**INDEX**

**TITLES PAGE NO**

**CONTENTS**

[ABSTRACT vi](#_TOC_250016)

1. [INTRODUCTION 1](#_TOC_250015)
2. LITERATURE SURVEY 5
3. [SYSTEM](#_TOC_250007) ANALYSIS 14
   1. 3.1 [EXISTING SYSTEM 16](#_TOC_250006)
   2. 3.2 [PROPOSEDSYSTEM 16](#_TOC_250005)
4. [FEASIBILITYSTUDY 17](#_TOC_250004)
   1. 4.1 [ECONOMICALFEASIBILITY 17](#_TOC_250003)
   2. 4.2 [TECHNICALFEASIBILITY 17](#_TOC_250002)
   3. 4.3 [SOCIALFEASIBILITY 18](#_TOC_250001)
5. [SYSTEMREQUIREMENTS 19](#_TOC_250000)

## SYSTEMDESIGN 20

* 1. 6.1 SYSTEMARCHITECTURE 20
  2. 6.2 DATAFLOW DIAGRAM 20
  3. 6.3 UMLDIAGRAMS 22

## IMPLEMENTATION 27

* 1. 7.1 MODULES 27
  2. 7.2 SAMPLE CODE 28

## SYSTEMTESTING 29

* 1. 8.1 UNITTESTING 31
  2. 8.2 INTEGRATION TESTING 32
  3. 8.3 ACCEPTANCETESTING 33

## INPUT DESIGN ANDOUTPUTDESIGN 34

* 1. 9.1 INPUTDESIGN 34
  2. 9.2 OUTPUTDESIGN 35

## SCREENSHOTS 37

## FUTUREWORK 50

## CONCLUSION 51

## BIBLOGRAPHY 52

**ABSTRACT**:

Constructing Sustainable Smart Water Supply systems are facing serious challenges all around the world with the fast expansion of modern cities. Water quality is influencing our life ubiquitously and prioritizing all the urban management. Traditional urban water quality control mostly focused on routine tests of quality indicators, which include physical, chemical and biological groups. However, the inevitable delay for biological indicators has increased the health risk and leads to accidents such as massive infections in many big cities. In this paper, we first analyze the problem, technical challenges, and research questions. Then we provide a possible solution by building a risk analysis framework for the urban water supply system. It takes indicator data we collected from industrial processes to perceive water quality changes, and further for risk detection. In order to provide explainable results, we propose an Adaptive Frequency Analysis (Adp-FA) method to resolve the data using indicators’ frequency domain information for their inner relationships and individual prediction. We also investigate the scalability properties of this method from indicator, geography and time domains. For the application, we select industrial quality data sets collected from a Norwegian project in 4 different urban water supply systems, as Oslo, Bergen, Strømmen and Alesund. We employ the proposed method to test spectrogram, prediction accuracy and time ˚ consumption, comparing with classical Artificial Neural Network and Random Forest methods. The results show our method better perform in most of the aspects. It is feasible to support industrial water quality risk early warnings and further decision support.

**Index Terms**—Sustainable Water Supply, Water Quality Control, Data Perception, Risk Evaluation, Frequency Analysis, Scalability.

**1. INTRODUCTION**

During the latest years of 21st century, two important phenomena have been emerging: urbanization and information technologies. The United Nations (UN) Department of Economic and Social Affairs (DESA) reports that for the first time ever, the majority of the world’s population lives in cities, and this proportion continues to grow with projections of 68% by 2050 [1]. Urban water supply systems are the most critical infrastructure all over the world. A Smart Water Supply system that integrates sensors, controllers, cloud computing and data technologies, are essential for the development of sustainable smart cities in the future. It is aiming to provide safe, stable and sufficient water for the increasing requirements in many expanding cities. However, the urban water quality is facing serious challenges from industrial, agriculture and social pollution. To emphasize the importance of water safety in urban supply is nowadays a truism. In 2015, the United Nations Development Programme published the Sustainable Development Goals (SDGs), including Clean Water and Sanitation as Goal 6 [2]. The dwindling supplies of safe drinking water is a major problem impacting every continent, around 2.1 billion people [3]. The concerns of the modern society regarding this issue are reflected in numerous legislative initiatives in this field, such as the European Union Water Framework Directive [4], United States Clean Water Act [5]. The prevalent water supply process can be divided into 3 sections, including water source management, treatment, and distribution. Traditional water quality control is taken after water treatment. But the current water sources are mainly groundwater and surface water. They are significantly prone to chemical and microbial contamination. The quality control after the water treatment apparently delays the risk detection and reduces the response time to take preventive measures. In Norway, the new national standard for water quality in the source area is in progress [6] [7]. Water quality refers to physical, chemical, and biological characteristics as indicators. Among the water quality indicators, biological indicators have a more direct impact over people’s health. Most of the national standards are made on biological indicator levels. Typical indicators include coliform, escherichia coli (Ecoli), intestinal enterococci (Int), clostridium perfringens (ClPerf), etc. Further treatment actions are made according to the test results [8]. Coliform itself is not usually causing serious illness, but their presence is a signal to indicate other active pathogenic organisms presentation. Some special types of Ecoli are the reason for water poisoning. Int is more dangerous to cause urinary tract infections, bacterial endocarditis, diverticulitis, and meningitis. The tests of biological indicators are primarily based on the bacterial culture in the laboratory. This process can take up to 24-48 hours. Compare to the effectual time on the human body, the danger is much higher than other indicators. In Norway, the giardia outbreak in Bergen 2004 affected more than 2500 people including young children due to the bacteria test delay results. Therefore, we have a severe requirement for early risk detection in smart water supply systems. There have been some trial work for water quality control based on data. In 2018, Hounslow [9] interpreted multiple water quality indicators. In 2015, Yagur-Kroll et al [10] showed a group of general bacterial sensor cells for water quality monitoring. There is some research work to use data for water quality prediction. Holger et al [11] designed an Artificial neural network to predict salinity level for an Australian river named Murray. Based on the data collected at Astane station in Sefidrood River, Iran, Orouji and his colleagues designed a series of models as ANFIS, GA and Shuffled FLA to predict water quality chemical indicators (sodium, potassium, magnesium, etc) in [12] [13] [14]. Chang et al [15] proposed a systematic analysis framework to predict NH3-H levels for Dahan River in Taiwan, China. However, their work is generally on individual quality indicator and ignored the inner relationship between them. Today the advanced ubiquitous sensing technologies cut across many areas of modern research, industry and daily life [16]. They offer the ability to detect, transmit and measure more environmental indicators. A sustainable smart water supply system adopts various sensors in order to manage resources and monitor water quality efficiently. In this process, data becomes an important tool to improve our understanding of existing systems. By observing data itself, through the appropriate methods, we can perceive the changes in our water supply system. In practice, we applied many different sensors in the water source areas, including multiple sensors for pH, temperature, conductivity, etc. The massive data collected by those low-cost sensors plus the recent data analysis technologies, help us greatly improve the water quality control process. At present, zettabytes of data are collected by these numerous sensors [17] [18]. At the same time, stronger data analysis tools have been developed. Water quality indicators are typical spatiotemporal variables. The analysis can be divided into correlation analysis and numerical prediction analysis. Early works with correlation analysis include Hardoon et al [19] used Kernel Correlation Analysis method for web page images and associated texts. For multiple variables, Principal component analysis (PCA) is often the first choice. Jolliffe et al [20] reviewed classical PCA and newly developed methods such as Robust PCA, Adaptive PCA,etc. Luo et al [21] applied tensor model in correlation analysis for gait recognition. But they did not consider the correlations in the time domain. As for spatiotemporal data analysis, most of the recent work is facing very huge data sets. For example, Gudmundsson et al [22] surveyed the player’s trajectories in team-sports with respect to behavior and prediction. Lecun et al [23] proposed the pioneer concept for Deep Learning to deal with spatiotemporal data. Liu et al [24] analyzed 3D human actions with modern LSTM method. Laptev et al [25] detects anomalies in the industrial platform data. However, their work has to rely on large training sets, which we cannot provide currently in water supply systems. In addition, the explanation with those methods cannot support the requirements for industrial use. In this paper, we introduce our preliminary experience in Norway. First, we analyze the problem, challenges and research questions. Second, based on water quality data collected from water supply systems, we propose a framework for water quality analysis with data perception. Third, we provide an adaptive frequency analysis method for risk detection and prediction. This method is scalable in multiple domains, including water quality indicators, geography and time. Furthermore, by application, we select industrial quality data sets collected from a national project in 4 different Norwegian city water supply systems, as Oslo, Bergen, Strømmen and Alesund. We show our preliminary ˚ findings of the frequency property relationship between water quality indicators, as well as risk detection, prediction and evaluation analysis. The results are compared also with classical Artifical Neural Network and Random Forest in their prediction accuracy and time consumption. In addition, scalability in time domain is also analyzed. There are several visible motivations for this research. First, it takes the advantage of the modern data analysis technologies to solve a water quality control problem in future Sustainable Smart Water Supply systems, especially in transferring the knowledge across different indicator, geography and time domains. Second, it copes with the practical water source monitoring process, applies the data directly collected from the industrial process. This avoids questions such as laboratory data reliability and industrial applicability. This is also valuable to the current water supply in urban infrastructure systems. Third, it builds the connection between easily accessible physical and chemical indicators with biological indicators that are critical to water quality risk. Fourth, this work provides the support for further reasoning of decision-making process and analysis over the pollution from industrial and residential activities in the corresponding water source areas.

**2. LITERATURE SURVEY**

**2.1 Urbanization and climate change impacts on surface water quality: Enhancing the resilience by reducing impervious surfaces**

**Authors:** **S. Franco, V. Gaetano, and T. Gianni** **Abstract:** Climate change and urbanization are key factors affecting the future of water quality in urbanized catchments. The work reported in this paper is an evaluation of the combined and relative impact of climate change and urbanization on the water quality of receiving water bodies in the context of a highly urbanized watershed served by a combined sewer system (CSS) in northern Italy. The impact is determined by an integrated modelling study involving two years of field campaigns. The results obtained from the case study show that impervious urban surfaces and rainfall intensity are significant predictors of combined sewer overflows (CSOs) and consequently of the water quality of the receiving water body. Scenarios for the year 2100 demonstrate that climate change combined with increasing urbanization is likely to lead to severe worsening of river water quality due to a doubling of the total phosphorus load from CSOs compared to the current load. Reduction in imperviousness was found to be a suitable strategy to adapt to these scenarios by limiting the construction of new impervious areas and decreasing the existing areas by only 15%. This information can be further utilized to develop future designs, which in turn should make these systems more resilient to future changes in climate and urbanization.

**2.2** **Sustainable development ´ goals: A need for relevant indicators**

**Authors:** **T. Hak, S. Janou ´ skov ˇ a, and B. Moldan** **Abstract:** At the UN in New York the Open Working Group created by the UN General Assembly proposed a set of global Sustainable Development Goals (SDGs) which comprises 17 goals and 169 targets. Further to that, a preliminary set of 330 indicators was introduced in March 2015. Some SDGs build on preceding Millennium Development Goals while others incorporate new ideas. A critical review has revealed that indicators of varied quality (in terms of the fulfilment certain criteria) have been proposed to assess sustainable development. Despite the fact that there is plenty of theoretical work on quality standards for indicators, in practice users cannot often be sure how adequately the indicators measure the monitored phenomena. Therefore we stress the need to operationalise the Sustainable Development Goals’ targets and evaluate the indicators’ relevance, the characteristic of utmost importance among the indicators’ quality traits. The current format of the proposed SDGs and their targets has laid a policy framework; however, without thorough expert and scientific follow up on their operationalisation the indicators may be ambiguous. Therefore we argue for the foundation of a conceptual framework for selecting appropriate indicators for targets from existing sets or formulating new ones. Experts should focus on the “indicator-indicated fact” relation to ensure the indicators’ relevance in order for clear, unambiguous messages to be conveyed to users (decision- and policy-makers and also the lay public). Finally we offer some recommendations for indicators providers in order to contribute to the tremendous amount of conceptual work needed to lay a strong foundation for the development of the final indicators framework.

# 2.3 A miniature porous aluminum oxide-based flowcell for online water quality monitoring using bacterial sensor cells

**Authors: S. Yagur-Kroll, E. Schreuder, C. J. Ingham, R. Heideman, R. Rosen, and S. Belkin**  **Abstract:** The use of live bacterial reporters as sensing entities in whole-cell biosensors allows the investigation of the biological effects of a tested sample, as well as the bioavailability of its components. Here we present a proof of concept for a new design for online continuous water monitoring flow-cell biosensor, incorporating recombinant reporter bacteria, engineered to generate an optical signal (fluorescent or bioluminescent) in the presence of the target compound(s). At the heart of the flow-cell is a disposable chip made of porous aluminum oxide (PAO), which retains the sensor microorganisms on its rigid planar surface, while its high porosity allows an undisturbed access both to the sample and to essential nutrients. The ability of the bacterial reporters to detect model toxic chemicals was first demonstrated using a "naked" PAO chip placed on solid agar, and later in a chip encased in a specially designed flow-through configuration which enables continuous on-line monitoring. The applicability of the PAO chip to simultaneous online detection of diverse groups of chemicals was demonstrated by the incorporation of a 6-member sensor array into the flow-through chip. The selective response of the array was also confirmed in spiked municipal wastewater effluents. Sensing activity was retained by the bacteria after 12-weeks storage of freeze-dried biochips, demonstrating the biochip potential as a simple minimal maintenance "plug-in" cartridge. This low-cost and easy to handle PAO-based flow-cell biosensor may serve as a basis for a future platform for water quality monitoring.

# 2.4 The use of a Neural Network technique for the prediction of water quality parameters

**Authors: H. R. Maier and G. C. Dandy Abstract:** This paper is concerned with the use of Neural Network models for the prediction of water quality parameters in rivers. The procedure that should be followed in the development of such models is outlined. Artificial Neural Networks (ANNs) were developed for the prediction of the monthly values of three water quality parameters of the Strymon river at a station located in Sidirokastro Bridge near the Greek — Bulgarian borders by using the monthly values of the other existing water quality parameters as input variables. The monthly data of thirteen parameters and the discharge, at the Sidirokastro station, for the time period 1980–1990 were selected for this analysis. The results demonstrate the ability of the appropriate ANN models for the prediction of water quality parameters. This provides a very useful tool for filling the missing values that is a very serious problem in most of the Greek monitoring stations.

**2.5 SOFTWARE ENVIRONMENT**

**Python** is a high-level, interpreted scripting language developed in the late 1980s by Guido van Rossum at the National Research Institute for Mathematics and Computer Science in the Netherlands. The initial version was published at the alt. Sources [newsgroup](https://en.wikipedia.org/wiki/Usenet) in 1991, and version 1.0 was released in 1994.

Python 2.0 was released in 2000, and the 2.x versions were the prevalent releases until December 2008. At that time, the development team made the decision to release version 3.0, which contained a few relatively small but significant changes that were not backward compatible with the 2.x versions. Python 2 and 3 are very similar, and some features of Python 3 have been back ported to Python 2. But in general, they remain not quite compatible.

Both Python 2 and 3 have continued to be maintained and developed, with periodic release updates for both. As of this writing, the most recent versions available are 2.7.15 and 3.6.5. However, an official [End of Life date of January 1, 2020](https://pythonclock.org/) has been established for Python 2, after which time it will no longer be maintained. If you are a newcomer to Python, it is recommended that you focus on Python 3, as this tutorial will do.

Python is still maintained by a core development team at the Institute, and Guido is still in charge, having been given the title of BDFL (Benevolent Dictator For Life) by the Python community. The name Python, by the way, derives not from the snake, but from the British comedy troupe [Monty Python’s Flying Circus](https://en.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus), of which Guido was, and presumably still is, a fan. It is common to find references to Monty Python sketches and movies scattered throughout the Python documentation.

**2.6 WHY CHOOSE PYTHON**

If you’re going to write programs, there are literally dozens of commonly used languages to choose from. Why choose Python? Here are some of the features that make Python an appealing choice.

**Python is Popular**

Python has been growing in popularity over the last few years. The 2018 [Stack Overflow Developer Survey](https://insights.stackoverflow.com/survey/2018) ranked Python as the 7th most popular and the number one most wanted technology of the year. [World-class software development countries around the globe use Python every single day.](https://realpython.com/world-class-companies-using-python/)

According to [research by Dice](https://insights.dice.com/2016/02/01/whats-hot-and-not-in-tech-skills/) Python is also one of the hottest skills to have and the most popular programming language in the world based on the [Popularity of Programming Language Index](https://pypl.github.io/PYPL.html).

Due to the popularity and widespread use of Python as a programming language, Python developers are sought after and paid well. If you’d like to dig deeper into [Python salary statistics and job opportunities, you can do so here](https://dbader.org/blog/why-learn-python).

**Python is interpreted**

Many languages are compiled, meaning the source code you create needs to be translated into machine code, the language of your computer’s processor, before it can be run. Programs written in an interpreted language are passed straight to an interpreter that runs them directly.

This makes for a quicker development cycle because you just type in your code and run it, without the intermediate compilation step.

One potential downside to interpreted languages is execution speed. Programs that are compiled into the native language of the computer processor tend to run more quickly than interpreted programs. For some applications that are particularly computationally intensive, like graphics processing or intense number crunching, this can be limiting.

In practice, however, for most programs, the difference in execution speed is measured in milliseconds, or seconds at most, and not appreciably noticeable to a human user. The expediency of coding in an interpreted language is typically worth it for most applications.

### Python is Free

The Python interpreter is developed under an OSI-approved open-source license, making it free to install, use, and distribute, even for commercial purposes.

A version of the interpreter is available for virtually any platform there is, including all flavors of Unix, Windows, macOS, smart phones and tablets, and probably anything else you ever heard of. A version even exists for the half dozen people remaining who use OS/2.

### Python is Portable

Because Python code is interpreted and not compiled into native machine instructions, code written for one platform will work on any other platform that has the Python interpreter installed. (This is true of any interpreted language, not just Python.)

### Python is Simple

As programming languages go, Python is relatively uncluttered, and the developers have deliberately kept it that way.

A rough estimate of the complexity of a language can be gleaned from the number of keywords or reserved words in the language. These are words that are reserved for special meaning by the compiler or interpreter because they designate specific built-in functionality of the language.

Python 3 has 33 keywords, and Python 2 has 31. By contrast, C++ has 62, Java has 53, and Visual Basic has more than 120, though these latter examples probably vary somewhat by implementation or dialect.

Python code has a simple and clean structure that is easy to learn and easy to read. In fact, as you will see, the language definition enforces code structure that is easy to read.

But It’s Not That Simple For all its syntactical simplicity, Python supports most constructs that would be expected in a very high-level language, including complex dynamic data types, structured and functional programming, and [object-oriented programming](https://realpython.com/python3-object-oriented-programming/).

Additionally, a very extensive library of classes and functions is available that provides capability well beyond what is built into the language, such as database manipulation or GUI programming.

Python accomplishes what many programming languages don’t: the language itself is simply designed, but it is very versatile in terms of what you can accomplish with it.

## Conclusion

This section gave an overview of the **Python** programming language, including:

* A brief history of the development of Python
* Some reasons why you might select Python as your language of choice

Python is a great option, whether you are a beginning programmer looking to learn the basics, an experienced programmer designing a large application, or anywhere in between. The basics of Python are easily grasped, and yet its capabilities are vast. Proceed to the next section to learn how to acquire and install Python on your computer.

**Python** is an [open source](https://simple.wikipedia.org/wiki/Open_source) [programming language](https://simple.wikipedia.org/wiki/Programming_language) that was made to be easy-to-read and powerful. A [Dutch](https://simple.wikipedia.org/wiki/Netherlands) programmer named [Guido van Rossum](https://simple.wikipedia.org/wiki/Guido_van_Rossum) made Python in 1991. He named it after the television show [Monty Python's Flying Circus](https://simple.wikipedia.org/wiki/Monty_Python%27s_Flying_Circus). Many Python examples and tutorials include jokes from the show.

Python is an interpreted language. Interpreted languages do not need to be [compiled](https://simple.wikipedia.org/wiki/Compiled_language) to run. A program called an [interpreter](https://simple.wikipedia.org/wiki/Interpreter_(computing)) runs Python code on almost any kind of computer. This means that a programmer can change the code and quickly see the results. This also means Python is slower than a compiled language like [C](https://simple.wikipedia.org/wiki/C_(programming_language)), because it is not running [machine code](https://simple.wikipedia.org/wiki/Machine_code) directly.

Python is a good programming language for beginners. It is a high-level language, which means a programmer can focus on what to do instead of how to do it. Writing programs in Python takes less time than in some other languages.

Python drew inspiration from other programming languages like C, [C++](https://simple.wikipedia.org/wiki/C%2B%2B), [Java](https://simple.wikipedia.org/wiki/Java_(programming_language)), [Perl](https://simple.wikipedia.org/wiki/Perl), and [Lisp](https://simple.wikipedia.org/wiki/LISP).

Python has a very easy-to-read syntax. Some of Python's syntax comes from C, because that is the language that Python was written in. But Python uses whitespace to delimit code: spaces or tabs are used to organize code into groups. This is different from C. In C, there is a [semicolon](https://simple.wikipedia.org/wiki/Semicolon) at the end of each line and curly braces ({}) are used to group code. Using whitespace to delimit code makes Python a very easy-to-read language.

**Python use [change / change source]**

Python is used by hundreds of thousands of programmers and is used in many

places. Sometimes only Python code is used for a program, but most of the time it is used to do simple jobs while another programming language is used to do more complicated tasks.

Its [standard library](https://simple.wikipedia.org/w/index.php?title=Standard_library&action=edit&redlink=1) is made up of many [functions](https://simple.wikipedia.org/wiki/Computable_function) that come with Python when it is installed. On the [Internet](https://simple.wikipedia.org/wiki/Internet) there are many other [libraries](https://simple.wikipedia.org/w/index.php?title=Library_(computing)&action=edit&redlink=1) available that make it possible for the Python language to do more things. These libraries make it a powerful language; it can do many different things.

Some things that Python is often used for are:

* Web development
* Scientific programming
* Desktop [GUIs](https://simple.wikipedia.org/wiki/GUI)
* Network programming
* [Game](https://simple.wikipedia.org/wiki/Video_game) programming

**3. SYSTEM ANALYSIS**

**3.1 EXISTING SYSTEM:**

Existing ANN and random forest will not have above dataset processing steps so its error rate will be high compare to propose adaptive frequency analysis algorithm.

In propose paper author has used Norwegian country water supply dataset but he did not publish that dataset on internet so we don’t have that dataset but we found Indian state water supply quality dataset.

**3.2 PROPOSED SYSTEM:**

In this project as extension we have added CNN (convolution neural network) and LSTM (long short term memory) and compare RMSE (root mean square error) with existing algorithms such as Random Forest, ANN and Adaptive Frequency. All existing algorithms will not filtered dataset multiple times to extract important features which helps in getting better prediction accuracy and reduce error rate. CNN and LSTM are the two most preferable deep learning algorithms which filter dataset multiple times to extract important features from dataset and then train a prediction model and all irrelevant features will be removed out by using DROPOUT functions and dataset will be filtered using function called DENSE which filtered dataset by using specified number of neurons.

More data processing organizations are switching towards CNN and LSTM classification or prediction model due to its increasing performance and popularity.

In CNN and LSTM we will define number of input and output layers and each layer will take number of data filtration as input. In below code screen you can read red colour comments to understand CNN implementation.

**4. FEASIBILITY STUDY**

The feasibility of the project is analyzed in this phase and 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 to 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

**4.1 ECONOMICAL FEASIBILITY**

This study is carried out to check the economic impact that the system will have on the organization. The amount of fund that the company can pour into the research and development of the system is limited. The expenditures must be justified. Thus the developed system 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.

**4.2 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 on the available technical resources. This will lead to high demands on the 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.

**4.3 SOCIAL FEASIBILITY**

The aspect of 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.

**5. SYSTEM REQUIREMENTS**

**5.1 HARDWARE REQUIREMENTS:**

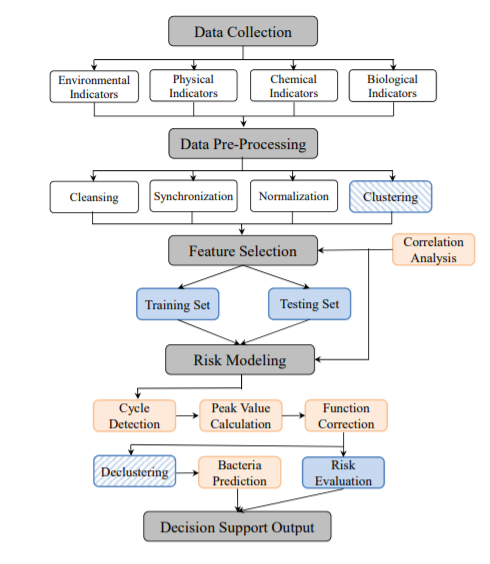
* System : Pentium Dual Core.
* Hard Disk : 120 GB.
* Monitor : 15’’ LED
* Input Devices : Keyboard, Mouse
* Ram : 1 GB

**5.2 SOFTWARE REQUIREMENTS:**

* Operating system : Windows 10
* Coding Language : python
* Tool : PyCharm
* Database : MYSQL
* Server : Flask

**6. SYSTEM DESIGN**

**6.1 SYSTEM ARCHITECTURE:**

****

**6.2 DATA FLOW DIAGRAM:**

1. The DFD is also called as bubble chart. It is a simple graphical formalism that can be used to represent a system in terms of input data to the system, various processing carried out on this data, and the output data is generated by this system.
2. The data flow diagram (DFD) is one of the most important modeling tools. It is used to model the system components. These components are the system process, the data used by the process, an external entity that interacts with the system and the information flows in the system.
3. DFD shows how the information moves through the system and how it is modified by a series of transformations. It is a graphical technique that depicts information flow and the transformations that are applied as data moves from input to output.
4. DFD is also known as bubble chart. A DFD may be used to represent a system at any level of abstraction. DFD may be partitioned into levels that represent increasing information flow and functional detail.

**User**

**Check**

**Unauthorized user**

**Yes NO**

**Upload Water Dataset**

**Preprocess& Normalized Dataset**

**Feature Selection**

**Run ANN Algorithm**

**Run Random Forest Algorithm**

**Run Propose Adaptive Frequency Analysis Algorithm**

**Run CNN Algorithm**

**Run LSTM Algorithm**

**RMSE Comparison Graph**

**Predict Water Quality & Risk**

**End process**

**6.3 UML DIAGRAMS:**

UML stands for Unified Modeling Language. UML is a standardized general-purpose modeling 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, Visualization, Constructing and documenting the artifacts of software system, as well as for business modeling and other non-software systems.

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

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

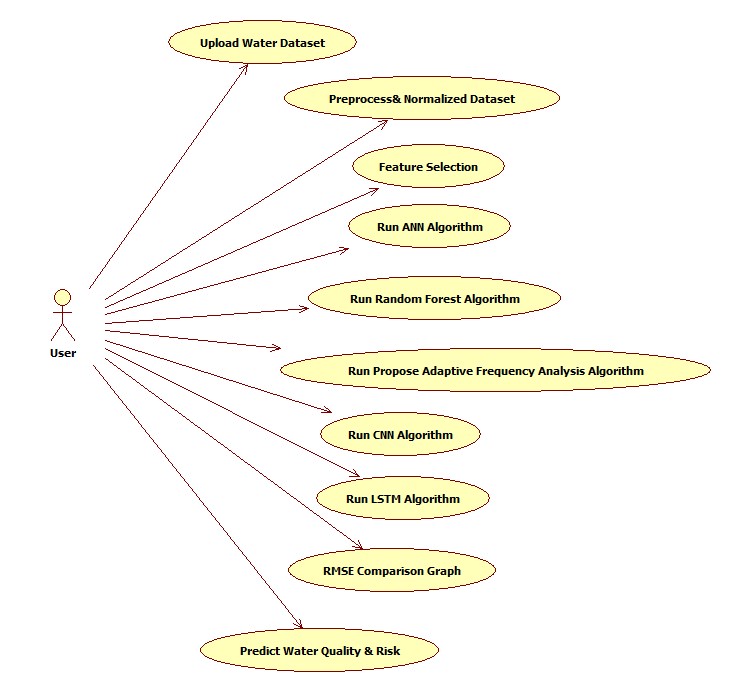
**GOALS:**

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

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

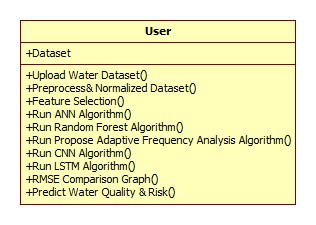
**USE CASE DIAGRAM:**

A use case diagram in the Unified Modeling Language (UML) is a type of behavioral 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. Roles of the actors in the system can bedepicted.



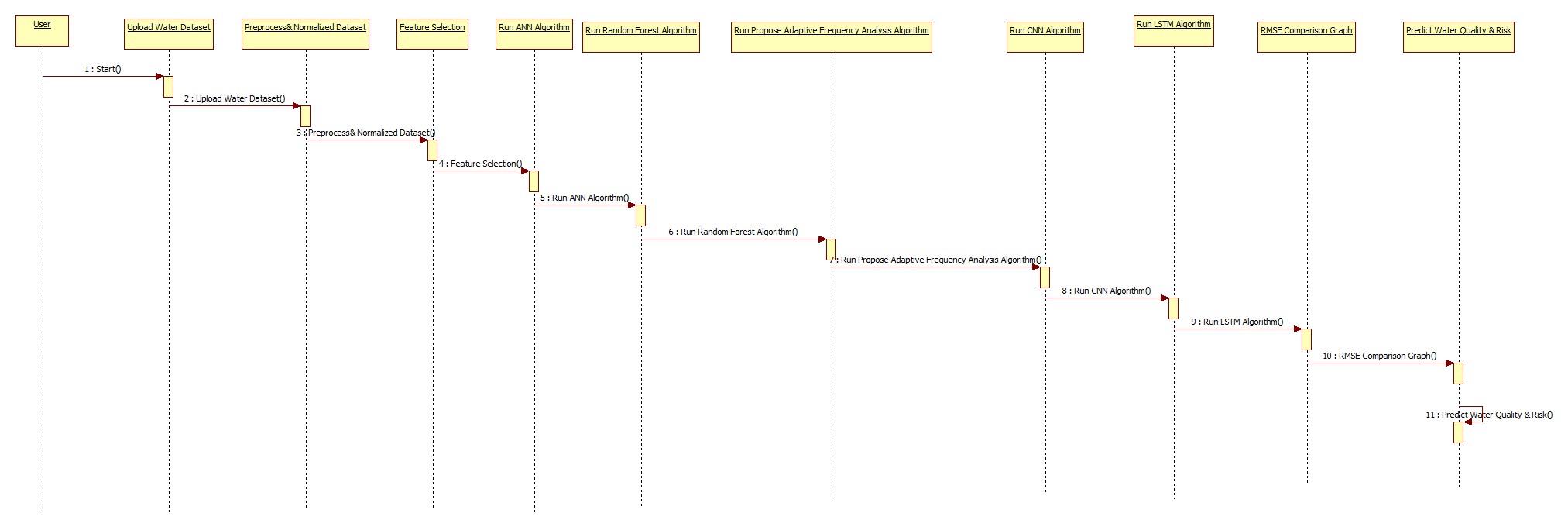
**CLASS DIAGRAM:**

In software engineering, a class diagram in the Unified Modeling Language (UML) is a type of static structure diagram that describes the structure of a system by showing the system's classes, their attributes, operations (or methods), and the relationships among the classes. It explains which class contains information.



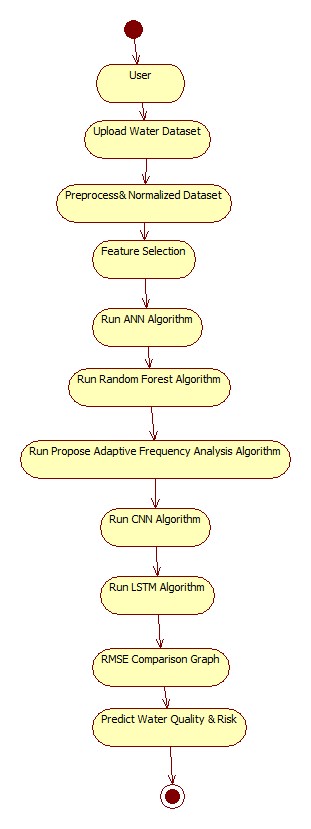
**SEQUENCE DIAGRAM:**

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.



**ACTIVITY DIAGRAM:**

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control.



**7. IMPLEMENTATION**

**MODULES DESCRIPTION:**

1. Physical data. Drinking water has to verify physicalattributes in water quality for thewhole supply process.
2. Chemical data. Chemical indicators are the traditionalrepresentation of water quality. They provide informationonwhat is impacting on the system aswell.
3. Biological data. Biological indicators are direct measuresof the health of the fauna and flora in the watersupply.
4. Environmental data. Environment data can be a leadingimpact factor forwater quality in some places.
5. Data Preprocessing: In this module missing and irrelevant data will be removed from dataset as data obtained from sensor contains huge size of data with noise values so by proprocessing we can remove such noise data.
6. Normalization: using this module we can normalize data between 0 and 1 to allow machine learning model to make better prediction
7. Clustering: Using this module we will cluster all dataset and this cluster will separate risk water quality data into one cluster and clean quality data into other cluster.
8. Synchronize: collect only recent data for evaluation

After processing dataset using above points then we will split dataset into train and test and then decluster data to predict risk value. In dataset if 100ML water contains any value of ECOLI bacteria then that water is not safe.

**7.2 SAMPLE CODE**

### 8. 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 in an unacceptable manner. There are various types of test. 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 a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic tests at 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 actually 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 satisfaction, 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 test:**

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 centered on the following items:

Valid Input : identified classes of valid input must be 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 exercised.

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

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify 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 Test:**

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.

**White Box Testing:**

White Box Testing is a testing in which in which the software tester has knowledge of the inner workings, structure and language of the software, or at least its purpose. It is 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, as 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 a 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.

**8.1 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.

# 8.2 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 encountered.

**8.3 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 encountered.

**9. INPUT DESIGN AND OUTPUT DESIGN**

**9.1 INPUT DESIGN:**

The input design is the link between the information system and the user. It comprises the developing specification and procedures for data preparation and those steps are necessary to put transaction data in to a usable form for processingcan be achieved by inspecting the computer to read data from a written or printed document or it can occur by having people keying the data directly into the system. The design of input focuses on controlling the amount of input required, controlling the errors, avoiding delay, avoiding extra steps and keeping the process simple. The input is designed in such a way so that it provides security and ease of use with retaining the privacy. Input Design considered the following things:

* What data should be given as input?
* How the data should be arranged or coded?
* The dialog to guide the operating personnel in providing input.
* Methods for preparing input validations and steps to follow when error occur.

**OBJECTIVES:**

1. Input Design is the process of converting a user-oriented description of the input into a computer-based system. This design is important to avoid errors in the data input process and show the correct direction to the management for getting correct information from the computerized system.

2.It is achieved by creating user-friendly screens for the data entry to handle large volume of data. The goal of designing input is to make data entry easier and to be free from errors. The data entry screen is designed in such a way that all the data manipulates can be performed. It also provides record viewing facilities.

3. When the data is entered it will check for its validity. Data can be entered with the help of screens. Appropriate messages are provided as when needed so that the user will not be in maize of instant. Thus the objective of input design is to create an input layout that is easy to follow

**9.2 OUTPUT DESIGN:**

A quality output is one, which meets the requirements of the end user and presents the information clearly. In any system results of processing are communicated to the users and to other system through outputs. In output design it is determined how the information is to be displaced for immediate need and also the hard copy output. It is the most important and direct source information to the user. Efficient and intelligent output design improves the system’s relationship to help user decision-making.

1. Designing computer output should proceed in an organized, well thought out manner; the right output must be developed while ensuring that each output element is designed so that people will find the system can use easily and effectively. When analysis design computer output, they should Identify the specific output that is needed to meet the requirements.

2. Select methods for presenting information.

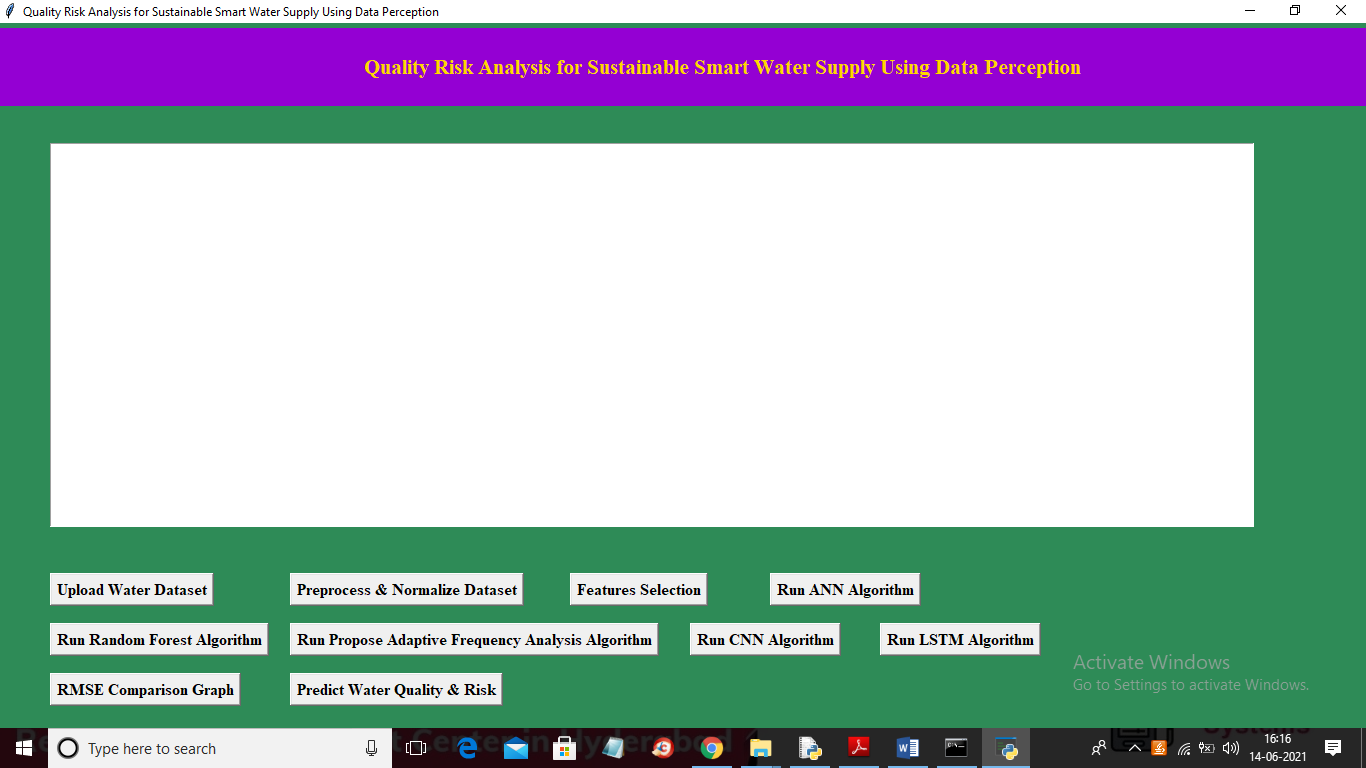
3. Create document, report, or other formats that contain information produced by the system.

The output form of an information system should accomplish one or more of the following objectives.

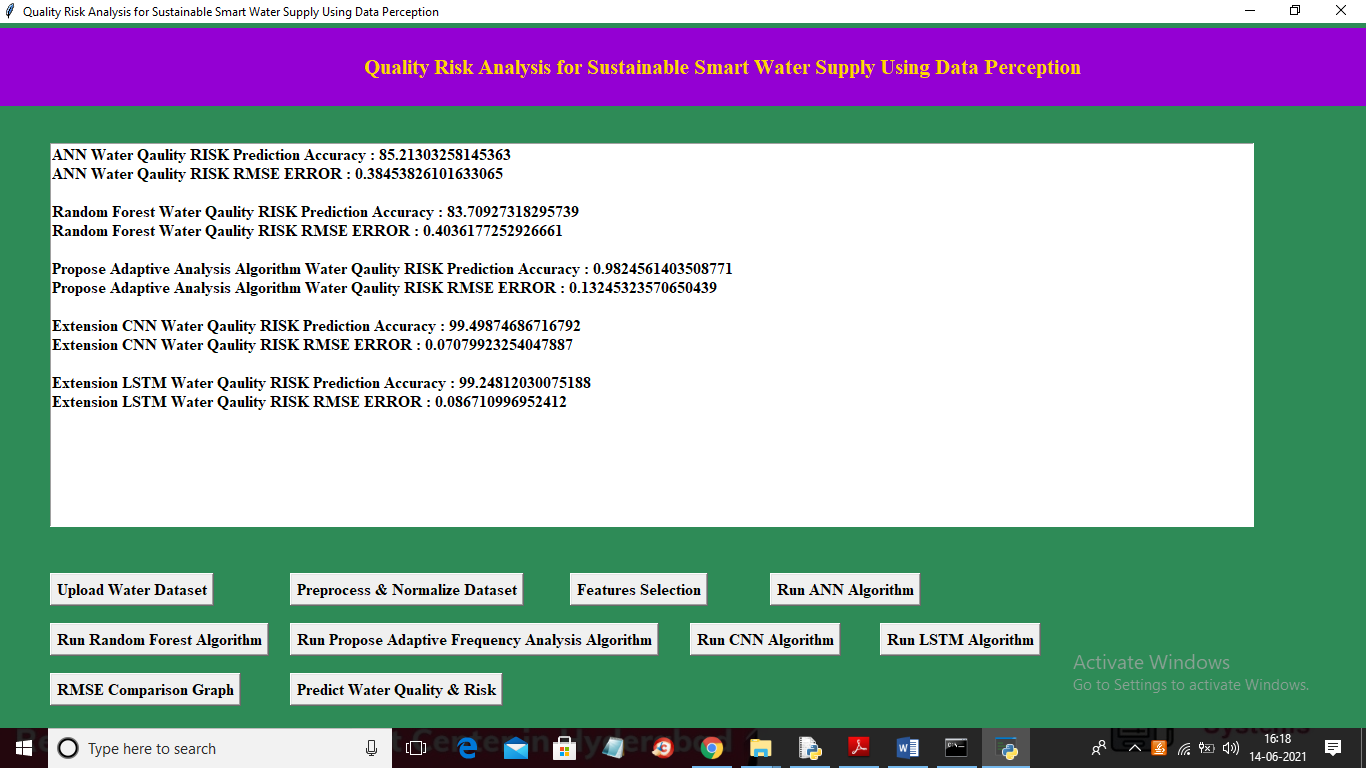
* Convey information about past activities, current status or projections of the
* Future.
* Signal important events, opportunities, problems, or warnings.
* Trigger an action.
* Confirm an action.

**10. SCREENSHOTS**

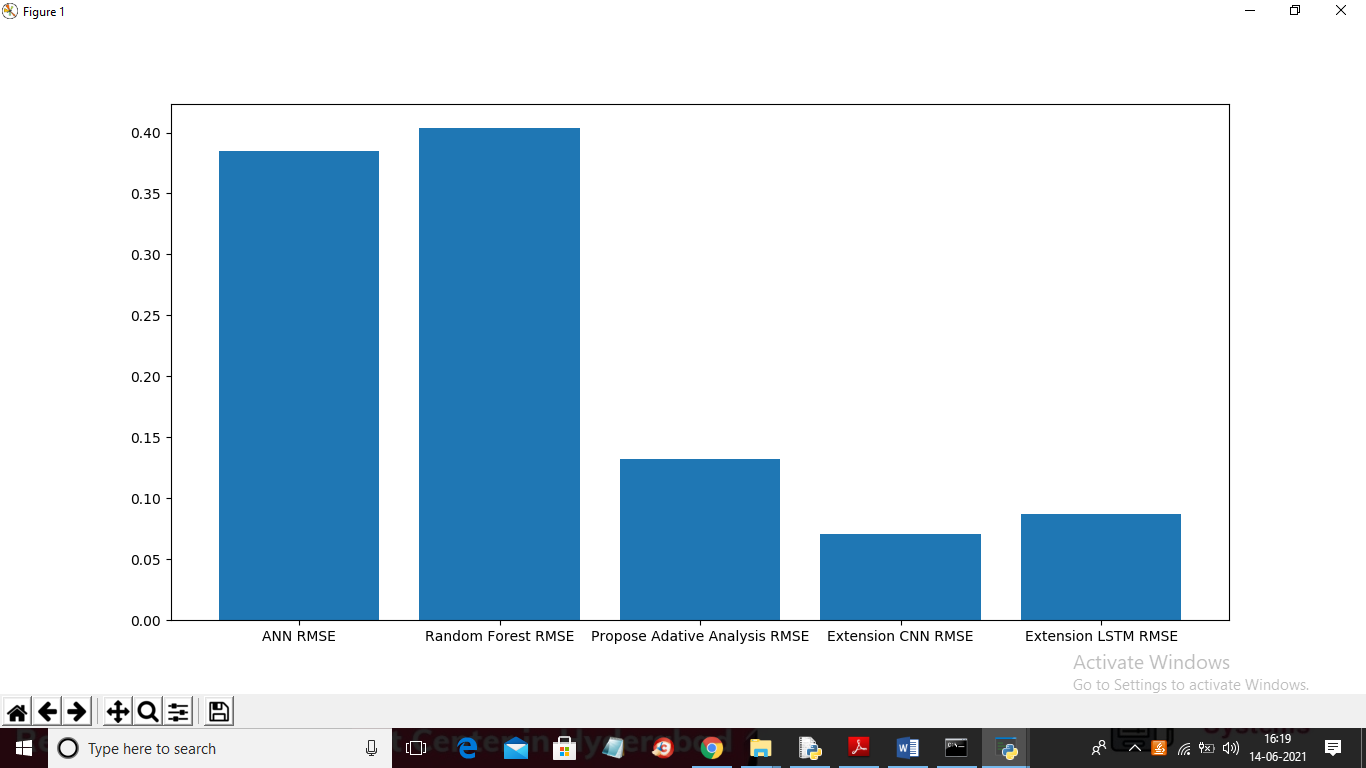
To run project double click on ‘run.bat’ file to get below screen



In above screen you can see we have added two more buttons to build CNN and LSTM model and like previous output upload dataset and then run all buttons one by one to train model of each algorithms and then compare accuracy and RMSE error



In above screen we trained all algorithms on same dataset and from above output we can see CNN has got more accuracy and less error rate compare to all other algorithms and below is then RMSE comparison graph for all algorithms



In above graph we can see CNN got less error compare to other algorithms and in machine learning model with less error rate and more accuracy can be consider as best prediction model

**11. FUTURE ENHANCEMENT**

Furthermore, we have to decluster the results and predict accurate bacteria indicators, both in tendency and values. These values can map to different risk modes according to practical water source management standards in different countries and regions. Future decision support in water treatment plants can adjust to both prediction and risk mode. Also, in practice, the models need to be evolved with both domain knowledge data set growing.

**12. CONCLUSION**

Water quality is a very critical issue in modern urban life all around the world, especially for Smart Water Supply system development. Traditional monitoring and risk control methods are difficult to detect bacteria broadcast on time and provide efficient decision support. In this paper, we propose an approach for water quality risk early warning using data perception. With the application among four different cities in Norway, we have proved the feasibility, accuracy, and efficiency of our approach. The preliminary results evaluated by domain experts are very promising. This work is beneficial in generally three aspects: • It provides an early warning mechanism from the water source areas using cost-less data analysis techniques. This prolongs the preventive measures response time, and support more decision options in the latter steps of water supply. • This approach integrates indicator, geography and time domains. It provides a new frequency domain analysis perspective to find the relationship between different indicators and their predictions. At the same time, it embraces scalability for these three domains. • This work is applied to real industrial water supply systems from 4 different Norwegian cities.

**13. BIBLIOGRAPHY**

[1] S. Franco, V. Gaetano, and T. Gianni, “Urbanization and climate change impacts on surface water quality: Enhancing the resilience by reducing impervious surfaces,” Water Research, vol. 144, pp. 491–502, 2018.

[2] T. Hak, S. Janou ´ skov ˇ a, and B. Moldan, “Sustainable development ´ goals: A need for relevant indicators,” Ecological Indicators, vol. 60, pp. 565–573, 2016.

[3] World Health Organization (WHO), Guidelines for drinking-water quality: recommendations. World Health Organization, 2004.

[4] E. Weinthal, Y. Parag, A. Vengosh, A. Muti, and W. Kloppmann, “The eu drinking water directive: the boron standard and scientific uncertainty,” European Environment, vol. 15, no. 1, pp. 1–12, 2005.

[5] R. W. Adler, J. C. Landman, and D. M. Cameron, The clean water act 20 years later. Island Press, 1993.

[6] D. Berge, “Overvaking av farrisvannet med tilløp fra 1958-2010,” ˚ 2011.

[7] I. W. Andersen, “EUs rammedirektiv for vann– miljøkvalitetsnormer for vannmiljøet i møte med norsk rett,” Kart og Plan, vol. 73, no. 5, pp. 355–366, 2013.

[8] V. Novotny, Water quality: prevention, identification and management of diffuse pollution. Van Nostrand-Reinhold Publishers, 1994.

[9] A. Hounslow, Water quality data: analysis and interpretation. CRC press, 2018.

[10] S. Yagur-Kroll, E. Schreuder, C. J. Ingham, R. Heideman, R. Rosen, and S. Belkin, “A miniature porous aluminum oxide-based flowcell for online water quality monitoring using bacterial sensor cells,” Biosensors and Bioelectronics, vol. 64, pp. 625–632, 2015.

[11] H. R. Maier and G. C. Dandy, “The use of artificial neural networks for the prediction of water quality parameters,” Water Resources Research, vol. 32, no. 4, pp. 1013–1022, 1996.

[12] H. Orouji, O. Bozorg Haddad, E. Fallah-Mehdipour, and M. Marino, “Modeling of water quality parameters using data- ˜ driven models,” Journal of Environmental Engineering, vol. 139, no. 7, pp. 947–957, 2013.

[13] O. Bozorg-Haddad, S. Soleimani, and H. A. Loaiciga, “Modeling ´ water-quality parameters using genetic algorithm–least squares support vector regression and genetic programming,” Journal of Environmental Engineering, vol. 143, no. 7, p. 04017021, 2017.

[14] N. Mahmoudi, H. Orouji, and E. Fallah-Mehdipour, “Integration of shuffled frog leaping algorithm and support vector regression for prediction of water quality parameters,” Water Resources Management, vol. 30, no. 7, pp. 2195–2211, 2016.

[15] F.-J. Chang, Y.-H. Tsai, P.-A. Chen, A. Coynel, and G. Vachaud, “Modeling water quality in an urban river using hydrological factors–data driven approaches,” Journal of Environmental Management, vol. 151, pp. 87–96, 2015.

[16] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things (iot): A vision, architectural elements, and future directions,” Future generation computer systems, vol. 29, no. 7, pp. 1645– 1660, 2013.

[17] V. Mayer-Schonberger and K. Cukier, ¨ Big data: A revolution that will transform how we live, work, and think. Houghton Mifflin Harcourt, 2013. [18] Y. Wu, F. Hu, G. Min, and A. Y. Zomaya, Big Data and Computational Intelligence in Networking. CRC Press, 2017.

[19] D. R. Hardoon, S. Szedmak, and J. Shawe-Taylor, “Canonical correlation analysis: An overview with application to learning methods,” Neural computation, vol. 16, no. 12, pp. 2639–2664, 2004.

[20] I. T. Jolliffe and J. Cadima, “Principal component analysis: a review and recent developments,” Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences, vol. 374, no. 2065, p. 20150202, 2016.