الجمهورية الجزائرية الديمقراطية الشعبية

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE YAHIA FARES DE MEDEA

FACULTE DE TECHNOLOGIE

Département de Génie Électrique

Mémoire de Fin d'Études de Master

Filière : Électronique

Spécialité : Instrumentation

Thème

Système Connecté d'Analyse Intelligente pour le Contrôle de la Qualité de l'Eau Potable

Préparé et présenté par,

Asma BOUCHERIT

Omar BENAMEUR

Proposé et Dirigé par,

Prof. Mounir BOUHEDDA

Dr. Hamza BENYEZZA

Promotion 2024

Travail préparé au sein du Laboratoire des Systèmes Électroniques Avancés (LSEA) de l'Université de Médéa, ALGÉRIE

Dédicaces

Au nom de Dieu, le Clément, le Miséricordieux.

Louange à Dieu, Seigneur des Mondes, et que la paix et la bénédiction soient sur le plus noble des prophètes et messagers, notre maître Mohammed, ainsi que sur sa famille et tous ses compagnons.

À ceux qui ont été mon soutien et mon appui à toutes les étapes de ma vie, à ceux qui ont consenti de nombreux efforts et sacrifices pour ma réussite et mon bonheur, à mes chers parents.

À ceux qui ne m'ont jamais privé de leurs conseils et de leurs orientations, à mes chers frères Abd Rahman, Youcef, Ibrahim, ma sœur Fatima Zahra,et les enfants de mes frères Ishaq, Israa, Mohamed, Mayssaa, Iyad, Djomana.

À mes honorables professeurs, qui n'ont ménagé aucun effort pour nous enseigner et nous guider, et qui ont été pour nous un exemple à suivre en matière de savoir et de morale.

À mes amis et collègues, qui ont partagé avec moi des moments de joie et de défis, et qui ont toujours été à mes côtés par leur soutien et leur assistance : Lina, Rofaida, Hadil, Nadjet, Khalida, et mon binôme Omar.

À tous ceux qui ont contribué, ne serait-ce que par un mot gentil ou un sourire, à soutenir mon parcours académique.

À vous tous, je dédie ce modeste travail, en espérant qu'il trouvera grâce à vos yeux et qu'il soit le début d'un avenir radieux.

Et louange à Dieu, par Sa grâce les bonnes actions s'accomplissent.

Asma Boucherit

Dédicaces

Avant tout, je tiens à remercier le Bon Dieu, l'Unique, qui m'a offert le courage et la volonté nécessaires pour affronter les différentes difficultés de la vie.

Ce travail, et bien au-delà, je le dois à mes très chers parents, qui m'ont fourni quotidiennement un soutien et une confiance sans faille. Je ne saurais exprimer ma gratitude uniquement par des mots.

À mon père : Mohamed et ma mère : Fatiha
À mes frères : Mokhtar, Djamel, Alaa,Hako et Hamada
À mes sœurs : Ouardia et Nacira
À mes amis et mes collègue : Nacer Eddine, Farid, Choaïb et Baghdad

En particulier, à ma collègue avant d'être un binôme, Boucherit Asmaa, qui n'a jamais manqué de travailler avec dévouement.

À notre encadreur, Pr. Bouhedda Mounir, et notre co-encadreur, Dr. Benyezza Hamza, pour leur immense patience, leur grande disponibilité et leurs conseils qui ont grandement contribué à la réalisation de ce travail.

Omar Benameur

Remerciements

Nous tenons à exprimer notre profonde gratitude à Dieu Tout-Puissant, qui nous a accordé la force et le courage nécessaires pour mener ce travail à terme.

Nous souhaitons exprimer nos plus sincères remerciements à notre encadreur, le Professeur Mounir BOUHEDDA de l'Université YAHIA FARES, pour sa disponibilité, sa lecture attentive, ses suggestions et remarques précieuses, et surtout pour la confiance illimitée qu'il a placée en nous tout au long de ce projet. Nous lui témoignons notre reconnaissance et notre respect les plus sincères.

Nous exprimons également notre gratitude au Dr. Hamza BENYEZZA pour ses orientations et précieuses aides.

Nos remerciements vont également à toute l'équipe pédagogique du département de génie électrique de l'Université de Médéa, dont l'efficacité nous a permis d'apprendre énormément de choses durant nos cinq années de formation.

Nous tenons à remercier particulièrement nos familles pour leur soutien moral et leurs encouragements tout au long de ce travail.

Nous adressons également nos remerciements à nos ami(e)s et collègues de classe, qui nous ont soutenus et aidés tout au long de ce projet

Enfin, nous remercions les membres du jury, notamment le Président, Dr. BETTACHE Farouk, et l'Examinateur, M. BAILICHE Amine, pour l'honneur qu'ils nous font en acceptant de juger ce modeste travail

نظام تحليل ذكي متصل لمراقبة جودة مياه الشرب

ملخص — يقدم هذا البحث نظامًا ذكيًا لمراقبة جودة مياه الشرب يستغل إنترنت الأشياء والذكاء الاصطناعي. تحلل خوارزميات التعلم الآلي مثل الشبكات العصبية، وأقرب الجيران، والانحدار اللوجستي، وغابات الاقتراحات العشوائية بيانات من أجهزة استشعار متقدمة لإجراء مراقبة دقيقة. متصلاً بإنترنت الأشياء، يكتشف هذا النظام المتطور للذكاء الاصطناعي أي خلل بسرعة وينقل البيانات للتدخل الفوري، مما يحمي بذلك الصحة العامة. يتم تفصيل تطويره الذي يجمع بين الأجهزة والبرامج وتقنيات تحليل البيانات والتحقق من صحته من خلال اختبارات صارمة.

الكلمات الرئيسية - أجهزة استشعار متقدمة، ذكاء اصطناعي، **Node-RED** ، إنترنت الأشياء، مراقبة في الوقت الفعلي.

Connected Intelligent Analysis System for Drinking Water Quality Control

Abstract–This thesis presents an intelligent drinking water quality monitoring system exploiting the Internet of Things and artificial intelligence. Machine learning algorithms such as neural networks, k-nearest neighbors, logistic regression, and random forests analyze data from advanced sensors for accurate monitoring. Connected to the IoT, this cutting-edge AI system rapidly detects anomalies and transmits data for immediate intervention, thereby protecting public health. Its development combining hardware, software, and data analysis techniques is detailed and validated through rigorous testing.

Keywords–Advanced sensors, Artificial intelligence, Internet of Things (IoT), Node-RED, Real-time monitoring.

Système Connecté d'Analyse Intelligente pour le Contrôle de la Qualité de l'Eau Potable

Résumé–Ce mémoire présente un système intelligent de contrôle de la qualité de l'eau potable exploitant l'Internet des Objets et l'intelligence artificielle. Des algorithmes d'apprentissage automatique comme les réseaux de neurones, les k-plus proches voisins, la régression logistique et les forêts aléatoires analysent les données de capteurs avancés pour une surveillance précise. Connecté à l'IoT, ce système à la pointe de l'IA détecte rapidement les anomalies et transmet les données pour une intervention immédiate, protégeant ainsi la santé publique. Son développement combinant matériel, logiciel et techniques d'analyse de données est détaillé et validé par des tests.

Mots clés : Capteurs avancés, Intelligence artificielle, Internet des Objets (IoT), Surveillance en temps réel, Node-RED

Sommaire

Dédicaces

Remerciements

Résumé

Sommaire

Liste de Figures

Liste des Tableaux

Liste des abréviations

Introduction Générale 1

CHAPITRE I: Étude générale sur la qualité d'eau potable

I. 1. Introduction 2

I. 2. Composition d'une eau naturelle 2

I. 3. Propriétés physico-chimiques 3

I. 3. 1. Propriétés Physiques 3

I. 3. 1. 1. Masse volumique 3

I. 3. 1. 2. Propriétés oxyde-réductrices 3

I. 3. 1. 3. Conductivité électrique de l'eau 3

I. 3. 1. 4. Propriétés Optiques 3

I. 3. 2. Propriétés Chimiques 4

I. 4. Cycle de l'eau 4

I. 4. 1.Evaporation 4

I. 4. 2.Condensation 4

I. 4. 3.Précipitations 5

I. 4. 4.Infiltration 5

I. 5. Eau alimentaire 5

I. 5. 1.Eaux de surface 5

I. 5. 2.Eaux souterraines 5

I. 5. 3.Eau de sources 6

I. 5. 4.Eau de puits 6

I .6. Méthodes classiques de contrôle qualité de l'eau potable 6

I.6.1 Échantillonnage des eaux 6

I.6.2Analyses 7

I.6.3 Éléments d'interprétation 7

I.6.3.1 Eaux souterraine 7

I.6.3.2 Eaux de surface *8*
I. 7. Analyse de l'eau potable **8**
I. 7. 1. Paramètres physiques 8
I. 7. 1. 1. Température *8*
I. 7. 1. 2. Potentiel d'hydrogène (pH) *8*
I. 7. 1. 3. Turbidité *8*
I. 7. 1. 4. Conductivité *8*
I. 7. 1. 5. Dureté *9*
I. 7. 1. 5. Solides (total des solides dissous-TDS) *9*
I. 7. 1. 6. Chloramines : *9*
I. 7. 1. 7. Sulfate : *9*
I. 7. 1. 8. Carbone organique : *10*
I. 7. 1. 9. Trihalométhanes *10*
I. 7. 2. Paramètres organoleptiques 10
I. 7. 2. 1. Couleur *10*
I. 7. 2. 2. Odeur *10*
I. 7. 2. 3. Goût et saveur *10*
I. 7. 3. Paramètres microbiologiques 10
I. 7. 3. 1. Germes totaux *12*
I. 7. 3. 2. Coliformes totaux *12*
I. 7. 3. 3. Coliformes fécaux *13*
I. 7 .3. 4. Escherichia coli *13*
I. 7. 3. 5. Streptocoques fécaux *14*
I. 7. 3. 6. Spores des bactéries anaérobies sulfite-réductrice *14*
I. 8. Contamination de l'eau **14**
I.9. Conclusion **15**

CHAPITRE II: Outils Software et Hardware utilisés

II.1 Introduction **16**
II.2 Outils Software **16**
II.2.1 Méthodes de machines Learning 16
II.2.1.1 Réseaux de neurones artificiels *16*
II.2.1.2 K plus proches voisins (KNN) *21*
II.2.1.3 Régression logistique (RL) 23
II.2.1.3.1 Principe de Régression logistique *24*
II.2.1.3. Décision *24*

II.2.1.3.3Fonction du coût *24*
II.2.1.3.4Gradients *25*
II.2.2 Node-Red ____ 29
II.2.2.1 Définition *29*
II.2.2.1Concepts Fondamentaux d'un Node-Red *29*
II.2.2.3 Création de Flows avec Node-RED *29*
II.2.2.4 Personnalisation et Extensibilité de Node-RED *29*
II.2.2.5 Déploiement et Gestion des Flows *29*
II.2.2.6 Créer un Dashboard *29*
II.2.3 Bibliothèques python utiles ____ 31
II.2.3.1 Pandas *31*
II.2.3.2 NumPy *31*
II.2.3.3 Sklearn *31*
II.2.3.4 Tensorflow *32*
II.3 Outils Hardware **32**
II.3.1 Arduino ____ 32
II.3.1.1 Caractéristiques principales *32*
II.3.1.2Environnement de la programmation *33*
II.3.1.3 Structure générale du programme (IDE Arduino) : *33*
II.3.2 Capteurs utilisés ____ 34
II.3.2.1 Capteur de PH *34*
II.3.2.2 Capteur de turbidité *35*
III.3.2.2.1Comprendre la turbidité *35*
III.3.2.2.2 Impact de la turbidité *36*
II.3.2.3 Capteur de conductivité électrique *37*
II.4 Conclusion **37**

CHAPITRE III: Réalisation pratique et déploiement

III.1 Introduction **37**
III.2 Cahier des charges **37**
III.2.1 Collecte De Données ____ 37
III.2.2 Transferts de données ____ 37
III.2.3 Traitement des données ____ 37
III.2.4 Interfaçage des données ____ 38
III.3 Système proposé : **38**
III.4 Interfaçage des capteurs **39**

III.4.1 pH-mètres 39
III.4.1.1 Composants requis ... *39*
III.4.1.2 Analyse du fonctionnement du capteur de pH analogique par gravité ... *39*
III.4.1.3 Interface Arduino avec le diagramme du capteur PH et son fonctionnement ... *39*
III.4.1.4Étalonnage de l'électrode de pH ... *40*
III.4.1.5 Programmation Arduino pour pH-mètre ... *40*
III.4.1.6 Test du dispositif de mesure du pH ... *40*
III.4.2.3 Connexion du capteur de turbidité avec Arduino ... *42*
III.4.2.4Code d'interface Arduino du capteur de turbidité de base ... *43*
III.4.2.6 Résultats et observations ... *43*
III.4.3 Capteur TDS et interface Arduino 45
III.4.3.1 Interface du capteur TDS analogique de gravité avec Arduino ... *45*
III.4.3.2 Étalonnage de l'électrode de TDS ... *46*
III.4.3.3 Code Arduino ... *46*
III.4.3.4 Réalisation réelle du capteur TDS ... *47*
III.4.3.5 Comparaison entre nos résultats précédents ... *48*
III.5 Réalisation ... **48**
III.5.1 Schéma synoptique global de système 48
III.5.2 Connexion de l'ensemble des capteurs à Arduino ... *49*
III.5.3 Code Arduino ... *49*
III.5.4Prototype du test réalisé ... *50*
III.5.5MESURES ET TESTS ... *51*
III.5.5.2Deuxième test ... *52*

III.6 Identification de la qualité d'eau par intelligence artificielle ... **52**
III.6.1 Base de données utilisée 52
III.6.2 Analyse des corrélations entre les paramètres de qualité de l'eau 53
III.6.3 Appel aux fichiers Excel dans un projet Python 54
III.6.4 Modèles de Machine Learning 55
III.6.4.1 Modèle ANN ... *55*
III.6.4.2 Classificateur LR ... *59*
III.6.4.3 Classificateur RF ... *61*
III.6.4.4Classificateur K-NN ... *63*
III.6.4.5Comparaison entre les quatre classificateurs ... *64*
III.7Node-RED ... **66**
III.7.1 Lancement sous Windows 66

III.7.2 Installation de Node-RED 66
III.7.3 Génération d'alerte 69
III.8 Utilisation des requêtes pour communiquer avec l'API HTTP de Node-RED depuis Python 70
III.9 Conclusion 71
Conclusion générale 72
Références bibliogrpahiques 73

Liste des Figures

N°	Titre	Page
Figure I.1	Cycle biogéochimique de l'eau	5
Figure I.2	Exemple d'appareil de filtration	7
Figure I.3	Exemple d'un kit d'analyse d'eau potable	7
Figure I.4	Groupes d'indicateurs bactériens	12
Figure II.1	Schéma simple d'un réseau de neurone	17
Figure II.2	Modèle KNN	22
Figure II.3	Valeurs de la fonction logistique sont comprises entre 0 et 1	24
Figure II.4	Exemple d'un arbre de décision	26
Figure II.5	Quelques nœuds du Dashboard de Node-RED	29
Figure II.6	Installation du tableau de bord	30
Figure II.7	Tableau de bord installé	30
Figure II.8	Éléments d'un tableau de bord (Dashboard)	30
Figure II.9	Carte Arduino Uno	32
Figure II.10	Composants principaux de l'Arduino	32
Figure II.11	Interface IDE d'Arduino	33
Figure II.12	Analog pH Meter Kit (SKUSEN0161)	33
Figure II.13	Capteur de turbidité	34
Figure II.14	Example de turbidité	35
Figure II.15	Capteur de conductivité	36
Figure III.1	Schéma fonctionnel du système de surveillance de la qualité de l'eau	38
Figure III.2	pH-mètre utilisant Arduino Uno	39
Figure III.3	Expérimentation de mesure de pH à base d'Arduino	40
Figure III.4	Programme Arduino pour la mesure du pH	40
Figure III.5	Expérimentation de la mesure de pH à base d'Arduino cas de l'eau potable	41
Figure III.6	Résultat de test de mesure du pH d'eau potable	41
Figure III.7	Mesure de pH d'eau citronnée	42
Figure III.8	Résultat de test de mesure du pH d'eau citronnée 76	42
Figure III.9	Utilisation de capteur de turbidité à base d'Arduino	42
Figure III.10	Programme Arduino pour la mesure de turbidité	43
Figure III.11	Réalisation expérimentale de capteur turbidité d'eau potable	43
Figure III.12	Résultats du premier test pour la mesure de turbidité à base d'Arduino	44
Figure III.13	Mesure de capteur de turbidité d'eau trouble à base d'Arduino	44
Figure III.14	Résultats du deuxième test de mesure de turbidité à base d'Arduino	45
Figure III.15	Mesure de TDS à base d'Arduino	46
Figure III.16	Programme Arduino pour la mesure de conductivité à base d'Arduino	46
Figure III.17	Réalisation expérimentale pour la mesure de conductivité cas d'eau potable	47
Figure III.18	Résultats de test pour la mesure de turbidité d'eau potable	47
Figure III.19	Mesure de turbidité à base d'Arduino en cas d'eau salée	48
Figure III.20	Résultat de mesure de turbidité d'eau salée	48
Figure III.21	Schéma synoptique global de mesure de pH, turbidité et TDS à base d'Arduino	49
Figure III.22	Code Arduino pour la mesure de pH, turbidité et conductivité	50
Figure III.23	Expérimentation pour la mesure de pH, turbidité et conductivité	50
Figure III.24	Mesure de pH, turbidité et conductivité à base d'Arduino d'une eau potable	51
Figure III.25	Résultat de mesure de pH, turbidité et conductivité à base d'Arduino d'une eau potable	51
Figure III.26	Résultats d'un deuxième test pour la mesure de pH, turbidité et conductivité à base d'Arduino	52
Figure III.27	Carte de corrélation des paramètres de qualité de l'eau	53

Figure III.28	Matrice de dispersion des paramètres de qualité de l'eau	54
Figure III.29	Programme python d'extraction de données à partir d'un fichier Excel	54
Figure III.30	Extraction de la colonne "Potabilité" d'un ensemble de données	55
Figure III.31	Ajustement des paramètres du modèle ANN	55
Figure III.32	Affichage de l'accuracy de l'ANN	56
Figure III.33	Matrice de confusion de l'apprentissage d'ANN pour un taux de test de 20%	57
Figure III.34	Matrice de confusion de l'apprentissage d'ANN pour un taux de test de 30%	57
Figure III.35	Matrice de confusion de l'apprentissage d'ANN pour un taux de test de 60%	57
Figure III.36	Mise à l'échelle de données pour le modèle RL	58
Figure III.37	Recherche par grille dans RL	58
Figure III.38	Paramètres du modèle RL 94	58
Figure III.39	Matrice de confusion de l'apprentissage RL pour un taux de test de 20%	59
Figure III.40	Matrice de confusion de l'apprentissage RL pour un taux de test de 50%	59
Figure III.41	Initialisation de RF	60
Figure III.42	Paramètres de RF	60
Figure III.43	Matrice de confusion de l'apprentissage RF pour un taux de test de 20%	61
Figure III.44	Matrice de confusion de l'apprentissage RF pour un taux de test de 30%	61
Figure III.45	Matrice de confusion de l'apprentissage RF pour un taux de test de 50%	61
Figure III.46	Création d'un Classificateur KNN	62
Figure III.47	Optimisation des hyperparamètres de KNN	62
Figure III.48	Paramètres de KNN	62
Figure III.49	Matrice de confusion de l'apprentissage KNN pour un taux de test de 20%	63
Figure III.50	Matrice de confusion de l'apprentissage KNN pour un taux de test de 20%	63
Figure III.51	Matrices de confusion des quatre classificateurs	64
Figure III.52	Distribution des précisions des quatre modèles de classification	65
Figure III.53	Lancement de Node-Red sur cmd	65
Figure III.54	Éditeur Node-Red	66
Figure III.55	Nœud serial in	66
Figure III.56	Configuration du nœud d'entrée série	66
Figure III.57	Fonction Json	67
Figure III.58	Exemple de configuration d'une jauge	67
Figure III.59	Nœuds du Dashboard	68
Figure III.60	Interface de surveillance de la qualité de l'eau	68
Figure III.61	Interface en cas d'Alerte	68
Figure III.62	Noeud "pythonshell in"	69
Figure III.63	Utilisation de python on Node Red	69
Figure III.64	Flux de traitement des paramètres de qualité de l'eau	70
Figure III.65	Test 1 de potabilité de l'eau	70
Figure III.66	Test 2 de potabilité de l'eau	71
Figure III.67	Test 3 de potabilité de l'eau	71

Liste des Tableaux

N°	Titre	Page
Tableau I.1	Principales constantes physiques de l'eau pure	7
Tableau I.2	Normes de potabilité d'une eau de consommation	11
Tableau II.1	Relation entre la sortie de l'électrode de pH et sa valeur	34
Tableau III.1	Exemple de données de potabilité de l'eau	52
Tableau III.2	Résultats des tests pour l'apprentissage à base d'ANN	56
Tableau III.3	Résultats des tests pour l'apprentissage RL	59
Tableau III.4	Résultats des tests pour les enseignantes RF	60
Tableau III.5	Résultats des tests pour l'apprentissage d'un modèle KNN	62

Liste des abréviations

ANN : Artificiel Neural Network (Réseau de neurones artificiels)

API : Automate programmable industriel

CA : Courant Alternatif

CE : Conductivité Électrique

COT : Carbone Organique Total

CTR: Click-Through Rate (Taux de clics)

E. Coli: Escherichia coli

GPG: GNU Privacy Guard

IA : Intelligence Artificielle

IDE : Integrated DevelopmentEnvironment (Environnement de développement intégré)

IOT : Internet of Things (Internet des objets)

K-NN : K-Nearest Neighbors (K plus proches voisins)

LR : logistic regression (régression logistique)

MES : Matériaux En Suspension

MLP : Multi-Layer Perceptron (Perceptron multicouche)

MON : Monochloramine

NTU : NephelometricTurbidity Unit (Unité de turbidité néphélométrique)

OMS : Organisation Mondiale de la Santé

PPM : Parts Per Million (Parties par million)

RF : Random Forest (Forêt aléatoire)

RN : Réseau de Neurones

RNA : Réseau de Neurones Artificiels (Redondant avec ANN)

TDS : Total Dissolved Solids (Solides dissous totaux)

THM : Trihalomethane

TSS : Total SuspendedSolids (Solides en suspension totaux)

TTC : ThermotolerantColiforms (Coliformes thermotolérants)

Introduction Générale

La qualité de l'eau potable représente un enjeu critique pour la santé publique et le bien-être des populations à l'échelle mondiale. Face à la croissance démographique, à l'urbanisation rapide et aux défis environnementaux actuels, garantir un accès à une eau saine et salubre est devenu une priorité absolue. Bien que performants, les systèmes traditionnels de contrôle de la qualité de l'eau peuvent s'avérer insuffisants pour répondre aux besoins croissants de surveillance en temps réel et de réactivité immédiate aux éventuelles contaminations [1].

C'est dans ce contexte que la conception de systèmes intelligents de contrôle de la qualité de l'eau basés sur l'Internet des Objets (IoT) émerge comme une solution prometteuse. Exploitant les dernières avancées technologiques, ces systèmes offrent une surveillance continue et une analyse précise des paramètres clés de l'eau potable grâce à des capteurs avancés, des microcontrôleurs performants et des algorithmes d'apprentissage automatique tels que les réseaux de neurones artificiels (ANN), les k plus proches voisins (KNN), La régression logistique (LR) et les forêts aléatoires (RF). Leur intégration à l'IoT permet une connectivité en réseau, offrant une réactivité accrue en détectant rapidement toute anomalie et en transmettant les données en temps réel pour une intervention immédiate visant à protéger la santé des consommateurs [2].

Les défis de la surveillance de la qualité de l'eau potable incluent la nécessité de détection rapide des anomalies et de réaction immédiate pour protéger la santé des consommateurs. Comment concevoir un système IoT capable de répondre efficacement à ces exigences en termes de fiabilité, précision et réactivité ?

Ce mémoire présente le processus de conception et de mise en œuvre d'un système intelligent de contrôle de qualité d'eau connecté basé sur l'IoT. La méthodologie adoptée inclut une revue des bases théoriques et des normes de qualité de l'eau, le choix et l'intégration des outils logiciels et matériels appropriés, et le développement de techniques d'apprentissage automatique pour l'analyse des données.

L'intégration de microcontrôleurs, de capteurs et de connectivité IoT a permis de développer un système de surveillance en temps réel capable de détecter rapidement toute contamination. Ce travail contribue à améliorer la sécurité de l'eau potable en proposant une solution technologique avancée et validée par des tests de performance rigoureux.

Structuré en trois chapitres principaux, ce mémoire aborde dans un premier temps les bases théoriques, les normes et critères réglementaires ainsi que les principales sources de contamination de l'eau. Le deuxième chapitre explore les outils logiciels et matériels utilisés, avec un focus sur les technologies IoT et les techniques d'apprentissage automatique. Enfin, le troisième chapitre décrit la réalisation concrète du système IoT, de l'assemblage des composants à la programmation des algorithmes, incluant des tests de performances. La conclusion synthétise les résultats obtenus, évalue l'efficacité du système et propose des perspectives de développements futurs.

Chapitre I

Étude générale sur la qualité d'eau potable

I. 1. Introduction

L'eau de boisson est l'eau utilisée pour la consommation humaine. C'est l'eau destinée à être bue pour se désaltérer. Sa principale qualité est d'être potable, c'est-à-dire exempte de germes pathogènes et d'organismes parasites [3][4].

En 2010, l'assemblée générale des Nations Unies a reconnu explicitement le droit à l'eau potable comme un droit de l'Homme. Chacun a le droit à un accès suffisant, continu, sûr, acceptable et abordable à de l'eau pour son usage personnel et domestique[5].

Il est très important pour la sante publique de disposer d'une eau salubre, que ce soit pour la boisson, pour un usage domestique, pour la production alimentaire ou pour les loisirs.

L'amélioration de l'approvisionnement et une meilleure gestion des ressources hydriques peuvent stimuler la croissance économique des pays et contribuer largement à réduire la pauvreté [3][5].

En effet l'eau est un élément clef de la vie terrestre. Elle est indispensable au fonctionnement de l'organisme. Elle représente 70% du poids corporel de l'homme adulte et 80% de celui de l'enfant. Ainsi, 2/3 de l'eau présente dans le corps humain sont repartis dans 51013 de cellules, 5% sous forme de plasma sanguin et 15% constituent le liquide interstitiel [6].

L'eau contaminée entrainent la transmission de maladies comme le choléra, la diarrhée, la dysenterie, l'hépatite A, la fièvre typhoïde et la poliomyélite.

La question de la qualité de l'eau au sein des programmes humanitaires se pose essentiellement en termes de consommation humaine et d'irrigation et la mauvaise qualité de l'eau peut être induite par des activités anthropiques ou par des phénomènes naturels. D'après l'organisation mondiale de la santé (OMS) la qualité d'une eau est définie par des paramètres physiques, chimiques et biologiques, mais aussi l'usage de cette eau.

Le principal objectif de cette étude est Contrôle qualité de l'eau de boisson, et de donner un aperçu sur les normes de qualité des eaux de consommation.

I. 2. Composition d'une eau naturelle

Dans une eau naturelle, les composants se répartissent selon leur nature chimique (organique ou minérale) ou leur état physique (matières dissoutes, colloïdales ou en suspension)[7].

- Matière minérale : Principalement des composés ioniques, tels que les anions HCO_3^-, Cl^-, SO_4^{2-}et les cations Ca^{2+}, Na^+, Mg^{2+}, K^+. Ils résultent de la dissolution des roches en contact avec l'eau. Cette dernière renferme également des gaz dissous, d'une importance majeure dans les phénomènes biologiques et chimiques, comme la corrosion.

- Matière organique : Issue de l'érosion des sols ou de la décomposition de matières animales ou végétales présentes dans l'eau. Ces substances se dégradent en raison

de leur instabilité chimique, sous l'action des microorganismes, formant des composés de plus en plus simples.

- Matières dissoutes : Des substances dispersées uniformément dans l'eau, faiblement polarisées ou ionisées. En raison de leur état dissous, ces molécules sont difficiles à éliminer par des procédés physiques classiques.

- Matières colloïdales : Elles se situent à mi-chemin entre les états dissous et en suspension. Composées de petites particules chargées négativement, elles sont difficiles à éliminer.

- Matières en suspension : Des particules solides d'une taille supérieure à 10 µm, dispersées dans l'eau sans être chimiquement liées à elle. Leur élimination est relativement simple.

I. 3. Propriétés physico-chimiques

L'eau existe sous différentes formes en fonction de son état physique. À l'état gazeux (vapeur), elle correspond principalement à la formule H_2O. En revanche, à l'état solide (glace) ou liquide, elle adopte une structure condensée plus complexe.

I. 3. 1. Propriétés Physiques

I. 3. 1. 1. Masse volumique

La masse volumique de l'eau varie selon la température et la pression. La glace a une masse volumique plus faible que l'eau liquide. La densité maximale de l'eau est atteinte à une température de 3,984°C[5]. Bien que l'eau soit souvent considérée comme un fluide incompressible, elle possède en réalité une légère élasticité : son volume diminue d'environ 0,048 % à chaque augmentation de pression d'une atmosphère[8].

I. 3. 1. 2. Propriétés oxyde-réductrices

L'eau joue un rôle crucial en tant que système oxyde-réducteur, dont la stabilité est limitée par ses réactions d'oxydation et de réduction [8]. Le pôle positif, déficient en électrons, agit comme un agent oxydant. En revanche, le pôle négatif, avec un excès d'électrons, agit comme un agent réducteur capable de céder des électrons[9][10] **.**

I. 3. 1. 3. Conductivité électrique de l'eau

L'eau présente une conductivité légère. La conductivité de l'eau la plus pure mesurée est de 4,2 microsiemens par mètre à 20 °C, équivalant à une résistivité de 23,8 mégohms centimètres. Cette conductivité augmente lors de la dissolution de sels dans l'eau et varie en fonction de la température.

I. 3. 1. 4. Propriétés Optiques

La transparence de l'eau est déterminée par la longueur d'onde de la lumière incidente. L'eau absorbe fortement les longueurs d'onde correspondant à l'orangé et au rouge dans le spectre visible, ce qui conduit à la teinte bleue de la lumière transmise en profondeur. Cette propriété de transparence est souvent exploitée pour évaluer certains types de pollution[6].

I. 3. 2. Propriétés Chimiques

Du fait de sa capacité à agir comme un excellent solvant, l'eau est facilement sujette à la pollution. Le processus de dissolution d'une substance implique la rupture de sa cohésion interne, causée par des forces interatomiques et intermoléculaires. En raison de ses propriétés électriques et de sa composition moléculaire, l'eau est particulièrement efficace pour dissoudre de nombreux gaz, liquides polaires et surtout des solides[11] (Tableau I.1) [9]

Tableau I.1: Principales constantes physiques de l'eau pure

Eau liquide	**Constante**
Température d ébullitions sous 760 mm Hg (101 325,02 Pa)	100%
Capacité thermique massique à 15°C	4,186 8 j. g^{-1}
Enthalpie de vaporisation à 100°C	2252,5 j. g^{-1}
Conductivité thermique à 20°C	5,98 mW. Cm^{-1}. K^{-1}
Résistivité à 20°C	23,8 MΩ .cm
Permittivité relative ε à 20°C	80
Indice de réfraction pour la raie D à 10°C	1,333
Masse volumique à 4°C	1g .cm^{-3} (par définition)
Eau solide	**Constante**
Température de fusion	0°C (par définition)
Capacité thermique massique	2,093 4 j. g^{-1}
Enthalpie de fusion sous 760 mm Hg (101 325 ,02Pa)	333,27 j. g^{-1}
Tension de vapeur à 0°C	877,128 Pa
Permittivité relative ε	3,26
Indice de réfaction pour la raie D	1,309 07
Densité (par rapport à l'eau à 4°C)	$0.91649 \neq 0.007$
Eau vapeur	
Conductivité thermique à 100°C	0,231mW .cm
Densité par rapport à l'air	0,62337
Indice de réfraction pour la raie D à100°C	1,00259

I. 4. Cycle de l'eau

Le cycle de l'eau est engendré par les échanges entre divers réservoirs d'eau, notamment les océans, les fleuves, les lacs, les glaciers, l'atmosphère, les nappes phréatiques, les végétaux, etc. Ce processus cyclique perpétuel implique de nombreux phénomènes (voir Figure 1)[12].

I. 4. 1.Evaporation

L'évaporation est le phénomène durant lequel l'eau liquide se convertit en gaz ou vapeur, nécessitant l'apport de chaleur fournie par le soleil. L'évaporation des océans constitue la principale voie par laquelle l'eau entre dans l'atmosphère.

I. 4. 2.Condensation

La condensation est le mécanisme par lequel la vapeur d'eau se convertit en eau liquide. Son rôle est crucial dans la formation des nuages.

I. 4. 3.Précipitations

La précipitation se produit lorsque l'eau contenue dans les nuages se décharge sous forme de pluie, de neige ou de grêle. Il s'agit du principal processus par lequel l'eau atmosphérique retourne sur Terre.

I. 4. 4.Infiltration

Lorsque l'eau sous forme de pluie ou de neige pénètre dans le sol et les roches, une partie est absorbée par la végétation avant d'être renvoyée dans l'atmosphère. Le reste s'accumule dans le sous-sol pour constituer des nappes phréatiques, qui peuvent à leur tour alimenter des sources émergentes à la surface du sol.

I. 5. Eau alimentaire

L'eau destinée à la consommation humaine doit être exempte de substances chimiques dangereuses et de micro-organismes nocifs pour la santé[13]. Cette eau peut provenir de différentes sources (Figure I.1), qu'elles soient superficielles ou souterraines [12].

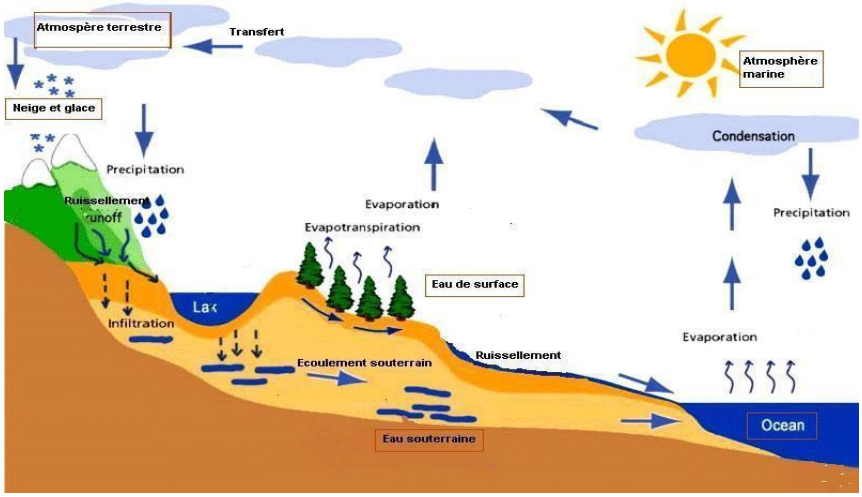


Figure I.1: Cycle biogéochimique de l'eau

I. 5. 1.Eaux de surface

Les eaux de surface comprennent les fleuves, les lacs, les rivières, les barrages, les réservoirs de stockage (tels que les citernes de récolte d'eau de pluie). Elles sont souvent contaminées, présentant une forte teneur en impuretés et une pollution biologique, principalement chimique. Cette pollution est largement causée par les rejets d'eaux usées dans l'environnement, qui contiennent des polluants toxiques. Avant toute utilisation domestique, ces eaux doivent être traitées [13].

I. 5. 2.Eaux souterraines

Les eaux souterraines se trouvent principalement dans les pores des sédiments ou dans les fissures des roches. Les plus adaptées à la consommation humaine sont celles situées à quelques dizaines de mètres sous la surface terrestre. En effet, à mesure que l'on s'enfonce dans le sol, l'eau devient plus chargée en sels minéraux et en métaux divers, ce qui peut la rendre impropre à la consommation. Ces eaux finissent par rejoindre la nappe phréatique [14].

I. 5. 3.Eau de sources

Une eau bactériologiquement saine, issue d'une nappe souterraine ou d'un gisement, provient d'une source exploitée par une ou plusieurs émergences naturelles ou forées. On distingue trois types de sources : les sources de versant, qui surgissent des pentes ou des hauteurs élevées ; les sources d'émergence (artésiennes), alimentées par une couche inférieure de la nappe ; et les sources d'affleurement, dont l'alimentation provient d'un substrat imperméable dans la vallée [13].

I. 5. 4.Eau de puits

L'eau provenant d'un puits provient généralement de la captation des eaux d'une nappe phréatique de profondeur moyenne. Il existe différents types de puits, tels que les puits ordinaires, les puits foncés, les puits forés et les puits artésiens [13].

Cette eau, considérée comme une source "alternative", n'est pas réglementée et est généralement considérée comme non potable. Elle présente un risque majeur à court terme en raison du risque microbiologique, exposant les consommateurs à des maladies telles que des diarrhées, des gastro-entérites ou des intoxications pouvant être graves. À long terme, il existe également un risque chimique en raison de la présence potentielle de pesticides, de métaux lourds, d'hydrocarbures et d'autres substances pouvant s'accumuler dans l'organisme [15].

I .6. Méthodes classiques de contrôle qualité de l'eau potable

Les méthodes classiques de contrôle de la qualité de l'eau reposent souvent sur une combinaison d'analyses physico-chimiques et microbiologiques. Les paramètres physico-chimiques incluent des mesures telles que le pH, la conductivité électrique, la turbidité et la teneur en oxygène dissous, qui donnent des indications sur la composition chimique et la clarté de l'eau. Les analyses microbiologiques visent à détecter la présence de bactéries indicatrices telles que Escherichia coli, qui peuvent indiquer une contamination fécale et présenter des risques pour la santé humaine. Les échantillons d'eau sont prélevés à des emplacements stratégiques, tels que des points de captage, des réservoirs ou des points de distribution, et sont analysés selon des protocoles normalisés pour assurer la fiabilité des résultats. Ces méthodes traditionnelles, bien que nécessaires et largement utilisées, sont en train d'être complétées par des approches plus avancées telles que la surveillance en temps réel et l'utilisation de capteurs à distance pour fournir une évaluation continue de la qualité de l'eau.

I.6.1 Échantillonnage des eaux

L'échantillonnage des eaux est une opération qui consiste à prélever des échantillons d'eau en vue d'analyse. L'échantillonnage est une étape primordiale de l'analyse car pour obtenir un résultat d'analyse pertinent, il est nécessaire que l'échantillon d'eau soit représentatif, qu'il soit conservé dans de bonnes conditions jusqu'au lieu de l'analyse, et qu'il ne soit pas contaminé par toute substance qui pourrait provoquer un résultat d'analyse erroné (Figure I.2).

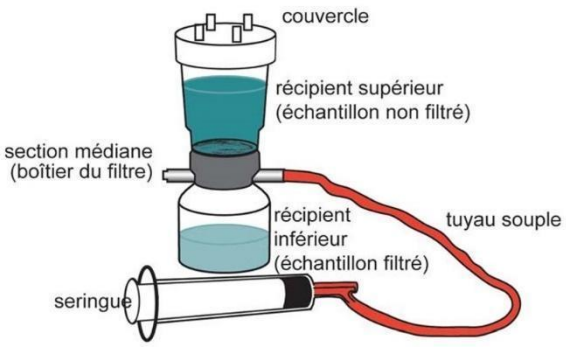


Figure I.2: Exemple d'appareil de filtration

.

I.6.2Analyses

Certains paramètres peuvent évoluer pendant le transport des échantillons au laboratoire et il est toujours préférable de faire ces déterminations sur le terrain pH, température, O_2, CO_2, H_2S, NH_3, potentiel d'oxydo-réduction, oxydants résiduels... La mesure de ces paramètres Repose souvent sur des méthodes de précision inférieure à celle des méthodes de laboratoire mais l'intérêt de la mesure immédiate peut être prépondérant compte tenu des variations susceptibles d'intervenir durant le transport et la conservation en laboratoire. Par ailleurs, ces analyses sont aussi nécessaires lors de l'étude prolongée d'une eau brute ou d'un effluent en vue de l'établissement d'un projet ou de l'optimisation d'uneinstallation de traitement (Figure I.3).



Figure I. 3: Exemple d'un kit d'analyse d'eau potable

.

I.6.3 Éléments d'interprétation

I.6.3.1 Eaux souterraine

La caractérisation des eaux souterraines concerne essentiellement l'analyse des paramètres de base (pH, température et conductivité) des ions majeurs et d'éventuels éléments traces. Les analyses des isotopes ne sont pas évoquées ici conductivité, la température et le pH permettent de définir les caractéristiques fondamentales de l'eau. La recherche des contrastes de ces paramètres entre différents points de mesures facilite la mise en évidence de zones d'alimentation, de pollution ou de géologie différente.

I.6.3.2 Eaux de surface

Le pH, la conductivité, l'oxygène dissous, la température et la turbidité sont mesurés in situ. Ils permettent une première estimation de la qualité du milieu. Phosphates, l'ammoniac, les nitrates et le fer permettent d'apprécier l'impact de rejets polluants. Les analyses bactériologiques confirment, en général, la pollution microbienne de ces eaux de surface et permettent

D'en fixer l'importance. Les indices biologiques sont surtout intéressants pour les eaux courantes. Les périodes et les stations d'études doivent être choisies pour cerner au mieux la variabilité spatiale et temporelle de la qualité du milieu (cf. paragraphe relatif à l'échantillonnage).

I. 7. Analyse de l'eau potable

Une analyse de l'eau potable est segmentée en divers groupes de paramètres, chacun comprenant des indicateurs spécifiques visant à assurer le maintien de la qualité globale de l'eau potable.

I. 7. 1. Paramètres physiques

I. 7. 1. 1. Température

Une température élevée accélère la plupart des réactions physico-chimiques, favorise le développement de micro-organismes, accentue le goût et la saveur, et peut provoquer des odeurs désagréables. De plus, elle peut également être un facteur important dans la corrosion des canalisations. Pour garantir une eau potable rafraîchissante, la température maximale admissible est de 15°C [16].

I. 7. 1. 2. Potentiel d'hydrogène (pH)

Le pH, exprimant la concentration d'ions hydrogène dans une solution, revêt une importance exceptionnelle, notamment dans les processus de traitement de l'eau. Il joue un rôle déterminant dans l'agressivité de l'eau envers les canalisations et dans l'action du chlore lors de la désinfection. Selon le décret nº 2.914/2011 du Ministère de la Santé, le pH de l'eau devrait être maintenu dans une plage de 6,0 à 9,5 dans le système de distribution[17].

I. 7. 1. 3. Turbidité

La turbidité de l'eau est générée par la présence de matières en suspension (MES) telles que l'argile, le limon, les particules organiques, le plancton et divers autres organismes microscopiques. Ce paramètre est mesuré dans les eaux souterraines, les eaux usées, les eaux de surface et l'eau potable). Les matières colloïdales, qu'elles soient d'origine organique ou minérale, contribuent à une faible proportion de la turbidité. La valeur moyenne de turbidité obtenue pour le Campus Wando Genet (0,98 NTU) est inférieure à la valeur recommandée par l'OMS de 5,00 NTU.

I. 7. 1. 4. Conductivité

Conductivité de L'eau pure n'est pas un bon conducteur de courant électrique, mais plutôt un bon isolant. L'augmentation de la concentration en ions améliore la conductivité électrique de l'eau. Généralement, la quantité de solides dissous dans l'eau détermine la

conductivité électrique. La conductivité électrique (CE) mesure en fait le processus ionique d'une solution qui lui permet de transmettre le courant. Selon les normes de l'OMS, la valeur EC ne doit pas dépasser 400 µS/cm.

I. 7. 1. 5. Dureté

La dureté de l'eau est une mesure de la concentration de certains minéraux dissous, principalement le calcium et le magnésium. Ces minéraux sont dissous dans l'eau lorsqu'elle passe à travers des couches de roches et de sols contenant des minéraux tels que le carbonate de calcium (calcaire) et le sulfate de magnésium (gypse).

La dureté de l'eau est généralement mesurée en unités de calcium carbonate (CaCO3) ou en grains par gallon (gpg) aux États-Unis. Elle est classée en plusieurs catégories :

- Très douce : Moins de 1 gpg ou 17.1 mg/L de $CaCO_3$
- Douce : De 1 à 3.5 gpg ou 17.1 à 60 mg/L de $CaCO_3$
- Modérément dure : De 3.5 à 7 gpg ou 60 à 120 mg/L de $CaCO_3$
- Dure : De 7 à 10.5 gpg ou 120 à 180 mg/L de $CaCO_3$
- Très dure : Plus de 10.5 gpg ou 180 mg/L de $CaCO_3$

La dureté de l'eau peut avoir un impact sur plusieurs aspects de la vie quotidienne, tels que l'efficacité des produits de nettoyage, l'entretien des appareils électroménagers, et la formation de dépôts dans les canalisations. C'est pourquoi il est souvent utile de connaître la dureté de l'eau dans une région donnée.

I. 7. 1. 5. Solides (total des solides dissous-TDS)

L'eau a la capacité de dissoudre une large gamme de minéraux ou de sels inorganiques et organiques tels que le potassium, le calcium, le sodium, les bicarbonates, les chlorures, le magnésium, les sulfates, etc. Ces minéraux produisent un goût indésirable. Et une couleur diluée ayant l'apparence de l'eau. C'est le paramètre important pour l'utilisation de l'eau. L'eau avec une valeur TDS élevée indique que l'eau est hautement minéralisée. Le boîtier TDS du camion-citerne d'eau potable à 500 ppm utilisera les recommandations de l'Organisation Mondiale de la Santé (OMS). Les enfants entre 300 et 500 ppm sont généralement considérés comme acceptables. Les climatiseurs neufs à 300 ppm sont à privilégier pour une meilleure qualité gustative et sensorielle.

I. 7. 1. 6. Chloramines :

Le chlore et la chloramine sont les principaux désinfectants utilisés dans les réseaux publics d'eau. Les Chloramines se forment le plus souvent lorsque de l'ammoniac est ajouté au chlore pour traiter l'eau potable. Des niveaux de chlore allant jusqu'à 4 milligrammes par litre (mg/L ou 4 parties par million (ppm) sont considérés comme sûrs dans l'eau potable.

I. 7. 1. 7. Sulfate :

Les sulfates sont des substances naturelles présentes dans les minéraux, le sol et les roches. Ils sont présents dans l'air ambiant, les eaux souterraines, les plantes et les aliments. La principale utilisation commerciale du sulfate se situe dans l'industrie chimique. La concentration de sulfate dans l'eau de mer est d'environ 2 700 milligrammes par litre

(mg/L). Elle varie de 3 à 30 mg/L dans la plupart des réserves d'eau douce, bien que des concentrations beaucoup plus élevées (1 000 mg/L) soient trouvées dans certaines zones géographiques

I. 7. 1. 8. Carbone organique :

Le carbone organique total (COT) dans les eaux de source provient de la matière organique naturelle (MON) en décomposition ainsi que de sources synthétiques. Le COT est une mesure de la quantité totale de carbone contenue dans les composés organiques de l'eau pure. Selon l'US EPA < 2 mg/L de COT dans l'eau traitée/potable, et < 4 mg/L dans l'eau de source utilisée pour le traitement.

I. 7. 1. 9. Trihalométhanes

Les THM sont des produits chimiques qui peuvent être présents dans l'eau traitée au chlore. La concentration de THM dans l'eau potable varie en fonction du niveau de matière organique présente dans l'eau, de la quantité de chlore requise pour traiter l'eau et de la température de l'eau traitée. Des niveaux de THM allant jusqu'à 80 ppm sont considérés comme sans danger dans l'eau potable.

I. 7. 2. Paramètres organoleptiques

I. 7. 2. 1. Couleur

La coloration de l'eau provient de diverses matières organiques telles que les substances humiques et les tanins, ainsi que des métaux tels que le fer et le manganèse, ainsi que des résidus industriels fortement colorés. Cette coloration est esthétiquement indésirable. Il est important de mesurer la couleur de l'eau, car une couleur élevée est esthétiquement indésirable. Le décret MS Nº 2914/2011 établit une valeur maximale permise de 15 pour la couleur apparente (voir Tableau 2).

I. 7. 2. 2. Odeur

Une eau destinée à la consommation doit être dépourvue d'odeur. En effet, la présence d'une odeur est un indicateur de pollution ou de la présence de matières organiques en décomposition. Ces substances sont généralement présentes en quantités si minimes qu'elles ne peuvent pas être détectées par les méthodes d'analyse ordinaires. Parfois, seul le sens olfactif peut les détecter[18](voir Tableau 2).

I. 7. 2. 3. Goût et saveur

Le goût représente l'ensemble des sensations gustatives, olfactives et chimiques perçues lorsqu'un aliment ou une boisson est dans la bouche, tandis que la saveur est l'ensemble des sensations perçues suite à la stimulation par certaines substances solubles des bourgeons gustatifs (Tableau I.2)[13].

I. 7. 3. Paramètres microbiologiques

Les microorganismes sont omniprésents dans l'environnement, comprenant à la fois des espèces bénéfiques et d'autres qui peuvent servir d'indicateurs microbiologiques de la qualité de l'eau potable. Parmi ces derniers, il est recommandé de réaliser l'analyse des cinq (05) groupes principaux voir (Figure I.4).

Tableau I. 2:Normes de potabilité d'une eau de consommation

Unité	Concentration	Caractéristiques
Caractéristiques organoleptiques		
Mg/l de platine	Au maximum 25	Couleur
	Au maximum 4	Odeur (seuil de perception à 25°C)
	Au maximum 4	Saveur (seuil de perception à 25°C)
Unité Jackson	Au maximum 2	Turbidité
Caractéristiques physico-chimiques		
Unité PH	6.5 à 8.5	pH
Us/cm	Au maximum 2.800	Conductivité (à 20°C)
Mg/l de $CaCO_3$	100 à 500	Dureté
Mg/l	200 à 500	Chlorures
Mg/l	200 à 400	Sulfates
Mg/l	75 à 200	Calcium
Mg/l	150	Magnésium
Mg/l	200	Sodium
Mg/l	20	Potassium
Mg/l	0.2	Aluminium
Mg/l en oxygène	Au maximum 3	Oxydabilité aux permanganates de potassium
Mg/l	1.5 à 2	Résidus secs après dessiccation à 180°C
Caractéristiques concernant les substances indésirables		
Mg/l	Au maximum 50	Nitrates
Mg/l	Au maximum 0.1	Nitrites
Mg/l	Au maximum 0.5	Ammonium
Mg/l	Au maximum 1	Azote
Mg/l	0.2 à 2	Fluor
Caractéristiques concernant les substances indésirables		
Mg/l	Au maximum 0.3	Fer
Mg/l	Au maximum 0.5	Manganèse
Mg/l	Au maximum 1.5	Cuivre
Mg/l	Au maximum 5	Zinc
Mg/l	Au maximum 0.05	Argent
Caractéristiques concernant les substances toxiques		
Mg/l	Au maximum 0.05	Arsenic
Mg/l	Au maximum 0.01	Cadmium
Mg/l	Au maximum 0.05	Cyanure
Mg/l	Au maximum 0.05	Chrome
Mg/l	Au maximum 0.01	Mercure
Mg/l	Au maximum 0.05	Plomb
Mg/l	Au maximum 0.01	Sélénium

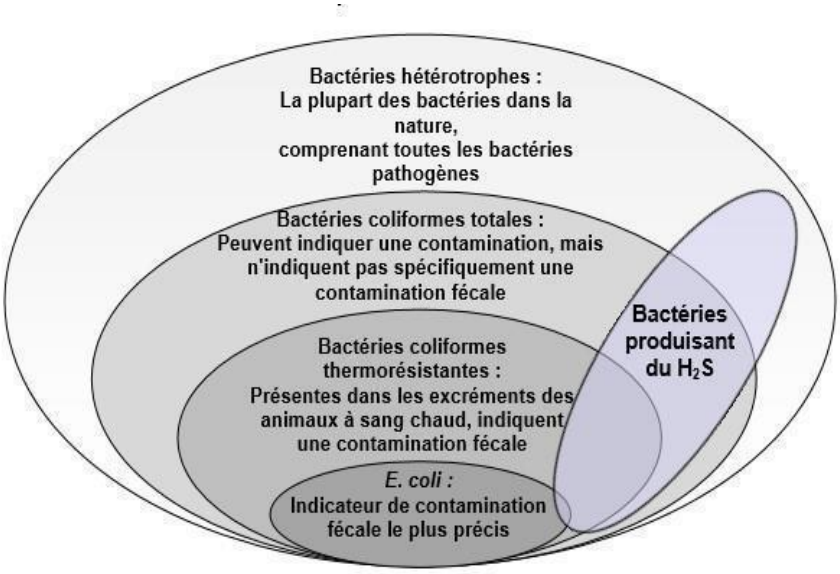


Figure I.4: Groupes d'indicateurs bactériens

I. 7. 3. 1. Germes totaux

Les microorganismes revivifiables désignent toute bactérie aérobie, levure ou moisissure capable de former des colonies dans un milieu spécifié et selon les conditions d'essai décrites dans la norme. Le dénombrement de ces germes, également appelés mésophiles aérobies en fonction de leurs conditions de croissance, est utilisé comme indicateur de pollution. Il est employé dans les environnements naturels de très bonne qualité microbiologique afin de contrôler une éventuelle contamination bactérienne. Ce type d'analyse est principalement effectué sur les eaux souterraines provenant de nappes profondes ou sur les réseaux d'eau. Une augmentation de la concentration bactérienne après le traitement peut indiquer une multiplication bactérienne dans le réseau ou une intrusion de bactéries à l'intérieur de celui-ci.

Parmi les bactéries se développant dans les conditions de la norme, on peut distinguer deux catégories hygiéniques :

- Les microorganismes se développant à 20°C, qui sont des saprophytes naturellement présents dans l'eau.
- Les microorganismes se développant à 37°C, température correspondant à celle du corps humain, qui peuvent provenir de l'homme ou d'animaux à sang chaud.

Bien qu'ils ne soient pas nécessairement des agents pathogènes, ces microorganismes peuvent indiquer une contamination de l'eau par des produits animaux, en particulier les matières fécales.

I. 7. 3. 2. Coliformes totaux

Les coliformes totaux sont un groupe de bactéries couramment présentes sur les végétaux, dans les sols et dans les intestins des humains ainsi que des animaux à sang chaud. En raison de leur ubiquité dans l'environnement, ils ont été largement utilisés comme outil de surveillance. Initialement, les coliformes totaux étaient composés de quatre groupes de bactéries : Escherichia, Klebsiella, Enterobacter et Nitrobacter, présents dans les excréments d'animaux à sang chaud, y compris les humains. Cependant, des recherches

récentes ont révélé que les coliformes totaux incluent en réalité un ensemble beaucoup plus vaste de bactéries que les quatre groupes d'origine.

À ce jour, 19 groupes de bactéries appartenant aux coliformes totaux ont été identifiés, dont seulement dix sont associés aux excréments[19]. De nombreux groupes de bactéries appartenant aux coliformes totaux dans l'environnement sont liés au sol, à la végétation ou aux sédiments aquatiques, ce qui signifie que toutes les bactéries coliformes totales ne proviennent pas d'excréments. De plus, certaines de ces bactéries sont capables de se reproduire dans l'eau naturelle, ce qui complique la détermination de la contamination fécale de l'eau.

En conclusion, les coliformes totaux ne devraient pas être utilisés comme indicateur de contamination fécale de l'eau potable, car ils ne répondent pas aux deux critères essentiels d'un bon indicateur : être associé uniquement aux excréments d'animaux et ne pas se multiplier dans les eaux naturelles [20].

I. 7. 3. 3. Coliformes fécaux

Les coliformes thermorésistants (CTR) sont un sous-groupe des coliformes totaux. Ce sont des bactéries à Gram négatif, aérobies ou anaérobies facultatifs, non sporulées, dépourvues d'oxydase, capables de se multiplier en présence de sels biliaires et de fermenter le lactose avec production d'acide et de gaz en 24 à 48 heures à une température comprise entre 36 et 37°C.

Les CTR sont présents dans les intestins des animaux à sang chaud. Parmi les coliformes provenant des excréments humains, 96,4% sont des coliformes thermorésistants. Ils se distinguent des coliformes totaux par leur capacité à se reproduire à des températures plus élevées (42-44.5°C) et sont étroitement associés à la pollution fécale plus étroitement que les coliformes totaux[21].

Historiquement, les coliformes fécaux ont été largement utilisés comme indicateurs bactériens de contamination fécale. Bien qu'Escherichia colis (E. coli) ait remplacé les coliformes thermorésistants en tant qu'indicateur plus spécifique de contamination fécale, la présence de coliformes thermorésistants dans un échantillon d'eau demeure un risque immédiat pour la santé publique.

I. 7 .3. 4. Escherichia coli

E. coli est le principal indicateur utilisé dans l'analyse de la qualité de l'eau depuis plus de 50 ans. Il s'agit d'une bactérie coliforme thermorésistante que l'on trouve principalement dans les excréments des animaux à sang chaud, y compris les humains. Une forte pollution fécale peut entraîner des niveaux d'E. coli de plusieurs millions de bactéries par litre d'eau. La plupart des souches d'E. coli sont inoffensives, mais certaines, comme O157 :H7, sont pathogènes et peuvent causer des diarrhées sévères et d'autres symptômes.

E. coli partage des propriétés biochimiques similaires aux autres coliformes, mais se distingue par la présence des enzymes bêta-glucuronidase et galactosidase (bêta-gal). De nombreuses méthodes d'analyse de l'eau utilisent la détection de cette enzyme bêta-glu pour identifier E. coli dans les échantillons d'eau. Plus de 95% des souches d'E. coli testées possèdent cette enzyme, qui est un excellent indicateur de la contamination fécale de l'eau potable. Cependant, certaines preuves suggèrent que E. coli peut se développer dans des

sols tropicaux riches en éléments nutritifs, bien que ce phénomène soit généralement limité par la compétition avec d'autres bactéries indigènes[22]. Quoi qu'il en soit, la détection d'E. coli dans un échantillon d'eau de boisson constitue une préoccupation immédiate pour la santé publique.

I. 7. 3. 5. Streptocoques fécaux

Les streptocoques fécaux, également appelés entérocoques intestinaux ou streptocoques du groupe « D » selon la classification de Landefeld (1928), sont des bactéries présentant une forme de Cocci à Gram positive, sphérique ou ovoïde, non sporulées, formant des chaînettes. Ils sont catalase négative et possèdent un antigène du groupe D. Ces bactéries se développent en anaérobiose à 44°C et à pH 9,6, et sont capables de dégrader l'esculine en présence de 40% de bile. Elles peuvent se multiplier en 24 à 48 heures à 37°C sur un milieu sélectif à l'azoture de sodium, formant des colonies caractéristiques qui réduisent le TTC et hydrolysent l'esculine en deux heures à 44°C après repiquage d'une colonie sur une gélose biliée à l'esculine et à l'azoture. Les streptocoques fécaux sont des pathogènes opportunistes pouvant causer des septicémies et des infections urinaires.

I. 7. 3. 6. Spores des bactéries anaérobies sulfite-réductrice

Les microorganismes anaérobies, dont les spores sont la forme de résistance, comprennent principalement les clostridies, les plus fréquentes et faciles à mettre en évidence. Bien qu'ils soient présents dans les matières fécales, leur quantité est généralement moindre que celle d'Escherichia colis et d'autres coliformes dans l'environnement, tels que le sol et les rivières. Ces bactéries peuvent provoquer un goût et une odeur désagréables en raison de la production de sulfure d'hydrogène. De plus, elles peuvent entraîner la corrosion des tuyaux en métal et en béton, ainsi que la tuberculisation des canalisations.

Les spores de ces microorganismes ont une similitude de comportement avec les parasites, notamment une résistance à la chloration, ce qui en fait un indicateur potentiel de la présence de microorganismes pathogènes. En raison de leur résistance aux conditions défavorables de l'environnement, telles que la chaleur, les radiations ultraviolettes, les désinfectants chimiques et la dessiccation, les spores sont recherchées plutôt que les bactéries elles-mêmes dans le cadre de l'analyse de la qualité de l'eau.

I. 8. Contamination de l'eau

Dans la nature, l'eau dite "pure" n'existe pas réellement, car toutes les sources d'eau contiennent naturellement des produits chimiques lessivés depuis l'environnement. Dans la plupart des cas, les niveaux de ces produits chimiques naturels sont bénéfiques ou ont des conséquences minimes. Cependant, il existe également des produits chimiques synthétiques issus d'activités humaines qui peuvent contaminer l'eau et affecter sa qualité.

L'Organisation mondiale de la Santé (OMS) catégorise les sources de produits chimiques en cinq groupes distincts : les substances naturellement présentes, celles provenant des activités agricoles, celles issues des sources industrielles et des peuplements humains, les produits chimiques liés au traitement de l'eau, ainsi que les pesticides utilisés dans l'eau à des fins de santé publique.

I.9. Conclusion

La qualité de l'eau potable est une préoccupation majeure pour la santé publique et l'environnement. Cette étude générale souligne l'importance de surveiller et de maintenir des normes rigoureuses pour garantir l'accès à une eau potable sûre et saine pour tous. Des mesures de protection et des politiques de gestion efficaces sont nécessaires pour prévenir la contamination et protéger les ressources en eau. En investissant dans la recherche, l'infrastructure et l'éducation, nous pouvons assurer un approvisionnement durable en eau potable pour les générations présentes et futures.

Chapitre II

Outils Software et Hardware utilisés

II.1 Introduction

Les avancées récentes dans les domaines de l'IoT, de l'IA et des langages de programmation comme Python ont ouvert de nouvelles perspectives pour le contrôle de la qualité de l'eau. En utilisant des plateformes telles que Node-RED, ces technologies permettent des solutions innovantes pour surveiller en temps réel les paramètres de l'eau. Python et Node-RED jouent un rôle clé dans le développement de solutions IoT, offrant flexibilité et facilité de développement. L'IoT permet le déploiement de capteurs connectés pour surveiller les indicateurs de qualité de l'eau, tandis que l'intégration de l'IA permet une analyse avancée des données, permettant une surveillance proactive et prédictive. En résumé, ces technologies combinées offrent des moyens efficaces pour garantir la potabilité de l'eau, en fournissant une surveillance en temps réel, une analyse avancée des données et une automatisation des processus pour une gestion plus efficace des ressources en eau.

II.2 Outils Software

II.2.1 Méthodes de machines Learning

II.2.1.1 Réseaux de neurones artificiels

II.2.1.1.1 Définition

Un réseau de neurones (RNA) est un système d'opérateurs non linéaires interconnectés, recevant des signaux de l'extérieur par ses entrées, et délivrant des signaux de sortie, ces réseaux de neurones sont une métaphore des structures cérébrales, de traitement parallèle, de distribution d'information et comportent plusieurs éléments de traitement appelé neurone.

Chaque neurone fonctionne indépendamment des autres de telle sorte que l'ensemble est un système parallèle fortement interconnecte. L'information détenue par le réseau de neurone est distribuée à travers l'ensemble des constituants et non localisée dans une partie de mémoire sous la forme d'un symbole.

Le réseau de neurone ne se programme jamais pour réaliser une ou telle tâche, il est entraîné sur des données acquises, grâce au mécanisme d'apprentissage qui agit sur les constituants du réseau afin de réaliser au mieux la tâche souhaitée. Le réseau de neurone ne se programme jamais pour réaliser une ou telle tâche, il est entraîné sur des données acquises, grâce au mécanisme d'apprentissage qui agit sur les constituants du réseau afin de réaliser au mieux la tâche souhaitée.

II.2.1.1.2 Structure des réseaux de neurones artificiels (RNA)

En général, on peut dire que les réseaux de neurones peuvent être caractérisés comme des modèles de calculs apprenant, généralisant et organisant des données (Figure II.1)

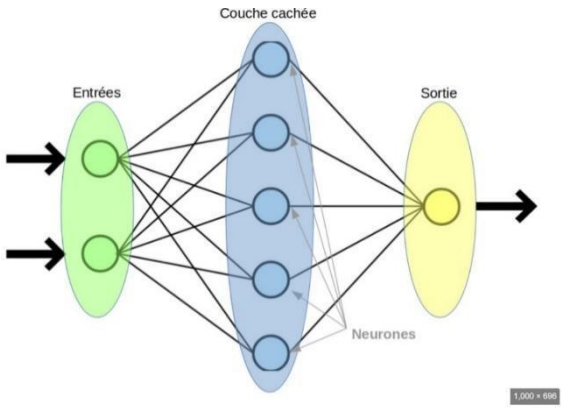


Figure II. 1: Schéma simple d'un réseau de neurone.

Un réseau de neurone artificiel contient un grand nombre d'unités, les neurones, qui communiquent entre eux en s'envoyant des signaux à travers de liens, appelées connexions synaptiques. En général le système de neurone possède trois types des neurones :

- Les neurones d'entrée qui reçoivent les données ;
- Les neurones de sortie qui envoient les données par la sortie du système ;
- Les neurones cachés, dont les signaux d'entrée et de sortie demeurent dans le système [24]

U_i Représente la somme pondérée des entrées du neurone, elle est donnée par la relation (II.1).

$$U_i = \sum_j w_{i,j} x_j + b_i \tag{II.1}$$

Où : x_i représente l'entrée j connectée au neurone i. b_i est le seuil interne du neurone. $w_{i,j}$ Désigne le poids de la connexion reliant l'entrée j au neurone. $O_i = g(U_i)$ désigne la sortie du neurone avec g comme fonction d'activation.

La fonction d'activation de chaque neurone détermine ses propres caractéristiques. Par conséquent, le type du neurone est caractérisé par sa fonction d'activation. Conformément au neurone biologique, les fonctions d'activation sont généralement croissantes et continues. Les fonctions les plus utilisées sont la fonction linéaire, tangente hyperbolique ou sigmoïde.

II.2.1.1.3 Domaines des RNA

Les RNA aujourd'hui ont des applications dans des domaines variés :

II.2.1.1.3.1 Application au traitement du signal

Dans ce domaine les applications réalisées traitent essentiellement de la reconnaissance de signatures radar ou sonner cette application a été construite à partir de signaux sonars et des traits caractéristiques de ces signaux déterminés par les experts comme utiles pour identifier une cible en environnement bruité. Le réseau agit comme un filtre pour éliminer le bruit.

II.2.1.1.3.2 Application à la planification

L'optimisation est souvent sollicitée pour la planification des actions ou des taches. À cet effet, l'utilisation du modèle de Hop Field parait très utiles pour la résolution des problèmes d'optimisation.

II.2.1.1.3.3 Application au contrôle

Grâce à leur propriétés (parallélisme de traitement, capacité d'adaptation, et de généralisation, etc.…). Les réseaux de neurones sont appliqués au contrôle intelligent. Par exemple plusieurs tentatives ont été faites pour appliquer les réseaux de neurones ; Ces applications peuvent être classifiés en plusieurs méthode, Telles que : le contrôle supervisé. Le contrôle inverse et le contrôle neuronal adaptatif [23].

II.2.1.1.3.4 Application au diagnostic

Les réseaux de neurones sont bien adaptés à la résolution des problèmes de diagnostic, utilisant la classification automatique des signaux et des formes. Dans ce contexte on distingue plusieurs applications des réseaux de neurones pour le diagnostic des défaillances et en particulier, pour le diagnostic des pannes des machines électriques [23].

II.2.1.1.4 Apprentissage

L'apprentissage et l'adaptation constituent deux caractéristiques essentielles des réseaux de neurones. Le rôle de l'apprentissage est de définir le poids de chaque connexion. De nombreuses règles existent pour modifier le poids des connexions et donc pour arriver à un apprentissage correct. Lorsque la phase d'apprentissage est achevée, le réseau doit être capable de faire les bonnes associations pour les vecteurs d'entrées qu'il n'aura pas appris. C'est l'une des propriétés importantes dans les réseaux de neurones, car elle permet de donner la capacité de reconnaître des formes ressemblantes et même dégradées des prototypes, c'est la phase de reconnaissance.

II.2.1.1.4.1 Types d'apprentissage

II.2.1.1.4.1.1 Apprentissage supervisé

Un superviseur, ou professeur, fournit au réseau des couples d'entrées-sorties. Il fait apprendre au réseau l'ensemble de ces couples, par une méthode d'apprentissage, comme la retro-propagation du gradient de l'erreur, en comparant pour chacun d'entre eux la sortie effective du réseau et la sortie désirée. L'apprentissage est terminé lorsque tous les couples entrées-sorties sont reconnus par le réseau. Ce type d'apprentissage se retrouve, entres autres dans le perceptron.

II.2.1.1.4.1.2 Apprentissage non supervisé

Cet apprentissage consiste à détecter automatiquement des régularités qui figurent dans les exemples présentés et à modifier des connexions pour que les exemples ayant les mêmes caractéristiques de régularité provoquent la même sortie. Les réseaux auto-organisateurs de Kohonen sont des réseaux à apprentissage non supervisé les plus connus

II.2.1.1.4.1.3 Apprentissage semi-supervisé

L'apprentissage semi-supervisé combine des données étiquetées et non étiquetées pour améliorer la performance des modèles d'apprentissage automatique. Seule une petite partie des données est annotée, tandis que la majorité ne l'est pas. Cette méthode exploite les

données non annotées pour améliorer la précision et la généralisation des modèles, tout en réduisant la nécessité de jeux de données entièrement annotés, souvent coûteux à produire. C'est une approche efficace dans les domaines où l'annotation manuelle est difficile, coûteuse ou peu pratique, élargissant ainsi les possibilités d'application des modèles d'apprentissage automatique.

II.2.1.1.4.2 Méthodes d'apprentissage

Dans les systèmes experts, les connaissances de l'expert ont une forme énumérée ; elles sont exprimées sous forme de règles. Dans le cas des réseaux de neurones, les connaissances ont une forme distribuée ; elles sont codées dans les poids des connexions, la topologie du réseau, les fonctions de transfert de chaque neurone, le seuil de ces fonctions, la méthode d'apprentissage utilisée. Il existe un certain nombre de méthodes d'apprentissage :

II.2.1.1.4.2.1 Règle de Hebb

C'est la méthode d'apprentissage la plus ancienne (1943) elle est inspirée de la biologie, Elle traduit le renforcement des connexions liant deux neurones activés. Si un des deux neurones au moins n'est pas activé, le poids de la connexion n'est pas modifié.

II.2.1.1.4.2.2 Retro-propagation du gradient de l'erreur

Cet algorithme est utilisé dans les réseaux de type feedforward, ce sont des réseaux de neurones à couche, ayant une couche d'entrée, une couche de sortie, et au moins une couche cachée. Il n'y a pas de récursivité dans les connexions, et pas de connexions entre neurones de la même couche. Le principe de la retro-propagation consiste à présenter au réseau un vecteur d'entrées, de procéder au calcul de la sortie n passant par les couches. Cette sortie obtenue est comparée à la sortie désirée, une erreur est alors obtenue. A partir de cette erreur, est calculé le gradient de l'erreur qui est à tour propagé de la couche de sortie vers la couche d'entrée, d'où le terme de retro-propagation. Cela permet la modification des poids du réseau et donc l'apprentissage. L'opération est réitérée pour chaque vecteur d'entrée et cela jusqu'à ce que le critère soit vérifié.

II.2.1.1.5 Types des RNA

II.2.1.1.5.1 Perceptron

Depuis les résultats des travaux de Mac Culloch & Pitt (1943), qui ont abouti à la définition du neurone formel, ainsi ceux de Hebb, expliquant les effets d'apprentissage de mémoire et de conditionnement à partir de groupes de cellules. Pour expliquer ces effets d'apprentissage, Hebb propose, que les cellules apprennent à modifier l'intensité des connexions qui les relient, en fonction de leur activité simultanée. L'idée de certains chercheurs fut d'utiliser les modélisations des neurones et de l'évolution des synapses pour simuler des réseaux de neurones. Le premier modèle solide fut présenté en 1959 par F. Rosenblatt. Il s'agit du perceptron, autrement dit un réseau réduit à un seul neurone formel.

II.2.1.1.5.2 Algorithme d'apprentissage :

L'algorithme d'apprentissage est la méthode qui va modifier les poids de connexion afin de converger vers une solution qui permettra au réseau d'accomplir la tâche désirée. L'apprentissage est une méthode d'identification paramétrique qui permet d'optimiser les valeurs des poids du réseau.

Plusieurs algorithmes itératifs peuvent être mis en œuvre, parmi lesquels on note :

II.2.1.1.5.2.1 Algorithme de rétro propagation :

L'algorithme de rétro propagation du gradient de l'erreur a été créé en généralisant les règles d'apprentissages de Widrow-Hoff, aux réseaux multicouches à fonctions de transfert non linéaire. C'est un algorithme utilisé avec des réseaux de types feedforward pour l'apprentissage de fonction, la reconnaissance de forme et la classification.

II.2.1.1.5.2.2 Principe :

La rétro propagation du gradient de l'erreur est utilisée pour ajuster les poids et le biais du réseau afin de minimiser l'erreur quadratique entre la sortie du réseau et la sortie réelle. A chaque couple entrée/sortie, une erreur est calculée, le gradient, ou pente, de l'erreur est déterminé. Ensuite les poids et les biais sont modifiés en ligne sur le réseau. On réitère ces calculs jusqu'à l'obtention du critère d'arrêt.

II.2.1.1.5.2.3 Algorithme :

L'algorithme de la rétro propagation du gradient de l'erreur se résume aux étapes suivantes :

- Initialisation des poids w^q à des petites valeurs aléatoires.
- Présentation d'une entrée x_k et de la sortie désirée d_k.
- Calcul de la sortie actuelle par propagation à travers les couches en utilisant la relation (II.2).

$$y_j^q = F\left(\sum_i w_{ji}^q y_i^{q-1}\right) \qquad \text{(II.2)}$$

Ou F est la fonction de transfert du neurone et [q] la sème couches du réseau.

- Accumulation des erreurs en sortie (relation II.3).

$$\varepsilon = \sum_k \left(d_k - y_k^s\right)^2 \qquad \text{(II.3)}$$

où d_k est la sortie désirée associée au vecteur d'entré x_k. y_k^s est la sortie obtenue sur la dernière couche au temps *t*. ε est l'erreur cumulée pour *k* présentations de couples (x_k, d_k)

- Retro-propagation du gradient de l'erreur δ depuis la dernière couche vers la première couche
  - Pour chaque cellule de sortie

$$\delta_i^s = -(d_i - y_i^s)F^l(p_i^s) \qquad \text{(II.4)}$$

  - Pour chaque cellule cachée :

$$\delta_i^s = -\sum_k \delta_k^{q+1} w_{ki} F^l(p_i^s) \qquad \text{(II.5)}$$

- Mise à jour des poids selon la règle :

$$W_j^q = W_j^q + \alpha(\delta_i^q x_i^q) \quad \text{(II.6)}$$

Où α est le coefficient d'apprentissage compris dans l'intervalle [0,1]

- Retour au point 2 tant qu'il y a des couples à présenter

II.2.1.2 K plus proches voisins (KNN)

La méthode des k plus proches voisins (k-Nearest Neighbors, KNN) est une méthode supervisée. Elle a été utilisée dans l'estimation statistique et la reconnaissance des modèles comme une technique non paramétrique, cela signifie qu'elle ne fait aucune hypothèse sur la distribution des données[24].

L'algorithme KNN est l'un des plus simples de tous les algorithmes d'apprentissage automatique. Il est un type d'apprentissage basé sur l'apprentissage paresseux (lazy Learning). En d'autres termes, il n'y a pas de phase d'entraînement explicite ou très minime. Cela signifie que la phase d'entraînement est assez rapide [25].

La méthode KNN suppose que les données se trouvent dans un espace caractéristique. Cela signifie que les points de données sont dans un espace métrique. Les données peuvent être des scalaires ou même des vecteurs multidimensionnels.

La méthode des k plus proches voisins est utilisée pour la classification et la régression. Dans les deux cas, l'entrée se compose des k données d'entraînement les plus proches dans l'espace de caractéristiques.

Pour trouver la classe d'un nouveau cas, cet algorithme se base sur le principe suivant : il cherche les k plus proches voisins de ce nouveau cas, ensuite, il choisit parmi Cette méthode utilise principalement deux paramètres : une fonction de similarité pour comparer les individus dans l'espace de caractéristiques et le nombre k qui décide combien de voisins influencent la classification [26].

Les candidats trouvés le résultat le plus proche et le plus fréquent [27],[28].

Pour affecter un nouvel individu à une classe, l'algorithme cherche les k plus proches voisins parmi les individus déjà classés. Ainsi, l'individu est affecté à la classe qui contient le plus d'individus parmi les candidats trouvés.

Cette méthode utilise principalement deux paramètres : une fonction de similarité pour comparer les individus dans l'espace de caractéristiques et le nombre k qui décide combien de voisins influencent la classification [29].

II.2.1.2.1 Algorithme

Exemple de classification kNN. L'échantillon d'essai (point vert) doit être classé soit en carrés bleus, soit en triangles rouges. Si k = 3 (cercle en trait plein), il est attribué aux triangles rouges car il y a 2 triangles et seulement 1 carré à l'intérieur du cercle intérieur. Si k = 5 (cercle en pointillés), il est attribué aux carrés bleus (3 carrés contre 2 triangles à l'intérieur du cercle extérieur).

Les exemples de formation sont des vecteurs dans un espace de fonctionnalités multidimensionnel, chacun avec une étiquette de classe. La phase de formation de

l'algorithme consiste uniquement à stocker les vecteurs de caractéristiques et les étiquettes de classe des échantillons de formation.

Dans la phase de classification, k est une constante définie par l'utilisateur et un vecteur non étiqueté (une requête ou un point de test) est classé en attribuant l'étiquette la plus fréquente parmi les k échantillons d'apprentissage les plus proches de ce point de requête.

Une mesure de distance couramment utilisée pour les variables continues est la distance euclidienne. Pour les variables discrètes, comme pour la classification de texte, une autre métrique peut être utilisée, telle que la métrique de chevauchement (ou distance de Hemming). Dans le contexte des données de puces à ADN d'expression génique, par exemple, KNN a été utilisé avec des coefficients de corrélation, tels que ceux de Pearson et Spearman, comme métrique. Souvent, la précision de la classification de KNN peut être considérablement améliorée si la métrique de distance est apprise à l'aide d'algorithmes spécialisés tels que l'analyse des composants du voisin le plus proche à grande marge ou du voisinage.

Un inconvénient de la classification de base du « vote majoritaire » se produit lorsque la répartition des classes est asymétrique. Autrement dit, les exemples d'une classe plus fréquente ont tendance à dominer la prédiction du nouvel exemple, car ils ont tendance à être communs parmi les k voisins les plus proches en raison de leur grand nombre. Une façon de résoudre ce problème consiste à pondérer la classification en tenant compte de la distance entre le point de test et chacun de ses k voisins les plus proches. La classe (ou la valeur, dans les problèmes de régression) de chacun des k points les plus proches est multipliée par un poids proportionnel à l'inverse de la distance entre ce point et le point de test. Une autre façon de surmonter l'asymétrie consiste à faire abstraction de la représentation des données. Par exemple, dans une carte autoorganisée (SOM), chaque nœud est un représentant (un centre) d'un groupe de points similaires, quelle que soit leur densité dans les données d'entraînement d'origine. K-NN peut ensuite être appliqué au SOM (relation II.7).

$$R^* \leq R_{KNN} \leq R^*\left(2 - \frac{MR^*}{M-1}\right) \tag{II.7}$$

Le taux d'erreur de Bayes, noté R^* est le taux d'erreur minimal possible. Le taux d'erreur asymptotique R_{KNN} du k- Plus Proche Voisin (KNN) et M est le nombre de classes dans le problème. La limite est étroite, signifiant que les limites inférieure et supérieure peuvent être atteintes par une certaine distribution. Pour $M = 2$ et comme R^* s'approche de zéro, cette limite se réduit à "pas plus de deux fois le taux d'erreur bayésien" (Figure II.2).

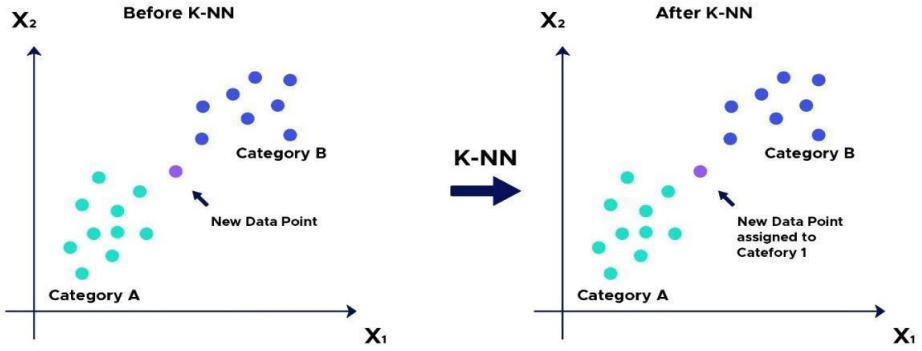


Figure II. 2: Modèle KNN

II.2.1.2.2Avantages de la méthode des k plus proches voisins

La méthode des k plus proches voisins représente des avantages tels que :

- L'algorithme KNN est robuste envers des données bruitées[30] .
- La méthode des k plus proches voisins est efficace si les données sont larges et incomplètes.
- Cette méthode est l'une des plus simples de tous les algorithmes d'apprentissage automatique.

II.2.1.2.3 Inconvénients de la méthode des k plus proches voisins

La méthode des k plus proches voisins comporte des inconvénients tels que :

- Le besoin de déterminer la valeur du nombre des plus proches voisins (le paramètre).
- Le temps de prédiction est très long puisqu'on doit calculer la distance de tous les exemples.
- Cette méthode est gourmande en espace mémoire car elle utilise une grande capacité de stockage pour le traitement des corpus[31] .

II.2.1.3 Régression logistique (RL)

La régression logistique est utilisée pour le classement et pas la régression. Mais, elle est considérée comme une méthode de régression puisqu'elle sert à estimer la probabilité d'appartenir à une classe. Il y a trois types de régression logistique :

- Régression logistique binaire : ici, le but de la classification est d'identifier si un échantillon appartient à une classe ou non.
- Régression logistique multinomiale : ici, le but de la classification est d'identifier à quelle classe appartient-t-il un échantillon parmi plusieurs classes.
- Régression logistique ordinale : ici, le but de la classification est de chercher la classe d'un échantillon parmi des classes ordonnées. Un exemple de classes : non satisfait, satisfait, très satisfait.

II.2.1.3.1 Principe de Régression logistique

Pour combiner entre les différentes caractéristiques, on utilise une fonction linéaire donnée par la fonction (II.8).

$$z(x) = \theta_0 + \theta_1 x_1 \, \theta_2 x_2 + \cdots + \theta_m x_m \qquad \text{(II.8)}$$

Cette valeur est transformée à une probabilité en utilisant la fonction logistique. Donc, la probabilité qu'un échantillon avec les caractéristiques *x1, …, xn* appartienne à une classe *y* est calculée par la relation (II.9).

$$h_\theta(x) = p\left(y = 1 \,\middle|\, x\right) = \frac{1}{1+e^{-z(x)}} \qquad \text{(II.9)}$$

Les valeurs de la fonction logistique sont comprises entre 0 et 1; comme il est indiqué dans le diagramme de la figure II.3.

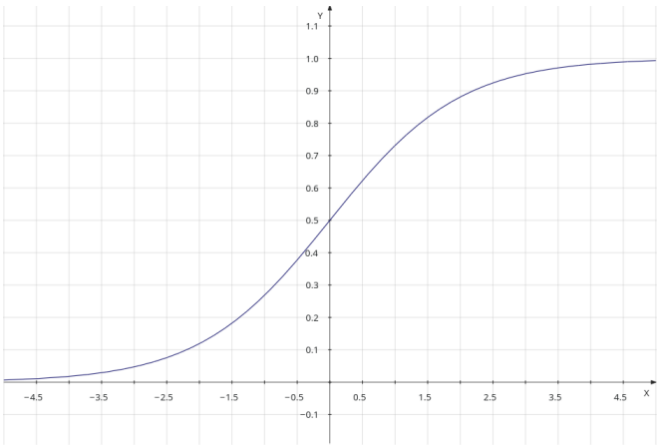


Figure II. 3 : Valeurs de la fonction logistique sont comprises entre 0 et 1

II.2.1.3. Décision

Pour prédire si un échantillon *x* appartient à une classe donnée (classe positive) *y=1*, on calcule sa probabilité en utilisant l'équation précédente. Ensuite, on utilise un seuil sur cette probabilité pour décider.

On peut utiliser le seuil **0.5** Dans ce cas :

- Si $p(y = 1|x) >= 0.5$ donc classe positive
- Sinon classe négative

En cas de plusieurs classes, on utilise une stratégie d'un-contre-le-reste. On entraine plusieurs classifieurs, chacun pour une classe. Pour décider quelle est la classe d'un échantillon, on prend celle avec la probabilité la plus élevée.

II.2.1.3.3Fonction du coût

L'erreur quadratique moyenne (MSE) ne peut pas être utilisée comme dans la régression linéaire. Ceci est dû au fait que la fonction de prédiction est non linéaire. La fonction du coût va être non-convexe avec plusieurs minimums locaux. Lors de la minimisation, on

peut tomber sur un minimum local et l'algorithme du gradient va s'arrêter sans converger vers la solution optimale.

Dans ce cas, on utilise l'entropie croisée. Etant donnée un ensemble de données avec *N* échantillons, où le résultat *y* est soit 1 ou 0. La fonction du coût est calculée comme suit, où *(i)* réfère au i-ème échantillon dans les données d'entrainement (formule (II.10)).

$$J = \frac{1}{N}\sum_{i=1}^{N} \begin{cases} -\log(h_\theta(x^{(i)})) & si\ y^{(i)} = 1 \\ -\log(1 - h_\theta\,(x^{(i)})) & si\ \ y^{(i)} = 1 \end{cases} \quad \text{(II.10)}$$

Puisque *y* peut prendre seulement les deux valeurs 0 et 1, cette fonction peut être simplifiée comme (II.11).

$$J = -\frac{1}{N}\sum_{i=1}^{N}(y^{(i)} \log h_\theta\,(x^{(i)})) + (1 - y^{(i)}) \log(\,1 - h_\theta\,(x^{(i)})) \quad \text{(II.11)}$$

II.2.1.3.4Gradients

Le gradient de chaque poids *θj* est calculé en utilisant le dérivé partiel de la fonction du coût par rapport à ce poids. Donc, le gradient d'un poids *θj* est calculé par (II.12).

$$\frac{\partial J}{\partial \theta_j} = \frac{1}{N}\sum_{i=1}^{N} x_j^{(i)}(h_\theta\,(x^{(i)}) - y^{(I)} \quad \text{(II.12)}$$

II.2.1.3.5 Avantages de la régression logistique

- **Simplicité** : Facile à comprendre et à implémenter.
- **Interprétabilité** : Coefficients interprétables pour comprendre l'impact des variables.
- **Efficacité** : Performante pour des ensembles de données de taille modérée.
- **Probabilités prédictives** : Fournit des probabilités pour les classes.
- **Régularisation** : Peut utiliser L1 et L2 pour éviter le surapprentissage.

II.2.1.3.6Inconvénients de la régression logistique

- **Linéarité** : Suppose une relation linéaire entre les variables et le logis.
- **Biais et variance** : Sensible aux hypothèses de linéarité et d'indépendance des erreurs.
- **Classes déséquilibrées** : Performances médiocres avec des classes très déséquilibrées.
- **Scalabilité** : Difficile à utiliser avec des ensembles de données très volumineux.
- **Sensibilité aux outlier**: Les coefficients peuvent être influencés par les outliers.

II.2.1.4 Random Forest

II.2.1.4.1 Définition

Random Forest est un algorithme d'apprentissage automatique supervisé polyvalent et puissant qui combine plusieurs arbres de décision pour former une "forêt". Il est utilisé pour la classification et la régression, et est populaire parmi les "Data Scientists" pour sa facilité d'interprétation, sa stabilité et ses bons résultats, couvrant ainsi de nombreux problèmes de Machine Learning [32][33].

II.2.1.4.2 Fonctionnement de l'algorithme Random Forest

Random Forest combine plusieurs arbres de décision pour améliorer la précision des prédictions. Chaque arbre donne une classification ou un vote, et la forêt choisit la classification majoritaire pour la classification, ou la moyenne des sorties pour la régression. La faible corrélation entre les arbres individuels permet au modèle global de compenser les erreurs individuelles, améliorant ainsi la fiabilité des prédictions [34][32].

II.2.1.4.3 Apprentissage automatique supervisé

L'apprentissage automatique supervisé est lorsque l'algorithme (ou le modèle) est créé en utilisant ce qu'on appelle un ensemble de données de formation. Le modèle est formé en utilisant de nombreux exemples différents de diverses entrées et sorties, et apprend ainsi à classer toutes les nouvelles données d'entrée qu'il reçoit à l'avenir. C'est ainsi que les algorithmes sont utilisés pour prédire les résultats futurs.

II.2.1.4.3.1 Régression et classification dans l'apprentissage automatique

Dans l'apprentissage automatique, les algorithmes sont utilisés pour classer certaines observations, événements ou entrées en groupes. Par exemple, un filtre anti-spam par courriel classera chaque courriel comme étant « spam » ou « non spam ». Toutefois, l'exemple par courriel n'est qu'un exemple simple ; dans un contexte commercial, les pouvoirs prédictifs de ces modèles peuvent avoir un impact majeur sur la façon dont les décisions sont prises et la façon dont les stratégies sont formées, mais plus à ce sujet plus tard.

Ainsi, La régression et la classification sont toutes deux des problèmes d'apprentissage automatique supervisés utilisés pour prédire la valeur ou la catégorie d'un résultat ou d'un résultat. Dans l'analyse de classification, l'attribut dépendant est catégorique. Les tâches de classification apprennent à attribuer une étiquette de classe à des exemples du domaine problématique. Comme mentionné précédemment, un exemple courant de classification est le filtre anti-spam de votre e-mail.

Dans l'analyse de régression, l'attribut dépendant est numérique à la place. La régression est utilisée lorsque la variable de sortie est une valeur réelle ou continue comme le salaire, l'âge ou le poids. Pour une façon simple de faire la distinction entre les deux, n'oubliez pas que la classification est de prédire une étiquette (p. ex. « spam » ou « non spam ») alors que la régression est de prédire une quantité.

II.2.1.4.4Arbres de décision

Comme nous le savons, le modèle Random Forest pousse et combine plusieurs arbres de décision pour créer une « forêt ». Comme montré dans la Figure (II.4) un arbre de décision est un autre type d'algorithme utilisé pour classer les données. En termes très simples, vous pouvez le penser comme un Filochant qui trace une voie claire vers une décision ou un résultat, il commence à un seul point, puis se ramifie dans deux directions ou plus, chaque branche de l'arbre de décision offrant des résultats différents possibles.

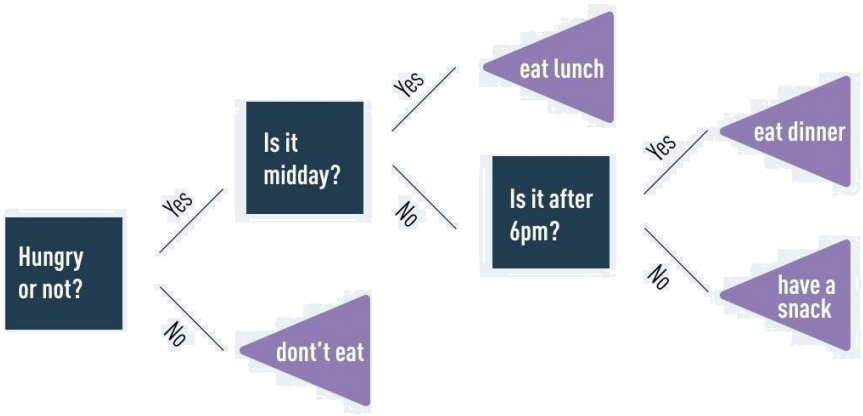


Figure II. 4: Exemple d'un arbre de décision

La classification est une branche importante et très précieuse de la science des données, et Random Forest est un algorithme qui peut être utilisé pour de telles tâches de classification. L'ensemble d'arbres de Random Forest produit le mode ou la moyenne des arbres individuels. Cette méthode permet des résultats plus précis et stables en s'appuyant sur une multitude d'arbres plutôt que sur un seul arbre de décision.

II.2.1.4.5 Fonctionnement de l'algorithme

Random Forest cultive plusieurs arbres de décision qui sont fusionnés pour une prédiction plus précise. La logique le modèle Random Forest est que plusieurs modèles non corrélés (les arbres de décision individuels) fonctionnent beaucoup mieux en tant que groupe qu'ils ne le font seuls. Lors de l'utilisation de Random Forest pour la classification, chaque arbre donne une classification ou un « vote ». La forêt choisit le classement avec la majorité des « votes ». Lors de l'utilisation de Random Forest pour la régression, la forêt choisit la moyenne des sorties de tous les arbres.

La clé ici réside dans le fait qu'il existe une faible corrélation (ou pas) entre les modèles individuels, c'est-à-dire entre les arbres de décision qui composent le modèle Random Forest plus large. Bien que les arbres de décision individuels puissent produire des erreurs, la majorité du groupe aura raison, déplaçant ainsi le résultat global derrière dans la bonne direction.

II.2.1.4.5.1Processus de formation des arbres dans une forêt aléatoire

Les arbres de décision d'un ensemble, comme les arbres d'une forêt aléatoire, sont habituellement formés selon la méthode de bagging (d'ensachage) La méthode d'ensachage est un type d'algorithme d'apprentissage automatique d'ensemble appelé Agrégation Bootstrap. Une méthode d'ensemble combine les prédictions de plusieurs algorithmes d'apprentissage automatique ensemble pour faire des prédictions plus précises qu'un modèle individuel. Random Forest est également une méthode d'ensemble.

Bootstrap effectue aléatoirement l'échantillonnage en ligne et l'échantillonnage des fonctionnalités à partir de l'ensemble de données pour former des ensembles de données d'échantillons pour chaque modèle. L'agrégation réduit ces ensembles de données d'échantillons en statistiques sommaires basées sur l'observation et les combine. Bootstrap Agrégation peut être utilisé pour réduire la variance des algorithmes de variance élevée tels que les arbres de décision.

La variance est une erreur résultant de la sensibilité aux petites fluctuations de l'ensemble de données utilisées pour la formation. Une variance élevée provoquera un algorithme pour modéliser des données non pertinentes, ou du bruit, dans l'ensemble de données au lieu des sorties prévues, appelées signal. Ce problème est appelé le overfitting (surapprentissage). Un modèle sur équité sera performant à l'entraînement, mais ne sera pas en mesure de distinguer le bruit du signal dans un test réel. La méthode du bagging est l'application de la méthode Bootstrap à un algorithme d'apprentissage automatique à haute variance.

II.2.1.4.5.2Multiples applications de l'algorithme Random Forest

La forêt aléatoire est utilisée sur le tas par les scientifiques de données dans beaucoup d'industries comprenant la banque, le commerce d'actions, la médecine, et le commerce électronique. Il est utilisé pour prédire les choses qui aident ces industries à fonctionner efficacement, comme l'activité des clients, les antécédents des patients et la sécurité.

Random Forest est utilisé dans les services bancaires pour détecter les clients qui sont plus susceptibles de rembourser leur dette à temps. Il est également utilisé pour prédire qui utilisera les services d'une banque plus fréquemment. Ils l'utilisent même pour détecter la fraude. Parlez du robin des bois des algorithmes !

Les traders utilisent Random Forest pour prédire le comportement futur d'une action. Il est utilisé par les entreprises de détail pour recommander des produits et prédire la satisfaction de la clientèle ainsi. Les scientifiques chinois ont utilisé Random Forest pour étudier les modes de combustion spontanés du charbon afin de réduire les risques pour la sécurité dans les mines de charbon.

Dans le domaine des soins de santé, Random Forest peut être utilisé pour analyser les antécédents médicaux d'un patient afin d'identifier les maladies. Les scientifiques pharmaceutiques utilisent Random Forest pour identifier la bonne combinaison de composants dans un médicament ou prédire la sensibilité aux médicaments. Parfois, Random Forest est même utilisé pour la biologie computationnelle et l'étude de la génétique.

II.2.1.4.6 Avantages de Random Forest

Les principaux avantages de l'utilisation de Random Forest sont les plus importants :

- Facilité d'utilisation
- Efficacité
- Précision
- Polyvalence – peut être utilisée pour la classification ou la régression
- Plus convivial pour les débutants que les algorithmes tout aussi précis comme les filets neuronaux.

II.2.1.4.7 Inconvénients de Random Forest

Bien que Random Forest présente peu d'inconvénients, son utilisation de nombreux arbres de décision peut nécessiter beaucoup de mémoire sur de grands projets, le rendant parfois plus lent que d'autres algorithmes plus efficaces. De plus, comme il est basé sur les arbres de décision, il peut être sujet à un surajustement, bien que ce soit généralement évité grâce

à l'utilisation de sous-ensembles aléatoires de fonctionnalités pour construire des arbres plus petits, ce qui peut ralentir le traitement mais augmenter la précision.

II.2.2 Node-Red

II.2.2.1 Définition

Node-RED est un outil de programmation visuelle open-source qui permet de connecter des périphériques, des APIs et des services en ligne de manière intuitive. Ce chapitre va aborder les concepts fondamentaux de Node-RED, de sa configuration à son futilisation pratique[35].

II.2.2.1Concepts Fondamentaux d'un Node-Red

Node-RED repose sur deux concepts clés : les nœuds (nodes) et les flux (flows). Les nœuds représentent des actions ou des éléments de données, tandis que les flux définissent la séquence d'exécution des nœuds. Comprendre ces concepts est essentiel pour tirer pleinement parti de Node-RED.

II.2.2.3 Création de Flows avec Node-RED

L'une des forces de Node-RED réside dans sa capacité à créer des flux complexes en quelques clics. Dans ce chapitre, nous allons explorer comment créer et connecter des nœuds pour automatiser des tâches, récupérer des données à partir de sources multiples et contrôler des appareils connectés.

II.2.2.4 Personnalisation et Extensibilité de Node-RED

Node-RED est hautement personnalisable grâce à son écosystème de nœuds supplémentaires. Dans ce chapitre, nous verrons comment étendre les fonctionnalités de Node-RED en ajoutant des nœuds tiers et en créant nos propres nœuds personnalisés pour répondre à des besoins spécifiques.

II.2.2.5 Déploiement et Gestion des Flows

Une fois que vous avez créé vos flows, il est temps de les déployer et de les gérer efficacement. Nous aborderons les meilleures pratiques pour déployer des flows sur des serveurs, les importer et les exporter, ainsi que surveiller et optimiser les performances de vos applications Node-RED.

En combinant ces aspects fondamentaux, vous serez en mesure de maîtriser Node-RED et de tirer pleinement parti de ses capacités pour automatiser des processus, connecter des systèmes et concevoir des applications IoT innovantes.

II.2.2.6 Créer un Dashboard

Un Dashboard est une interface graphique composée de plusieurs éléments, chacun représenté par un Node. On peut les diviser en deux types : éléments d’entrée et éléments de sortie. Avec le module node-red-dashboard, vous avez accès à tous les éléments d’entrée suivants :

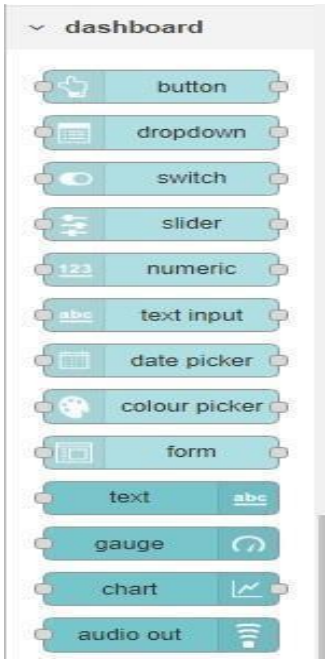


Figure II. 5: Quelques nœuds du Dashboard de Node-RED

Les nœuds du tableau de bord ne sont pas préinstallés. Pour les installer, ces deux procédures doivent être suivies :

- Tout d'abord, les trois lignes horizontales doivent être cliquées, puis la gestion de la palette sélectionnée.

- Ensuite, "Dashboard" doit être recherché dans la barre de recherche et l'option "Installer" doit être sélectionnée. Après quelques minutes, les nœuds seront ajoutés à la palette de gauche (voir Figures II.6 et II.7).

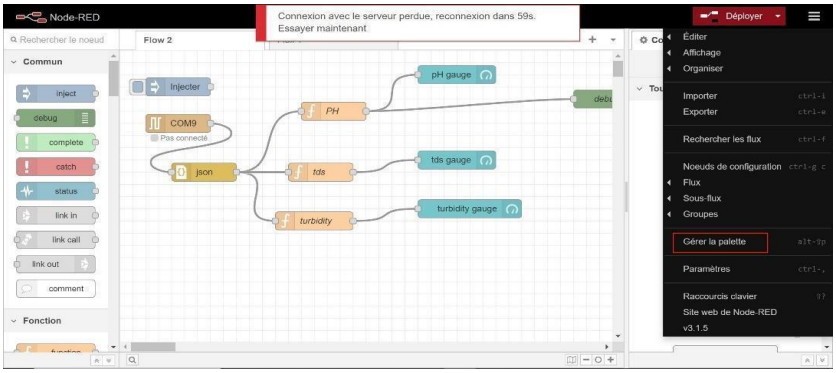


Figure II. 6: Installation du tableau de bord

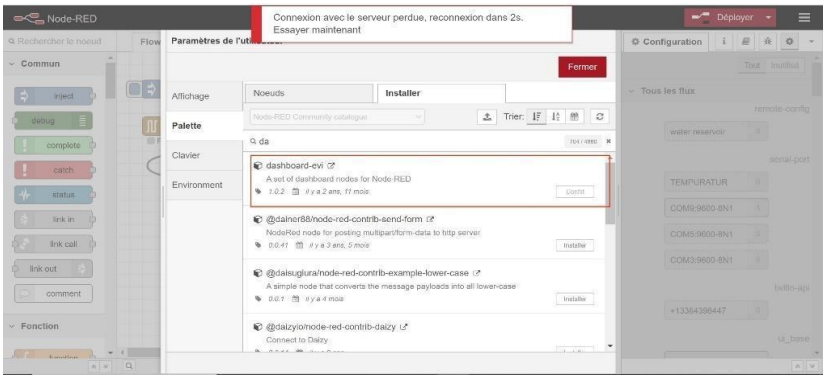


Figure II. 7: Tableau de bord installé

La figure II.8 montre un exemple d'une IHM réalisée à l'aide de Node-RED.

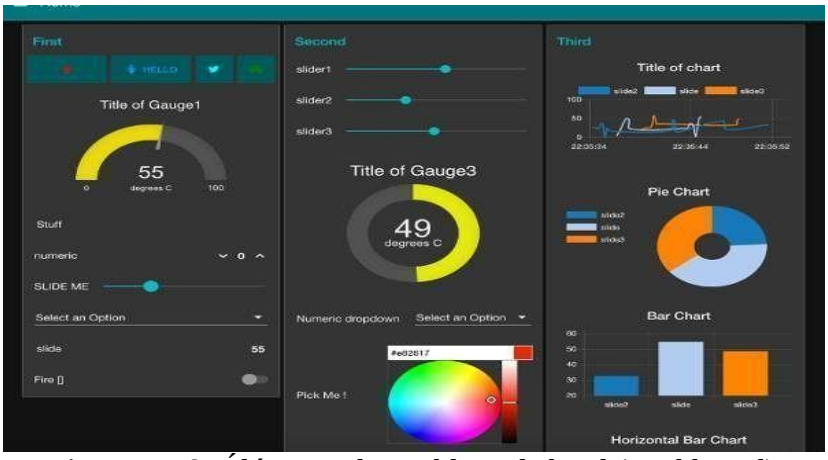


Figure II. 8: Éléments d'un tableau de bord (Dashboard**)**

II.2.3 Bibliothèques python utiles

Pour la phase d'apprentissage et de tests, le langage Python est utilisé, en tirant parti de nombreuses bibliothèques open source.

II.2.3.1 Pandas

Pandas Data Frame est une structure de données tabulaire bidimensionnelle variable en taille, potentiellement hétérogène, avec des axes étiquetés (lignes et colonnes). Une trame de données est une structure de données bidimensionnelle, c'est-à-dire que les données sont alignées de manière tabulaire en lignes et en colonnes. Pandas Data Frame se forme de trois composants principaux, les données, les lignes et les colonnes.

II.2.3.2 NumPy

Un projet open source permettant le calcul numérique avec Python. Il a été créé en 2005, en s'appuyant sur les premiers travaux des bibliothèques Numericable et Numarray. Il est toujours 100% open source, gratuit pour les utilisateurs (« NumPy ») [36].

II.2.3.3 Sklearn

Scikit-learn est une bibliothèque clé pour le langage de programmation Python qui est généralement utilisé dans les projets d'apprentissage automatique. Scikit-learn se concentre sur les outils d'apprentissage automatique, y compris les algorithmes mathématiques, statistiques et à usage général qui constituent la base de nombreuses technologies d'apprentissage automatique. En tant qu'outil gratuit, Scikit-learn est extrêmement

important dans de nombreux types de développement d'algorithmes pour l'apprentissage automatique et les technologies associées.

II.2.3.4 Tensorflow

Tensorflow est un Framework open source, développé par des chercheurs de Google pour exécuter l'apprentissage automatique, l'apprentissage en profondeur et d'autres charges de travail d'analyse statistique et prédictive. Comme des plates-formes similaires, il est conçu pour rationaliser le processus de développement et d'exécution d'applications d'analyse avancées pour les utilisateurs tels que les scientifiques des données, les statisticiens et les modélisateurs prédictifs. Le logiciel Tensorflow gère les ensembles de données qui sont organisés en nœuds de calcul sous forme de graphe. Les arêtes qui relient les nœuds d'un graphe peuvent représenter des vecteurs ou des matrices multidimensionnelles, créant ce que l'on appelle des tenseurs. Parce que les programmes Tensorflow utilisent une architecture de

Flux de données qui fonctionne avec des résultats intermédiaires généralisés des calculs, ils sont particulièrement ouverts aux applications de traitement parallèle à très grande échelle, les réseaux de neurones étant un exemple courant. Des calculs, ils sont particulièrement ouverts aux applications de traitement parallèle à très grande échelle, les réseaux de neurones étant un exemple courant. Des calculs, ils sont particulièrement ouverts aux applications de traitement parallèle à très grande échelle, les réseaux de neurones étant un exemple courant.

II.3 Outils Hardware

II.3.1 Arduino

Le module Arduino est un circuit imprimé en matériel libre (plateforme de contrôle) dont les plans de la carte elle-même sont publiés en licence libre dont certains composants de la carte : comme le microcontrôleur et les composants complémentaires qui ne sont pas en licence libre. Un microcontrôleur programmé peut analyser et produire des signaux électriques de manière à effectuer des tâches très diverses. Arduino est utilisé dans beaucoup d'applications comme l'électrotechnique industrielle et embarquée ; le modélisme, la domotique mais aussi dans des domaines différents comme l'art contemporain et le pilotage d'un robot, commande des moteurs et faire des jeux de lumières, communiquer avec l'ordinateur, commander des appareils mobiles (modélisme). Chaque module d'Arduino possède un régulateur de tension +5 V et un oscillateur à quartez 16 MHz (ou un résonateur céramique dans certains modèles). Pour programmer cette carte, on utilise l'logiciel IDE Arduino. Figure (II.9) [37]



Figure II. 9: Carte Arduino Uno

II.3.1.1 Caractéristiques principales

Arduino comme microntrôleur, possède les caractéristiques suivantes :

- Alimentation via port USB (Programmer ou peur être alimenter) ou 7 à 12 V sur connecteur Jack DC 5,5 x 2,1 mm
- Microcontrôleur ATMega328
- Un oscillateur à quartz 16 MHz.
- Mémoire flash 32 Kb, mémoire SRAM 2 kb et mémoire EEPROM 1 Kb.
- 14 broches d'E/S dont 6 PWM (Pulse Wirth Modulation)
- 6 entrées analogiques 10 bits.
- Bus série I2C et SPI.
- Gestion des interruptions, bouton Reset.
- Régulateur de tension

La figure (II.10) présente les composants de la carte Arduino.

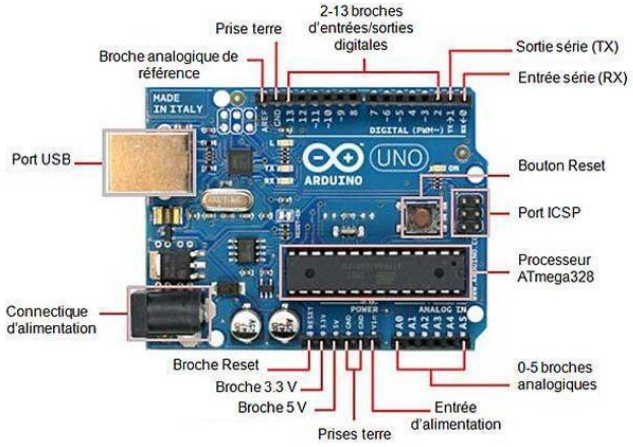


Figure II. 10: Composants principaux de l'Arduino

II.3.1.2Environnement de la programmation

Le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche du C). Une fois, le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers de la liaison USB. Le câble USB alimente à la fois en énergie la carte et transporte aussi l'information ce programme appelé IDE Arduino [38].

II.3.1.3 Structure générale du programme (IDE Arduino) :

Comme n'importe quel langage de programmation, une interface souple et simple est exécutable sur n'importe quel système d'exploitation Arduino basé sur la programmation en C.

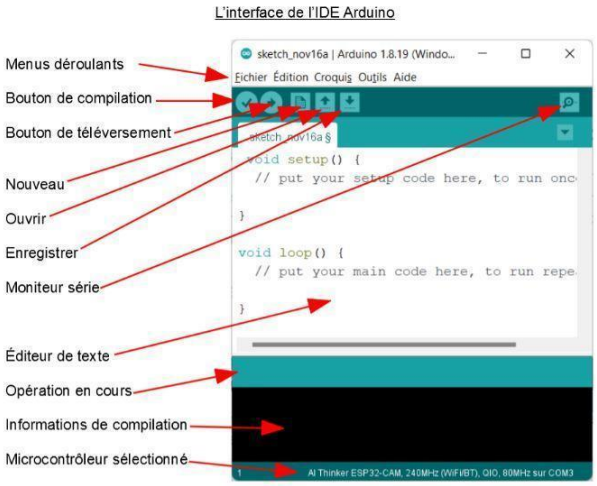


Figure II.11: Interface IDE d'Arduino

II.3.2 Capteurs utilisés

Il existe plusieurs capteurs qu'on peut utiliser pour surveiller et contrôler la qualité de l'eau. Parmi les plus courants, on retrouve le capteur de pH, de turbidité et celui de conductivité électrique. Ils sont cruciaux pour détecter les anomalies et maintenir des normes de qualité adéquates dans la gestion de l'eau.

II.3.2.1 Capteur de PH

Un pH-mètre analogique, spécialement conçu pour les contrôleurs Arduino et a connexion simple, commode et pratique et les caractéristiques. Il dispose d'une LED qui fonctionne comme le témoin d'alimentation, un connecteur et PH2.0 interface de capteur BNC (Figure II.12).

Pour l'utiliser, il suffit de connecter le capteur de pH avec connecteur BNC, et branchez l'interface PH2.0 dans le port d'entrée analogique d'un contrôleur Arduino. Des préprogrammé, vous obtiendrez la valeur du pH facilement. Livré dans une boîte en plastique compact avec des mousses pour un meilleur stockage mobile[39].



Figure II. 12: Analog pH Meter Kit (SKU: SEN0161) [39]

II.3.2.1.1 Spécification du capteur pH

Le capteur de pH se distingue par les éléments suivants :

- Module d'alimentation : 5.00V
- Taille du module : 43mm × 32mm
- Plage de mesure : 014 PH
- Mesure de la température : 060°C
- Précision : ± 0.1pH (25 °C)

- Temps de réponse : ≤ 1 min
- Capteur de pH avec connecteur BNC
- Interface PH2.0 (3 pieds patch)
- Réglage du gain Potentiomètre
- Indicateur d'alimentation LED
- Longueur du câble du capteur au connecteur BNC : 660mm

II.3.2.1.2 Caractéristiques du capteurs pH

La sortie de l'électrode de pH est millivolts, et la valeur du pH de la relation est représentée comme suit (25 °C) :

Tableau II. 1:Relation entre la sortie de l'électrode de pH et sa valeur

Tension(mv)	Valeur de pH	Tension (mv)	Valeur de Ph
414.12	0.00	-414.12	14.00
354.96	1.00	-354.96	13.00
295.80	2.00	-295.80	12.00
236.64	3.00	-236.64	11.00
177.48	4.00	-177.48	10.00
118.32	5.00	-118.32	9.00
59.16	6.00	-59.16	8.00
0.00	7.00	0.00	7.00

II.3.2.2 Capteur de turbidité

Le niveau de turbidité de l'eau est mesuré par un capteur de turbidité destiné lui aussi à la détection de la qualité de l'eau à travers des mesures. Il a la capacité de détecter les particules pendantes dans l'eau. Cela se fait en mesurant le taux de transmission et de diffusion de la lumière, taux de diffusion ; cela permet de déterminer les changements dans la quantité de solides en suspension (TSS) dans l'eau (Figure II.13)[40].



Figure II. 13: Capteur de turbidité

III.3.2.2.1Comprendre la turbidité

La turbidité est la nébulosité ou le flou d'un fluide provoqué par un grand nombre de particules individuelles généralement invisibles à l'œil nu, semblables à de la fumée dans l'air. La mesure de la turbidité est un test clé de la qualité de l'eau (Figure III.14).[41]



Figure II. 14: Exemple de turbidité (gauche <10 NTU ; milieu = 200 NTU ; droit = 1500 NTU)

La turbidité est causée par des particules en suspension ou dissoutes dans l'eau qui diffusent la lumière, donnant à l'eau un aspect trouble ou trouble. Les matières particulaires peuvent inclure des sédiments, en particulier de l'argile et du limon, de fines matières organiques et inorganiques, des composés organiques colorés solubles, des algues et d'autres organismes microscopiques.

III.3.2.2.2 Impact de la turbidité

Une turbidité élevée peut réduire considérablement la qualité esthétique des lacs et des cours d'eau. Cela peut augmenter le coût du traitement de l'eau potable et de la transformation des aliments. Il peut nuire aux poissons et à d'autres organismes aquatiques en réduisant les réserves de nourriture, en dégradant les frayères et en affectant la fonction des branchies.

III.3.2.2.2.1 Sources de turbidité

Les sédiments arrivent souvent en tête de liste des substances ou polluants responsables de la turbidité. Les sources naturelles peuvent inclure l'érosion provenant des zones de hautes terres, des zones riveraines, des berges et des chenaux des cours d'eau. Les algues qui se développent en se nourrissant des nutriments entrant dans le cours d'eau par la décomposition des feuilles ou d'autres processus de décomposition naturels peuvent également être une source de turbidité.

Le mouvement des cours d'eau peut également libérer des sédiments. Les matières organiques provenant des rejets d'eaux usées, en particulier lors des contournements des stations d'épuration, peuvent contribuer à la turbidité. Les activités humaines qui perturbent les terres, telles que la construction, l'exploitation minière et l'agriculture, peuvent entraîner des niveaux élevés de sédiments pénétrant dans les plans d'eau pendant les tempêtes de pluie en raison du ruissellement des eaux pluviales.

III.3.2.2.2.2 Mesurer la turbidité

La turbidité est mesurée à l'aide d'un équipement optique spécialisé en laboratoire ou sur le terrain. La lumière est dirigée à travers un échantillon d'eau et la quantité de lumière diffusée est mesurée.

L'unité de mesure est appelée unité de turbidité néphélométrie (NTU), qui se décline en plusieurs variantes. Plus la diffusion de la lumière est grande, plus la turbidité est élevée. De faibles valeurs de turbidité indiquent une clarté d'eau élevée ; des valeurs élevées indiquent une faible clarté de l'eau.

II.3.2.3 Capteur de conductivité électrique

La capacité d'une solution à faire passer ou à transporter un courant électrique est appelée conductivité de la solution. La mesure de la conductivité spécifique est importante pour déterminer les impuretés dans l'eau. Les particules solides dissoutes totales (TDS) dans l'eau, déterminent la quantité de sels et de minéraux existant dans l'eau, Figure (II.15).[42]

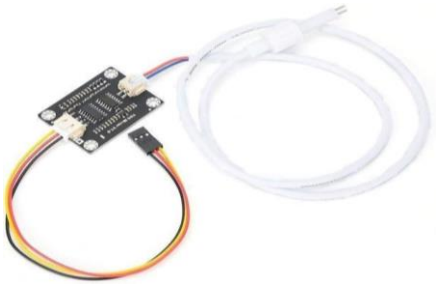


Figure II. 15: Capteur de conductivité

II.3.2.3.1 Caractéristiques du capteur TDS

- Tension d'entrée : cc 3,3 ~ 5,5 V.
- Tension de sortie : 0 ~ 2,3 V
- Courant de fonctionnement : 3 ~ 6 mA
- Plage de mesure TDS : 0 ~ 1000 ppm
- Précision de mesure TDS : ± 10 % FS (25 ℃)
- Interface de modules : XH2.54-3P
- Interface d'électrode : XH2.54-2P

II.3.2.3.2 Composants de capteur TDS

II.3.2.3.2.1 Sonde TDS

- - Fabriquée en acier inoxydable pour la résistance à la corrosion.
- - Longueur du câble : Généralement 1 mètre.
- - Connecteur : Souvent BNC.

II.3.2.3.2.2 Module de Capteur TDS

- - Convertisseur de signal analogique.
- - Potentiomètre pour ajuster la précision.
- - Connecteurs pour la sonde TDS, alimentation et sortie analogique.

II.4 Conclusion

Dans ce chapitre, nous avons examiné l'intégration de diverses technologies logicielles telles que Node-RED et Python avec des algorithmes d'apprentissage automatique, y compris les RNA, les k-plus proches voisins (K-NN), Régression logistique (RL) et les forêts aléatoires (RF). Ces outils, combinés aux capteurs de pH, de turbidité et de conductivité électrique TDS, ainsi qu'aux plateformes matérielles comme Arduino, ouvrent la voie à des solutions innovantes et efficaces pour surveiller et gérer la qualité de l'eau.

Chapitre III

Réalisation pratique et déploiement

III.1 Introduction

Après avoir introduit les outils utilisés, tant matériels que logiciels, ce chapitre se penchera sur une exploration détaillée de la mise en œuvre technique et pratique de la solution envisagée : un système connecté intelligent de contrôle qualité de l'eau potable basé sur l'IA. Ce chapitre aborde les aspects cruciaux de la partie pratique du projet, présentant les étapes de conception et de réalisation comme suit :

- Cahier des charges
- Solution proposée
- Connexions des entrées et sorties avec l'Arduino
- Synthèse du modèle intelligent
- Mise en œuvre de l'IHM
- Tests et validation

III.2 Cahier des charges

Dans le cadre de la réalisation de notre système intelligent de contrôle qualité d'eau connecté, nous avons défini les grandes lignes qui serviront de référence pour notre projet, clarifiant ainsi ses objectifs de manière précise.

- Acquisition des données à l'aide de trois capteurs en tant qu'entrées : capteur de pH, capteur de turbidité et capteur de conductivité électrique TDS.

- Les données acquises par les différents capteurs, nécessitant un microcontrôleur, ont été traitées à l'aide d'Arduino dans notre cas. Ce dernier est également responsable de l'envoi des données vers un PC ou via Wifi pour un contrôle en temps réel.

- La solution envisagée est mise en œuvre dans une plateforme combinant l'intelligence artificielle et l'internet des objets à base de la plateforme Node-RED.

Les données utilisées pour le contrôle de la qualité de l'eau subissent plusieurs opérations du point de vue matériel, notamment la collecte, le transfert, le traitement et l'interfaçage.

III.2.1 Collecte De Données

Dans cette première étape, les trois capteurs sont placés afin que chacun puisse collecter une valeur de paramètre pertinente dans l'échantillon d'eau. Ces données sont ensuite transmises du processus de collecte au traitement.

III.2.2 Transferts de données

Pendant cette étape, les données sont lues à partir des capteurs puis transmises pour être surveillées dans l'interface via une communication série. Ce processus représente un point de connexion crucial entre les périphériques matériels et les services logiciels.

III.2.3 Traitement des données

Cette étape est d'une importance capitale, car elle englobe la réception des données, la prise de décision concernant la potabilité de l'eau en utilisant un modèle intelligent, ainsi que la décision éventuelle d'envoyer des e-mails d'alerte et des notifications. Ainsi, au

cœur de cette phase, nous mettons en lumière les nombreux avantages de l'IoT et de Node-RED.

III.2.4 Interfaçage des données

Dans cette phase finale, une interface est élaborée à l'aide de Node-RED pour visualiser en temps réel les données provenant des dispositifs externes sur un navigateur Web et/ou une application mobile. En cas de variations anormales des paramètres de l'eau, cette interface déclenche une alerte en affichant un message vocal sur l'écran, en envoyant des e-mails, des notifications et des SMS en l'absence d'accès à Internet, assurant ainsi que tout le personnel est informé et conscient de l'état actuel de l'eau en temps réel.

III.3 Système proposé :

Les paramètres clés surveillés dans le système proposé sont la conductivité, la turbidité et le pH. Le schéma fonctionnel du système proposé est présenté sur la figure III.1

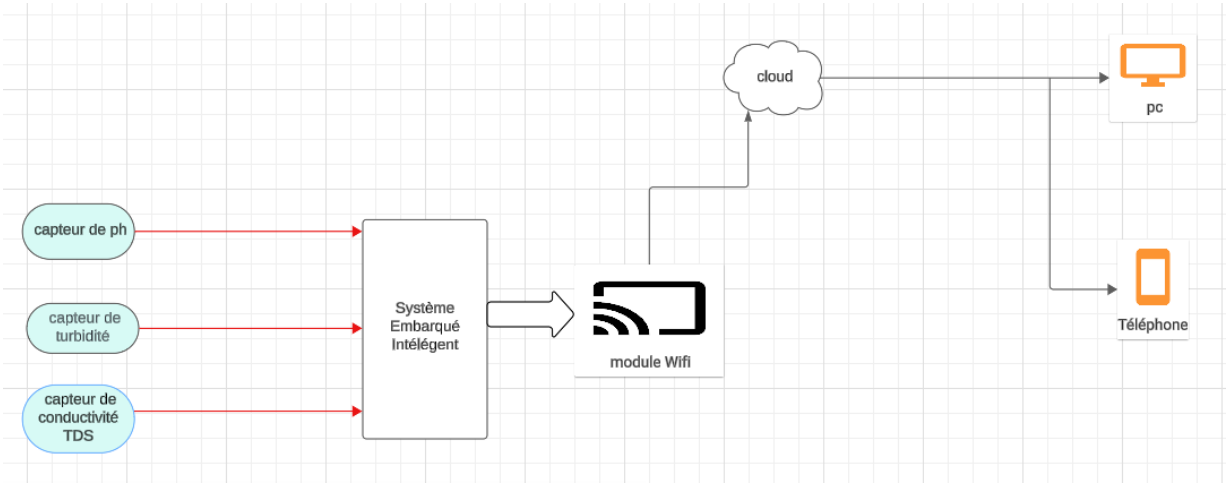


Figure III. 1: Schéma fonctionnel du système de surveillance de la qualité de l'eau

Un contrôleur constitue la partie centrale du système de surveillance de la qualité de l'eau compatible avec la solution IoT. Les capteurs sont directement interfacés au contrôleur puisque le système proposé est de surveiller qualité de l'eau domestique. Les paramètres du capteur tels que la conductivité, la turbidité et le pH sont mesurés en plaçant le capteur dans différentes solutions d'eau. Les données des capteurs sont envoyées vers le cloud à l'aide du contrôleur. Le seuil est défini dans le cloud sur la base des normes fournies par l'OMS. Le message est envoyé du cloud au mobile de l'utilisateur si la valeur dépasse le seuil. Une application mobile a été développée dans laquelle les valeurs obtenues par chaque capteur dans le cloud peuvent être visualisé. Cela peut être utilisé à la fois par la qualité de l'eau autorités de contrôle ainsi que les utilisateurs.

Le contrôleur est lié aux différents capteurs tels que le capteur de pH, le capteur de conductivité, le capteur de turbidité et de nombreux capteurs peuvent être ajoutées. Les fils du capteur sont placés dans l'eau à tester. Les valeurs du capteur, de grandeurs analogiques, seront traitées par un ADC disponible au niveau du contrôleur qui lit la valeur et elle sera téléchargée par la suite sur le cloud. Les valeurs seront surveillées en permanence à l'aide d'un modèle intelligent.

III.4 Interfaçage des capteurs

Trois capteurs sont utilisés : le pH-*mètre*, celui turbidité et celui de la TDS.

III.4.1 pH-mètres

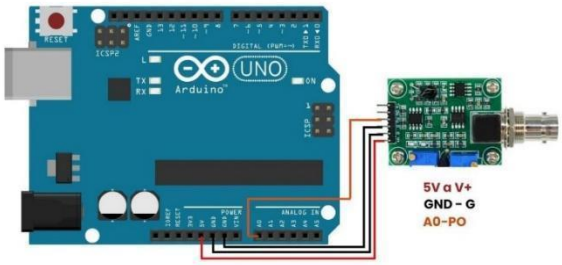


Figure III. 2: pH-mètre utilisant Arduino Uno

Dans ce projet, nous allons concevoir un pH-mètre à l'aide d'Arduino et mesurer le pH d'une solution liquide à l'aide d'un capteur de pH gravitationnel.

III.4.1.1 Composants requis

- Arduino Uno
- Capteur de pH analogique par gravité
- Fils de connexion

III.4.1.2 Analyse du fonctionnement du capteur de pH analogique par gravité

Le capteur de pH analogique est conçu pour mesurer la valeur du pH d'une solution et afficher l'acidité ou l'alcalinité de la substance. Il est couramment utilisé dans diverses applications telles que l'agriculture, le traitement des eaux usées, les industries, la surveillance de l'environnement, etc. Le module dispose d'une puce de régulateur de tension intégrée qui prend en charge la large alimentation en tension de 3,3 à 5,5 V CC, compatible avec 5 V et 3,3 V de n'importe quelle carte de contrôle comme Arduino. Le signal de sortie est filtré par un faible jeter matériel.

III.4.1.2.1Description de la broche :

- V+ : entrée 5 V CC
- G : broche de masse
- Po : sortie analogique pH
- D_O : sortie 3,3 V CC
- T_O : Sortie de température

III.4.1.3 Interface Arduino avec le diagramme du capteur PH et son fonctionnement

Comme le montre la figure III.4, les trois broches du capteur de pH sont connectées à la carte Arduino.

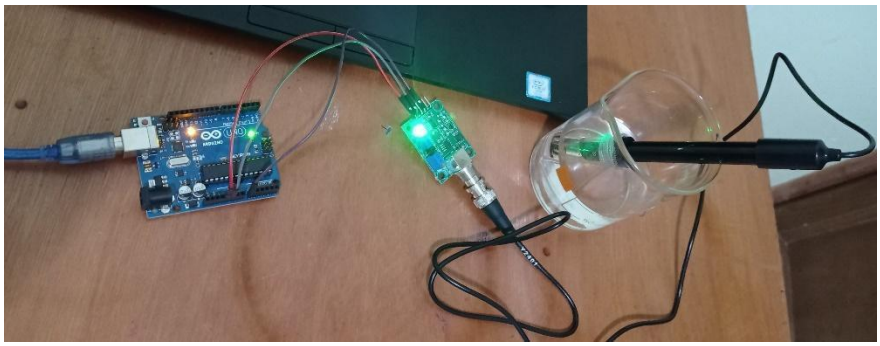


Figure III. 3: Expérimentation de mesure de pH à base d'Arduino

III.4.1.4Étalonnage de l'électrode de pH

L'étalonnage de l'électrode pH est très important dans ce projet. Pour cela, nous avons besoin d'une solution dont nous connaissons la valeur. Celle-ci peut être prise comme solution de référence pour l'étalonnage du capteur.

Supposons que nous ayons une solution dont la valeur pH est de 7 (eau distillée). Maintenant, lorsque l'électrode est plongée dans la solution de référence et que la valeur du pH affichée sur par Arduino est de 6,5. Ensuite pour le calibrer, il suffit d'ajouter 7-6.5=0.5 dans la variable de calibrage « calibration_value » dans le code.

III.4.1.5 Programmation Arduino pour pH-mètre

Une fois que le capteur est connecté au microcontrôleur, il est temps de procéder à la programmation de l'Arduino, en n'omettant surtout pas le processus de calibrage. Le code complet pour ce pH-mètre avec Arduino est présenté dans la Figure (III.4)

sketch_may13b.ino

```
// Déclaration des broches du capteur de pH
int pH_sensor = A0;

void setup() {
  // Initialisation de la communication Série
  Serial.begin(9600);
}

void loop() {
  // Lecture de la valeur du capteur de pH
  float pH_value = analogRead(pH_sensor);

  // Conversion de la valeur lue en pH
  float pH = map(pH_value, 0, 1023, 0, 14); // Les valeurs de pH peuvent varier

  // Affichage du pH sur le moniteur série
  Serial.print("pH: ");
  Serial.println(pH);

  delay(1000); // Attente d'une seconde avant la prochaine lecture
}

```

Figure III. 4: Programme Arduino pour la mesure du pH

III.4.1.6 Test du dispositif de mesure du pH

Le pH-mètre Arduino a été testé en plongeant dans de l'eau pure et de l'eau citronnée (Figure III.5). Lors du premier test, une eau potable avec un pH de 7.25, tel qu'indiqué sur l'étiquette de la bouteille, a été utilisée.

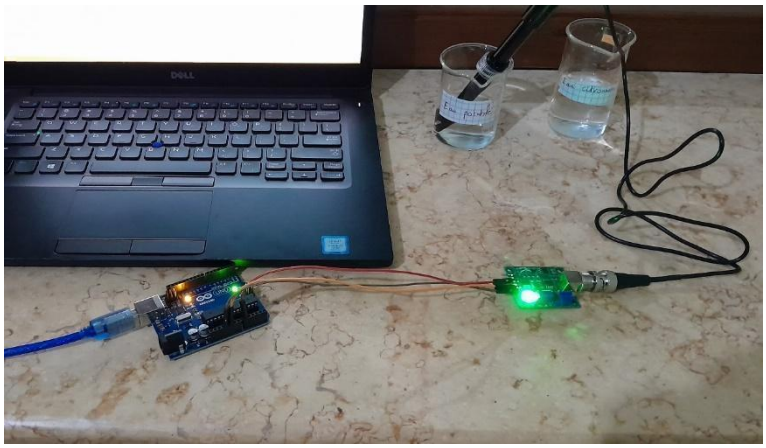


Figure III. 5: Expérimentation de la mesure de pH à base d'Arduino cas de l'eau potable

La figure III.6 donne les résultats de premier test ou la valeur de la tension est indiquée ainsi que le pH correspondant.

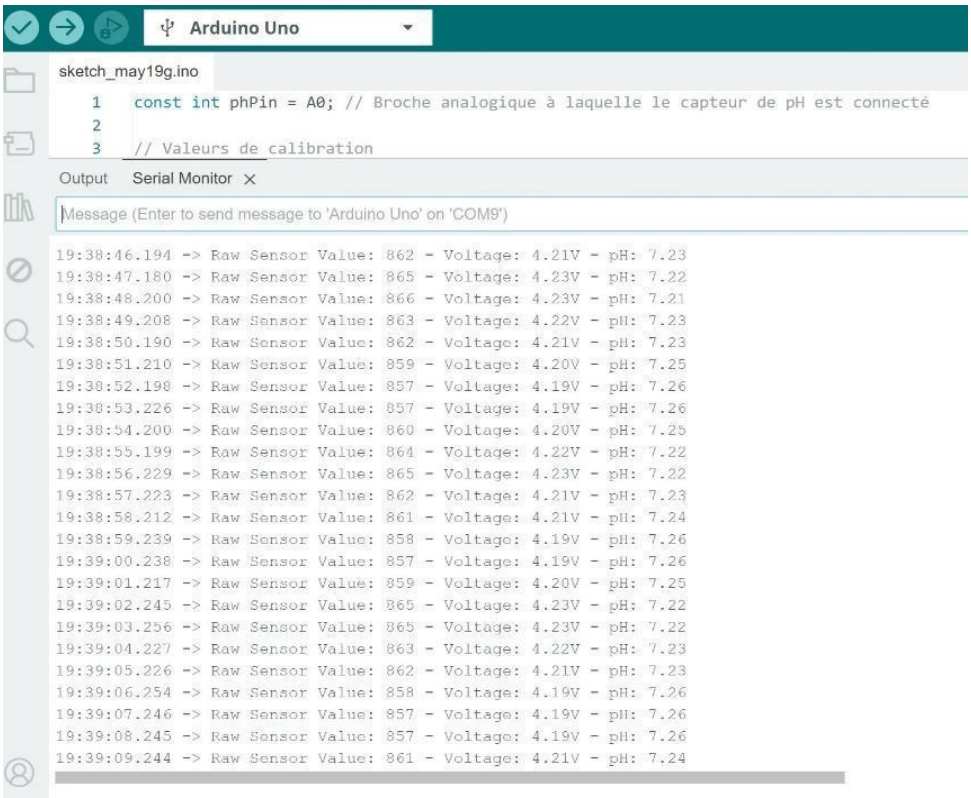


Figure III. 6: Résultat de test de mesure du pH d'eau potable

Dans le deuxième test, une eau citronnée est utilisée (Figure III.7).



Figure III. 7: Mesure de pH d'eau citronnée

Les résultats de deuxième test sont donnés par la figure (III.8), on remarque la diminution de la valeur du pH et c'est tout à fait normal puisqu'il s'agit d'une solution acide.



Figure III. 8: Résultat de test de mesure du pH d'eau citronnée

III.4.2.3 Connexion du capteur de turbidité avec Arduino

Nous connectons le VCC du capteur de turbidité avec Arduino 5V, GND à GND et sortie analogique à la broche Arduino A0, comme indiqué dans l'image de la Figure (III.9).

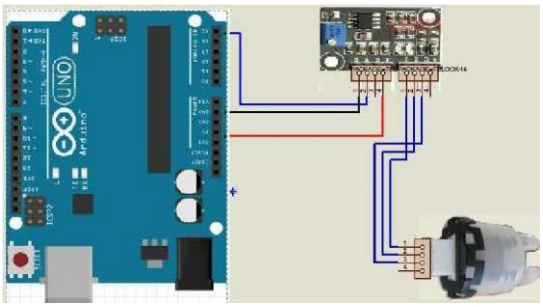


Figure III. 9: Utilisation de capteur de turbidité à base d'Arduino

III.4.2.4Code d'interface Arduino du capteur de turbidité de base

La figure (III.10) montre le code Arduino/capteur de turbidité. Ce code lira la valeur analogique du capteur et l'affichera sur le moniteur série.



Figure III. 10: Programme Arduino pour la mesure de turbidité

III.4.2.6 Résultats et observations

Pour vérifier le fonctionnement du projet, nous pouvons plonger le capteur dans différents types de sources d'eau. Nous pouvons mélanger de la boue ou de l'argile et vérifier la turbidité de l'eau en NTU. Par exemple, dans de l'eau propre, la tension restera à 4,2V. Si elle n'affiche pas 4,2, nous devons ajuster l'étalonnage en tournant le potentiomètre dans le capteur (Figure III.11).

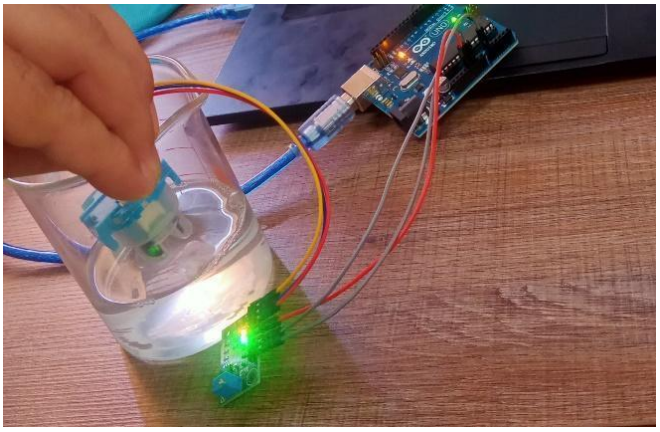


Figure III. 11: Réalisation expérimentale de capteur turbidité d'eau potable

La figure (III.12) donne les résultats d'un test effectué sur Arduino/Capteur de turbidité sur une eau potable.

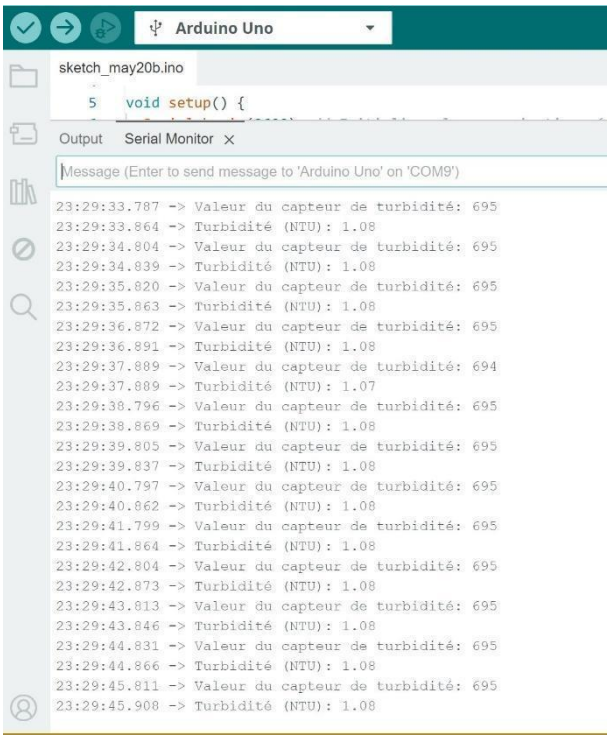


Figure III. 12: Résultats du premier test pour la mesure de turbidité à base d'Arduino

Pour effectuer un test expérimental de turbidité, nous avons préparé deux échantillons d'eau avec des niveaux différents de turbidité en ajoutant des quantités contrôlées de particules en suspension. Ces échantillons ont été placés dans des récipients transparents. En immergeant le capteur dans chaque échantillon, les valeurs de tension analogique correspondantes ont été lues par l'Arduino. Un script a permis de convertir ces tensions en valeurs NTU à l'aide d'une courbe d'étalonnage préalablement établie. Les résultats ont été affichés en temps réel sur le moniteur série Arduino, permettant d'observer l'évolution de la turbidité en fonction de la concentration de particules.

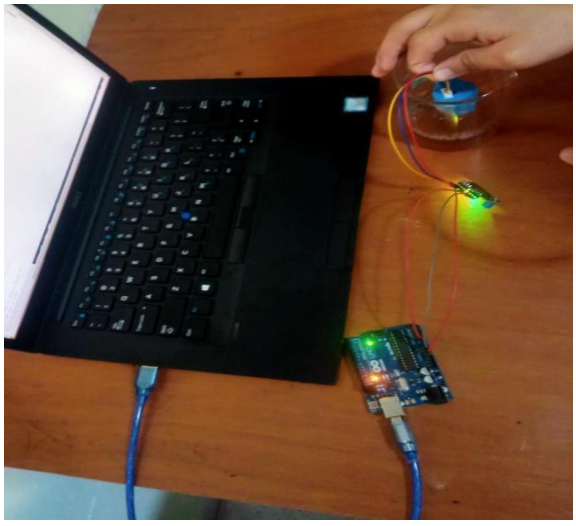


Figure III. 13: Mesure de capteur de turbidité d'eau trouble à base d'Arduino

La figure III.14 illustre les résultats d'un test effectué sur Arduino/Capteur de turbidité sur une eau trouble où l'on remarque une tension autour de 0V et NTU entre 39 et 45.

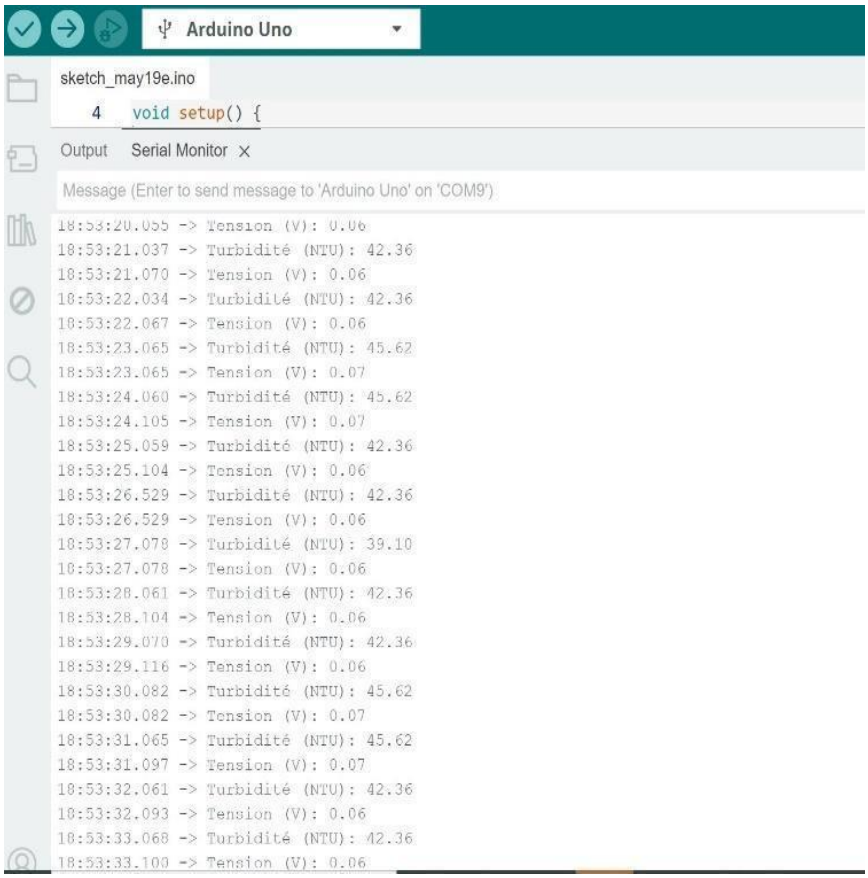


Figure III. 14: Résultats du deuxième test de mesure de turbidité à base d'Arduino

Les résultats des tests de turbidité pour les deux échantillons d'eau présentent des niveaux différents de contamination. Le premier échantillon affiche une turbidité de 1.08 NTU, ce qui indique une faible quantité de particules en suspension. En revanche, le deuxième échantillon présente une turbidité beaucoup plus élevée, mesurée à 42 NTU, ce qui suggère une concentration significativement plus importante de particules en suspension. Une différence marquée de clarté et de pureté est observée entre les deux échantillons, le second étant nettement plus trouble que le premier. Il est essentiel de surveiller de près la qualité de l'eau et de prendre des mesures pour réduire la turbidité, en particulier pour garantir sa potabilité et protéger la santé des personnes qui en consomment.

III.4.3 Capteur TDS et interface Arduino

Le capteur TDS analogique par gravité est un kit de capteur/mètre TDS compatible Arduino pour mesurer la valeur TDS de l'eau. Nous pouvons l'appliquer à l'eau domestique, à la culture hydroponique et à d'autres domaines de test de la qualité de l'eau. Ce produit prend en charge une entrée de tension large de 3,3 à 5,5 V et une sortie de tension analogique de 0 à 2,3 V, ce qui le rend compatible avec les systèmes ou cartes de contrôle de 5 V ou 3,3 V. La source d'excitation est un signal CA, qui peut efficacement empêcher la polarisation de la sonde et prolonger la durée de vie de la sonde, tout en contribuant à augmenter la stabilité du signal de sortie. La sonde TDS est étanche, nous pouvons l'immerger dans l'eau pour des mesures de longue durée.

III.4.3.1 Interface du capteur TDS analogique de gravité avec Arduino

Le schéma de la figure III.15 montre l'interfaçage du capteur TDS avec l'Arduino. La connexion du capteur TDS avec Arduino est assez simple. Le VCC doit être connecté au

5V d'Arduino et le GND au GND. Ensuite, la broche analogique du capteur doit être connectée à n'importe quelle broche analogique d'Arduino. Dans ce cas, la broche analogique A2 d'Arduino a été utilisée.

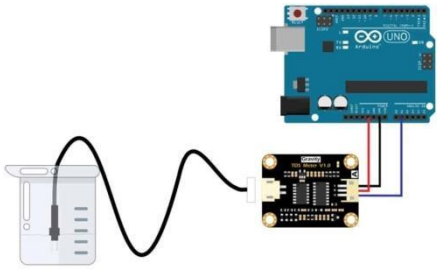


Figure III. 15: Mesure de TDS à base d'Arduino

III.4.3.2 Étalonnage de l'électrode de TDS

Pour ajuster correctement la conversion de la valeur analogique lue par le capteur de TDS en unités de TDS (ppm), il est essentiel que le capteur soit calibré avec des solutions de référence ayant des valeurs de TDS connues. Un guide pour la calibration du capteur de TDS est fourni ci-dessous:

- La préparation ou l'achat de solutions de référence avec des valeurs de TDS connues (par exemple, 0 ppm, 100 ppm, 500 ppm, etc.) est requis.
- Le capteur de TDS doit être immergé dans chaque solution de référence et les valeurs analogiques lues (tds_Value) doivent être enregistrées.
- Un graphique avec les valeurs analogiques (tds_Value) sur l'axe des x et les valeurs de TDS connues (ppm) sur l'axe des y doit être tracé.
- La relation entre la tension mesurée et la valeur de TDS doit être déterminée en utilisant une régression linéaire ou une autre méthode d'ajustement. Une formule de la forme TDS = a×tension+b sera obtenue, où a et b sont des constantes déterminées à partir de la calibration.
- La formule de conversion dans le code doit être mise à jour en utilisant les valeurs a et b déterminées.

III.4.3.3 Code Arduino

Un code simple pour interfacer le capteur TDS avec Arduino est présenté par la Figure III.16.

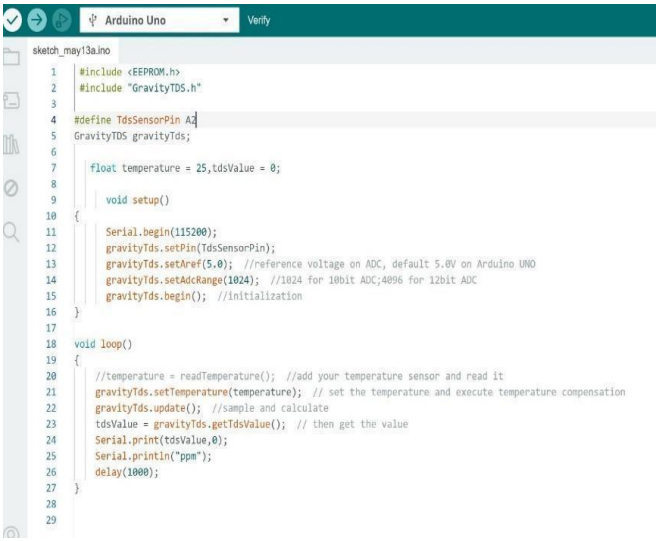


Figure III. 16: Programme Arduino pour la mesure de TDS à base d'Arduino

Maintenant, la bibliothèque DfRobot Gravity: Capteur analogique TDS doit être installée dans l'IDE Arduino, et elle peut être incluse dans les croquis en allant dans le menu "Sketch", en sélectionnant "Inclure la bibliothèque", puis en choisissant DfRobot Gravity: Capteur analogique TDS dans la liste..

III.4.3.4 Réalisation réelle du capteur TDS

Le premier test sera réalisé avec de l'eau potable (Figure III.17) et les résultats sont donnés par la figure (III.18)

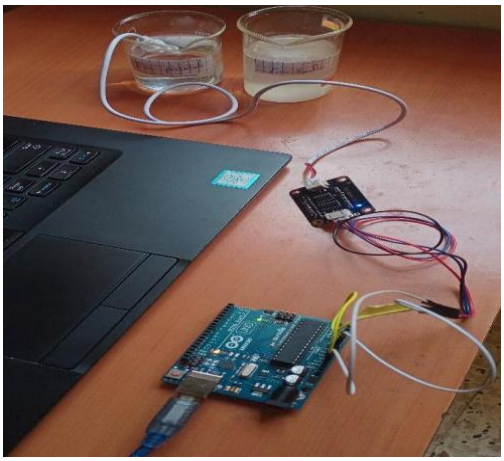


Figure III. 17: Réalisation expérimentale pour la mesure de TDS cas d'eau potable

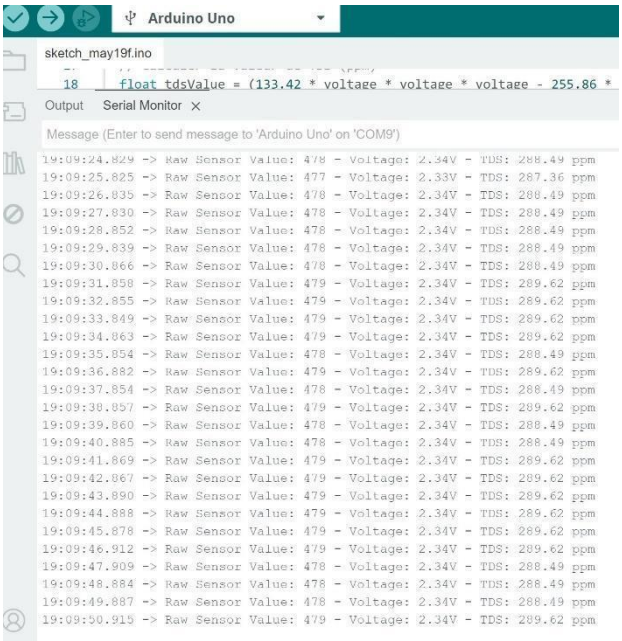


Figure III. 18:Résultats de test pour la mesure de turbidité d'eau potable

Pour le deuxième test, on utilise de l'eau salée (Figure III.19) et les résultats sont donnés par la figure III.20.

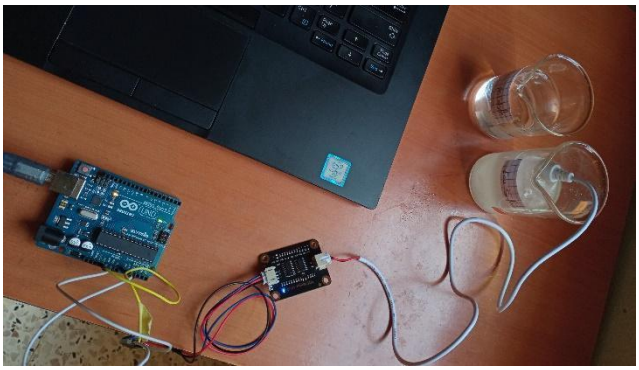


Figure III. 19: Mesure de turbidité à base d'Arduino en cas d'eau salée

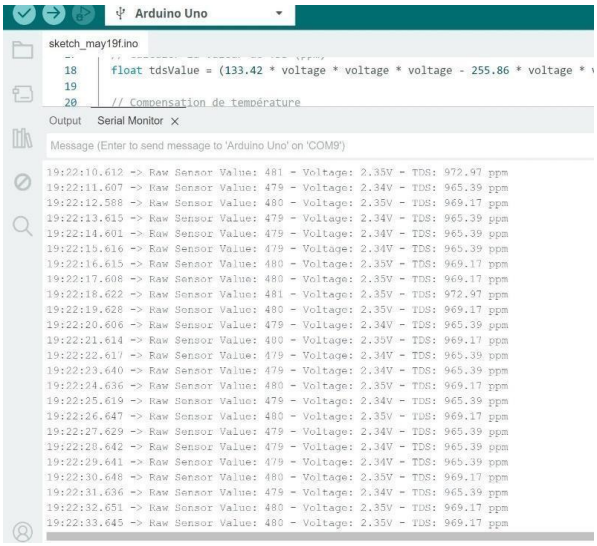


Figure III. 20:Résultat de mesure de turbidité d'eau salée

III.4.3.5 Comparaison entre nos résultats précédents

Une différence significative est observée entre nos échantillons. Le premier échantillon avec un TDS de 250 ppm est bien situé dans les limites acceptables pour l'eau potable. Cependant, le deuxième échantillon avec un TDS de 970 ppm est presque quatre fois plus élevé que le premier, dépassant largement les recommandations de potabilité.

III.5 Réalisation

III.5.1 Schéma synoptique global de système

La figure ci-dessus représente le schéma global de système pour la mesure des trois grandeurs à savoir : le pH, la turbidité et la conductivité (Figure III.21).

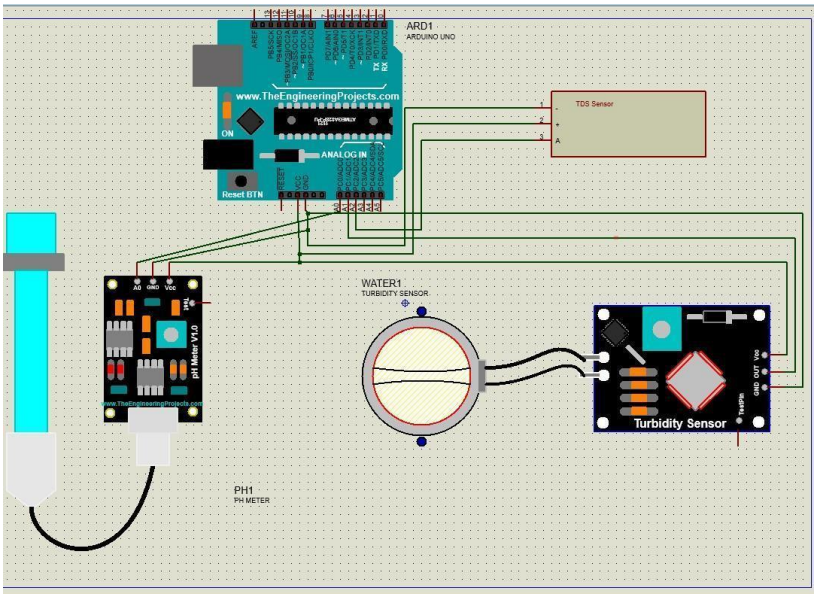


Figure III. 21: Schéma synoptique global de mesure de pH, turbidité et TDS à base d'Arduino

III.5.2 Connexion de l'ensemble des capteurs à Arduino

Le capteur est connecté aux signaux analogiques de l'Arduino.

- Le capteur de pH est connecté sur la broche analogique A0.
- Le capteur de turbidité est connecté sur la broche analogique A1.
- Le capteur de TDS est connecté sur la broche analogique A2.
- L'alimentation (+5V) et la masse (GND) sont ajustées pour chaque capteur.

Pour les capteurs :

- Le pH est converti en signal analogique par le capteur de pH.
- La turbidité est convertie en signal analogique par le capteur de turbidité.
- L'affichage TDS est converti en signal analogique par le capteur TDS.

Concernant l'Arduino :

- Les données sont acquises en utilisant les signaux analogiques et les capteurs via les broches A0, A1 et A2 respectivement.
- La méthode de conversion est utilisée pour convertir les valeurs analogiques (0-1023) via l'ADC (Analogie-to-Digital Converter) vers l'Arduino.
- Les paramètres sont calculés en convertissant les nombres en unités (pH, NTU en turbidité, ppm en TDS) à l'aide des formules d'étalonnage utilisées dans le condensateur.
- Les résultats sont affichés sur un écran à l'Arduino (Sortie).

III.5.3 Code Arduino

La figure (III.22) donne le code Arduino global du circuit final incluant les trois capteurs.

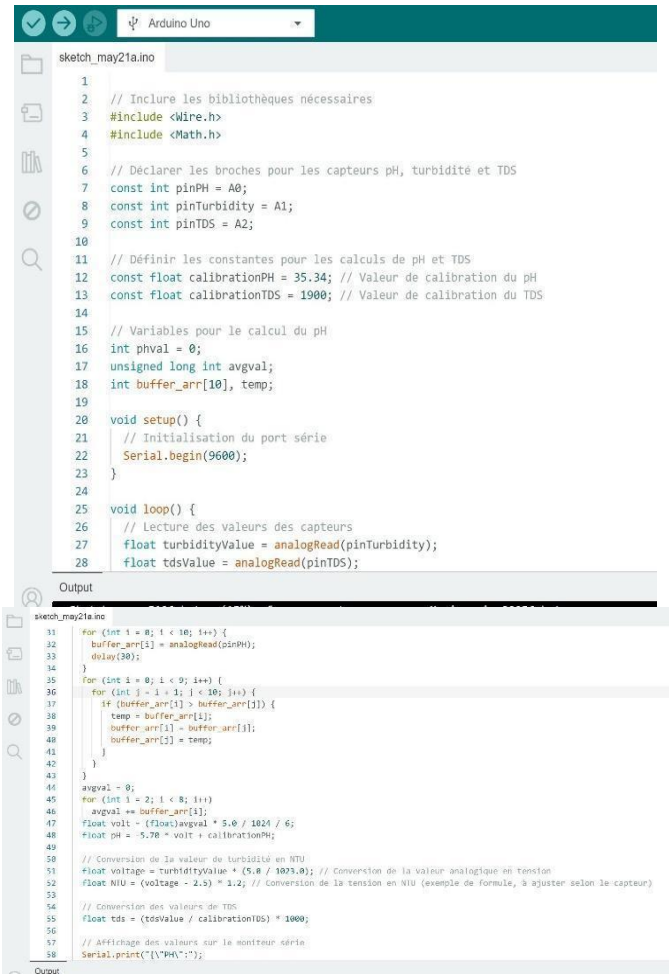


Figure III. 22: Code Arduino pour la mesure de pH, turbidité et conductivité

III.5.4Prototype du test réalisé

Un code Arduino a été réalisé pour connecter les capteurs utilisés. Ensuite, une interface homme-machine (IHM) a été réalisée pour la supervision en utilisant la plateforme Node-RED d'IBM. Une communication entre l'IHM et l'Arduino a été établie en utilisant le port série. Le prototype réel est présenté sur la figure III.23.

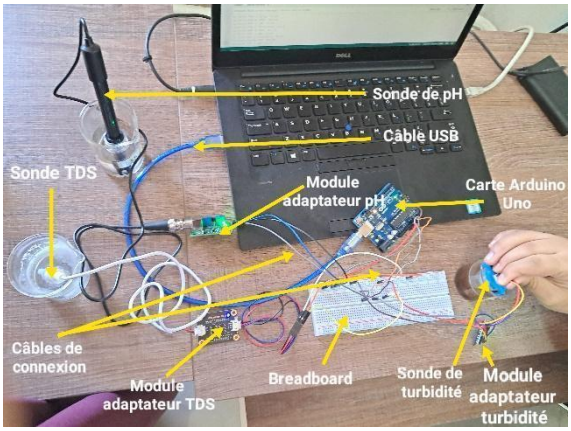


Figure III. 23: Expérimentation pour la mesure de pH, turbidité et conductivité

III.5.5MESURES ET TESTS

III.5.5.1 Premier test

Un premier test a été réalisé sur de l'eau potable (Figure III.24) et les résultats de mesure des trois capteurs sont donnés par la Figure III.25.

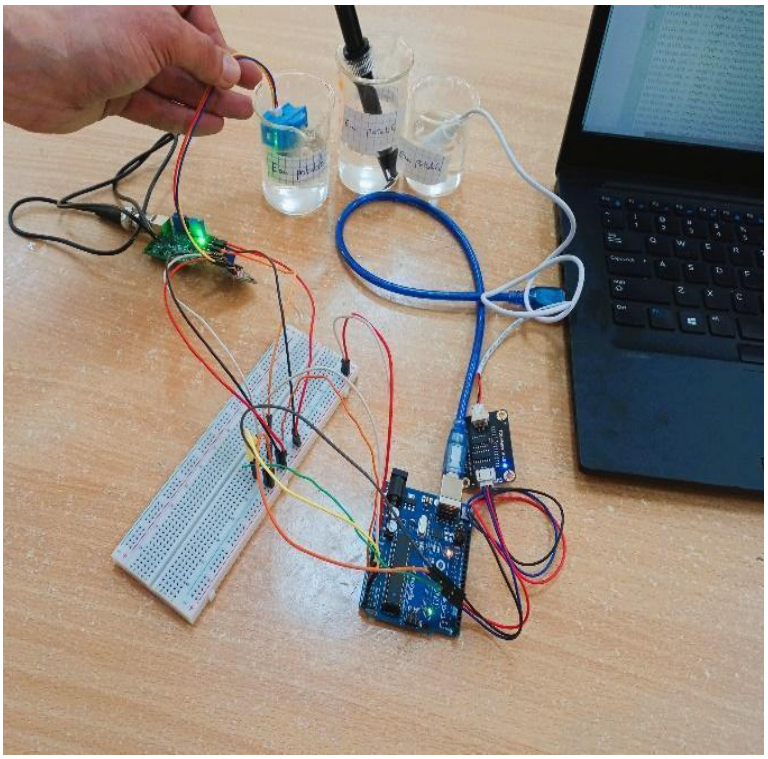


Figure III. 24: Mesure de pH, turbidité et conductivité à base d'Arduino d'une eau potable .

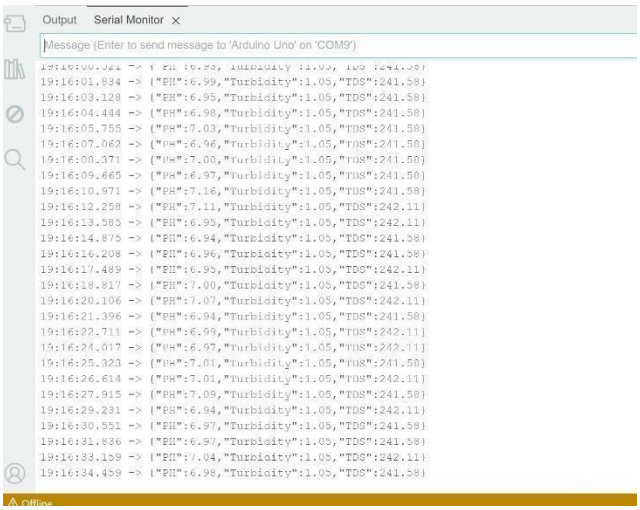


Figure III. 25: Résultat de mesure de pH, turbidité et conductivité à base d'Arduino d'une eau potable

III.5.5.2Deuxième test

Un deuxième test a été réalisé et les résultats sont donnés par la Figure III.26.

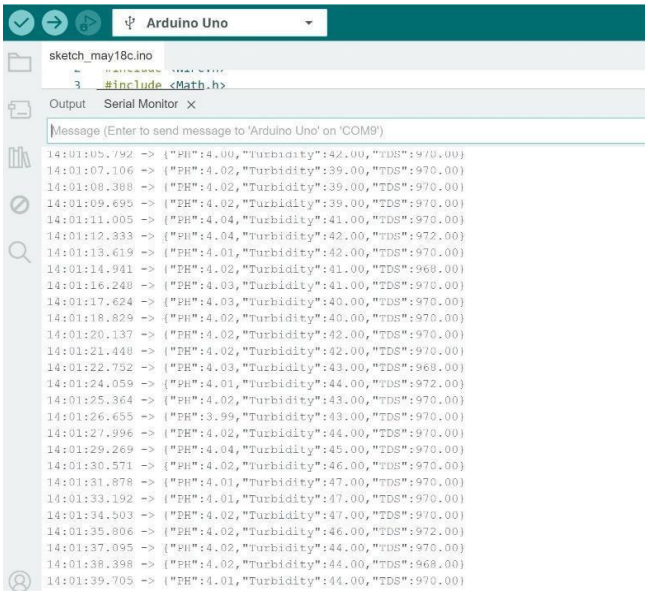


Figure III. 26:Résultats d'un deuxième test pour la mesure de pH, turbidité et conductivité à base d'Arduino

III.6 Identification de la qualité d'eau par intelligence artificielle

Des modèles intelligent de machine learning sont utilisés pour l'aide à la décision sur la potabilité de l'eau. Pour cela on a utilisé une base de données existante [43] pour l'apprentissage de quatre modèles à savoir: RL, RF, ANN et KNN.

III.6.1 Base de données utilisée

La base de données utilisée est disponible en ligne sur Kaggle [43]. Quelques mesures de la base de données sur la qualité de l'eau pour plusieurs échantillons sont présentées dans le tableau III.1. Chaque ligne du tableau représente un échantillon et chaque colonne représente une caractéristique spécifique de l'eau mesurée pour cet échantillon. Cette feuille de calcul est extraite de la base de données utilisée.

Tableau III. 1: Exemple de données de potabilité de l'eau

	A	B	C	D	E	F	G	H	I	J
1	ph	Hardness	Solids	Chloramines	Sulfate	Conductivity	Organic_carb	Trihalometha	Turbidity	Potability
2		204.890456	20791.319	7.30021187	368.516441	564.308654	10.3797831	86.9909705	2.96313538	0
3	3.71608008	129.422921	18630.0579	6.63524588		592.885359	15.1800131	56.3290763	4.50065628	0
4	8.09912419	224.236259	19909.5417	9.2758836		418.606213	16.8686369	66.4200925	3.05593375	0
5	8.31676588	214.373394	22018.4174	8.05933238	356.886136	363.266516	18.4365245	100.341674	4.62877054	0
6	9.09222346	181.101509	17978.9863	6.54659997	310.135738	398.410813	11.5582794	31.9979927	4.07507543	0
7	5.58408664	188.313324	28748.6877	7.54486879	326.678363	280.467916	8.39973464	54.9178618	2.55970823	0
8	10.2238622	248.071735	28749.7165	7.51340847	393.663396	283.651634	13.7896953	84.6035562	2.67298874	0
9	8.63584872	203.361523	13672.0918	4.56300869	303.309771	474.607645	12.3638167	62.798309	4.40142472	0
10		118.988579	14285.5839	7.80417355	268.646941	389.375566	12.706049	53.9288458	3.59501718	0
11	11.1802845	227.231469	25484.5085	9.07720002	404.041635	563.885482	17.9278064	71.976601	4.37056194	0
12	7.36064011	165.520797	32452.6144	7.55070091	326.624354	425.38342	15.5868104	78.7400157	3.66229178	0
13	7.97452165	218.693301	18767.6567	8.1103845		364.098231	14.5257457	76.4859112	4.01171811	0
14	7.11982438	156.704993	18730.8137	3.60603609	282.344051	347.715027	15.9295359	79.5007783	3.44575622	0
15		150.174923	27331.362	6.83822347	299.415781	379.761835	19.3708072	76.5099955	4.41397418	0
16	7.49623221	205.344982	28388.0049	5.07255777		444.645352	13.2283111	70.3002127	4.77738234	0
17	6.34727176	186.732881	41065.2348	9.62959628	364.487687	516.743282	11.5397812	75.0716173	4.37634829	0
18	7.0517858	211.049406	30980.6008	10.094796		315.141267	20.3970218	56.6516038	4.26842886	0
19	9.18156001	273.813807	24041.3263	6.90498973	398.350517	477.974642	13.3873408	71.4573622	4.5036608	0
20	8.97546435	279.357167	19460.3981	6.20432086		431.44399	12.8887591	63.8212371	2.43608559	0

water_potabilitycsv

Entrées

1) *pH*: Ce champ indique le pH de l'eau. Le pH est une mesure de l'acidité ou de l'alcalinité de l'eau, allant de 0 à 14, avec 7 étant neutre, moins de 7 acide et plus de 7 alcalin.
2) *Dureté*: Ce champ indique la dureté de l'eau, qui est une mesure de la quantité de minéraux dissous dans l'eau, tels que le calcium et le magnésium, en mg/L.
3) *Solides*: Ce champ indique la quantité totale de solides dissous dans l'eau en ppm. Les solides dissous sont des substances qui ne peuvent pas être éliminées par filtration.
4) *Chloramines*: Ce champ indique la quantité de chloramines dans l'eau en ppm. Les chloramines sont des désinfectants utilisés pour traiter l'eau potable.
5) *Sulfate*: Ce champ indique la quantité de sulfates dissous dans l'eau en mg/L. Les sulfates sont des minéraux naturels ou ajoutés comme coagulants.
6) *Conductivité*: Ce champ indique la conductivité électrique de l'eau en µS/cm, qui est une mesure de sa capacité à conduire l'électricité.
7) *Organic_carbon*: Ce champ indique la quantité de carbone organique dissous dans l'eau en ppm, provenant de sources naturelles ou anthropiques.
8) *Trihalométhanes*: Ce champ indique la quantité de trihalométhanes dans l'eau en µg/L, qui sont des sous-produits potentiellement cancérigènes de la désinfection.
9) *Turbidité*: Ce champ indique la turbidité de l'eau en NTU, qui est une mesure de sa clarté et de la présence de particules en suspension.

Sortie

10) *Potabilité:* Ce champ indique si l'eau est propre à la consommation humaine, avec une valeur de 1 pour potable et 0 pour non potable.

III.6.2 Analyse des corrélations entre les paramètres de qualité de l'eau

Le Tableau III.2 présente la corrélation entre les différents paramètres de l'eau. On remarque des corrélations presque nulles entre les différents paramètres, ce qui peut indiquer que les entrées sont totalement indépendantes deux à deux.

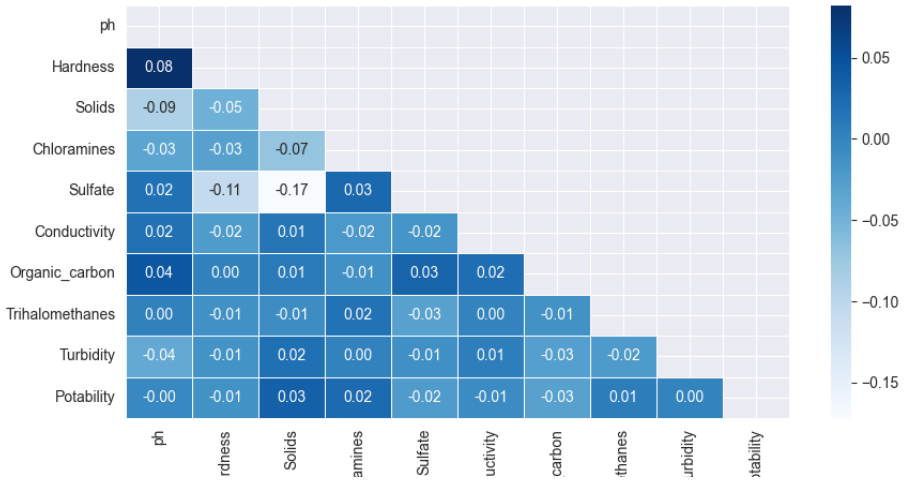


Figure III. 27:Carte de corrélation des paramètres de qualité de l'eau

La figure III.27 représente la matrice de dispersion des paramètres de qualité de l'eau en fonction de la potabilité. Une analyse de cette matrice permet de faire les affirmations suivantes :

1) On observe une grande dispersion des valeurs pour la plupart des paramètres, que l'eau soit potable ou non.
2) Certains paramètres, comme le pH et la dureté, semblent avoir une plage de valeurs plus restreinte pour l'eau potable par rapport à l'eau non potable.
3) D'autres paramètres, tels que les solides dissous totaux (TDS) et la conductivité, présentent un chevauchement important des valeurs entre l'eau potable et non potable, rendant leur discrimination difficile.
4) La matrice de dispersion met en évidence la complexité de la tâche de classification de la potabilité de l'eau en se basant uniquement sur les paramètres physico-chimiques individuels.

Cette analyse souligne l'importance d'utiliser des techniques d'apprentissage automatique avancées, capables de traiter les relations non linéaires et les interactions entre les différents paramètres, pour une classification précise de la potabilité de l'eau.

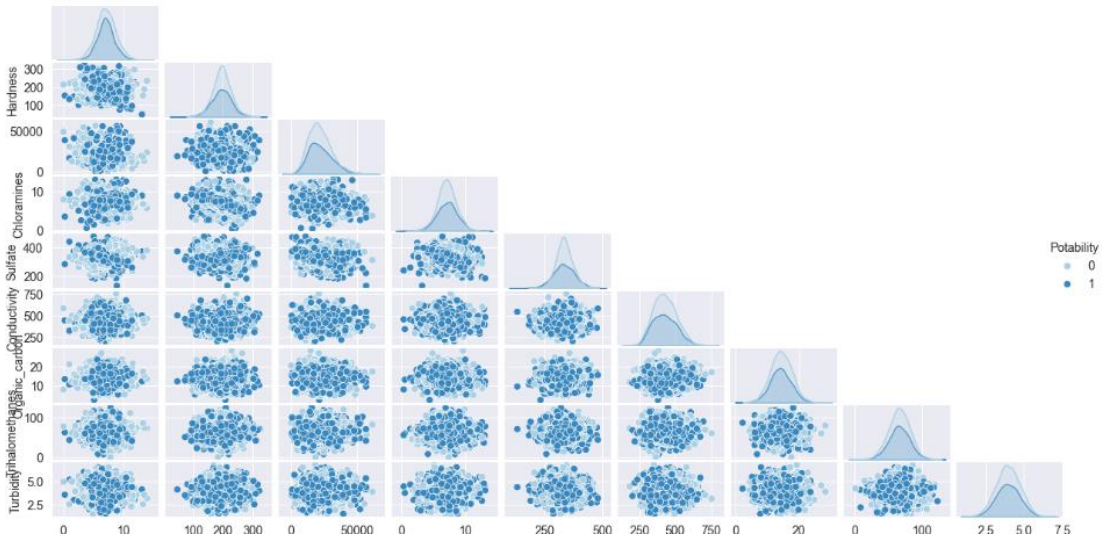


Figure III. 28: Matrice de dispersion des paramètres de qualité de l'eau

III.6.3 Appel aux fichiers Excel dans un projet Python

Pour synthétiser les différents modèles de machine learning en utilisant le langage Python, un fichier Excel contenant la base de données utilisée pour l'apprentissage et l'évaluation des modèles a dû être utilisé. La bibliothèque pandas a été employée pour lire les données Excel dans une structure de données appelée DataFrame. Un exemple de code illustrant cette approche est présenté dans la Figure III.28. Un fichier Excel nommé "donnees.xlsx" y est chargé et les premières lignes sont affichées.

```python
import pandas as pd

# Chargement du fichier Excel dans un DataFrame
df = pd.read_excel("donnees.xlsx")

# Affichage des premières lignes du DataFrame
print(df.head())
```

Figure III. 29: Programme python d'extraction de données à partir d'un fichier Excel

Il a été vérifié que le fichier Excel se trouvait dans le même répertoire que le script Python. Si ce n'était pas le cas, le chemin complet du fichier a été spécifié. Après le chargement des données dans un DataFrame, diverses opérations ont pu être effectuées, telles que le filtrage, le tri et le regroupement, en exploitant les fonctionnalités puissantes de pandas.

III.6.4 Modèles de Machine Learning

Dans notre projet Python, nous avons intégré plusieurs modèles de machine Learning pour effectuer différentes tâches. Les quatre modèles que nous avons utilisés sont Random Forest, Régression Logistique (RL), K plus proches voisins (KNN) et Perceptron multicouche (MLP/ANN).

III.6.4.1 Modèle ANN

Ce modèle fait appel à un ensemble de bibliothèques complémentaires

- **Pandas :** est utilisé pour le traitement des données, garantissant leur organisation et leur préparation pour l'analyse.
- **Scikit-learn** : intervient dans la segmentation des données, permettant de les diviser en ensembles distincts pour l'entraînement, la validation et le test du modèle.
- **Tensorflow** et **Keras** : quant à eux, s'associent pour la création et la formation du réseau neuronal, exploitant la puissance des calculs matriciels pour optimiser les paramètres du modèle et améliorer ses performances.

Séparation des données en caractéristiques (X) et cible (y)

```python
X = data.drop(columns='Potability')
y = data.Potability
```

Figure III. 30: Extraction de la colonne "Potabilité" d'un ensemble de données

Séparation des données :

- X : contient toutes les colonnes de données sauf la colonne 'Potabilité'. Ces colonnes sont appelées les caractéristiques ou fonctionnalités.
- y : contient uniquement la colonne 'Potabilité', qui est la cible ou le label que nous voulons prédire.

Fonction pour Entraîner et Évaluer un Modèle MLP (Figure III.30).

```
def mlp_classifier_model(X_train, y_train, X_test, y_test):
    parameters = {
        'hidden_layer_sizes': [500],
        'activation': ['logistic'],
        'alpha': [0.0001],
        'batch_size': [200],
    }
```

Figure III. 31: Ajustement des paramètres du modèle ANN

Dictionnaire des Paramètres :

- Paramètres = { ... } :Contient les hyperparamètres que GridSearchCV testera pour trouver les meilleures valeurs.
- 'hidden_layer_sizes' : [500] : Indique que le réseau de neurones aura une seule couche cachée avec 500 neurones. La configuration des canapés cachés peut être modifiée selon les besoins du problème.
- 'activation': ['logistique'] : Spécifie que la fonction d'activation utilisée par les neurones sera la fonction logistique (sigmoïde). Cette fonction est utilisée pour introduire la non-linéarité dans le réseau de neurones.
- 'alpha' : [0,0001] : Définit le terme de régularisation L2 (pénalité de poids). Cela aide à prévenir le surapprentissage en pénalisant les poids élevés.
- 'batch_size' : [200] : Indique que la taille des lots pour l'entraînement sera de 200. La taille du lot déterminant combien d'échantillons sera propagée à travers le réseau avant de mettre à jour les poids.

Précision

- La précision est une métrique de performance qui calcule le pourcentage de prédictions correctes par rapport au nombre total de prédictions, évaluant ainsi la performance de notre modèle.

Arrêt précoce

- Cette technique interrompt l'entraînement si la performance du modèle sur un ensemble de validation commence à se dégrader, prévenant ainsi le surapprentissage en empêchant le modèle de mémoriser les données d'entraînement.

```
print(colored('MLPClassifier Accuracy : {}'.format(round(accuracy, 3)), 'green'))
```

Figure III. 32 : Affichage de l'accuracy de l'ANN

- ## Test

Les tests sont effectués en modifiant les paramètres: Hidden-Layer-size (nombre de neurone dans la couche cachée) , fonction d'activation, le parmètre d'apprentissage alpha et le batch-size (Tableau III.3). Le résultat est comparé en fonction de l'accuracy.

Tableau III. 2:Résultats des tests pour l'apprentissage à base d'ANN

Test_Size	Hidden-Layer	Activation	Alpha	Batch-size	**Accuracy**
20%	50	Relu	0.1	50	0.70
	100	Logistic	0.0001	200	0.78
	500	Logistic	0.0001	200	0.82
30%	100	Logistic	0.0001	200	0.72
	300	Relu	0.1	50	0.81
	800	Relu	0.01	100	0.69
60%	100	Logistic	0.001	200	0.84
	500	Logistic	0,01	200	0.77
	1000	Relu	0.01	50	0.81

- **Résultats**

Les meilleures valeurs de précision et matrices de confusion ont été obtenues en fonction du pourcentage de données utilisées pour les tests. En comparant les matrices de confusion des figures III.32, III.33 et III.34 avec les valeurs de précision, nous avons constaté que la précision est la plus élevée lorsque 80 % des données sont utilisées pour l'entraînement et 20 % pour les tests.

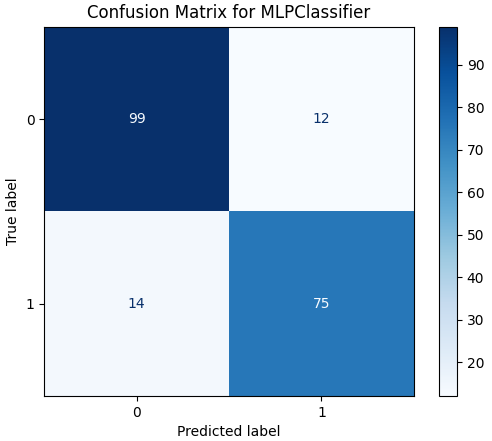


Figure III. 33: Matrice de confusion de l'apprentissage d'ANN pour un taux de test de 20%

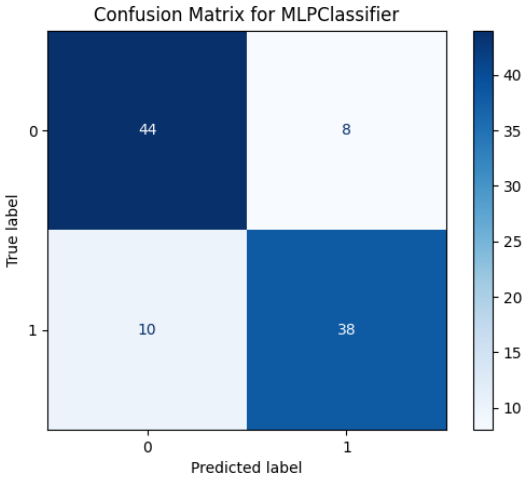


Figure III. 34: Matrice de confusion de l'apprentissage d'ANN pour un taux de test de 30%

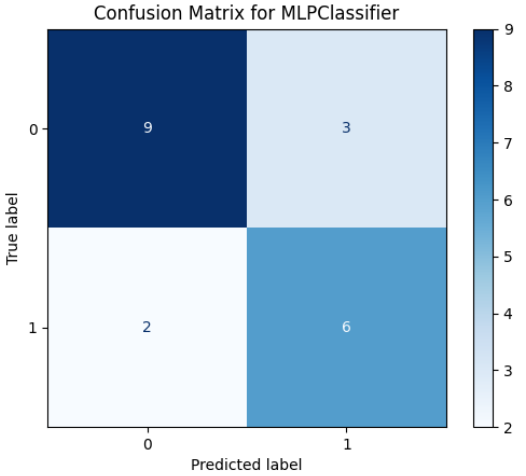


Figure III. 35: Matrice de confusion de l'apprentissage d'ANN pour un taux de test de 60%

III.6.4.2 Classificateur LR

L'algorithme de régression logistique (LR) utilise la bibliothèque «Logistic Regression», qui offre des outils puissants pour créer et entraîner des modèles prédictifs. Cette bibliothèque facilite l'importation de fichiers, la segmentation des données et la sélection des fonctionnalités Les données sont normalisées à l'aide de la fonction «StandardScaler()», ce qui permet de les mettre à la même échelle. Cette normalisation est illustrée à la Figure (III.35).

```
scaler = StandardScaler()
```

Figure III. 36 : Mise à l'échelle de données pour le modèle RL

Nous avons configuré l'algorithme de régression logistique (LR) et déterminé un réseau de paramètres hyperparamétriques candidats, ce qui permet de rechercher les meilleurs paramètres pour le modèle logistique. Ensuite, nous avons configuré la recherche en grille en utilisant "GridSearchCV", rendant ainsi le modèle prêt à être entraîné avec "x_train_scaled" et "y_train". Après l'entraînement, nous exécutons des prédictions sur l'ensemble de test pour évaluer la précision, comme illustré à la Figure III.36.

```
grid_search = GridSearchCV(estimator=logistic_classifier, param_grid=param_grid, cv=5, scoring='accuracy')
```

Figure III. 37: Recherche par grille dans RL

```
parameters = {
    'C' : [0.001, 1, 1000],
    'class_weight' : ['balanced', None],
    'solver' : ['liblinear', 'sag'],
    'penalty' : ['l2'],
    'n_jobs' : [-1],
}
```

Figure III. 38: Paramètres du modèle RL

Les valeurs de C ont été choisies dans une plage logarithmique de 0,001 à 1000. La valeur de class_weight a été définie sur "balanced" pour pondérer les classes en fonction de leur importance relative. L'algorithme solver a été défini sur "liblinear" pour utiliser l'algorithme liblinear pour résoudre le problème d'optimisation. La fonction de pénalité a été définie sur "12" pour utiliser la régularisation L2. Le nombre de threads a été défini sur -1 pour utiliser tous les threads disponibles. Le modèle a été évalué sur un ensemble de données de test et a obtenu une précision de 95%.

- **Test**

Les tests sont effectués en modifiant les paramètres C,max-iter et solver. Pour les détails concernant la taille de la couche, qui est un outil précieux pour les utilisateurs. Tableau III.4

Tableau III. 3:Résultats des tests pour l'apprentissage RL

Test_Size	max_iter	C Punishment power	Solver	**Accuracy**
20%	100	100	Liblinear	0.70
	1000	1	newton-cg	0.88
	2000	10	lbfgs', 'saga'	0.72
	200	100	Liblinear	0.78
	500	0.01	lbfgs, saga	0.77
50%	100	100	Liblinear, saga	0.79
	200	0.01	newton-cg	0.67
	500	1	newton-cg	0.77

- **Résultats**

Après nos tests, nous avons attribué plusieurs valeurs à la meilleure précision et à la meilleure matrice de confusion du tableau, en fonction de la proportion de données fournies pour les tests. Après avoir comparé les valeurs des matrices de confusion des figures III.38, III.39 et III.1avec les valeurs de précision, nous avons constaté que la précision est élevée lorsque 80 % des données sont utilisées pour l'entraînement et le reste pour tester le modèle.

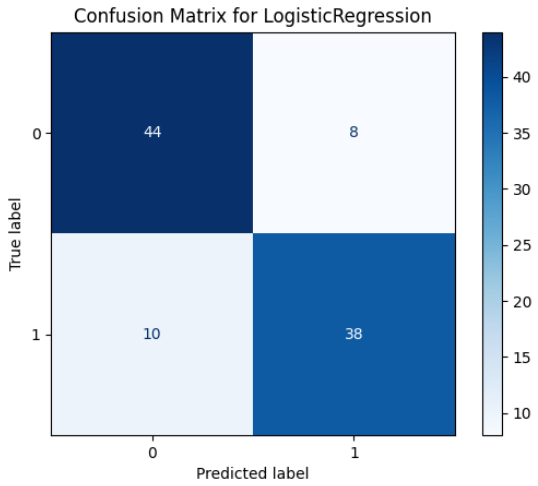


Figure III. 39: Matrice de confusion de l'apprentissage RL pour un taux de test de 20%

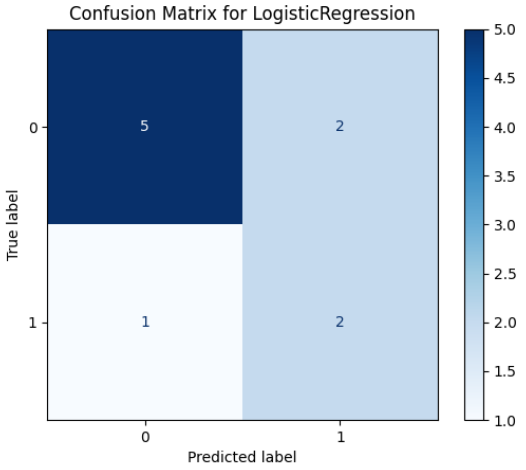


Figure III. 40: Matrice de confusion de l'apprentissage RL pour un taux de test de 50%

III.6.4.3 Classificateur RF

Nous devons utiliser l'algorithme RF (RandomForestClassifier), qui propose des outils performants pour créer et saisir des modèles prédictifs. Cette bibliothèque facilite l'importation de fichiers, la segmentation des données et la sélection de fonctionnalités. Nous commençons par utiliser le classificateur principal et spécifions une grille de paramètres pour trouver la meilleure combinaison possible (Figure III.40)

```
rf_classifier = RandomForestClassifier()
```

Figure III. 41: Initialisation de RF

```
parameters = {
    'n_estimators' : [1000],
    'criterion' : ['log_loss'],
    'max_features' : ['sqrt'],
    'n_jobs' : [-1]
}
```

Figure III. 42: Paramètres de RF

- **n_estimators : [1000]** :Utilise 1000 arbres ou itérations dans le modèle, augmentant potentiellement les performances mais aussi le temps de calcul.
- **criterion : ['log_loss'] :**Utilise la fonction de perte "log_loss" (ou "cross-entropy loss") pour l'optimisation, adaptée aux problèmes de classification.
- **max_features : ['sqrt']** :Considère la racine carrée du nombre total de caractéristiques pour chaque split, ce qui aide à réduire la corrélation entre les arbres.
- **n_jobs : [-1]** :Utilise tous les processeurs disponibles pour l'entraînement, accélérant le calcul grâce au parallélisme.

- **Test**

  En modifiant ces paramètres, nous pouvons évaluer et tester le modèle comme indiqué dans le tableau III.5.

Tableau III. 4:Résultats des tests pour les enseignantes RF

Test_Size	n_Estimators	min_samples_leaf	max_Depth	**Accuracy**
20%	50	2	None	0.89
	200	4	10	0.92
	500	10	20	0.90
30%	50	**2**	None	0.77
	200	4	10	0.92
	500	10	20	0.90
50%	50	2	None	0.91
	200	4	10	0.88
	500	10	20	0.90

-

- **Résultat**

Les figures de III.42 à III.44 donnent les matrices de confusion correspondante aux tailles des données de test de 20%, 30% et 50% respectivement. On note que la taille de 20% donne de meilleurs résultats.

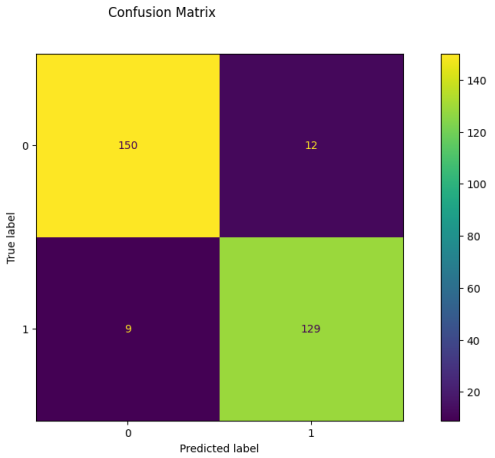


Figure III. 43: Matrice de confusion de l'apprentissage RF pour un taux de test de 20%

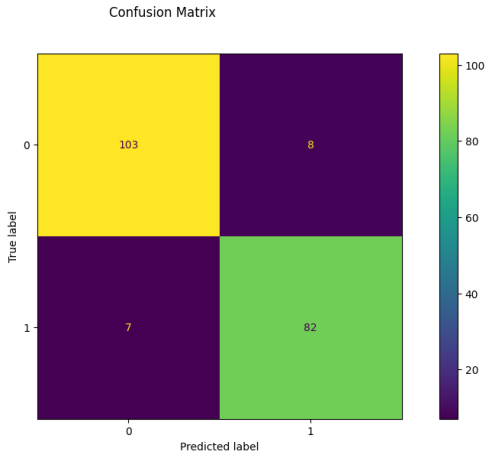


Figure III. 44: Matrice de confusion de l'apprentissage RF pour un taux de test de 50%

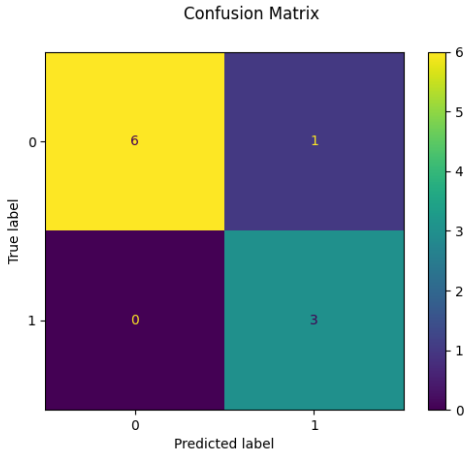


Figure III. 45: Matrice de confusion de l'apprentissage RF pour un taux de test de 50%

III.6.4.4Classificateur K-NN

Nous avons créé une instance de la classe KNeighborsClassifier en utilisant l'instruction knn = KNeighborsClassifier(). Cet objet représente l'algorithme KNN avec ses paramètres par défaut (Figure III.45).

```
knn = KNeighborsClassifier()
knn_cv = GridSearchCV(estimator=knn, cv=10, param_grid=parameters).fit(X_train, y_train)
```

Figure III. 46: Création d'un Classificateur KNN

L'extrait de code fourni démontre le processus de réglage des hyperparamètres pour un classificateur KNN à l'aide de la classe GridSearchCV. Cette approche nous permet d'évaluer systématiquement différentes combinaisons de valeurs d'hyperparamètres et d'identifier l'ensemble qui offre les meilleures performances sur notre base de données spécifique (Figure III.46).

```
knn = KNeighborsClassifier(**knn_cv.best_params_).fit(X_train, y_train)
```

Figure III. 47: Optimisation des hyperparamètres de KNN

```
parameters = {
    'n_neighbors' : list(np.arange(3, 50, 2)),
    'weights': ['uniform', 'distance'],
    'p' : [1, 2, 3, 4]
}
```

Figure III. 48: Paramètres de KNN

Dans ce cas précis, le code explore les paramètres suivants :

- **n_neighbors**: le nombre de voisins à utiliser pour le calcul de la distance
- **weights:** le type de pondération à utiliser pour les voisins
- **p :** le paramètre **p** utilisé pour le calcul de la distance de Minkowski

- **Test**

Nous pouvons ajuster les paramètres 'n_ neighbors 'et 'leaf-size' pour effectuer une sélection minutieuse, et l'optimisation de ces paramètres est cruciale pour garantir des performances et une généralisation optimale du modèle, comme indiqué dans le tableau III.6.

Tableau III. 5:Résultats des tests pour l'apprentissage d'un modèle KNN

Test_Size	leaf_size	n_neighbors	**Accuracy**
20%	10	10	0.61
	30	50	0.77
	50	100	0.90
	30	50	0.93
	50	100	0.89
50%	10	10	0.91
	30	50	0.70
	50	100	0.88

- **Résultat**

Les figures III.48 et III.49 donnent les matrices de confusion correspondante aux tailles des données de test de 20% et 50% respectivement. On note que la taille de 20% donne de meilleurs résultats.

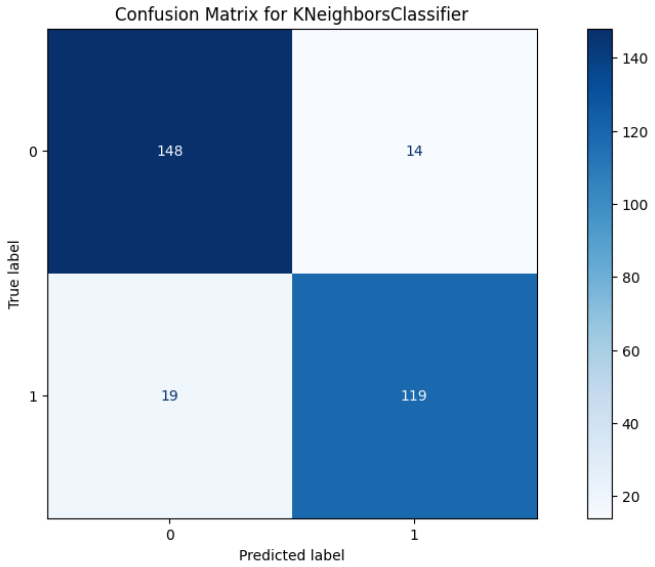


Figure III. 49: Matrice de confusion de l'apprentissage KNN pour un taux de test de 20%

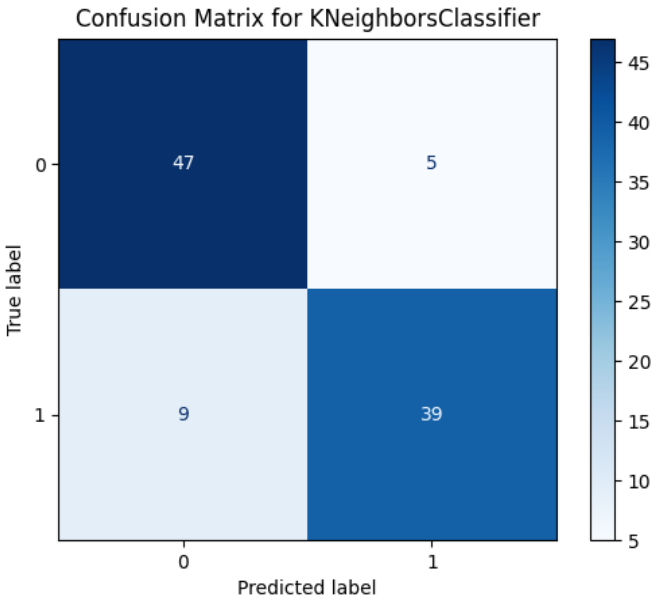


Figure III. 50: Matrice de confusion de l'apprentissage KNN pour un taux de test de 50%

III.6.4.5Comparaison entre les quatre classificateurs

Après avoir testé les quatre algorithmes en modifiant les paramètres de chacun d'eux, nous avons choisi la meilleure précision et la meilleure matrice de confusion pour chaque algorithme afin de déterminer celui sur lequel nous nous appuierons pour la classification.

III.6.4.5.1Comparaison de la matrice de confusion

Dans cette section, les matrices de confusion de quatre modèles sont comparées, comme le montre la figure III.50. L'analyse de ces matrices fournit des informations précieuses sur la précision des tâches de classification. Le modèle RF montre de bons résultats par rapport aux autres..

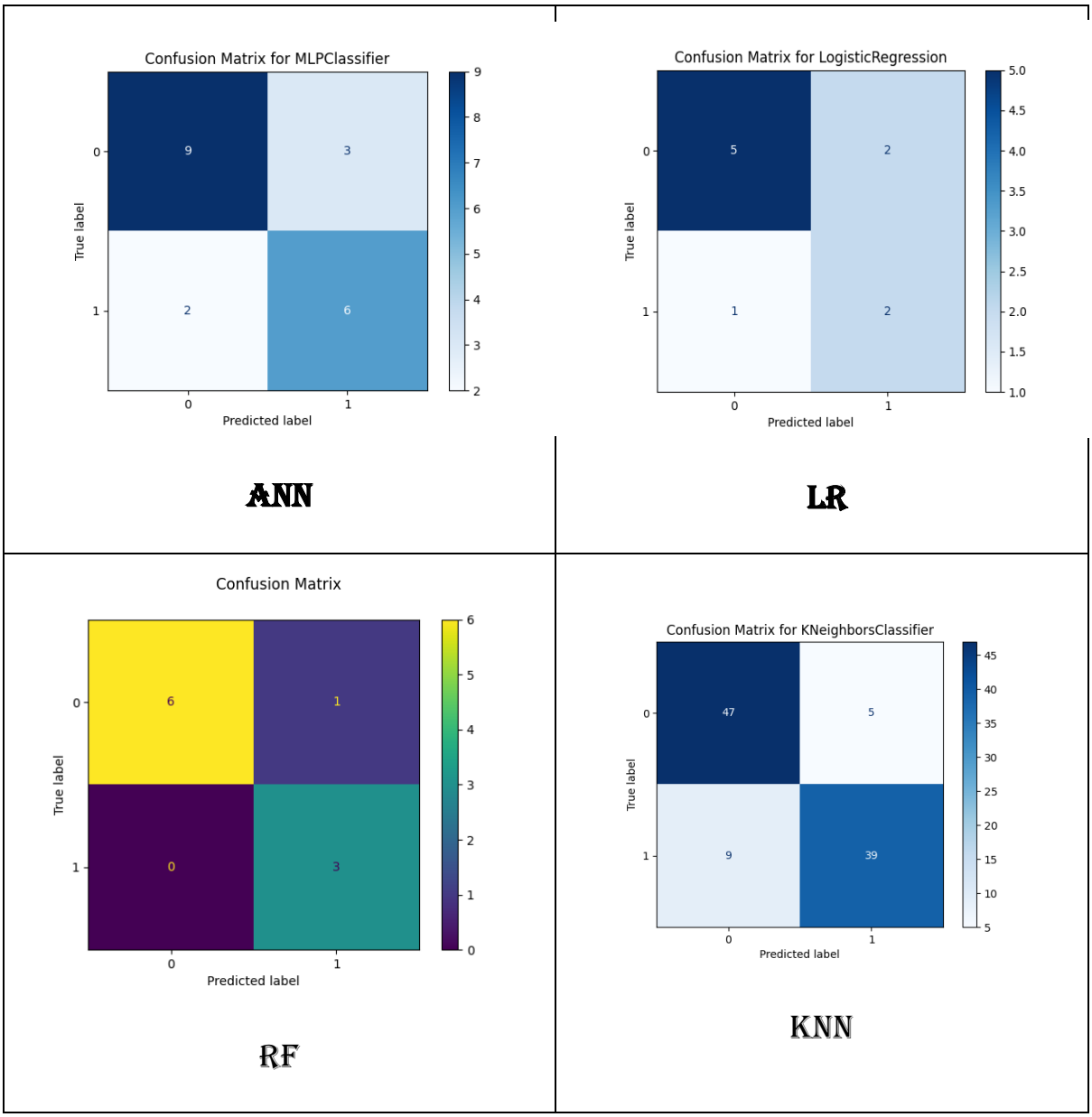


Figure III. 51: Matrices de confusion des quatre classificateurs

III.6.4.5.2Comparaison de précision

Nous comparons les performances des quatre modèles dans la figure III.52. En termes de précision (accuracy), le modèle RF se révèle également supérieur par rapport aux autres.

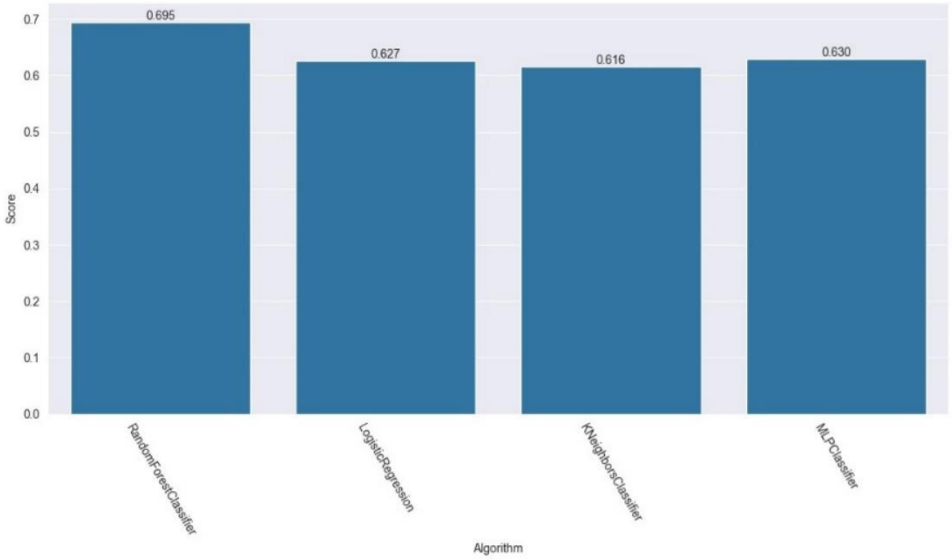


Figure III. 52: Distribution des précisions des quatre modèles de classification

III.7Node-RED

La plateforme Node-RED d'IBM est utilisé pour le développement de l'IHM de supervision et notification.

III.7.1 Lancement sous Windows

La console Windows doit être lancée en mode administrateur. Ensuite, il faut naviguer vers le dossier : c:\users\sb\appdata\roaming\npm. La commande Node RED doit être exécutée, et l'accès doit être autorisé (Figure III.53).

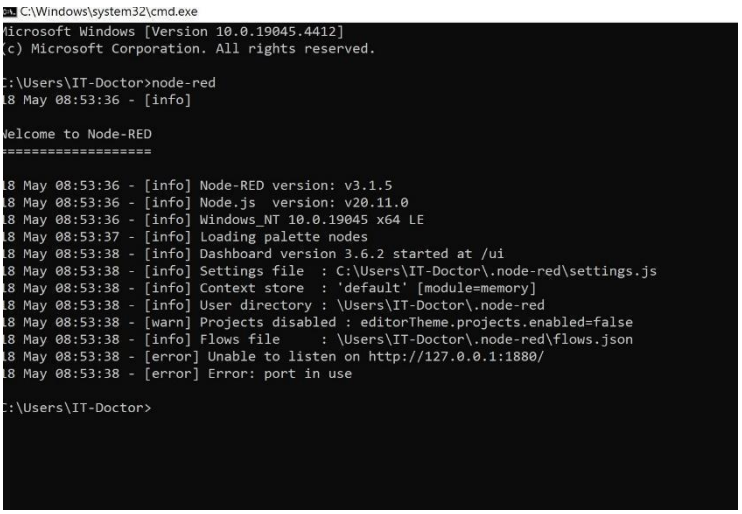


Figure III. 53: Lancement de Node-Red sur cmd

Ensuite accédez avec un navigateur à l'IP http://127.0.0.1:1880/ où l'interface de Node RED fonctionne. Pour tester le Dashboard, utilisez l'IP: http://127.0.0.1:1880/ui/#

III.7.2 Installation de Node-RED

Pour installer de nouveaux nœuds (Nodes) dans la palette, on utilise le menu contextuel en choisissant : "manage palette" (Figure III.53).

Dans "manage palette", accédez à l'onglet "Install" et effectuez une recherche, par exemple : "serial" (pour Bluetooth) ou "modbustcp" (pour Adam). Installez les nœuds qui vous intéressent. Une fois installés, ces nœuds apparaîtront dans la palette.

Ensuite, un flux est créé dans Node-Red en connectant le nœud "Serial In" qui reçoit les données de l'IDE Arduino (Figure III.54).

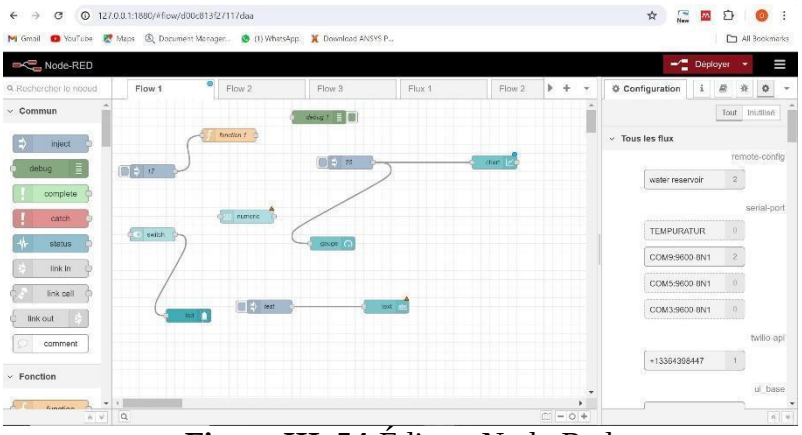


Figure III. 54:Éditeur Node-Red



Figure III. 55: Nœud serial in

L'entrée série est choisie pour récupérer des données provenant de services externes, puis elle est implémentée dans Node-Red. Après avoir configuré l'entrée série avec les détails d'Arduino, comme illustré ci-dessous (Figure III.55).

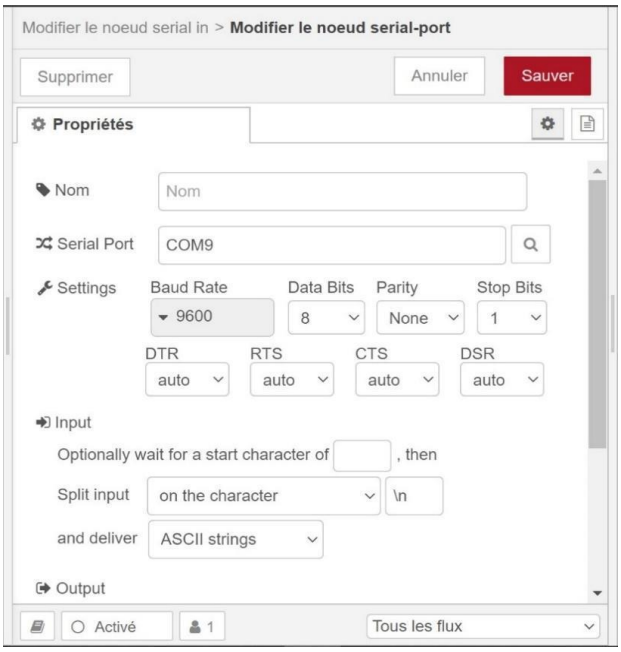


Figure III. 56: Configuration du nœud d'entrée série

Ensuite, les données sont envoyées aux nœuds de traitement où elles sont d'abord divisées par le nœud JSON pour être organisées en leurs propres entités, telles que le nœud de fonction pH pour le pH (Figure III.56).

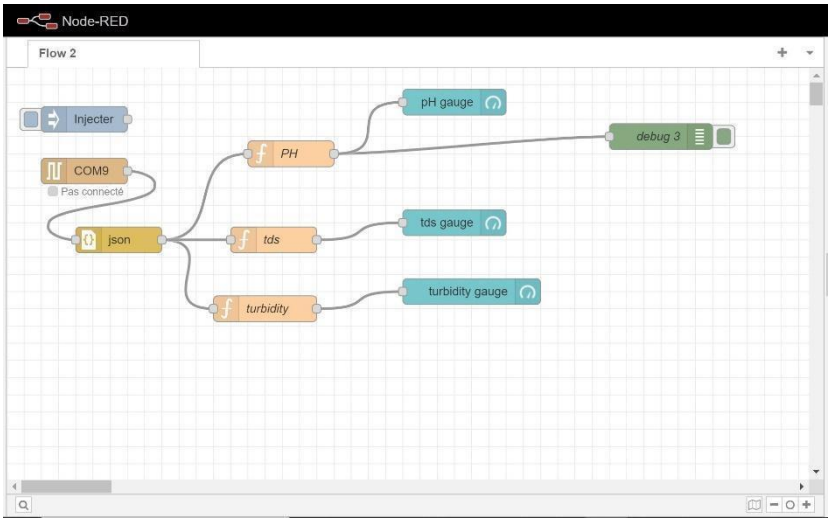


Figure III. 57: Fonction Json

Dans le même temps, ces entrées sont affichées dans un tableau de bord de l'interface utilisateur en utilisant des nœuds de tableau de bord comme les nœuds de jauge et les nœuds de notification, qui doivent également être configurés avec les propriétés nécessaires (Figure III.58).

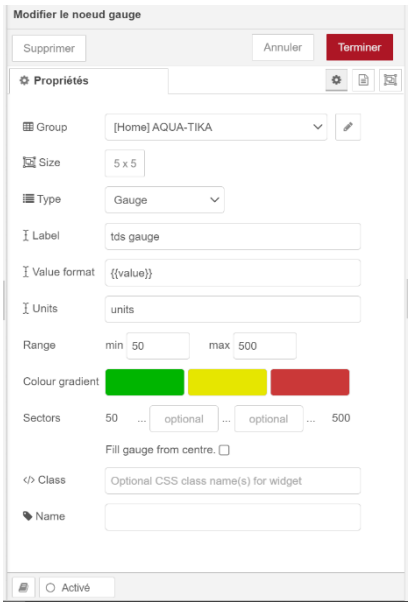


Figure III. 58: Exemple de configuration d'une jauge

Un ensemble de fenêtres au niveau du flux est utilisé pour modifier les paramètres dans l'interface, comme illustré par la Figure III.58

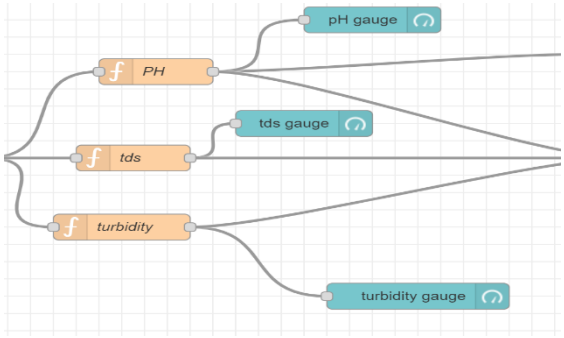


Figure III. 59: Nœuds du Dashboard

Cette figure montre l'IHM resultanat qui peut être obtenu sur l'adresse : http://127.0.0.1:1880/ui

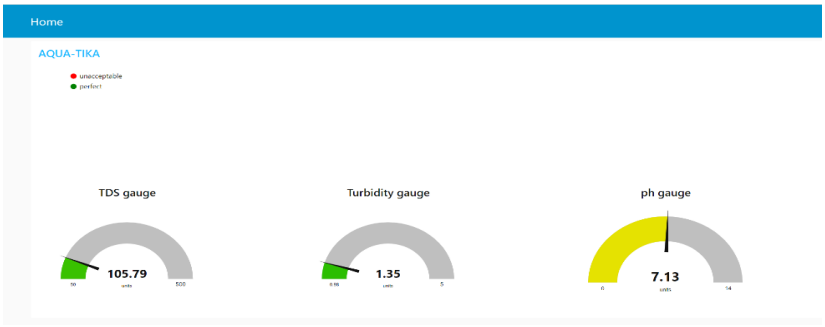


Figure III. 60: Interface de surveillance de la qualité de l'eau

III.7.3 Génération d'alerte

Pour présenter la surveillance continue et l'interface des données en temps réel, un exemple est donné où les seuils suivants sont dépassés : pH à 7, turbidité à 10 et TDS à 500. En conséquence, les nouvelles modifications sont immédiatement affichées sur l'interface, accompagnées d'une notification et d'une sortie audio, comme indiqué dans la figure III.60.

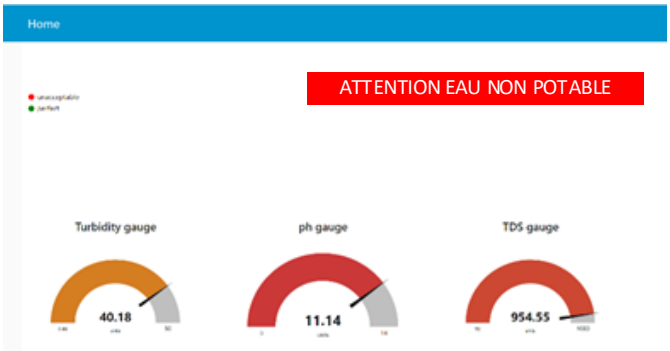


Figure III. 61: Interface en cas d'Alerte

III.8 Utilisation des requêtes pour communiquer avec l'API HTTP de Node-RED depuis Python

Node-RED, Python et les bibliothèques nécessaires pour notre script Python ont été téléchargés.

- Un nœud "Inject" est ajouté : Un nœud "Inject" est fait glisser depuis la palette sur le tableau et configuré pour envoyer un message à intervalle régulier ou lors d'un événement spécifique.
- Un nœud "Python Shell" est ajouté : Un nœud "Python Shell" est fait glisser depuis la palette sur le tableau.

La sortie du nœud "Inject" est connectée à l'entrée du nœud "Python Shell"..



Figure III. 62: Noeud "pythonshell in"

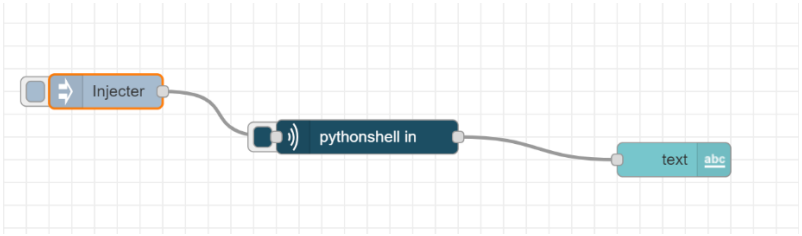


Figure III. 63: Utilisation de python on Node Red

- **Déployer le Flux** :

Nous cliquons sur le bouton "Deploy" en haut à droite pour déployer notre flux.

- **Tester l'Injection** :

Nous cliquons sur le bouton du nœud "Inject" pour déclencher l'exécution du script Python via le nœud "Function".

- Vérification des Résultats

Après avoir injecté le message et exécuté le script, nous pouvons voir la sortie dans la console de Node-RED ou dans le panneau de débogage si nous avons connecté un nœud "debug" à la sortie du nœud "python shell". Figure (III.63)

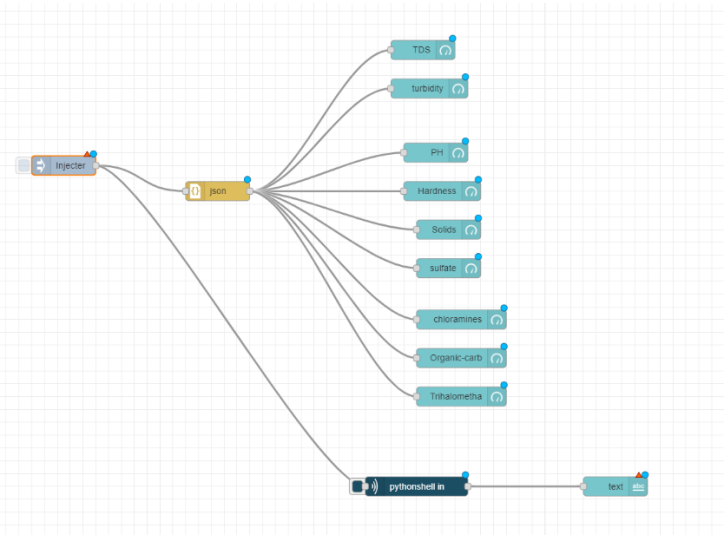


Figure III. 64: Flux de traitement des paramètres de qualité de l'eau

Après avoir mené les tests, au cours desquels nous avons intégré le programme Python à Node-Red et utilisé les quatre modèles d'apprentissage machine, nous avons conclu que RF fournissait les meilleurs résultats pour notre système. RF a donc été choisi comme modèle d'apprentissage machine pour notre système, et les tests suivants ont été réalisés.

- **Test**

Dans la figure III.64, il est observé que les résultats des tests nous ont donné 1, ce qui signifie eau potable

Potabilité: 1



Figure III. 65: Test 1 de potabilité de l'eau

Dans la figure III.65, il est observé que les résultats des tests nous ont donné 0, ce qui signifie eau non potable.

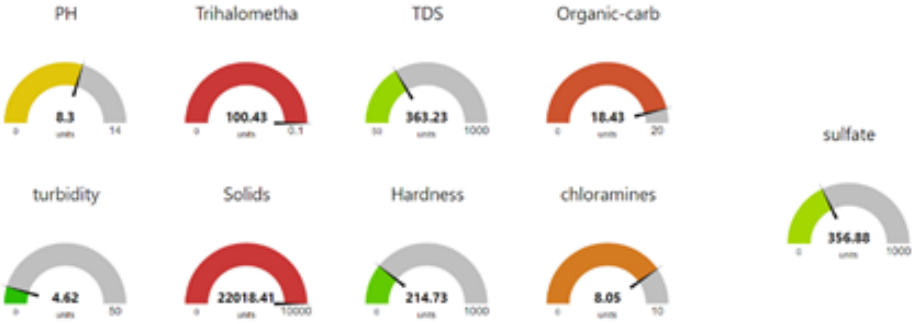


Figure III. 66: Test 2 de potabilité de l'eau

Sur la figure III.66, on remarque que les résultats des tests nous a donné 1, signifiant eau potable. Mais on remarque que dans la jauge de TDS donne un résultat inapproprié, et on en conclut que l'apprentissage automatique RF fait parfois des erreurs justifié par une accuracy de une peu plus de 68%.

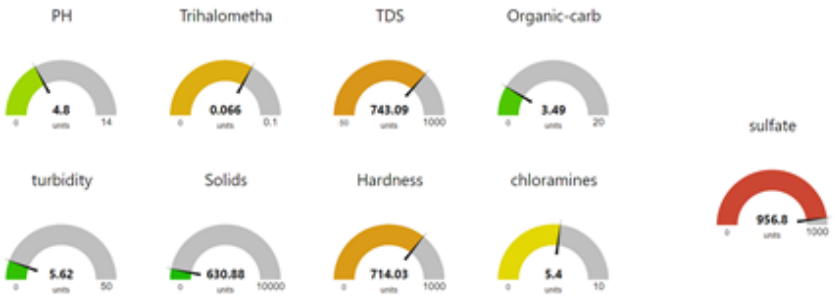


Figure III. 67: Test 3 de potabilité de l'eau

III.9 Conclusion

Ce chapitre a présenté un aperçu complet du système proposé pour le contrôle de la qualité de l'eau, en détaillant l'interfaçage des capteurs avec Arduino, notamment les pH-mètres, les capteurs de turbidité et les capteurs TDS. Il a inclus l'analyse du fonctionnement, l'étalonnage, la programmation Arduino, et les tests.

Le chapitre a également couvert la réalisation pratique du système, y compris le développement et le test du prototype, validant ainsi son efficacité et sa précision. L'analyse des corrélations entre les paramètres de qualité de l'eau et l'utilisation d'algorithmes de machine Learning (ANN, LR, RF, K-NN) ont amélioré l'analyse des données.

Enfin, la mise en place et la configuration de Node-RED, ainsi que son intégration avec Python pour la gestion des alertes, ont renforcé la capacité du système à réagir rapidement en cas de détection de paramètres anormaux. En conclusion, le système conçu est efficace et robuste pour le contrôle de la qualité de l'eau.

Conclusion générale

La conception et la mise en œuvre d'un système intelligent de contrôle de la qualité de l'eau représentent une avancée technologique majeure dans la gestion des ressources en eau. Cette étude répond efficacement à la problématique de la surveillance en temps réel et de la réactivité face aux contaminations de l'eau potable. À travers cette mémoire, nous avons exploré chaque étape, de la conception initiale au développement final, démontrant ainsi le potentiel de ce système à améliorer la sécurité et la surveillance de l'eau potable.

L'intégration dispositifs électroniques telles que les capteurs de pH, de turbidité et de conductivité, les microcontrôleurs Arduino, ainsi que les algorithmes d'apprentissage automatique (ANN, KNN, SVM, RF), représente l'un des principaux avantages de notre système. Ces technologies permettent une analyse précise et continue des paramètres de l'eau, assurant une surveillance en temps réel et une réactivité accrue face aux variations de qualité.

Nous avons également mis en place une interface utilisateur conviviale grâce à Node-RED, permettant la supervision en temps réel des données de qualité de l'eau. Cette interface permet non seulement de visualiser les données mais aussi de générer des alarmes en cas de dépassement des seuils critiques. Cette fonctionnalité renforce la réactivité du système en alertant immédiatement les opérateurs en cas de situation anormale.

Nos tests ont confirmé la performance et la fiabilité du système, mais ont également mis en lumière des axes d'amélioration pour renforcer ses capacités. Des ajustements sont nécessaires pour optimiser son efficacité et sa robustesse dans diverses conditions environnementales. Cela pourrait inclure l'intégration de nouveaux capteurs, l'optimisation des algorithmes d'apprentissage automatique, et l'amélioration de la connectivité réseau pour une transmission plus rapide et plus fiable des données.

Ce travail ouvre la voie à des recherches futures visant à perfectionner et à élargir les capacités de ces systèmes. Les prochaines études pourraient explorer l'utilisation de technologies émergentes telles que l'intelligence artificielle et le machine learning pour améliorer davantage la précision et la rapidité de détection des anomalies. De plus, une adoption plus large de ces systèmes pourrait être facilitée par des collaborations avec des organismes de régulation et des initiatives de sensibilisation au sein des communautés locales.

En conclusion, la mise en œuvre de ce système intelligent de contrôle de la qualité de l'eau, intégrant Node-RED pour la supervision en temps réel et la génération d'alarmes, contribue significativement à la protection de la santé publique et à la sécurité des ressources en eau. Elle pose également les bases pour des recherches futures visant à optimiser et à étendre les capacités de ces systèmes, garantissant ainsi un accès durable et fiable à une eau potable de qualité pour les générations futures.

Références bibliographiques

[1] Benyezza, Hamza, Bouhedda, Mounir, et Rebouh, Samia. Zoning irrigation smart system based on fuzzy control technology and IoT for water and energy saving. Journal of Cleaner production, 2021, vol. 302, p. 127001..

[2] Liu, L., Sun, X., Fang, J., & Hossin, Md. A. (2024). Exploring Prototypicality Adherence Effects on User Engagement in User-Generated Video Platforms. *International Journal of Electronic Commerce*, *28*(3), 381–415. https://doi.org/10.1080/10864415.2024.2361400

[3] Kanouté, Yamadou. Evaluation de la qualité de l'eau de boisson à Mopti et Sévaré. 2019. Thèse de doctorat. USTTB, Mali.

[4] N'Diaye, A. Etude bactériologique des eaux de boissons vendues en sachet dans quatre communes d'Abidjan. Université de Bamako: Faculté de Médecine, de Pharmacie et d'Odontostomatologie. Mali, 2008.

[5] "Normes de l'OMS sur l'eau potable." Accessed: Mar. 06, 2024. [Online]. Available: https://www.who.int.

[6] Ibrahim, Berraf et Abdraouf, Ramdane. Contrôle qualité de l'eau potable. Thèse de doctorat. Université Kasdi Merbah de Ouargla..

[7] REJSEK, Franck. *Analyse des eaux: Aspects réglementaires et techniques*. Centre régional de documentation pédagogique d'Aquitaine, 2002.

[8] Serrat and Pierre, "Compte rendu de : MUSY A. et HIGY C., 2004, Hydrologie (1. Une science de la nature), Presses polytechniques et universitaires romandes, (21), 314 p.," *http://journals.openedition.org/geocarrefour*, vol. 81, no. vol. 81/1, pp. 72-, Jan. 2006, doi: 10.4000/GEOCARREFOUR.1794.

[9] D. SA, J.-L. Chaussade, and G. Mestrallet, *Memento technique de l'eau. Tome 1*, 10e édition. Rueil Malmaison: Degrémont Suez, 2005.

[10] F. Nordmann and G. P. Legry, "Chimie de l'eau et corrosion dans les REP," *Ref : TIP180WEB - "Génie nucléaire,"* Oct. 2000, doi: 10.51257/A-V1-BN3750.

[11] Mehta, Mayuri, PALADE, Vasile, et Chatterjee, Indranath (ed.). Explainable AI: Foundations, methodologies and applications. Springer Nature, 2022.

[12] Maillot, Richard. Mémento technique des granulats. Presses des MINES, 2001.

[13] Gromaire· Mertz, Marie-Christine. La pollution des eaux pluviales urbaines en réseau d'assainissement unitaire-Caractéristiques et origines. La Houille Blanche, 2000, vol. 86, no 2, p. 66-70.

[14] Cazoulat, A., Lecompte, Y., Bohand, S., *et al.* Self-absorption alpha particle factor in water: interest in the monitoring of specific military sites. *Radioprotection*, 2007, vol. 42, no 1, p. 11-27.

[15] Bouziani, Mustapha. *L'eau: de la pénurie aux maladies*. Ed. Ibn-Khaldoun, 2000.

[16] Jetha, Nina, Robinson, Kerry, Wilkerson, Tricia, *et al.* Supporting knowledge into action: the Canadian best practices initiative for health promotion and chronic disease prevention. *Canadian Journal of Public Health/Revue Canadienne de Santé Publique*, 2008, p. I1-I8.

[17] Sigg, Laura, Behra, Philippe, et Stumm, Werner. Chimie des milieux aquatiques. Paris, France : Dunod, 2001.

[18] A. N. Ghenim and A. Megnounif, "Analysis of rainfall in Northwestern Algeria," *Sécheresse*, vol. 24, no. 2, pp. 107–114, Apr. 2013, doi: 10.1684/sec.2013.0380.

[19] "Management of Legionella in Water Systems," *Management of Legionella in Water Systems*, Feb. 2019, doi: 10.17226/25474.

[20] Fondation Nationale de la Santé. “Manuel pratique d’analyse de l’eau, 4ème édition.” Brasilia, 2013, pp. 57, 67. Accessed: Mar. 10, 2024. [Online].

[21] Kitchener, Ben GB, Wainwright, John, et Parsons, Anthony J. A review of the principles of turbidity measurement. Progress in Physical Geography, 2017, vol. 41, no 5, p. 620-642.

[22] CEAEQ. Détermination de la conductivité: méthode électrométrique, MA. 115–Cond. 1.1, rév. 1 (Determination of Conductivity: Electrometric Method, MA. 115–Cond. 1.1, rev. 1). 2015.

[23] Pandolfi, Denis et Dumas, M. D. Caractérisation morphologique et physiologique de la biomasse des boues activées par analyse d’images. Institut National Polytechnique de Lorraine: Nancy, France, 2006.

[24] A. Rannou, “Connaissance actuelle des sources d’irradiation naturelle,” *Radioprotection*, vol. 34, no. 4, pp. 505–519, Oct. 1999, doi: 10.1051/radiopro:1999114.

[25] Rodier, Jean, Legube, Bernard, et Merlet, Nicole. L'analyse de l'eau-10e éd. Dunod, 2016..

[26] Alley, E. Roberts, et al. Water quality control handbook. New York : McGraw-Hill, 2007.

[27] Bouyer, Amandine. *Impact du type de fluide laitier sur les performances du procédé d'osmose inverse pour la production d'eau de procédé*. 2020. Thèse de doctorat. Université Laval.

[28] Guo, Gongde, Wang, Hui, Bell, David, et al. KNN model-based approach in classification. In : On The Move to Meaningful Internet Systems 2003: CoopIS, DOA, and ODBASE: OTM Confederated International Conferences, CoopIS, DOA, and ODBASE 2003, Catania, Sicily, Italy, November 3-7, 2003. Proceedings. Springer Berlin Heidelberg, 2003. p. 986-996.

[29] Mondiale de la Santé, Assemblée. *Le financement futur de l’OMS: Organisation mondiale de la Santé: réformes pour un avenir sain: rapport du Directeur général: résumé d’orientation*. Organisation mondiale de la Santé, 2011.

[30] OMS (Organisation mondiale de la Santé). “Évaluation rapide de la qualité de l’eau de boisson: Manuel de mise en œuvre.” OMS, Genève, Suisse, 2012. Accessed: Mar. 10, 2024. [Online].

[31] S. Zhang and J. Li, “KNN Classification With One-Step Computation,” *IEEE Trans Knowl Data Eng*, vol. 35, no. 3, pp. 2711–2723, Mar. 2023, doi: 10.1109/TKDE.2021.3119140.

[32] Parmar, Aakash, Katariya, Rakesh, et Patel, Vatsal. A review on random forest: An ensemble classifier. In : International conference on intelligent data communication technologies and internet of things (ICICI) 2018. Springer International Publishing, 2019. p. 758-763.

[33] PAL, Mahesh. Random forest classifier for remote sensing classification. International journal of remote sensing, 2005, vol. 26, no 1, p. 217-222.

[34] Shaik, Anjaneyulu Babu et Srinivasan, Sujatha. A brief survey on random forest ensembles in classification model. In : International Conference on Innovative Computing and Communications: Proceedings of ICICC 2018, Volume 2. Springer Singapore, 2019. p. 253-260.

[35] Hammyani, Asma. Amélioration des forêts aléatoires: Application au diagnostic médical. 2014. Thèse de doctorat, Université de Tlemcen.

[36] Nguyen, Nicolas. IPT-Informatique pour tous: Algorithmique, programmation Python, ingénierie numérique et simulation (bibliothèques Numpy et Scipy de Python)-Tout le programme de prépas scientifiques en 1 clin d'oeil. 2018.

[37] Barrett, Steven F. Arduino microcontroller processing for everyone!. Springer Nature, 2022.

[38] BLUM, Jeremy. Exploring Arduino: tools and techniques for engineering wizardry. John Wiley & Sons, 2019.

[39] Festy, Bernard, Hartemann, Philippe, Ledrans, Martine, et al. Qualité de l'eau. Environnement et santé publique-Fondements et pratiques, 2003, p. 333-368.

[40] Stoppiglia, Hervé. *Méthodes statistiques de sélection de modèles neuronaux; applications financières et bancaires*. 1997. Thèse de doctorat. ESPCI ParisTECH.

[41] Dégbey, Cyriaque, Makoutode, Michel, Ouendo, Edgard-Marius, *et al.* The quality of well water in the municipality of Abomey-Calavi in Benin. *Environnement, Risques & Santé,* 2008, vol. 7, no 4, p. 279-283..

[42] Martin, Hondjenou, Rodrigue, Landeou Codjo, Thierry, Azonhe, *et al.* Facteurs associes a la qualité des eaux consommées dans la Commune d'Abomey-Calavi au Sud du Benin en Afrique de L'Ouest. *European Scientific Journal ESJ*, 2019, vol. 15, no 30.

[43] "Kaggle: Your Machine Learning and Data Science Community." Accessed: Jun. 19, 2024. [Online]. Available: https://www.kaggle.com/