expansions and steepest-descent minimization. A general gradient descent “boosting” paradigm is developed for additive expansions based on any fitting criterion.Specific algorithms are presented for least-squares, least absolute deviation, and Huber-M loss functions for regression, and multiclass logistic likelihood for classification. Special enhancements are derived for the particular case where the individual additive components are regression trees, and tools for interpreting such “TreeBoost” models are presented. Gradient boosting of regression trees produces competitive, highly robust, interpretable procedures for both regression and classification, especially appropriate for mining less than clean data. Connections between this approach and the boosting methods of Freund and Shapire and Friedman, Hastie and Tibshirani are discussed.

**“Wastewater quality monitoring system using sensor fusion and machine learning techniques.”**

A multi-sensor water quality monitoring system incorporating an UV/Vis spectrometer and a turbidimeter was used to monitor the Chemical Oxygen Demand (COD), Total Suspended Solids (TSS) and Oil & Grease (O&G) concentrations of the effluents from the Chinese restaurant on campus and an electrocoagulation-electroflotation (EC-EF) pilot plant. In order to handle the noise and information unbalance in the fused UV/Vis spectra and turbidity measurements during the calibration model building, an improved boosting method, Boosting-Iterative Predictor Weighting-Partial Least Squares (Boosting-IPW-PLS), was developed in the present study. The Boosting-IPW-PLS method incorporates IPW into boosting scheme to suppress the quality-irrelevant variables by assigning small weights, and builds up the models for the wastewater quality predictions based on the weighted variables. The monitoring system was tested in the field with satisfactory results, underlying the potential of this technique for the online monitoring of water quality.

**“Comparing the performance of different neural networks for binary classification problems.”**

Classification problem is a decision making task where many researchers have been working on. There are a number of techniques proposed to perform classification. Neural network is one of the artificial intelligent techniques that has many successful examples when applying to this problem. This paper presents a comparison of neural network techniques for binary classification problems. The classification performance obtained by five different types of neural networks for comparison are back propagation neural network (BPNN), radial basis function neural network (RBFNN), general regression neural network (GRNN), probabilistic neural network (PNN), and complementary neural network (CMTNN). The comparison is done based on three benchmark data sets obtained from UCI machine learning repository. The results show that CMTNN typically provide better classification results when comparing to techniques applied to binary classification problems.

**“Water quality assessment based on the water quality index method in Lake Poyang.”**

Twenty-four samplings were conducted every 3 months at 15 sites from January 2009 to October 2014 in Lake Poyang, and 20 parameters were analysed and classified into three groups (toxic metals, easily treated parameters, and others). The assessment results based on water quality index (WQI) showed that the water quality in Lake Poyang was generally “moderate”, according to the classification of the surface water quality standard (GB3838-2002) in China, but a deteriorating trend was observed at the interannual scale. Seasonally, the water quality was best in summer and worst in winter. Easily treated parameters generally determined the WQI value in the assessment, especially total nitrogen (TN) and total phosphorus (TP), while toxic metals and other parameters in Lake Poyang were generally at low and safe levels for drinking water. Water level (WL) has a net positive effect on water quality in Lake Poyang through dilution of environmental parameters, which in practice means TN. Consequently, local management agencies should pay more attention to nutrient concentrations during the monitoring schedule, as well as during the low-water periods which manifest a relatively bad water quality state, especially with the prevailing low WL observed recently in Lake Poyang.

**“Estimation of Water Quality Index by Weighted Arithmetic Water Quality Index Method.”**

The current study is about water quality parameters of Vijayawada, Krishna district of Andhra Pradesh. This study is intended to estimate water quality Index (WQI) of study area using Weighted Arithmetic water quality index method. In this study the quality of water of study area is determined using the various physico - chemical parameters such as pH , Total Dissolved Solids (TDS), Cl , SO4 , Na, K, Ca, Mg, and Total Hardness (TH). Physico - Chemical analysis of water samples was carried out at one hundred and ninety different stations of study area for pre monsoon and post monsoon seasons of 2014 which is collected and analyzed out of which nineteen samples were selected and water quality index estimated. Out of one hundred and ninety samples nineteen prime locations (wherein physiochemical property values were maximum) were selected and the quality index was analysed and explained in this paper. This study is to investigate the suitability of water for drinking purpose based on Water quality Index (WQI) estimated. Keywords : Water quality index (WQI) , sulphate ( SO4), Sodium(Na),Potassium( K),Calcium( Ca), Magnesium(Mg), and Total Hardness (TH). Physico - Chemical parameters.

**“Comparing the performance of different neural networks for binary classification problems.”**

Classification problem is a decision making task where many researchers have been working on. There are a number of techniques proposed to perform classification. Neural network is one of the artificial intelligent techniques that has many successful examples when applying to this problem. This paper presents a comparison of neural network techniques for binary classification problems. The classification performance obtained by five different types of neural networks for comparison are back propagation neural network (BPNN), radial basis function neural network (RBFNN), general regression neural network (GRNN), probabilistic neural network (PNN), and complementary neural network (CMTNN). The comparison is done based on three benchmark data sets obtained from UCI machine learning repository. The results show that CMTNN typically provide better classification results when comparing to techniques applied to binary classification problems.

**“Physico-chemical analysis of water samples.”**

The water quality is determined in five blocks (water samples taken from Urban and rural locations of Bangalore(HAL 3rd stage (Kaveri water), HRBR lay out (kaveri water mixed with Bore well water), Borewell water (Srinivasapura,Kolar District)and Mineral water samples). where from each block water samples are under studied for Physico- chemical status of water samples. In Physico-chemical analysis , various quality parameter are measured including pH, Specific conductivity(SP) , total dissolved solids (TDS),total hardness, compared with WHO standards of water quality; also in present research paper classification ofwater samples of five blocks was investigation on the basis of TDS, anions, cations and TH. .T he pH of all water samples were found almost neutral . The TDS, conductance, hardness increased towards the urban water as compared to rural wate All Parameters were within the permissible limits. The results indicated and discussed.

**3. SYSTEM ANALYSIS**

**3.1 Existing System**

The CNN system with perfect images without noise. Next step is to build our own dataset of fish in the ocean. Because it is difficult to obtain images from other kinds of objects such as the sea turtle, coral and so on. This part of the research, fish is the only object to be detected. For the collection of 410 images, many of them have multiple fish in one image, so the detection is challenging. The same method was chosen to create ground truth image. And all the parameters introduced before remain the same, only the class information is made up of a 1 × 1 × 18 vector instead of 1 × 1 × 30 because of reduce in the classes. Figure 11 illustrates one labelled image example. It is obvious that this data set is totally different from the ideal images from ImageNET.

**Disadvantages**

1. It takes more time
2. Less accuracy

**3.2 Proposed System**

The paper author is analysing water quality by applying machine learning algorithm to predict fish diseases. If water contains high toxins or viruses then it will affect fish lungs which causes disease inside fish or fish may die. So we can collect water samples and then apply on machine learning model to predict water quality and if quality is not good then we can say fish is not healthy. In propose paper author has used water quality dataset from KAGGLE website and then trained with Gradient Boosting algorithm. This algorithm giving more than 95% accurate prediction accuracy on test data.

**Advantages**

1.It takes less time

2.More prediction

**Modules:**

1. Upload Water Quality Dataset: using this module we will upload dataset to application
2. Preprocess & Normalize Dataset: using this module will convert all non-numeric data to numeric data and then normalize all values
3. Features Selection: using this module application will select X training features and Y class label and then split dataset into train and test where application using 80% dataset for training and 20% for testing
4. Train Gradient Boosting Algorithm: 80% training data will be input to Gradient Boosting algorithm to train a model and this model will be applied on test data to calculate prediction accuracy
5. Predict Fish Condition: using this module we will upload test data and then algorithm will predict weather fish is healthy or disease affected.

**3.3. PROCESS MODEL USED WITH JUSTIFICATION**

**SDLC (Umbrella Model):**

**Umbrella Activity**

**Umbrella Activity**

**Umbrella Activity**

1. Feasibility Study
2. TEAM FORMATION
3. Project Specification PREPARATION

Business Requirement Documentation

ANALYSIS & DESIGN

CODE

UNIT TEST

DOCUMENT CONTROL

ASSESSMENT

TRAINING

INTEGRATION & SYSTEM TESTING

DELIVERY/INSTALLATION

ACCEPTANCE TEST

Requirements Gathering

SDLC is nothing but Software Development Life Cycle. It is a standard which is used by software industry to develop good software.

**Stages in SDLC:**

* Requirement Gathering
* Analysis
* Designing
* Coding
* Testing
* Maintenance

**Requirements Gathering** **stage:**

The requirements gathering process takes as its input the goals identified in the high-level requirements section of the project plan. Each goal will be refined into a set of one or more requirements. These requirements define the major functions of the intended application, define operational data areas and reference data areas, and define the initial data entities. Major functions include critical processes to be managed, as well as mission critical inputs, outputs and reports. A user class hierarchy is developed and associated with these major functions, data areas, and data entities. Each of these definitions is termed a Requirement. Requirements are identified by unique requirement identifiers and, at minimum, contain a requirement title and textual description.



These requirements are fully described in the primary deliverables for this stage: the Requirements Document and the Requirements Traceability Matrix (RTM). The requirements document contains complete descriptions of each requirement, including diagrams and references to external documents as necessary. Note that detailed listings of database tables and fields are *not* included in the requirements document.

The title of each requirement is also placed into the first version of the RTM, along with the title of each goal from the project plan. The purpose of the RTM is to show that the product components developed during each stage of the software development lifecycle are formally connected to the components developed in prior stages.

In the requirements stage, the RTM consists of a list of high-level requirements, or goals, by title, with a listing of associated requirements for each goal, listed by requirement title. In this hierarchical listing, the RTM shows that each requirement developed during this stage is formally linked to a specific product goal. In this format, each requirement can be traced to a specific product goal, hence the term requirements traceability.

The outputs of the requirements definition stage include the requirements document, the RTM, and an updated project plan.

* Feasibility study is all about identification of problems in a project.
* No. of staff required to handle a project is represented as Team Formation, in this case only modules are individual tasks will be assigned to employees who are working for that project.
* Project Specifications are all about representing of various possible inputs submitting to the server and corresponding outputs along with reports maintained by administrator.

**Analysis Stage:**

The planning stage establishes a bird's eye view of the intended software product, and uses this to establish the basic project structure, evaluate feasibility and risks associated with the project, and describe appropriate management and technical approaches.



The most critical section of the project plan is a listing of high-level product requirements, also referred to as goals. All of the software product requirements to be developed during the requirements definition stage flow from one or more of these goals. The minimum information for each goal consists of a title and textual description, although additional information and references to external documents may be included. The outputs of the project planning stage are the configuration management plan, the quality assurance plan, and the project plan and schedule, with a detailed listing of scheduled activities for the upcoming Requirements stage, and high level estimates of effort for the out stages.

**Designing Stage:**

The design stage takes as its initial input the requirements identified in the approved requirements document. For each requirement, a set of one or more design elements will be produced as a result of interviews, workshops, and/or prototype efforts. Design elements describe the desired software features in detail, and generally include functional hierarchy diagrams, screen layout diagrams, tables of business rules, business process diagrams, pseudo code, and a complete entity-relationship diagram with a full data dictionary. These design elements are intended to describe the software in sufficient detail that skilled programmers may develop the software with minimal additional input.

  
When the design document is finalized and accepted, the RTM is updated to show that each design element is formally associated with a specific requirement. The outputs of the design stage are the design document, an updated RTM, and an updated project plan.

**Development (Coding) Stage:**

The development stage takes as its primary input the design elements described in the approved design document. For each design element, a set of one or more software artefacts will be produced. Software artefacts include but are not limited to menus, dialogs, and data management forms, data reporting formats, and specialized procedures and functions. Appropriate test cases will be developed for each set of functionally related software artefacts, and an online help system will be developed to guide users in their interactions with the software.



The RTM will be updated to show that each developed artefact is linked to a specific design element, and that each developed artefact has one or more corresponding test case items. At this point, the RTM is in its final configuration. The outputs of the development stage include a fully functional set of software that satisfies the requirements and design elements previously documented, an online help system that describes the operation of the software, an implementation map that identifies the primary code entry points for all major system functions, a test plan that describes the test cases to be used to validate the correctness and completeness of the software, an updated RTM, and an updated project plan.

**Integration & Test Stage:**

During the integration and test stage, the software artefacts, online help, and test data are migrated from the development environment to a separate test environment. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite confirms a robust and complete migration capability. During this stage, reference data is finalized for production use and production users are identified and linked to their appropriate roles. The final reference data (or links to reference data source files) and production user list are compiled into the Production Initiation Plan.



The outputs of the integration and test stage include an integrated set of software, an online help system, an implementation map, a production initiation plan that describes reference data and production users, an acceptance plan which contains the final suite of test cases, and an updated project plan.

* **Installation & Acceptance Test:**

During the installation and acceptance stage, the software artefacts, online help, and initial production data are loaded onto the production server. At this point, all test cases are run to verify the correctness and completeness of the software. Successful execution of the test suite is a prerequisite to acceptance of the software by the customer.

After customer personnel have verified that the initial production data load is correct and the test suite has been executed with satisfactory results, the customer formally accepts the delivery of the software.



The primary outputs of the installation and acceptance stage include a production application, a completed acceptance test suite, and a memorandum of customer acceptance of the software. Finally, the PDR enters the last of the actual labor data into the project schedule and locks the project as a permanent project record. At this point the PDR "locks" the project by archiving all software items, the implementation map, the source code, and the documentation for future reference.

**Maintenance:**

Outer rectangle represents maintenance of a project, Maintenance team will start with requirement study, understanding of documentation later employees will be assigned work and they will undergo training on that particular assigned category. For this life cycle there is no end, it will be continued so on like an umbrella (no ending point to umbrella sticks).

**3.4. Software Requirement Specification**

**3.4.1. Overall Description**

A Software Requirements Specification (SRS) – a [requirements specification](http://en.wikipedia.org/wiki/Requirements_specification) for a [software system](http://en.wikipedia.org/wiki/Software_system) is a complete description of the behavior of a system to be developed. It includes a set of [use cases](http://en.wikipedia.org/wiki/Use_case) that describe all the interactions the users will have with the software. In addition to use cases, the SRS also contains non-functional requirements. [Non-functional requirements](http://en.wikipedia.org/wiki/Non-functional_requirements) are requirements which impose constraints on the design or implementation (such as [performance engineering](http://en.wikipedia.org/wiki/Performance_engineering) requirements, [quality](http://en.wikipedia.org/wiki/Quality_%28business%29) standards, or design constraints).

System requirements specification: A structured collection of information that embodies the requirements of a system. A [business analyst](http://en.wikipedia.org/wiki/Business_analyst), sometimes titled [system analyst](http://en.wikipedia.org/wiki/System_analyst), is responsible for analyzing the business needs of their clients and stakeholders to help identify business problems and propose solutions. Within the [systems development lifecycle](http://en.wikipedia.org/wiki/Systems_development_life_cycle) domain, the BA typically performs a liaison function between the business side of an enterprise and the information technology department or external service providers. Projects are subject to three sorts of requirements:

* [Business requirements](http://en.wikipedia.org/wiki/Business_requirements) describe in business terms what must be delivered or accomplished to provide value.
* Product requirements describe properties of a system or product (which could be one of several ways to accomplish a set of business requirements.)
* Process requirements describe activities performed by the developing organization. For instance, process requirements could specify .Preliminary investigation examine project feasibility, the likelihood the system will be useful to the organization. The main objective of the feasibility study is to test the Technical, Operational and Economical feasibility for adding new modules and debugging old running system. All system is feasible if they are unlimited resources and infinite time. There are aspects in the feasibility study portion of the preliminary investigation:
* **ECONOMIC FEASIBILITY**

A system can be developed technically and that will be used if installed must still be a good investment for the organization. In the economical feasibility, the development cost in creating the system is evaluated against the ultimate benefit derived from the new systems. Financial benefits must equal or exceed the costs. The system is economically feasible. It does not require any addition hardware or software. Since the interface for this system is developed using the existing resources and technologies available at NIC, There is nominal expenditure and economical feasibility for certain.

* **Operational Feasibility**

Proposed projects are beneficial only if they can be turned out into information system. That will meet the organization’s operating requirements. Operational feasibility aspects of the project are to be taken as an important part of the project implementation. This system is targeted to be in accordance with the above-mentioned issues. Beforehand, the management issues and user requirements have been taken into consideration. So there is no question of resistance from the users that can undermine the possible application benefits. The well-planned design would ensure the optimal utilization of the computer resources and would help in the improvement of performance status.

* **TECHNICAL FEASIBILITY**

Earlier no system existed to cater to the needs of ‘Secure Infrastructure Implementation System’. The current system developed is technically feasible. It is a web based user interface for audit workflow at NIC-CSD. Thus it provides an easy access to .the users. The database’s purpose is to create, establish and maintain a workflow among various entities in order to facilitate all concerned users in their various capacities or roles. Permission to the users would be granted based on the roles specified. Therefore, it provides the technical guarantee of accuracy, reliability and security.

**3.4.2. External Interface Requirements**

**User Interface**

The user interface of this system is a user friendly python Graphical User Interface.

**Hardware Interfaces**

The interaction between the user and the console is achieved through python capabilities.

**Software Interfaces**

The required software is python.

**HARDWARE REQUIREMENTS:**

# Processor - I3(min)

* Speed - 1.1 GHz
* RAM - 4GB(min)
* Hard Disk - 500GB
* Key Board - Standard Windows Keyboard
* Mouse - Two or Three Button Mouse
* Monitor - SVGA

**SOFTWARE REQUIREMENTS:**

* Operating System - Windows 10/above
* Programming Language - Python 3.7

**4. SYSTEM DESIGN**

**UML Diagram:**

The Unified Modelling Language allows the software engineer to express an analysis model using the modelling notation that is governed by a set of syntactic semantic and pragmatic rules.

A UML system is represented using five different views that describe the system from distinctly different perspective. Each view is defined by a set of diagram, which is as follows.

* + **User Model View**
    1. This view represents the system from the users perspective.
    2. The analysis representation describes a usage scenario from the end-users perspective.
  + **Structural Model view**
    1. In this model the data and functionality are arrived from inside the system.
    2. This model view models the static structures.
* **Behavioural Model View**

It represents the dynamic of behavioural as parts of the system, depicting the interactions of collection between various structural elements described in the user model and structural model view.

* **Implementation Model View**

In this the structural and behavioural as parts of the system are represented as they are to be built.

* **Environmental Model View**

In this the structural and behavioural aspects of the environment in which the system is to be implemented are represented.

**Class Diagram:**

The class diagram is the main building block of object oriented modeling. It is used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code. Class diagrams can also be used for data modeling. The classes in a class diagram represent both the main objects, interactions in the application and the classes to be programmed. In the diagram, classes are represented with boxes which contain three parts:

* The upper part holds the name of the class
* The middle part contains the attributes of the class
* The bottom part gives the methods or operations the class can take or undertake.



**Use case Diagram:**

A **use case diagram** at its simplest is a representation of a user's interaction with the system and depicting the specifications of a use case. A use case diagram can portray the different types of users of a system and the various ways that they interact with the system. This type of diagram is typically used in conjunction with the textual use case and will often be accompanied by other types of diagrams as well.



**Sequence diagram:**

A sequence diagram 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. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called event diagrams, event scenarios, and timing diagrams.



**Collaboration diagram:**

A collaboration diagram describes interactions among objects in terms of sequenced messages. Collaboration diagrams represent a combination of information taken from class, sequence, and use case diagrams describing both the static structure and dynamic behaviour of a system.



**Component Diagram:**

In the Unified Modelling Language, a component diagram depicts how components are wired together to form larger components and or software systems. They are used to illustrate the structure of arbitrarily complex systems.

Components are wired together by using an assembly connector to connect the required interface of one component with the provided interface of another component. This illustrates the service consumer - service provider relationship between the two components.



**Deployment Diagram:**

A **deployment diagram** in the Unified Modeling Language models the *physical* deployment of artifacts on nodes. To describe a web site, for example, a deployment diagram would show what hardware components ("nodes") exist (e.g., a web server, an application server, and a database server), what software components ("artifacts") run on each node (e.g., web application, database), and how the different pieces are connected (e.g. JDBC, REST, RMI).

The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within the boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as a cluster of database servers.



**Activity Diagram:**

Activity diagram is another important diagram in UML to describe dynamic aspects of the system. It is basically a flow chart to represent the flow form one activity to another

activity.The activity can be described as an operation of the system. So the control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent

upload

preprocess

feature selection

run Gradient Boosting

predict

**Data Flow Diagram:**

Data flow diagrams illustrate how data is processed by a system in terms of inputs and outputs. Data flow diagrams can be used to provide a clear representation of any business function. The technique starts with an overall picture of the business and continues by analyzing each of the functional areas of interest. This analysis can be carried out in precisely the level of detail required. The technique exploits a method called top-down expansion to conduct the analysis in a targeted way.

As the name suggests, Data Flow Diagram (DFD) is an illustration that explicates the passage of information in a process. A DFD can be easily drawn using simple symbols. Additionally, complicated processes can be easily automated by creating DFDs using easy-to-use, free downloadable diagramming tools. A DFD is a model for constructing and analyzing information processes. DFD illustrates the flow of information in a process depending upon the inputs and outputs. A DFD can also be referred to as a Process Model. A DFD demonstrates business or technical process with the support of the outside data saved, plus the data flowing from the process to another and the end results.

User

1. upload 2.upload successfully

3. preprocess 4.preprocess successfully

5. feature selection 6.feature selection successfully

7. run Gradient Boosting 8. Run Gradient Boosting Successfully

9. predict 10. Predicted successfully

**5. IMPLEMETATION**

**5.1 Python**

Python is a general-purpose language. It has wide range of applications from Web development (like: Django and Bottle), scientific and mathematical computing (Orange, SymPy, NumPy) to desktop graphical user Interfaces (Pygame, Panda3D). The syntax of the language is clean and length of the code is relatively short. It's fun to work in Python because it allows you to think about the problem rather than focusing on the syntax.

**History of Python:**

Python is a fairly old language created by Guido Van Rossum. The design began in the late 1980s and was first released in February 1991.

**Why Python was created?**

In late 1980s, Guido Van Rossum was working on the Amoeba distributed operating system group. He wanted to use an interpreted language like ABC (ABC has simple easy-to-understand syntax) that could access the Amoeba system calls. So, he decided to create a language that was extensible. This led to design of a new language which was later named Python.

**Why the name Python?**

No. It wasn't named after a dangerous snake. Rossum was fan of a comedy series from late seventies. The name "Python" was adopted from the same series "Monty Python's Flying Circus".

**Features of Python:**

**A simple language which is easier to learn**

Python has a very simple and elegant syntax. It's much easier to read and write Python programs compared to other languages like: C++, Java, C#. Python makes programming fun and allows you to focus on the solution rather than syntax.

If you are a newbie, it's a great choice to start your journey with Python.

**Free and open-source**

You can freely use and distribute Python, even for commercial use. Not only can you use and distribute software’s written in it, you can even make changes to the Python's source code.

Python has a large community constantly improving it in each iteration.

**Portability**

You can move Python programs from one platform to another, and run it without any changes.

It runs seamlessly on almost all platforms including Windows, Mac OS X and Linux.

**Extensible and Embeddable**

Suppose an application requires high performance. You can easily combine pieces of C/C++ or other languages with Python code.

This will give your application high performance as well as scripting capabilities which other languages may not provide out of the box.

**A high-level, interpreted language**

Unlike C/C++, you don't have to worry about daunting tasks like memory management, garbage collection and so on.

Likewise, when you run Python code, it automatically converts your code to the language your computer understands. You don't need to worry about any lower-level operations.

**Large standard libraries to solve common tasks**

Python has a number of standard libraries which makes life of a programmer much easier since you don't have to write all the code yourself. For example: Need to connect MySQL database on a Web server? You can use MySQLdb library using import MySQLdb .

Standard libraries in Python are well tested and used by hundreds of people. So you can be sure that it won't break your application.

**Object-oriented**

Everything in Python is an object. Object oriented programming (OOP) helps you solve a complex problem intuitively.

With OOP, you are able to divide these complex problems into smaller sets by creating objects.

**Applications of Python:**

**1. Simple Elegant Syntax**

Programming in Python is fun. It's easier to understand and write Python code. Why? The syntax feels natural. Take this source code for an example:

a = 2

b = 3

sum = a + b

print(sum)

**2. Not overly strict**

You don't need to define the type of a variable in Python. Also, it's not necessary to add semicolon at the end of the statement.

Python enforces you to follow good practices (like proper indentation). These small things can make learning much easier for beginners.

**3. Expressiveness of the language**

Python allows you to write programs having greater functionality with fewer lines of code. Here's a link to the source code of Tic-tac-toe game with a graphical interface and a smart computer opponent in less than 500 lines of code. This is just an example. You will be amazed how much you can do with Python once you learn the basics.

**4. Great Community and Support**

Python has a large supporting community. There are numerous active forums online which can be handy if you are stuck.

**5.2 Sample Code:**

from tkinter import messagebox

from tkinter import \*

from tkinter import simpledialog

import tkinter

from tkinter import filedialog

import matplotlib.pyplot as plt

import numpy as np

from tkinter.filedialog import askopenfilename

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

import pandas as pd

from sklearn.preprocessing import LabelEncoder

from sklearn.ensemble import GradientBoostingClassifier

from sklearn.preprocessing import normalize

from sklearn.metrics import confusion\_matrix

from sklearn.metrics import precision\_score

from sklearn.metrics import recall\_score

from sklearn.metrics import f1\_score

import seaborn as sns

main = tkinter.Tk()

main.title("A Machine Learning Approach for Early Detection of Fish Diseases by Analyzing Water Quality") #designing main screen

main.geometry("1300x1200")

global filename

global X\_train, X\_test, y\_train, y\_test

global X, Y

global classifier

global dataset

global le1, le2, le3, le4, le5

def upload(): #function to upload tweeter profile

global filename

global dataset

filename = filedialog.askopenfilename(initialdir="dataset")

text.delete('1.0', END)

text.insert(END,filename+" loaded\n\n");

dataset = pd.read\_csv(filename,encoding='iso-8859-1')

dataset.fillna(0, inplace = True)

text.insert(END,str(dataset))

def getAdaptiveFFT(data): #function to calculate FFT on recordings

return np.fft.fft(data)/len(data)

def preprocess():

text.delete('1.0', END)

global X

global Y

global dataset

global before\_features

global le1, le2, le3, le4, le5

le1 = LabelEncoder()

le2 = LabelEncoder()

le3 = LabelEncoder()

le4 = LabelEncoder()

le5 = LabelEncoder()

Y = []

#taking dataset as input which contains M cities, N indicators and T recordings

dataset['LOCATION'] = pd.Series(le1.fit\_transform(dataset['LOCATION'].astype(str)))

dataset['STATE'] = pd.Series(le2.fit\_transform(dataset['STATE'].astype(str)))

dataset['MONTH'] = pd.Series(le3.fit\_transform(dataset['MONTH'].astype(str)))

dataset['CIPerf\_(cfu/100\_ml)'] = pd.Series(le4.fit\_transform(dataset['CIPerf\_(cfu/100\_ml)'].astype(str)))

dataset['TERMOTOL\_COLIFORM\_(cfu/100\_ml)'] = pd.Series(le5.fit\_transform(dataset['TERMOTOL\_COLIFORM\_(cfu/100\_ml)'].astype(str)))

dataset = normalize(dataset.values) #dataset normalization

for i in range(len(dataset)): #looping each indicatior from dataset to calculate water quality

fft = getAdaptiveFFT(dataset[i]) #calculating FFT on indicators

signal = np.amax(fft) #getting max signal from FFT

signal = str(signal)

signal = signal[1:4]

signal = float(signal)

T = dataset[i,10] #getting indicator recording from dataset

if signal < T/2: #if signal < indicator/2 then calculated value will be 1 else0

Y.append(1)

else:

Y.append(0)

Y = np.asarray(Y)

X = dataset #dataset normalization

text.insert(END,"Dataset Preprocessing Completed\n\n")

text.insert(END,str(X))

def featureSelection():

global X\_train, X\_test, y\_train, y\_test

text.delete('1.0', END)

global X, Y

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, Y, test\_size=0.2)

text.insert(END,"Total records found in dataset : "+str(X.shape[0])+"\n\n")

text.insert(END,"Dataset train and test split details\n")

text.insert(END,"80% dataset records used for training : "+str(X\_train.shape[0])+"\n")

text.insert(END,"20% dataset records used for testing : "+str(X\_test.shape[0])+"\n")

coliform = X[:,6]

ecoli = X[:,7]

plt.figure(figsize=(10,6))

plt.grid(True)

plt.xlabel('Sampling Data')

plt.ylabel('Indicator Values')

plt.plot(coliform, 'ro-', color = 'blue')

plt.plot(ecoli, 'ro-', color = 'orange')

plt.legend(['Coliform', 'Ecoli'], loc='upper left')

#plt.xticks(wordloss.index)

plt.title('Coliform & Ecoli Virus Found in Dataset')

plt.show()

def runGradientBoosting():

text.delete('1.0', END)

global classifier

global X\_train, X\_test, y\_train, y\_test

rfc = GradientBoostingClassifier()

rfc.fit(X\_train, y\_train)

predict = rfc.predict(X\_test)

classifier = rfc

precision = precision\_score(y\_test, predict,average='macro') \* 100

recall = recall\_score(y\_test, predict,average='macro') \* 100

fscore = f1\_score(y\_test, predict,average='macro') \* 100

accuracy = accuracy\_score(y\_test,predict)\*100

text.insert(END,"Gradient Boosting Accuracy : "+str(accuracy)+"\n")

text.insert(END,"Gradient Boosting Precision : "+str(precision)+"\n")

text.insert(END,"Gradient Boosting Recall : "+str(recall)+"\n")

text.insert(END,"Gradient Boosting FSCORE : "+str(fscore)+"\n\n")

labels = ['Healthy Fish', 'Diseases Detected']

conf\_matrix = confusion\_matrix(y\_test, predict)

ax = sns.heatmap(conf\_matrix, xticklabels = labels, yticklabels = labels, annot = True, cmap="viridis" ,fmt ="g");

ax.set\_ylim([0,len(labels)])

plt.title("Gradient Boosting Confusion matrix")

plt.ylabel('True class')

plt.xlabel('Predicted class')

plt.show()

def predict():

global pca

global le1, le2, le3, le4, le5

text.delete('1.0', END)

filename = filedialog.askopenfilename(initialdir="dataset")

test = pd.read\_csv(filename,encoding='iso-8859-1')

test.fillna(0, inplace = True)

temp = test.values

test['LOCATION'] = pd.Series(le1.transform(test['LOCATION'].astype(str)))

test['STATE'] = pd.Series(le2.transform(test['STATE'].astype(str)))

test['MONTH'] = pd.Series(le3.transform(test['MONTH'].astype(str)))

test['CIPerf\_(cfu/100\_ml)'] = pd.Series(le4.transform(test['CIPerf\_(cfu/100\_ml)'].astype(str)))

test['TERMOTOL\_COLIFORM\_(cfu/100\_ml)'] = pd.Series(le5.transform(test['TERMOTOL\_COLIFORM\_(cfu/100\_ml)'].astype(str)))

test = test.values

#test = normalize(test)

y\_pred = classifier.predict(test)

print(y\_pred)

for i in range(len(test)):

predict = y\_pred[i] #np.argmax(y\_pred[i])

if predict == 0:

text.insert(END,"X=%s, Predicted = %s" % (temp[i], '=====> Healthy Fish')+"\n\n")

else:

text.insert(END,"X=%s, Predicted = %s" % (temp[i], '=====> Fish will get affected by Coliform & Ecoli Virus')+"\n\n")

font = ('times', 16, 'bold')

title = Label(main, text='A Machine Learning Approach for Early Detection of Fish Diseases by Analyzing Water Quality')

title.config(bg='darkviolet', fg='gold')

title.config(font=font)

title.config(height=3, width=120)

title.place(x=0,y=5)

font1 = ('times', 12, 'bold')

text=Text(main,height=20,width=150)

scroll=Scrollbar(text)

text.configure(yscrollcommand=scroll.set)

text.place(x=50,y=120)

text.config(font=font1)

font1 = ('times', 12, 'bold')

uploadButton = Button(main, text="Upload Water Quality Dataset", command=upload)

uploadButton.place(x=50,y=550)

uploadButton.config(font=font1)

processButton = Button(main, text="Preprocess & Normalize Dataset", command=preprocess)

processButton.place(x=290,y=550)

processButton.config(font=font1)

featureButton = Button(main, text="Features Selection", command=featureSelection)

featureButton.place(x=570,y=550)

featureButton.config(font=font1)

gbButton = Button(main, text="Train Gradient Boosting Algorithm", command=runGradientBoosting)

gbButton.place(x=770,y=550)

gbButton.config(font=font1)

predictButton = Button(main, text="Predict Fish Condition", command=predict)

predictButton.place(x=50,y=600)

predictButton.config(font=font1)

main.config(bg='sea green')

main.mainloop()

**6. TESTING**

**Implementation and Testing:**

Implementation is one of the most important tasks in project is the phase in which one has to be cautions because all the efforts undertaken during the project will be very interactive. Implementation is the most crucial stage in achieving successful system and giving the users confidence that the new system is workable and effective. Each program is tested individually at the time of development using the sample data and has verified that these programs link together in the way specified in the program specification. The computer system and its environment are tested to the satisfaction of the user.

## **Implementation**

## The implementation phase is less creative than system design. It is primarily concerned with user training, and file conversion. The system may be requiring extensive user training. The initial parameters of the system should be modifies as a result of a programming. A simple operating procedure is provided so that the user can understand the different functions clearly and quickly. The different reports can be obtained either on the inkjet or dot matrix printer, which is available at the disposal of the user. The proposed system is very easy to implement. In general implementation is used to mean the process of converting a new or revised system design into an operational one.

## **Testing**

Testing is the process where the test data is prepared and is used for testing the modules individually and later the validation given for the fields. Then the system testing takes place which makes sure that all components of the system property functions as a unit. The test data should be chosen such that it passed through all possible condition. Actually testing is the state of implementation which aimed at ensuring that the system works accurately and efficiently before the actual operation commence. The following is the description of the testing strategies, which were carried out during the testing period.

### **System Testing**

Testing has become an integral part of any system or project especially in the field of information technology. The importance of testing is a method of justifying, if one is ready to move further, be it to be check if one is capable to with stand the rigors of a particular situation cannot be underplayed and that is why testing before development is so critical. When the software is developed before it is given to user to use the software must be tested whether it is solving the purpose for which it is developed. This testing involves various types through which one can ensure the software is reliable. The program was tested logically and pattern of execution of the program for a set of data are repeated. Thus the code was exhaustively checked for all possible correct data and the outcomes were also checked.

**Module Testing**

To locate errors, each module is tested individually. This enables us to detect error and correct it without affecting any other modules. Whenever the program is not satisfying the required function, it must be corrected to get the required result. Thus all the modules are individually tested from bottom up starting with the smallest and lowest modules and proceeding to the next level. Each module in the system is tested separately. For example the job classification module is tested separately. This module is tested with different job and its approximate execution time and the result of the test is compared with the results that are prepared manually. The comparison shows that the results proposed system works efficiently than the existing system. Each module in the system is tested separately. In this system the resource classification and job scheduling modules are tested separately and their corresponding results are obtained which reduces the process waiting time.

**Integration Testing**

After the module testing, the integration testing is applied. When linking the modules there may be chance for errors to occur, these errors are corrected by using this testing. In this system all modules are connected and tested. The testing results are very correct. Thus the mapping of jobs with resources is done correctly by the system.

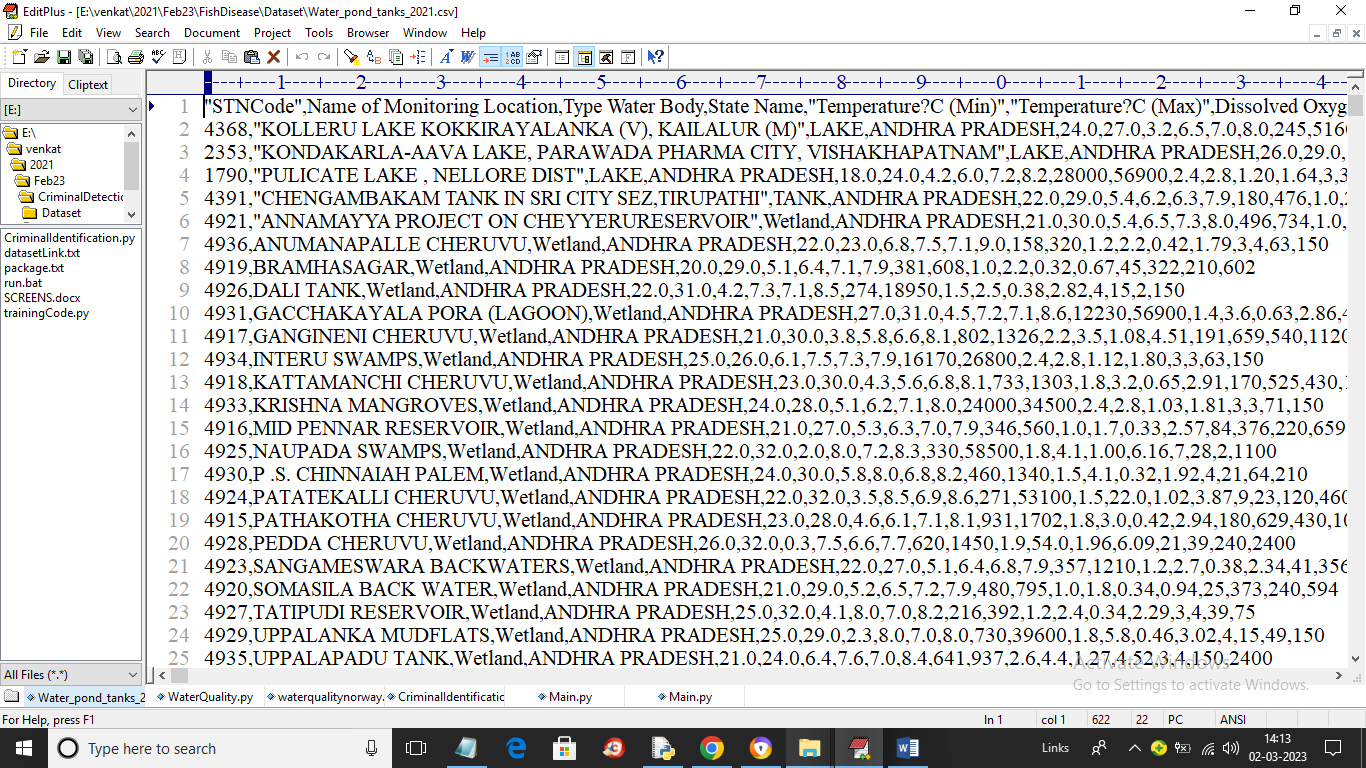
**Acceptance Testing**

When that user fined no major problems with its accuracy, the system passers through a final acceptance test. This test confirms that the system needs the original goals, objectives and requirements established during analysis without actual execution which elimination wastage of time and money acceptance tests on the shoulders of users and management, it is finally acceptable and ready for the operation

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Test Case Id** | **Test Case Name** | **Test Case Desc.** | **Test Steps** | | | **Test Case Status** | **Test Priority** |
| **Step** | **Expected** | **Actual** |
| 01 | Upload Water Quality Dataset | Test whether Water Quality Dataset is uploaded or not into the system | If the water Dataset may not uploaded | We cannot do further operations | Water Quality Dataset uploaded we will do further operations | High | High |
| 02 | Preprocess& Normalized Dataset | Test whether the Pre-process & NormalizedDataset Successfully or not | If the Pre-process & Normalized Dataset may not Run Successfully | We cannot do further operations | we will do further  operations | High | High |
| 03 | Feature Selection | Test whether the Feature Selection  Successfully or not | If the  Feature Selection  may not Successfully | We cannot do further operations | we will do further  operations | High | High |
| 04 | Train Gradient Boosting Algorithm | Test whether Gradient Boosting Algorithm Run Successfully or not | If the  Gradient Boosting Algorithm may not Run Successfully | We cannot do further operations | we will do further  operations | High | High |
| 05 | Predict Fish Condition | Test Fish condition whether Successfully or not | If the  Predict the fish condition may not Successfully | We cannot do further operations | we will do further  operations | High | High |

**7.SCREENSHOTS**:

Below screen showing dataset details



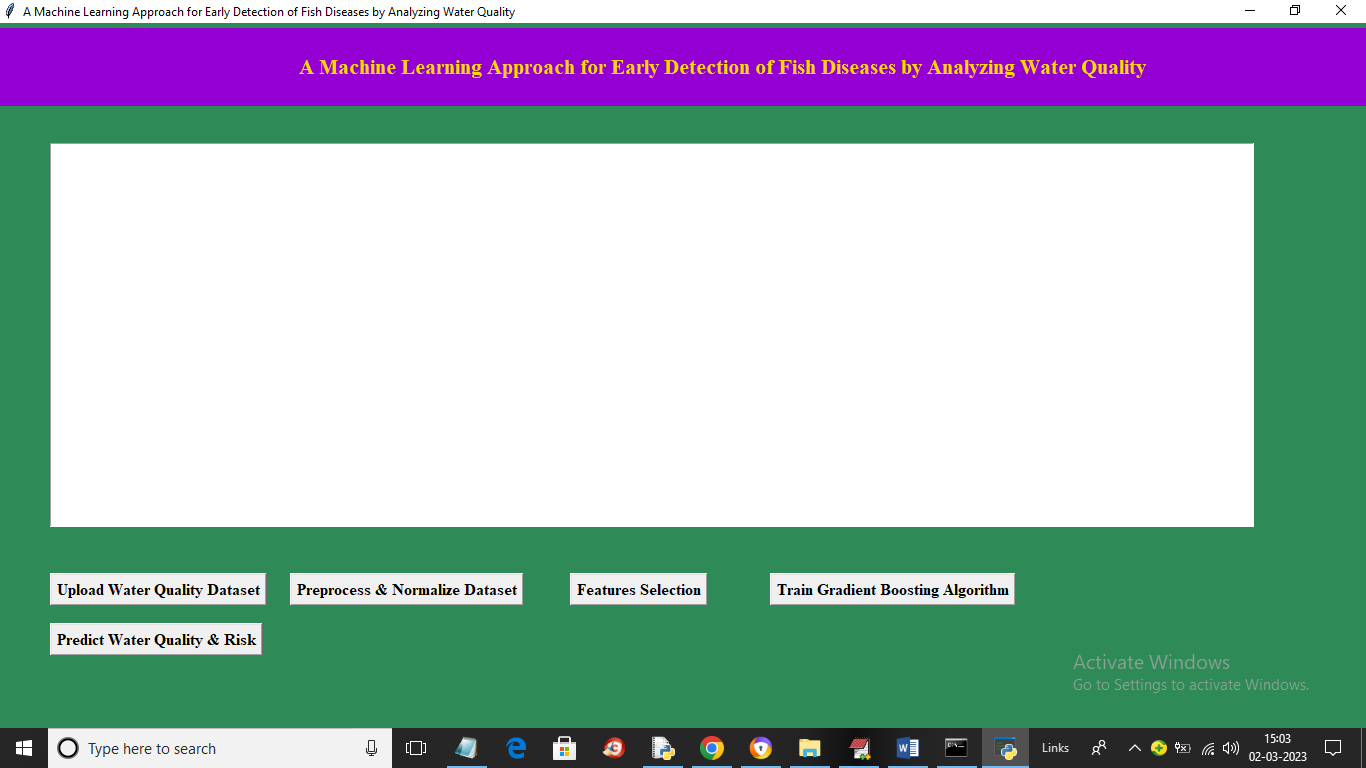
In above dataset screen first row contains dataset column names and remaining rows contains dataset values such as presence of chemical and ECOLI and other viruses and by using this dataset we will train Gradient Boosting algorithm for fish disease prediction.

To implement this project we have designed following modules

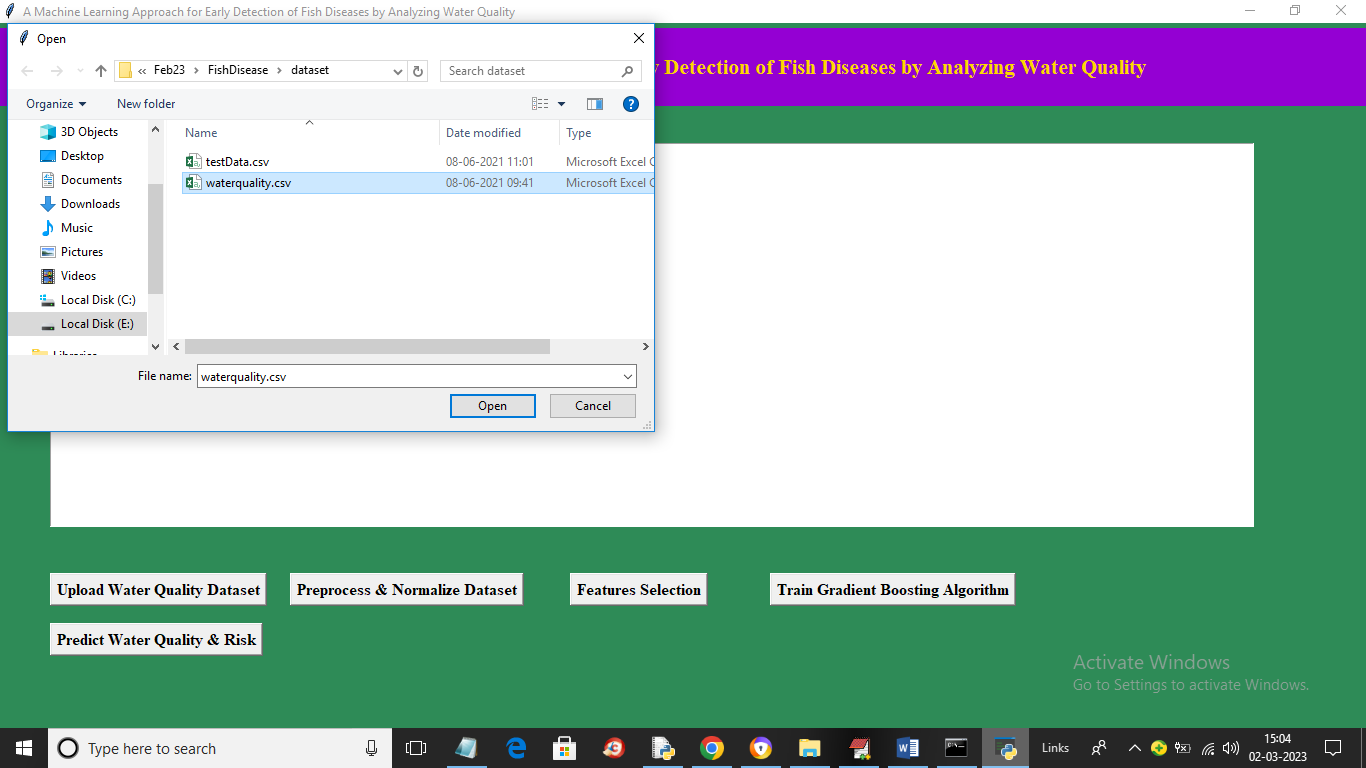
1. Upload Water Quality Dataset: using this module we will upload dataset to application
2. Preprocess & Normalize Dataset: using this module will convert all non-numeric data to numeric data and then normalize all values
3. Features Selection: using this module application will select X training features and Y class label and then split dataset into train and test where application using 80% dataset for training and 20% for testing
4. Train Gradient Boosting Algorithm: 80% training data will be input to Gradient Boosting algorithm to train a model and this model will be applied on test data to calculate prediction accuracy
5. Predict Fish Condition: using this module we will upload test data and then algorithm will predict weather fish is healthy or disease affected.

**SCREENSHOTS:**

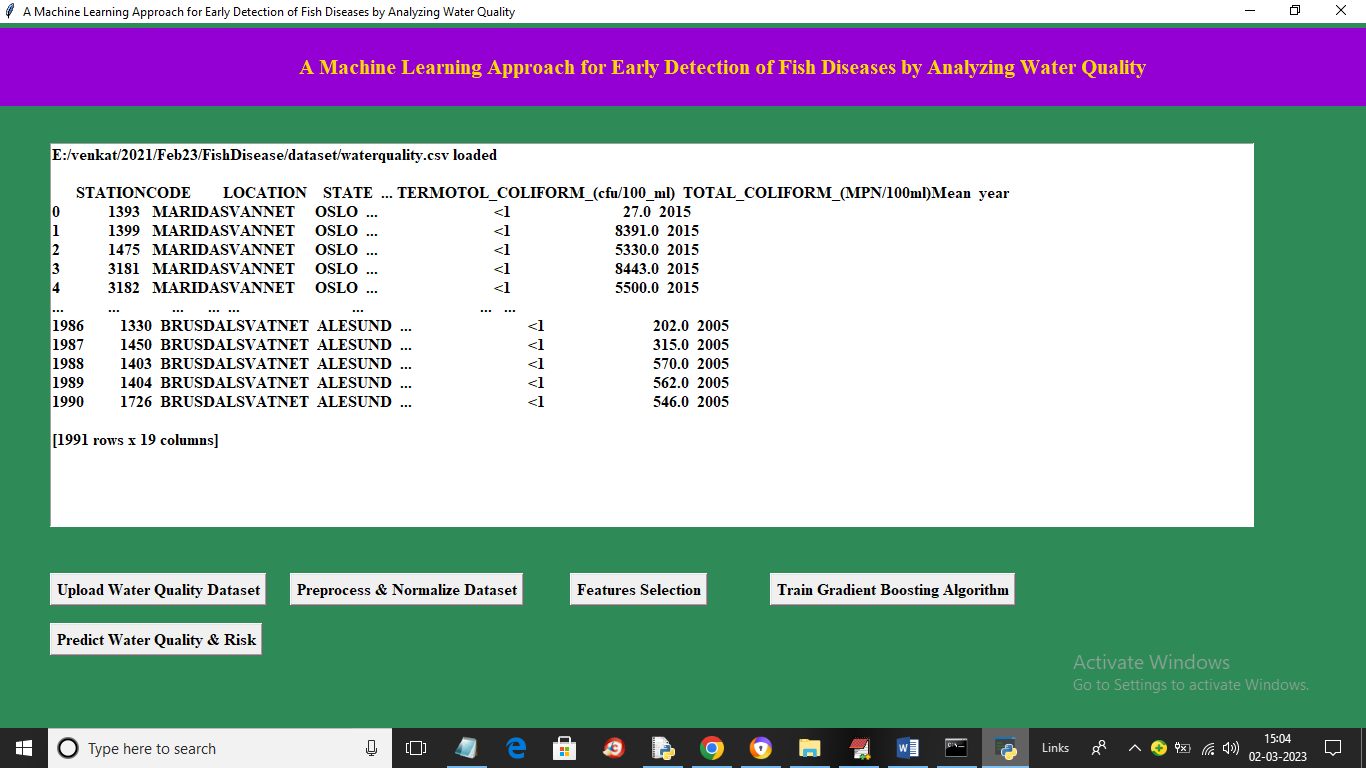
To run project double click on ‘run.bat’ file to get below screen



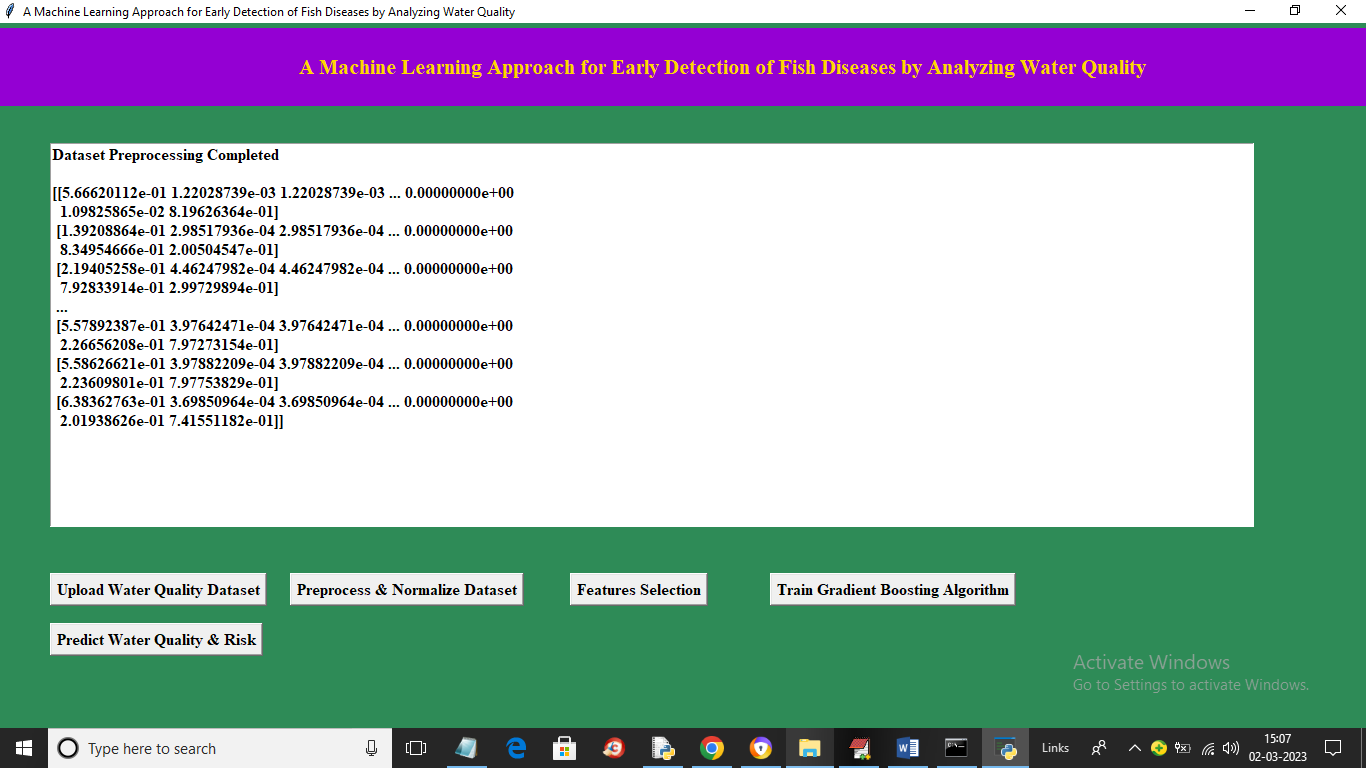
In above screen click on ‘Upload Water quality Dataset’ button to upload dataset and get below output



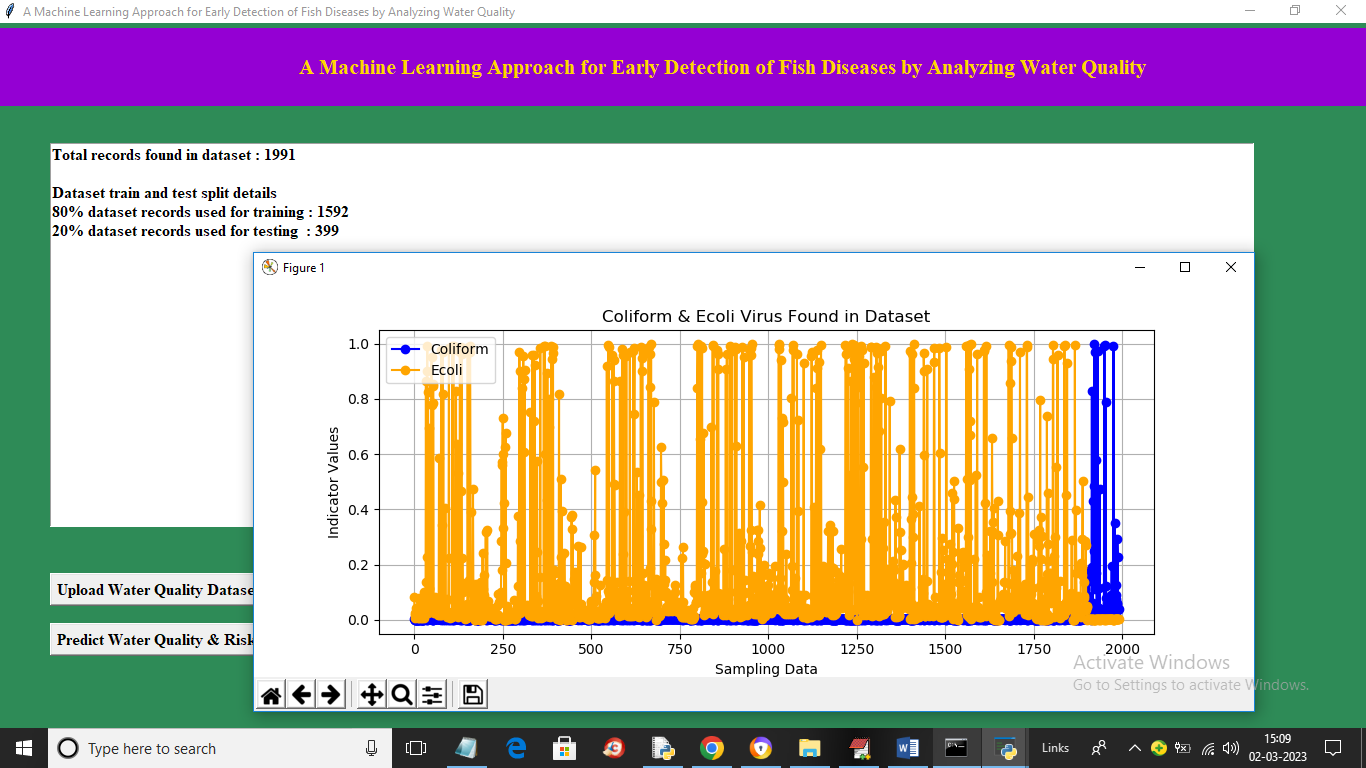
In above screen click on ‘Open’ button to load dataset and get below output



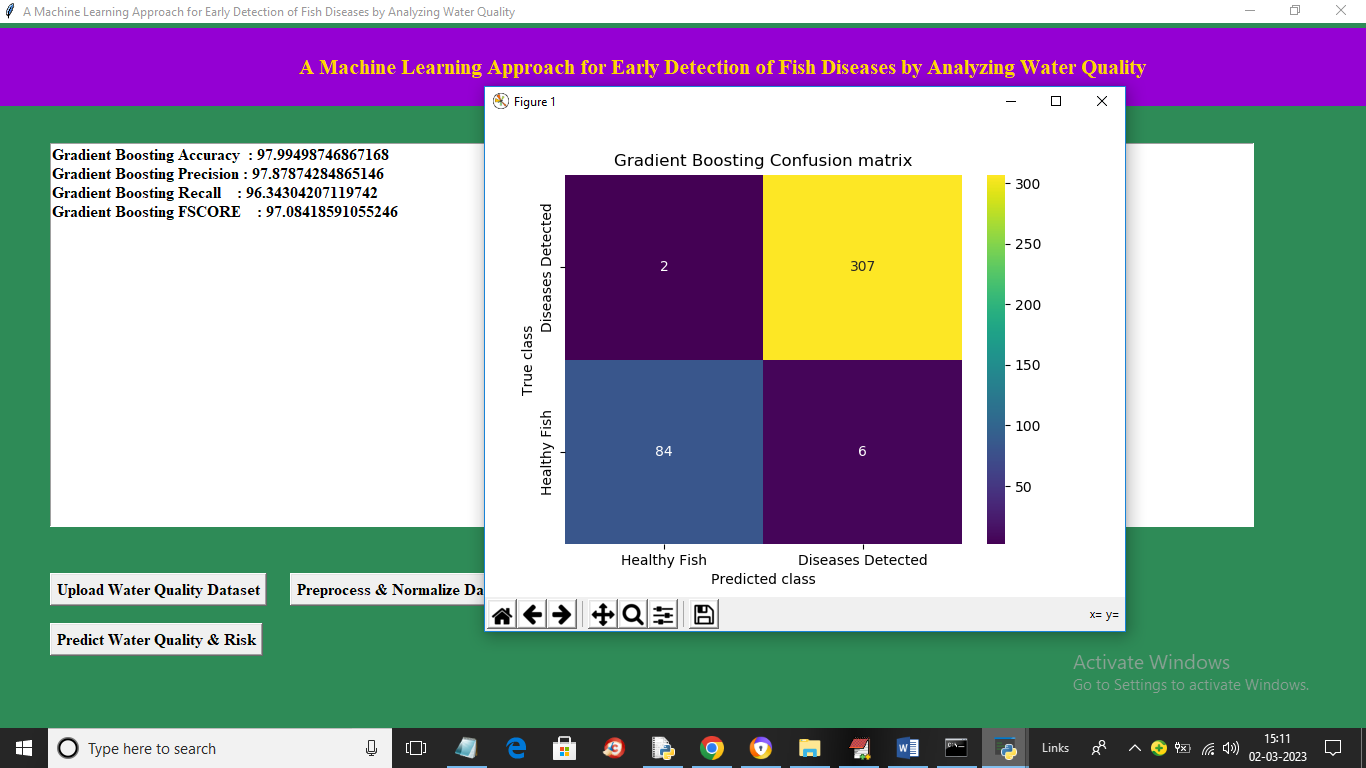
In above screen dataset loaded and we can see dataset contains numeric and non-numeric values and machine learning algorithms accept only numeric dataset so by applying label encoder class we can convert non-numeric data to numeric values so click on ‘Preprocess & Normalize Dataset’ button to get below output



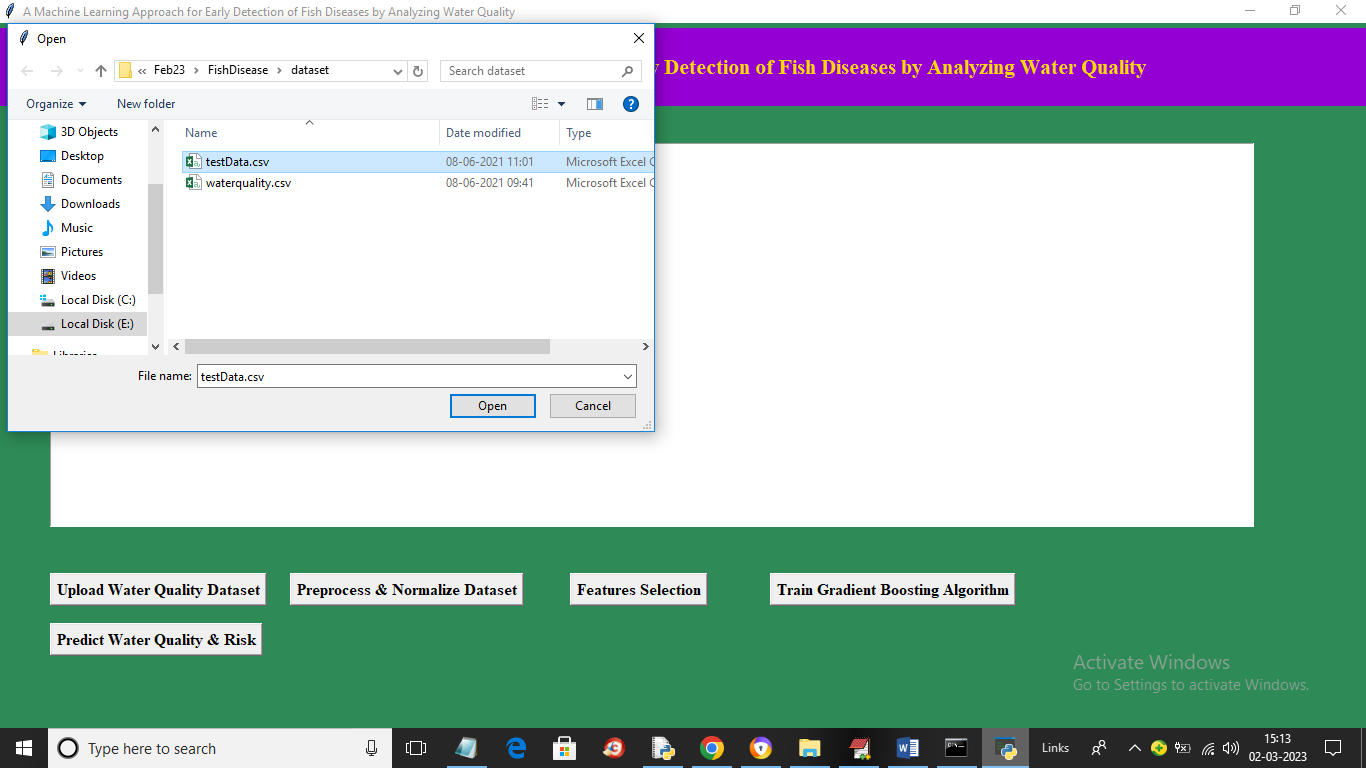
In above screen all values are converted to numeric format and now click on ‘Features Selection’ button to extract X and Y features from dataset and then split into train and test values and get below output



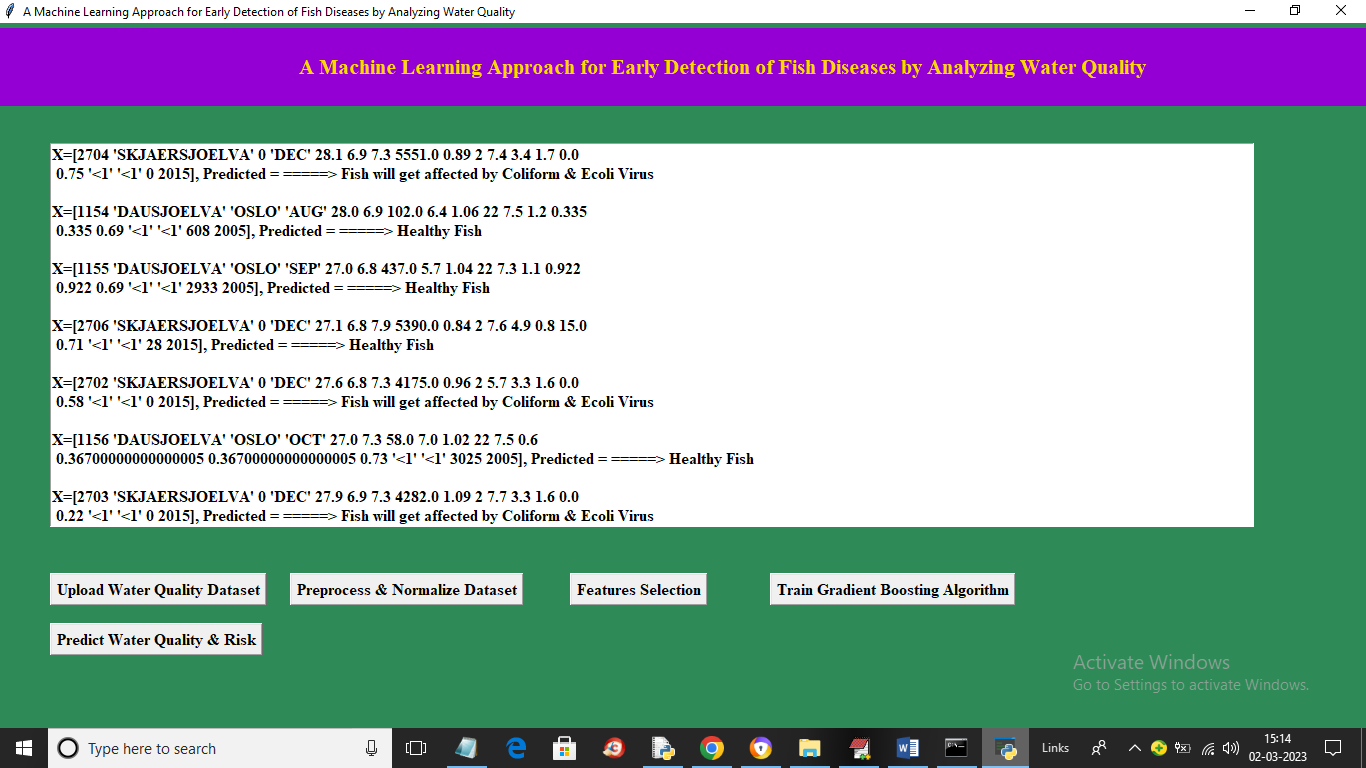
In above screen we can see dataset total values and then we can see training and testing dataset size and in graph x-axis represents number of records and y-axis represents presence quality of ‘Coliform and Ecoli’ virus where yellow line is for Ecoli and blue line for Coliform and now close above graph and then click on ‘Train Gradient Boosting Algorithm’ button to train algorithm and get below output



In above screen with Gradient Boosting we got 97% accuracy and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and blue colur boxes represents Incorrect prediction count which is 2 only and different colour boxes contains correct prediction count. Now close above graph and then click on ‘Predict Water Quality & Risk’ button to get below output



In above screen selecting and uploading ‘testData.csv’ file and then click on ‘Open’ button to load dataset and get below output



In above screen in square bracket we can see test data values and after arrow symbol we can see predicted values as healthy or disease affected fish.

**8. CONCLUSION:**

In this paper, authors built a neural network model to accomplish fish detection. To support the training process with enough dataset, the data augmentation approach was conducted. Dropout algorithm was selected to solve the overfitting problem. Moreover, loss function was refined to update the parameters inside the network. By these approaches, both the training time and the training loss were reduced dramatically. To summarize the contribution of this article: (1) Establish the data set to include real blur ocean water condition; (2) Revise loss function and other parameters in CNN to explore an applicable solution for fish detection; (3) The system is targeted at an embedded system for AUV design with all possible optimizations.

**9. REFERENCES:**

[1]Greenfacts.Org, ―Fisheries Latest data.‖ Available at: https://www.greenfacts.org/en/fisheries/l-2/06- fish-consumption.htm Last accessed: May 3, 2020

[2] DoF. 2018. ―Yearbook of Fisheries Statistics of Bangladesh, 2017-18. Fisheries Resources Survey System (FRSS).‖ Department of Fisheries. Bangladesh: Ministry of Fisheries, vol. 35, (2018),pp. 129.

[3] N.Bagum, M. Monir and M.S. Khan., ―Present Status of Fish Diseases and Economic Losses Due to Incidence of Disease in Rural Freshwater Aquaculture of Bangladesh.‖ J. Innov. Dev. Strategy,vol. 7, no. 3, (2013), 48-53.

[4] M.Chaurasia, I.Gupta, R.Kumar, P.Prasad and R. A.Sohony,―Water quality analysis of surface water: A Web approach.‖ Environmental Monitoring and Assessment, vol. 185, (2013), pp. 5987–5992.

[5] N.Rahmanian, S. B.Hajar, M.Homayoonfard, N. J.Ali, M.Rehan, Y.Sadefand A. S.Nizami, ―Analysis of Physiochemical Parameters to Evaluate the Drinking Water Quality in the State of Perak, Malaysia.‖ Journal of Chemistry, (2015).

[6] C.Nagamani, C. S.Devi, and A.Shalini,―Physico-chemical analysis of water samples.‖ International Journal of Scientific & Engineering Research, vol. 6, no. 1, (2015), pp. 2149-2155.

[7] M.Tanekhy,―Fish Health and Diseases.‖ ICMALPS 2015, Alexandria University, Egypt, (2015), pp. 222-232.

[8] N.Emanet, H. R.Öz, N.Bayram and D. A.Delen,―Comparative analysis of machine learning methods for classification type decision problems in healthcare.‖ Decision Analytics, (2014).

[9] P.Jeatrakuland K. W.Wong,―Comparing the performance of different neural networks for binary classification problems.‖ Eighth International Symposium on Natural Language Processing, (2009).

[10] K. L.Mosier, L. J.Skitka, S.Heers, and M.Burdick,―Automation Bias: Decision Making and Performance in High-Tech Cockpits.‖ The International Journal of Aviation Psychology,vol. 8, no. 1, (2009), pp. 47-63.

[11] J. G.Harris and T. H.Davenport,―Automated decision making comes of age.‖ MIT Sloan Management Review, (2005).

[12] C. M.Han, S. Y.Yang, H.Ceongand J. S.Park,―Clinical Sign-Based Fish Disease Diagnosis Aid System.‖ Electrical Engineering and Control, vol. 98,(2011), pp. 1039-1046.

[13] P.Mădălina And I. G.Breabăn,―Water Quality Index – An Instrument for Water Resources Management.‖ AERUL ŞI APA COMPONENTE ALE MEDIULUI, (2014),pp. 391-398.

[14] I. A.Ani,―Mathematical Computation of Water Quality Index for the Assessment of Al-Hilla River Ecosystem.‖ International Journal of Civil Engineering and Technology, vol. 10, no. 1, (2019), pp. 1862-1869.

[15] S.Chandra, S. S.Asadiand M. V. S.Raju,―Estimation of Water Quality Index by Weighted Arithmetic Water Quality Index Method: A Model Study.‖ International Journal of Civil Engineering and Technology, vol. 8, no. 4, (2017), pp.1215-1222.

[16] N. H.Omer,―Water Quality Parameters. Water Quality – Science.‖ Assessments and Policy,(2019).

[17] A.Appavu, S.Thangavelu, S.Muthukannan, J. S.JesudossandB.Pand,―Study of Water Quality Parameters of Cauvery River Water in Erode Region.‖ Journal of Global Biosciences, vol. 5, no. 9,(2016),pp. 4556-4567.

[18] S. S.Sagar, R. P.Chavan, C. L.Patil, D. N.Shinde and S. S.Kekane,―Physico-chemical parameters for testing of water- A review.‖ International Journal of Environmental Sciences, vol. 3, no.3, (2012), pp. 1194-1207.

[19] Z.Wu, D.Zhang, Y.Cai, X.Wang, L.Zhang and Y.Chen,―Water quality assessment based on the water quality index method in Lake Poyang: The largest freshwater lake in China.‖ Scientific Reports, vol.7, no. 1, (2017).

[20] A.Natekinand A.Knoll,―Gradient Boosting Machines, A Tutorial.‖ Frontiers in Neurorobotics, (2013).

[21] J. H.Friedman,―Stochastic Gradient Boosting.‖ Computational Statistics & Data Analysis, vol. 38, no. 4,(2002), pp. 367-378.

[22] J. H.Friedman,―Greedy Function Approximation: A Gradient Boosting Machine.‖ Annals of Statistics,vol. 29, no. 5, (2001), pp.1189-1232.

[23] X.Qin, F.Gao And G. Chen,―Wastewater quality monitoring system using sensor fusion and machine learning techniques.‖ Water Research, vol. 46, no. 4, (2012), pp. 1133-1144.

[24] Y. H.Kim, J.Im, H. K.Ha., J. K.Choi and S.Ha,―Machine learning approaches to coastal water quality monitoring using GOCI satellite data.‖GIScience& Remote Sensing, vol. 2, (2014), pp. 158-174.

[25] D. G. Shah and P. S. Patel,―Water quality of some drinking water samples of Kathalal territory.‖ Guidelines for Drinking-water Quality, 3rd ed.(2012).