

République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



UNIVERSITE SAAD DAHLEB DE BLIDA

Faculté des Sciences

Département Informatique

Projet de fin d'étude pour l'obtention du diplôme Master

Option

Ingénierie des logiciels + Systèmes informatique et réseaux

Conception et Réalisation d'un Système de Gestion et de Contrôle d'Anomalies dans un Service de Radiothérapie

Présenté par :

HAMIDA abderrahmene

DJEDRI aimen lotfi

Organisme d'accueil :



Soutenu le 09/12/2020, devant le jury composé de :

Dr. GUESSOUM dalila	Présidente
Dr. CHERIGUENE soraya	Examinatrice
Dr. LAHIANI nesrine	Promotrice
Dr. ALLAM abdelkrim	Encadreur

Promotion 2019/2020

Remerciements

Nous tenons tout d'abord à remercier Dieu le tout puissant et miséricordieux, qui nous a donné la force et la patience d'accomplir ce modeste travail. Ce travail est l'aboutissement d'un long cheminement au cours duquel nous avons bénéficié d'encadrement, des encouragements et du soutien de plusieurs personnes, à qui nous tenons à dire profondément et sincèrement merci.

Nous exprimons notre grande gratitude à notre promotrice Mme LAHIANI Nesrine, d'avoir accepté de nous encadrer durant notre travail et pour ses précieux conseils et ses orientations.

Nous remercions très chaleureusement Mr. ALLAM Abdelkrim pour son soutien, ses précieux conseils tout au long de notre stage au niveau du CDTA. Nous avons eu le privilège de travailler parmi votre équipe et d'apprécier vos qualités et vos valeurs, votre sérieux et votre compétence.

Nous remercions également les membres du jury pour l'intérêt qu'ils ont porté à notre travail et pour les remarques et critiques qui permettront l'amélioration de ce travail. Bien évidemment, nous remercions nos familles et nos amis pour leurs encouragements et leurs soutiens.

Enfin, nous tenons à remercier tous ceux et toutes celles qui ont contribué, de près ou de loin, à la réalisation de ce travail.

DEDICATION

C' est un grand honneur pour nous de pouvoir dédier ce modeste travail :

À tous ceux qui nous sont très chers.

À la source éternelle d'honneur et d'affection, nos très chers et honorables parents, qui ont toujours veillé sur notre bonheur en nous assurant tous moyens.

À ceux qui ont partagé nos joies et nos peines.

À nos frères et sœurs que dieu les garde.

À toutes nos grandes familles.

À tous nos amis et camarades sans exception.

À tous nos enseignants.

À vous tous,

Abderrahmene et Lotfi

Résumé

La radiothérapie est le dernier stade de traitement du cancer après la chirurgie et la chimiothérapie, c'est un domaine extrêmement sensible où chaque erreur peut coûter la vie des patients, plusieurs causes entrent en jeu dont la gestion des patients, l'organisation des traitements et le contrôle des équipements et des machines. Notre objectif est de réduire le maximum possible les erreurs et les risques pendant la chaîne de traitement du patient en se focalisant sur les anomalies provenant des machines.

Dans ce travail nous présentons une partie d'un système de gestion et de contrôle d'anomalies que nous avons réalisé pour aider à éviter les erreurs de manipulation et la gestion dans les salles de traitement de cancer consacré aux services de radiothérapie au niveau de l'hôpital cpmc. Ce système contient un check-list des machines de traitement et reçoit les données des capteurs de températures, d'humidité et pression d'eau placés dans les salles de radiothérapie. Notre travail a permis de classifier ces données en utilisant le réseau de neurone et de créer un système de décision pour le traitement du patient, nous avons contribué encore plus à la conception d'un système expert pour les anomalies acquis depuis l'application. Les résultats obtenus justifient que le système que nous proposons améliore la qualité de la cure de traitement et nous a permis de réduire considérablement les erreurs.

Mots clés :

radiothérapie, système, anomalies, check-list, machines de traitement, capteurs, réseau de neurone, système de décision pour le traitement, système expert.

Abstract

Radiotherapy is the last stage of cancer treatment after surgery and chemotherapy, it is an extremely sensitive area where every mistake can cost the life of patients, several causes go into this including patient management, organization of treatment, control of equipment and machinery. Our goal is to reduce errors and risks during the patient treatment chain as much as possible by focusing on the abnormalities coming from the machines.

In this work we present part of a management and abnormalities control system we have realized to help avoid handling errors and management in the cancer treatment rooms dedicated to radiotherapy services at the CPMC hospital level. This system contains a checklist of treatment machines and receives data from temperature, humidity and water pressure sensors placed in the radiotherapy rooms. Our work has allowed us to classify this data using the neural network and to create a decision system for the treatment of the patient, we have further contributed to the design of an expert system for the abnormalities acquired since the application. The results obtained justify that the system we propose improves the quality of the treatment cure and has allowed us to considerably reduce errors.

Keywords :

radiotherapy, system, abnormalities, checklist, treatment machines, sensors, neural network, system for the treatment of the patient, expert system.

Table des matières

Liste des figures	xi
Liste des tableaux	xii
Liste des abréviations	xiii
Introduction générale	1
1 Généralités sur l'IA et les reseaux de neurones	3
Introduction	3
1.1 L'intelligence artificielle	4
1.1.1 Définition de L'IA	4
1.1.2 Domaine d'application de l'IA	4
1.1.3 Quelques exemples d'usage d'IA	5
1.1.4 Les avantages et les inconvénients de l'intelligence artificielle	6
1.1.4.1 Les avantages	6
1.1.4.2 Les inconvénients	7
1.2 Les réseaux de neurones	8
1.2.1 Historique	8
1.2.2 Le Neurone	8
1.2.2.1 Modèle d'un neurone biologique	8
1.2.2.2 Modèle d'un neurone artificielle (forme)	9
1.2.3 Fonctionnement d'un neurone artificielle	10
1.2.4 Fonction d'activation	10
1.2.5 Architecture des réseaux de neurones	11
1.2.5.1 Les Réseaux de neurones mono couches	11
1.2.5.2 Les Réseaux de neurones multicouches	12
1.2.5.3 Les Réseaux de neurones à connexions locales	13
1.2.6 Caractéristiques des RN	13
1.2.6.1 Le parallélisme	13
1.2.6.2 L'apprentissage	13

1.2.6.2.1	L'apprentissage supervisé	13
1.2.6.2.2	L'apprentissage non supervisé	13
1.2.6.2.3	Le renforcement	14
1.2.6.2.4	Le mode hybride	14
1.2.7	Les avantages et les inconvénients de réseaux de neurone	14
1.2.7.1	Les avantages	14
1.2.7.2	Les inconvénients	14
	Conclusion	15
2	État de l'art sur les systèmes experts	16
	Introduction	16
2.1	Définition d'un système expert	16
2.2	Structure d'un système expert	17
2.2.1	Moteur d'inférence	18
2.2.1.1	Chaînage avant	18
2.2.1.2	Chaînage arrière	19
2.2.1.3	Chaînage Mixte	20
2.3	Domaine d'application des systèmes experts	20
2.3.1	Quelques systèmes experts classique	20
2.3.2	Représentation des connaissances	21
2.3.2.1	Les réseaux sémantiques	21
2.3.2.2	Les représentations logiques	22
2.3.2.3	Les règles de production	22
2.3.2.4	Les objets	23
2.4	Le cycle de base d'un moteur d'inférence	24
2.4.1	phase d'évaluation	25
2.4.1.1	Le filtrage	25
2.4.1.2	La sélection avec résolution de conflits	25
2.4.2	La Phase d'exécution	25
2.5	Stratégie de recherche	26
2.5.1	La recherche en largeur	26
2.5.2	La recherche en profondeur	26
2.6	Réalisation d'un SE	26
2.6.1	Phase de conception	26
2.6.2	Phase d'implantation	27
2.7	les avantages et les inconvénients d'un système expert	27
2.7.1	Les avantages d'un système expert	27

2.7.2	Les inconvénients d'un système expert	28
	Conclusion	28
3	Application du réseau de neurone pour la classification des données de capteurs	29
	Introduction	29
3.1	Présentation des capteurs	29
3.1.1	Le capteur de température et humidité	29
3.1.2	Le capteur de pression d'eau	30
3.2	Pourquoi le réseau de neurone?	30
3.3	Procédure du construction d'un réseau de neurones	30
3.3.1	Collecte d'une base de données	31
3.3.2	Séparation de la base de données	32
3.3.3	Choix de l'architecte de réseau de neurone	32
3.4	Résultat et discussion	33
3.4.1	Les modèles utilisés	34
3.4.1.1	Résultat modèle 1 (5x5) :	34
3.4.1.2	Résultat modèle 2 (10x10) :	35
3.4.1.3	Résultat modèle 3 (15x15) :	35
	Conclusion	36
4	Etude conceptuelle	37
4.1	Introduction	37
4.2	Solution proposée	37
4.2.1	Présentation des composants	37
4.3	UML	38
4.3.1	Définition d'UML	38
4.4	La Modélisation de l'Application web	39
4.5	Expression des besoins	39
4.5.1	Les besoin techniques	39
4.5.2	besoin fonctionnels	39
4.5.2.1	Exigences liées au système	39
4.5.2.2	Les cas d'utilisation (UC)	40
4.5.2.2.1	Identification des Acteurs	40
4.5.2.2.2	Identification des cas d'utilisation	40
4.5.2.2.3	Diagrammes de cas d'utilisation	41
4.5.2.2.4	Description textuelle de cas d'utilisation	44
4.6	L'analyse des besoins	46

4.6.1	Diagramme de séquence (DES)	46
4.6.1.1	Diagramme de séquence de cas d'utilisation (S'authentifier)	47
4.6.1.2	Diagramme de séquence de cas d'utilisation (Réparer les anomalies)	47
4.6.1.3	Diagramme de séquence de cas d'utilisation (Remplir le check-list)	48
4.6.1.4	Diagramme de séquence de cas d'utilisation (Gérer les utilisateurs)	49
4.6.1.5	Diagramme de séquence de cas d'utilisation (Configurer les machines)	50
4.7	Conception détaillée	50
4.7.1	Diagramme de classe	50
4.8	Conception de l'application labview	51
4.8.1	système de décision de traitement	51
4.8.1.1	Le schéma conceptuel du système de décision de traitement	52
4.8.1.2	Raisonnement du système de décision	53
4.8.2	Le système expert	53
4.8.2.1	Représentation des Connaissances d'expertise	53
4.8.2.2	Le schéma représentatif du système expert	54
4.9	Conclusion	54
5	Implémentation	55
5.1	Introduction	55
5.2	Choix techniques	55
5.2.1	Langages de programmation	55
5.2.2	Architecture du système (Architecture à trois niveaux)	58
5.2.3	Protocole	58
5.3	Outils matériels	59
5.4	Environnement de développement	59
5.4.1	Outils de développement	59
5.4.1.1	VS Code	59
5.4.1.2	WampServer	59
5.4.1.3	MySQL	60
5.5	Présentation de l'application web CPMC	60
5.5.1	introduction	60
5.5.2	Présentation des interfaces de L'application web CMPC	60
5.5.2.1	Interface Authentification	60

5.5.2.2	Page d'accueil	61
5.5.2.3	Page gestion des utilisateurs (admin)	62
5.5.2.4	Page gestion machine (admin)	63
5.5.2.5	Page check-list	63
5.5.2.6	Page d'intervention (réparation)	64
5.5.2.7	Pages d'aides	64
5.5.2.8	Pages de statistiques	66
5.6	Présentation de l'application labview	67
5.6.1	introduction	67
5.6.2	Présentation de l'interface de l'application labview	67
5.6.3	Interface module de decision de traitement	68
5.6.4	Interface système expert	68
5.7	Conclusion	69
	Conclusion générale	70
	Annexe	71
	Bibliographie	78

Liste des figures

1.1	Le neurone biologique[12]	8
1.2	Le neurone artificielle[13]	9
1.3	Les Réseaux de neurones monocouches.	12
1.4	Réseaux de neurones multicouches.	12
1.5	Réseaux de neurones à connexions locales.	13
2.1	Architecture d'un Système Expert [19]	17
2.2	Représentation d'un réseau sémantique [25]	21
2.3	Représentation de la classe Véhicule [29]	23
2.4	Exemple d'héritage	24
2.5	Cycle de base d'un moteur d'inférence [31]	25
3.1	Flux de base pour la conception d'un modèle de réseau neuronal ar- tificial.	31
3.2	Model 1 accuracy.	34
3.3	Model 1 loss.	34
3.4	Model 2 accuracy	35
3.5	Model 2 loss	35
3.6	Model 3 accuracy.	35
3.7	Model 3 loss.	35
3.8	La Matrice de confusion du modèle 3.	36
4.1	le schéma des composants du système	38
4.2	le diagramme de cas d'utilisations générale de l'application web	41
4.3	Diagramme de cas d'utilisation du "Remplir le check-list"	42
4.4	Diagramme de cas d'utilisation du "Réparer les anomalies"	42
4.5	Diagramme de cas d'utilisation du "Gérer les utilisateurs"	43
4.6	Diagramme de cas d'utilisation du "configurer les machines"	43
4.7	Diagramme de séquence de cas d'utilisation (S'authentifier)	47
4.8	Diagramme de séquence de cas d'utilisation (Réparer les anomalies) . . .	47
4.9	Diagramme de séquence de cas d'utilisation (Remplir le check-list) . . .	48

4.10	Diagramme de séquence de cas d'utilisation (Gérer les utilisateurs) . . .	49
4.11	Diagramme de séquence de cas d'utilisation (Configurer les machines)	50
4.12	Diagramme de classe de l'application web	51
4.13	le schéma conceptuel du système de decision	52
4.14	Le schéma représentatif du système expert.	54
5.1	Interface Authentification	60
5.2	Page d'accueil administrateur	61
5.3	Page d'accueil utilisateur	61
5.4	Page gestion des utilisateurs	62
5.5	Page d'ajout utilisateur	62
5.6	Page gestion machine	63
5.7	Page checklist	63
5.8	Page d'intervention	64
5.9	Page de Codes des verrouillages (anomalies)	65
5.10	Page TOP	65
5.11	Graphe données capteurs avec résultat du reseau de neurone	66
5.12	Page de suivi des réparations	66
5.13	Interface de l'application labview	67
5.14	Interface module de decision de traitement	68
5.15	Interface système expert	68
5.16	Description des composants de capteur MS6503.	71
5.17	Vue du capteur MS6503 avec les 2 principaux boutons à manipuler (Power et Hold).	72
5.18	Mode de fonctionnement du capteur en mode extinction automatique du mode de fonctionnement continu.	73
5.19	fenêtre « panneau de configuration ».	74
5.20	fenêtre « Outils d'administrateur ».	74
5.21	fenêtre « ODBC ».	75
5.22	fenêtre «configuration ODBC ».	75
5.23	Le diagramme de bloc connexion bd labview	76
5.24	Le diagramme de bloc de l'application labview	76
5.25	Algorithme réseau de neurone 1.	77
5.26	Algorithme réseau de neurone 2.	77

Liste des tableaux

1.1	Les fonctions d'activations [12]	10
3.1	Paramètres utilisés	31
3.2	Les données des capteurs.	32
3.3	Les paramètres de chaque modèle	34
4.5	Valeur de l'entrée 2 du SDT	52
4.6	Le raisonnement du SDT	53

Liste des abréviations

- ADD** Arbre de Décision.
- BC** Base de Connaissances
- BF** Base de Faits
- BR** Base de Règles
- CDTA** Centre de Développement Des Technologies Avancées
- CPMC** Centre Pierre et Marie-Curie
- DASM** division Architecture des Systèmes et Multimédia
- IA** Intelligence Artificielle
- ICQ** instrumentation et contrôle de qualité
- KNN** K-Nearest Neighbor
- LabVIEW** Laboratory Virtual Instrument Engineering Workbench
- MI** Moteur d'Inférence
- PHP** HyperText PreProcessor
- RN** réseau de neurone
- SE** Système Expert
- SDT** Système de décision de traitement
- SGBD** Système de Gestion de Base de Données
- SQL** Structured Query Language
- SVM** Support Vector Machine
- TALN** traitement automatique du langage naturel
- UC** Use Case
- UML** Unified Modeling Language

Introduction générale

Vu le nombre de patients venus de plusieurs régions d'Algérie, qui fréquentent le service de radiothérapie du CPMC d'Alger, la gestion de ce service devient de plus en plus complexe et ainsi le processus de traitement est perturbé, ce qui a engendré une qualité de service rendu moyennement appréciée vu la taille des risques des accidents.

La problématique scientifique est de trouver une procédure fiable permettant d'exclure les risques provenant des machines de traitement en radiothérapie.

Les procédures actuelles au niveau mondial sont insuffisantes pour assurer une assurance qualité de traitement en radiothérapie. Il y'a énormément d'intervenants (oncologues radiothérapeutes, physiciens médicaux, manipulateurs, etc.) et les accidents peuvent survenir au niveau de ces équipements modernes et complexes. Des accidents surviennent régulièrement même dans les pays avancés. L'avantage de ce système consiste à optimiser les ressources humaines, avoir une traçabilité pour tout acte médical pendant la chaîne de traitement d'un patient dans le service de radiothérapie et d'aboutir à une meilleure gestion des équipements.

Le travail se fera au niveau de l'équipe instrumentation et contrôle de qualité (ICQ) de la division Architecture des Systèmes et Multimédia (DASM) dans le cadre du projet intitulé « Détection d'anomalies dans une chaîne de traitement médical ». Plusieurs personnes travaillent dans la même thématique (aspect réseau, télémédecine, E-gouvernement et base de données). 6 personnes font partie du projet cité ci-dessus.

l'objectif de ce travail est de proposer une plateforme contenant un check-list qui sera capable de contrôler les anomalies présentent au niveau des machines des salles de traitement du service radiothérapie du "Centre Pierre et Marie Curie" d'alger, ainsi de visualiser sur un écran en utilisant le logiciel LABVIEW les données des capteurs de température, d'humidité et de pression d'eau placées dans les salles de traitement de cancer et de collecter ces données.

Nous allons faire un système de décision de traitement du patient incluant les réseaux de neurones, pour traiter ces données et les classifier en trois classes :

- **T(0)** : Pas d'anomalies, le traitement du cancer peut se dérouler normalement.
- **TA(1)** : Anomalies à corriger durant la phase de traitement du cancer.
- **NT(2)** : Ne pas traiter car la ou les anomalies sont graves et doivent être corrigées, la phase du traitement du cancer est à l'arrêt.

Enfin nous allons proposer un système expert pour la réparation des anomalies des machines de traitements.

Le présent mémoire est organisé en cinq chapitres : Le premier chapitre dresse à savoir l'intelligence artificielle et le réseau de neurone .Le deuxième chapitre dresse l'état de l'art sur les systèmes experts. Le troisième chapitre traite l'application du réseau de neurones pour la classification des données de capteurs. Le quatrième chapitre concerne l'étude conceptuelle de notre système. Le dernier chapitre conclut notre travail en présentant les matériels utilisés, les technologies utilisées et notre interface d'application.

Chapitre 1

Généralités sur l'IA et les reseaux de neurones

Introduction

Vivant dans un monde de technologies où les systèmes informatiques ont pris une place énorme au sein des populations, dans le but de servir l'être humain et son bien être, même si parfois au péril de son métier et de son travail, en le remplaçant par un système qui réfléchit et agit comme lui, tout en étant plus performant et productif que ce dernier.

les réseaux de neurones élargissent le domaine de l'intelligence artificielle. Bien qu'ils y fassent parti, les réseaux de neurones, forment un sujet auquel se penchent plusieurs spécialistes de la recherche informatique et de l'intelligence artificielle, afin de rapprocher les décisions humaines expertes des systèmes informatiques, pour que ces derniers puissent déduire les mêmes résultats attendus ou prédire des nouvelles solutions. Sachant qu'un réseau de neurone peut donner de plus amples résultats en un temps très réduit par rapport à un humain, ceci augmente l'intérêt des chercheurs à travailler dessus et les intégrer dans plusieurs domaines et métiers.

Durant ce chapitre nous allons en premier lieu décrire quelques notions sur l'intelligence artificielle. En deuxième lieu nous définirons les réseaux de neurones en allant de l'historique au avantages et inconvénients.

1.1 L'intelligence artificielle

1.1.1 Définition de L'IA

L'intelligence artificielle (IA) implique la mise en œuvre de diverses technologies conçues pour que les machines imitent une forme d'intelligence réelle. L'IA est impliquée dans de plus en plus de domaines d'application. [1]

L'intelligence artificielle peut être définie comme la science de la simulation des processus cognitifs. Ces processus comprennent :

- l'acquisition des connaissances.
- l'archivage des connaissances.
- l'application des connaissances. [2]

De Google à Microsoft en passant par Apple, IBM et Facebook, toutes les grandes entreprises du monde informatique travaillent désormais sur l'intelligence artificielle, essayant de l'appliquer dans plusieurs domaines spécifiques. De cette manière, chacun a créé des réseaux de neurones artificiels constitués de serveurs et capables de traiter des calculs lourds dans des bases de données géantes. [3]

1.1.2 Domaine d'application de l'IA

L'intelligence artificielle est présente sous plusieurs formes parfois inattendues et dans des domaines variés, mais toutes ses applications ont pour but de simplifier la vie de l'être humain. En effet, elle est présente dans notre vie quotidienne avec notamment les moteurs de recherches, les GPS, l'informatique et les jeux, ou dans d'autres domaines bien plus importants tels que le militaire, la santé ou encore la robotique.

Nous allons présenter ceux qui à nos yeux nous semblent les plus importants, mais cette liste reste incomplète du fait du nombre de domaines existants.

- **Le traitement automatique du langage naturel** : on regroupe sous le vocable de traitement automatique du langage naturel (TALN) l'ensemble des recherches et développements visant à modéliser et reproduire, à l'aide de machines, la capacité humaine à produire et à comprendre des énoncés linguistiques dans des buts de communication. [4].
- **La Santé** : lien patient-médecin, recherche, prévention, diagnostic, traitement. L'intelligence artificielle infiltre tous les domaines de l'industrie médicale. Elle est déjà meilleure que les médecins pour détecter le mélanome cancéreux ou analyser les radiographies pulmonaires.[5]

- **La reconnaissance** : Que ce soit la parole, l'écriture ou le visage, l'intelligence artificielle a fait de grands progrès dans ces actions. [6]
- **L'informatique** : Surtout via les moteurs de recherche : les personnes recherchant des informations sur Internet utiliseront des moteurs de recherche. Ensuite, il fournira tous les sites référençant les informations recherchées par la personne. Le plus connu est Google, Bing ... [6]
- **L'apprentissage automatique** : pour qu'une intelligence artificielle puisse faire son devoir elle doit apprendre les notions données par l'homme. [6]
- **Les systèmes experts** : Un système expert est un logiciel capable de simuler le comportement d'un expert humain effectuant une tâche précise. Le succès de l'intelligence artificielle dans ce domaine est indéniable. [7]

1.1.3 Quelques exemples d'usage d'IA

Les Voitures autonomes : leur technologie se base sur l'apprentissage automatique (ou le machine learning). Ce sont des algorithmes informatiques qui peuvent apprendre par eux-mêmes et donc effectuer des tâches auparavant impossibles. Ces engins apparaissent progressivement dans le trafic réel. Uber a commencé à tester avec de vrais clients en 2016. De plus, lors du dernier salon de l'automobile de Francfort en septembre 2017, chaque groupe de constructeurs automobiles a présenté au moins un prototype de véhicule autonome. Certains proposent des options de «conduite autonome».[8]

Les jeux : Stockfish, champion du monde des logiciels classiques en 2016, a été battu par l'intelligence artificielle d'une filiale de Google à Londres le 5 décembre 2017. Un tournoi d'échecs a été joué entre deux machines dotées de l'IA : l'algorithme d'intelligence artificielle AlphaZero de la société britannique DeepMind rachetée en 2014 par Google et Stockfish, champion du monde des programmes d'échecs en 2016 et très supérieur à n'importe quel champion humain. L'algorithme AlphaZero a gagné contre Stockfish. [8]

La médecine : au dernier meeting annuel (2017) de l'American Society of Clinical Oncology (ASCO) on a évoqué largement de la qualité des conseils de Watson en termes d'aide pour le traitement de cancers. En plus d'obtenir de très bons résultats pour la détection et le traitement de divers cancers, Watson est capable d'analyser les mutations et conseiller un traitement personnalisé pour les patients atteints de cancers du sein et des poumons 78% plus vite qu'un praticien habituel (24 minutes contre une moyenne de 110 minutes). L'intelligence artificielle d'IBM Watson est déjà beaucoup utilisée à travers le monde, notamment dans des centres de soins en Australie, au Mexique, au Brésil et en Asie. IBM a annoncé l'équipement de neuf nouveaux centres médicaux afin d'assister les médecins dans le traitement de

leurs patients. Plus Watson apprend des expériences traitées, plus il pourra analyser efficacement les différents types de cancers.[8]

l'astronomie : le 14 décembre 2017, en cours de recherche de nouvelles planètes, la NASA et Google ont annoncé lors d'une conférence de presse conjointe qu'ils avaient découvert deux nouvelles exo planètes Kepler-80g et Kepler-90i à l'aide de la technologie d'apprentissage automatique : Les astronomes ont étudié 15 000 enregistrements et soumis ces enregistrements au réseau de neurones artificiels. Grâce à eux, l'ordinateur a appris à faire la différence entre les exo planètes, les systèmes à deux étoiles ou les simples erreurs d'observation, avec un taux de réussite de 98,8%. [8]

1.1.4 Les avantages et les inconvénients de l'intelligence artificielle

Examinons maintenant les avantages et les inconvénients de l'intelligence artificielle.

1.1.4.1 Les avantages

- **L'intelligence artificielle trie les données** : Le flux d'actualité Facebook, les algorithmes de prédiction du prix des billets et les recommandations sur Netflix ou Amazon proviennent tous de la puissance de l'IA. [9]
- **L'intelligence artificielle sauve des vies** : dans le domaine médical, elle est de plus en plus utilisée à des fins analytiques. En conséquence, elle est maintenant capable de visualiser et d'analyser avec précision des images médicales pour détecter d'éventuelles tumeurs. Son taux de détection est proche de celui des médecins, et surtout, elle fonctionne en continu 24 heures par jour.
- **L'intelligence artificielle numérise notre écosystème** : nous utilisons l'IA dans presque tous les domaines. L'agriculture devient connectée pour intervenir uniquement au bon moment afin de limiter l'utilisation de produits phytosanitaires, et récolter les céréales, les fruits ou les légumes au meilleur moment.

En termes d'urbanisme, la Smart City se construit aussi grâce à l'intelligence artificielle. Cette dernière collecte de multiples données pour connecter des bâtiments, des rues, des lieux publics, etc.[9]

1.1.4.2 Les inconvénients

- **Un coût élevé** : L'intelligence artificielle est très coûteuse à créer car ce sont des machines très complexes. Ils sont également très coûteux à réparer et à entretenir.[10]
- **Aucune initiative** : Les machines n'ont aucune valeur émotionnelle et morale. Ils exécutent le contenu prédéterminé et ne peuvent juger le bien ou le mal. S'ils rencontrent une situation qu'ils ne connaissent pas, ils ne peuvent même pas prendre de décision. [10]
- **Pas de créativité** : Bien qu'ils puissent vous aider à concevoir et à créer, ils n'ont pas la capacité de réflexion du cerveau humain et n'ont même pas le caractère unique de la pensée créative.[10]
- **Le chômage** : Le remplacement des personnes par des machines entraînera un chômage élevé. [10]

1.2 Les réseaux de neurones

1.2.1 Historique

Les RN sont apparus avec la description d'un modèle de neurone par McCulloch et Petts en 1943, et d'une règle d'apprentissage par Hebb en 1950, la recherche dans le domaine des RN est essentiellement centrée sur les architectures selon lesquelles les neurones sont combinés et les méthodologies par lesquelles les poids d'interconnexions sont calculés ou ajustés. L'idée des RN vient originalement de la modélisation biophysique du cerveau, qui tente d'expliquer la bio-physiologie du cerveau et ses fonctionnalités. Actuellement, les études des circuits neuronaux sont en plein expansion, et le domaine des RN est prêt à prendre la relève.[11]

1.2.2 Le Neurone

1.2.2.1 Modèle d'un neurone biologique

Le cerveau humain est un réseau qui contient des milliards de neurones. Il reçoit, produit des signaux et effectue un certain apprentissage en s'auto organisant d'une manière précise et très compliquée.

Dans le cerveau humain, le neurone est l'élément de base. Il reçoit des signaux en provenance de neurones voisins, les traite, engendre, conduit et transmet l'influx nerveux à d'autres neurones. La figure 1.1 représente les éléments d'un neurone biologique. [12]

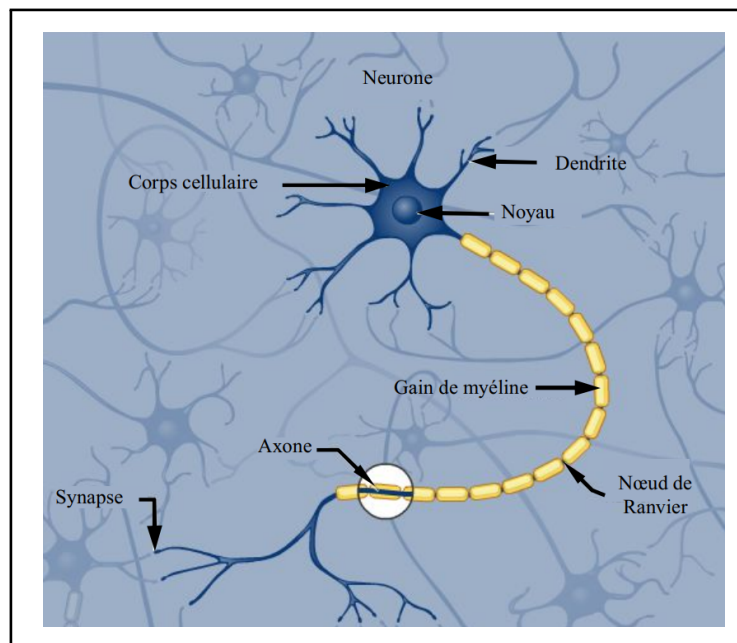


FIGURE 1.1 – Le neurone biologique[12]

Le neurone est constitué de :

- Les dendrites : ce sont les récepteurs principaux du neurone pour capter les signaux qui lui parviennent.

- Le corps cellulaire : Il contient le noyau. C'est un sommateur à seuil. Il effectue une sommation des influx nerveux transmis par ses dendrites. Si la somme est supérieure au seuil, le neurone répond par un influx nerveux ou potentiel d'action qui se propage le long de son axone. Si la somme est inférieure au seuil, il reste inactif.

- L'axone : Il sert de moyen de transport pour les signaux émis par le neurone. Il se ramifie à son extrémité, là où il communique avec d'autres neurones.

- Les synapses : Ils permettent aux cellules nerveuses de communiquer entre elles. Les synapses se rencontrent surtout entre les axones et les dendrites.

1.2.2.2 Modèle d'un neurone artificielle (forme)

Un neurone est une fonction algébrique non linéaire bornée dont la valeur dépend des paramètres appelés coefficients ou poids du neurone . Il reçoit un nombre variable d'entrée en provenance de neurones amont après il s'active suivant la somme pondérer de ses entrées, puis la valeur de cette sommation sera modulée par une fonction qu'on appelle fonction d'activation (Figure 1.2).

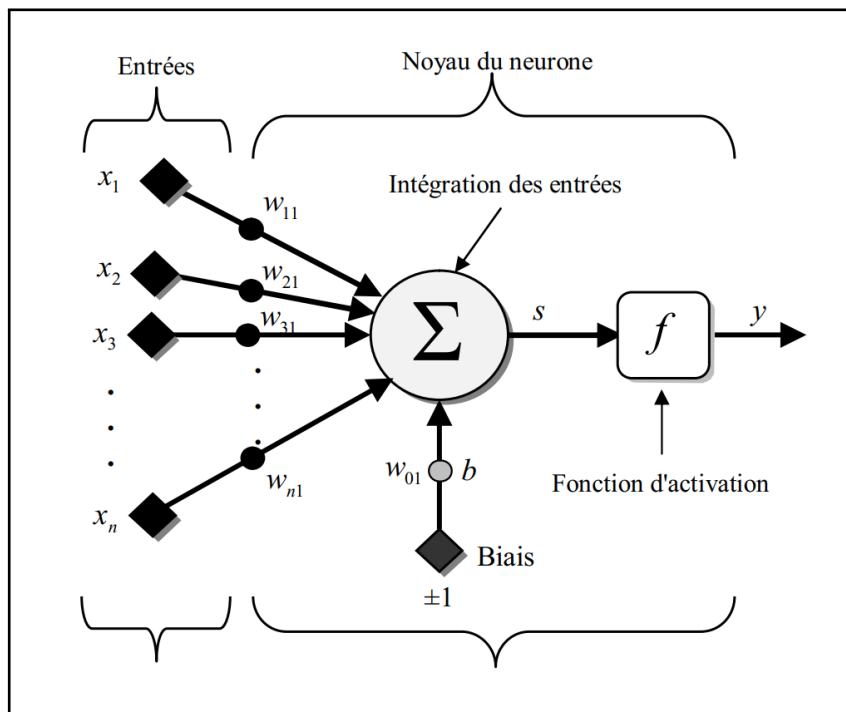


FIGURE 1.2 – Le neurone artificielle[13]

Avec :

- Les x_i = représentent le vecteur d'entrées $[x_0, x_1, x_2, \dots, x_n]$, elles proviennent soit des sorties d'autres neurones, soit des variables d'entrées.

- les w_{ij} sont les poids du neurone j associé a chaque entrée i , ces poids pondèrent les entrées et peuvent être modifiés pendant l'apprentissage.
- Biais : entrée prend souvent les valeurs -1 ou +1 qui permet d'ajouter de la flexibilité au réseau en permettant de varier le seuil de déclenchement du neurone par l'ajustement des poids et du biais lors de l'apprentissage
- Noyau : prend en valeur la somme pondérée avec le biais et calcule la sortie selon une fonction d'activation.

1.2.3 Fonctionnement d'un neurone artificielle

Le neurone calcule la somme des entrées puis cette valeur passe à travers la fonction d'activation pour produire sa sortie . En règle générale, le calcul de la valeur de cette fonction peut se décomposer en deux étape : [14]

- Une combinaison linéaire des entrées : $s = b + \sum_{i=0}^n w_i * x_i$ (figure 1.2)
- La sortie du neurone est : $y = f(s) = f(b + \sum_{i=0}^n w_i * x_i)$ (figure 1.2)

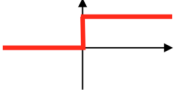
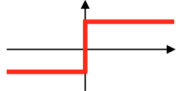
1.2.4 Fonction d'activation

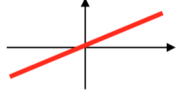
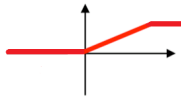
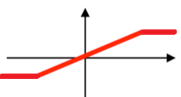
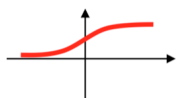
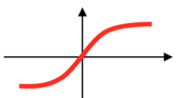
La fonction d'activation , elle est aussi appelé fonction de transfert ou fonction de seuil est utilisée pour introduire la non-linéarité dans le comportements des neurones. La fonction de seuil a généralement trois intervalles :

- en dessous du seuil : le neurone est inactif (Habituellement dans ce cas, sa valeur de sortie est 0 ou -1).
- Près du seuil : la valeur entre dans la phase de transition.
- Au dessus du seuil : le neurone est actif (généralement 1). [15]

(Le tableau 1.1) englobe l'ensemble des fonctions d'activations qui sont généralement utilisées.

TABLE 1.1: Les fonctions d'activations [12]

Nom de la fonction	Type	Equation	Allure
Seuil	Binaire (fonction de heaviside)	$f(x) = 1$ si $x > 0$ $f(x) = 0$ si $x \leq 0$	
	Signe	$f(x) = 1$ si $x > 0$ $f(x) = -1$ si $x \leq 0$	

Linéaire	Identité	$f(x) = x$	
	Saturé positif	$f(x) = 0$ si $x \leq 0$ $f(x) = x$ si $0 < x < 1$ $f(x) = 1$ si $x \geq 1$	
	Saturé symétrique	$f(x) = -1$ si $x \leq -1$ $f(x) = x$ si $-1 < x < 1$ $f(x) = 1$ si $x \geq 1$	
Sigmoide	Positive (type logistique)	$f(x) = \frac{1}{1+e^{-x}}$	
	Symétrique (type tanh)	$f(x) = \frac{2}{1+e^{-2x}} - 1$	

1.2.5 Architecture des réseaux de neurones

Dans un réseau, les neurones sont regroupés généralement en couche. Chaque couche est connectée à tous les neurones dans la couche précédente et la couche suivante. L'information donnée a un réseau de neurone est propagée couche par couche d'entrée a celle de la sortie en passant soit par une ou plusieurs couche intermédiaire. Dépendant de l'algorithme d'apprentissage, il est aussi possible d'avoir une propagation de l'information dans le sens inverse, c'est-a-dire de la couche de sortie a la couche d'entrée, c'est ce qu'on appelle la Rétro-propagation.[16]

1.2.5.1 Les Réseaux de neurones mono couches

La structure d'un réseau monocouche est telle que des neurones organisés en entrée soient entièrement connectés à d'autres neurones organisés en sortie par une couche modifiable de poids.(figure 1.3)

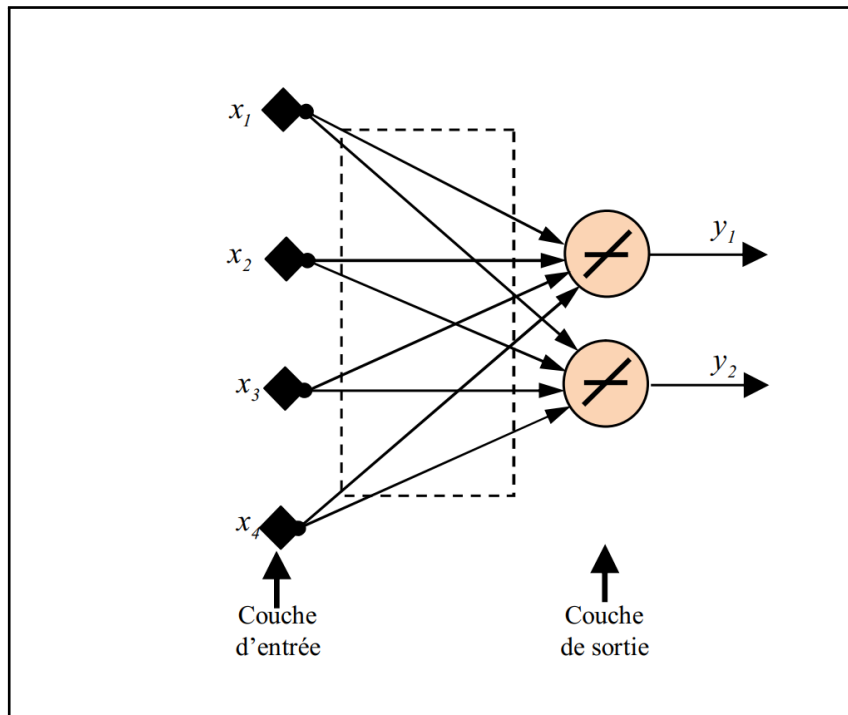


FIGURE 1.3 – Les Réseaux de neurones monocouches.

1.2.5.2 Les Réseaux de neurones multicouches

Les neurones sont arrangés par couche. Il n'y a pas de connexion entre neurones d'une même couche, et les connexions ne se font qu'avec les neurones de couches avales. Habituellement, chaque neurone d'une couche est connecté à tous les neurones de la couche suivante et celle-ci seulement. Ceci nous permet d'introduire la notion de sens de parcours de l'information (de l'activation) au sein d'un réseau et donc définir les concepts de neurone d'entrée, neurone de sortie. Par extension, on appelle couche d'entrée l'ensemble des neurones d'entrée, couche de sortie l'ensemble des neurones de sortie. Les couches intermédiaires sont appelées couches cachées. (figure 1.4)

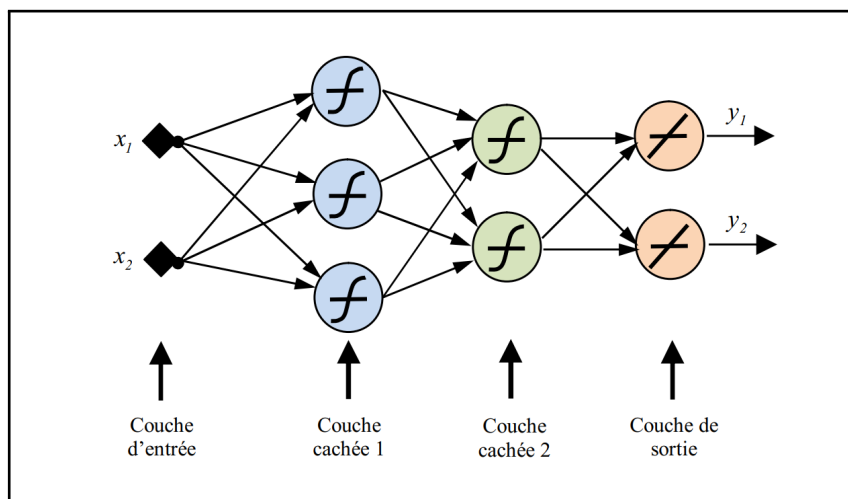


FIGURE 1.4 – Réseaux de neurones multicouches.

1.2.5.3 Les Réseaux de neurones à connexions locales

Il s'agit d'une structure multicouche, mais qui à l'image de la rétine conserve une certaine topologie. Chaque neurone entretient des relations avec un nombre réduit et localisé de neurones de la couche avale. Les connexions sont donc moins nombreuses que dans le cas d'un réseau multicouche classique.(figure 1.5)

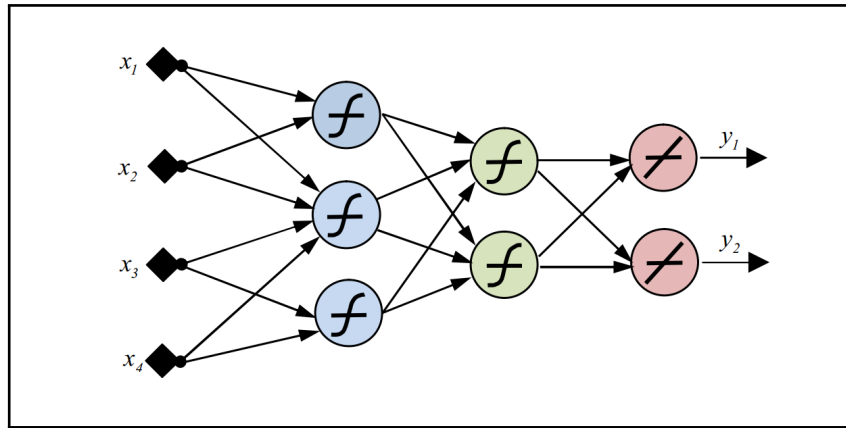


FIGURE 1.5 – Réseaux de neurones à connexions locales.

1.2.6 Caractéristiques des RN

1.2.6.1 Le parallélisme

Il résulte de l'architecture des RN considérés comme des ensembles d'entités élémentaires qui travaillent simultanément.

1.2.6.2 L'apprentissage

On associe généralement aux RN un algorithme d'apprentissage permettant de modifier de manière plus ou moins automatique le traitement effectué afin de réaliser une tâche donnée. L'apprentissage peut être considéré par la mise à jour des poids des connexions au sein des réseaux, afin de réussir la tâche qui lui est demandée. L'apprentissage peut se faire de différentes manières et selon différentes règles :

1.2.6.2.1 L'apprentissage supervisé Dans ce type de d'apprentissage, le réseau s'adapte par comparaison entre le résultat qu'il a calculé, en fonction des entrées fournies, et la réponse attendue en sortie. Ainsi, le réseau va se modifier jusqu'à ce qu'il trouve la bonne sortie.

1.2.6.2.2 L'apprentissage non supervisé L'apprentissage non supervisé implique la fourniture à un réseau autonome d'une quantité suffisante d'exemples contenant des répétitions (autrement dit de la redondance), telles que celui-ci en dégage les

régularités automatiquement. Ces réseaux sont souvent appelés auto-organiseurs, ou encore à apprentissage compétitif. Dans l'apprentissage non supervisé, les données ne contiennent pas d'informations sur une sortie désirée, il n'y a pas de superviseur ou expert humain. Il s'agit de déterminer les paramètres du réseau de neurones suivant un critère à définir. Dans ce cas, les exemples présentés à l'entrée provoquent une auto adaptation du réseau afin de produire des valeurs de sortie qui soient proches en réponse pour des valeurs d'entrées similaires. L'architecture du réseau, préalablement définie par son utilisateur, est une forme de supervision. [17]

1.2.6.2.3 Le renforcement Le renforcement est en fait une sorte d'apprentissage supervisé et certains auteurs le classe d'ailleurs, dans la catégorie des modes supervisés. Dans cette approche le réseau doit apprendre la corrélation entrée/sortie via une estimation de son erreur, c-à-d du rapport échec/succès. Le réseau va donc tendre à maximiser un index de performance qui lui est fourni, appelé signal de renforcement. Le système étant capable ici, de savoir si la réponse qu'il fournit est correcte ou non, mais il ne connaît pas la bonne réponse.

1.2.6.2.4 Le mode hybride Le mode hybride reprend les deux approches supervisée et non supervisée, puisque une partie des poids va être déterminée par apprentissage supervisé et l'autre partie par apprentissage non supervisé.

1.2.7 Les avantages et les inconvénients de réseaux de neurone

1.2.7.1 Les avantages

- Faculté d'apprentissage à partir d'exemples représentatifs, par rétro-propagation des erreurs. L'apprentissage (ou construction du modèle) est automatique.
- Simple à manier, beaucoup moins de travail personnel à fournir que dans l'analyse statistique classique.
- Aucune compétence en mathématique, informatique statistique requise.

1.2.7.2 Les inconvénients

- L'absence de méthode systématique permettant de définir la meilleure topologie du réseau et le nombre de neurones à placer dans la (ou les) couche(s) cachée(s) .
- Le choix des valeurs initiales des poids du réseau .

- Le problème du sur-apprentissage (apprentissage au détriment de la généralisation).
- Les réseaux de neurones sont des boîtes noires que les utilisateurs ne peuvent pas comprendre.

Conclusion

Dans ce chapitre nous avons introduit quelques notions de base sur l'intelligence artificielle ainsi que ces domaine d'utilisation puis nous avons présenté le fonctionnement, les différentes architectures ainsi que les caractéristiques principales des réseaux de neurones artificiels qui seront utilisés dans notre travail afin de traiter les données des capteurs.

l'objet de notre étude dans le chapitre suivant est de faire l'état de l'art sur les systèmes experts .

Chapitre 2

État de l'art sur les systèmes experts

Introduction

Les systèmes expert sont conçus pour travailler dans un domaine précis, ils suivent les comportement et les méthode d'un expert du domaine afin d'aboutir a des résultats et des conclusions précise. Les systèmes experts appartiennent a l'intelligence artificielle ,leur comportement ne consiste pas simplement a exécuter un algorithme simple mais ils apprennent au fur et a mesure, c'est ce qu'il les caractérise.

2.1 Définition d'un système expert

Selon la définition proposée par J.C. Pomerol, un système expert est un outil informatique d'intelligence artificielle est conçue pour simuler les connaissances d'experts dans un domaine précis et bien défini, grâce au développement d'une certaine quantité de connaissances clairement fournies par des experts du domaine. Il permet de modéliser le raisonnement des experts, de manipuler les connaissances , de promouvoir l'acquisition, la modification et la mise à jour des connaissances et d'expliquer comment les résultats sont obtenus. [18]

2.2 Structure d'un système expert

La structure générale d'un système expert est constituée de trois modules comme le montre la figure ci-dessous :

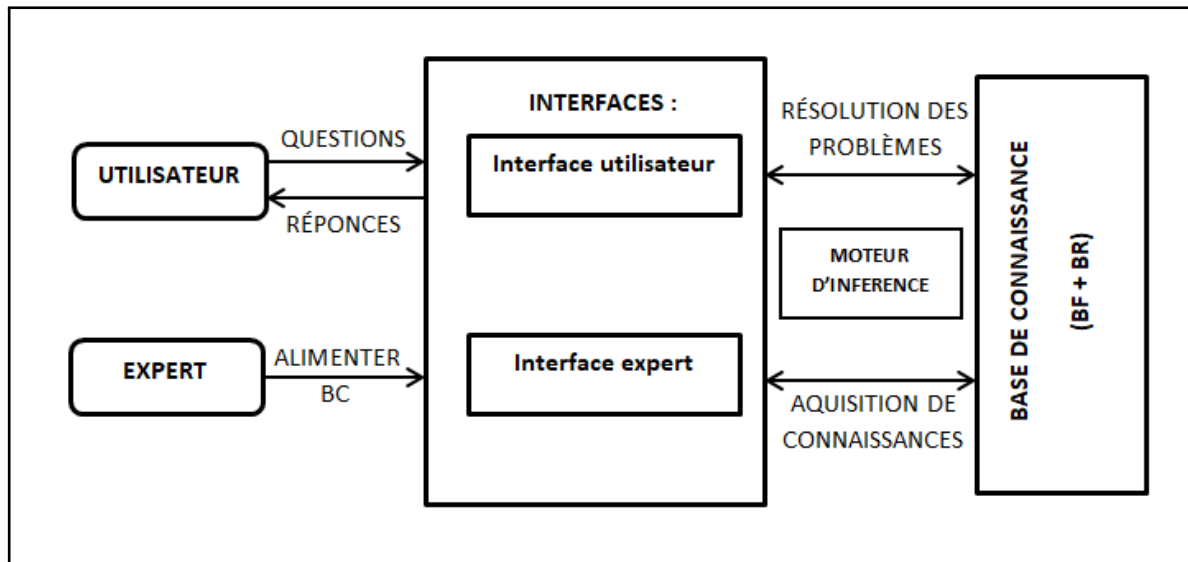


FIGURE 2.1 – Architecture d'un Système Expert [19]

- **L'interface expert** : sert à introduire les règles et approvisionner la base de faits. [19]
- **L'interface utilisateur** : sert à simplifier la communication, elle peut utiliser la forme question-réponse, le menu, le langage naturel etc. [19]
- **La base de connaissances (BC)** : contient les connaissances concernant la résolution du problème et dispose des deux bases suivantes :

-La Base de Faits (BF) : C'est l'une des entrées du moteur d'inférence elle contient des connaissances qui représentent un état prouvable, elle est la mémoire de travail de SE , elle peut être variable pendant l'exécution et sera vide une fois l'exécution terminée Les faits peuvent prendre des formes plus ou moins complexes.[19]

-La Base de Règles (BR) : contient les connaissances et les raisonnements effectués par l'expert, ces règles sont appelées les une a la suite des autre créant ainsi un enchaînement de raisonnements généralement sous la forme de « si condition alors action ». [19].

2.2.1 Moteur d'inférence

Un moteur d'inférence fait des raisonnements logiques en utilisant la base de faits et la base de règles. Le moteur d'inférence utilise des règles, les interprète, les enchaîne jusqu'à arriver à un état représentant une condition d'arrêt suivant différentes méthodes de chaînage. [20]

2.2.1.1 Chaînage avant

En mode Chaînage avant, le moteur d'inférence part des fait pour atteindre l'objectif, c'est-à-dire qu'il ne sélectionne que des règles dont les conditions de la partie gauche sont vérifiées, puis applique la première de ces règles, en ajoutant la conclusion à la base. Cette procédure est répéter jusqu'à atteindre l'objectif ou qu'il n'y ait plus de règle a appliquer.

Voici l'algorithme du chaînage avant :

Algorithme du chainage avant

BF (base de faits), BR (base de règles), F (proposition à vérifier)

DEBUT

TANT QUE F n'est pas dans BF **ET QU'**il existe dans BR une règle applicable

FAIRE

Prendre la première règle applicable R

BR = BR - R (désactivation de R)

BF = BF union conclusion(R)

(déclenchement de la règle R, sa Conclusion est rajoutée à la base de faits)

FIN TANT QUE SI F appartient à BF **ALORS**

F est établi (succès)

SINON F n'est pas établi (échec)

FIN . [21]

Exemple :

Soit la base de faits $\{\mathbf{B}, \mathbf{C}\}$ et l'objectif \mathbf{H} et la base de règles :

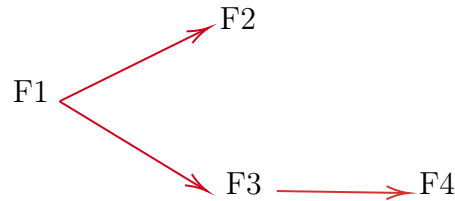
1. Si $\mathbf{B}, \mathbf{D}, \mathbf{E} \rightarrow \mathbf{F}$
2. Si $\mathbf{G}, \mathbf{D} \rightarrow \mathbf{A}$
3. Si $\mathbf{C}, \mathbf{F} \rightarrow \mathbf{A}$
4. Si $\mathbf{B} \rightarrow \mathbf{X}$
5. Si $\mathbf{D} \rightarrow \mathbf{E}$
6. Si $\mathbf{X}, \mathbf{A} \rightarrow \mathbf{H}$
7. Si $\mathbf{C} \rightarrow \mathbf{D}$
8. Si $\mathbf{X}, \mathbf{B} \rightarrow \mathbf{A}$

Execution :

1. Evaluation de la règle 4 \rightarrow BF = $\{\mathbf{B}, \mathbf{C}, \mathbf{X}\}$
2. Evaluation de la règle 7 \rightarrow BF = $\{\mathbf{B}, \mathbf{C}, \mathbf{X}, \mathbf{D}\}$
3. Evaluation de la règle 8 \rightarrow BF = $\{\mathbf{B}, \mathbf{C}, \mathbf{X}, \mathbf{D}, \mathbf{A}\}$
4. Evaluation de la règle 6 \rightarrow BF = $\{\mathbf{B}, \mathbf{C}, \mathbf{X}, \mathbf{D}, \mathbf{A}, \mathbf{H}\}$
5. \mathbf{H} est trouvé \rightarrow Fin avec réussite.

2.2.1.2 Chaînage arrière

Le chaînage arrière ou raisonnement arrière est un mécanisme d'inférence qui fonctionne à l'envers c'est-à-dire il travaille en remontant en arrière de l'objectif. Si B est faux et si A implique B alors A est faux. Le chaînage arrière traduit un raisonnement déductif :



- de F4 est déduit F3.
- de F3 est déduit F1. [21]

Algorithme du chaînage arrière

Fonction chaînage Arrière

BF(base de faits),BR(base de règles),listeDesObjectifs.

SI Vide(listeDesObjectifs) **ALORS**

Résultat \leftarrow SUCCES

SINON

SI demBut (premier (listeDesObjectifs)) **ALORS**

Résultat \leftarrow chaînage Arriere (suite(listeDesObjectifs))

SINON

Résultat \leftarrow ECHEC

FIN SI

FIN SI Retourner Résultat.

Exemple : Soit la base de faits $\{A,M\}$ et l'objectif C .

Soit la base de règles suivante :

1. SI $E,F \rightarrow C$
2. SI $M,F \rightarrow E$
3. SI $H,A \rightarrow F$
4. SI $F,H \rightarrow M$
5. SI $A,M \rightarrow H$

Execution :

1. Evaluation de la règle 1 \rightarrow listeDesObjectifs $\{F,E\}$
2. Evaluation de la règle 2 \rightarrow listeDesObjectifs $\{M,F\}$
3. Evaluation de la règle 3 \rightarrow listeDesObjectifs $\{M,A,H\}$
4. Evaluation de la règle 4 \rightarrow listeDesObjectifs $\{M,A\}$
5. $A,M \in BF$ donc elle sont vérifier \rightarrow listeDesObjectifs $\{\}$ SUCCÈS.

2.2.1.3 Chaînage Mixte

Le chaînage mixte est une combinaison du chaînage avant et arrière qui améliore ce dernier car le chaînage arrière pose le problème du retard à l'évaluation. En effet Lorsque un but est évalué, l'évaluation peut conduire à des conclusions sur d'autres attributs, qui ne sont pas faites et pour lesquelles d'autres procédures de chaînage arrière sont inutilement invoquées. En chaînage mixte après chaque évaluation d'une prémisse de règle, une propagation en avant de cette évaluation est faite.[21]

2.3 Domaine d'application des systèmes experts

De nos jours, les systèmes experts se retrouvent utilisés dans divers domaines d'applications tels que :

- **La médecine** : aide au diagnostic, aide à l'interprétation d'image, surveillance des malades en réanimation.
- **L'informatique** : aide à la programmation, configuration de système informatique, aide à l'exploitation et à la maintenance.
- **Banque, finance , assurance** : gestion financière, aide au placement, évaluation des risques de prêts.
- **Aéronautique et espace** : aide au pilotage, planification du trafic aérien, contrôle de satellites(correction d'attitude, contrôle des opérations de lancement).
- **Militaire** : identification et suivi de cibles, aide au pilotage d'engins (chars, hélicoptères, avions de chasse).

2.3.1 Quelques systèmes experts classique

MYCIN, 1974 : médecine, système d'aide au diagnostic et au traitement de maladies bactériennes du sang.

MOLGEN, 1977 : biologie, engendre un plan de manipulations génétiques en vue de construire une entité biologique donnée.

CRYBALIS, 1979 : chimie, recherche la structure de protéines à partir de résultats d'analyse cristallographique.

MUSCADET, 1984 : mathématiques, démonstration de théorèmes. [22].

2.3.2 Représentation des connaissances

c'est un ensemble d'outils et de méthodes destinés d'une part à représenter et d'autre part à organiser le savoir humain pour l'utiliser et le partager.

- la logique avec le calcul des prédicats d'ordre 0 ou 1 ou intermédiaire (0+).
- les règles (déduction, réécritures, actions conditionnelles, ...).
- le formalisme objet (objets, classes, instances, héritage). [23]

Pour représenter les connaissances nous avons les différents modèles suivants :

- **Les réseaux sémantiques.**
- **Les représentations logiques.**
- **Les règles de production.**
- **Les objets structurés.**

Le développement du Web, en particulier le développement du Web sémantique, a mis à jour son domaine en introduisant l'ontologie . Un certain nombre de langages ont été développés , comme les standards RDFS SKOS et OWL .

2.3.2.1 Les réseaux sémantiques

les réseaux sémantiques sont des graphes (schémas) qui servent à représenter les connaissances. Dans ces réseaux sémantiques les concepts du domaine à modéliser sont représentés par des noeuds et la relation entre ces concepts sont représentés par des arcs étiquetés. [24]

l'un des concepts qu'un réseau sémantique peut modéliser le mieux est l'hiérarchisation des domaines où les arcs représentent : "sous-ensemble de" ou "est un élément de" comme le montre la figure :

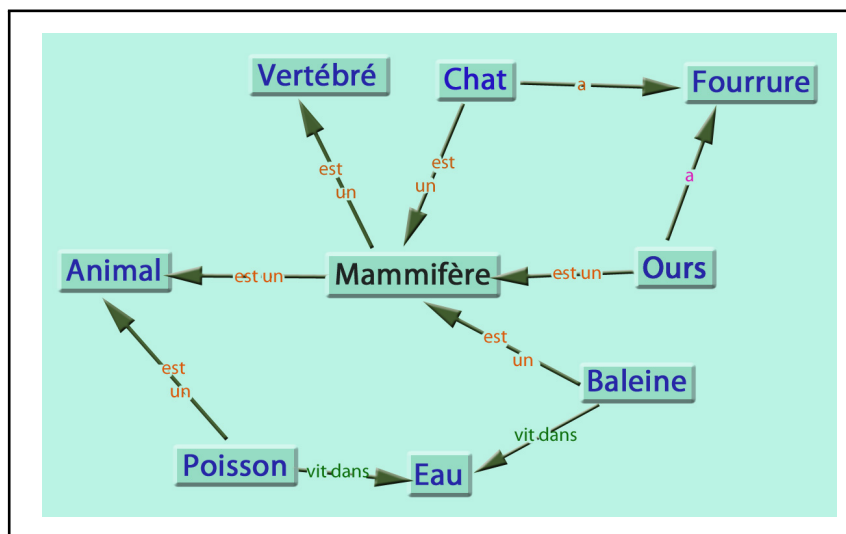


FIGURE 2.2 – Représentation d'un réseau sémantique [25]

2.3.2.2 Les représentations logiques

- **Logique d'ordre 0 :**

c'est la logique des proposition, elle est considéré comme la plus simples des logiques. Les formules sont appelées des propositions. Ces derniers, ne peuvent prendre que deux valeurs, vrai ou faux. Les formules sont construites à l'aide un ensemble fini ou dénombrable $P = p_0, p_1, \dots, p_n$ de symboles que l'on appelle variables propositionnelles.[26]

Exemple : Aucun homme ne marche.

$$\forall X \text{homme}(X) \Rightarrow \neg \text{marche}(X).$$

- **Logique d'ordre 0+ :**

Les prédicats d'ordre supérieur sont des prédicats qui prennent un ou plusieurs autres prédicats comme paramètres. Généralement, un prédicat d'ordre n prend comme paramètre un ou plusieurs prédicats d'ordre n-1, avec $n > 1$. La même chose pour les fonctions .[26]

Exemple :

$$\underbrace{\text{Si heure} = 09}_{\text{Prémisse}} \quad \underbrace{\text{Alors temps} = \text{«Jour»}}_{\text{Conclusion}}$$

- **Logique d'ordre 1 :**

C'est la logique du calcul des prédicats. Un prédicat : est une expression logique dont la valeur peut être vraie ou fausse selon la valeur des arguments, c'est-à-dire que la fonction renvoie "vrai" ou "faux". La logique d'ordre 1 comporte :

- Des constantes (a, b ,c ...).
- Des variables (x, y ,z ...).
- Quantificateurs logique (\exists, \forall).
- Connecteurs ($\wedge, \vee, \rightarrow, \leftrightarrow$).
- Des fonctions (f, g ,h ...). [27]

Exemples :

- $\text{homme}(x) \Rightarrow x \text{ est un homme.}$
- $\forall(x) \text{voiture}(x) \Rightarrow \text{avoir_roues}(x).$

2.3.2.3 Les règles de production

Une règle de production se compose d'une partie gauche appelée la «prémisse» et d'une partie droite appelée la « conclusion ». Sa forme est la suivante : Si condition(s)

alors action(s) Une règle permet de représenter des connaissances utiles pour une analyse, un diagnostic, une classification, etc.

Le langage de description des règles peut comporter des conjonctions, des disjonctions, des négations ...[28]

Exemples :

(R1) SI l'animal a des poils, ALORS c'est un mammifère.

(R2) SI l'animal est un mammifère, et s'il mange de la viande, ALORS c'est un carnivore.

(R3) SI X possède un micro-ordinateur et X maîtrise php ALORS il pourrait créer des applications web.

2.3.2.4 Les objets

Un objet est une entité rassemblant des données et du code travaillant sur ses données là. Une classe décrit la structure interne d'un objet ,elle est considérée comme un moule à partir duquel on peut instancier des objets. Considérons par exemple la modélisation d'un véhicule telle que présentée par la figure suivante :

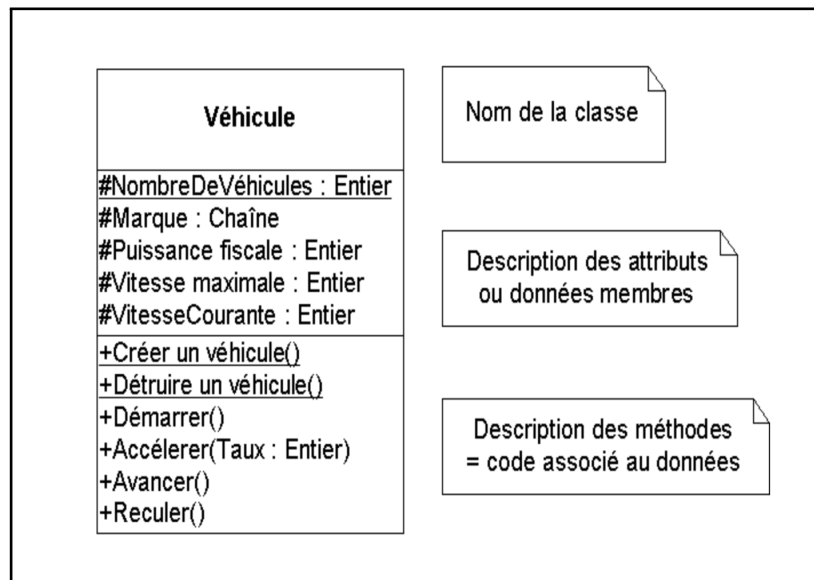


FIGURE 2.3 – Représentation de la classe Véhicule [29]

Les principales caractéristiques dans la structure d'objet sont :

● **Encapsulation :**

L'encapsulation est un mécanisme consistant à protéger les données et les méthodes contenues dans un objet. Cette protection consiste à empêcher la modi-

fication de ces données depuis l'extérieur. On peut accéder à ces informations uniquement par les services proposés.[29]

- **Héritage :**

l'héritage ou dérivation de classe permet de créer une nouvelle classe à partir d'une classe existante. La classe dérivée contient tout les attribues et les méthodes de sa classe mère (superclass) plus ceux qui lui sont propre. [29]

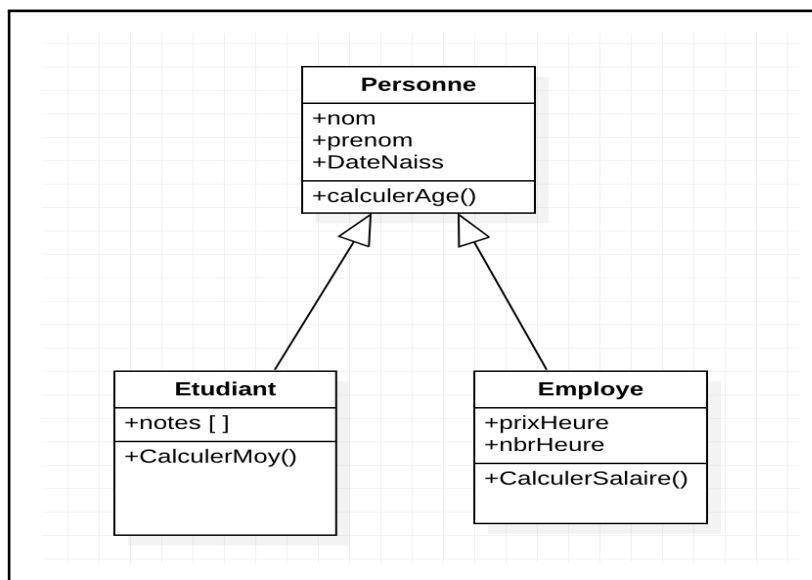


FIGURE 2.4 – Exemple d'héritage

- **Polymorphisme :**

Le polymorphisme en informatique est l'idée d'autoriser le même code à être utilisé avec différents types. Le polymorphisme peut être vu comme la capacité de choisir dynamiquement la méthode qui correspond au type réel de l'objet.[29]

2.4 Le cycle de base d'un moteur d'inférence

Le moteur d'inférence est le cerveau d'un système expert. Il exploite la base des connaissances afin de résoudre les problèmes posés par les utilisateurs.

Au cours de son raisonnement, le moteur d'inférence enchaîne des cycles de travail comportant chacun deux phases : une phase **d'évaluation** et une phase **d'exécution**.[30]

2.4.1 phase d'évaluation

Cette phase d'évaluation s'effectue généralement en deux étapes :

2.4.1.1 Le filtrage

Cette étape consiste à définir l'ensemble des règles de BC, les règles qui sont effectivement applicables, celles dont le côté gauche (prémisses) est satisfait au vu de l'état actuel de la base de faits, cet ensemble de règles est appelé ensemble de conflit.[30]

2.4.1.2 La sélection avec résolution de conflits

La résolution de conflit ou choix de règle consiste à choisir la règle qui va être appliquée et utilisée.[30]

2.4.2 La Phase d'exécution

Cette phase comme son nom l'indique, concerne l'exécution de la partie « action » de la règle sélectionnée lors de la phase d'évaluation et à modifier la base de faits.[30]

Dans la figure ci-dessus (figure 2.5), il est illustré comment, en général, un moteur d'inférence enchaîne des cycles de travail composés de ces deux phases :

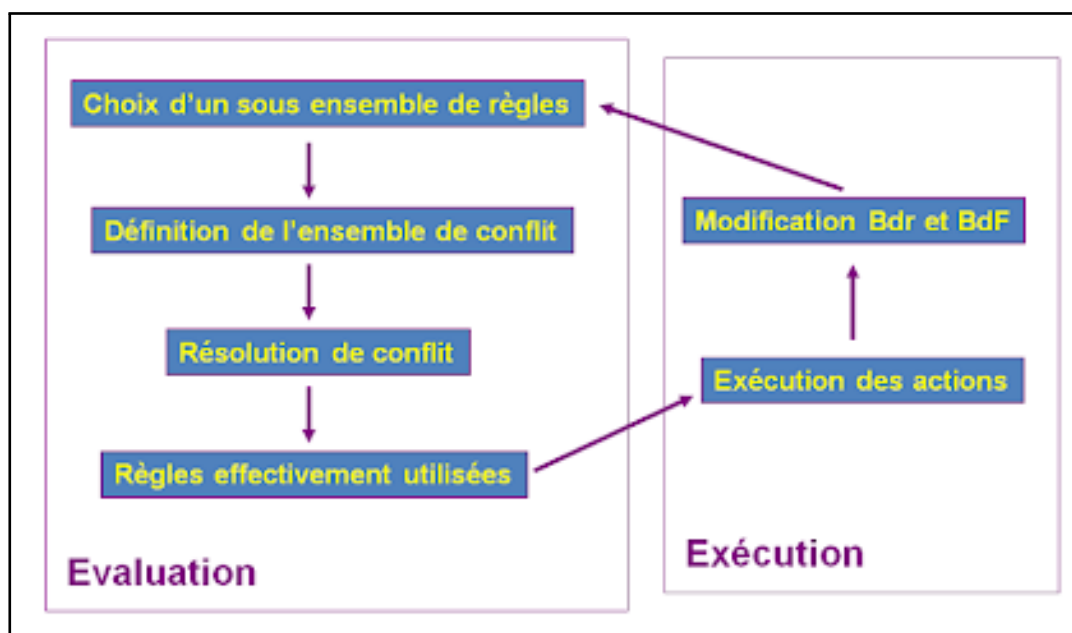


FIGURE 2.5 – Cycle de base d'un moteur d'inférence [31]

2.5 Stratégie de recherche

Au cours des cycles de recherche d'un moteur d'inférence un arbre de recherche ce crée, ou chaque niveau correspond à l'ensemble des règles applicables. Chaque règle déclenchée crée une nouvelle situation et de nouvelles règles à invoquer.[30] Deux principales stratégies de recherche se présentent :

2.5.1 La recherche en largeur

on applique toutes les règles d'un même niveau les une après les autres avant de passer au niveau suivant.

2.5.2 La recherche en profondeur

on applique une règle d'un niveau à un autre et on ne revient aux règles restantes jusqu'à arriver a une condition d'arrêt (succès de la branche ou échec).

2.6 Réalisation d'un SE

La réalisation d'un système expert se fait en deux phases :

2.6.1 Phase de conception

La phase de conception d'un SE est indispensable, dans cette phase on définit tout les éléments de l'application : Les exigences ainsi que le type d'entrées et de sorties nécessaire, quel type de problèmes peut rencontrer l'utilisateur, comment exprime-il ses besoins à l'ordinateur et quel réponse souhaite-il obtenir. Nous pouvons donner ainsi un plan d'actions, sous la forme d'une série d'étapes.[32]

La phase de conception SE peut alors être définie en suivant ces étapes :

1. Contacter un ou plusieurs experts dans le domaine.
2. Établir la liste des données à déduire et toutes les règles de déduction.
3. Si le calcul des règles ne fournit pas de résultats sûrs, associer à chaque règle un degré de certitude.
4. Préparation de la base de connaissances : choix des prédicats et représentation de l'ensemble des règles.
5. Définition de l'interface utilisateur :
 - Moyen de communication (entrées/sorties en langage naturel).
 - La représentation des écrans et boites de dialogue.

2.6.2 Phase d'implantation

Pour développer un SE, il est indispensable d'utiliser un système informatique qui offre un moyen de représentation de la connaissance (BC) et un module pour faire les raisonnements logique (MI).[32]

Deux choix se présentent :

1. Soit en utilise des langages de programmation spécifique au système expert qui dispose déjà d'un moteur d'inférence par exemple : les langages de programmation logique (Prolog,Oz, CLIPS,LISP).
2. Soit en utilise les langages de programmation dit (general-purpose programming language) comme (JAVA,C,PYTHON), l'avantage de ce choix est de proposer un moteur d'inférence suivant les exigences de l'application à réaliser.

2.7 les avantages et les inconvénients d'un système expert

2.7.1 Les avantages d'un système expert

Les systèmes experts présentent plusieurs caractéristiques intéressantes :

- Le coût de mise à disposition de l'expérience pour l'utilisateur est considérablement réduit.
- L'expérience est permanente. Contrairement aux experts humains, la connaissance d'un système expert est préservée indéfiniment.
- Les connaissances de nombreux spécialistes peuvent être disponibles pour un travail simultané et continu sur le problème, L'expérience combinée de nombreux systèmes experts peut dépasser celle d'un seul spécialiste humain.
- Les systèmes experts réagissent plus rapidement et sont plus disposés que les experts humains.
- Facilité de mise à jour.[33]

2.7.2 Les inconvénients d'un système expert

Les principaux inconvénients des systèmes experts sont :

- Ils créent le chômage parce qu'ils émulent les humains.
- Dans les systèmes experts, on fait inférence à des connaissances même si elles sont dépassées.
- Coûts et Difficulté de développement
- Mise à jour continuels requis
- L'expert a besoin toujours un cognicien.[33]

Conclusion

au cour de ce chapitre nous venons de présenter les système expert ainsi que leur mode de fonctionnement en passant par les modules qui les composent et les différents stratégies de recherche et de chaînage qu'ils adoptent.

Dans le chapitre suivant nous allons parler de l'application du réseau de neurones pour la classification des données de capteurs.

Chapitre 3

Application du réseau de neurone pour la classification des données de capteurs

Introduction

Ce chapitre présente l'application du réseau neuronal supervisé comme étant une solution dans la classification des données des capteurs (température, humidité et d'eau) en trois classes (traitement, traitement avec anomalies et pas de traitement).

Le but est de valider et d'évaluer les performances de chaque modèles présentés, les exigences principales d'efficacité sont formulées sur deux points de base à savoir, les tests qui vérifient que l'algorithme réalise bien la tâche pour laquelle il a été conçu, et les tests de performance qui seront utilisés pour mesurer l'efficacité de l'exécution de cette tâche. Afin de mener une étude comparative et de sélectionner de manière décisive le modèle le plus adapté à l'application indiquée, des paramètres liés au taux de reconnaissance et au temps d'apprentissage seront évalués.

Une discussion des résultats conclura cette étude afin de sélectionner le modèle le plus appropriée.

3.1 Présentation des capteurs

3.1.1 Le capteur de température et humidité

Le MS6503 série RS232 de la société Mastech est un capteur de température et humidité, il est utilisé dans les salles de traitement de radiothérapie au niveau du CPMC d'Alger. Les caractéristiques de ce capteur sont présentées dans l'annexe.

3.1.2 Le capteur de pression d'eau

Le capteur PT205 de la marque Atech, il est utilisé pour mesurer la pression d'eau dans les machines de traitement de radiothérapie au niveau du CPMC d'Alger.

3.2 Pourquoi le réseau de neurone ?

Il existe plusieurs algorithmes de classification supervisée qui reposent sur différentes techniques, telles que : Machine à vecteur de support (SVM), arbre de décision (ADD), classificateur bayésien, les k plus proche voisin (KNN), réseau de neurones (RN) etc...

Nous nous intéressons dans ce travail aux réseaux de neurones qui sont des machines parallèles, ils allient fiabilité et tolérance aux fautes. Lorsque ces réseaux sont organisés en couches, ils s'avèrent être des classificateurs très efficaces, qualité qui leur est donnée par les propriétés non linéaires des neurones formels, les algorithmes d'apprentissage permettent d'adapter ces réseaux à toutes sortes de problèmes de classification. Ainsi, la rétro-propagation du gradient de l'erreur permet l'ajustement de surfaces de décisions entre les classes de n'importe quelle catégorie de problème.

3.3 Procédure de construction d'un réseau de neurones

La procédure de conception d'un réseau de neurone peut être résumée en cinq étapes :

- La collecte d'une base de données.
- La séparation de la base de données en deux sous-ensembles (base d'apprentissage, base de validation et test).
- Prétraitement des données.
- Entraînement du réseau de neurones sur les bases d'apprentissage et de validation.
- Mesure des performances du réseau de neurone sur la base de test.

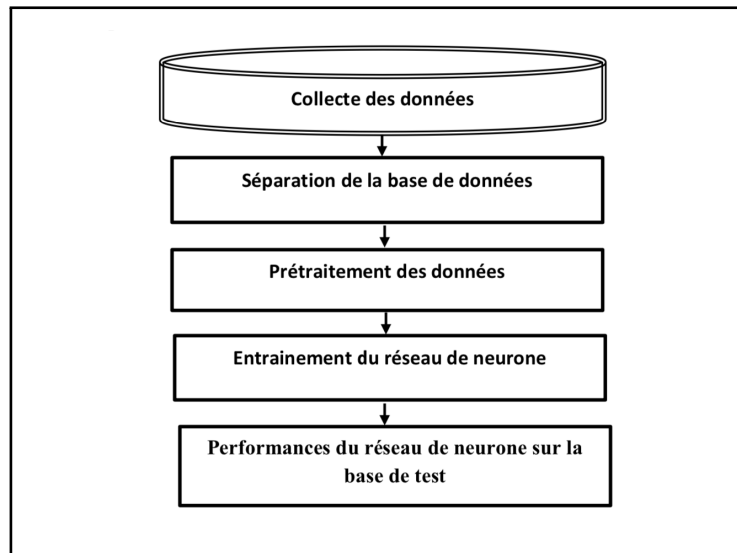


FIGURE 3.1 – Flux de base pour la conception d’un modèle de réseau neuronal artificiel.

3.3.1 Collecte d’une base de données

L’objectif de cette étape est de collecter un nombre de données plus suffisant pour construire une base représentative de données, qui servira à l’apprentissage et au test du réseau de neurones.

Cette base de données constitué l’entrée du réseau de neurone, et par conséquent c’est elle qui détermine à la fois la taille du réseau, et les performances du système.

Pour notre application, nous avons utilisé les données de mois de Juin 2020 comptant 838 valeurs fournie par les capteurs placés dans les salles de CPMC.

Nous avons travaillé avec les valeurs des capteurs pour prévoir la possibilité de traitement. Nous avons utilisé comme inputs : température, humidité et la pression d’eau, les unités et les intervalles utilisés sont représentés dans le tableau 3.1 ci-dessous. Pour l’output : **0** (traitement), **1** (traitement avec anomalies) et **2** (pas de traitement). Le tableau 3.2 représente l’ensemble de données des capteur avec leurs classes.

TABLE 3.1: Paramètres utilisés .

Variable	Unité	Intervalle	Type
Température	°C	0°C – 40°C	Numérique
Humidité	RH	20 – 80%RH	
Pression d’eau	PSI	10 – 100PSI	

TABLE 3.2: Les données des capteurs.

N°	Temperature	Humidité	Pression	Classe
1	26.1	38.9	45	0
2	29.8	62.0	40	1
3	19.8	38.5	62	0
4	16.7	68.0	19	2
.....				
.....				
838	22.5	57.4	47	0

3.3.2 Séparation de la base de données

Après la collecte des données suffisantes, il faut procéder à la phase de séparation de ces données en deux ensembles. Une partie pour effectuer l'apprentissage et l'autre pour la validation qui permet de contrôler la phase d'apprentissage et pour tester le réseau obtenu et déterminer ses performances.

En général, la plupart des données sont utilisées pour l'apprentissage et une petite partie des données est utilisée pour la validation et le test. Dans notre cas nous avons utilisé 80% (669 valeurs) pour l'apprentissage et 20% (169 valeurs) pour la validation et le test.

3.3.3 Choix de l'architecte de réseau de neurone

La définition de l'architecture de réseau de neurone utilisée pour résoudre un problème donné est encore un problème ouvert. Ce problème peut généralement être résolu de deux méthodes : la première consiste à ajouter successivement des neurones et des connexions à une petite architecture, la deuxième quant à elle consiste à supprimer des neurones et des connexions d'une architecture initiale maximale. Ces deux méthodes ont souvent comme inconvénient le temps d'apprentissage élevé et imprévisible.[34]

L'architecture d'un réseau de neurone est l'organisation des neurones dans un même réseau. En d'autres termes, il s'agit de la façon dont ils sont connectés et ordonnés.

Pour pouvoir utiliser le réseau de neurone, on doit avoir une méthode indiquant comment choisir une architecture de réseau. C'est-t-à-dire, pouvoir répondre aux questions suivantes : combien de couches cachées ? combien de neurones par couches cachées ? quelle fonction d'activation utiliser ?

La structure du réseau dépend étroitement de la base de données. Constituée de couples entrée/ sortie (s), Pour la couche d'entrée le nombre de neurones est déterminé par le nombre de variables utilisées en entrée. Dans notre cas 3 neurones. Pour la couche de sortie, le nombre de neurones est déterminé par le nombre de sorties à approximer, (la variable qu'on veut prédire) dans notre travail on a soit la possibilité de traitement normal, traitement avec anomalies ou bien pas de traitement, donc on a utilisé trois neurones dans la couche de sortie.

3.4 Résultat et discussion

Les réseaux RNA sont testés et validés avec un macbook pro en utilisant notre algorithme d'apprentissage. On a utilisé le langage python et la bibliothèque Tensorflow. Voici le pseudo-code :

```
1 TRAIN_DATAS = LIRE_FICHER ("TRAIN.CSV");
2 TEST_DATAS  = LIRE_FICHER ("TEST.CSV");
3
4 MODEL = CREER_MODEL (
5     INPUT_LAYER      = 3,
6     (HIDEN_LAYER_1  = 15, FONCTION_ACTIVATION="SIGMOID"),
7     (HIDEN_LAYER_2  = 15, FONCTION_ACTIVATION="SIGMOID"),
8     (OUTPUT_LAYER   = 3  , FONCTION_ACTIVATION="SOFTMAX"),
9 );
10
11 COMPILER_MODEL (MODEL , OPTIMISER="ADAM");
12 (ACCURACY_TRAIN , LOSS_TRAIN) = ENTRAINEUR_MODEL (MODEL ,
13     TRAIN_DATAS);
14 (ACCURACY_TEST , LOSS_TEST) = EVALUER_MODEL (MODEL ,
15     TEST_DATAS);
16 MATRICE_CONFUSION = MC (MODEL , TEST_DATAS);
17 SAUVGARDER_MODEL (MODEL , "model.h5");
```

Listing 3.1 – Pseudo-code algorithme d'apprentissage

Pour les couches cachées 1 et 2 nous avons utilisé comme fonction d'activation la fonction **SIGMOID**, et pour la couche de sortie la fonction d'activation **SOFTMAX** qui permet de choisir le neurone qui a comme valeur la plus grande probabilité et l'active en mettant sa valeur à 1, quant aux autres neurones ils seront désactivés en mettant leurs valeurs à 0.

Le code python de l'apprentissage automatique est présenté dans l'annexe.

3.4.1 Les modèles utilisés

Nous avons adopté une démarche qui consiste à considérer d'abord l'architecture la plus simple (avec un minimum de neurones dans la couche cachée). Puis à la rendre plus complexe et à retenir celle qui présente les meilleures performances. Pour la démonstration on a sélectionné 3 modèles.

Les résultats de l'apprentissage et de test dans la base de données de dimension 3 sont présentés dans le tableau 3.3. Les paramètres tels que les différents nombres de neurones dans les couches cachées (NNCC), nombre d'itération, le temps d'apprentissage et le taux de précision pour les trois modèles pendant la phase d'entraînement et test.

TABLE 3.3 – Les paramètres de chaque modèle

base de données	Réseau	NNCC	Nombres d'itérations	Temps d'apprentissage (sec)	Taux de précision (entraînement)	Taux de précision (test)
3 Entrées (C, Rh,PSI)	1	5 - 5	500	16,26	0.8475 %	0.8107 %
	2	10 - 10	500	16,43	0.9537 %	0.9476%
	3	15 - 15	500	16,58	0.9704%	0.9704%

3.4.1.1 Résultat modèle 1 (5x5) :

Dans ce modèle on a utilisé un réseau qui se compose de deux couches cachées chaque couche contient 5 neurones, de 3 entrées et 3 neurones de sortie.

Dans la figure 3.4 on va voir la précision de cette architecture, dans la figure 3.5 la perte de cette architecture. Pendant la phase d'entraînement et test.

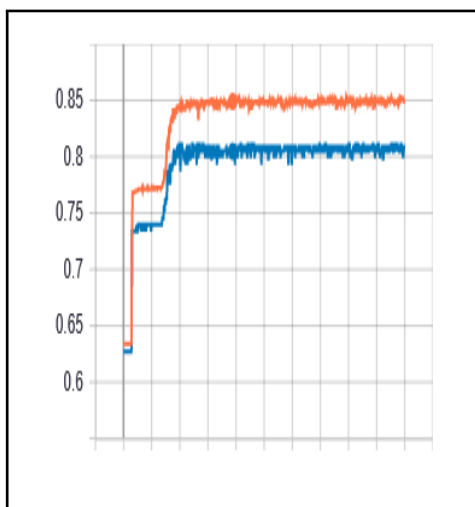


FIGURE 3.2 – Model 1 accuracy.

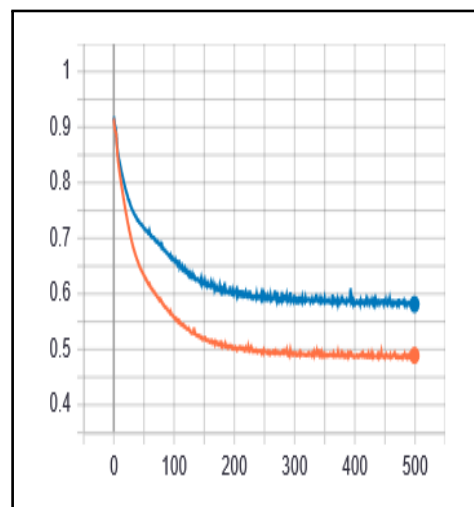


FIGURE 3.3 – Model 1 loss.

■ entraînement ■ test

3.4.1.2 Résultat modèle 2 (10x10) :

Dans ce modèle on a utilisé un réseau qui se compose de deux couches cachées chaque couche contient 10 neurones, de 3 entrées et 3 neurones de sortie.

Dans la figure 3.6 on va voir la précision de cette architecture, dans la figure 3.7 la perte de cette architecture. Pendant la phase d'entraînement et de test.

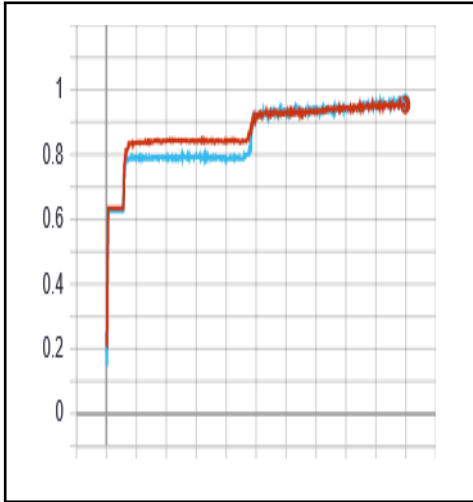


FIGURE 3.4 – Model 2 accuracy .

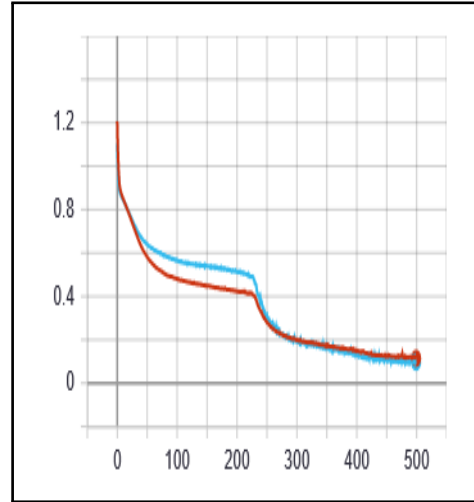


FIGURE 3.5 – Model 2 loss .

3.4.1.3 Résultat modèle 3 (15x15) :

Dans ce modèle on a utilisé un réseau qui se compose de deux couches cachées chaque couche contient 15 neurones, de 3 entrées et 3 neurones de sortie.

Dans la figure 3.8 on va voir la précision de cette architecture, dans la figure 3.9 la perte de cette architecture. Pendant la phase d'entraînement et de test.

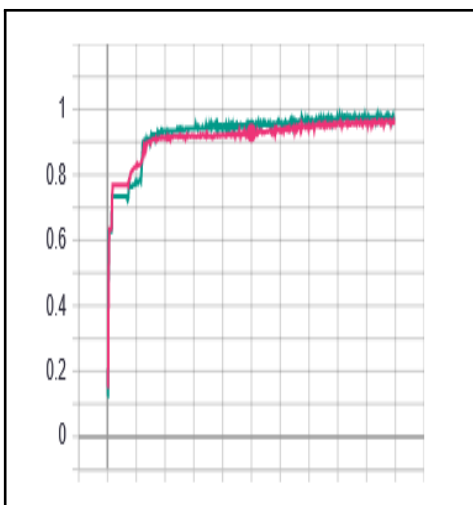


FIGURE 3.6 – Model 3 accuracy.

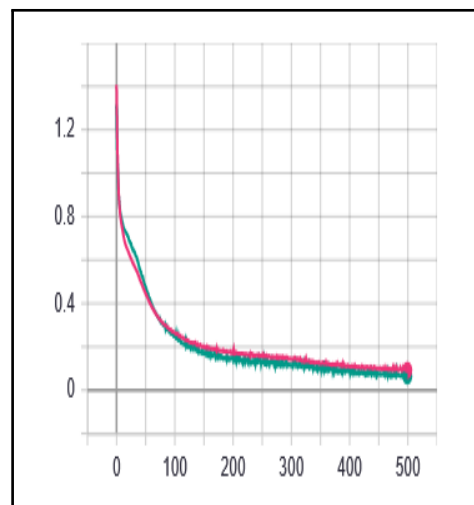


FIGURE 3.7 – Model 3 loss.

En résumé le **modèle 3** est le meilleur modèle car il a une perte plus petite (**0.05%**), un pourcentage de précision total (**0.97%**) plus grand que les autres modèles.

La figure 3.10 présente la matrice de confusion. Les sorties du réseau sont très précises, comme on peut le voir par le nombre élevé de réponses correctes dans les carrés bleus (diagonale principale) et le faible nombre de réponses incorrectes dans les carrés blancs. Signalons que notre modèle sur un ensemble de test contenant 169 valeurs, il a fait 4 prédictions fausses.

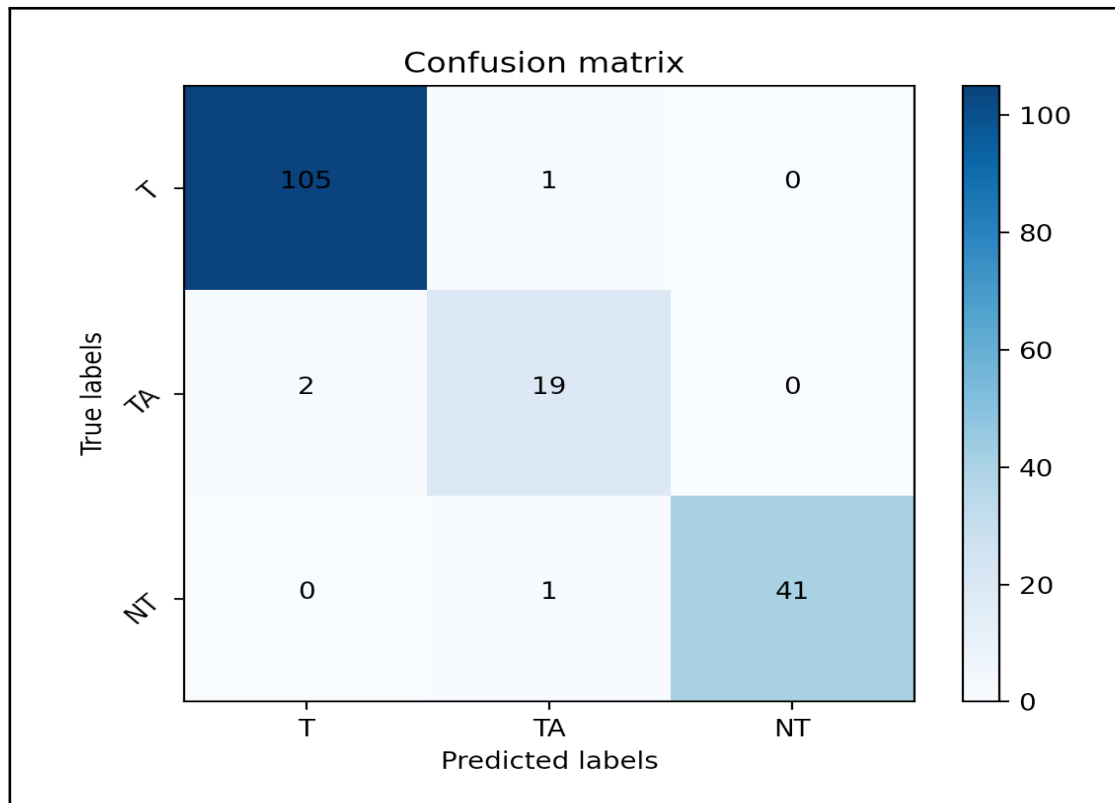


FIGURE 3.8 – La Matrice de confusion du modèle 3.

Conclusion

Dans ce chapitre, trois (03) modèles de classification basés sur le réseau de neurone ont été utilisés pour classer les données des capteurs. Pour l'évaluation de ces modèles, nous avons effectué des tests sur la base de données test. Les trois modèles ont été entraînés pour une période de 500 époques. Les 3 modèles ont donné des résultats différents. Le modèle qui a donné les meilleurs résultats de validation a été testé et validé. Nous pouvons constater que l'utilisation des réseaux de neurones avec apprentissage supervisé a permis d'avoir des résultats très satisfaisants, à savoir un taux de prédiction moyen de **97.04%**.

La sortie du réseau de neurone va être utilisée par la suite comme une entrée dans le système de décision de traitement.

Chapitre 4

Etude conceptuelle

4.1 Introduction

Dans ce chapitre, nous allons entamer la conception de notre système (application web et application labview) dont le but de déterminer les caractéristiques qu'on doit satisfaire pour aboutir à la réalisation de notre projet. Pour l'application web, nous utilisons le langage UML (Unified Modeling Language ou langage unifié de modélisation) qui permet de modéliser les applications. Pour les autres systèmes, nous allons expliquer leurs raisonnements.

4.2 Solution proposée

4.2.1 Présentation des composants

notre système est divisé de deux parties : **une plateforme web** et **une application labview**.

Nous avons développé une plateforme web sous php. Cette dernière est utilisée pour gérer et contrôler plusieurs tâches citons par exemple les plus essentiels (l'historique des détections et réparations des anomalies, la configuration des machines et la validation des check-list). Une fois l'information est récupérée le serveur web l'enregistre dans la base de données, l'application labview récupère ces données et les traite. Cette dernière est composée de deux systèmes :

1. **Un système de décision de traitement** :
 - (a) **Processus Automatique** : qui fait le traitement en temps réel des données de capteurs en utilisant le réseau de neurone.
 - (b) **Processus Manuelle** : qui fait la vérification du check-list.
2. **Un système expert** : listant les actions à prendre selon les codes de verrouillages présents sur la machine.

La figure suivante montre le schéma des composants du notre système.

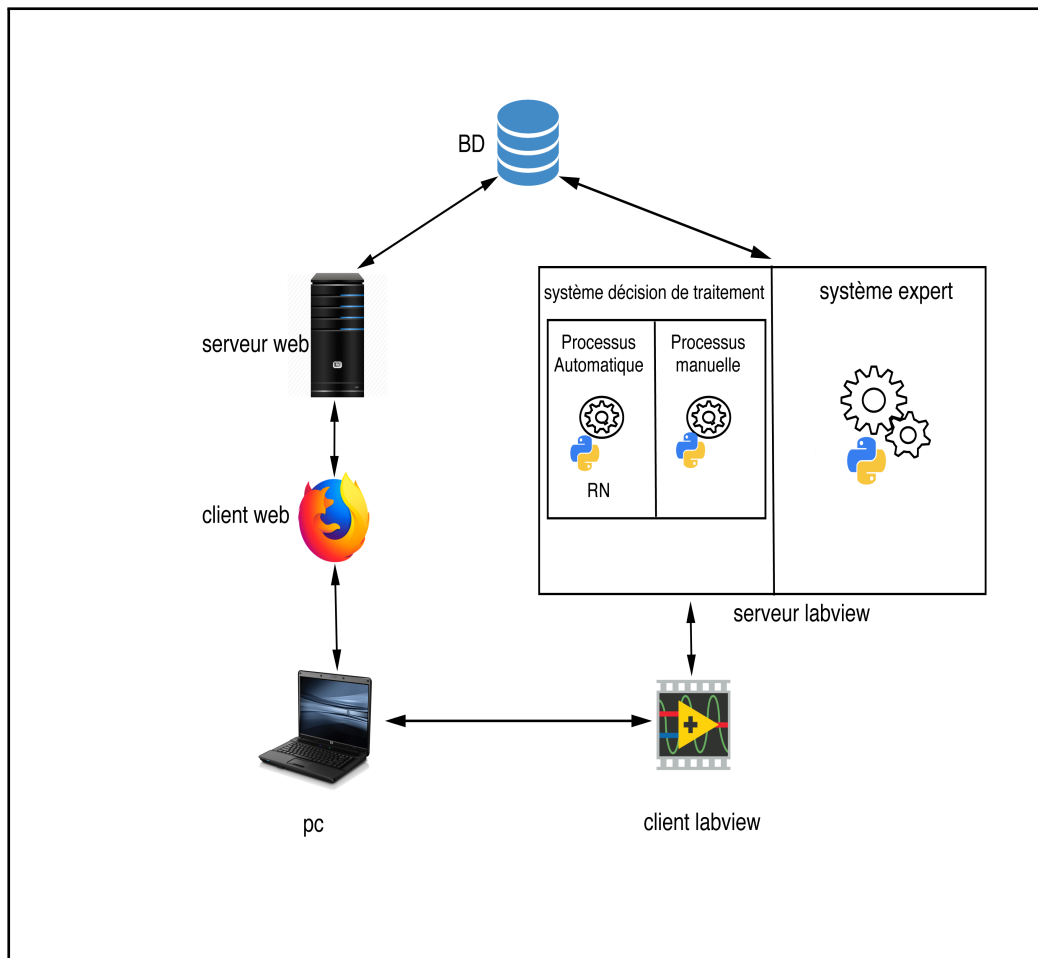


FIGURE 4.1 – le schéma des composants du système

4.3 UML

4.3.1 Définition d'UML

UML (Unified Modeling Language ou langage unifié de modélisation) est un langage graphique destiné à la modélisation des systèmes et de processus. C'est un langage basé sur l'approche par objets, celle qui a d'abord conduit à la création des langages de programmation comme Java, C++ ou Smalltalk. UML est unifié et permet de décrire un système à l'aide des diagrammes. Ces derniers permettant d'exprimer une représentation simplifiée du problème. Chaque type de diagramme offre une vue d'un système. La combinaison des différents types du diagramme offre une vue complète du système. [35]

4.4 La Modélisation de l'Application web

Pour la Modélisation de l'application , on est passé par trois étapes :

- Expression des besoins.
- Analyse des besoins.
- Conception détaillée.

4.5 Expression des besoins

L'objectif fondamental de cette étape est de déterminer les acteurs et les fonctionnalités de l'application à développer.

Nous nous intéressons donc, dans cette partie, à la réalisation de diagramme de cas d'utilisation qui nous montre les fonctionnalités de notre application. Les principaux cas d'utilisation seront accompagné d'une description textuelle.

4.5.1 Les besoin techniques

- **Interface utilisateur :**

L'interface doit être simple, intuitive et facile à manipuler pour l'utilisateur pour que celui-ci ne trouve pas de problèmes dans l'utilisation.

- **Développement de l'application**

- Environnement de développement : Visual Studio Code.
- Langage de programmation : Php , html/css et javascript.
- SGBD : MySQL.
- Environnement d'exécution : Windows , MacOS.

4.5.2 besoin fonctionnels

4.5.2.1 Exigences liées au système

- **Coté administrateur (chef de service) :**

- l'application web devra reconnaître l'administrateur grâce à son login et mot de passe.

- l'application web devra offrir une interface permettant à l'administrateur d'accéder à toutes les fonctionnalités de application :
 - i. De manipuler les utilisateurs.
 - ii. De Configurer les machines.
 - iii. De Consulter les états des anomalies.
 - iv. De faire le Suivi des réparations.
 - v. de consulter les statistiques.
- **Coté Utilisateur (Manipulateur, Radio-physicien, Radiothérapeute et technicien de maintenance) :**
 - l'application web devra reconnaître le type d'utilisateur grâce à son login et mot de passe.
 - l'application web devra offrir une interface permettant à l'utilisateur d'accéder à l'application.
 - l'application web devra permettre à l'utilisateur de :
 - i. De Consulter les états des anomalies.
 - ii. De faire le Suivi des réparations.
 - iii. de consulter les statistiques.

4.5.2.2 Les cas d'utilisation (UC)

4.5.2.2.1 Identification des Acteurs Un acteur représente l'abstraction d'un rôle joué par des entités externes (utilisateur, dispositifs matériels ou autres systèmes) qui interagissent directement sur le système étudié. [36]

notre application web est utilisé par deux types d'utilisateurs, ceux-ci sont distingués grâce à la nature des manipulations qu'ils exercent sur le système.

- **administrateur (chef service) :** C'est la personne qui est chargée de gérer tout l'application.
- **Utilisateur :** Cet utilisateur pourrait être soit un Manipulateur, Radio-physicien, Radiothérapeute et technicien de maintenance ou enfin technicien de maintenance.

4.5.2.2.2 Identification des cas d'utilisation Chaque cas d'utilisation doit faire l'objet d'une définition qui décrit l'intention de l'acteur lorsqu'il utilise le système et les séquences d'actions principales qu'il est susceptible d'effectuer. Ces définitions servent à fixer les idées lors de l'identification des cas d'utilisation. Grâce aux besoins récoltés et aux acteurs détectés nous avons déduit les cas d'utilisations suivants :

- S'Authentifier.
- gérer les utilisateurs.
- Configurer les machines
- remplir le check-list.
- réparer les anomalies.

4.5.2.2.3 Diagrammes de cas d'utilisation Le diagramme de cas d'utilisation est un schéma qui montre les cas d'utilisation reliés par des associations (lignes) à leurs acteurs. Chaque association signifie simplement « participe à ». Un cas d'utilisation doit être relié à au moins un acteur. [36]

La figure suivante représente le diagramme générale de cas d'utilisation :

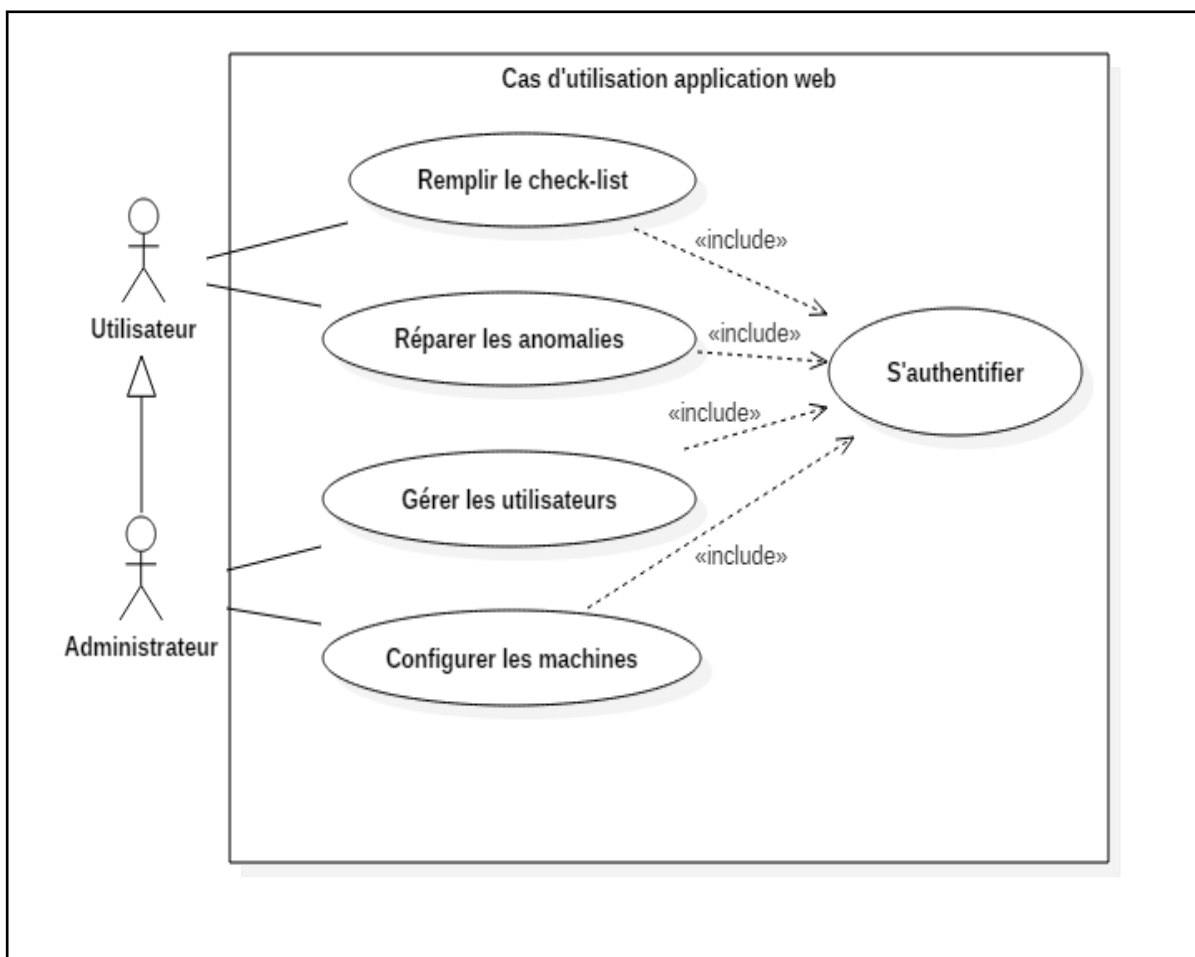


FIGURE 4.2 – le diagramme de cas d'utilisations générale de l'application web

- Diagramme de cas d'utilisation du "Remplir le check-list" :

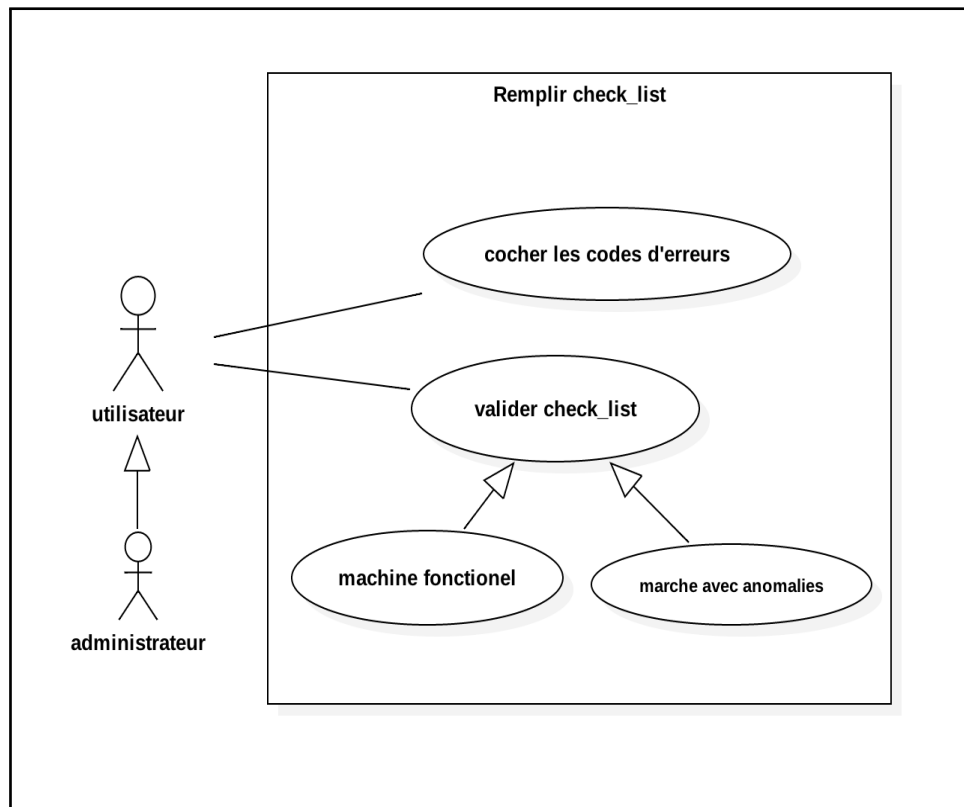


FIGURE 4.3 – Diagramme de cas d'utilisation du "Remplir le check-list"

- Diagramme de cas d'utilisation du "Réparer les anomalies" :

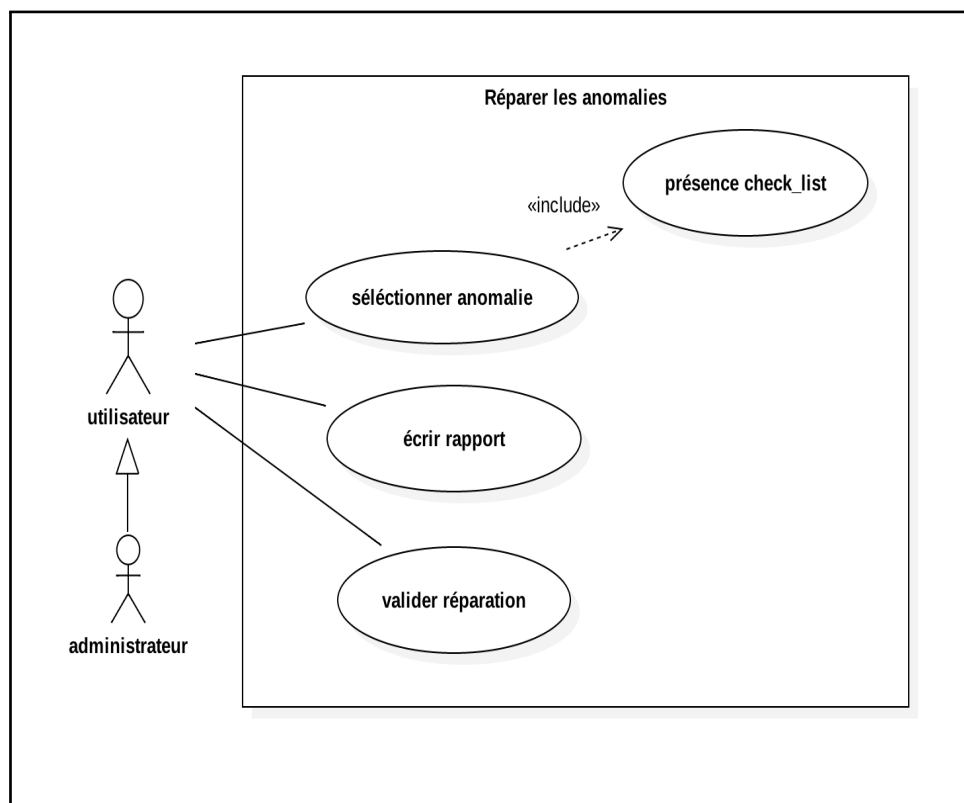


FIGURE 4.4 – Diagramme de cas d'utilisation du "Réparer les anomalies"

- Diagramme de cas d'utilisation du "Gérer les utilisateurs" :

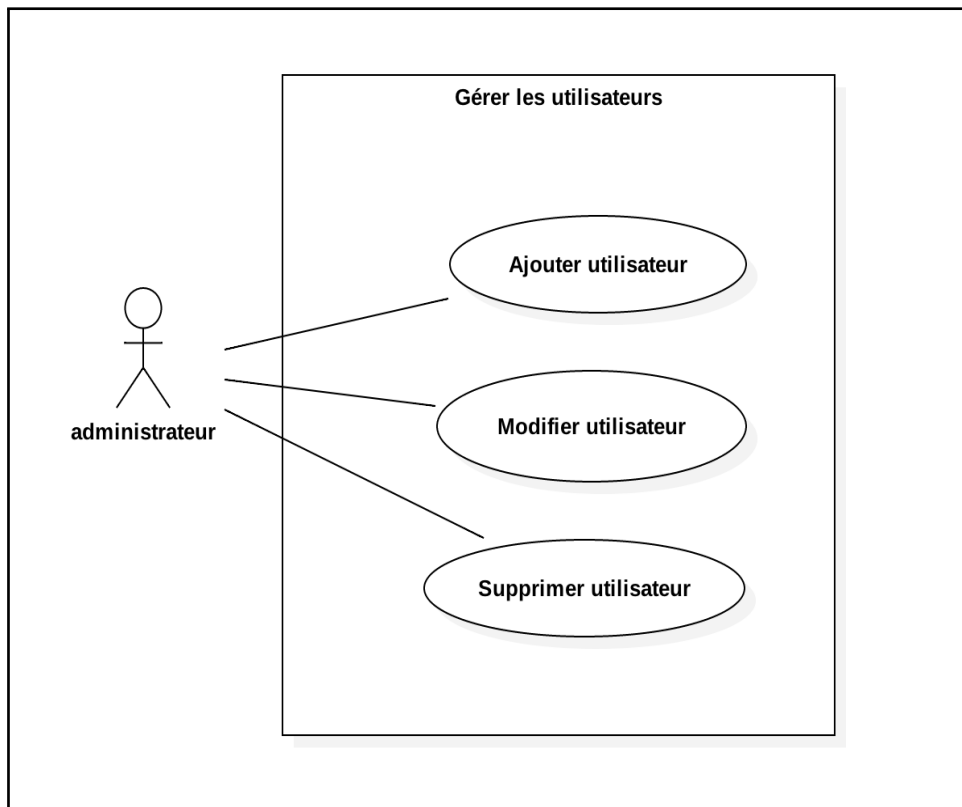


FIGURE 4.5 – Diagramme de cas d'utilisation du "Gérer les utilisateurs"

- Diagramme de cas d'utilisation du "configurer les machines" :

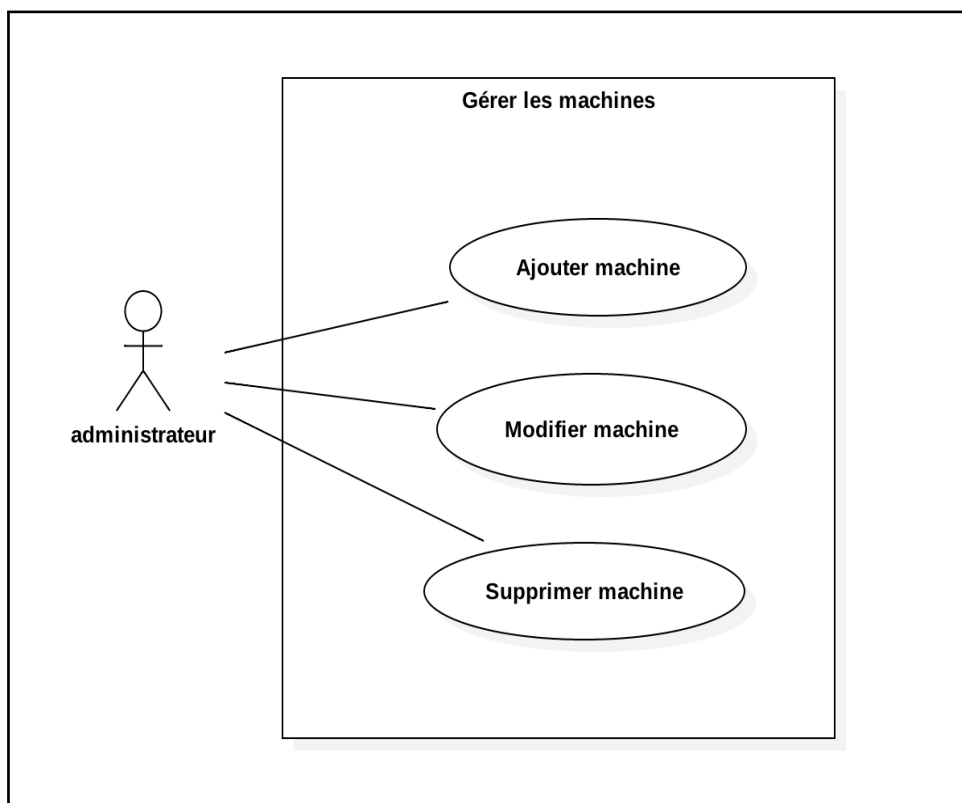


FIGURE 4.6 – Diagramme de cas d'utilisation du "configurer les machines"

4.5.2.2.4 Description textuelle de cas d'utilisation À chaque cas d'utilisation doit être associée une description textuelle des interactions entre l'acteur et le système et les actions que le système doit réaliser en vue de produire les résultats attendus par les acteurs. [36]

- **Objectif** : Décrire le contexte et les résultats attendus du cas d'utilisation.
- **Acteurs concernés** : Le ou les acteurs concernés par le cas doivent être identifiés en précisant globalement leur rôle.
- **Scénario nominal** : Il s'agit là du scénario principal qui doit se dérouler sans incident et qui permet d'aboutir au résultat souhaité.
- **Scénarios alternatifs** : Les autres scénarios, secondaires ou correspondants à la résolution d'anomalies, sont à décrire à ce niveau. Le lien avec le scénario principal se fait à l'aide d'une numérotation hiérarchisée.
- **Pré conditions** : Si certaines conditions particulières sont requises avant l'exécution du cas, elles sont à exprimer à ce niveau.
- **Post conditions** : si certaines conditions particulières doivent être réunies après l'exécution du cas, elles sont à exprimer à ce niveau.

- **Description textuelle du cas d'utilisation "Remplir le check-list" :**

Description du cas «Remplir le check-list»
<u>Identification</u> acteur principale : Administrateur , Utilisateur .
<u>Séquencement</u> Pré-conditions :s'authentifier. Enchaînement nominal :l'utilisateur remplit les codes d'erreurs et valide soit la machine est fonctionnel soit elle marche avec anomalies si elle affiche des code d'erreurs Enchaînement alternatif :l'utilisateur déclare que la machine est fonctionnel sans anomalies en cochant au moins un code d'erreur ou l'inverse. Post-conditions :le check-list est remplie dans la base de données.

- Description textuelle du cas d'utilisation "Réparer les anomalies" :

Description du cas «Réparer les anomalies»
<u>Identification</u> acteur principale : Administrateur , Utilisateur .
<u>Séquencement</u> Pré-conditions :s'authentifier. Enchaînement nominal :l'utilisateur sélectionne les anomalies et écrit un rapport pour chacune d'elles puis il valide . Post-conditions :l'anomalie est modifiée dans la base de données .

- Description textuelle du cas d'utilisation "Gérer les utilisateurs" :

Description du cas «Gérer les utilisateurs»
<u>Identification</u> acteur principale : Administrateur .
<u>Séquencement</u> Pré-conditions :s'authentifier avec les droits d'administrateur. Enchaînement nominal :l'administrateur remplit les information d'un utilisateur pour l'ajouter ou le modifier , sinon le supprimer Enchaînement alternatif :l'utilisateur figure déjà dans la base de donnée (l'ajout d'un nouveau utilisateur). Post-conditions :l'utilisateur est ajouté,modifié ou supprimé de la base de données .

- Description textuelle du cas d'utilisation"configurer les machines" :

Description du cas «configurer les machines»
<u>Identification</u> acteur principale : Administrateur .
<u>Séquencement</u> Pré-conditions :s'authentifier avec les droits d'administrateur. Enchaînement nominal :l'administrateur remplit les information de la machine pour l'ajouter ou la modifier , sinon la supprimer . Enchaînement alternatif :la machine figure déjà dans la base de donnée (l'ajout d'une nouvelle machine). Post-conditions :la machine est ajoutée,modifiée ou supprimée de la base de données .

4.6 L'analyse des besoins

L'analyse consiste à étudier précisément la spécification fonctionnelle de manière à obtenir une idée de ce que va réaliser le système en termes de métier. [36]

4.6.1 Diagramme de séquence (DES)

L'objectif du diagramme de séquence est de représenter les interactions entre objets en indiquant la chronologie des échanges. [36]

- **Messages synchrone et message asynchrone**

Dans un diagramme de séquence, deux types de messages peuvent être distingués :

- **Message synchrone** : Dans ce cas l'émetteur reste en attente de la réponse à son message avant de poursuivre ses actions. La flèche avec extrémité pleine symbolise ce type de message.
 - **Message asynchrone** : Dans ce cas, l'émetteur n'attend pas la réponse à son message, il poursuit l'exécution de ses opérations. C'est une flèche avec une extrémité non pleine qui symbolise ce type de message.
- **Ligne de vie** Elle représente la période de temps durant laquelle l'objet "existe". Les objets communiquent en échangeant des messages représentés sous forme de flèches. Lors de la création d'un objet, un message pointe sur le symbole de l'objet et quand il est détruit sa ligne de vie se termine par une croix en trait.

4.6.1.1 Diagramme de séquence de cas d'utilisation (S'authentifier)

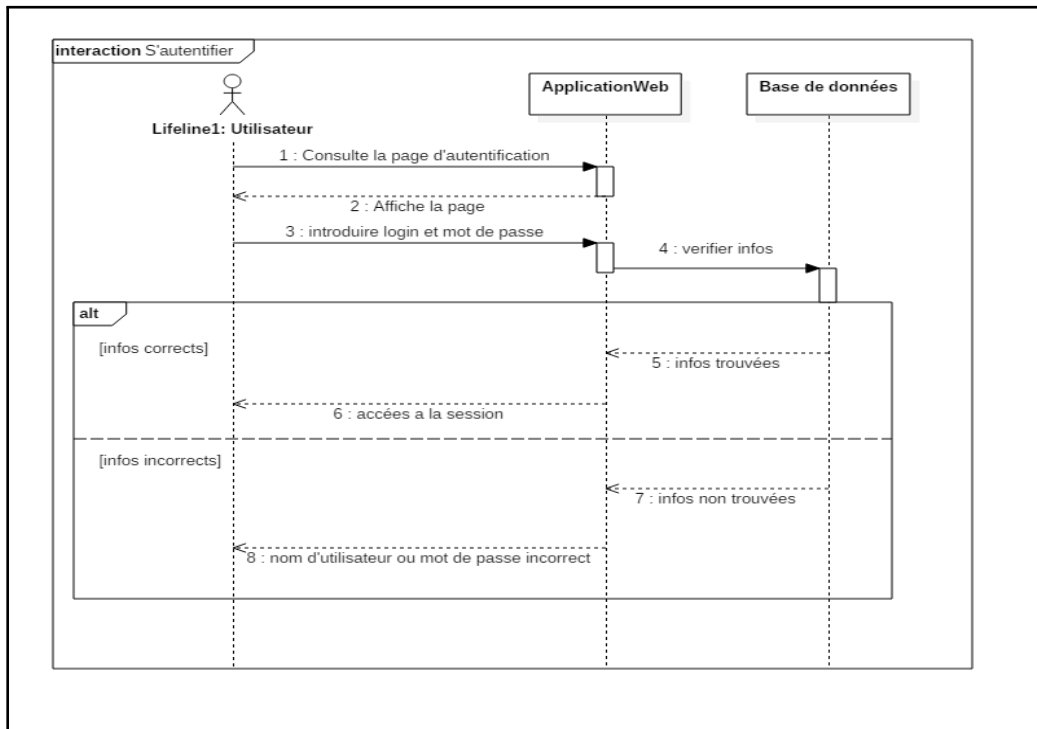


FIGURE 4.7 – Diagramme de séquence de cas d'utilisation (S'authentifier)

4.6.1.2 Diagramme de séquence de cas d'utilisation (Réparer les anomalies)

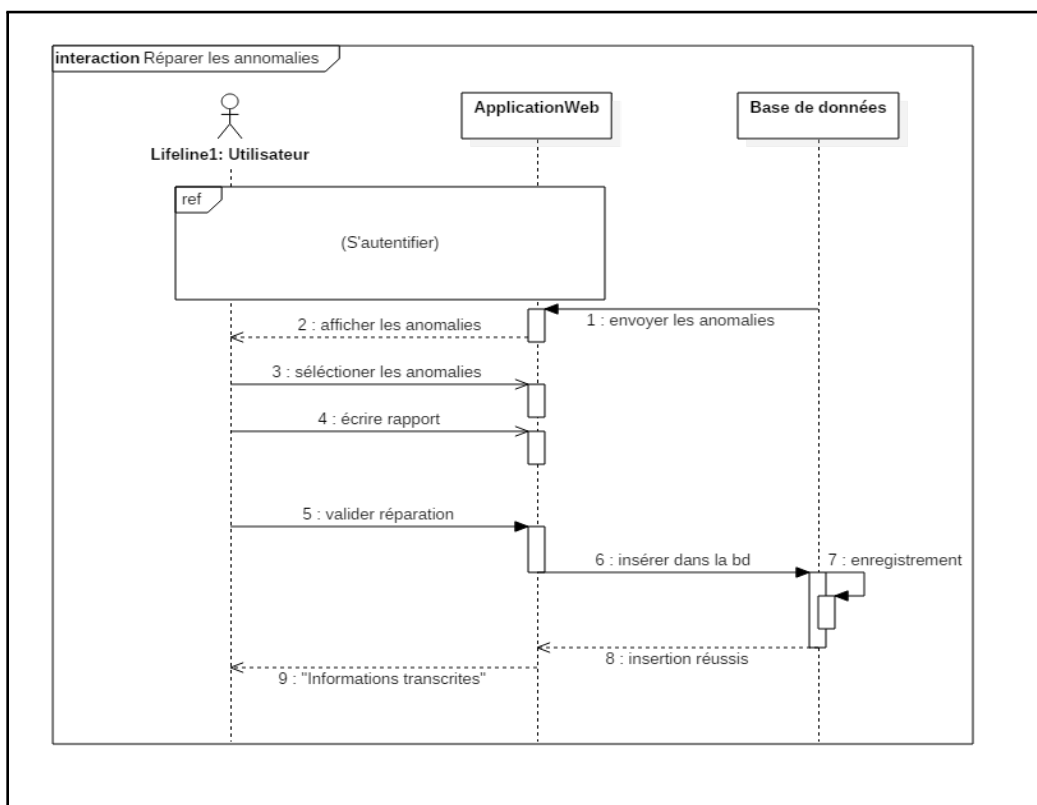


FIGURE 4.8 – Diagramme de séquence de cas d'utilisation (Réparer les anomalies)

4.6.1.3 Diagramme de séquence de cas d'utilisation (Remplir le check-list)

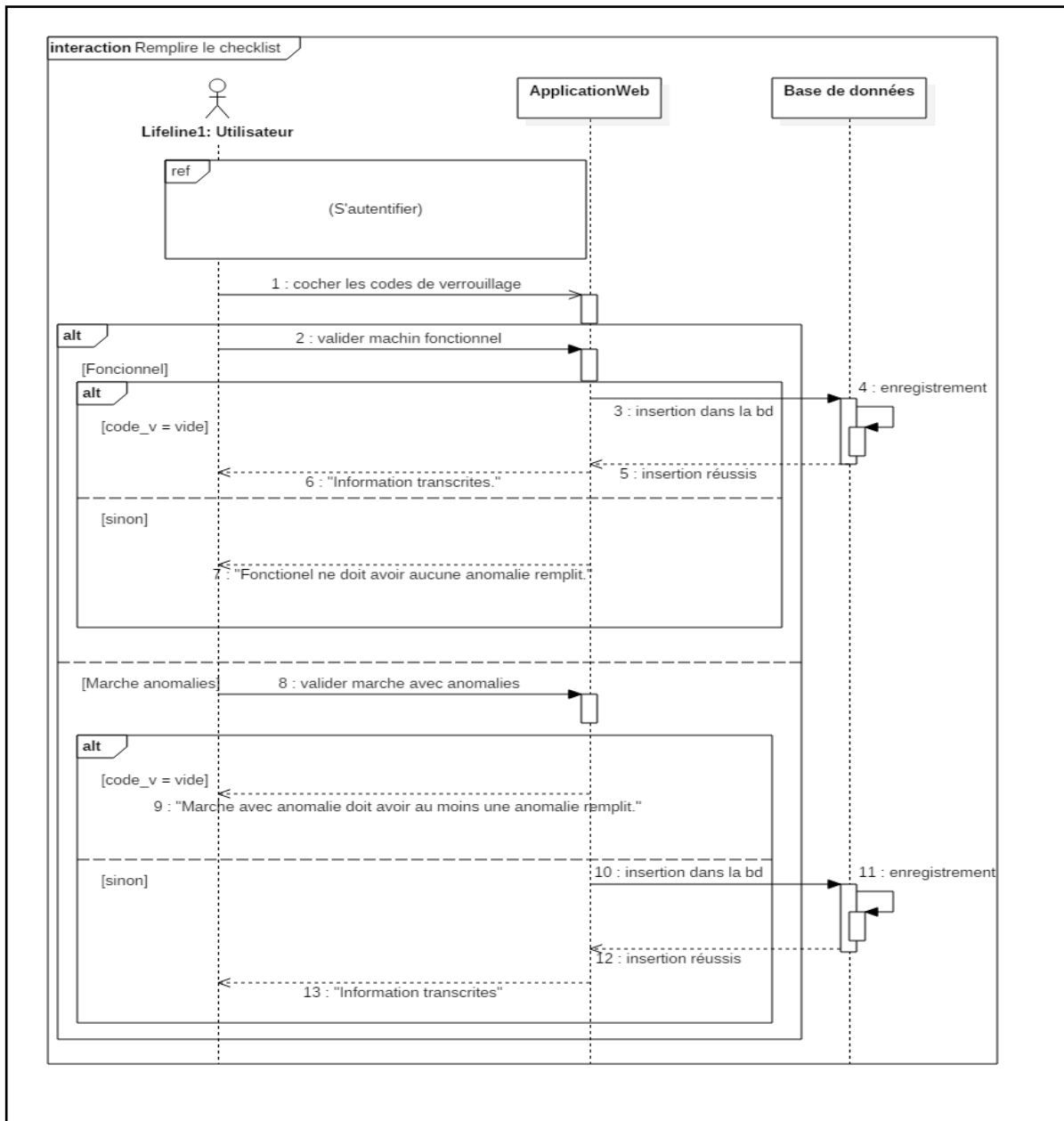


FIGURE 4.9 – Diagramme de séquence de cas d'utilisation (Remplir le check-list)

4.6.1.4 Diagramme de séquence de cas d'utilisation (Gérer les utilisateurs)

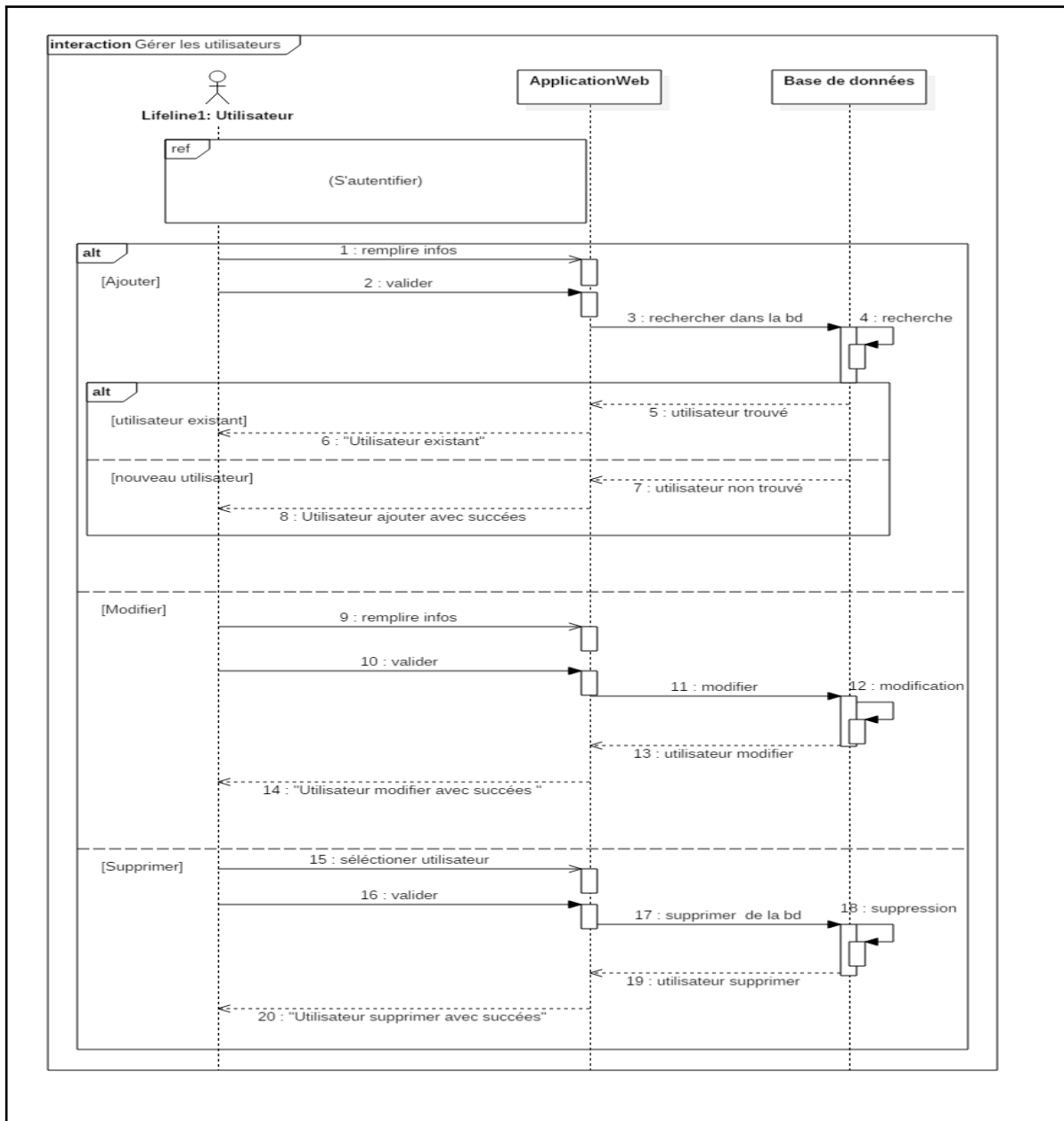


FIGURE 4.10 – Diagramme de séquence de cas d'utilisation (Gérer les utilisateurs)

4.6.1.5 Diagramme de séquence de cas d'utilisation (Configurer les machines)

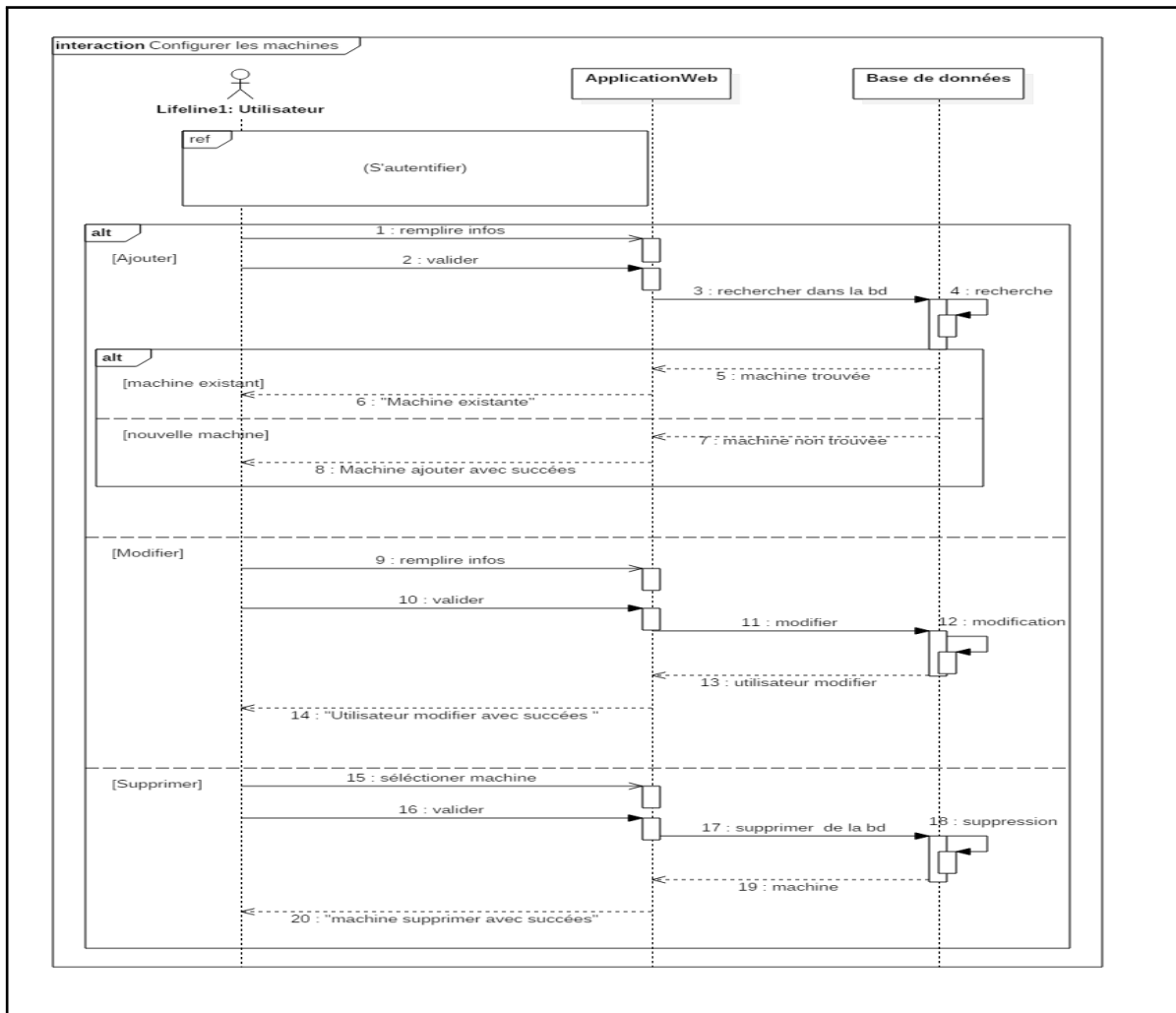


FIGURE 4.11 – Diagramme de séquence de cas d'utilisation (Configurer les machines)

4.7 Conception détaillée

Dans cette étape nous allons présenter le diagramme de classe de " l'application CPMC ".

4.7.1 Diagramme de classe

Le diagramme de classes est considéré comme le diagramme le plus important dans la modélisation orientée objet. Alors que le diagramme de cas d'utilisation montre un système du point de vue fonctionnel, le diagramme de classes montre la structure interne du système. Il contient principalement des classes. Une classe contient des attributs et des opérations (méthodes). [36]

la figure suivante montre le diagramme de calsse de l'application web :

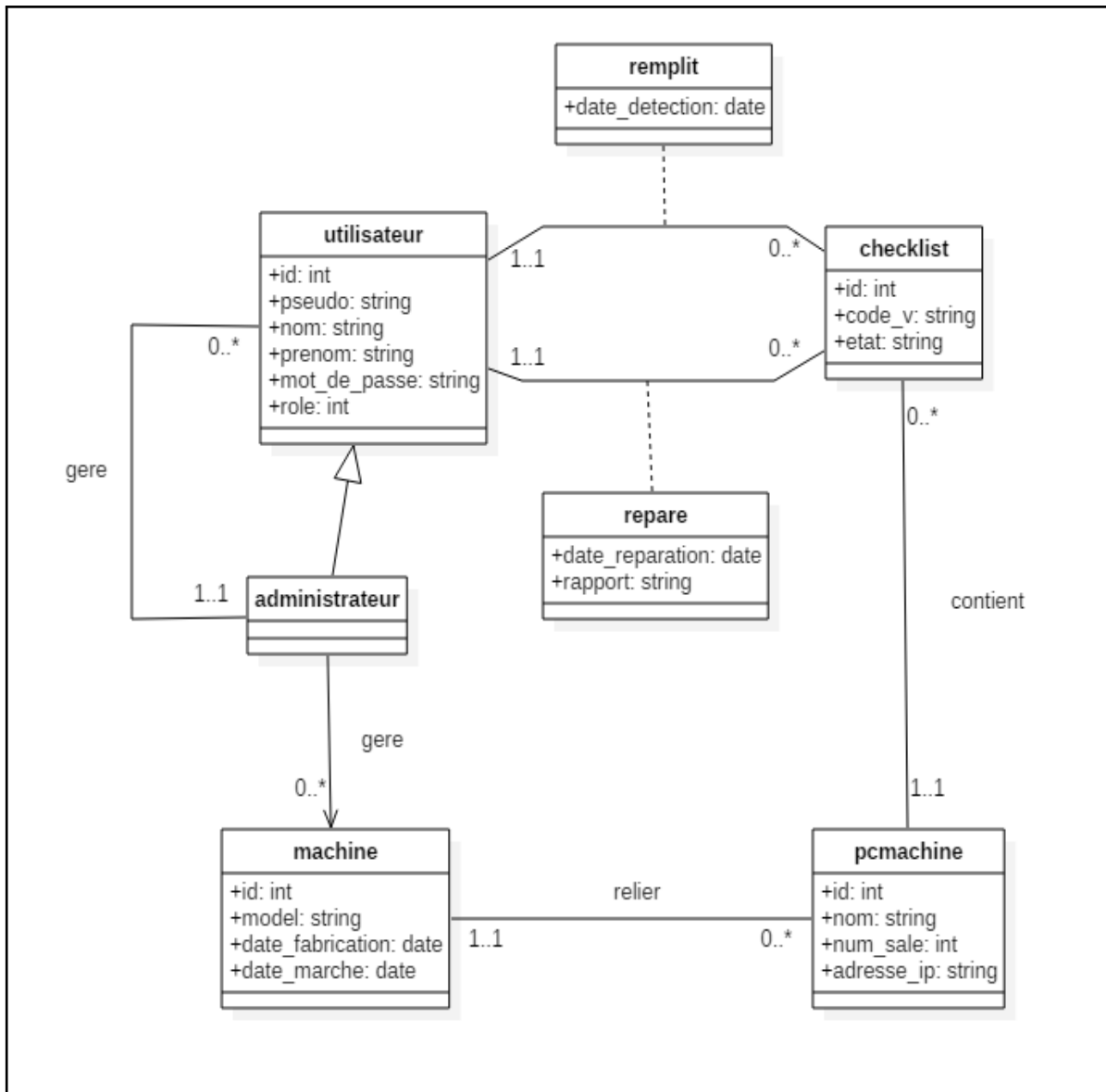


FIGURE 4.12 – Diagramme de classe de l'application web

4.8 Conception de l'application labview

4.8.1 système de décision de traitement

Le but de ce système est de donner une décision sur l'éventualité du traitement , il est composé de deux processus :

- **un processus manuelle** : vérification hebdomadaire du check-list ainsi que la gravité de codes de verrouillages présents sur la machine.
- **un processus automatique** : traitement des données des capteurs avec le réseau de neurone en temps réel.

Le résultat du **SDT** est la combinaison des deux processus (manuelle et automatique).

Trois types de résultat ont été discutés qui sont :

- **T** : Pas d'anomalies, le traitement du cancer peut se dérouler normalement.
- **TA** : Anomalies à corriger durant la phase de traitement du cancer.
- **NT** : Ne pas traiter car la ou les anomalies sont graves et doivent être corrigées, la phase du traitement du cancer est à l'arrêt.

4.8.1.1 Le schéma conceptuel du système de décision de traitement

Pour mieux expliquer le système de décision de traitement, on présente le schéma conceptuel suivant (figure 4.13) :

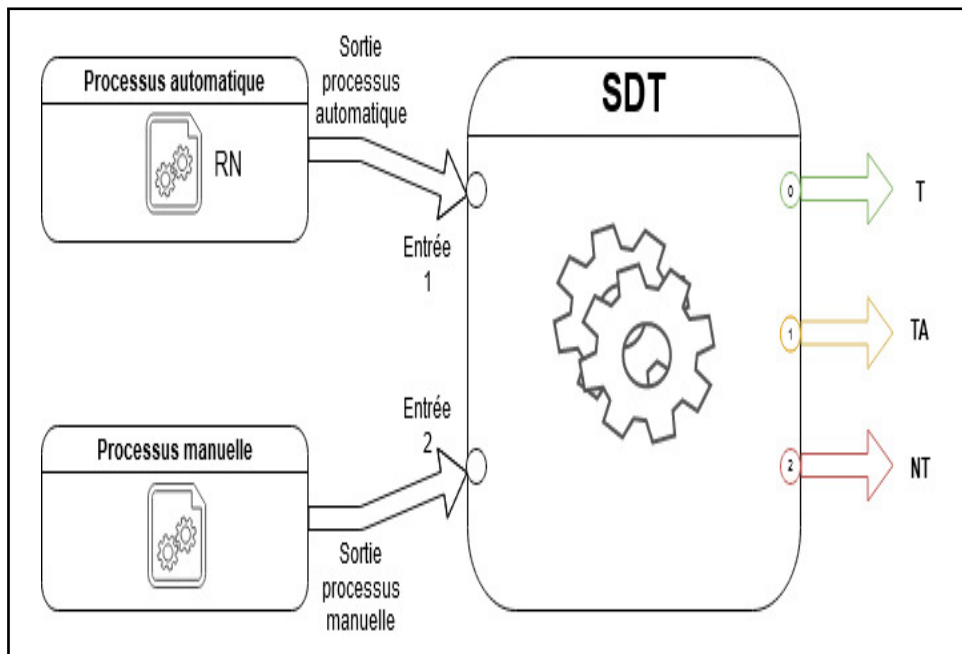


FIGURE 4.13 – le schéma conceptuel du système de décision .

- L'entrée 1 prend comme valeur le résultat du réseau de neurone.
- L'entrée 2 prend les valeurs suivantes selon certaines conditions présentés dans le tableau suivant :

TABLE 4.5 – Valeur de l'entrée 2 du SDT

valeur	condition
T (0)	checklist hebdomadaire fait ET aucune anomalie (code) détecter
TA (1)	checklist hebdomadaire fait ET anomalie (code) mineur détecter
NT (2)	absence checklist hebdomadaire OU anomalie (code) majeur détecter

4.8.1.2 Raisonnement du système de décision

Comme nous avons mentionné précédemment le SDT fait la combinaison de ses deux entrées (processus manuelle et automatique).

Le Résultat de cette combinaison est décrit sous forme de règles illustré dans le tableau suivant :

TABLE 4.6 – Le raisonnement du SDT

N° règle	Combinaison	Résultat
1	$PM = 'NT'$	'NT'
2	$PM = 'T' \text{ ET } (PA = 'T' \text{ OU } PA = 'TA' \text{ OU } PA = 'NT')$	PA
3	$PM = 'TA' \text{ ET } (PA = 'T' \text{ OU } PA = 'TA')$	PM
4	$PM = 'TA' \text{ ET } PA = 'NT'$	PA

avec :

- PM : Processus Manuelle.
- PA : Processus Automatique.
- 'T' : 'Traitement'.
- 'TA' : 'Traitement avec anomalies'.
- 'NT' : 'Pas de traitement'.

4.8.2 Le système expert

le rôle du système expert est de dicter les actions à prendre concernant la réparation des machines en temps réel selon les codes de verrouillages signalés depuis l'application web.

4.8.2.1 Représentation des Connaissances d'expertise

Après avoir discuter avec l'expert du domaine nous avons pu lister les différents actions a faire sous formes de règles :

Si "CODE" Alors "ACTION"

au cours de notre étude nous avons compté 68 codes d'erreurs pour un total théorique de 2.95×10^{20} d'actions possibles, en règle générale le nombre de combinaisons se calcule avec la formule suivante :

NbrsCombinaisons = $\sum_{i=0}^n \frac{n!}{i!(n-i)!}$ où n représente le nombre total de codes d'erreurs.[37]

Mais dans la pratique nous avons considérablement réduis ce nombre en classant dans un dictionnaire de données tous les codes qui donnent les mêmes actions, pour enfin arriver a un totale de 35 actions a faire pour toute les combinaisons possibles.

4.8.2.2 Le schéma représentatif du système expert

Le schéma suivant (figure 4.14) illustre le mode de fonctionnement du système expert proposé :

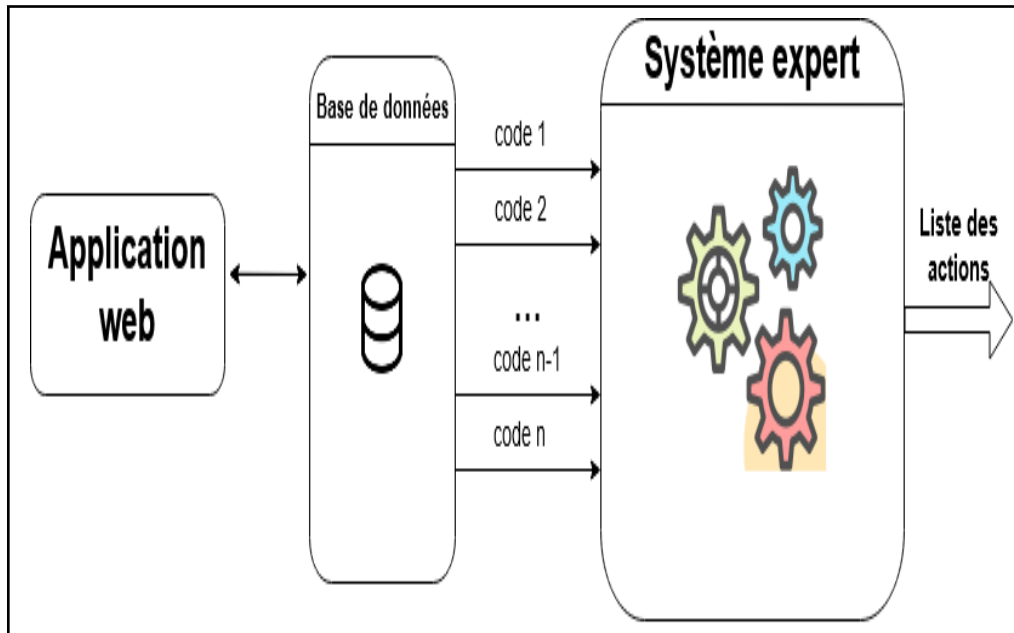


FIGURE 4.14 – Le schéma représentatif du système expert.

4.9 Conclusion

Tout au long de ce chapitre, nous avons exposé les différents besoins auxquels doit répondre l'application à réaliser ainsi que la solution que nous allons adopter pour la satisfaction de ces exigences stipulées. Dans le chapitre suivant, nous allons aborder l'implémentation de cette solution proposée tout en évoquant l'ensemble des choix techniques et logiciels qui nous ont aidés dans la réalisation de cette tâche.

Chapitre 5

Implémentation

5.1 Introduction

La réalisation est la phase la plus importante après celle de la conception. Le choix des outils de développement détermine énormément le coût en temps de programmation, ainsi que la flexibilité du produit à réaliser. Cette phase consiste à transformer le modèle conceptuel établi en des composants logiciels formant notre système. Dans un premier temps, nous allons procéder à la spécification de l'environnement matériel et logiciel utilisé dans notre projet. Dans un deuxième temps, nous nous intéresserons à décrire les différentes parties de notre système « Plateforme de Gestion et de Contrôle d'Anomalies dans un Service de Radiothérapie ».

5.2 Choix techniques

5.2.1 Langages de programmation

Pour l'**application web** on a utilisé :

- **PHP** : Hypertext Preprocessor, plus connu sous son sigle PHP (sigle auto-référentiel), est un "langage de programmation" libre, principalement utilisé pour produire des pages Web dynamiques via un serveur HTTP.[38]

PHP a permis de créer un grand nombre de sites web célèbres, comme Facebook et Wikipédia Il est considéré comme une des bases de la création de sites web dits dynamiques mais également des applications web.

- **HTML** : Le HyperText Markup Language, généralement abrégé HTML ou dans sa dernière version HTML5, est le langage de balisage conçu pour représenter les pages web. C'est un langage permettant d'écrire de l'hypertexte, d'où son nom. HTML permet également de structurer sémantiquement

la page, de mettre en forme le contenu, de créer des formulaires de saisie, d'inclure des ressources multimédias dont des images, des vidéos, et des programmes informatiques. Il est souvent utilisé conjointement avec le langage de programmation JavaScript et des feuilles de style en cascade (CSS) . Il s'agit d'un format ouvert.[39]

- **CSS** : Les feuilles de style en cascade, généralement appelées CSS de l'anglais Cascading Style Sheets, forment un langage informatique qui décrit la présentation des documents HTML et XML.CSS devient couramment utilisé dans la conception de sites web et bien pris en charge par les navigateurs web dans les années 2000.[40]

- **JAVASCRIPT** : est un langage de programmation interprété, orienté objet, qui est principalement utilisé dans la forme côté client.

Ce langage de programmation est utilisé pour tout : applications web, serveurs, applications mobiles... Sa particularité est qu'il ne nécessite aucune compilation, puisque c'est le navigateur lui-même qui lit le code et effectue les actions qu'il indique. Cela en fait l'un des meilleurs langages de programmation pour créer des sites web dans lesquels vous souhaitez inclure des éléments interactifs ou très visuels.[41]

- **AJAX** : désigne une technologie qui s'est popularisée dans le domaine de la création de sites internet. Elle est principalement utilisée pour apporter de l'interactivité au sein des pages d'un site web tout en économisant les ressources serveur.

En effet, AJAX permet de communiquer avec le serveur à l'aide de code Javascript en arrière-plan pendant que la page est affichée à l'écran. Ainsi le contenu de la page peut être modifié sans qu'il soit nécessaire de faire transiter et afficher la page en entier. Elle est particulièrement utilisée pour la mise à jour des formulaires et des paniers sur la plupart des sites web. C'est une technologie asynchrone : le code de la page continue de s'exécuter pendant que l'appel vers le serveur est effectué.[42]

- **SQL** : (sigle de Structured Query Language, en français langage de requête structurée) est un langage informatique normalisé servant à exploiter des bases de données relationnelles. La partie langage de manipulation des données de SQL permet de rechercher, d'ajouter, de modifier ou de supprimer des données dans les bases de données relationnelles. [43]

Pour l'application **desktop** on a utilisé :

- **Labview** : (contraction de Laboratory Virtual Instrument Engineering Workbench) est le cœur d'une plate-forme de conception de systèmes de mesure et de contrôle, fondée sur un environnement de développement graphique de National Instruments. LabVIEW est utilisé principalement pour la mesure par acquisition de données, pour le contrôle d'instruments et pour l'automatisme industriel. La plate-forme de développement s'exécute sous différents systèmes d'exploitation comme Microsoft Windows, Linux et Mac OS X. LabVIEW peut générer du code sur ces systèmes d'exploitation mais également sur des plate-formes temps réel, des systèmes embarqués ou des composants reprogrammables .[44]
- **Toolkit database connectivity** : est un toolkit qui permet aux programmes Labview de communiquer avec une base de données, ce Toolkit est utilisés pour obtenir les informations générales sur la base de données on peut voir les noms des tables les champs (colonnes) dans un tableau. On peut également consulter et régler les différents paramètres de connexion pour la base de données . Et aussi on peut stocker des enregistrements de déposer et replacez-les dans de Labview dans un fichier.
Il est fourni avec un outil permettant d'effectuer un test de connexion rapide à n'importe quelle base de données.[45]
- **Python** : est un langage de script de haut niveau, structuré et open source, Développé à l'origine par Guido van Rossum en 1989, il est comme la plupart des applications et outils open source, maintenu par une équipe de développeurs un peu partout dans le monde. Conçu pour être orienté objet, Il est le langage de programmation le plus utilisé dans le domaine du Machine Learning, du Big Data et de la Data Science.[46]
- **Tensorflow** : est une plate-forme Open Source développé par Google de bout en bout dédiée au machine learning. Elle propose un écosystème complet et flexible d'outils, de bibliothèques et de ressources communautaires permettant aux chercheurs d'avancer dans le domaine du machine learning, et aux développeurs de créer et de déployer facilement des applications qui exploitent cette technologie.[47]
- **Numpy** : est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux.[48]

- **Matplotlib** : est une bibliothèque du langage de programmation Python destinée à tracer et visualiser des données sous formes de graphiques.[49]

5.2.2 Architecture du système (Architecture à trois niveaux)

Dans l'architecture à trois niveaux, les applications au niveau serveur sont délocalisées, c'est-à-dire que chaque serveur est spécialisé dans une tâche (serveur application/ serveur de base de données par exemple). L'architecture à trois niveaux permet :

- Une plus grande flexibilité/souplesse.
- Une sécurité accrue car la sécurité peut être définie indépendamment pour chaque service, et à chaque niveau.
- Des meilleures performances, étant donné le partage des tâches entre les différents serveurs.

Cette architecture fait intervenir trois parties indépendantes les unes des autres :

1. **La couche de données liée au serveur de base de données (SGBD)** : qui permet le stockage et l'accès aux données. Le système de stockage des données a pour but de conserver une quantité plus ou moins importante de données de façon structurée.
2. **La logique applicative** : elle se compose généralement d'un script ou d'un programme qui constitue les traitements métier nécessaires sur l'information afin de la rendre exploitable par chaque utilisateur.
3. **La couche présentation (ou affichage)** : C'est la partie visible pour l'utilisateur. Elle a donc une importance primordiale pour rendre l'information lisible, compréhensible et accessible.

5.2.3 Protocole

Dans notre projet, nous avons utilisé le protocole **HTTP** pour l'application web, afin de communiquer les données entre la partie cliente web et le serveur web pour la synchronisation. En effet, Le HTTP est un protocole qui définit la communication entre un serveur et un client. Pour l'application labview, nous avons implémenté notre propre protocole applicatif à l'aide du protocole **TCP** en utilisant les sockets.

5.3 Outils matériels

La programmation a été effectuée sur deux (03) ordinateurs :

1. **Macbook Pro** doté de la configuration suivante :

- Intel(R) Core(TM) i5 CPU @2.7GHZ
- 8.00 Go de RAM
- Système d'exploitation macOS Mojave 64bits

2. **PC Bureau** doté de la configuration suivante :

- Intel(R) Core(TM) i5 CPU @3.3GHZ
- 8.00 Go de RAM
- Système d'exploitation Windows 10 Professionnel.

3. **LENOVO thinkpad** doté de la configuration suivante :

- Intel(R) Core(TM) i7 CPU @2.4GHZ
- 4.00 Go de RAM
- Système d'exploitation Windows 10 Professionnel

5.4 Environnement de développement

5.4.1 Outils de développement

5.4.1.1 VS Code

Visual Studio Code est un éditeur de code open-source, gratuit et multi-plateforme (Windows, Mac et Linux), développé par Microsoft, à ne pas confondre avec Visual Studio, l'IDE propriétaire de Microsoft. VS Code est développé avec Electron . Principalement conçu pour le développement d'application avec JavaScript, TypeScript et Node.js, l'éditeur peut s'adapter à d'autres types de langages grâce à un système d'extension bien fourni.[50]

5.4.1.2 WampServer

WampServer est une plate-forme de développement Web sous Windows. Il permet de développer des applications Web dynamiques à l'aide du serveur Apache2, du langage de scripts PHP et d'une base de données MySQL. Il possède également PHPMyAdmin pour gérer plus facilement les bases de données.[51]

5.4.1.3 MySQL

MySQL est un système de gestion de bases de données relationnelles (SGBDR). Il est distribué sous une double licence GPL et propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) .[52]

5.5 Présentation de l'application web CPMC

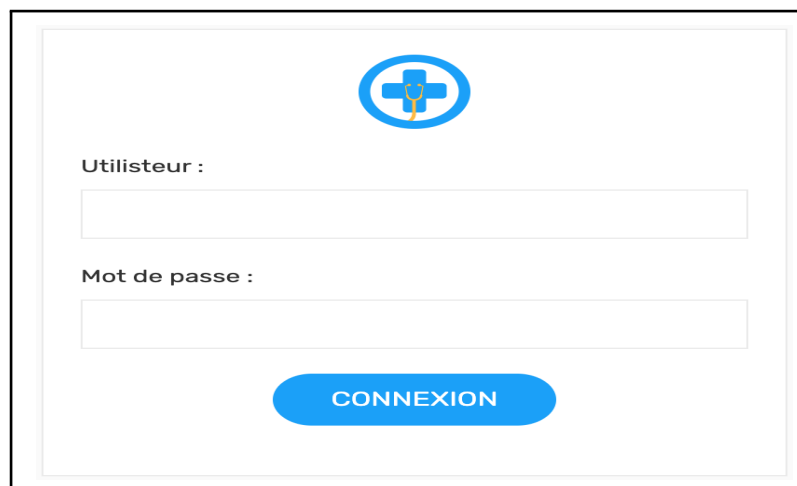
5.5.1 introduction

La gestion des machines médical programmé par un système informatique connecté à une base de données est nécessaire et semble obligatoire pour assurer une meilleure qualité des soins. Cela permettra d'avoir une meilleure traçabilité et évitera des accidents aux patients (traitement par radiothérapie non conforme). Le personnel médical doit se déclarer et s'authentifier avant toute procédure de traitement. En cas d'accidents (il en arrive souvent au niveau mondial), le médecin traitant (physicien médical) ou l'anomalie technique sera facilement détectée et identifiée.

5.5.2 Présentation des interfaces de L'application web CMPC

5.5.2.1 Interface Authentification

L'application CMPC couvre deux sessions fonctionnelles Administrateur (chef service) et utilisateur. Pour chaque cas d'utilisation, les personnes après leur avoir attribué les informations relatives à leurs comptes par le chef de service doivent s'authentifier au système (figure 5.1).



The image shows a web-based authentication form. At the top center is a blue circular icon containing a white medical cross with a yellow caduceus symbol. Below the icon, the text 'Utilisateur :' is followed by a white rectangular input field. Underneath that, the text 'Mot de passe :' is followed by another white rectangular input field. At the bottom center of the form is a blue rounded rectangular button with the word 'CONNEXION' written in white capital letters.

FIGURE 5.1 – Interface Authentification

5.5.2.2 Page d'accueil

Après avoir s'identifier, l'admin accède à la page d'accueil (figure 5.2).



FIGURE 5.2 – Page d'accueil administrateur

Après avoir s'identifier, l'utilisateur accède à la page d'accueil (figure 5.3).



FIGURE 5.3 – Page d'accueil utilisateur

5.5.2.3 Page gestion des utilisateurs (admin)

La figure (5.4) donne l'interface à travers laquelle l'admin gère tous les utilisateurs (modifier, rechercher et supprimer un utilisateur) .

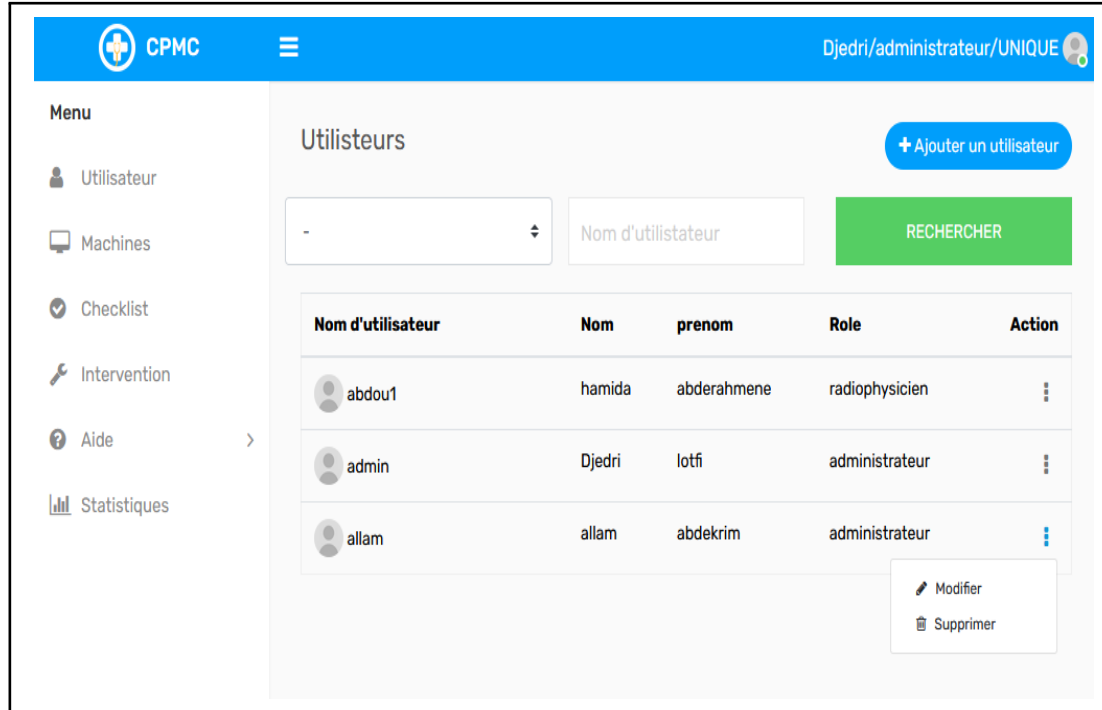


FIGURE 5.4 – Page gestion des utilisateurs

La figure (5.5) donne la fenêtre à travers laquelle l'admin remplit le informations et ajoute un nouveau utilisateur.

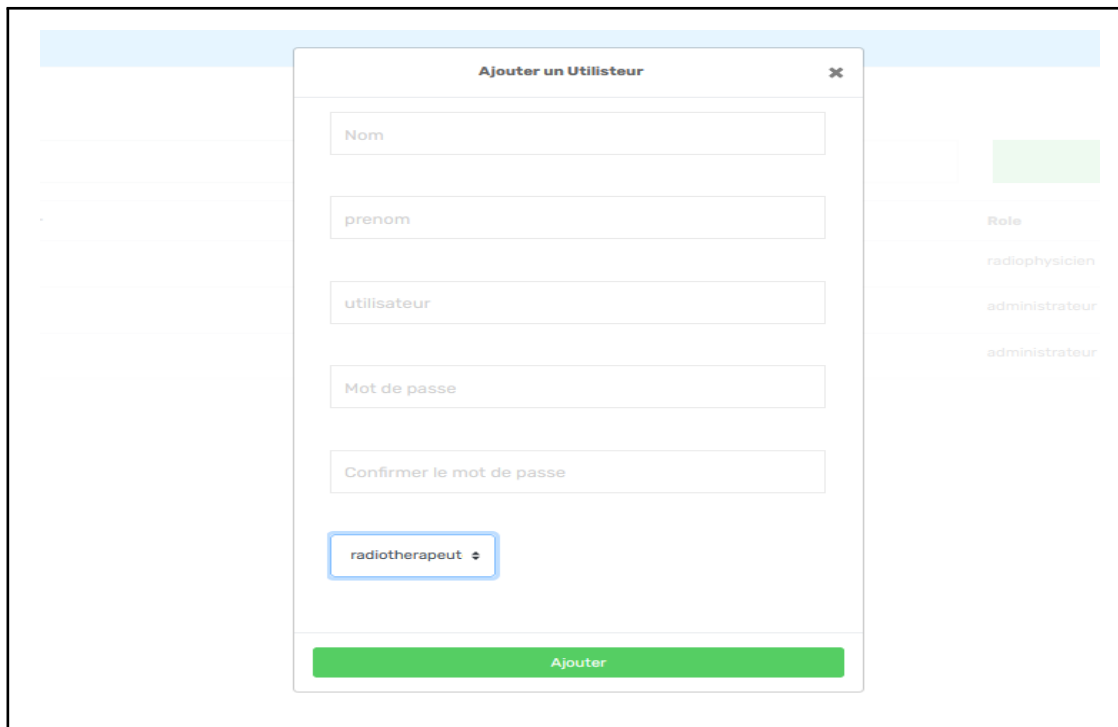


FIGURE 5.5 – Page d'ajout utilisateur

5.5.2.4 Page gestion machine (admin)

La figure (5.6) donne l'interface à travers laquelle l'admin gère tous les machines (ajouter, modifier et supprimer une machine) .

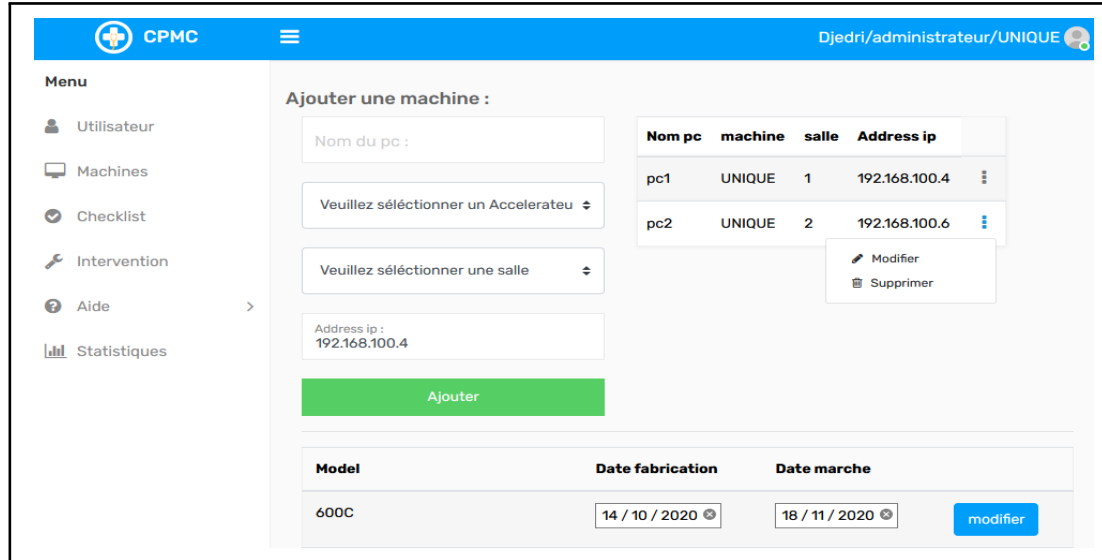


FIGURE 5.6 – Page gestion machine

5.5.2.5 Page check-list

Dans cette interface (figure 5.7) l'utilisateur coche les codes d'erreurs présents sur la machine et valide par marche avec anomalies. Si aucun code n'est coché, il valide par fonctionnel.

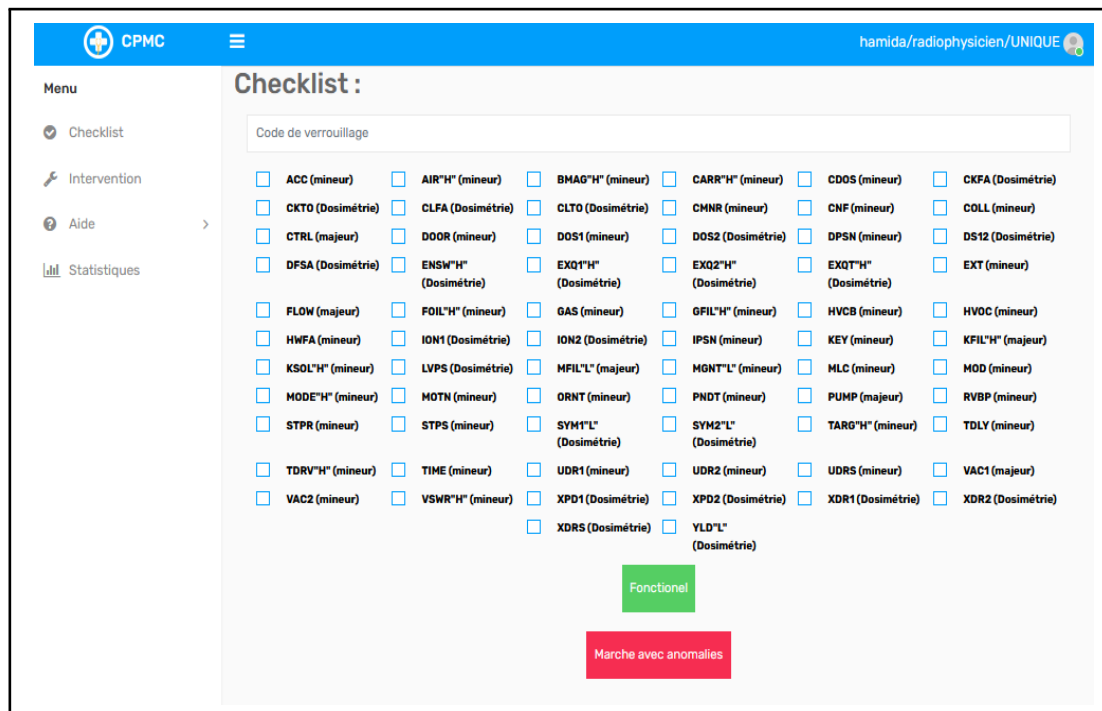


FIGURE 5.7 – Page checklist

5.5.2.6 Page d'intervention (réparation)

Dans cette interface (figure 5.8) l'utilisateur coche les codes d'erreurs à réparer puis il écrit son rapport sur chaque code d'erreur et enfin il valide en cliquant sur Réparer.



FIGURE 5.8 – Page d'intervention

5.5.2.7 Pages d'aides

Pour mieux administrer l'application deux menus d'aides à la décision dans le cas d'une intervention ou déclenchement d'une Anomalie ou panne (figure 5.9 et 5.10).

- **Codes des verrouillages de sécurité :** Ce menu est utilisé pour comprendre les codes des verrouillages des pannes.
- **TOP :** La procédure du TOP.

La figure (5.9) donne l'interface qui décrit les codes des verrouillages des pannes (anomalies).



FIGURE 5.9 – Page de Codes des verrouillages (anomalies)

La figure (5.10) donne l'interface qui décrit la procédure du top des radios physiques.

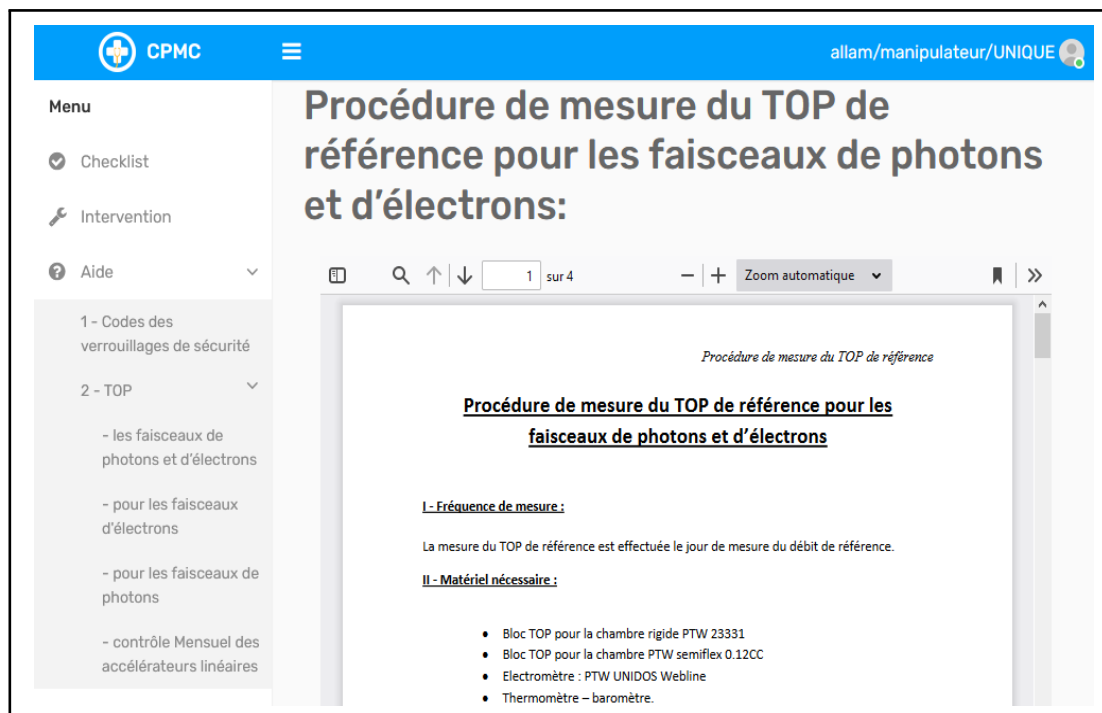


FIGURE 5.10 – Page TOP

5.5.2.8 Pages de statistiques

La figure (5.11) donne l'interface qui affiche les données des capteurs traités par l'application labview.

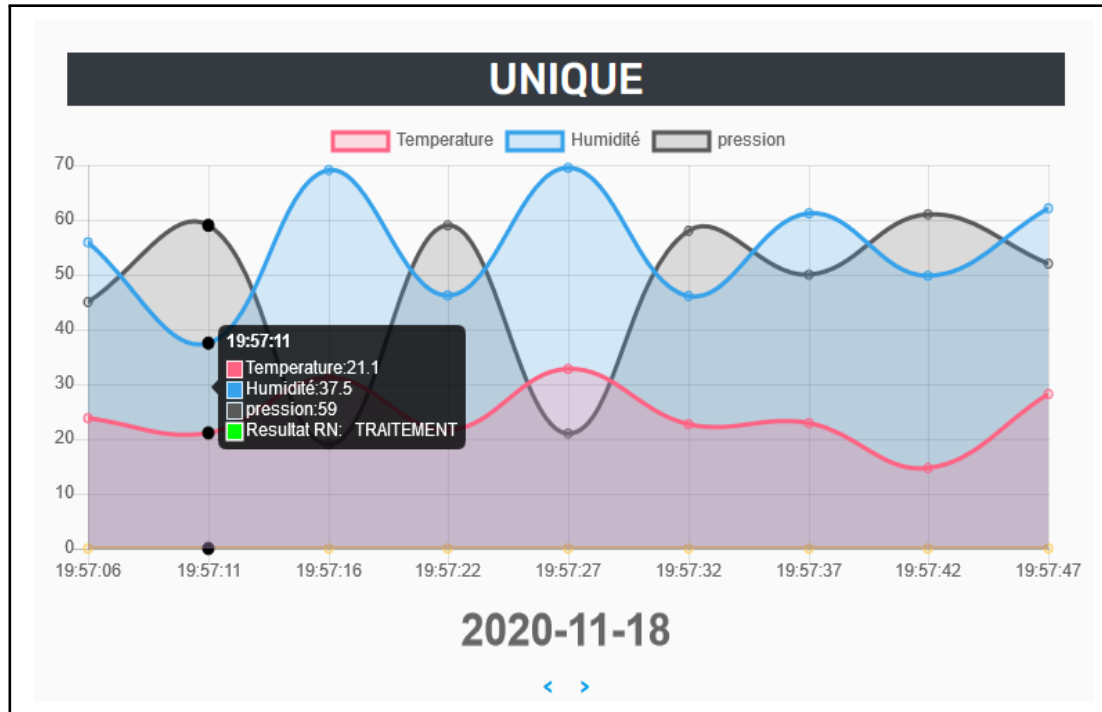


FIGURE 5.11 – Graphe données capteurs avec résultat du reseau de neurone

La figure (5.12) donne l'interface qui affiche le suivi des réparations des codes de verrouillages (anomalies).

Code verrouillages:

Code:	Manipulateur:	Date detection	Reparateur	Date reparation	Etat	Rapport
ACC	hamida abderahmene	2020-11-18 19:43:12	hamida abderahmene	2020-11-18 19:43:22	REPARER	
AIR	hamida abderahmene	2020-11-18 19:43:12	hamida abderahmene	2020-11-18 19:43:22	REPARER	
BMAG	hamida abderahmene	2020-11-18 19:43:12	hamida abderahmene	2020-11-18 19:43:22	REPARER	
CLTO	hamida abderahmene	2020-11-18 19:43:12	hamida abderahmene	2020-11-18 19:43:22	REPARER	
DOS1	hamida abderahmene	2020-11-18 19:43:12			NON REPARER	

< 1/39 >

FIGURE 5.12 – Page de suivi des réparations

5.6 Présentation de l'application labview

5.6.1 introduction

Cette application consiste à faire la classification des données de capteurs en utilisant le réseau de neurone et de visualiser la décision de traitement du patient en temps réel avec les actions à prendre en cas d'anomalies.

5.6.2 Présentation de l'interface de l'application labview

La figure (5.13) présente l'interface de notre application labview (avant exécution). Cette dernière visualise :

- Le nom de la machine de traitement et l'adresse ip du pc liée virtuellement a cette machine.
- Elle visualise les données de capteurs placés dans la salle de traitement .
- Le résultat du processus manuelle .
- Le résultat du réseau de neurone des données de capteurs .
- Le résultat du système de décision de traitement.
- les actions suggéré par le système expert .

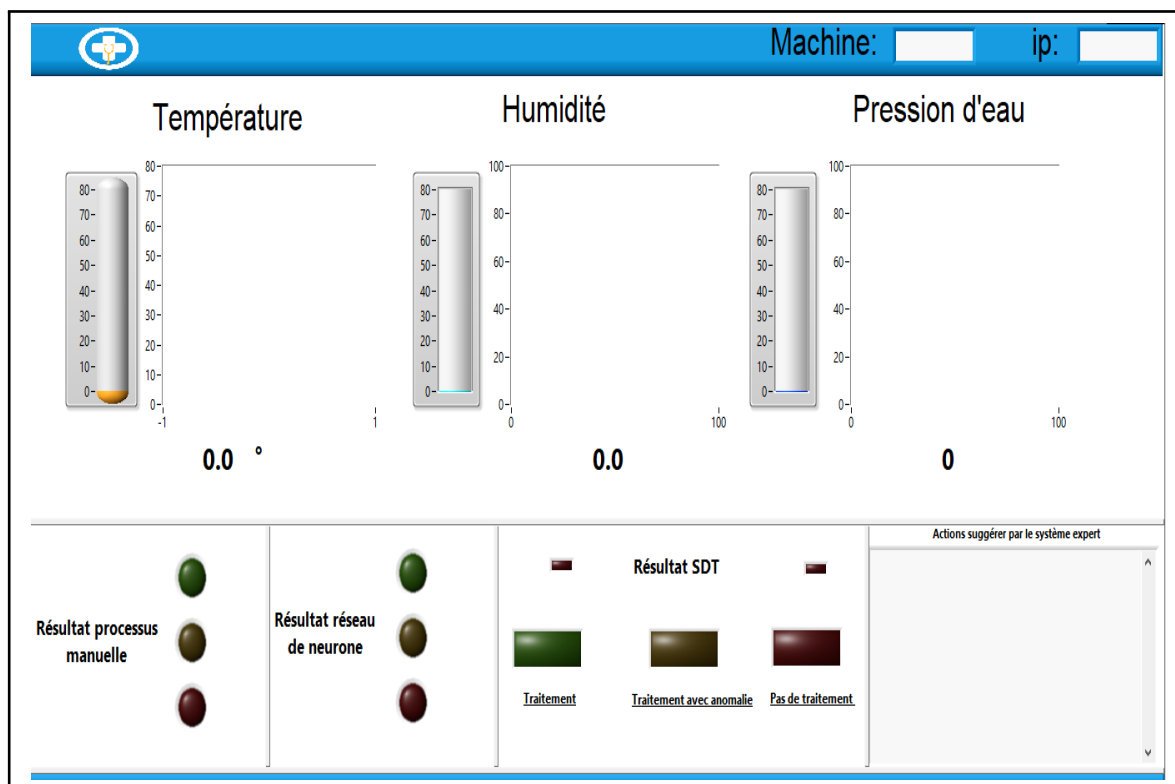


FIGURE 5.13 – Interface de l'application labview

5.6.3 Interface module de decision de traitement

La figure (5.14) visualise le résultat du module de décision de traitement en temps réel .

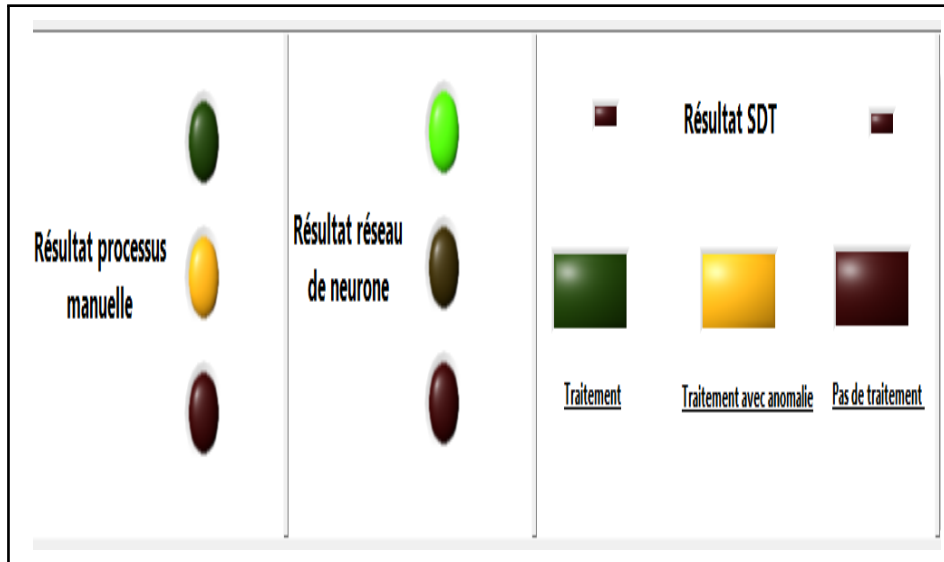


FIGURE 5.14 – Interface module de decision de traitement

5.6.4 Interface système expert

La figure (5.14) visualise les actions suggérés par le système expert envers les anomalies signalées depuis l'application web en temps réel .

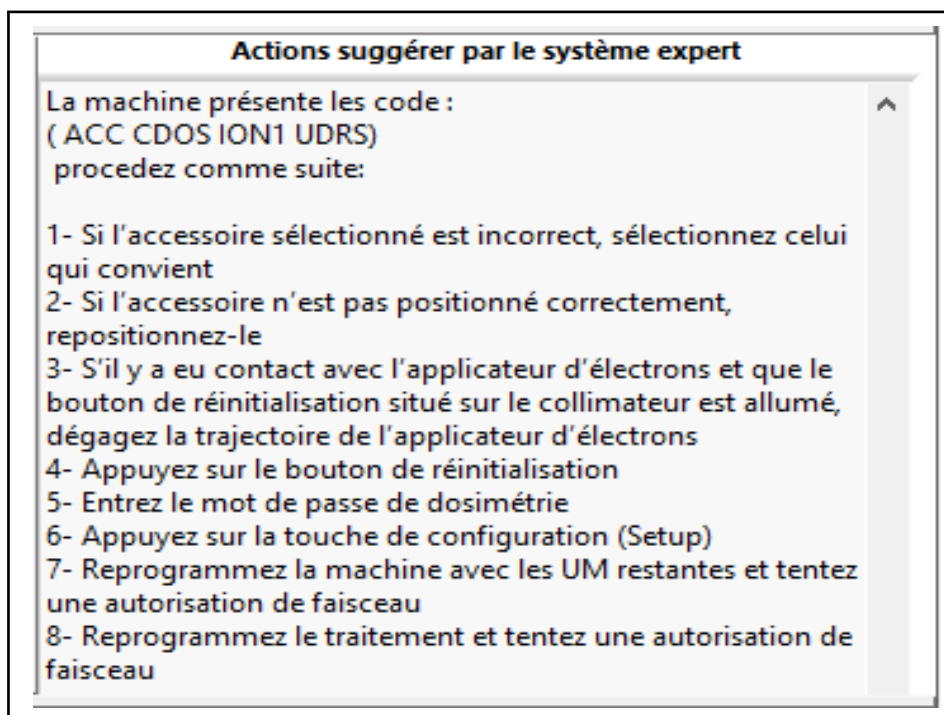


FIGURE 5.15 – Interface système expert

5.7 Conclusion

Au cours de ce chapitre, nous avons décrit toutes les étapes nécessaires de l'implémentation de notre application web y compris la présentation de l'environnement logiciel et les langages de programmation. Aussi nous avons parlé de l'application Labview qui est utilisé pour la supervision des machines de traitement. Ensuite, nous avons montré quelques captures d'écran comme un guide d'utilisateur de notre application afin d'assurer le bon fonctionnement du système.

Conclusion générale

Ce travail s'inscrit dans le cadre de la gestion et de contrôle d'anomalies dans le service de radiothérapie de l'hôpital cpmc. Dans le cadre de ce mémoire nous avons traité une problématique qui touche le monde entier même dans les pays avancés.

Pour répondre à cette problématique nous nous avons conçu et implémenté un système composé de deux parties. En effet la première partie est une application web qui sert d'interface entre le personnel et les machines de traitements. Quant à la deuxième partie comporte deux modules à savoir :

- module système de décision de traitement, ce module s'appuie partiellement sur l'apprentissage automatique pour classer les données de capteurs ainsi les anomalies signalées.
- module système expert, ce module vise à suggérer les actions à faire en cas d'anomalies présentes sur les machines de traitements.

Aujourd'hui le système de gestion et de contrôle d'anomalies est fonctionnel. Il est à présent possible de visualiser sur l'écran d'un ordinateur la liste des problèmes intervenants ainsi que les actions à prendre pour la réparation des anomalies dans le service cpmc en temps réel à l'aide d'un système expert et avoir une traçabilité de toutes les anomalies, et les enregistrements des capteurs classifiés par le réseau de neurone . Néanmoins ce système nécessite encore des améliorations, que ce soit de sécurité ou d'ajouts de nouvelles fonctionnalités. Il pourrait être intéressant de concevoir un module d'envoi via téléphone et de faire une analyse des données permet au chef de service de prendre des décisions pour :

1. Le matériel.
2. Le personnel.
3. Les patients.

De notre côté nous avons beaucoup apprécié de travailler sur ce projet, nous considérons qu'il nous a été bénéfique vu qu'il nous a permis de consolider nos connaissances vers la conception d'un système qui sera utile dans le domaine de la santé et on espère que ce projet va encore évoluer par la suite.

Annexe

Le capteur de température et d'humidité MS6503

1. Description du capteur

- Cet outil est un environnement numérique thermo-hygromètre qui est basé sur un capteur à semi-conducteur et la caractéristique capacitive d'un polymère.
- Cet outil est pour des services professionnels avec grand écran LCD rétro éclairé.
- Cet instrument est interfacé, via le logiciel, avec un PC afin que vous puissiez enregistrer une grande quantité de données avec le traitement ultérieur.
- Cet instrument est équipé du "HOLD".
- Cet outil permet l'identification de la MAX / MIN.
- Pendant l'utilisation peut changer l'unité de mesure de la chaleur.
- Cet instrument est équipé de la mise hors tension automatique (de mise hors tension automatique).

2. Nomenclature des composants

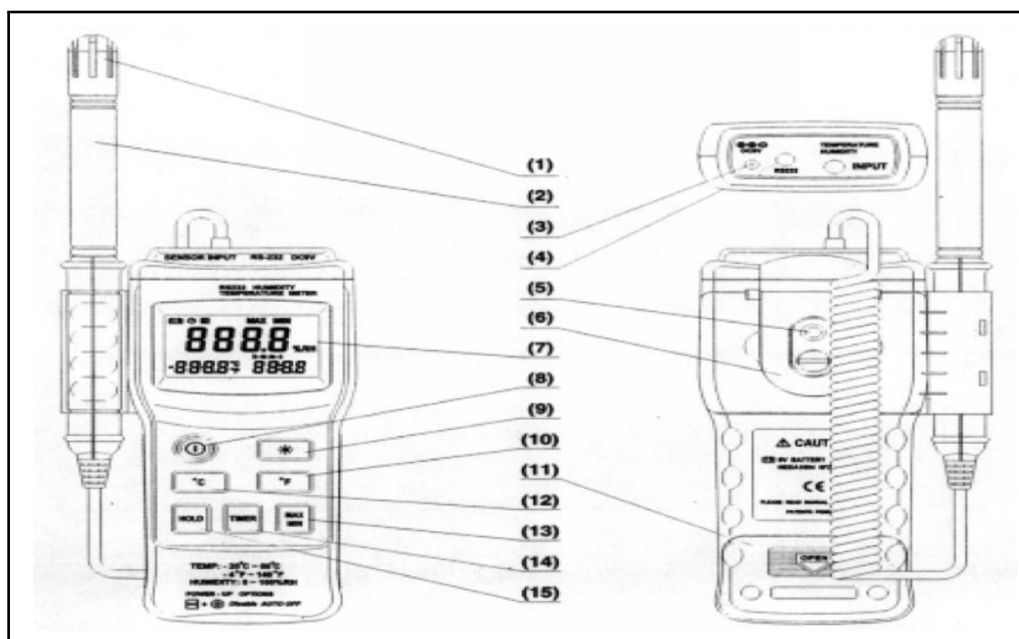


FIGURE 5.16 – Description des composants de capteur MS6503.

- (1) Capteur de protection anti-poussièr.
- (2)le support de capteur de la sonde.
- (3)Branchez l'AC.
- (4) sortie connecteur RS-232.
- (5) Connecteur pour support optionnel.
- (6) Tab benne pour accrocher l'outil.
- (7) Ecran LCD.
- (8) Boutton "ON / OFF".
- (9) Bouton pour rétro éclairage.
- (10) Bouton "° F " à l'échelle en degrés Fahrenheit.
- (11) de la housse de siège à partir de la batterie de 9V 6LR61.
- (12) Bouton "° C" à l'échelle en degrés centigrades.
- (13)" MAX / MIN " valeurs de contrôle de bouton MAX, MIN.
- (14) touche "Timer" pour afficher la minuterie. (15) Touche "HOLD" la fonction HOLD.

3. Mode d'utilisation du capteur MS6503

Le capteur est programmé de façon à s'éteindre automatiquement au bout de 30 minutes de marche si on l'allume en appuyant seulement sur le bouton Power.



FIGURE 5.17 – Vue du capteur MS6503 avec les 2 principaux boutons à manipuler (Power et Hold).

4. Configuration du capteur MS6503 dans les 2 modes de fonctionnement

Continu et Extinction automatique

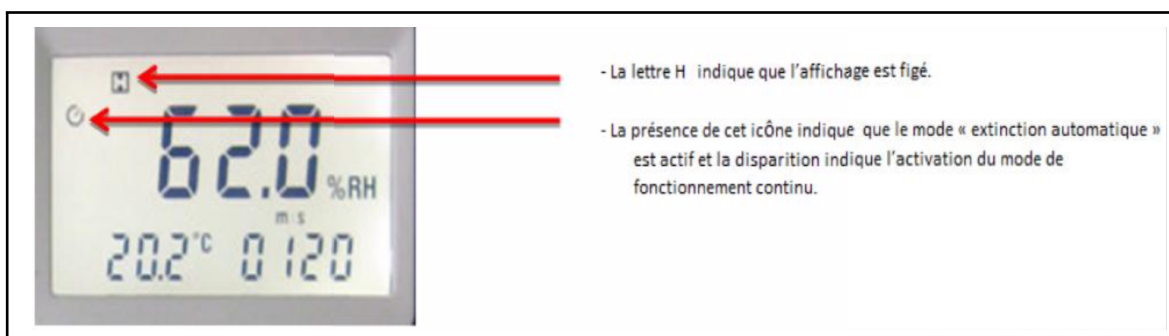


FIGURE 5.18 – Mode de fonctionnement du capteur en mode extinction automatique du mode de fonctionnement continu.

5. Sélection de la grille de température

Lorsque l'appareil est allumé, l'échelle par défaut est définie sur Celsius (° C). Pour activer la température Fahrenheit F, appuyez sur "F", et pour revenir à la température Celsius (C), appuyez sur "C".

Configuration de ODBC

1. Téléchargement et installation du pilote Mysql Connector/ODBC :

- consulter cette page : <https://dev.mysql.com/downloads/connector/odbc/> et télécharger la dernière version de Mysql Connector.
- lancer le programme et suivez les étapes d'installation.

2. Ajouter une entrée ODBC :

- Ouvrir le panneau de configuration et sélectionner « Outils d'administration »

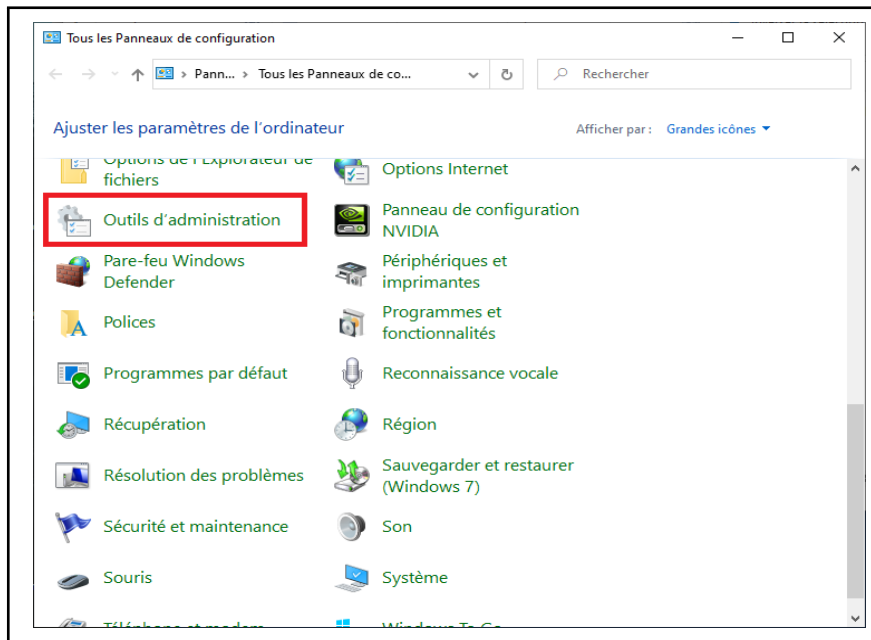


FIGURE 5.19 – fenêtre « panneau de configuration ».

- sélectionner « ODBC Data Sources »

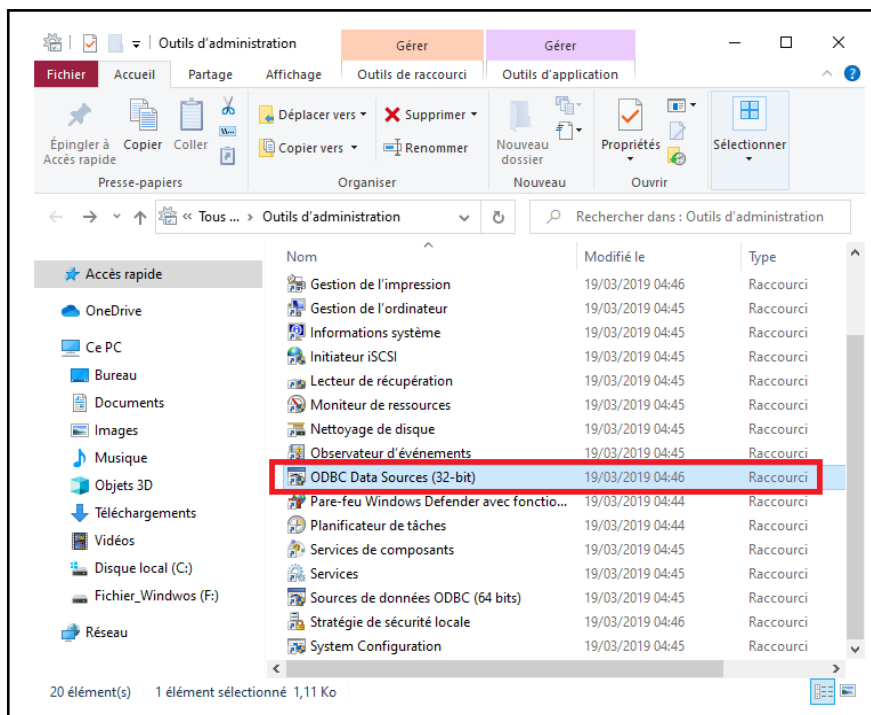


FIGURE 5.20 – fenêtre « Outils d'administrateur ».

- cliquer sur ajouter puis sélectionner MySQL ODBC Unicode Driver

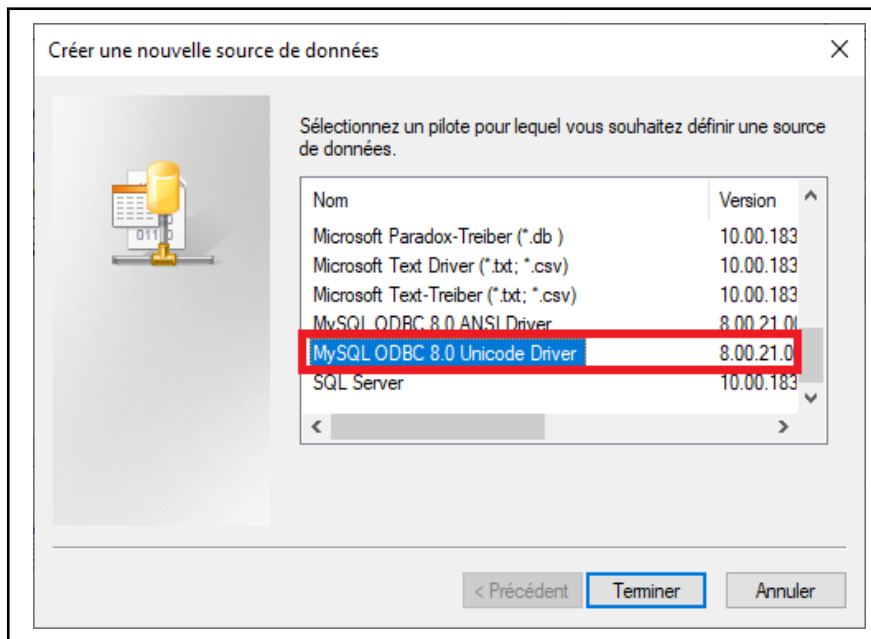


FIGURE 5.21 – fenêtre « ODBC ».

- inscrivez les information sur la base de données

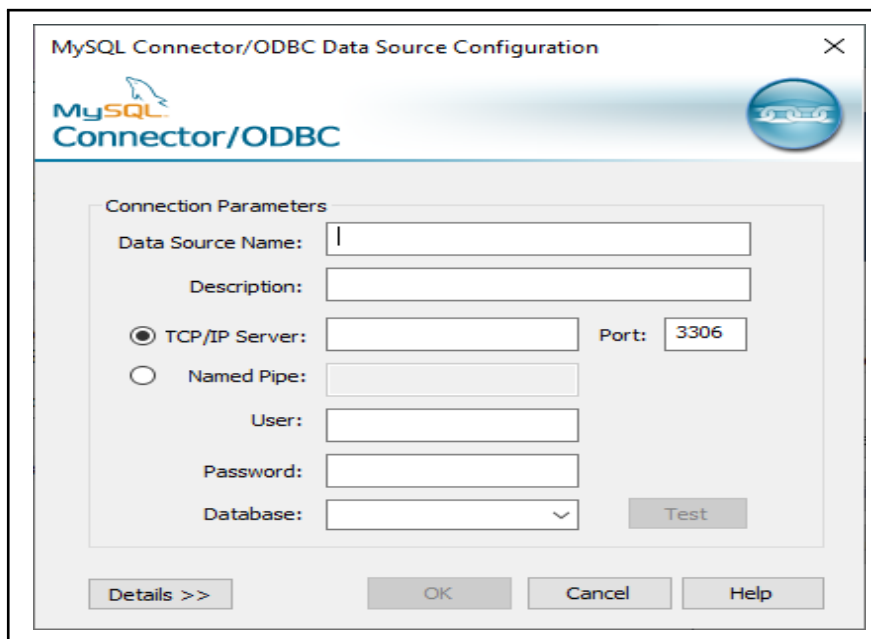


FIGURE 5.22 – fenêtre «configuration ODBC ».

Présentation diagramme de bloc de l'application labview

la figure (5.23) montre comment établir une connexion avec la base de données en utilisant le toolkit database connectivity labview.

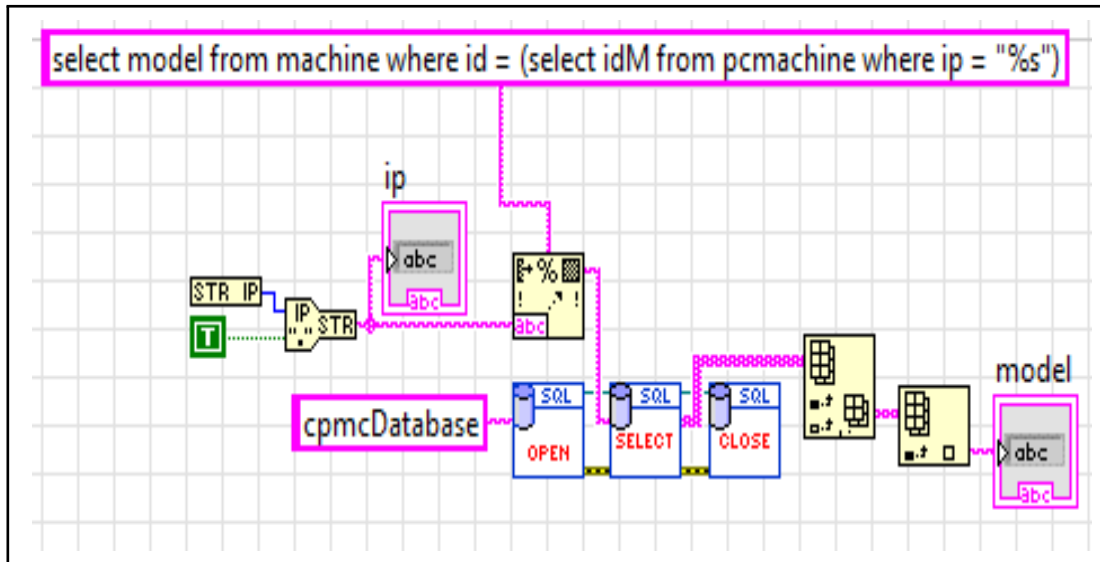


FIGURE 5.23 – Le diagramme de bloc connexion bd labview

la figure (5.24) montre une partie des blocs de construction de notre application labview.

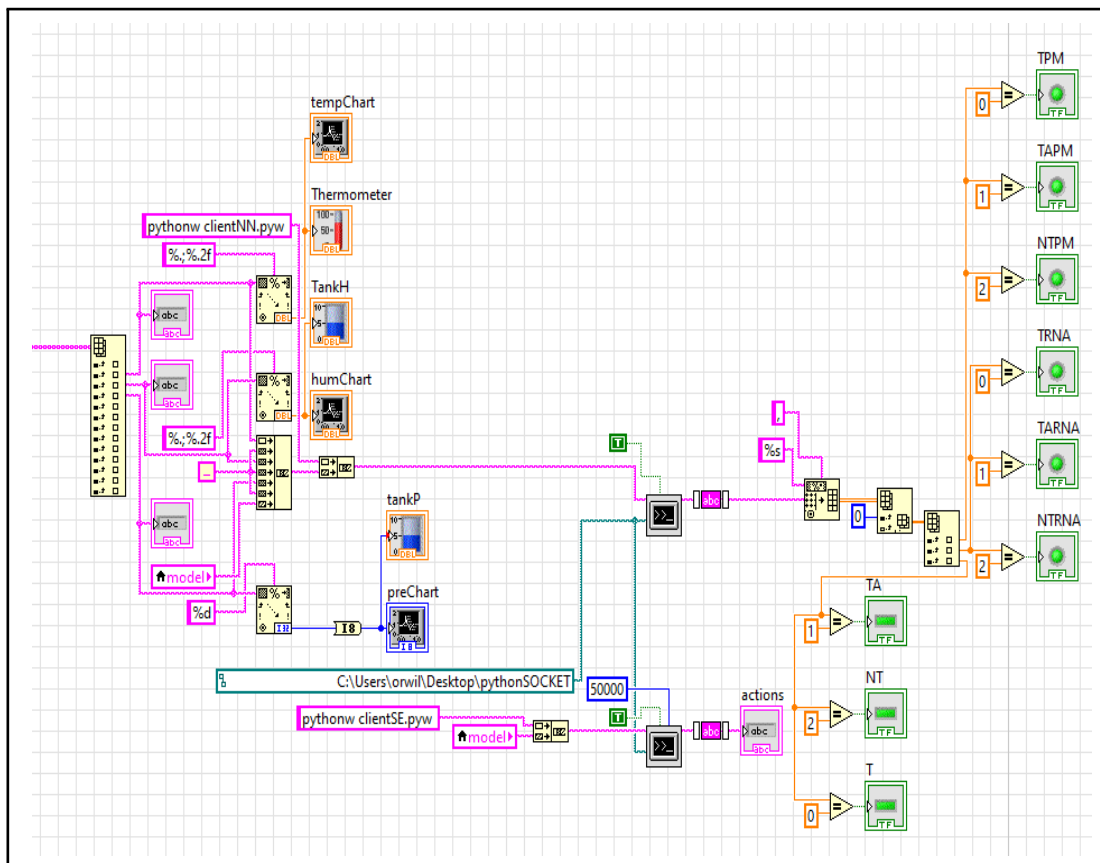


FIGURE 5.24 – Le diagramme de bloc de l'application labview

Code python d'apprentissage automatique

les figures (5.25 et 5.26) montrent le code python d'apprentissage pour la classification des données de capteurs.

```
1  from datetime import datetime
2  import os
3  os.environ["TF_CPP_MIN_LOG_LEVEL"] = "2"
4  import tensorflow as tf
5  from tensorflow import keras
6  import numpy as np
7  import matplotlib.pyplot as plt
8  from tensorflow.keras.callbacks import TensorBoard
9  import time
10 import csv
11 NAME = "capteurs_{}".format(int(time.time()))
12 tf_callback = TensorBoard(log_dir='logs/{}'.format(NAME), histogram_freq=1)
13 #CONSTRUCT TRAIN DATAS
14 train_datas = []
15 train_labels = []
16 with open("train_datas.csv") as csvfile:
17     reader = csv.reader(csvfile, delimiter=',')
18     for row in reader:
19         train_datas.append([])
20         train_labels.append([])
21         for key, val in enumerate(row):
22             if key != 0 and key != 4:
23                 train_datas[-1].append(float(val))
24             elif key == 4:
25                 train_labels[-1].append(int(val))
26 train_datas = np.array(train_datas)
27 train_labels = np.array(train_labels)
28 #CONSTRUCT TEST DATAS
29 test_datas = []
30 test_labels = []
31 with open("test_datas.csv") as csvfile:
32     reader = csv.reader(csvfile, delimiter=',')
33     for row in reader:
34         test_datas.append([])
35         test_labels.append([])
36         for key, val in enumerate(row):
37             if key != 0 and key != 4:
38                 test_datas[-1].append(float(val))
39             elif key == 4:
40                 test_labels[-1].append(int(val))
41 test_datas = np.array(test_datas)
42 test_labels = np.array(test_labels)
43
```

FIGURE 5.25 – Algorithme réseau de neurone 1.

```
43
44 # CREATE A MODEL
45 model = keras.Sequential([
46     keras.layers.Dense(3),
47     keras.layers.Dense(15, activation='sigmoid'),
48     keras.layers.Dense(15, activation='sigmoid'),
49     keras.layers.Dense(3, activation='softmax')
50 ])
51
52 #compiler model
53 model.compile(optimizer='adam',
54               loss="sparse_categorical_crossentropy",
55               metrics=['accuracy'])
56 #TRAIN THE MODEL
57 model.fit(train_datas, train_labels,
58         validation_data=(test_datas, test_labels), epochs=500, callbacks=[tf_callback])
59
60 #EVALUATE THE MODEL
61 test_loss, test_acc = model.evaluate(test_datas, test_labels)
62 print('\n test accuracy :', test_acc, ' || test loss :', test_loss)
63
64
65 #SAVE THE MODEL
66 model.save("modelCapteurs.h5")
67
```

FIGURE 5.26 – Algorithme réseau de neurone 2.

Bibliographie

- [1] V. Perrin . *L'intelligence artificielle*. conférence-Débat, Jeudi 18 octobre 2018, Edition Saint-Die. .
- [2] A. Neumann Docteur de l'Ecole Nationale des Ponts et Chaussées. *Introduction d'outils de l'intelligence artificielle dans la prévision de pluie par radar*. thèse doctorale : Soutenue le 23 Novembre 2007.
- [3] <https://www.futura-sciences.com/tech/definitions/informatique-intelligence-artificielle-555/>. Consulté le 09-06-2020.
- [4] L. Audibert Université Paris 13 – Laboratoire d'Informatique de Paris-Nord (LIPN). 4 novembre 2010
- [5] <https://www.nouvelobs.com/societe/20180228.OBS2855/dans-ces-8-domaines-l-intelligence-artificielle-va-changer-nos-vies.html>. Consulté le 15-06-2020.
- [6] <http://intelart.e-monsite.com/pages/ii-l-utilisation-de-l-ia/domaines-d-applications.html>. Consulté le 15-06-2020.
- [7] J.P. Rennard . *Réseaux neuronaux : Une introduction accompagnée d'un modèle Java*. 13 février 2006.
- [8] <https://lte.ma/quelques-exemples-dapplication-de-lintelligence-artificielle>. Consulté le 09-06-2020.
- [9] <https://www.ilv.fr/les-avantages-de-lintelligence-artificielle/>. Consulté le 12-06-2020.
- [10] <https://www.axiocode.com/avantages-inconvenients-intelligence-artificielle/>. Consulté le 13-06-2020.
- [11] J.R. Jang, G. T. Sun et E. Mizutani. *Neuro-Fuzzy and Computing*,. Englewood Cliffs, NJ : Prentice-Halle, 1997.
- [12] https://www.academia.edu/37046526/Les_Réseaux_de_Neurones_Artificiels. Consulté le 14-06-2020.
- [13] C. Touzet. *Les réseaux de neurones artificiels introduction au connexionnisme*, cours, exercices et travaux pratiques, Juillet 1992.

- [14] G. N. Messelem Omar *Influence du dopage sur les performances électrique d'un transistor CNTFET modélisée par les réseaux de neurones*, ingénieur, L'arbi Ben Mhidi, 2011.pp.39.
- [15] P. Campolucci *A circuit Theory Approach to Recurrent Neural Network Architecture and Learning Methods*,These Doctorat, Université de Bologne, Février 1998.
- [16] Y. Hachemi Imad. *Optimisation d'entraînement des réseaux de neurone*, L'arbi Ben Mhidi, 2010-2011.
- [17] G. Dreyfus, J. Martinez, M. Samuelides, M. Gordon, F. Badran, S. thiria et L.Herault. *réseaux de neurones, méthodologies et application* edition Eyrolles.2002.
- [18] C. Pomerol , G. Benchimol, P. Levine .*Developing Expert Systems"*. (décembre 1987).
- [19] G.F. Luger and W.A. Stubblefield.*Artificial Intelligence : Structures and Strategies for Complex Problem Solving*. 1999.
- [20] A. Martel.*MPI-2 les systèmes experts*. Romain Bouleis Université de Savoie. 2007.
- [21] https://perso.liris.cnrs.fr/alain.mille/enseignements/DEA-ECD/site_ia_emiage/Session2/session2.htm#\T1\textsection2.Consulté le 22-06-2020.
- [22] F. Denis. *Intelligence artificielle : les systèmes experts* . Laboratoire d'Informatique fondamentale, Marseille Laurent Miclet .Juillet 2009.
- [23] D. Pastre. *livre L'intelligence artificielle définition - historique - domaines* .Édition Fayard 2005.
- [24] J.L. Crowley. *Conception de systèmes intelligents : Programmation des Systèmes Experts* . 2001-2002.
- [25] https://fr.wikipedia.org/wiki/Réseau_sémantique.Consulté le 24-06-2020.
- [26] <https://www.enseignement.polytechnique.fr/informatique/INF423/uploads/Main/chap3-good.pdf>.Consulté le 24-06-2020.
- [27] <http://www.linguist.univ-paris-diderot.fr/~amsili/Ens06/mainBx.pdf>.Consulté le 25-06-2020.
- [28] P. Jackson *Introduction to Expert Systems*. Addition-wesley, 1986.
- [29] <http://www.bruno-garcia.net/www/Cours/introobj.html> .Consulté le 30-06-2020.
- [30] D. Ben ayed . *Intelligence Artificielle-systèmes experts et moteurs d'inférence*Institut supérieur informatique.

- [31] <https://sites.google.com/site/iadescartes/2-les-systemes-experts/2-1-les-syt/2-1-3-le-moteur-d-inference-un-systeme-en-deux-phases>. Consulté le 30-06-2020.
- [32] N. Chenfour. *Systèmes experts et moteurs d'inférence*. Université Sidi Mohamed Ben Abdellah, Fès.
- [33] https://www.memoireonline.com/09/10/3851/m_Realisation-dun-systeme-expert-pour-la-therapeutique-et-le-diagnostic-des-maladies-de-la-tuber19.html. Consulté le 05-07-2020.
- [34] Y. Jiang. *Prediction of monthly mean daily diffuse solar radiation using artificial neural networks and comparison with other empirical models*. Energy Policy, (2008).
- [35] G. Joseph. *Merise et UML pour la modélisation des systèmes d'information*. Editeur Dunod, 11/03/2004, Nb de pages 289p.
- [36] R. Pascal. *UML2 EN ACTION*. 4ème édition, Eyrolles 01/03/2007.
- [37] https://fr.wikiversity.org/wiki/Combinatoire/Combinaisons_sans_répétition. Consulté le 05-11-2020.
- [38] <https://www.php.net>. Consulté le 05-11-2020.
- [39] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203257-html5-hypertext-markup-langage5-definition-traduction/>. Consulté le 05-11-2020.
- [40] http://www.standard-du-web.com/feuilles_de_style_en_cascade.php. Consulté le 05-11-2020.
- [41] https://programmation.developpez.com/langages/?page=populaires_web. Consulté le 05-11-2020.
- [42] <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203617-ajax-asynchronous-javascript-and-xml-definition-traduction/>. Consulté le 05-11-2020.
- [43] <http://perso.univ-lemans.fr/~cpiau/BD/SQL.html>. Consulté le 05-11-2020.
- [44] <https://www.ni.com/en-lb/shop/labview.html>. Consulté le 06-11-2020.
- [45] <https://www.ni.com/en-lb/support/downloads/software-products/download.labview-database-connectivity-toolkit.html>. Consulté le 06-11-2020.
- [46] <https://www.lebigdata.fr/python-langage-definition>. Consulté le 06-11-2020.
- [47] <https://www.tensorflow.org/?hl=fr>. Consulté le 06-11-2020.
- [48] <https://intelligence-artificielle.agency/numpy/>. Consulté le 06-11-2020.

- [49] https://dvillers.umons.ac.be/wiki/teaching:progappchim:matplotlib_simple. Consulté le 06-11-2020.
- [50] <https://code.visualstudio.com>. Consulté le 07-11-2020.
- [51] <https://www.wampserver.com>. Consulté le 07-11-2020.
- [52] https://www.memoireonline.com/10/17/10090/m_Conception-et-implementation-d-un-site-web-pour-l-archivage-et-la-commande-en-ligne-des-journaux-da22.html. Consulté le 07-11-2020.