People's Democratic Republic of Algeria
Ministry of High Education and Scientific Research
University of Akli Mohand Oulhadj Bouira
Faculty of Science and Applied Science
Computer Science department

Master Thesis

In Computer Science

***Specialty**: Computer Systems Engineering*

Theme

Intelligent Approaches for IoT: Water Quality Prediction

Supervised by:

- Dr.Akli Abbas

Realized by:

- Rihab Mecheri
- Smail Dahmani

2021/2022

Acknowledgments

First and foremost we thank **Allah**, the Almighty, for providing us with the health, courage, and patience to complete this project.

We would like to express our gratitude and appreciation to our supervisor Dr. Akli Abbas for providing guidance and feedback throughout this project.

We would also like to thank the members of the jury who accepted to evaluate our work. As well as all the professors, who ensured our path in university course.

Thanks to our families who supported and encouraged us throughout our studies.

Finally, special thanks to all our friends and everyone contributed in this work by encouraging and supporting us.

Dedication

I dedicate this work to my family, my friends and anyone who supported me.

Rihab Mecheri.

Dedication

This work is dedicated to:

My parents, siblings and to all my friends who have provided me with their encouragement, love and their support.

Smail Dahmani.

Abstract

The amount of data generated by the Internet of Things(IoT) is very large. Managing and analyzing all this data is a major challenge. Artificial Intelligence (AI) can do it faster and with greater precision. Thus, AI, and particularly Machine Learning(ML), is an effective ally for processing a growing volume of data. The objective of this work is to propose an intelligent approach for the IoT, in this case we have chosen to work on one of the applications where we can use the IoT, which is water quality prediction . We used three supervised learning algorithms K-Nearest Neighbor (KNN), Decision Tree (DT), and Random Forest (RF) on a database to develop such an approach.
The RF algorithm was more efficient than KNN, and DT, as we got the highest accuracy **(90%)** with the RF algorithm.

Key words: Internet of Things, Artificial Intelligence, Machine Learning, water quality prediction, K-Nearest Neighbor, Decision Tree, Random Forest.

Résumé

La quantité de données générée par l'Internet des Objets est très grande. La prise en charge et l'analyse de toutes ces données constitue un défi majeur. L'Intelligence Artificielle peut le faire plus rapidement et avec une plus grande précision. Ainsi, L'Intelligence Artificielle, et particulièrement l'apprentissage automatique, se trouve être un allié efficace pour traiter un volume de données croissant. L'objectif de ce travail est de proposer une approche intelligente pour l'Internet des Objets, dans ce cas nous avons choisi de travailler sur l'une des applications où nous pouvons utiliser l'Internet des Objets, qui est la prédiction de la qualité de l'eau. Nous avons utilisé l'algorithme d'apprentissage supervisé Random Forest (RF) sur une base de données pour développer une telle approche.

Mots clés: Internet des Objets, Intelligence Artificielle, apprentissage automatique, prédiction de la qualité de l'eau, K-plus proche voisin, arbre de décision, Random Forest.

ملخص

كمية البيانات الناتجة عن إنترنت الأشياء كبيرة جدًا. تعد إدارة وتحليل كل هذه البيانات تحديًا كبيرًا. يمكن للذكاء الاصطناعي القيام بذلك بشكل أسرع وبدقة أكبر. وبالتالي ، يعد الذكاء الاصطناعي ، وخاصة التعلم الآلي ، حليفًا فعالًا لمعالجة حجم متزايد من البيانات. الهدف من هذا العمل هو اقتراح نهج ذكي لإنترنت الأشياء ، وفي هذه الحالة اخترنا العمل على أحد التطبيقات حيث يمكننا استخدام إنترنت الأشياء ، وهو التنبؤ بجودة المياه. استخدمنا خوارزمية التعلم الخاضع للإشراف الغابة العشوائية في قاعدة بيانات لتطوير مثل هذا النهج.

الكلمات المفتاحية: إنترنت الأشياء ، الذكاء الاصطناعي ، التعلم الآلي ، التنبؤ بنوعية المياه ، الجار الأقرب لـ ك ، شجرة القرار ، الغابة العشوائية.

Contents

List of Figures

List of Tables

Abbreviations list

IoT	Internet of Things
RFID	Radio Frequency Identification
WQM	Water Quality Monitoring
WSN	Wireless Sensor Network
AI	Artificial Intelligence
ML	Machine Learning
KNN	K-Nearest Neighbour
DT	Decision Tree
RF	Random Forest
DL	Deep Learning
DBN	Deep Belief Networks
RBM	Restricted Boltzmann Machines
CNN	Convolutional Neural Network
RNN	Recurrent Neural Networks
WHO	World Health Organization
EPA	Environmental Protection Agency
USA	United States of America

General introduction

The Internet of Things (IoT) is a collection of many interconnected objects and devices that can communicate, share data, and information with the help of the internet from anywhere and at any time to achieve a common goal in various areas and applications. Sensors deployed in the IoT generates huge volumes of data for a wide range of applications such as smart health, smart cities, smart living, smart environment monitoring, and so on.

In addition to their large volume, these data are very different and uncertain. The analysis of this data by humans or classical analysis methods is difficult.

In order to take advantage and make sense of this data, one of the challenges facing the industry (or individuals) is to migrate traditional data analysis models to modern and intelligent analysis models where the concept smart data is a good representative of IoT data. These models, based on Machine Learning techniques and Artificial Intelligence. Artificial Intelligence is playing a starring role in IoT because of its ability to quickly extract insights from data. Machine Learning, an Artificial Intelligence branche, brings the ability to automatically identify patterns and detect anomalies in the data that smart sensors and devices generate information such as temperature, pressure, humidity, air quality, water quality, etc.

The main objective of this work is to propose an approach based on Machine Learning techniques to analyze data and make decisions.

Our work is divided into four chapters:

- **Chapter 1:** Internet of Things

  In this chapter, we will discuss about what IoT is, the key technology of The IoT, about its architecture, characteristics applications. And we will talk about sensors and its types.

- **Chapter 2:** Artificial Intelligence

  In this chapter, we will talk about some basics like Artificial Intelligence passing after that to Machine Learning, and some of its types, Finale we will have a close look on Deep Learning.

- **Chapter 3:** Water Quality Prediction With KNN, DT and RF

  In this chapter, we will describe the proposed approach.

- **Chapter 4:** Implementation and Evaluation

  In this chapter, we will discuss the results obtained.

Finally we will conclude our work with a general conclusion.

Chapter 1

Internet of Things

1.1 Introduction

Physical electronic devices that can communicate with each other over the Internet to exchange useful information are called the Internet of Things. IoT is more than just electronic devices connected to the internet. The Internet of Things is the latest technology to form a system capable of capturing and responding to the movement of information without human interference.

This chapter provides an overview of the basic concepts including the definition of the Internet of Things, its properties, architecture and applications. We also, introduce the importance of the Internet of Things in Water Quality.

1.2 Internet of Things

1.2.1 Definition

The Internet of Things (IoT) is the network of physical devices, vehicles, buildings and so on embedded with electronics, software, sensors and network connectivity that enable these objects to collect and transmit data via internet [1].

Figure 1.1: Internet of Things [1].

1.2.2 Characteristics

The fundamental characteristics of the IoT are as follows [12].

- **Interconnectivity:** Regarding IoT, everything can be connected to the global information and communication infrastructure.
- **Things-related services:** The IoT is capable of providing thing-related services within the constraints of things, such as privacy protection and semantic consistency between physical things and their associated virtual things. In order to provide thing-related services within the constraints of things.
- **Heterogeneity:** Devices in IoT are heterogeneous because they are based on different hardware platforms and networks. They can interact with other devices or service platforms through different networks.
- **Dynamic changes:** Device state changes dynamically, such as sleep and wake, connect and/or disconnect, and the context of the device, including location and speed. Furthermore, the number of devices can change dynamically.
- **Enormous scale:** The number of devices that need to manage and communicate with each other will be at least an order of magnitude greater than the current number of Internet-connected devices.

1.2.3 IoT Architecture

IoT architecture consists of different layers of technologies supporting IoT. It serves to illustrate how various technologies relate to each other and to communicate the scalability, modularity and configuration of IoT deployments in different scenarios.

Each layer is defined by its functions and the devices that are used in that layer. There are different opinions regarding the number of layers in IoT. However, according to many researchers, the IoT mainly operates on three layers termed as Perception, Network, and Application layers.
Figure 1.2 shows the basic three layer architectural framework of IoT with respect to the devices and technologies that encompass each layer [1] [2].

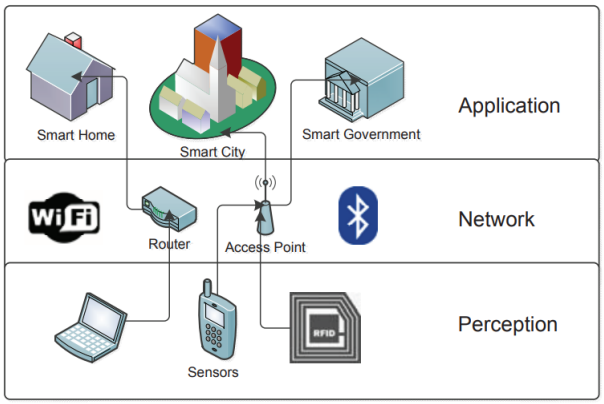

Figure 1.2: Three-layer IoT architecture [2].

- **Perception Layer:** The bottom layer consists of smart objects integrated with sensors. Sensors enable the physical and digital worlds to be connected so that real-time information can be collected and processed. There are different types of sensors used for different purposes. Depending on the type of sensor, the information can be location, temperature, orientation, motion, vibration, acceleration, humidity, pressure, flow, and chemical changes in the air, among others. The collected information is then forwarded to the network layer for secure transmission to the information processing system.

- **Network layer:** The network layer of IoT is used to route and transmit data to various IoT centers and devices over the Internet. At this level, cloud computing platforms, Internet gateways, switching and routing devices, etc. Use some of the

latest technologies, such as WiFi, LTE, Bluetooth, 3G, Zigbee, etc. Network gateways act as an intermediary between different IoT nodes by aggregating, filtering and transmitting data to and from different sensors.

- **Application Layer:** IoT applications include "smart" environments/spaces in the following areas : transportation, buildings, cities, lifestyle, retail, agriculture, factories, supply chain, emergency response, healthcare, user interaction, culture and tourism, environment and energy . The application layer guarantees the authenticity, integrity and confidentiality of the data.

1.2.4 The Key Technology of The Internet of Things

The key technologies of the IoT include RFID technology, sensor technology, network communication technology, embedded system technology, etc [13].

1. **Radio Frequency Identification (RFID) Technology**

   RFID technology is a comprehensive integration technology that combines wireless radio frequency technology with embedded technology as a whole, and it has a wide range of applications in the entire network system. In general, an RFID system consists of an RFID electronic label, a reader-writer, and an information processing system. When the matter with the electronic label passes through a specific information reader-writer, the label is activated by the reader-writer and the information carried in the label is transmitted through radio waves to the reader-writer and the information processing system, completing the work of automatic information acquisition. The information processing system is responsible for information control and processing.

2. **Sensor Technology**

   The Sensor is responsible for collecting network information. Sensors are usually composed of sensitive elements and conversion elements, which can sense sound, light, electricity, heat, force, displacement, moisture through signal perception, and provide raw information for the Internet of Things, work detection, analysis and feedback.

   Sensor network technology integrates sensor technology, embedded computing technology, modern network and wireless communication technology, distributed in-

formation processing technology and so on, the micro sensor collaboration of the integrated real-time monitoring, sensing and collecting a variety of environmental or monitoring information. A typical sensor network structure usually consists of sensor nodes, sink, internet or communication satellite, task management node etc.

3. **Network Communication Technology**

   No matter how far the concept of IoT goes, perception and thing-to-thing communication are irreplaceable key technologies. The sensor network communication technology includes short-distance communication technology and long-distance network communication technology. Commonly used network communication technologies for sensors include Bluetooth, RFID, ZigBee, IrDA, Wi-Fi, UWB, NFC, Wireless Hart, etc.

4. **Embedded System Technology**

   Embedded system technology is the combination of computer hardware and software, sensor technology, integrated circuit technology and electronic technology applications. After decades of development, embedded system technology intelligent terminal products can be seen everywhere, mobile phones, automotive instruments, robots, medical equipment, set-top boxes, aerospace equipment and industrial control systems are all embedded systems.

1.2.5 Advantages of IoT

- **Minimize human efforts:** As devices interact and communicate with each other and accomplish many tasks for us, human effort is reduced.
- **Save time:** IoT saves us time by reducing manpower. Instead of repeating the same task every day, it allows people to do other creative tasks.
- **Efficient resources utilization:** If we know the functionality and the way that how each device works we can increase the optimum utilization of energy and resources. Here we can save money by using IoT technology.
- **Improving quality of life:** As IoT (technology) increased confort, convenience and better management, hence it improves the quality of life.

- **Better monitoring of devices:** The IoT allows us to automate and control the tasks that are done on a daily basis and reducing human intervention, we can monitor the devices connected to IoT and take necessary action in case of emergencies.
- **Enhance data collection:** It helps collecting data information.
- Ability to access information from anywhere at anytime on any device.

1.2.6 Limitations and Security Issues in IoT

Security in IoT is a double-edged sword. Now we have an interconnected system devices on the network, it makes the system quite secure and efficient. Although connected devices generate huge amounts of data and constantly exchange between them device, so it in turn generates a lot of internet traffic. Devices can make these data useful, but with this accessibility of the internet comes security and privacy question. There are many reasons for these problems.

Because IoT devices cannot be patched or fixed, bugs in software are often found. This makes them vulnerable to security risks like DDoS attacks, one of the most common problems caused by setting a device password as a default password that can be easily cracked by hackers. Both ransomware and malware rely on encryption to completely lock down a user's device and steal their data. Privacy remains one of the biggest concerns for IoT as data is transferred, stored, processed and bullied by large corporations. This data is then sold to various other companies, violating our data protection and data security rights. Home intrusion, one of the most terrifying threats IoT can have, as most homes and offices rely on automated processes to function for deploying IoT devices in these places. The IP addresses of these devices are exposed to hackers, who can use it to track location and access our personal information, which is then sold to underground sites and can be used for criminal activities [14].

1.2.7 Application Areas of IoT

The IoT can find its applications in almost every aspect of our daily life. Here we discuss a few of them [12].

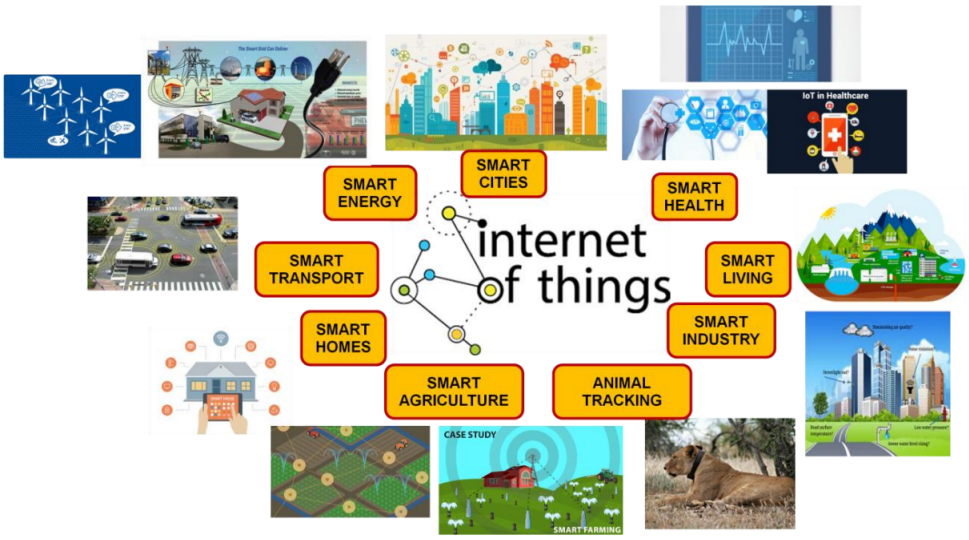

Figure 1.3: IoT Applications [1].

1. **Smart Health**

   - **Patients Surveillance:** Monitor the condition of patients in hospitals and nursing homes.
   - **Fall Detection:** Assistance for elderly or disabled people living independently.
   - **Medical Fridges:** Control the conditions of refrigerators that store vaccines, medicines and organic elements.
   - **Sleep Control:** Wireless sensors placed above the mattress capture small movements such as breathing and heart rate, as well as large movements caused by tossing and turning during sleep, with data available through an app on the smartphone.

2. **Smart Living**

   - **Intelligent Shopping Applications:** Provide recommendations at the point of sale based on customer habits, preferences, presence of allergic components for them, or expiring dates.
   - **Smart Home Appliances:** Refrigerator with LCD screen showing what's inside, food about to expire, ingredients we need to buy and all the information available on a smartphone app. The washing machine that allows us to remotely monitor laundry, as well. The kitchen cooktop connects with a

smartphone app to remotely control the temperature and monitor the oven's self-cleaning function.

- **Weather Station:** Displays outdoor weather conditions such as humidity, temperature, air pressure, wind speed and rainfall, and transmits data over long distances.
- **Safety Monitoring:** Cameras and home alarm systems making people feel safe in their daily life at home.
- **Smart Jewelry:** Increased personal safety by wearing a piece of jewelry inserted with bluetooth enabled technology used in a way that a simple push establishes contact with our smartphone, which through an app will send alarms to selected people in our social circle with information that we need help and our location.

3. **Smart Environment Monitoring**

   - **Forest Fire Detection:** Monitoring of combustion gases and preemptive fire conditions to define alert zones.
   - **Air Pollution:** Control of CO2 emissions of factories, pollution emitted by cars and toxic gases generated in farms.
   - **Water Quality:** Study of water suitability in rivers and the sea for eligibility in drinkable use.
   - **River Floods:** Monitoring of water level variations in rivers, dams and reservoirs during rainy days.

4. **Smart Industry**

   - **Explosive and Hazardous Gases:** Detection of gas levels and leakages in industrial environments, surroundings of chemical factories and inside mines, monitoring of toxic gas and oxygen levels inside chemical plants to ensure workers and goods safety, monitoring of water, oil and gas levels in storage tanks and Cisterns.
   - **Maintenance and repair:** Early predictions on equipment malfunctions and service maintenance can be automatically scheduled ahead of an actual part failure by installing sensors inside equipment to monitor and send reports.

5. **Smart City**

   - **Smart Parking:** Real-time monitoring of parking spaces availability in the city making residents able to identify and reserve the closest available spaces.
   - **Traffic Congestion:** Optimizing the driving and walking routes depending on number of pedestrian and vehicles traffic.
   - **Smart Lighting:** Intelligent and weather adaptive lighting in street lights.

6. **Smart Agriculture**

   A network of different sensors can sense data, perform data processing and inform the farmer through communication infrastructure e.g., mobile phone text message about the portion of land that need particular attention. This may include smart packaging of seeds, fertilizer and pest control mechanisms that respond to specific local conditions and indicate actions. Intelligent farming system will help agronomists to have better understanding of the plant growth models and to have efficient farming practices by having the knowledge of land conditions and climate variability. This will significantly increase the agricultural productivity by avoiding the inappropriate farming conditions [15].

1.3 Sensors/Electronic Devices

1.3.1 Definition

A sensor is a device that detects and responds to input from the physical environment. The input can be light, heat, motion, humidity, pressure, or any number of other environmental phenomena. The output is usually a signal that is converted to a human-readable display at the sensor site, or transmitted electronically over a network for reading or further processing.

In a way, sensors are used to capture signals and convert those signals into the desired format. There are different types of sensors used to collect information such as: sound sensors, humidity sensors, fog sensors and smoke sensors. Electronic devices such as laptops, computers, tablets, cell phones, and other security systems that connect to the Internet through a wireless connection [16] [17].

Sensors are very important in every IoT role. They are able to create an ecosystem to collect and process data about a given environment, making it easier and more efficient to monitor, manage and control it. IoT sensors are used in homes, out in the field, in automobiles, on airplanes, in industrial settings and in other environments. Sensors bridge the gap between the physical and logical worlds, acting as the eyes and ears of the computing infrastructure, analyzing the data collected by the sensors and processing them appropriately [17].

1.3.2 Types of Sensors

There are different types of sensors, from very simple to complex. Sensors can be classified according to their specifications, conversion method, type of material used, detection of physical phenomena, measured characteristics and field of application.

1. **Pressure Sensors**

   Pressure sensors are used to measure gas or liquid pressure including water level, flow, speed, and altitude. Practical examples include sensors for pumps and compressors, hydraulic systems, and refrigerators. The pressure sensor usually acts as a transducer where it generates a signal as a function of the pressure applied [18].

2. **Flow Sensors**

   Flow sensors are used to detect and record fluid flow rates in a pipe or system. Flow sensors can also be used to measure the flow / heat transfer caused by moving media. Feeling and measuring flow is very important for many applications down to more serious applications such as flow monitoring for high purity acids and others [18].

3. **Motion Sensors**

   Motion sensors can detect physical movement within a defined space and can be used to control lights, cameras, parking barriers, faucets, security systems, automatic door openers and many other systems. Sensors typically emit some type of energy such as microwaves, ultrasound, or light beams and can detect when the flow of energy is interrupted by something in its path [17].

4. **Temperature Sensors**

   These sensors can determine the temperature of the target medium, be it gas, liquid

or air. Temperature sensors are used in various devices and environments, such as in appliances, machines, airplanes, cars, computers, greenhouses, farms, thermostats and many others [17].

5. **Humidity Sensors**

   These sensors can detect the level of water vapors in the air to determine the relative humidity. Humidity sensors often include temperature readings because relative humidity is dependent on the air temperature. These sensors are used in a variety of industries and environments, including agriculture, manufacturing, data centers, meteorology, ventilation and air conditioning [17].

6. **Chemical Sensors**

   Chemical sensors detect specific chemicals in a medium (gas, liquid or solid). Chemical sensors can be used to detect soil nutrient levels in agricultural fields, smoke or carbon monoxide in a room, pH in water bodies, or many other conditions [17].

7. **Water Quality Sensors**

   Water quality sensors are used for ion monitoring. Water quality is measured by water quality sensors. Researchers in [19] proposed the design of a low-cost system that measures water temperature, pH, turbidity, conductivity, dissolved oxygen and monitors water quality in IoT scenarios [20].

1.4 Data Processing

A system that receives data from sensors and processes that information is called a data processing system. Embedded systems have to process the data and extract useful information and send the extracted information to a cloud-based system or a server's database [16].

1.5 Water Quality

In the part of our project. We will focus on one of the applications where we can use the IoT, which is Water Quality.
Water quality refers to the chemical, physical and biological properties of water which determines the suitability of that water for a particular area values such as drinkability,

ecosystem health, agriculture and industry. However, obtaining accurate and efficient water quality models remains a challenge often in complex water systems due to variation, complexity, anomalies and noise encountered in the real world during data collection and model structure [21].

Currently, quality control techniques of water monitoring analysis are primarily focused on analytical laboratory tests requiring toxic chemicals, trained personnel and longtime. Therefore, conventional data processing techniques are no longer sufficient in addressing water quality issues. However, the advent of the Internet of Things (IoT) provides opportunities for reporting operational data in real-time conditions to ensure a good quality outflow from the water bodies [22].
Monitoring of water systems using soft computing and communication technology provides a better alternatives as it is fast, efficient and eco-friendly, does not require harmful chemicals and provide a level of real-time public health security compared to the current laboratory based on approaches that are too slow to develop functional response.

1.5.1 Water Quality Anomaly Detection

Water quality is a term that describes the physical, chemical and biological properties of water in relation to its suitability for its intended purpose or use. It is critical to ensure that water quality meets expectations according to recognized standards. Therefore, the overall goal of water quality anomaly is to detect contamination faster and more accurately [23].

1.5.2 Traditional Manual Water Quality Monitoring Approach

Over years, Water Quality Monitoring (WQM), for detecting anomalies, has evolved from traditional laboratory based on manual methods to traditional manual in situ methods. More recently, techniques based on Wireless Sensor Networks (WSNs) have been used, where real-time sensor measurements are analysed using data analytic methods to detect abnormal events. Common traditional WQM methods are briefly discussed below [24]:

- **Physical observation**

  Water quality is usually assessed by observing contaminants in the water. To the extent that this is still common today, water company personnel and consumers still act as sensors, providing insights into drinking water safety based on the presence or absence of certain water quality indicators observed.

- **Laboratory-based analysis method**

  This method is still widely used by water companies, where water samples are taken randomly or at specified time intervals for laboratory analysis by highly qualified personnel using special equipment to verify the safety of drinking water. There are several laboratory-based methods that differ in the contaminants to be detected, the sensitivity, and the time required for detection. However, these methods can only take into account the time and place of sampling.

- **Handheld detection devices**

  Portable handheld devices, sometimes with wireless networking capabilities, have been developed and are currently used to detect water contamination in small bodies of water.

Drawbacks

These traditional WQM methods are time-consuming and challenging in detecting contaminants in low concentrations. Moreover, they cannot meet the needs of real-time, multiple and heterogeneous water quality parameters, as well as the need for high accuracy detection of water quality events across the entire water distribution system, hence the need for detection using Artificial Intelligence techniques on data obtained from stationed on-site multiple water quality sensors [24].

1.6 Internet of Things In Water Quality Monitoring

Researchers are seeking more intricate techniques for conducting real-time monitoring of the quality of surface and groundwater that is assessable to the human population across various locations. Digital communication technologies are now the bedrock of modern society and IoT enabled water quality monitoring is a vital aspect of that [25]. The Internet of Things (IoT) is an integrated system that has been widely used in a

variety of applications ranging from smart supply chains, smart cities, and smart power grids to smart wearables. IoT has sparked considerable interest as a method of connecting everyday objects via sensors and establishing a network to send and receive data at a lower cost. Also, data from a network of different chipboard (such as ARM-based Raspberry Pi microcomputers, ARDUINO microprocessors, ESP8266, and ESP32) are attached to sensors to collect real-time data, which is displayed by a server computer and processed by Artificial Intelligence such as Machine Learning tools [22].

Traditional water quality monitoring methods require manual use of instruments to take readings, and recording data is considered inefficient, slow and expensive. This section demonstrates the importance of IoT in water quality monitoring and its advantages over traditional water sampling and analysis methods used by environmental engineers and scientists in water quality monitoring [25].

1. The most significant advantage of IoT in water quality monitoring is the possibility of real-time monitoring. The water quality status (based on various indicators) can be called up here at any time. This is made possible by the speed of internet communication, where data can be transmitted from sensors in fractions of a second. These incredible speeds are unattainable with traditional water quality monitoring.

2. IoT in water quality monitoring can be automated. This means that it does not require the presence of human personnel to take readings and log data. Moreover, these IoT systems would require less human resources and eliminate human errors in data logging and computations. Automation is the foundational concept of smart cities and its associated technologies.

3. Alongside the advantage of automation, IoT has led to the use of adaptive and responsive systems in water quality monitoring. These smart-systems can alert authorities or personnel regarding impending danger (such as high water level of an impending flood) or non-optimal conditions (such as in aquaponic systems).

4. IoT in water quality monitoring is cheaper than hands-on personnel conducting the monitoring. The cost of human resources is minimised, and an IoT based system would not require [25].

1.7 Conclusion

The Internet of Things is becoming an emerging concept. Billions or trillions of machines communicate with each other and exchange vast amounts of data to extract information and make real-time or non-real-time decisions.
In this chapter, we introduced how can Internet of Things be helpful to provide researches to obtain a suitable water.

In the next chapter we will focus on artificial intelligence based on introducing machine learning, deep learning and their different learning methods.

Chapter 2

Artificial Intelligence

2.1 Introduction

Internet of Things devices are used to collect real-time data from a given water sample or station. At the same time, Artificial Intelligence (AI) deals with the evaluation, simulation and prediction of data for easy interpretation and future use.
AI is the most fascinating and most discussed Technology of the current decade because of its nature to mimic human intelligence. AI systems are able to do the specific types of jobs for which they were trained.

This chapter starts with the definition of artificial intelligence, and then introduces machine learning and its types. Finally, we will look at deep learning, give some definitions, and we will detail its different types.

2.2 Artificial Intelligence

2.2.1 Definition

Artificial Intelligence (AI) is the theory and development of computer systems able to perform tasks usually requiring human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages [26].
In a sense, AI is a technique of getting machines to work and bahave like humans. In the rest part, AI has been able to accomlish this by creating machines and robots that have been used in wide range of fields including healthcare, robotics marketing, business analytics, and many more.

2.2.2 Branches of Artificial Intelligence

Artificial intelligence is a technology that exists in almost every field. In order to better deal with the various problems of society, it is divided into many branches, each one related to a specific problem. The major branches of Artificial Intelligence are Experts Systems, Robotics, Machine Learning, Neural Network, Fuzzy Logic and Natural Language Processing [27].
Among these branches, we will concentrate on Machine Learning.

2.3 Machine Learning

Machine Learning (ML) is a subset of methods of Artificial Intelligence. Its aim is to develop algorithms that learn interpretation principles from training samples, and apply them to new data from the same domain to make informed decisions [28].

2.3.1 Types of Machine Learning

Machine Learning systems are not explicitly programmed for a specific task, they are trained for some special tasks. Learning techniques can be divided into four categories. These are four types of machine learning supervised learning, unsupervised learning,semi-supervised-learning, and reinforcement learning.

2.3.1.1 Supervised Learning

Supervised Learning is simply a formalization of the idea of learning from examples. In supervised learning, the learner (typically, a computer program) is provided with two sets of data, a training set and a test set. The idea is for the learner to "learn" from a set of labeled examples in the training set so that it can identify unlabeled examples in the test set with the highest possible accuracy. That is, the goal of the learner is to develop a rule, a program, or a procedure that classifies new examples (in the test set) by analyzing examples it has been given that already have a class label [29].

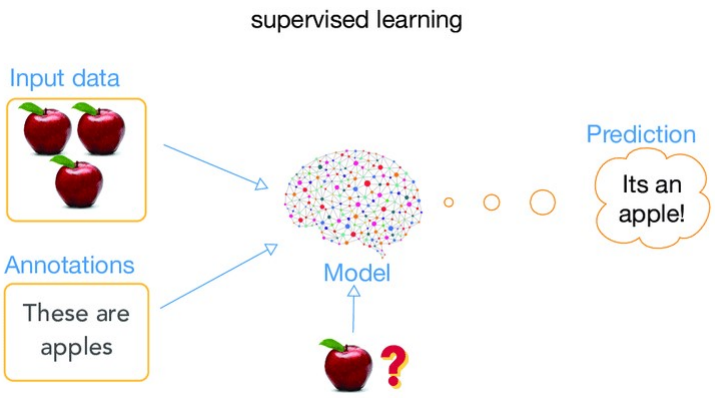

Figure 2.1: Example of Supervised Learning.

Supervised learning solves two problems:

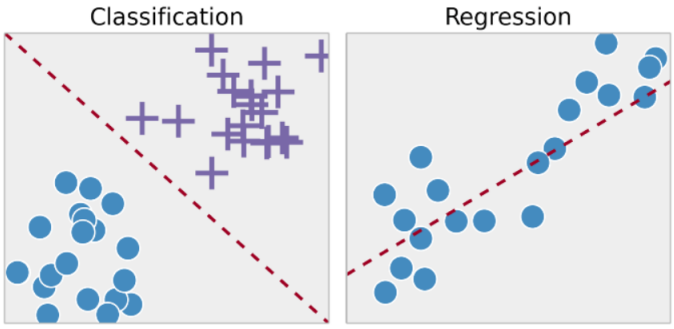

Figure 2.2: Types of Supervised Machine Learning Techniques [3].

1. **Regression**

   Regression techniques use training data to predict a single output value. For example, we can use regression to predict house prices from training data. The input variables are the location, the size of the house. etc. [3].

2. **Classification**

   Classification means to group the output into a class. When an algorithm tries to classify the input into two different classes, it is called binary classification. To choose between more than two classes is called multi-class classification [3].

2.3.1.2 Unsupervised Learning

The goal is to have the computer learn how to do something that we don't tell it how to do! Unsupervised learning techniques require only the input feature values in the training data and the learning algorithm discovers hidden structure in the training data based on

them Clustering techniques that try to partition the data into coherent groups fall into this category [30].

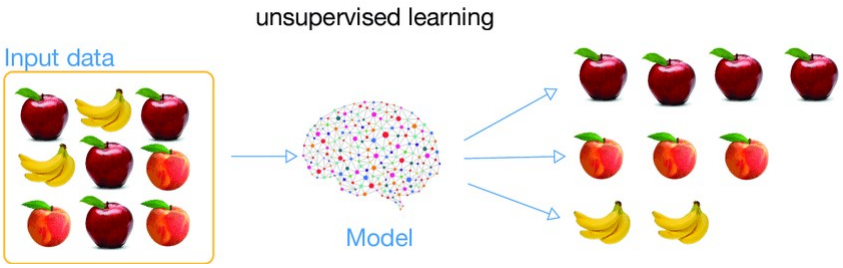

Figure 2.3: Example of Unsupervised Learning.

Unsupervised learning problems further grouped into clustering and association problems.

1. **Clustering**

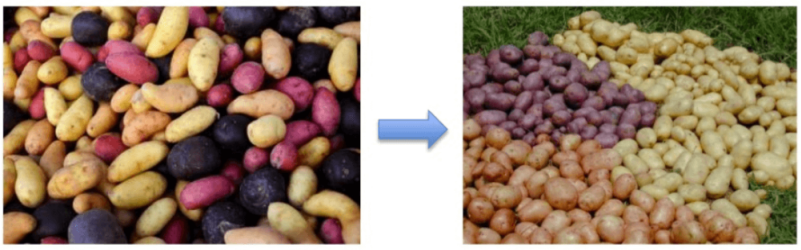

Figure 2.4: Clustering.

Clustering is an important concept when it comes to unsupervised learning. It mainly deals with finding a structure or pattern in a collection of uncategorized data. Clustering algorithms process our data and find natural clusters (groups) when they exist in the data. We can also change the number of clusters we want the algorithm to recognize. This allows us to adjust the granularity of these groups. [3].

2. **Association**

   Association rules allow us to establish associations between data objects in large databases. This unsupervised technique is about discovering interesting relationships between variables in large databases. For example, people buying new homes are most likely to buy new furniture [3].

Other Examples:

- A subgroup of cancer patients grouped by their gene expression measurements.
- Groups of shopper based on their browsing and purchasing histories.
- Movie group by the rating given by movies viewers.

2.3.1.3 Semi-Supervised Learning

Supervised learning requires labeled data, and using supervised learning to classify unlabeled data is not possible. However, most of the available data are unlabeled. On the other hand, clustering unlabeled data using unsupervised learning presents significant challenges for describing clusters and determining the optimal number of clusters. In this case, semi-supervised learning becomes useful. This is where the model is trained on labeled and unlabeled data simultaneously. Usually large amount of unlabeled data and some labeled data to identify the cluster and determining a optimum numbers of clusters. The main benefit of this is labeling all the data is not required and some unknown pattern or clusters may be discovered. Some of applications are protein sequence analysis, web page classifications and speech classification [31].

2.3.1.4 Reinforcement Learning

Reinforcement learning enables an agent to learn by interacting with its environment. The agent will learn to take the best actions that maximize its long-term rewards by using its own experience [32].

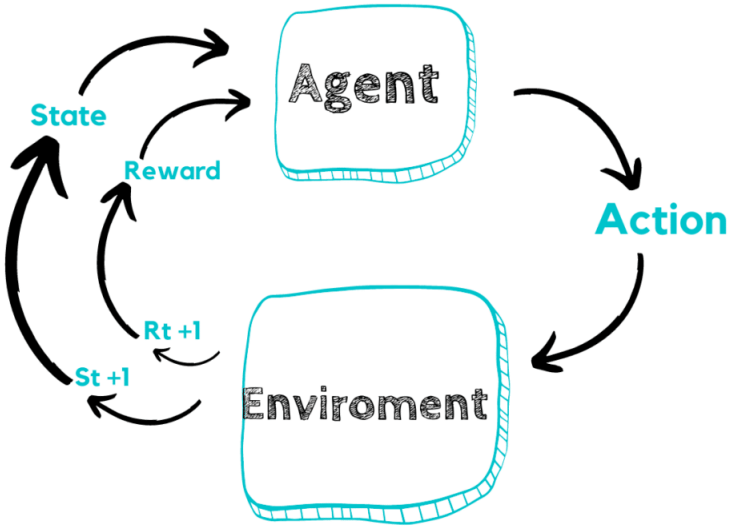

Figure 2.5: Reinforcement Learning.

2.3.2 Machine Learning Algorithms

Machine Learning refers to giving machines the ability to make decisions using past experiences through learning. Machine learning algorithms are classified into supervised and unsupervised algorithms. These algorithms are classified based on the task in hand. Supervised machine learning algorithms are used for classification, regression, and predictions whereas unsupervised machine learning approaches are used for clustering applications, outlier detection,etc.

2.3.2.1 Supervised Machine Learning Algorithms

1. **K-Nearest Neighbour (KNN)**

   KNN algorithm is one of the supervised machine learning algorithm that can be used for classification and regression. It stores the dataset and performs operations on the dataset during classification instead of learning straight from the training set, it is widely known as the lazy learning algorithm. Because it does not learn anything during the training phase [4].

   How does KNN make a prediction?

   To make a prediction, the KNN algorithm will use the whole dataset. Indeed, for an observation, which is not part of the dataset, that we want to predict, the algorithm will look for the K instances of the dataset closest to our observation. Then for these K neighbors, the algorithm will rely on their output variables y to calculate

the value of the variable y of the observation we want to predict [33].

On the other hand:

- If KNN is used for regression, the mean (or median) of the y variables of the K nearest observations will be used for prediction [33].
- If KNN is used for classification, it is the mode of the y variables of the K nearest observations that will be used for prediction [33].

Algorithmic writing

The steps of the KNN algorithm are described by the following pseudo code [33]:

The kNN algorithm

Start Algorithm

Input data:

- a set of data D
- a function of definition distance d
- An integer K

For a new observation X whose output variable y is to be predicted Do:

(a) Compute all distances of this observation X from the other observations in the dataset D

(b) Retain the K observations of the dataset D that are closest to X by using the distance calculation function d

(c) Take the y values of the K selected observations:

  i. If we perform a regression, compute the mean (or the median) of the selected y

  ii. If we perform a classification, compute the mode of the selected y

(d) Return the value calculated in step (c) as the value that was predicted by KNN for the observation X

End Algorithm

Similarity calculation in the KNN algorithm

As we have just seen in the algorithm writing, KNN needs a function to calculate the distance between two observations. The closer two points are to each other, the more similar they are and vice versa.

There are several distance functions, such as the Euclidean distance, the Manhattan distance,etc. We choose the distance function according to the type of data we are handling. Thus for quantitative data (example: weight, wages, height, amount of electronic basket, etc.) and of the same type, the Euclidean distance is a good candidate. As for the Manhattan distance, it is a good measure to use when the data (input variables) are not of the same type (example: age, sex, length, weight, etc.) [33].

Here are the mathematical definitions of the distances we just mentioned:

- **The Euclidean distance:**

  Distance that calculates the square root of the sum of the square differences between the coordinates of two points [33]:

$$D(x, y) = \sqrt{\sum_{j=1}^{n} (x_j - y_j)^2} \tag{2.1}$$

  Where: x and y are vectors.

- **The Manhattan distance:**

  Calculates the sum of the absolute values of the differences between the coordinates of two points [33]:

$$d(x, y) = \sum_{i=1}^{k} |x_i - y_i| \tag{2.2}$$

  Where: x and y are vectors.

Choosing the right value for K

The k in kNN is a parameter that refers to the number of nearest neighbors to include in the majority voting process.

The choice of the K value to be used to make a prediction with KNN, varies according to the dataset. In order to choose the right K for the data, we run the KNN algorithm several times with different values of K and choose a K that reduces the number of errors we encounter while maintaining the algorithm's ability to accurately make predictions when it's given data it hasn't seen before [34].

The selection of different values for k can generate different classification results for the same sample object.

Figure 2.6 shows an illustration of how the KNN works to classify a new object.

For K=3, the new object (star) is classified as 'black' because there are 2 black points, however, it has been classified as 'red' when K=5.

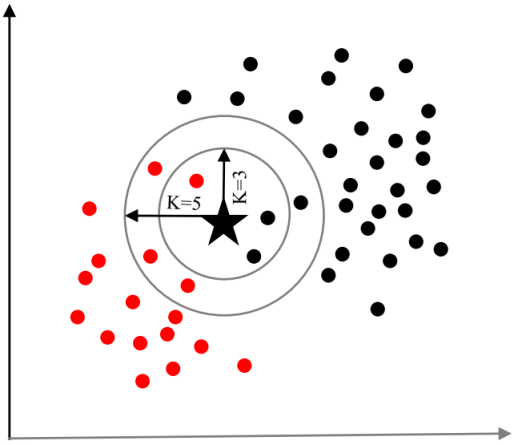

Figure 2.6: Illustration of how the KNN works to classify a new object [4].

Advantages of the KNN algorithm

- The algorithm is simple and easy to implement.
- The algorithm is versatile. It can be used for classification and regression.
- There is no need to train a model [34].

Disadvantages of the KNN algorithm

- The testing phase of K-nearest neighbor classification is slower and costlier in terms of time and memory.
- It requires large memory for storing the entire training dataset for prediction.
- Stores all the training data [34].

2. **Linear regression**

   The goal of the linear regression, as a part of the family of regression algorithms, is to find relationships and dependencies between variables. It represents a modeling relationship between a continuous scalar dependent variable y (also label or target in machine learning terminology) and one or more (a D-dimensional vector) explanatory variables (also independent variables, input variables, features, observed data, observations, attributes, dimensions, data point, etc.) denoted X using a linear function. In regression analysis the goal is to predict a continuous target variable, whereas another area called classification is predicting a label from a finite set. The model for a multiple regression which involves linear combination of input variables takes the form [5]:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + ... + e$$

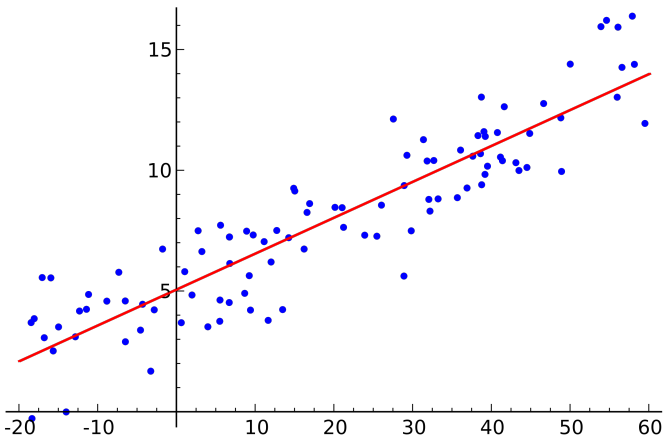

Figure 2.7: Illustration of Linear regression [5].

As shown on Figure 2.7, the model (red line) is calculated using training data (blue points) where each point has a known label (y axis) to fit the points as accurately as possible by minimizing the value of a chosen loss function. We can then use the model to predict unknown labels (we only know x value and want to predict y value) [5].

3. **Decision Tree (DT)**

   Decision Tree is a supervised machine learning algorithm. It can be used for both classification and regression. The goal is to create a model that predicts the class or value of the target variable by learning simple decision rules derived from previous (training) data [35].

   Decision Tree Terminologies

   - **Root Node:** The top node of the decision tree. It represents the entire dataset, which is further divided into two or more homogeneous sets.
   - **Splitting:** Is the process of dividing a node into sub-nodes.
   - **Decision Node:** When a sub-node splits into further sub-nodes, it is called the decision node.
   - **Branch/Sub Tree:** A tree formed by splitting the tree.
   - **Leaf/Terminal Node:** Is the final output node, after getting a leaf node, the tree cannot be split further [6].

Figure 2.8 shows the general structure of a decision tree.

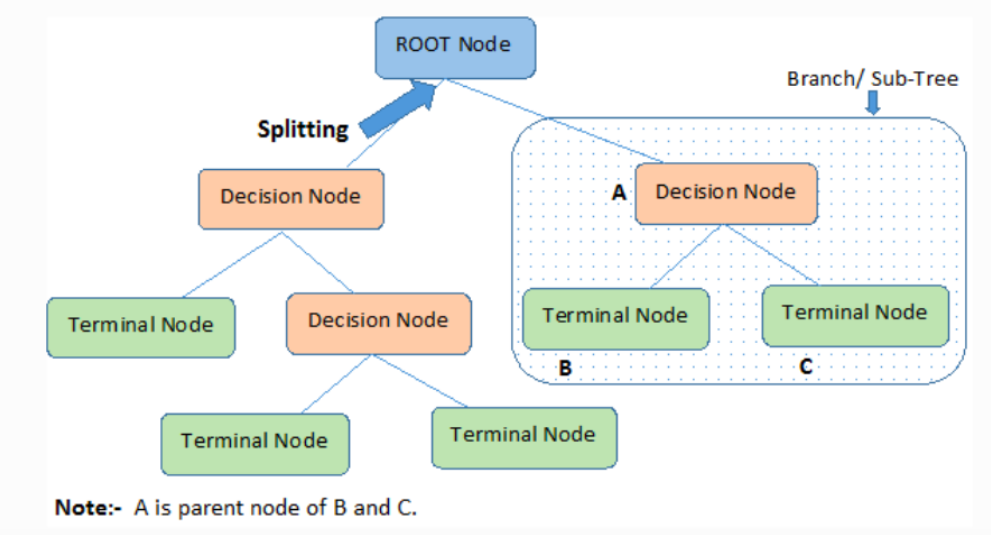

Figure 2.8: General structure of a decision tree [6].

How does the Decision Tree algorithm Work?

Decision trees make decisions by splitting nodes into sub-nodes. This process is repeated many times during training until only homogeneous nodes remain. The algorithm starts from the root node of the tree. The algorithm compares the value of the root attribute with the record attribute (real dataset) and follows the branch and jumps to the next node based on the comparison. For the next node, the algorithm again compares the attribute value with other sub-nodes and moves on. It continues this process until it reaches the leaf node of the tree. The complete process can be better understood using the below algorithm [36].

- **Step-1:** Start the tree from the root node, says S, which contains the complete dataset.
- **Step-2:** Use **Attribute Selection Measures (ASM)** to find the best attributes in the dataset.
- **Step-3:** Divide S into subsets containing the possible values of the best attribute.
- **Step-4:** Generate the decision tree node with the best attributes.
- **Step-5:** Build a new decision tree recursively using the subset of the dataset created in **Step-3**. Continue this process until you cannot further classify the nodes and call the last node a leaf node.

Attribute Selection Measures (ASM):

The main issue when implementing a decision tree is determining the best attributes for the root and sub-nodes. To solve this type of problem, there is a technique called Attribute Selection Measure, is used. With this measurement, we can easily select the best attributes for the tree's nodes.

Information Gain is one of the ASM techniques [36].

First, the entropy of target is determined. Next, the dataset is split on the attributes and the sum of entropy for all the classes in the attributes is calculated. The resulting entropy is then subtracted from the target entropy before splitting. The attribute with the highest information gain is chosen to split the dataset. This process is repeated over and over again until we get a pure classification.

The entropy is calculated using the following formula [14]:

$$Entropy(S) = \sum_{i=1}^{c} -p(i) \log_2 p(i) \tag{2.3}$$

Here, "c" is the number of classes of an attribute. "pi" is the fraction of examples of the class "i" [14].

$$InformationGain(S, A) = Entropy(S) - Entropy(A) \tag{2.4}$$

Here, S is the parent and A is the attribute that we want to split. Information gain is basically the decrease in entropy due to partitioning of data set based on an attribute. Decision tree is split based on attributes with higher information gain [14].

Advantages of the DT algorithm

- It is simple to understand because it follows the same process that humans use when making decisions in real life.
- It can be very helpful in solving decision-making problems.
- It is beneficial to consider all possible solutions to a problem.
- When compared to other algorithms, it requires less data cleaning [36].

Disadvantages of the DT algorithm

- The decision tree has many layers, which makes it complex.
- It may have an overfitting problem, which the Random Forest algorithm can solve.
- The computational complexity of the decision tree may increase as more class labels are added [36].

4. **Random Forest (RF)**

   RF algorithm creates the forest with many Decision Trees. It is a supervised learning algorithm that can be used for both Classification and Regression problems in ML. It is an attractive classifier due to the high execution speed. Many Decision Trees ensemble together to form a random forest, and it predicts by averaging the predictions of each component tree. DTs that are grown very deep often cause overfitting of the training data, resulting a high variation in classification outcome for a small change in the input data. They are very sensitive to their training data, which makes them error-prone to the test dataset. The different DTs of an RF are trained using the different parts of the training dataset. To classify a new sample, the input vector of that sample is required to pass down with each DT of the forest. Each DT then considers a different part of that input vector and gives a classification outcome. The forest then chooses the classification of having the most 'votes' (for discrete classification outcome) or the average of all trees in the forest (for numeric classification outcome). Since the RF algorithm considers the outcomes from many different DTs, it can reduce the variance resulted from the consideration of a single DT for the same dataset [37][38].

   The working process can be explained with the following steps:

   - **Step 1:** Choose random K data points from the training set.
   - **Step 2:** Create a decision tree related to the selected data points (subset).
   - **Step 3:** Each decision tree produces an output.
   - **Step 4:** Consider final output based on majority vote or average [39].

Figure 2.9 shows an illustration of the RF algorithm.

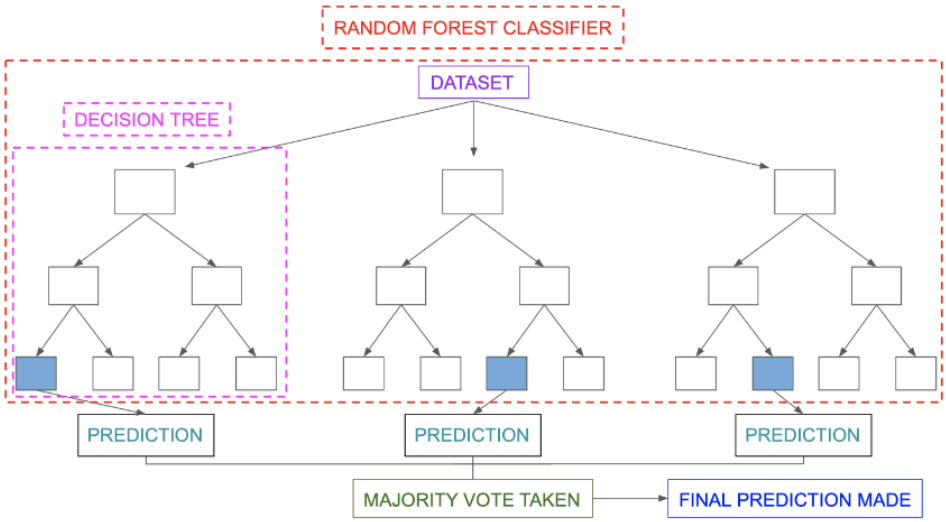

Figure 2.9: Illustration of a Random Forest [7].

Advantages of the RF algorithm

- It can be used in classification and regression problems.
- It solves the problem of overfitting as output is based on majority voting or averaging.
- It enhances the accuracy of the model [39].

Disadvantages of the RF algorithm

Training time is more compared to other models. Whenever it has to make a prediction each decision tree has to generate output for the given input data [39].

2.3.2.2 Unsupervised Machine Learning Algorithms

There are several unsupervised learning algorithms such as K-means clustering, Hierarchical clustering and Recommender system, etc. We quote one of these algorithms which is K-means clustering.

- **K-Means Clustering**

  K-means is one of the simplest unsupervised learning algorithms that solve the well known clustering problem. The procedure follows a simple and easy way to classify a given data set through a certain number of clusters. The main idea is to define

k centers, one for each cluster. These centers should be placed in a cunning way because of different location causes different result. So, the better choice is to place them is much as possible far away from each other.

The next step is to take each point belonging to a given data set and associate it to the nearest center. When no point is pending, the first step is completed and an early group age is done. At this point we need to re-calculate k new centroids as bary center of the clusters resulting from the previous step [8].

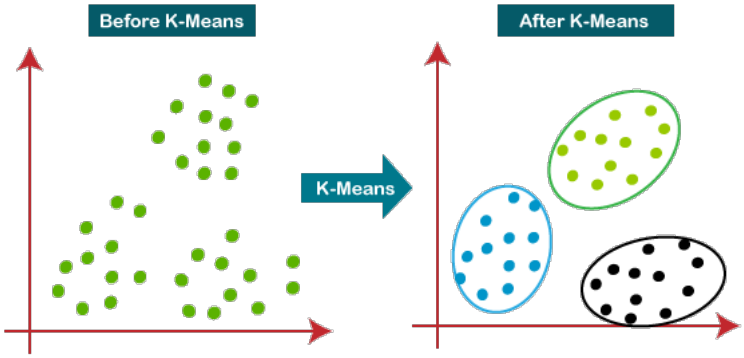

Figure 2.10: Illustration of K-Means Clustering [8].

2.4 Deep Learning

2.4.1 Definition

Deep Learning (DL) is a subset of Machine Learning, which in turn is a subset of Artificial Intelligence.

Deep Learning is a new area of Machine Learning research, inspired by the structure of the human brain, in terms of Deep Learning this structure is called an artificial neural networks. It is based on learning several levels of representations, corresponding to hierarchy of features or concepts, where higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts [40].

2.4.2 Deep Learning algorithms

There are many kinds of deep learning algorithms, and different algorithms can be used to represent different data sources. For example, convolutional neural networks are the most popular image recognition algorithms, while recurrent neural networks are better suited for sequential tasks like handwriting or speech recognition. In this section, we

briefly outline three well-established algorithms: Deep Belief Networks (DBN), Convolutional Neural Networks (CNN), and Recurrent Neural Networks (RNN).

1. **Deep Belief Networks (DBN)**

   DBN consists of multi-layers of neurons. These neurons are divided into hidden units and visible units. Visible units are used to receive data, while hidden units are used to extract features, so they are also called feature detectors [41].

   A DBN is composed of multiple layers of Restricted Boltzmann Machines (RBM). which is a type of neural network. An RBM is a two-layer model that consists of a single visible layer and a single hidden layer. In an RBM, the hidden and visible layers are fully connected via symmetrical and undirected weights, and the units within a layer are not connected. The hidden units are trained to model higher-order data correlations that are observed at the visible units [9].

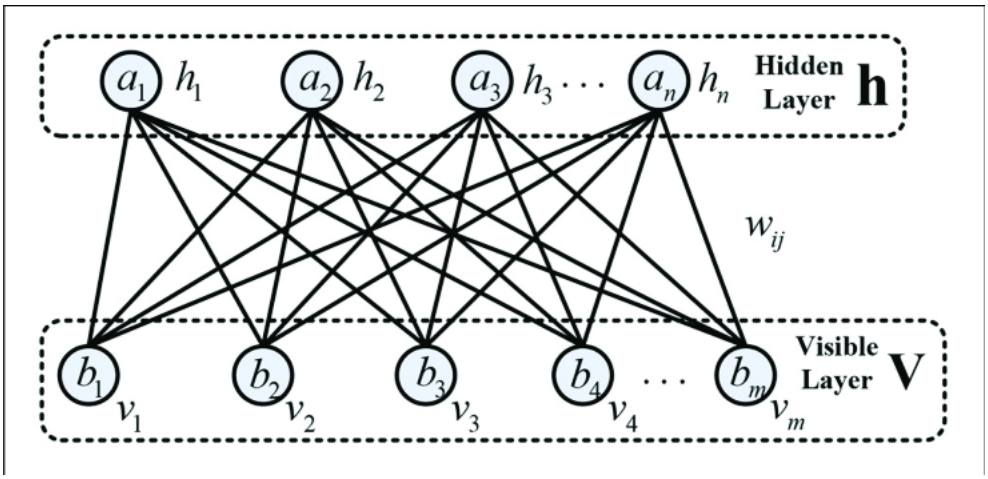

Figure 2.11: Illustration of Restricted Boltzmann Machines [9].

To determine the generative weights of an RBM, a greedy layerwise pretraining algorithm is adopted. The pretraining algorithm is an unsupervised algorithm that uses unlabeled data to determine the generative weights of an RBM. The weights and biases of an RBM determine the energy of a joint configuration for the hidden and visible [9].

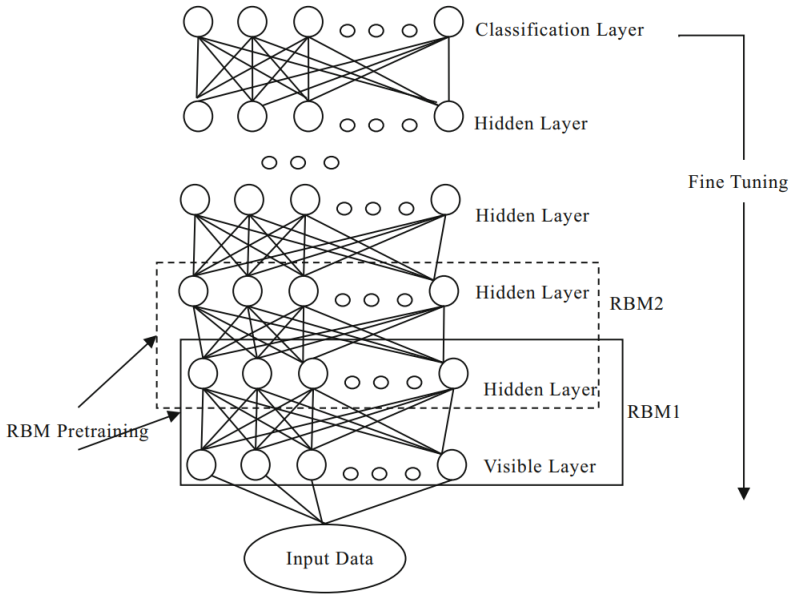

Figure 2.12: Illustration of Deep Belief Networks [9].

Before fine-tuning, a layer-by-layer pre-training of RBMs is performed: the outputs of a RBM are fed as inputs to the next RBM and the process repeats until all the RBMs are pretrained. This layer-by-layer unsupervised learning is critical in DBN training as practically it helps avoid local optima and alleviates the overfitting problem that is observed when millions of parameters are used. Furthermore, the algorithm is very efficient in terms of its time complexity, which is linear to the number and size of RBMs. Features at different layers contain different information about data structures with higher-level features constructed from lower-level features [9] [41].

After pretraining a stack of RBMs, we can use the bottom-up recognition weights of the resulting DBN to initialize the weights of a multi-layer feed-forward neural network, which can then be discriminatively fine-tuned by using back-propagating error derivatives. If working as a classifier, the feed-forward neural network should have a final "softmax" layer added that calculates a probability distribution over class labels, and the derivative of the log probability of the correct class (during this phase, we use labeled data to determine the weights of the network) should be back-propagated to train the incoming weights of the final layer and to discriminatively fine-tune the weights in all lower layers [9] [41].

2. **Convolutional Neural Network (CNN)**

   CNN is a type of feed-forward artificial neural network which uses convolution in at least one of their layers. It was inspired by biological neural networks. CNN combines artificial neural networks and discrete convolution for image processing which can be used to automatically extract features. Thus it is particularly designed for recognizing two-dimensional data, such as images and videos. Images can be directly used as the input of the network, which avoids the complex feature extraction and data reconstruction process in traditional image recognition algorithms [10].

3. **Recurrent Neural Networks (RNN)**

   RNN aim to process sequential data. In the traditional feed forward neural network model, like CNNs, data flows from the input layer to the hidden layer and then the output layer. There is no connection between neurons in the same layer. Some problems cannot be fully handled by this kind of neural network. This architecture cannot solve the problem where the input data have relationships with each other. For example, we need the previous word in a sentence to predict the next one, because the words in the same sentence are not independent.

   In RNN, the current output and previous output are relevant. To be more specific, the output of the previous step is stored and used to calculate the currentoutput, that is, the input of the network contains both the data from the input layer and the output of the hidden layers from the previous step. This makes them applicable to tasks such as handwriting recognition and speech recognition [10].

   The architecture of RNN is shown in Figure 2.13.

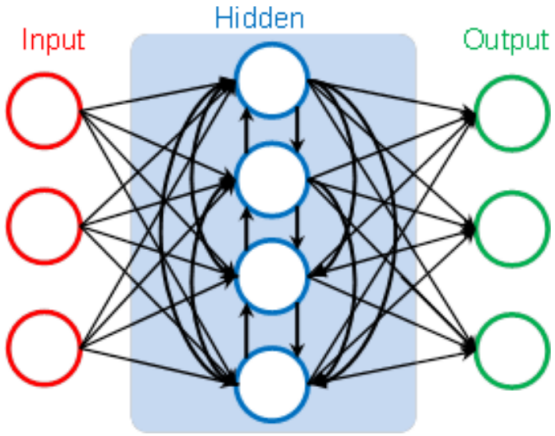

Figure 2.13: Illustration of Recurrent Neural Networks [10].

2.5 Conclusion

Artificial intelligence, machine learning, and deep learning are basically machine perception. It is the ability to interpret sensory data. The training algorithm uses supervised learning to name things and unsupervised learning to group things. The difference between supervised and unsupervised learning is whether we a labeled training set to work with.

In this chapter, we have detailed the basic concepts necessary for the realization of our work. Starting by defining ML and presenting some algorithms, then we talked about deep learning and its algorithms.

In the next chapter, we present the architecture of our proposed approach.

Chapter 3

Water Quality Prediction Based on KNN, DT and RF

3.1 Introduction

We have seen that traditional water quality monitoring methods require manual use of instruments into take read and record data is considered inefficient and slow. On one side we have sensors that collect data in real time and on the other side we have machine learning that has the ability to analyze this data and make decisions. Therefore, it is very important to propose an intelligent approach to predict the water quality.

In this chapter we will present the architecture of the proposed approach, describe the content of our dataset, then we will present some steps followed during the preprocessing phase.

3.2 Related work

In this section, we mention one of the existing projects in this field which is Comparison of Machine Learning Algorithms in Statistically Imputed Water Potability Dataset.

Diwash Poudel, Dhadkan Shrestha, Sulove Bhattarai and Abhishek Ghimire compared in [11] (February 2022) four different machine learning techniques to predict the potability of water. They used Logistic Regression, K-Nearest Neighbours, Artificial Neural network, and Random Forest techniques to obtain the prediction. To achieve the research goals, a technique was developed including dataset collection, data preprocessing, building model

using LR, KNN, ANN, and RF, and training the model using the training dataset, and evaluating the built model with the help of testing dataset. From the results obtained (figure 3.1), the accuracy of LR was 60.51%, KNN was 60.98%, ANN was 69.5% and RF was 70.42%.

They found that using RF was more efficient than ANN, LR and K-NN, as RF had the highest accuracy [11].

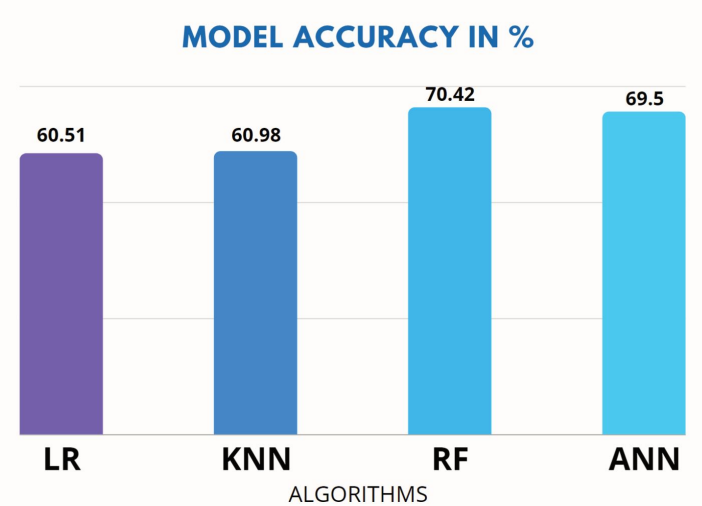

Figure 3.1: Accuracy comparison of various algorithms [11].

3.3 Architecture of our approach

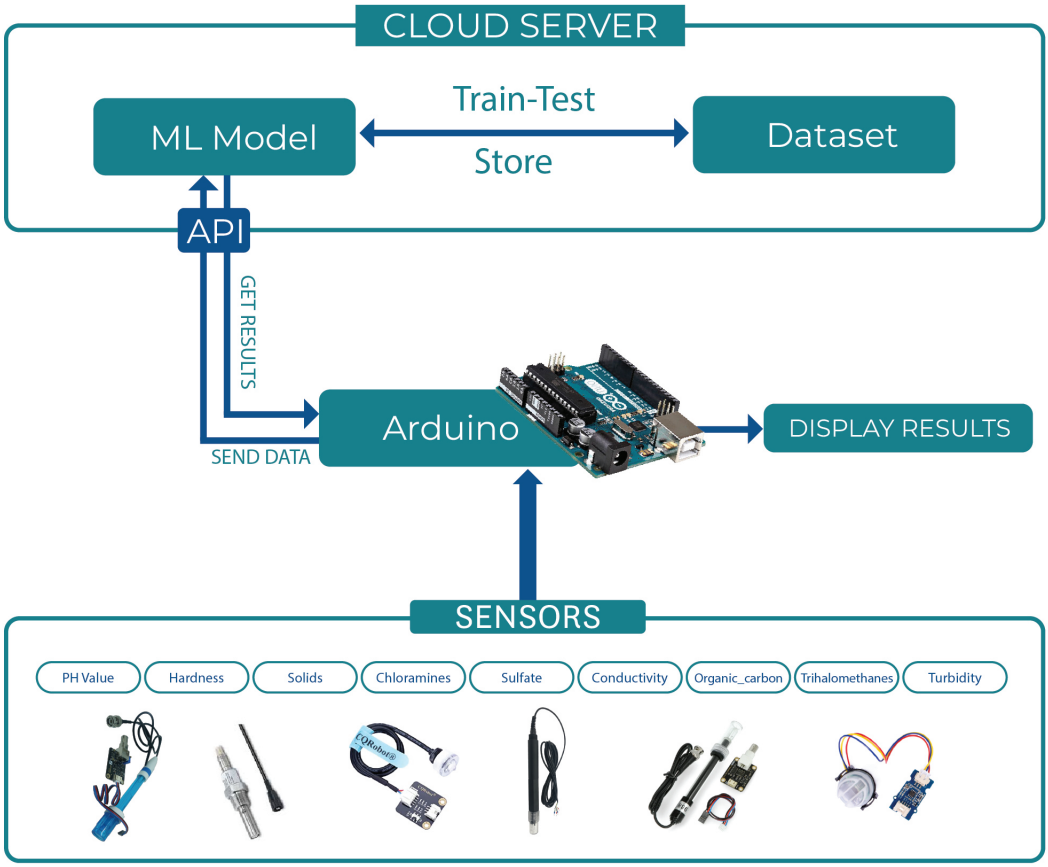

Figure 3.2: Architecture of our approach.

Firstly, We collect data from water bodies (potable and not potable) which we measure their different parameters such as PH value, Hardness, Solids, Chloramines, Sulfate, Conductivity, Organic-carbon, Trihalomethanes and Turbidity. We format the data to arrays rows each row represents values from one test.

Secondly, those classified rows are split to two parts one used to train the machine learning model and the 2nd part which represents only 30% of the dataset is used to test the model accuracy and precision.

Finally, Water bodies that are not yet classified, are measured the same way using sensors and formatted to be sent to the trained model, then the model returns back the results: '0' for not potable and '1' for potable.

3.4 Data Collection

The dataset is the most important resource in any artificial intelligence study. Nowadays, the only thing that is much more expensive than money is information. unfortunately many datasets are not free or publically accessible due to privacy concerns.
Below is shown the free dataset that we find.

3.4.1 Dataset

The dataset used in this study (which is obtained from Kaggle [42].) contains water quality metrics for 3276 different water bodies. The dataset has 10 parameters,namely, pH value, Hardness, Solids (Total dissolved solids), Chloramines, Sulfate, Conductivity, Organic-carbon, Trihalomethanes, Turbidity and Potability. These will be detailed below.

- **pH value**

  The pH of water is a measure of the acid–base equilibrium. World Health Organization (WHO) has recommended 6.5- 8.5 as the range of pH value which is suitable for drinking [43].

- **Hardness**

  Water hardness is related to the amount of dissolved calcium and magnesium in water. It is traditionally defined as the ability of water to react with soap [44].

- **Solids (Total dissolved solids - TDS)**

  TDS include inorganic salts and small amounts of organic matter dissolved in water. The main constituents are usually the cations calcium, magnesium, sodium and potassium and the anions carbonate, bicarbonate, chloride and sulfate. Desirable limit for TDS is 500 mg/l and maximum limit is 1000 mg/l which prescribed for drinking purpose [45].

- **Chloramines**

  Chloramines are disinfectants used to treat drinking water which are usually formed when ammonia is added to chlorine to treat drinking water. Chloramines can provide longer-lasting disinfection as water travels through pipes to consumers. Environmental Protection Agency (EPA) in the USA has established a maximum level for chloramine to be 4 mg/L [46].

- **Sulfate**

  Sulfates occurs naturally in numerous minerals. According to Environmental Protection Agency, the limit is 250 mg/L which is safe for drinking [47].

- **Conductivity**

  Conductivity is a measure of the ability of water to pass an electrical current. Because dissolved salts and other inorganic chemicals conduct electrical current, conductivity increases as salinity increases. According to EPA, the electrical conductivity of water should not cross the limit of 500S/ cm [48].

- **Organic-carbon**

  Organic-carbon is the amount carbon atoms tied up in organic compounds in water [49].

- **Trihalomethanes**

  Trihalomethanes are a group of disinfection by-products that are formed when chlorine compounds used to disinfect water react with other naturally occurring chemicals in the water. They are colorless and evaporate from water into the air [50].

- **Turbidity**

  Turbidity is a measure of the relative clarity of a liquid. It is an optical property of water and is a measure of the amount of light scattered by substances in the water as it passes through a water sample. The higher the scattered light intensity, the higher the turbidity. Materials that cloud water include clays, silts, very small inorganic and organic materials, algae, dissolved colored organic compounds, plankton, and other microorganisms. WHO has established the limit for turbidity of water that should not cross more than 5 Nephelometric Turbidity Units (NTU) [51].

- **Potability**

  Indicates if water is safe for human consumption where 1 means Potable and 0 means Not potable [42].

3.5 Data Preprocessing

When developing a machine learning project, we do not always come across clean and formatted data. And, before performing any operation on data, it must be cleaned and formatted. Therefore, we use the data preprocessing.
Data preprocessing is the first and most important step that cleaning and preparing raw data to use it for a machine learning model. It can help to improves the accuracy and efficiency of the machine learning model.

3.5.1 Dealing With Missing Values

The df.isnull().sum() shows that we have 3 features with missing values (NaN):

- **PH:** 449 missing values
- **Sulfate:** 694 missing values
- **Trihalomethanes:** 147 missing values

The blank area in the missingno matrix also helps visualize the data. (Figure 3.3)

Figure 3.3: Missing values visualization.

To handle missing values present in the dataset we will fill it with mean value by the following code:

```
for x in col:
  df[x] = df[x].fillna(df.groupby(['Potability'])[x].transform('mean'))
```

Figure 3.4: Filling the missing values.

3.5.2 Dealing With Outliers

The boxplot is a statistical plot to visualize a descriptive statistics median quartile 1, quartile 2, quartile 3 and minimum-maximum values. Outliers are numbers outside the group of the rest of the data.

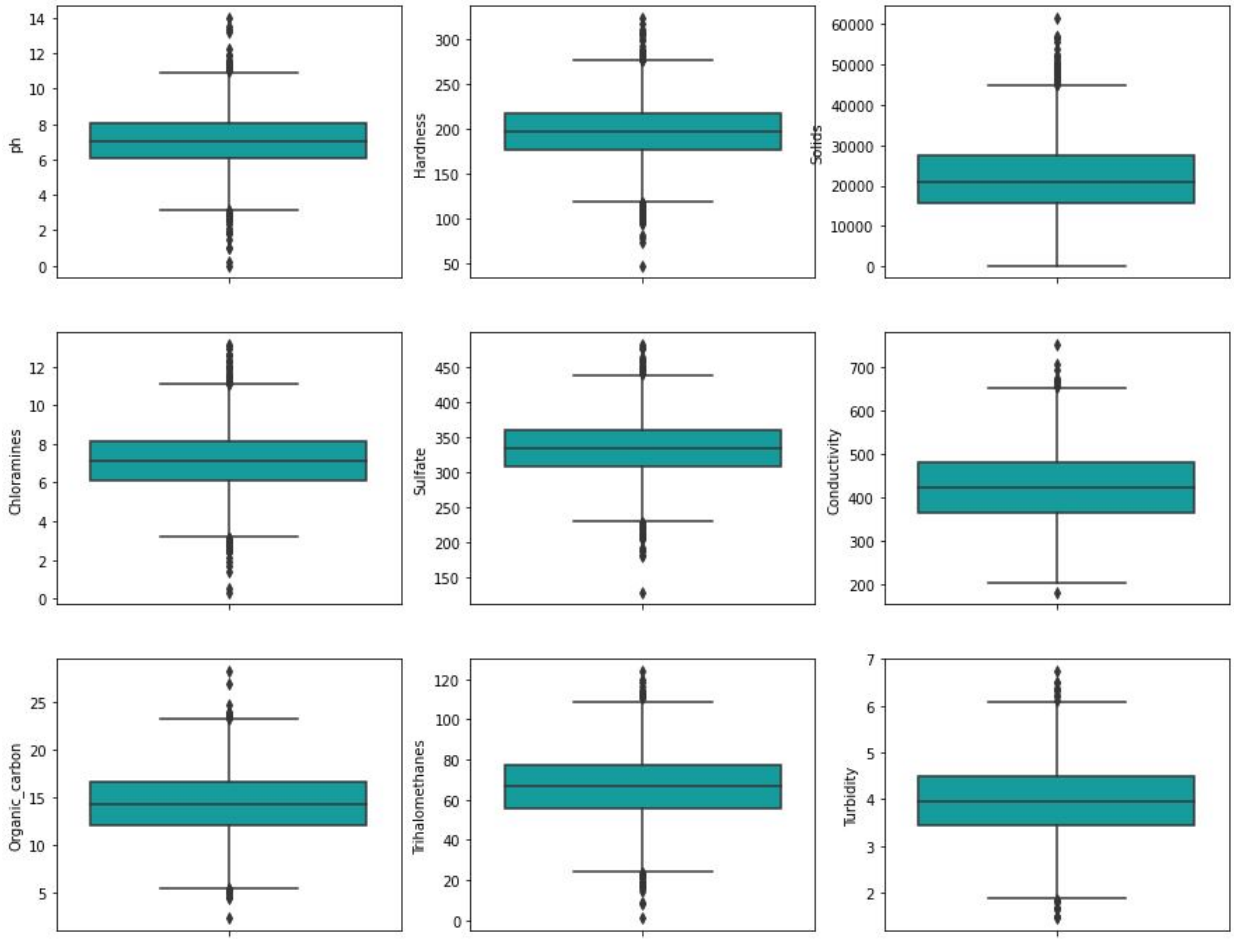

Figure 3.5: Boxplot visualization.

To deal with outliers we used Interquartile Range IQR method:

```
for x in col:
    q75,q25 = np.percentile(df.loc[:,x],[75,25])
    intr_qr = q75-q25

    max = q75+(1.5*intr_qr)
    min = q25-(1.5*intr_qr)

    df.loc[df[x] < min,x] = np.nan
    df.loc[df[x] > max,x] = np.nan
```

Figure 3.6: Outlier removal code snippet.

The results after executing the method shown in the following figure:

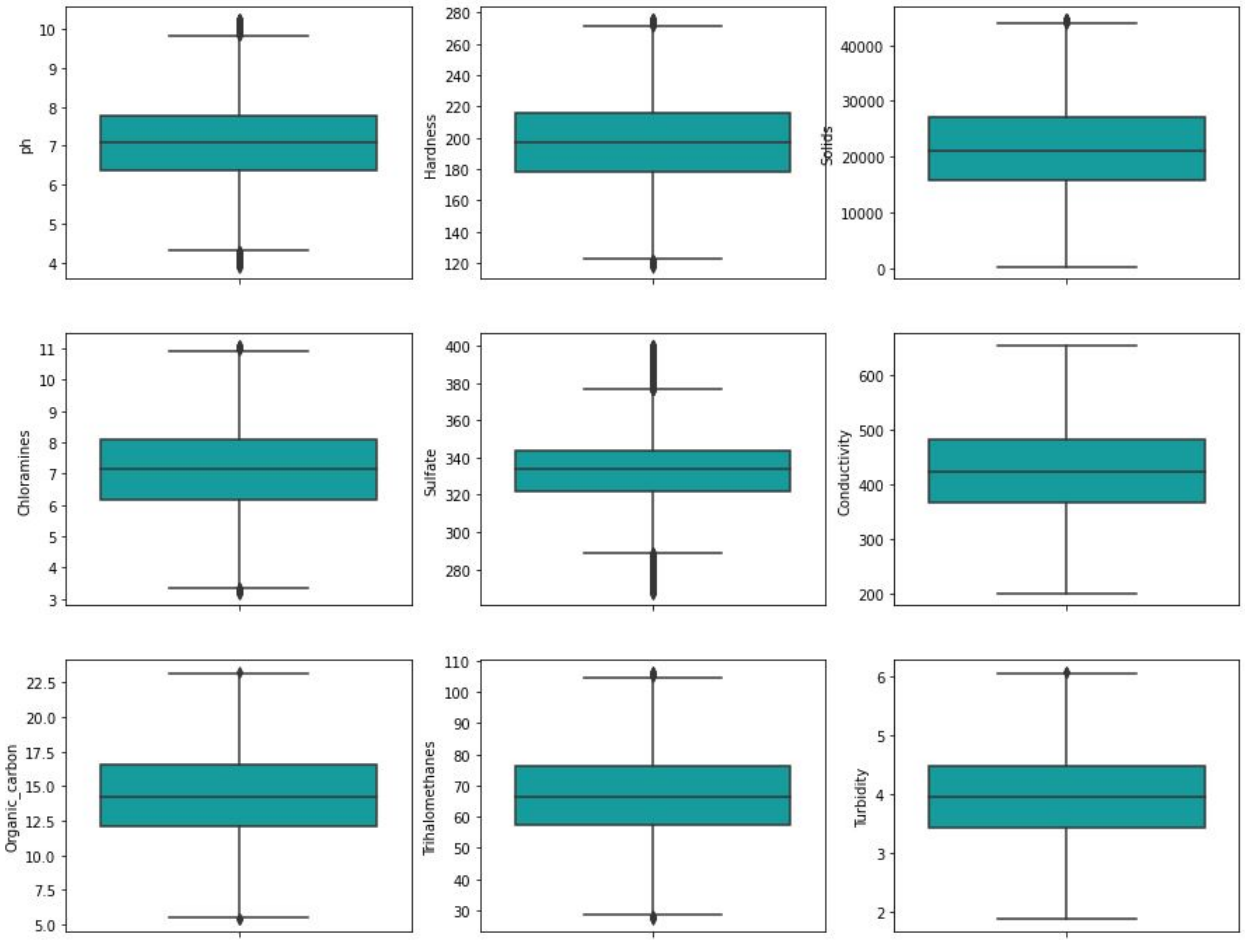

Figure 3.7: Boxplot after removing outliers.

3.5.3 Handling imbalanced data

The dataset we chose contains 3276 samples from which 1278 are potable and 1998 are not potable.

The dataset has imbalanced number of samples from each type, which can affect the performance of our model.

	Potable	**Not Potable**	**Total**
Samples	1278	1998	3276

Table 3.1: Potability classes count.

Figure (3.8) shows the percentage of potable and not potable water samples in our dataset, where 39% are potable and 61% are not potable.

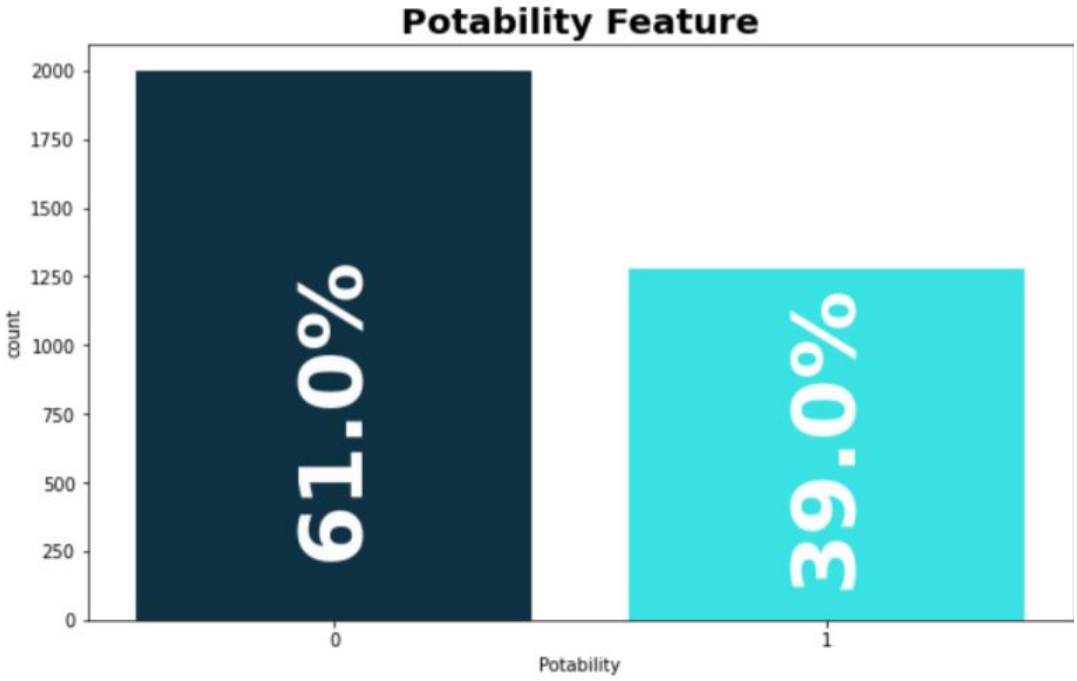

Figure 3.8: Potability classes percentage.

To overcome this problem, we used resample() method from sklearn.utils to up-sample the minority class (potable water class) to have an equal number of samples as not potable water class.

```
notpotable = df[df['Potability'] == 0]
potable = df[df['Potability'] == 1]
max_len = len(notpotable)

df_minority_upsampled = resample(potable, replace=True, n_samples=max_len)

df = pd.concat([notpotable, df_minority_upsampled])
df = shuffle(df)
```

Figure 3.9: Up-sampling script.

After running this previous script, the data becomes evenly distributed with 1998 sample from each class as shown in table 3.2¡.

	Potable	**Not Potable**	**Total**
Samples	1998	1998	3996

Table 3.2: Number of samples after up-sampling.

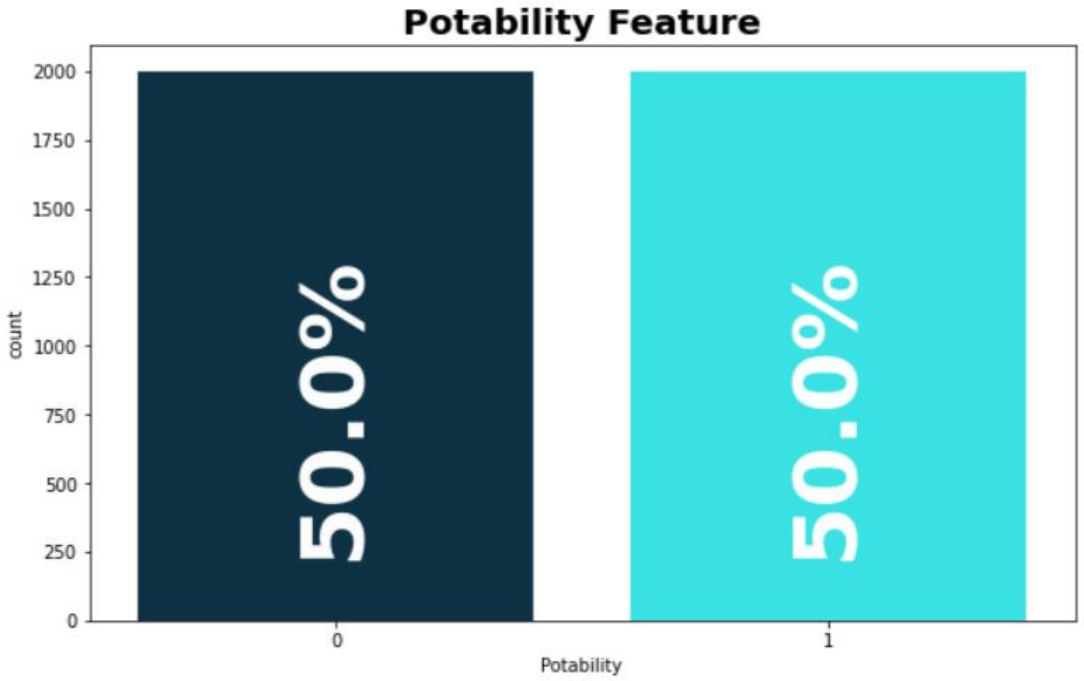

Figure 3.10: Potability classes percentage after up-sampling.

3.5.4 Feature scaling

Our dataset has multiple parameters that differs vastly in their range of values, and this vast difference can make some parameters with significant numbers to have more influence than the rest parameters with smaller numbers, To solve this issue we used StandardScaler function from the Sklearn library which is used on normally distributed data to standardizes each feature by subtracting the mean and then scaling to unit variance.

```
scale = StandardScaler()
X_train=scale.fit_transform(X_train)
X_test=scale.transform(X_test)
```

Figure 3.11: StandardScaler script.

3.5.5 Splitting Dataset

We divide our dataset into a training set and testing set:

- **Training set:** 70% are used in the training phase.
- **Testing set:** 30% are used in the testing phase.

	Potable	**Not Potable**	**Total**
Training	1399	1398	2797
Testing	599	600	1199
Total	1998	1998	3996

Table 3.3: Dataset representation after splitting.

3.6 Classification performance metrics

In order to evaluate our trained model performance, We used Accuracy, Precision, Recall and F1-Score These evaluation metrics are totally compatible with binary classification as in our case for which we chose them.

To measure performance, We used the confusion matrix where we count four kinds of results:

- **True Positive (TP):** Prediction is positive and the water is potable.
- **False Positive (FP):** Prediction is positive but the water is not potable.
- **True Negative (TN):** Prediction is negative and the water is not potable.
- **False Negative (FN):** Prediction is negative but the water is potable

All used metrics are explained below:

- **Accuracy:**

  Accuracy is the ratio of true positives and true negatives to the total observations [52].

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{3.1}$$

- **Precision:**

  Precision is the ratio of true positives observations to the total positives observations true and false [52].

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

- **Recall:**

  Recall is the ratio of correctly predicted positive observations to the all observations in actual class [52].

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

- **F1-Score:**

  F1 Score is the weighted average of Precision and Recall [52].

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3.4}$$

3.7 Conclusion

In this chapter, we defined the dataset we used. Also, we presented the architecture of our proposed approach and some steps followed during the preprocessing phase.

In the next chapter, we will discuss the obtained results.

Chapter 4

Implementation and Evaluation

4.1 Introduction

After presenting our approach proposed in the previous chapter. We will talk in this chapter about the different libraries, environments and programming language that used to implement our model, then we will present the results, and finally, we will compare the result obtained by our model with the result of another existing project.

4.2 Programming environment and tools

In this section, we will present the different libraries, environments and programming language used to implement our model.

4.2.1 Programming language

Python

Python is an easy to learn, powerful programming language. It has efficient high-level data structures and a simple but effective approach to object-oriented programming. Python's elegant syntax and dynamic typing, together with its interpreted nature, make it an ideal language for scripting and rapid application development in many areas on most platforms. Python offers several libraries (packages) for data processing, matrix calculations, analysis and data visualization [53].

Advantages Of Python

- It's Free.
- It has Lots of Libraries.

- It is easy to learn, read, understand, use and write.
- It works on all major operating systems and computer platforms.

4.2.2 Developing environment

Google Colab

Google Colab is a product of Google Research. Colab allows anyone to write and run Python code of their choice through the browser. It is an environment particularly suited to the machine learning, data analysis and education. In more technical terms, Colab is a hosted Jupyter notebook service that requires no configuration and allows free access to computing resources, including GPUs [54].

Benefits of Colab

- Python 2.7 and Python 3.6 support.
- Free GPU acceleration.
- Pre-installed libraries: All major Python libraries like TensorFlow, Scikit-learn, Matplotlib among many others are pre-installed and ready to be imported.
- Built on top of Jupyter Notebook.
- Collaboration feature (works with a team): Google Colab allows developers to use and share Jupyter notebook among each other without having to download, install, or run anything other than a browser.
- Google Colab notebooks are stored on the drive [55].

4.2.3 Used libraries

Pandas

pandas is a fast, powerful, flexible and easy to use open source data analysis and manipulation tool, built on top of the Python programming language [56].

NumPy

NumPy is the fundamental package for scientific computing in Python. It is a Python library that provides a multidimensional array object, and an assortment of routines for

fast operations on arrays, including mathematical, logical, sorting, selecting, and much more [57].

Matplotlib

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python [58].

Missingno

Missingno is a Python library that provides the ability to understand the distribution of missing values through information visualization. Visualizations can take the form of heatmaps or bar charts. Using this library, we can observe where missing values appear and examine the correlation of columns containing missing values with the target column. Missing values are better handled once the dataset is fully explored [59].

Scikit-learn

Scikit-learn is an open source machine learning library that supports supervised and unsupervised learning. It also provides various tools for model fitting, data preprocessing, model selection, model evaluation, and many other utilities [60].

4.3 Results

We used three supervised learning algorithms KNN, DT and RF. All these algorithms are detailed in chapter 2.
The results and the parameters used in each algorithm are shown in the following figures:

- **KNN:**

  The best parameters for KNN algorithm are shown in the figure below:

```
KNeighborsClassifier(metric='manhattan', n_neighbors=33, weights='distance')
```

Figure 4.1: Best Parameters of KNN algorithm.

	precision	recall	f1-score	support
0	0.77	0.62	0.69	600
1	0.68	0.82	0.74	599
accuracy			0.72	1199
macro avg	0.73	0.72	0.72	1199
weighted avg	0.73	0.72	0.72	1199

	Predicted Negative	Predicted Positive
Actual Negative	371	229
Actual Positive	108	491

Figure 4.2: KNN algorithm performance.

From the figure (4.2), we notice that the KNN algorithm predicted 371 true negatives, 108 false negatives, 229 false positives and 491 true positives. As well as the precision was 73%, recall was 72%, f1-score was 72% and the accuracy was 72%.

- **DT:**

  The best parameters for DT algorithm is shown in the figure below:

```
DecisionTreeClassifier(criterion='entropy', max_depth=42, max_features='log2')
```

Figure 4.3: Best Parameters of DT algorithm.

	precision	recall	f1-score	support
0	0.84	0.78	0.80	600
1	0.79	0.85	0.82	599
accuracy			0.81	1199
macro avg	0.81	0.81	0.81	1199
weighted avg	0.81	0.81	0.81	1199

	Predicted Negative	Predicted Positive
Actual Negative	465	135
Actual Positive	91	508

Figure 4.4: DT algorithm performance.

From the figure (4.4), we notice that the DT algorithm predicted 465 true negatives, 91 false negatives, 135 false positives and 508 true positives. As well as the precision

was 81%, recall was 81%, f1-score was 81% and the accuracy was 81%.

- **RF:**

  The best parameters for RF algorithm is shown in the figure below:

```
RandomForestClassifier(criterion='entropy', max_depth=14, max_features='sqrt',
                       n_estimators=850)
```

Figure 4.5: Best Parameters of RF algorithm.

	precision	recall	f1-score	support
0	0.91	0.88	0.89	600
1	0.88	0.91	0.90	599
accuracy			0.90	1199
macro avg	0.90	0.90	0.90	1199
weighted avg	0.90	0.90	0.90	1199

	Predicted Negative	Predicted Positive
Actual Negative	528	72
Actual Positive	52	547

Figure 4.6: RF algorithm performance.

From the figure (4.6), we notice that the RF algorithm predicted 528 true negatives, 52 false negatives, 72 false positives and 547 true positives. As well as the precision was 90%, recall was 90%, f1-score was 90% and the accuracy was 90%.

4.4 Comparison of Results

Model	Precision	Recall	F1-Score	Accuracy
KNN	73%	72%	72%	72%
DT	81%	81%	81%	81%
RF	90%	90%	90%	90%

Table 4.1: Performance evaluation results.

According to the results above we see that the RF algorithm has the highest Accuracy with 90% followed by DT with 81% and KNN with 72%.

We conclude that using RF was more efficient than KNN, and DT.

4.5 Comparison with other work

The figure below shows the comparison between our model and the existing model in [11].

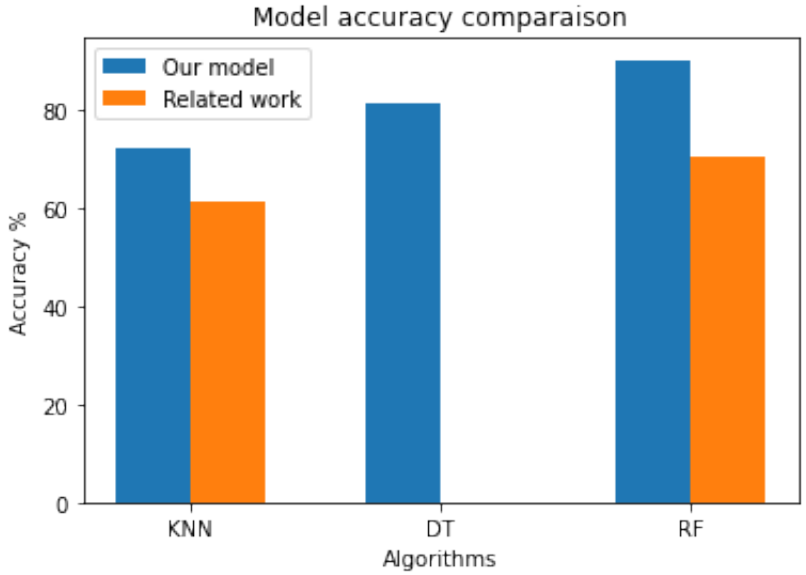

Figure 4.7: Comparison between our model and the related work.

It is visible that our model is better than the related workl.

4.6 Conclusion

In this last chapter, we started by listing the different environments and development tools used for the implementation of our proposed approach, Then we presented the results obtained.

Conclusion and Perspectives

Artificial Intelligence brings value to the Internet of Things through Machine Learning capabilities that can be used to transform proving IoT data into useful information for better decision making. While IoT adds value to AI through connectivity and data exchange.

In this project, we demonstrate how machine learning can be used into the Internet of Things. We have shown how to process huge amounts of information proving from multiple sensors by building a learning model through training data and a machine learning algorithm.
In this work, we have designed and implemented a water quality prediction system, using a dataset containing information, and the KNN, DT and RF algorithm.
We found that using RF was more efficient than KNN, and DT, as we got the highest accuracy **(90%)** with the RF algorithm.

This project was the subject of an interesting experience, very beneficial for us because we enriched our theoretical and practical knowledge.
Perspective and future works:

It is known that there is no perfect work, so it is our work. In the close future we are looking forward to make our models even more accurate by using more data and other models as deep learning models and make them applicable in the real world hoping to help people make sure that the water that they are drinking is potable without the need for laboratories and make it accessible.

REFERENCES

[1] K. K. Patel, S. M. Patel, and P. Scholar, "Internet of things-iot: definition, characteristics, architecture, enabling technologies, application & future challenges," *International journal of engineering science and computing*, vol. 6, no. 5, 2016.

[2] R. Mahmoud, T. Yousuf, F. Aloul, and I. Zualkernan, "Internet of things (iot) security: Current status, challenges and prospective measures," in *2015 10th international conference for internet technology and secured transactions (ICITST)*. IEEE, 2015, pp. 336–341.

[3] "GURU99", *"Supervised vs Unsupervised Learning: Key Differences"*, https://www.guru99.com/supervised-vs-unsupervised-learning.html.

[4] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC medical informatics and decision making*, vol. 19, no. 1, pp. 1–16, 2019.

[5] V. Nasteski, "An overview of the supervised machine learning methods," *Horizons. b, vol. 4, p. 51-62*, 2017.

[6] N. S. Chauhan, "Decision tree algorithm, explained," https://www.kdnuggets.com/2020/01/decision-tree-algorithm-explained.html.

[7] O. Mbaabu, "Introduction to random forest in machine learning," https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/.

[8] B. Mahesh, "Machine Learning Algorithms - A Review." *International Journal of Science and Research (IJSR)*, 2020.

[9] Y. Ding and S. Zhu, "Malware detection based on deep learning algorithm." *Neural Comput Applic*, 2017.

[10] X. Hao, G. Zhang, and S. Ma, "Deep learning," *International Journal of Semantic Computing*, vol. 10, no. 03, pp. 417–439, 2016.

[11] D. Poudel, D. Shrestha, S. Bhattarai, and A. Ghimire, "Comparison of machine learning algorithms in statistically imputed water potability dataset," February 2022, DOI:10.13140/RG.2.2.25767.21925.

[12] O. Vermesan, P. Friess *et al.*, *Internet of things-from research and innovation to market deployment*. River publishers Aalborg, 2014, vol. 29.

[13] X. Xingmei, Z. Jing, and W. He, "Research on the basic characteristics, the key technologies, the network architecture and security problems of the internet of things," in *Proceedings of 2013 3rd International Conference on Computer Science and Network Technology*. IEEE, 2013, pp. 825–828.

[14] A. R. Arko, S. H. Khan, A. Preety, and M. H. Biswas, "Anomaly detection in iot using machine learning algorithms," Ph.D. dissertation, Brac University, 2019.

[15] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future internet: the internet of things architecture, possible applications and key challenges," in *2012 10th international conference on frontiers of information technology*. IEEE, 2012, pp. 257–260.

[16] S. Tayyaba, S. A. Khan, M. W. Ashraf, and V. E. Balas, "Home automation using iot," in *Recent trends and advances in artificial intelligence and internet of things*. Springer, 2020, pp. 343–388.

[17] R. Sheldon, "What is a sensor?" https://www.techtarget.com/whatis/definition/sensor.

[18] A. Sulastri, "Internet of things technology development," *ITEJ (Information Technology Engineering Journals)*, vol. 4, no. 1, pp. 52–66, 2019.

[19] N. Vijayakumar, Ramya, and R, "The real time monitoring of water quality in iot environment," in *2015 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. IEEE, 2015, pp. 1–5.

[20] D. Sehrawat and N. S. Gill, "Smart sensors: Analysis of different types of iot sensors," in *2019 3rd International Conference on Trends in Electronics and Informatics (ICOEI)*. IEEE, 2019, pp. 523–528.

[21] A. Najah, A. El-Shafie, O. A. Karim, and A. H. El-Shafie, "Application of artificial neural networks for water quality prediction," *Neural Computing and Applications*, vol. 22, no. 1, pp. 187–201, 2013.

[22] H. M. Mustafa, A. Mustapha, G. Hayder, and A. Salisu, "Applications of iot and artificial intelligence in water quality monitoring and prediction: A review," in *2021 6th International Conference on Inventive Computation Technologies (ICICT)*. IEEE, 2021, pp. 968–975.

[23] M. Meybeck, N. E. Peters, and D. V. Chapman, "91: Water quality," 2005.

[24] E. M. Dogo, N. I. Nwulu, B. Twala, and C. Aigbavboa, "A survey of machine learning methods applied to anomaly detection on drinking-water quality data," *Urban Water Journal*, vol. 16, no. 3, pp. 235–248, 2019.

[25] J. O. Ighalo, A. G. Adeniyi, and G. Marques, "Internet of things for water quality monitoring and assessment: a comprehensive review," *Artificial intelligence for sustainable development: theory, practice and future applications*, pp. 245–259, 2021.

[26] K. Keshari, "Artificial intelligence tutorial : All you need to know about ai," https://www.edureka.co/blog/artificial-intelligence-tutorial/.

[27] Z. Lateef, "Types of artificial intelligence you should know," https://www.edureka.co/blog/types-of-artificial-intelligence/.

[28] S. Vieira, W. H. L. Pinaya, and A. Mechelli, "Introduction to machine learning," in *Machine learning*. Elsevier, 2020, pp. 1–20.

[29] E. G. Learned-Miller, "Introduction to supervised learning," *I: Department of Computer Science, University of Massachusetts*, p. 3, 2014.

[30] Y. Baştanlar and M. Özuysal, "Introduction to machine learning," *miRNomics: MicroRNA biology and computational analysis*, pp. 105–128, 2014.

[31] V. E. Balas, R. Kumar, and R. Srivastava, *Recent trends and advances in artificial intelligence and internet of things*. Springer, 2020.

[32] T. O. Ayodele, "Types of machine learning algorithms," *New advances in machine learning*, vol. 3, pp. 19–48, 2010.

[33] Y. Benzaki, "Introduction à l'algorithme k nearst neighbors (k-nn)," https://mrmint.fr/introduction-k-nearest-neighbors.

[34] O. Harrison, "Machine learning basics with the k-nearest neighbors algorithm," https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm-6a6e71d01761.

[35] "Decision trees," https://scikit-learn.org/stable/modules/tree.html#decision-trees.

[36] "Decision tree classification algorithm," https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm.

[37] S. Uddin, A. Khan, M. E. Hossain, and M. A. Moni, "Comparing different supervised machine learning algorithms for disease prediction," *BMC medical informatics and decision making*, vol. 19, no. 1, pp. 1–16, 2019.

[38] H. Mahmudul, I. Md Milon, I. Z. Md Ishrak, and H. M.M.A, "Attack and anomaly detection in IoT sensors in IoT sites using machine learning approaches." *Internet of Things 7*, 2019.

[39] S. E. R, "Understanding random forest," https://www.analyticsvidhya.com/blog/2021/06/understanding-random-forest/.

[40] D. Yu and L. Deng, "Deep Learning: Methods and Applications." 2014.

[41] X.-W. CHEN and X. LIN, "Big Data Deep Learning: Challenges and Perspectives." *IEEE access*, 2014.

[42] https://www.kaggle.com/datasets/adityakadiwal/water-potability?fbclid=IwAR34lIsRJ9Qu-lfUZvMRYEjgMbNq1PghF-BsmgvSv3WNoObkQboEPYH6S8c.

[43] "pH in Drinking-water,," *World Health Organization*, 2007.

[44] "Hardness in drinking water," https://www.healthvermont.gov/environment/drinking-water/hardness-drinking-water.

[45] "Guidelines for canadian drinking water quality: Guideline technical document – total dissolved solids (tds)," https://www.canada.ca/en/health-canada/services/publications/healthy-living/guidelines-canadian-drinking-water-quality-guideline-technical-document-total-dissolved-solids-td
html.

[46] "Basic information about chloramines and drinking water disinfection," https://www.epa.gov/dwreginfo/basic-information-about-chloramines-and-drinking-water-disinfection#:~:text=Chloramines.

[47] R. Sheldon, "Sulfate in drinking water," https://archive.epa.gov/water/archive/web/html/sulfate.html.

[48] "Indicators: Conductivity," https://www.epa.gov/national-aquatic-resource-surveys/indicators-conductivity.

[49] "Total organic carbon (toc)," https://maineenvironmentallaboratory.com/?p=1095.

[50] "Tthm in drinking water: Information for consumers, howpublished= https://www.mass.gov/service-details/tthm-in-drinking-water-information-for-consumers,."

[51] "Turbidity and water, howpublished= https://www.usgs.gov/special-topics/water-science-school/science/turbidity-and-water,."

[52] H. N. B, "Confusion matrix, accuracy, precision, recall, f1 score," https://medium.com/analytics-vidhya/confusion-matrix-accuracy-precision-recall-f1-score-ade299cf63cd.

[53] "The python tutorial," https://docs.python.org/3/tutorial/.

[54] https://research.google.com/colaboratory/faq.html.

[55] V. Lall, "Google colab the beginner's guide," https://medium.com/lean-in-women-in-tech-india/google-colab-the-beginners-guide-5ad3b417dfa.

[56] https://pandas.pydata.org/.

[57] "Numpy documentationl," https://numpy.org/doc/stable/.

[58] "Matplotlib: Visualization with python," https://matplotlib.org/.

[59] B. Madhukar, "Tutorial on missingno python tool to visualize missing values," https://analyticsindiamag.com/tutorial-on-missingno-python-tool-to-visualize-missing-values/.

[60] https://scikit-learn.org/stable/getting_started.html.