

République Algérienne Démocratique et Populaire
Ministre de L'Enseignement Supérieur et de la Recherche
Scientifique



Université Abderrahmane MIRA-BÉJAÏA
Faculté des Sciences Exactes
Département d'Informatique

En vue de l'obtention du diplôme de Master Professionnel en Informatique option Génie
Logiciel

THÈME :

*Conception et réalisation d'un système
d'analyse et de prédiction de ventes
(cas : Sonelgaz Distribution Direction de BÉJAÏA)*

Réalisé par :

M^{elle} AIROUCHE Kafia
M^{elle} ALOUANE Kahina

Devant un jury composé de :

Président : M. BEDJOU Khaled	MAA	Université A MIRA-BÉJAÏA
Examineur : M. DJEBARI Nabil	MCB	Université A MIRA-BÉJAÏA
Encadrant : M. MIR Foudil	MAA	Université A MIRA-BÉJAÏA

Promotion 2021-2022

Remerciements

Au terme de notre travail, nous remercions Dieu le tout puissant de nous avoir donné la santé, le courage et la patience de réaliser ce modeste travail.

Tout d'abord, ce mémoire est le fruit des efforts fournis et des sacrifices consentis par plusieurs personnes, sans lesquelles ce travail n'aurait pas pu aboutir.

Nos remerciements vont particulièrement à notre encadrant monsieur **MIR Foudil**, pour nous avoir orienté durant l'élaboration de ce travail. Il a toujours été disponible à l'écoute de nos nombreuses questions.

Notre reconnaissance et notre estime sont également portés à **Mr. HADJOUT Dalil** responsable au service statistiques pour sa disponibilité et surtout ces judicieux conseils et pour tout le temps qu'ils nous a consacré. Ainsi que, tout le personnel de la **Sonelgaz Distribution Direction de BÉJAIA** .

Nous tenons à remercier également les membres du jury qui nous font le grand honneur d'avoir accepté de juger ce travail.

Enfin, nos vifs remerciements s'adressent à toutes nos familles qui nous ont soutenus et encouragés durant toute notre vie et pendant notre cursus d'étude, également à toutes les personnes qui ont contribué de près ou de loin à la réalisation de ce modeste travail.

AIROUCHE Kafia

ALOUANE Kahina

Dédicaces

“

Je dédie ce modeste travail aux personnes chères à mon cœur,

*À ma très chère mère Yamina et mon très cher père Hamid,
pour tous leurs sacrifices, leur amour, leur tendresse, leur
soutien et leurs prières tout au long de mes études, pour tout
ce que vous avez fait pour moi.*

*Que dieu, le tout puissant, vous préserve et vous procure
santé et longue vie afin que je puisse à mon tour vous
combler.*

À la mémoire de mes grands-parents.

À mon frère Arezki.

À mes sœurs Rekia, Sonia et Cylia.

À ma tante Khadidja.

À mes amis : Kahina, Kenza, Yacine.

*À tous ceux qui sont chers, proches de mon cœur et à tous
ceux qui m'aiment et qui voudraient partager ma joie.*

Merci d'être toujours là pour moi.

”

- Kafia

Dédicaces

“

Je dédie ce modeste travail,

À mes très chers parents

*Je vous remercie infiniment pour l'éducation que vous m'avez apporté,
pour les sacrifices, l'amour, la tendresse que vous m'avez donné pour
avoir toujours cru en moi et m'avoir soutenu durant tout mon cursus de
formation.*

*Que Dieu vous préserve et vous procure santé et longue vie afin que je
puisse à mon tour vous récompenser.*

À mon cher frère Nabil

*Ma profonde estime pour l'aide que tu m'as apportée. Tu m'as toujours
soutenu, réconforté et encouragé.*

À ma chère petite sœur Adorable Asma

À mes cousines

Silia, Dihia, Ryma, Linda, Kenza, Hmila et Lina.

À mes tantes

Saida et Rachida.

À ma grand mère Akria à qui je souhaite une bonne santé.

À la mémoire de mes grands-pères et mon oncle

À mes amis

Kafia, Kenza, Silia, Lounja et Soria.

*À tous ceux qui sont chers, proches de mon cœur et à tous ceux qui
m'aiment et qui voudraient partager ma joie.*

”

- **Kahina**

TABLE DES MATIÈRES

Table des figures	viii
Liste des tableaux	x
Liste des abréviations	xi
Introduction générale	1
1 Etat de l'art	3
1.1 Introduction	3
1.2 Séries temporelles	4
1.2.1 Définition	4
1.2.2 Composantes d'une Série Chronologique	4
1.2.3 Types des schémas de série temporelle	4
1.2.4 Types séries temporelles	6
1.2.5 Quelques modèles des séries temporelles	6
1.3 Machine Learning	8
1.3.1 Types d'apprentissage automatique	9
1.3.1.1 Apprentissage supervisé	9
1.3.1.2 Apprentissage non supervisé	9
1.3.1.3 Apprentissage semi supervisé	9

1.3.1.4	Apprentissage par renforcement	10
1.3.2	Modèles d'apprentissage automatique	10
1.3.2.1	Support Vector Machine (SVM)	10
1.3.2.2	K plus proche voisins	10
1.3.2.3	Régression linéaire	10
1.3.2.4	Les Forêts Aléatoires (Random Forest)	11
1.4	Deep Learning	11
1.4.1	Neurone artificiel	11
1.4.2	Perceptron	11
1.4.2.1	Fonctionnement	12
1.4.3	Réseau de neurone	12
1.4.4	Les types de réseaux neuronaux	13
1.4.4.1	Les réseaux de neurones feed-forwarded	13
1.4.4.2	Les réseaux neuronaux récurrents	13
1.5	Marché de l'électricité	16
1.5.1	Analyse des ventes	16
1.5.2	Prévision des ventes	17
1.6	Gestion de projet	17
1.7	Les méthodes agiles	18
1.7.1	Les principales méthodes agiles	18
1.7.1.1	XP (eXtreme Programming)	18
1.7.1.2	RAD (Rapid Application Development)	19
1.7.1.3	Scrum	19
1.8	Application web	21
1.8.1	Architecture des applications web	21
1.9	Conclusion	22
2	Analyse des besoins et conception	23
2.1	Introduction	23
2.2	Présentation du groupe SONELGAZ	23
2.3	Les différents types de client Sonelgaz	24

2.4	Présentation de l'organisme d'accueil (<i>Sonelgaz Distribution Direction de BÉJAIA</i>)	25
2.4.1	Organisation de la Sonelgaz Distribution Direction de BÉJAIA	25
2.4.1.1	Division relations commerciales	26
2.5	Problématique	27
2.6	Solution proposée et objectif	27
2.7	Délimitation du domaine d'étude	28
2.7.1	Les rôles scrum	28
2.7.2	Identification des acteurs	29
2.7.3	Diagramme de contexte	29
2.8	Pilotage de notre projet avec SCRUM	30
2.8.1	Backlog produit	31
2.8.2	Planification des sprints	32
2.9	Diagramme de cas d'utilisation	33
2.10	Description des cas d'utilisation	35
2.10.1	Cas d'utilisation «S'authentifier»	36
2.10.2	Cas d'utilisation «Visualiser les consommations»	37
2.10.3	Cas d'utilisation «Prédire les ventes»	39
2.11	Diagramme séquence détaillé	41
2.11.1	Diagramme de séquence détaillé du cas d'utilisation «S'authentifier »	42
2.11.2	Diagramme de séquence détaillé du cas d'utilisation « Visualiser les consommations»	43
2.11.3	Diagramme séquence détaillé du cas d'utilisation « Prédire les ventes »	44
2.12	Diagramme de navigation	45
2.13	Diagramme de classe	47
2.14	Schéma de Base de donnée NoSQL	48
2.15	Conclusion	49
3	Réalisation	50
3.1	Introduction	50
3.2	Environnement de développement de l'application	50
3.2.1	Architectures	50
3.2.2	Matériel	52

3.2.3	Langages utilisés	52
3.2.4	Frameworks	53
3.2.5	Logiciels	53
3.2.6	Librairies Python	54
3.3	Modélisation de la prédiction de vente	55
3.3.1	Modélisation de la méthode SARIMA	56
3.3.2	Modélisation de la méthode LSTM	59
3.4	Présentation de notre application	62
3.4.1	Application web	62
3.4.2	Quelques interfaces graphiques	63
3.4.2.1	Interface d'authentification	63
3.4.2.2	Interface de tableau de bord	64
3.4.2.3	Interface Utilisateurs	64
3.4.2.4	Interface visualisation des ventes	65
3.4.2.5	Interface localisation des clients	66
3.4.2.6	Interface prédiction des ventes	66
3.5	Conclusion	68
	Conclusion générale et perspectives	69
	Bibliographie	70

TABLE DES FIGURES

1.1	Représentation graphique d'un schéma additif	5
1.2	Représentation graphique d'un schéma multiplicatif	5
1.3	Représentation graphique d'un schéma mixte	6
1.4	Schéma d'un neurone.	11
1.5	Schéma d'un perceptron.	12
1.6	Schéma réseau de neurone.	13
1.7	Architecture long et court terme mémoire.	14
1.8	Architecture Unité récurrente à grille.	15
2.1	Organigramme de la Sonelgaz Distribution Direction de BÉJAIA	25
2.2	Diagramme de contexte.	29
2.3	Diagramme de cas d'utilisation.	34
2.4	Diagramme de séquence système du cas d'utilisation « S'authentifier ».	37
2.5	Diagramme de séquence système du cas d'utilisation « Visualiser les consommations ».	39
2.6	Diagramme de séquence système du cas d'utilisation « Prédire les ventes ».	41
2.7	Diagramme de séquence détaillée du cas d'utilisation « S'authentifier ».	43
2.8	Diagramme séquence détaillé du cas d'utilisation « Visualiser les consommations ».	44
2.9	Diagramme séquence détaillé du cas d'utilisation « Prédire les ventes ».	45
2.10	Diagramme de navigation.	46
2.11	Diagramme de classes.	47

2.12	Diagramme de classes.	47
2.13	schéma BDD	49
3.1	L'architecture MVT.	51
3.2	Les composantes de la série temporelle du client 967.	57
3.3	Les diagnostics du modèle sarima-Histogram plus estimated density.	58
3.4	Les diagnostics du modèle sarima-Graphique Q-Q.	58
3.5	Les diagnostics du modèle sarima-Standardized residual.	58
3.6	Les diagnostics du modèle sarima-Correlogram.	59
3.7	Interface Authentification.	63
3.8	Interface Tableau de bord.	64
3.9	Interface Utilisateurs.	64
3.10	Interface visualisation des ventes-Tableau.	65
3.11	Interface visualisation des ventes-Graphe.	65
3.12	Interface localisation des clients.	66
3.13	Interface prédiction des ventes -SARIMA.	67
3.14	Interface prédiction des ventes -LSTM.	67

LISTE DES TABLEAUX

1.1	Tableau comparatif des modèles des séries temporelles.	8
1.2	Tableau comparatif des méthodes agile.	20
2.1	Messages échangés entre le système et les acteurs	30
2.2	Backlog produit.	32
2.3	Planification des Sprints.	33
2.4	Formalisme de description des cas d'utilisations.	35
2.5	Description du cas d'utilisation «S'authentifier».	36
2.6	Description du cas d'utilisation «Visualiser les consommations».	38
2.7	Description du cas d'utilisation «Prédire les ventes».	40
2.8	Description UML du diagramme de séquence détaillé.	42
2.9	Dictionnaire de données.	48
3.1	Tableau des résultats LSTM.	62

Liste des abréviations

AIC : Akaike Information Criterion
AR : AutoRégressive
ARIMA : AutoRegressive Integrated Moving Average
ARMA : AutoRegressive Moving Average
BDD : Base de données
BP : Basse Pression
BT : Basse Tension
CSS : Cascading Style Sheets
DL : Deep Learning
DRC : Division Relation Commercial
GRU : Gated Recurrent Unit
HP : Haute Pression
HT : Haute Tension
HTA : Haute Tension A
HTB : Haute Tension B
HTML : HyperText Markup Language
HTTP : HyperText Transfer Protocol
KDE : Kernel Density Estimation
K-NN : K-Nearest Neighbor
LSTM : Long Short Term Memory
MA : Moving Average
ML : Machine Learning
MT : Moyenne Tension
MVT : Modèle-Vue-Template
ODM : Object Data Manager
RAD : Rapid Application Development
RNN : Recurent Neural Network
SARIMA : Seasonal AutoRégressiveIntegrated Moving Average
SARMA : Seasonal AutoRégressive Moving Average
SGC : Service Grand Compte
SVM : Support Vector Machine
UML : Unified Modeling Language
URL : Uniform Resource Locator
WSCI : Web Server Getway Interface
XP :Extreme Programming

Introduction générale

La prévision de la production électrique est essentielle de nos jours pour des raisons économiques et environnementales. Étant que l'électricité est une ressource non stockable, une production proportionnée à la demande actuelle doit être satisfaite afin d'atteindre un équilibre entre la production et la consommation, cela limitera les pertes et l'approvisionnement continu des consommateurs sera assuré. Grâce à des outils de prédictions, on peut s'assurer que l'énergie nécessaire à la production est connue a priori. Ces outils se basent sur des approches statistiques et d'Intelligence Artificielle reconnues dans le domaine de l'analyse des séries temporelles pour prédire les valeurs ultérieures de cette dernière. Avec l'évolution de l'informatique et l'introduction de plusieurs approches intelligentes, le machine learning et deep learning ont marqué leur puissance dans le domaine de prévision.

La performance est aujourd'hui vue comme l'atteinte des objectifs de l'entreprise et la contribution à l'amélioration de sa valeur. Il faut donc maîtriser la manière d'agir en assemblant des outils de gestion générale et des outils de prévision en particulier. La prévision est l'un des outils importants que les dirigeants utilisent pour garder le contrôle, mais elle est souvent considérée comme l'aspect le plus problématique du contrôle de gestion, c'est-à-dire la manière dont une entreprise utilise les prévisions de ventes pour répondre à la demande, gérer sa production et orienter ses politiques commerciales, ses politiques de ressources humaines, etc.

Ainsi, l'objectif de notre projet est de développer un système d'analyse et de prévision des ventes au profit de la Sonelgaz Distribution Direction de BÉJAIA, afin de faciliter, améliorer le processus d'analyse des statistiques de ventes électriques des clients HTA (Haute Tension A) et de permettre de prévoir la consommation d'électricité de ces derniers. Étant donné que cet organisme ne dispose pas d'un outil de prévision et utilise des techniques traditionnelles (fichier excel) pour analyser ses ventes.

Dans ce mémoire nous allons utiliser deux méthodes pour la prévision de la consommation électrique : méthode statistique SARIMA (Seasonal Auto Régressive Integrated Moving Average) utilisé dans machine Learning et méthode LSTM (Long Short Term Memory) utilisé dans deep Learning pour la prévision des séries temporelles, pour la gestion de notre projet nous allons opter pour la méthode agile scrum.

Organisation du mémoire

Ce mémoire est structuré comme suit :

Dans le **premier chapitre**, nous allons donner un aperçu sur les différentes technologies utilisées dans notre projet à savoir Machine Learning, Deep Learning ainsi que l'analyse et prévision des séries temporelles. Nous étudierons également, le processus de l'analyse prédictive.

Après, nous allons définir la gestion de projet. Ensuite, nous allons exposer les différentes méthodes agiles et mettre en avant la méthode Scrum qui sera appliquée à notre projet.

Puis, dans le **deuxième chapitre** nous enchaînerons avec la présentation de l'organisme d'accueil, la problématique suivie par la solution ainsi que les objectifs de l'application à réaliser. Nous allons identifier les acteurs et les différents cas d'utilisation. De plus, ces différents cas d'utilisation seront modélisés ensuite par des diagrammes de séquences.

Par la suite, les interactions entre les objets du système seront modélisées par les diagrammes de séquences détaillés. Un diagramme de navigation sera établi pour comprendre l'architecture de notre application. À la fin de ce chapitre, nous allons présenter le diagramme de classes.

Ensuite, dans le **troisième et dernier chapitre** nous allons énumérer les étapes de la construction des deux modèles de prédiction. Enfin, nous allons expliquer l'architecture de l'application et son fonctionnement à l'aide des interfaces graphiques.

Nous terminerons ce rapport par une conclusion générale et perspectives.

1.1 Introduction

Depuis plus d'une décennie, l'intelligence artificielle (IA) vit une accélération dans son développement et son adoption. En entreprise, il est de plus en plus impliqué dans la prise de décision, ce qui est un facteur d'une importance primordiale. À mesure que les nouvelles technologies se développent, de plus en plus de décisions sont prises par des algorithmes fournis par Machine Learning et Deep Learning.

L'analyse de séries chronologiques est devenue un domaine d'intérêt récent pour l'intelligence artificielle, car des prévisions précises deviennent de plus en plus importantes dans tous les types d'industries afin de prendre des décisions plus éclairées. Aujourd'hui, la production d'énergie en général est considérée comme l'un des piliers du développement économique d'un pays, en particulier l'électricité. L'électricité est un facteur très important dans l'amélioration des conditions de vie des citoyens, étant un produit non stockable, le besoin de prévoir devient nécessaire. Pour répondre aux problématiques de qualité décisionnelle et d'analyse prédictive, certaines entreprises optent pour des solutions basées sur des systèmes intelligents.

La finalisation du projet dans les délais de livraison est le souci majeur de chaque équipe de développement d'un logiciel. L'un des problèmes les plus fréquemment affrontés lors de la construction du logiciel est la mauvaise spécification et le changement brusque des besoins. Dans cette finalité, nous adoptons la méthodologie agile pour la gestion de notre projet.

Ce chapitre est consacré à définir quelques généralités sur les différentes technologies utilisés dans notre projet ML, DL, série temporelle ainsi que la gestion de projet et ces méthodologies plus particulièrement, la méthodologie Scrum qui est appliqué durant le développement de notre application.

1.2 Séries temporelles

1.2.1 Définition

Une série temporelle (ou encore une série chronologique), noté X_t , est une suite de données (valeurs numériques) indexées par le temps. L'indice temps peut être selon les cas : la minute, l'heure, le jour, l'année etc. telle que :

$$X_t = x_t, t = 1, \dots, T, T \in \mathbb{R}, \mathbb{Z}, \mathbb{N}, \dots \quad (1.1)$$

une série temporelle peut être représentée sous format d'un graphe construit de la manière suivante :

- en abscisse le temps.
- en ordonnée la valeur de l'observation à chaque instant [17].

1.2.2 Composantes d'une Série Chronologique

On considère qu'une série chronologique X_t est la résultante des différentes composantes fondamentales [21] :

1. **La tendance (ou trend) :** notée Z_t représente l'évolution à long terme de la série étudiée. Elle traduit le comportement « moyen » de la série.
2. **La composante saisonnière (ou saisonnalité) :** notée S_t correspond à un phénomène qui se répète dans des intervalles de temps réguliers (périodiques). Elle est donc liée au rythme imposé par les saisons (toutes les 12 périodes pour des données mensuelles, toutes les 4 périodes pour des données trimestrielles).
3. **La composante résiduelle (ou bruit ou résidu) :** notée ε_t correspond à des fluctuations irrégulières, en général de faible intensité mais de nature aléatoire.

1.2.3 Types des schémas de série temporelle

une série temporelle est modélisée par un des modèles suivant :

- **Modèle additif :** modèle classique de décomposition dans le traitement des modèles d'ajustement. La variable X_t s'écrit comme la somme des trois composantes :

$$X_t = Z_t + S_t + \varepsilon_t, t = 1, \dots, T. \quad (1.2)$$

La figure 1.1 [17] représente graphiquement un schéma additive .

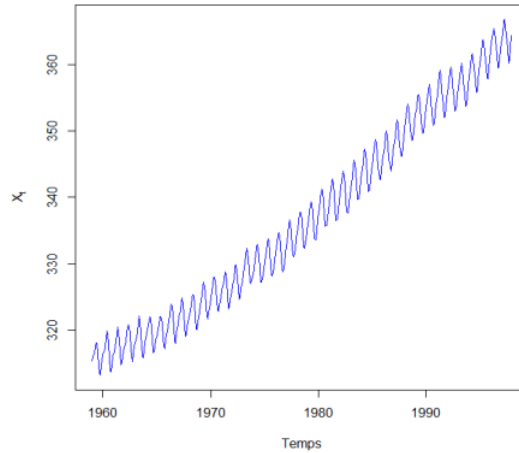


FIGURE 1.1 – Représentation graphique d'un schéma additif

- Modèle multiplicatif : la série X_t s'écrit comme le produit des composantes :

$$X_t = Z_t * S_t * \varepsilon_t, t = 1, \dots, T. \quad (1.3)$$

La figure 1.2 [17] représente graphiquement un schéma multiplicatif .

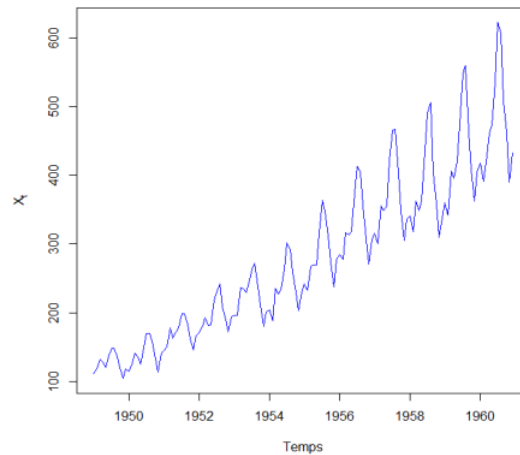


FIGURE 1.2 – Représentation graphique d'un schéma multiplicatif

- Modèle mixte : Il s'agit des différentes combinaisons de modèles additifs et de modèles multiplicatifs.

$$X_t = Z_t * S_t + \varepsilon_t \quad (1.4)$$

La figure 1.3 [17] représente graphiquement un schéma mixte .

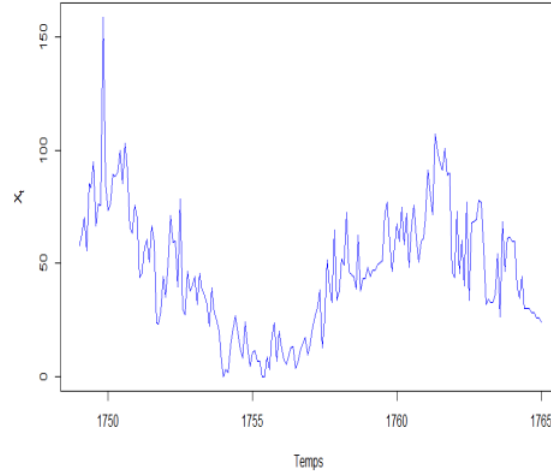


FIGURE 1.3 – Représentation graphique d'un schéma mixte

1.2.4 Types séries temporelles

- Univarié : une seule variable qui dépend du temps.
- Multivarié : plusieurs variables qui dépendent du temps.
- Stationnaire : la moyenne et la variance sont constantes dans le temps.
- Non stationnaire : la moyenne et la variance ne sont pas constantes dans le temps.

1.2.5 Quelques modèles des séries temporelles

1. Processus stationnaires

- **Processus Autorégressif $AR(p)$** : Un processus autorégressif est un modèle de régression pour séries temporelles dans lequel la série est expliquée par ses valeurs passées plutôt que par d'autres variables [1]. Un processus autorégressif d'ordre p , noté $AR(p)$ est donné par :

$$AR(p) : X_t = c + \varphi_1 X_{t-1} + \varphi_2 X_{t-2} + \cdots + \varphi_p X_{t-p} + \varepsilon_t. \quad (1.5)$$

où $\varphi_1, \dots, \varphi_p$ sont les **paramètres** du modèle, c est une constante et ε_t un bruit blanc.

- **Processus De Moyenne Mobile $MA(q)$** : Les modèles à moyenne mobile suggèrent que la série présente des fluctuations autour d'une valeur moyenne. Il est défini de manière explicite [24]. Un processus moyenne mobile d'ordre q , noté $MA(q)$ est donné par :

$$\forall t \in \mathbb{Z}, X_t = \eta_t - \theta_1 \eta_{t-1} - \cdots - \theta_q \eta_{t-q} \quad (1.6)$$

où $\theta_1, \theta_2, \dots, \theta_q$ sont des réels fixés et $(\eta_t)_{t \in \mathbb{Z}}$ est un bruit blanc de variance σ^2 .

- **Processus ARMA** : Autorégressive Moving Average ou méthode de Box-Jenkins est un processus qui permet de modéliser les séries temporelles stationnaires. Il est caractérisé par 2 paramètres :

p : ordre de la partie autorégressive (AR), l'observation courante est la combinaison linéaire de ses instants passés (1 à p).

q : ordre de la moyenne mobile(MA) [10].

- **Processus SARMA**

Les modèles SARMA sont des cas particuliers de processus stochastiques de type ARMA. L'identification des modèles SARMA $(p, q) \times (P, Q)$ est beaucoup plus compliquée que pour les ARMA(p, q) car il faut chercher toutes les combinaisons possibles $(p, q) \times (P, Q)$ qui peut-être extrêmement long en temps de calcul .

2. Processus non stationnaires

- **Processus ARIMA**

ARIMA : Autoregressive Integrated Moving Average est une extension de ARMA, Il est désigné par 3 paramètres :

d : ordre de la partie autorégressive (AR), permet d'intégrer l'effet des valeurs passées dans notre modèle.

d : ordre de la différence ou dérivation du processus, cela inclut les termes du modèle qui intègrent la quantité de différenciation (c'est-à-dire le nombre de points passés à soustraire de la valeur actuelle) à appliquer à la série chronologique.

q : ordre de la moyenne mobile(MA), cela nous permet de définir l'erreur de notre modèle comme une combinaison linéaire des valeurs d'erreur observées à des moments antérieurs dans le passé [46].

ARIMA traite les données comportant une tendance. Sa limitation est qu'elle ne prend pas en compte les séries comportant une saisonnalité.

- **Processus SARIMA**

SARIMA : Seasonal ARIMA ou ARIMA saisonnier est une extension du modèle ARIMA. Il permet de modéliser les séries temporelles comportant une composante saisonnière et désigné par 7 paramètres [10] :

p, d, q : les mêmes que ceux de ARIMA.

P : ordre de la partie autorégressive saisonnière.

D : ordre de la différence saisonnière.

Q : ordre de la moyenne mobile saisonnière.

m : la période de la composante saisonnière.

Lorsque nous traitons des effets saisonniers, nous utilisons l'ARIMA saisonnier, qui est noté $ARIMA(p,d,q)(P,D,Q)_s$. (p, d, q) les paramètres non saisonniers décrits ci-dessus, (P, D, Q) suivent la même définition mais sont appliqués à la composante saisonnière de la série chronologique. Le terme s est la périodicité de la série chronologique (4 pour des périodes trimestrielles, 12 pour des périodes mensuelles, etc.).

Le tableau 1.1 représente une comparaison entre les différents modèles de series temporelles cité ci-dessus

ARMA	SARMA	ARIMA	SARIMA
-Traite les séries temporelles stationnaires -Traite les séries comportant une moyenne mobile	-Traite les séries temporelles stationnaires -Traite les séries comportant une moyenne mobile et une saisonnalité	-Généralisation de ARMA -Traite les séries temporelles non stationnaires -Traite les séries temporelles comportant une tendance	-Généralisation de SARMA -Traite les séries temporelles non stationnaires -Traite les séries temporelles comportant une tendance et une saisonnalité

TABLE 1.1 – Tableau comparatif des modèles des séries temporelles.

Les séries temporelles utilise le machine learning et deep learning pour la prévision.

1.3 Machine Learning

Le «Machine Learning» ou « apprentissage automatique » en français est un champ d'étude de l'intelligence artificielle qui se fonde sur des approches mathématiques et statistiques pour donner aux ordinateurs la capacité d'apprendre à partir de données sans être explicitement programmés à le faire. Machine Learning est la science qui consiste à programmer des algorithmes permettant aux ordinateurs de modifier un programme qui sais rien faire au début puis apprend avec le temps et l'expérience.

Selon Fabien Benureau (2015) : "L'apprentissage est une modification d'un comportement sur la base d'une expérience " [13].

1.3.1 Types d'apprentissage automatique

L'apprentissage automatique est reparti en quatre types d'apprentissages :

1.3.1.1 Apprentissage supervisé

Un processus dans l'apprentissage automatique qui consiste à apprendre à une fonction faire des prédictions, à partir d'une liste d'exemples étiquetés, c'est-à-dire accompagnés de la valeur à prédire. Les étiquettes servent de « professeur » et supervisent l'apprentissage de l'algorithme [13]. En apprentissage supervisé, on distingue deux types de tâches :

- La classification : dans un problème de classification, on cherche à classer un objet dans différentes classes, c'est-à-dire que l'on cherche à prédire la valeur d'une variable discrète (qui ne prend qu'un nombre fini de valeurs).
- La régression : les tâches de régression se concentrent sur la recherche à prédire la valeur d'une variable continue, c'est-à-dire une variable qui peut prendre une infinité de valeurs.

1.3.1.2 Apprentissage non supervisé

Dans le cadre de l'apprentissage non supervisé, les données ne sont pas étiquetées. Il s'agit alors de modéliser les observations pour mieux les comprendre. Aucun exemple n'est requis. L'algorithme doit découvrir par lui-même la structure en fonction des données et ainsi par lui-même arriver à trouver les bons poids [13].

- Clustering : un sous-problème d'apprentissage non supervisé, est un regroupement en classes homogènes consistant à représenter un nuage des points d'un espace quelconque en un ensemble de groupes appelé Cluster. C'est un traitement sur un ensemble d'objets qui n'ont pas été étiquetés par un superviseur. L'algorithme doit découvrir par lui-même la structure en fonction des données.

1.3.1.3 Apprentissage semi supervisé

L'apprentissage semi-supervisé est le mélange des approches d'apprentissage supervisé et non supervisé. Ils classent l'ensemble de données sans étiquette à l'aide de l'ensemble de données étiquetées [13].

1.3.1.4 Apprentissage par renforcement

L'apprentissage par renforcement est un processus de Machine Learning où un logiciel apprend à effectuer une tâche à partir de ses échecs et de ses succès [13].

1.3.2 Modèles d'apprentissage automatique

L'apprentissage automatique comprends plusieurs modèles, les plus utilisées sont :

1.3.2.1 Support Vector Machine (SVM)

Les SVMs ont pour but de séparer les données en classes à l'aide d'un hyperplan, ou une frontière entre deux classes de données qui maximisera la marge entre les deux classes .Il y a plusieurs plans qui peuvent séparer les deux classes, mais un seul plan peut maximiser la marge ou la distance entre les classes [47].

1.3.2.2 K plus proche voisins

k-Nearest Neighbors en anglais (d'où l'appellation K-NN) est un algorithme de machine learning qui appartient à la classe des algorithmes d'apprentissage supervisé simple qui peut être utilisé pour résoudre les problèmes de classification et de régression. K-NN est un algorithme assez simple à appréhender car il ne va pas calculer un modèle prédictif à partir d'un training set pour effectuer une prédiction et il n'existe pas de phase d'apprentissage. K-NN stocke tout le jeu de données pour effectuer une prédiction, en calculant la similarité entre une observation en entrée et les différentes observations du jeu de données [15].

1.3.2.3 Régression linéaire

Le principe de la régression linéaire est de modéliser une variable dépendante quantitative Y , au travers d'une combinaison linéaire de p variables explicatives quantitatives, X_1, X_2, \dots, X_p . Il y a trois étapes clés pour construire le meilleur modèle de régression linéaire :

- Définir une fonction coût : c'est une fonction mathématique qui mesure l'erreur que nous com-mettons en approximant les données. Nous parlons aussi d'erreur induite par la modélisation,
- Minimiser cette fonction coût : il faut trouver les bons paramètres de notre modèle pour minimiser l'erreur de modélisation,
- Choisir une méthode de résolution du problème (la descente de gradient...) [13].

1.3.2.4 Les Forêts Aléatoires (Random Forest)

L'algorithme de forêt aléatoire est un algorithme de classification supervisé. Comme son nom l'indique, cet algorithme crée la forêt avec un certain nombre d'arbres. En général, plus il y a d'arbres dans la forêt, plus la forêt est robuste. De la même manière dans le classificateur de forêt aléatoire, plus le nombre d'arbres dans la forêt est élevé, plus les résultats de précision sont élevés. Le modèle random forest implique la création d'arbres décisionnels (decision tree) multiples en utilisant ensembles de données fractionnés à partir des données d'origine. Et en sélectionnant aléatoirement un sous-ensemble de variables à chaque étape de l'arbre décisionnel. Le modèle sélectionne ensuite le mode de toutes les prédictions de chaque arbre décisionnel [48].

1.4 Deep Learning

Le «Deep Learning» ou « apprentissage profond » est un type d'intelligence artificielle dérivé de l'apprentissage automatique, s'appuie sur un réseau de neurones artificiels s'inspirant du cerveau humain, ce réseau est composé de plusieurs couches de neurones afin de laisser la machine apprendre des formes contenues dans nos données [35].

1.4.1 Neurone artificiel

Comme un neurone biologique, un neurone est simplement comme une boîte qui va prendre des informations en entrée et va envoyer un signal en sortie.

La figure 1.4 [36] illustre le schéma d'un neurone.

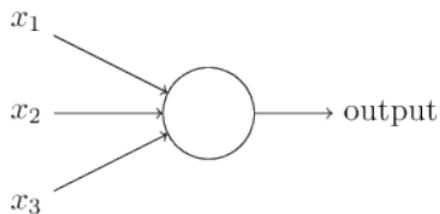


FIGURE 1.4 – Schéma d'un neurone.

1.4.2 Perceptron

C'est un neurone artificiel, aussi l'unité de base des réseaux de neurones. Il s'agit d'un modèle de classification binaire, capable de séparer linéairement deux classes de points. Il joue

un rôle important dans les projets de machine learning souvent utilisé pour la classification des données [38].

La figure 1.5 [36] ci dessous représente le schéma d'un perceptron.

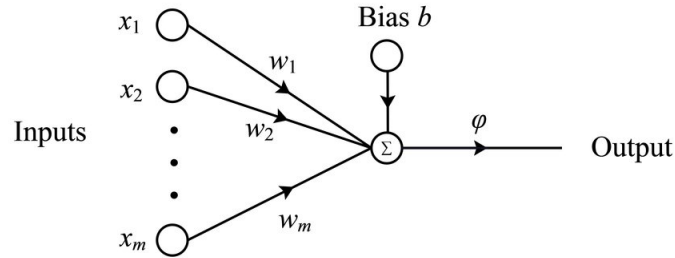


FIGURE 1.5 – Schéma d'un perceptron.

1.4.2.1 Fonctionnement

Le perceptron en réalité est représenté par une fonction mathématique :

$$\hat{Y} = \phi\left(b + \sum_{i=1}^m (x_i w_i)\right)$$

où

\hat{Y} :c'est la sortie.

ϕ :la fonction d'activation(sigmoïde, tangente hyperbolique, ...).

b :le biais .

x_i :l'entrée i.

w_i :le poids i pour l'entrée x_i .

Les données d'entrée (x) sont multipliées par les coefficients de poids (w). Le résultat produit est une valeur qui peut être positif ou négatif. Le neurone artificiel s'active si la valeur est positive c'est à dire que si le poids calculé des données d'entrée dépasse un certain seuil.

1.4.3 Réseau de neurone

Connu sous le nom de réseau neuronal artificiel (artificial neural network en anglais) est une imitation algorithmique des fonctions du cerveau humain. Ces algorithmes sont utilisés pour résoudre des problématiques d'apprentissage de la machine (Machine Learning). Un réseau de neurone est un ensemble de neurone artificiel organisé en couche (couche d'entrée, une couche de sortie et une

ou plusieurs couche caché) ou chaque neurone des couches caché est un perceptron [30]. La figure 1.6 [36] représente le schéma d'un réseau de neurone .

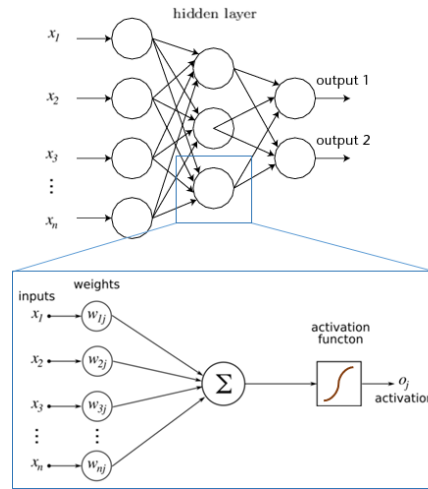


FIGURE 1.6 – Schéma réseau de neurone.

1.4.4 Les types de réseaux neuronaux

1.4.4.1 Les réseaux de neurones feed-forwarded

Les réseaux de neurones feed-forwarded (propagation avant) signifie que l'information est propagée uniquement vers l'avant et ne retourne pas en arrière. Cette famille de réseaux neuronaux comprend à son tour deux catégories de neural networks : le perceptron simple et le perceptron multicouches [45].

- Perceptron simple : Est un réseau de neurone qui possède que deux couche : couche entrée et couche sortie. Ces deux couches sont directement reliées entre elles, ce qui fait que le réseau ne possède qu'une seule matrice de poids. Ce type d'algorithme est utile que pour la classification linéaire d'un ensemble d'informations en deux catégories distinctes.
- Perceptron multicouche : A la différence du perceptron simple, le perceptron multicouche dispose entre la couche en entrée et la couche en sortie une ou plusieurs couches dites cachées. Il intègre plusieurs matrices de poids, ce qui en fait l'algorithme le plus adapté pour le traitement des fonctions non linéaires.

1.4.4.2 Les réseaux neuronaux récurrents

Recurrent Neural Network en anglais (RNN) procèdent à un traitement en cycle de l'information. Les données peuvent être véhiculées à la fois en propagation en avant et en rétropropagation,

sont spécifiquement conçus pour traiter les données séquentielles telles que les séquences de mots dans les problèmes liés à la traduction automatique, des données audio dans la reconnaissance vocale ou des séries temporelles dans les problèmes de prévision [45].

Voici deux exemples des RNN utilisées pour la prévision des séries temporelle :

- a) **Mémoire à long et court terme** : Long Short-Term Memory en anglais (LSTM) Les RNNs standards de base souffrent du problème du gradient évanescent, qui consiste en une diminution du gradient au fur et à mesure que le nombre de couches augmente. Ces réseaux ont une mémoire à court terme et n'obtiennent pas de bons résultats lorsqu'ils traitent de longues séquences qui nécessitent de mémoriser toutes les informations contenues dans la séquence complète. Pour cette raison les réseaux récurrents à mémoire à long et court terme sont apparus afin de résoudre le problème du gradient évanescent.

Le LSTM utilise trois portes pour conserver les informations pertinentes de longue date et éliminer les informations non pertinentes. Ces portes sont les suivantes :

Γ^f **porte d'oubli** : décide quelles informations doivent être jetées ou sauvegardées.

- valeur proche de 0 : l'information passée est oubliée.
- valeur proche de 1 : l'information est conservée.

Γ^u **porte de mise à jour** : décide quelle nouvelle information \tilde{c}_t utiliser pour mettre à jour l'état de la mémoire c_t . Ainsi c_t est mis à jour en utilisant à la fois Γ^f et Γ^u .

Γ^o **la porte de sortie** : décide quelle est la valeur de sortie qui sera l'entrée de l'unité cachée suivante [45].

La figure 1.7 [45] ci dessus illustre l'architecture long et court terme mémoire.

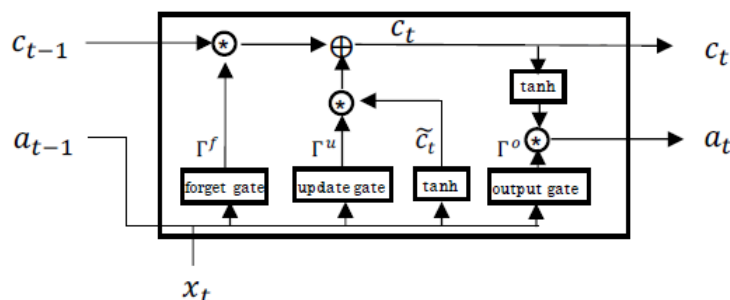


FIGURE 1.7 – Architecture long et court terme mémoire.

L'information de l'unité cachée précédente a_{t-1} et l'information de l'entrée actuelle x_t sont passées par la fonction d'activation σ sigmoïde pour calculer toutes les valeurs de la porte et par la fonction d'activation \tanh pour calculer la nouvelle information \tilde{c}_t qui sera utilisée pour la mise à jour. Les équations définissant une unité LTSM sont les suivantes :

$$\tilde{c}_t = \tanh(W_c[a_{t-1}, x_t] + b_c) \quad (1.7)$$

$$\Gamma^u = \sigma(W_u[a_{t-1}, x_t] + b_u) \quad (1.8)$$

$$\Gamma^f = \sigma(W_f[a_{t-1}, x_t] + b_f) \quad (1.9)$$

$$\Gamma^o = \sigma(W_o[a_{t-1}, x_t] + b_o) \quad (1.10)$$

$$c_t = \Gamma^u * \tilde{c}_t + \Gamma^f * c_{t-1} \quad (1.11)$$

$$a_t = \Gamma^o * \tanh(c_t) \quad (1.12)$$

où W_u , W_f , W_o , b_u , b_f et b_o sont les poids et les biais qui régissent le comportement des portes Γ^u , Γ^f et Γ^o , respectivement, W_c et b_c sont les poids et les biais du candidat de la cellule mémoire \tilde{c}_t .

- b) **Unités récurrentes à grille** : Gated Recurrent Unit en anglais (GRU) Sont des réseaux de mémoire à long terme comme les LSTM, mais ils sont apparus en 2014 comme une simplification des LSTM en raison du coût de calcul élevé des réseaux LSTM. GRU est l'une des versions les plus utilisées sur laquelle les chercheurs ont trouvé robuste et utile pour de nombreux problèmes différents. le GRU est un modèle plus simple et plus rapide à calculer car il ne possède que deux portes, la porte de mise à jour Γ^u et la porte de pertinence Γ^r [45].

La porte Γ^u décidera si l'état de la mémoire c_t est ou n'est pas mis à jour en utilisant le candidat de l'état de la mémoire c_t .

La porte Γ^r détermine la pertinence de c_{t-1} pour calculer le candidat pour c_t , c'est-à-dire \tilde{c}_t . La figure 1.8 [45] ci-dessus illustre l'architecture unité récurrente à grille.

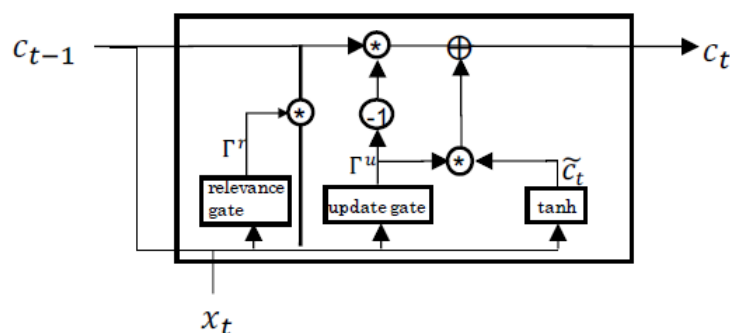


FIGURE 1.8 – Architecture Unité récurrente à grille.

Une unité GRU est définie par les équations suivantes :

$$\Gamma^u = \sigma(W_u[c_{t-1}, x_t] + b_u) \quad (1.13)$$

$$\Gamma^r = \sigma(W_r[c_{t-1}, x_t] + b_r) \quad (1.14)$$

$$\tilde{c}_t = \tanh(W_c[\Gamma^r * c_{t-1}, x_t] + b_c) \quad (1.15)$$

$$c_t = \Gamma^u * \tilde{c}_t + (1 - \Gamma^u) * c_{t-1} \quad (1.16)$$

$$a_t = c_t \quad (1.17)$$

où W_u , W_r , b_u et b_r sont les poids et le biais qui régissent le comportement des portes Γ^u et Γ^r , respectivement, W_c et b_c sont les poids et le biais de la cellule mémoire candidate \tilde{c}_t .

L'Intelligence artificielle et ses sous-catégories (machine learning et deep learning) sont couramment utilisées dans plusieurs domaines comme la prévision des ventes électriques.

1.5 Marché de l'électricité

L'expression **marché de l'électricité** désigne, de façon générale, les différentes formes d'organisation du secteur de la production et de la commercialisation de l'électricité apparus, principalement dans les années 1990, dans la plupart des pays industrialisés. Le marché de l'énergie est composé de plusieurs acteurs. Tous interviennent à différents niveaux pour produire, commercialiser, acheminer et finalement distribuer l'électricité dans les foyers des particuliers ou dans les locaux des entreprises.

1.5.1 Analyse des ventes

L'analyse des ventes décrit le processus d'identification et de traitement des données liées aux ventes et au chiffre d'affaires sur une période donnée, ce traitement de données fournit des explications et ses conclusions serviront de base au travail de l'équipe pour exécuter des actions commerciales et marketing, basées sur l'utilisation de tableaux de bord commerciaux et de techniques statistiques.

L'analyse des ventes est également utilisée pour identifier tout changement dans le comportement des clients et la saisonnalité.

Généralement, une analyse complète des ventes est placée dans le contexte du marché. Grâce à ce processus, une entreprise peut déterminer sa position sur le marché par rapport à ses concurrents et peut suivre l'évolution de cette position [2].

L'analyse des ventes d'une entreprise demeure une fonction primordiale à savoir :

- L'analyse des ventes permet d'évaluer la situation commerciale, de mesurer la Performance, suivre la réalisation des objectifs et la rentabilité de l'entreprise.
- Elle permet de prévoir les changements et de prendre les décisions appropriées.
- Elle permet d'identifié les forces et les faiblesses de la mise en œuvre des ventes de l'entreprise.
- Elle permet aussi d'identifier les produits, les clients, les régions et les vendeurs les plus rentables pour l'entreprise.

1.5.2 Prévision des ventes

La prévision des ventes est une activité dans laquelle une entreprise cherche à calculer ou à prévoir des événements futurs, sur la base d'une analyse raisonnable des données disponibles, expérience passée et tout autre événement pertinent.

La prévision recouvre également un ensemble de méthodes statistiques très diverses qui ont pour objectif de chercher à réduire l'incertitude liée à la non connaissance du futur.

L'activité de prévision est essentielle pour les entreprises, car elle est utile pour la planification des approvisionnements et de la production [34]. Les prévisions des ventes ont, donc pour principal objectif d'aider les entreprises à anticiper la demande des produits ou des services qu'elles offrent et optimiser leurs ressources, dans le but d'avoir estimé les quantités qui sont à prévoir dans le futur. Ainsi, une bonne prévision des ventes permet de déterminer les meilleures stratégies de production et d'établir quelle capacité de production est requise afin d'ajuster l'offre à la demande.

1.6 Gestion de projet

La gestion de projet est le mode de réalisation d'un projet, c'est aussi l'ensemble des outils, méthodes et technique qui permettent au chef de projet et à l'équipe de planifier le projet, suivre le budget, maîtriser les risques, respecter la qualité demandé.

Afin de bien mener les projets, plusieurs méthodes existes parmi elles on retrouve *la méthode classique et la méthode agile* :

- **La gestion de projet classique** : La méthode classique de gestion de projet ou méthode traditionnelle, prévoit la planification totale du projet, les besoins et la vision du projet final sont clairement définit. Cette méthode s'appuie sur un processus stricte, la rédaction d'une documentation détaillé et une implication faible du client. En effet, la méthode classique est un modèle linéaire, où les différentes phases de développement suivent un ordre précis. Chaque tâche doit être terminée avant de passer à la tâche suivante et la livraison du produit se fait à la fin du processus de développement.

Il existe différentes approche de gestion de projet classiques parmi elle, on trouve :

Approche systémique : Fait référence à une approche analytique, une façon de traiter des systèmes complexes dans une perspective globale sans prêter attention aux détails. Il vise à mieux appréhender la complexité sans trop simplifier la réalité [3]. Exemple : Merise.

Approche orienté objet : L'approche orientée objet considère le logiciel comme une collection d'objets dissociés définis par des propriétés. Une propriété est soit un attribut soit une opération. Exemple : (UP, 2TUP, RUP ...) [42].

- **La gestion de projet Agile** : L'approche agile est apparue en 2001 dans « Manifeste pour le développement Agile de logiciels ». L'agilité est la capacité à favoriser le changement, née de la volonté d'être en rupture avec l'approche classique qui n'est pas la mieux adaptée notamment pour certains projets complexes [29]. la méthodologie agile caractérise par une démarche itérative et incrémentale, elle décompose le projet en un ensemble d'itérations, elle fixe les objectifs à court terme et les tests se font dès le début du projet et à chaque sprint ce qui permet d'avoir un produit de qualité. Le délai est fixe, le client prévoit une date de livraison pour le produit [29].

1.7 Les méthodes agiles

La méthode AGILE est une méthodologie de gestion de projet, se base sur un cycle de développement qui porte le client au centre. Le client est impliqué dans la réalisation du début à la fin du projet ce qui permet à l'équipe d'obtenir un feedback régulier afin d'appliquer directement les changements nécessaires [26].

La méthode agile repose sur quatre grands principes :

- ✓ **Equipe** : Les individus et leurs interactions plus que les processus.
- ✓ **Application** : des logiciels opérationnelle plus qu'une documentation exhaustive.
- ✓ **Acceptation** : l'adaptation au changement plus que le suivi d'un plan.
- ✓ **Collaboration** : la collaboration avec le client plus que la négociation contractuelle.

1.7.1 Les principales méthodes agiles

Parmi les méthodes Agiles les plus utilisé on trouve :

1.7.1.1 XP (eXtreme Programming)

C'est une méthode de gestion de projet qui applique à l'extrême les principes du développement agile, c'est-à-dire se concentrer sur les besoins du client, mettre en place un développement itératif et l'intégration continue. L'objectif de cette méthode est de réduire les coûts du changement. La différence et l'originalité de cette méthode réside dans le fait qu'ils sont poussés à l'extrême. XP repose sur quatre valeurs fondamentales : Communication, Simplicité, Feedback et Courage [39].

1.7.1.2 RAD (Rapid Application Development)

L'une des premières méthodes Agile et celle qui a été la première à être en rupture avec les méthodes traditionnelles. Elle a introduit les notions d'itération et d'incrément, se base sur une durée courte d'un maximum de 120 jours.

1.7.1.3 Scrum

Le terme SCRUM fait référence à la mêlée de rugby, est une méthodologie incrémentale, organisée en sprints pendant lesquels l'équipe scrum va produire un incrément potentiellement livrable du produit. La durée des sprints est définie en amont du projet, elle est généralement de 2 ou 3 semaines pour permettre à l'équipe de produire l'incrément du produit. Cette méthode a été initialement prévue pour le développement de projets informatiques, mais elle peut être appliquée à tout type de projet, du plus simple au plus innovant, et ce, de manière très simple [29].

- **La répartition des rôles**

1. **Le responsable du produit** : est le représentant des clients et des utilisateurs, il définit les spécifications fonctionnelles et porte la vision finale du produit à réaliser. Il hiérarchise les fonctionnalités à développer ou à corriger et valide la fonctionnalité développée. Il doit jouer le rôle du client final.
2. **Scrum Master** : est le garant de la bonne application de la méthode Scrum, il élimine tous les obstacles qui peuvent empêcher l'équipe d'atteindre les objectifs fixés pour chaque sprint. Il assiste chaque rôle de l'équipe scrum dans son activité afin de les guider et les coacher dans le framework Scrum.
3. **Les membres de l'équipe** : L'équipe de développement rassemble les personnes qui réalisent le projet. Elle peut comprendre toute sorte de profils tels que : designers, concepteurs, développeurs, et bien d'autres personnes. Cette équipe a pour rôle de développer le meilleur produit possible. Selon les organisations. Elle a pour responsabilité de livrer à chaque fin d'itération une nouvelle version de l'application enrichie de nouvelles fonctionnalités et respectant le niveau de qualité.

- **Le fonctionnement de la méthode Scrum**

Un product backlog est élaboré avant le lancement du sprint par le product owner pour planifier l'ensemble des exigences du client, le backlog est composé de toutes les fonctionnalités qui doivent être intégré dans l'application ou chaque fonctionnalité est classé par ordre de priorité.

Ensuite, à chaque sprint, lors de la réunion de planification, l'équipe et le product-owner décident ensemble du sprint planning, le sous-ensemble qui sera réalisé lors du sprint.

Chaque sous-ensemble est divisé en user stories. Pour chaque user stories, l'équipe détaille les sous-tâches à réaliser et y attribue une personne en charge de la réaliser.

Chaque sprint suit, en boucle, la même chronologie pour incrémenter le produit, il est passé par les réunions suivantes [29] :

- **Sprint Planning** : au cours de cette réunion, l'équipe Scrum définit le sprint backlog, détaille les user stories et identifie les sous tâches qui sont classé par ordre de priorité, chaque sous tâches est attribué a un membre de l'équipe et elle doit être réaliser dans une période donnée.
Durée : 3 heures.
- **Daily Scrum** : également appelé "mêlée journalière". C'est une réunion généralement debout qui a lieu tous les jours au même endroit. Durant cette réunion l'équipe présente : ce qui a été fait la veille, les problèmes rencontrés, ce qu'elle compte faire aujourd'hui.
Durée : 15 minutes.
- **Sprint Review** : aussi appelée "Revue de Sprint ". L'équipe se réunit avec le client pour présenter ce qu'elle a été réalisée pendant le sprint et recueillir les feedbacks du client.
Durée : 2 heures.
- **Rétrospective** : cette réunion clôture le sprint, elle permet à l'équipe de faire le point sur ce qui a marché et ce qu'il faut améliorer et proposer un plan d'action pour améliorer le processus.
Durée : 1heure 30 minutes.

Le tableau 1.2 représente une comparaison entre les diifférents méthodes agiles

XP	SCRUM	RAD
-Itération courte(1 à 2 semaines). -Les fonctionnalités sont hiérarchisées par le client -Met l'accent sur de solides pratiques d'ingénierie	-Sprint long(2 à 4 semaines) -L'équipe Scrum décide de l'ordre des fonctionnalités. -Met l'accent sur l'auto-organisation.	-application rapide -cycle de développement court. -Accueille les changements meme tard dans le processus développement.

TABLE 1.2 – Tableau comparatif des méthodes agile.

Les méthodes agiles sont généralement appliquées pour la gestion, la conception et développement d'une application web.

1.8 Application web

Une application web est une interface web applicative disponible uniquement sur le web et accessible via un navigateur web. C'est un ensemble de pages qui interagissent avec les utilisateurs, les unes avec les autres, ainsi qu'avec les différentes ressources d'un serveur web, notamment les bases de données [4].

Parmi ces avantages :

- offre une grande mobilité.
- accès facile à partir de n'importe quel appareil.
- Il n'y a pas d'installation préalable à réaliser contrairement à un logiciel ou une application mobile.
- les sauvegardes sont centralisées sur le serveur et il n'y a plus besoin de réaliser des mises à jour sur tous les postes de travail [4].

1.8.1 Architecture des applications web

Une architecture est un modèle générique et conceptuel qui se rapporte à un sujet et qui représente la fonctionnalité, la structure, le positionnement, l'interrelation des différents types d'éléments (hardware, logiciels, infrastructure) qui la composent.

En règle générale, une application est découpée en 3 niveaux (couches) d'abstraction :

La couche présentation : La couche présentation est chargée du traitement de l'interaction avec l'utilisateur. C'est un rôle d'affichage et d'interaction.

La couche métier : La couche métier effectue les traitements applicatifs. Elle effectue de plus le tampon entre la présentation et les données. Elle effectue aussi les règles de gestion de l'application.

La couche accès aux données : Elle consiste en la partie gérant l'accès aux données du système. Ces données peuvent être propres au système, ou gérées par un autre système.

Il existe différentes architectures pour une application web [22] :

Architecture 2-tiers : Ce sont les systèmes clients/serveurs où le client demande une ressource et le serveur la lui fournit directement. Cela signifie que le serveur ne fait pas appel à une autre application afin de fournir le service.

Architecture 3-tiers : Est une extension de l'architecture à 2 niveaux. L'architecture à 3 niveaux comprend les couches suivantes :

- Couche de présentation (votre PC, tablette, mobile, etc.).
- Couche d'application (serveur).

- Serveur de base de données.

1.9 Conclusion

Dans ce chapitre, nous avons défini une application web et les différentes architectures web. Puis, nous avons abordé les séries temporelles en citant ces différents types et modèles. Ensuite, nous avons introduit le domaine de l'intelligence artificielle, machine et deep Learning. Nous avons aussi défini le marché de l'électricité et comment se font l'analyse et la prédiction de ventes.

Enfin, nous avons abordé la gestion de projet qui est une étape primordiale pour le bon déroulement de tout projet informatique.

Dans le chapitre suivant, nous ferons une analyse approfondie du fonctionnement et des besoins de l'organisme d'accueil afin d'identifier les fonctionnalités de notre application.

CHAPITRE 2

ANALYSE DES BESOINS ET CONCEPTION

2.1 Introduction

L'analyse des besoins et la conception constituent une phase importante d'analyse du projet. C'est la clé pour créer des applications de qualité.

Ce chapitre sera réservé à l'étude des besoins et la conception de l'application développée, commençant par l'identification de l'équipe Scrum du projet, les acteurs principaux de l'application, le backlog de produit, la planification des sprints jusqu'à la modélisation des différents diagrammes représentant notre système.

2.2 Présentation du groupe SONELGAZ

Sonelgaz (acronyme de **SO**ciété **N**ationale de l'**E**lectricité et du **GAZ**) est l'opérateur historique dans le domaine de la fourniture des énergies électrique et gazière en Algérie. Créée en 1969, Sonelgaz, œuvre depuis un demi-siècle au service du citoyen algérien en lui apportant cette source énergétique essentielle à la vie quotidienne. A la faveur de la promulgation de la loi sur l'électricité et la distribution du gaz par canalisations, Sonelgaz est passée d'une entreprise verticalement intégrée à une holding pilotant un Groupe industriel multi-sociétés et multi-métiers. Sonelgaz a toujours joué un rôle majeur dans le développement économique et social du pays. Sa contribution dans la concrétisation de la politique énergétique nationale est à la mesure des importants programmes réalisés, en matière d'électrification rurale et de distribution publique gaz ; ce qui a permis de hisser le taux de couverture en électricité à 98% pour 10 983 538 clients et un taux de pénétration du gaz

à 65% pour 6 886 407 clients. Aujourd'hui, le groupe Sonelgaz est composé de 14 sociétés filiales, gérées directement par la holding et de 09 sociétés en participations avec des tiers [5].

Parmi ces filiales :

- *La Société de Production de l'Electricité (SPE).*
- *Sharikat Kahraba wa Takat Moutadjadida (SKTM) .*
- *La Société de l'Engineering de l'Electricité et du Gaz (CEEG).*
- *La Société de Gestion du Réseau de Transport de l'Electricité (GRTE).*
- *La Société de Gestion du Réseau de Transport Gaz (GRTG) .*
- *L'Opérateur Système électrique (OS), chargée de la conduite du système de production et de transport de l'électricité.*
- *Sonelgaz-Distribution (anciennement Société Algérienne de Distribution de l'Electricité et du gaz - SADEG - créée en 2017 par fusion des sociétés SDC, SDA, SDE et SDO).*
- *la Société des Grands travaux d'électricité et de gaz (Kahragaz).*
- *la Société de réalisation des infrastructures énergétiques et industrielles (Inerkib).*

2.3 Les différents types de client Sonelgaz

Les types de client sont définis en fonction du type de courant et du type de tension (basse tension ou haute tension) qui lui sont attribués.

Parmi ces derniers on distingue :

- **Client HTB (Haute Tension B) :** Il s'agit des clients qui utilisent des lignes HTB qui constituent le réseau de répartition ou d'alimentation régionale et permettent le transport à l'échelle régionale ou locale. Ces lignes acheminent l'électricité aux industries lourdes, aux grands consommateurs électriques comme les transports ferroviaires et font le lien avec le second réseau. Leur tension est de 63 kV ou 90 kV.
- **Client HTA (Haute Tension A) :** C'est le type client bénéficiant des lignes HTA qui permettent le transport de l'électricité à l'échelle locale vers les petites industries, les PME et les commerces. Ces lignes ont une tension comprise entre 15 kV et 30 kV.
- **Client BT (Basse Tension) :** Ce genre de client utilise les plus petites lignes du réseau qui sont les lignes BT. Leur tension est de 230V ou 400V. Ce sont celles qui nous servent tous les jours pour alimenter nos appareils ménagers. Elles permettent donc la distribution d'énergie électrique vers les ménages et les artisans.

2.4 Présentation de l'organisme d'accueil (*Sonelgaz Distribution Direction de BÉJAIA*)

La Direction de Distribution de Bejaia est rattachée à la Société Algériennes de Distribution de l'Electricité et du Gaz de l'Est (SDE), dont le siège se trouve à Constantine, cette dernière est composée d'une Direction à laquelle sont reliés directement :

- Le secrétariat.
- Les assistants du Directeur de Distribution.
- Le chargé des affaires juridiques.
- Le chargé de la communication.
- Le chargé de sécurité.
- 9 Divisions et 10 Agences commerciales (BÉJAIA CITE TOBAL, BÉJAIA 4 CHEMAIN, EL KSEUR, AMIZOUR, SIDI AICH, SEDDOUK, AKBOU, TAZMALT, AOKAS, KHER-RATA).
- 5 Districts (BÉJAIA, AKBOU, SIDI AICH, AMIZOUR, KHERRATA).

2.4.1 Organisation de la Sonelgaz Distribution Direction de BÉJAIA

La structure organisationnel des différentes divisions de la Sonelgaz Direction Distribution de BÉJAIA est présentée dans la figure 2.1

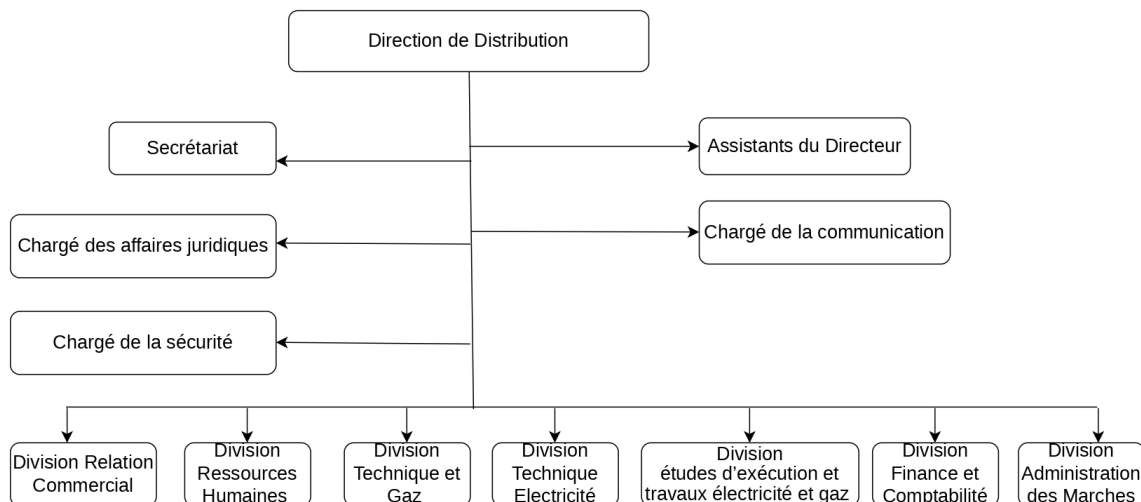


FIGURE 2.1 – Organigramme de la Sonelgaz Distribution Direction de BÉJAIA .

2.4.1.1 Division relations commerciales

Cette division joue un rôle primordial au sein de la Direction de Distribution, ses missions principales sont :

- La vente de l'électricité et du gaz .
- La gestion des relations clients.
- facturation des FSM (facturation sur mémoire) .
- Analyse statistique (taux de pertes).
- Assurer la qualité de service et la satisfaction client.

Elle est scindée en 3 services :

a. **Service développement des ventes (RCN) :**

- Prise en charge toutes les demandes de raccordements des clients en électricité et gaz.
- Orienter et conseiller les clients MT/MP en matière de mode de raccordement, la détermination du niveau de tension/ pression.
- Facturation des devis RCN et de toutes les prestations.
- Etablir les ordres d'exécutions travaux après paiement des devis par les clients.

b. **Service recouvrement :**

- Assurer la gestion des comptes clients HT/HP, MT/MP, BT/BP.
- Respecter les délais de facturations des groupes en HT/HP, MT/MP, MT/MP, BT/BP.
- Avertissement des clients lors des retards de paiements de factures (ordres de coupures, mise en demeure, ...);
- Contrôler et archiver toutes les pièces comptables.
- Assurer le suivi des clients grands comptes.
- Suivi de la facturation et du recouvrement FSM (facturation sur mémoire).

c. **Services grands comptes :**

- Assurer la gestion, la télé relève et la facturation mensuelle des clients MT/MP, HT/HP.
- Etablir le contrat de 5 ans à signer avant la mise en service d'un poste client (livraison) MT/MP.
- Veillez à la qualité des prestations de service.

2.5 Problématique

L'électricité est un facteur essentiel au développement économique dans tous les pays du monde. Son importance s'accroît avec le progrès technique, l'industrialisation, l'augmentation de production est synonyme d'amélioration de qualité de vie. C'est pour cela que l'entreprise est marquée par de nombreuses décisions. Elles sont prises chaque jour, depuis sa création jusqu'à sa mort. L'entreprise a du mal à prendre certaines décisions, dont on trouve celles de production électrique. D'après les informations acquises durant notre stage, nous avons recensé quelques problèmes majeurs :

En premier lieu, l'entreprise ne satisfait pas le vrai besoin du client dans le cas d'insuffisance de production d'une part, d'autre part, si elle la maximise elle est amenée à un nouveau problème qui est la perte d'énergie car l'électricité n'est pas une source qui peut être stockée.

En deuxième lieu, d'une part, l'entreprise est confrontée au problème de détection de fraude en cas de défaillance du système de comptage de l'énergie électrique consommée par une entreprise cliente. D'autre part, si l'équipe technique détecte un problème de fraude, la Sonelgaz ne peut évaluer la quantité d'énergie à faire facturer pour le client.

En dernier lieu, l'entreprise utilise des méthodes classiques pour l'analyse des ventes et ne possède pas des systèmes d'aide à la décision.

2.6 Solution proposée et objectif

Comme l'entreprise rencontre des problèmes de décision concernant les ventes, elle doit mettre en œuvre des méthodes d'analyse et de prévision de la structure de ses ventes.

L'émergence de l'intelligence artificielle ces dernières années notamment le machine learning et le deep learning motive les entreprises à intégrer ces nouvelles technologies dans le cadre de l'amélioration de la gestion et de faire face aux différentes problématiques pratiques.

Dans cette optique, Sonelgaz Distribution Direction de BÉJAIA souhaite se doter d'un système intelligent qui permet l'analyse des ventes des clients HTA pour mieux gérer le problème de décision.

Le but de notre travail est de répondre à leur aspiration et développer une application informatique d'un système Intelligent pour l'analyse et la prédiction de ventes énergie des clients HTA.

L'objectif principal de ce mémoire est principalement de fournir à l'entreprise un système qui leur permet de prévoir la consommation électrique future des clients HTA dans le but de minimiser la perte d'énergie et détecter la fraude. Ainsi que d'aider les analystes de la Sonelgaz a mieux visualiser la consommation de ventes des clients afin de prendre connaissance de sa clientèle, de tirer une idée

sur l'évolution de la consommation électrique pour déduire la période d'accélération de ventes. Les fonctionnalités principales de notre application sont :

1. Création et affichage de la liste des utilisateurs.
2. Localisation des clients sur la carte géographique.
3. Visualisation des ventes électriques des différents clients.
4. La prédiction des consommations électriques avec un modèle statistique utilisé dans machine learning.
5. La prédiction des futures consommations électriques de tous les clients avec un modèle deep learning en utilisant les réseaux de neurones artificiels.

2.7 Délimitation du domaine d'étude

La délimitation du périmètre de l'intervention est la composante la plus importante d'une gestion de projet réussie. Elle comprend la compréhension du contexte du projet et l'identification des acteurs impliqués. Notre domaine d'étude implique uniquement le service grands comptes (SGC) qui se situe dans la division relations commerciales (DRC), car tout ce qui concerne les analyses et les prévisions de ventes se fait dans ce service. Il reste maintenant à énumérer les acteurs intervenant dans ce même service. En effet, le succès de la mise en œuvre d'une démarche de conception agile repose sur l'implication du client dans le projet et l'identification des différents acteurs qui interagissent avec le système.

2.7.1 Les rôles scrum

Notre équipe Scrum est constituée d'un propriétaire de produit, de l'équipe de développement et d'un Scrum Master. Le modèle d'équipe Scrum est conçu pour optimiser la flexibilité, la créativité et la productivité. Nous allons tout d'abord tenter de cerner notre équipe Scrum :

- **Product Owner** : *M. Dalil HADJOUT*, son rôle est d'assurer la présentation des caractéristiques et des fonctionnalités du produit à développer et l'approbation du produit à livrer.
- **Scrum Master** : *M. Foudil MIR*, qui assure globalement la supervision de l'avancement du projet et des activités de l'équipe. Il assure également l'organisation des réunions et la bonne application de la méthode AGILE.
- **L'équipe de développement** : Composée de *M^{elle} Kahina ALOUANE* et *M^{elle} Kafya AIROUCHE* qui se chargent de la réalisation des user stories et l'élaboration des sprints.

2.7.2 Identification des acteurs

Un acteur est une entité externe qui joue un rôle dans le système. Il peut être un utilisateur humain, un dispositif matériel ou un autre système qui interagit avec l'application [19]. Dans le cas de notre application, on a deux acteurs qui sont censés interagir avec notre système :

1. **Le responsable de statistique** : qui joue le rôle d'administrateur de notre application, il aura l'accès aux différentes fonctionnalités du système. Après s'être authentifié, il pourra par exemple créer des comptes utilisateurs, effectuer l'analyse et la consultation des ventes, localisation des grands clients ou faire de la prédiction.
2. **Les utilisateurs** : qui correspondent aux différents responsables du service grand comptes au sein de Sonelgaz. Ils pourront effectuer l'analyse, la consultation des ventes et faire de la prédiction. Mais, ils auront un accès restreint aux fonctionnalités suivantes (Création et consultation de la liste des utilisateurs).

2.7.3 Diagramme de contexte

Le diagramme de contexte présente les acteurs qui interagissent avec le système avec des différents messages. La figure 2.2 représente le diagramme de contexte de notre système.

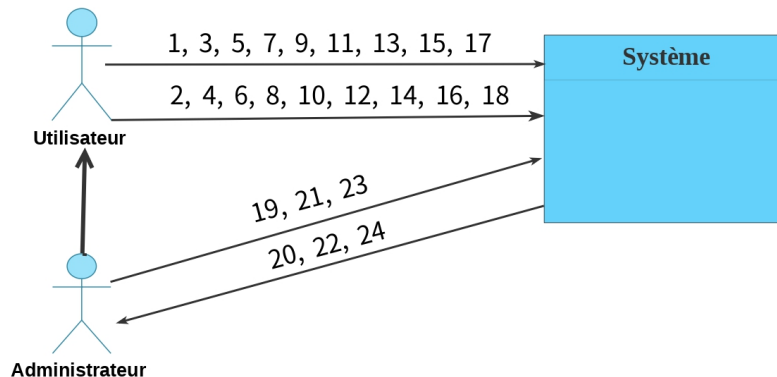


FIGURE 2.2 – Diagramme de contexte.

La description des messages est donnée dans le tableau 2.1.

Acteur	Numéro	Message	Numéro	Message
Utilisateur	1	Demande d'authentification	2	afficher l'interface d'accueil
	3	Demande d'analyse statistiques	4	Affichage de tableau de bord
	5	Demander d'afficher la localisation des clients	6	Affichage de la carte avec la localisation des clients.
	7	Demande de consultations de ventes (consommation électrique)	8	Affichage d'un choix de format
	9	Effectuer un choix	10	Affichage de la page de consommation choisi
	11	Effectuer une recherche	12	Affichage du résultat de la recherche
	13	Demande de prédiction	14	Afficher d'un choix
	15	Effectuer un choix de prédiction	16	Affichage du formulaire
	17	Remplir et valider les champs du formulaire	18	Affichage du résultat de prédiction
Administrateur	19	Demande de gestion des utilisateurs	20	Affichage d'un choix
	21	Demande d'ajout de profils utilisateurs	22	Affichage de l'interface d'ajout
	23	Demande de consulter la liste des utilisateurs	24	Affichage de la liste des utilisateurs

TABLE 2.1 – Messages échangés entre le système et les acteurs .

2.8 Pilotage de notre projet avec SCRUM

Après l'identification des besoins fonctionnels, nous devons les décomposer en users stories, les classifier en tenons compte de deux facteurs principaux qui sont la priorité et la complexité.

- **La complexité** : Elle sert à estimer l'effort nécessaire à une équipe pour implémenter une fonctionnalité. Trois éléments doivent être pris en compte pour l'estimation : l'effort pour le développement, la complexité et le risque.
- **La priorité** : Le « Product Owner » classe les users stories dans le « backlog du produit » par ordre de priorité selon leur importance pour le bon déroulement du projet.

2.8.1 Backlog produit

Le « backlog du produit » constitue l'artefact le plus important de Scrum.

Le tableau 2.2 résume le backlog de produit de notre projet :

ID	Fonctionnalités	User stories	Priorité	complexité
1	Authentification	-En tant qu'admin je dois pouvoir m'identifier au système pour avoir accès aux différentes fonctionnalités. -En tant qu'utilisateur je dois pouvoir m'identifier au système pour avoir accès aux différentes fonctionnalités.	Haute	2
2	Gérer les utilisateurs	-En tant qu'admin je doit pouvoir créer des utilisateurs. -En tant qu'admin je dois pouvoir consulter la liste des utilisateurs crée.	Moyenne	2
3	Localiser les clients	-En tant qu'admin je dois pouvoir consulter la position géographique des clients. -En tant qu'utilisateur je dois pouvoir consulter la position géographique des clients.	Faible	2
4	Visualisation de la consommation des clients	-En tant qu'admin je dois pouvoir visualiser la consommation d'un client dans une année sous format tableau. -En tant qu'admin je dois pouvoir visualiser la consommation d'un client dans une année sous format graphe. -En tant qu'utilisateur je dois pouvoir visualiser la consommation d'un client dans une année sous format tableau. -En tant qu'utilisateur je dois pouvoir visualiser la consommation d'un client dans une année sous format graphe.	Moyenne	10

ID	Fonctionnalités	User stories	Priorité	complexité
5	Prédiction de ventes avec un modèle de machine learning	-En tant qu'admin je dois pouvoir visualiser la prédiction de la consommation des 5 grands clients dans les années (2020,2021,2022,2023) sous format Tableau. -En tant qu'utilisateur je dois pouvoir visualiser la prédiction de la consommation des 5 grands clients dans les année (2020, 2021, 2022, 2023) sous format graphe. -En tant qu'admin je dois pouvoir visualiser la prédiction de la consommation des 5 grands clients dans un seul graphe pour les années (2020, 2021, 2022, 2023). -En tant qu'utilisateur je dois pouvoir visualiser la prédiction de la consommation des 5 grands clients dans un seul graphe pour les années (2020, 2021, 2022, 2023).	Haute	13
6	Prédiction de ventes avec un modèle de deep learning	-En tant qu'admin je dois pouvoir visualiser la prédiction de la consommation de tous les clients sous format graphe. -En tant qu'utilisateur je dois pouvoir visualiser la prédiction de la consommation de tous les clients sous format graphe.	Haute	100

TABLE 2.2 – Backlog produit.

2.8.2 Planification des sprints

Les « User stories » précédemment définis dans le « Backlog du produit » sont triés par ordre de priorité. Le but étant d'implémenter en premier ce qui a le plus de valeur. Le travail sera planifié selon des sprints que nous avons définis et chacun dure environ une ou deux semaines. Après une réunion avec l'équipe, on a identifié quatre sprints.

Dans ce tableau 2.3, nous présentons la planification des sprints : Cette planification est préliminaire, les Backlogs seront définis au fur et à mesure à la fin de chaque sprint et ceci dépendra de la capacité de l'équipe par lesquelles on entend la rapidité de l'équipe dans la finalisation des tâches.

Sprint	Nom du sprint	Période
Sprint 1	Localiser les clients	02 jours
Sprint 2	Visualisation de la consommation des clients	15 jours
Sprint 3	Prédiction de ventes avec un modèle de machine learning	15 jours
Sprint 4	Prédiction de ventes avec un modèle de deep learning	20 jours

TABLE 2.3 – Planification des Sprints.

2.9 Diagramme de cas d'utilisation

Formalisme qui permet de modéliser le fonctionnement d'un système par un découpage de celui-ci en fonctionnalités. Il illustre également la nature des interactions avec ces fonctionnalités, offertes à titre de services à des acteurs externes au système. La nature des interactions peut être décrite de manière sommaire ou détaillée selon le niveau de détail recherché par le modélisateur [40]. Le diagramme des cas d'utilisation est représenté dans la figure 2.3

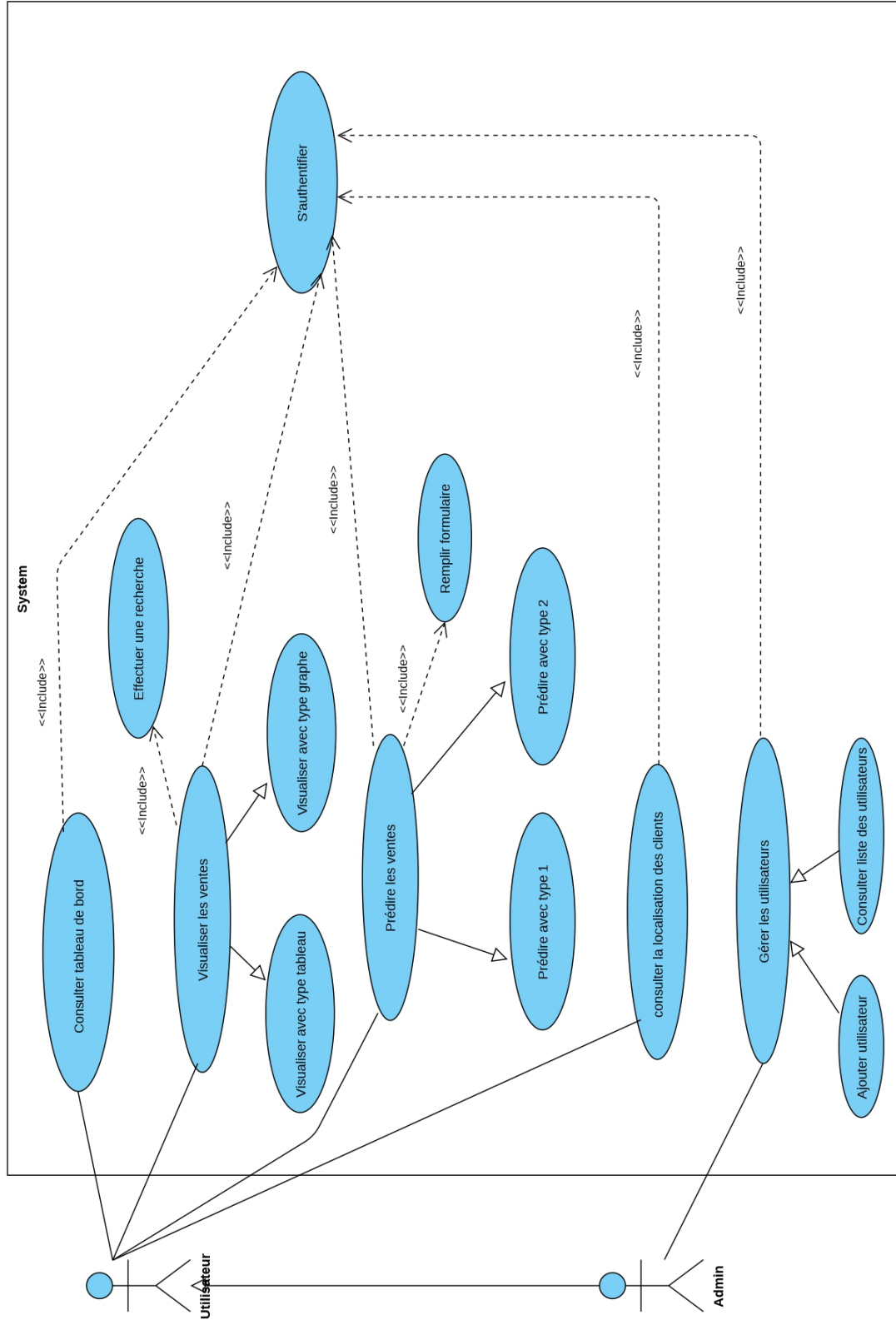


FIGURE 2.3 – Diagramme de cas d'utilisation.

2.10 Description des cas d'utilisation

Nous produisons une fiche technique et un diagramme de séquence pour les cas d'utilisation pertinents de manière à montrer le déroulement des différentes actions.

Description textuelle

Le diagramme de cas d'utilisation décrit les grandes fonctions d'un système du point de vue des acteurs, mais n'expose pas de façon détaillée le dialogue entre les acteurs et les cas d'utilisation. Il est recommandé de rédiger une description textuelle, car c'est une forme souple qui convient dans bien des situations.

Dans le tableau 2.4, on retrouve la description du formalisme suivi pour la description textuelle des cas d'utilisation de notre système :

Nom du cas d'utilisation	
Acteurs	Les acteurs qui vont réaliser le cas d'utilisation.
Description	Une description résumée de cas d'utilisation.
Auteurs	Les intervenants de l'élaboration de la fiche.
Date	La date de rédaction ou de mise à jour de la fiche.
Pré-condition	Les conditions nécessaires pour déclencher le cas d'utilisation.
Description des scénarios	
Scénario nominal	La description des interactions entre acteur et système, dans le meilleur des cas ou plus précisément dans le meilleur des scénarios .
Scénario alternatif	La description d'une suite d'étapes qui se termine : soit par la satisfaction, soit par l'abandon de l'objectif.
Post-condition	Etat du système après réalisation du cas d'utilisation et qui pourra témoigner du bon fonctionnement.

TABLE 2.4 – Formalisme de description des cas d'utilisations.

Diagramme de séquence

Diagramme de séquence permet de représenter les interactions entre objets en indiquant la chronologie des échanges. Il montre également les événements systèmes déclenchés par les acteurs. L'ordre chronologique des messages doit suivre la séquence décrite dans le cas d'utilisation ou chaque cas d'utilisation peut donner lieu à un (ou plusieurs) diagramme de séquences [40].

2.10.1 Cas d'utilisation «S'authentifier»

La description textuelle de ce cas est donnée dans le tableau 2.5 :

S'authentifier	
Acteurs	Utilisateur.
Description	Permet d'accéder aux différentes fonctionnalités du système.
Auteurs	ALOUANE Kahina AIROUCHE Kafia.
Date	26/05/2022.
Pré-condition	L'utilisateur existe dans la base de données.
Description des scénarios	
Scénario nominal	<ol style="list-style-type: none"> 1. Système affiche l'interface d'authentification à l'utilisateur après son lancement par ce dernier. 2. L'utilisateur introduit le nom d'utilisateur et le mot de passe. 3. Le système vérifie les informations envoyées, ensuite il donne l'accès à l'utilisateur pour accéder au système.
Scénario alternatif	<ol style="list-style-type: none"> 3.1. L'utilisateur a mal saisi le nom d'utilisateur ou le mot de passe. 3.2. Le système renvoie un message d'erreur et indique à l'utilisateur de ressaisir les identifiants.
Post-condition	<ul style="list-style-type: none"> • Utilisateur authentifié. • Page d'accueil s'affiche.

TABLE 2.5 – Description du cas d'utilisation «S'authentifier».

Le diagramme de la figure 2.4 représente le scénario lorsque l'utilisateur essaye de se connecter au système.

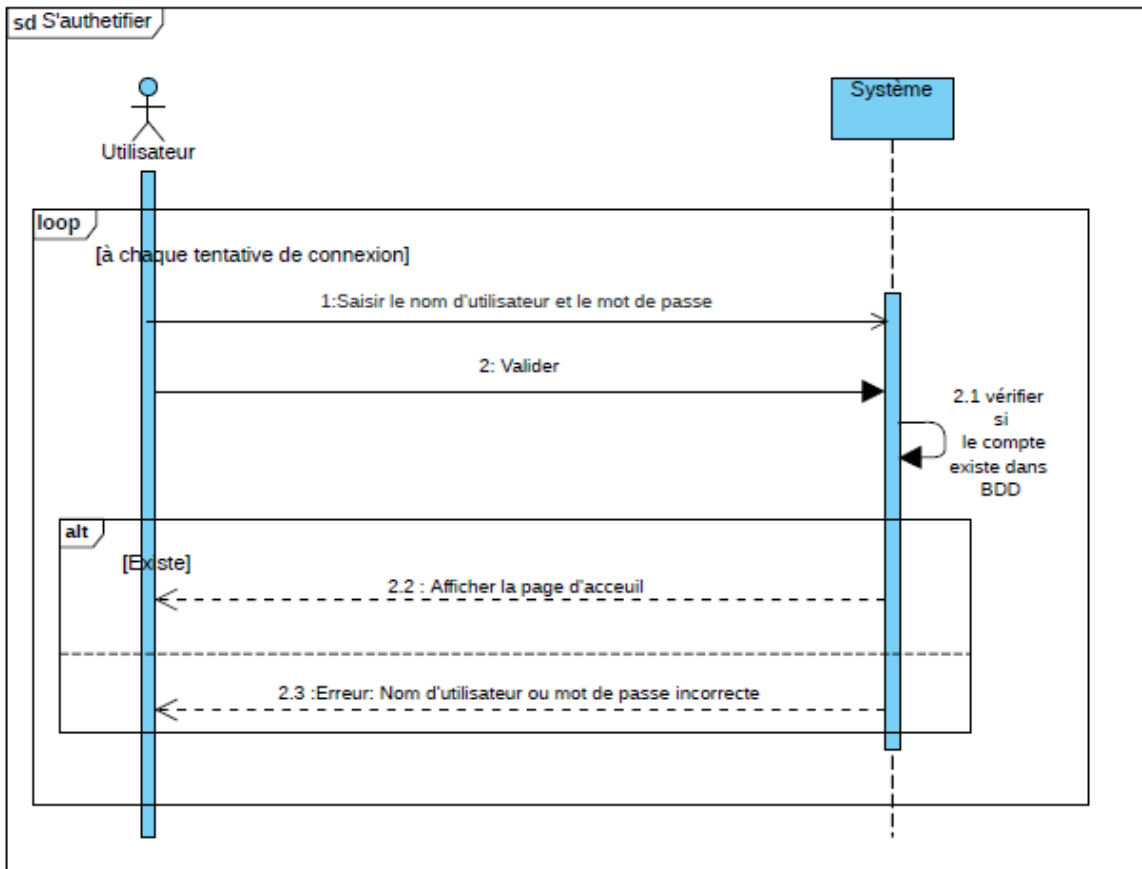


FIGURE 2.4 – Diagramme de séquence système du cas d'utilisation « S'authentifier ».

2.10.2 Cas d'utilisation « Visualiser les consommations »

La description textuelle de la visualisation des consommations est donnée dans le tableau 2.6 Le diagramme de la figure 2.5 représente le scénario lorsque l'utilisateur essaye de visualiser les consommations.

Visualiser les consommations	
Acteurs	Utilisateur.
Description	Permet à l'utilisateur de visualiser les consommations des différents client avec différents formats.
Auteurs	ALOUANE Kahina. AIROUCHE Kafia.
.Date	26/05/2022
Pré-condition	L'utilisateur doit être authentifié.
Description des scénarios	
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur demande l'accès à l'espace de visualisation des ventes . 2. Le système lui affiche un choix de type de visualisation. 3. L'utilisateur effectue un choix. 4. L'utilisateur saisie un identifiant du client et l'année dont il veut afficher les consommations. 5. Le système affiche les résultats de recherche.
Scénario alternatif	<ol style="list-style-type: none"> 4.1. L'utilisateur n'a pas fait aucune recherche. 4.2. Le système demande de remplir les champs pour effectuer la recherche. 4.3. L'utilisateur a saisie des information inexistante dans la BDD . 4.4. Le système demande de remplir les champs avec des informations valides.
Post-condition	<ul style="list-style-type: none"> • le résultat s'affiche.

TABLE 2.6 – Description du cas d'utilisation «Visualiser les consommations».

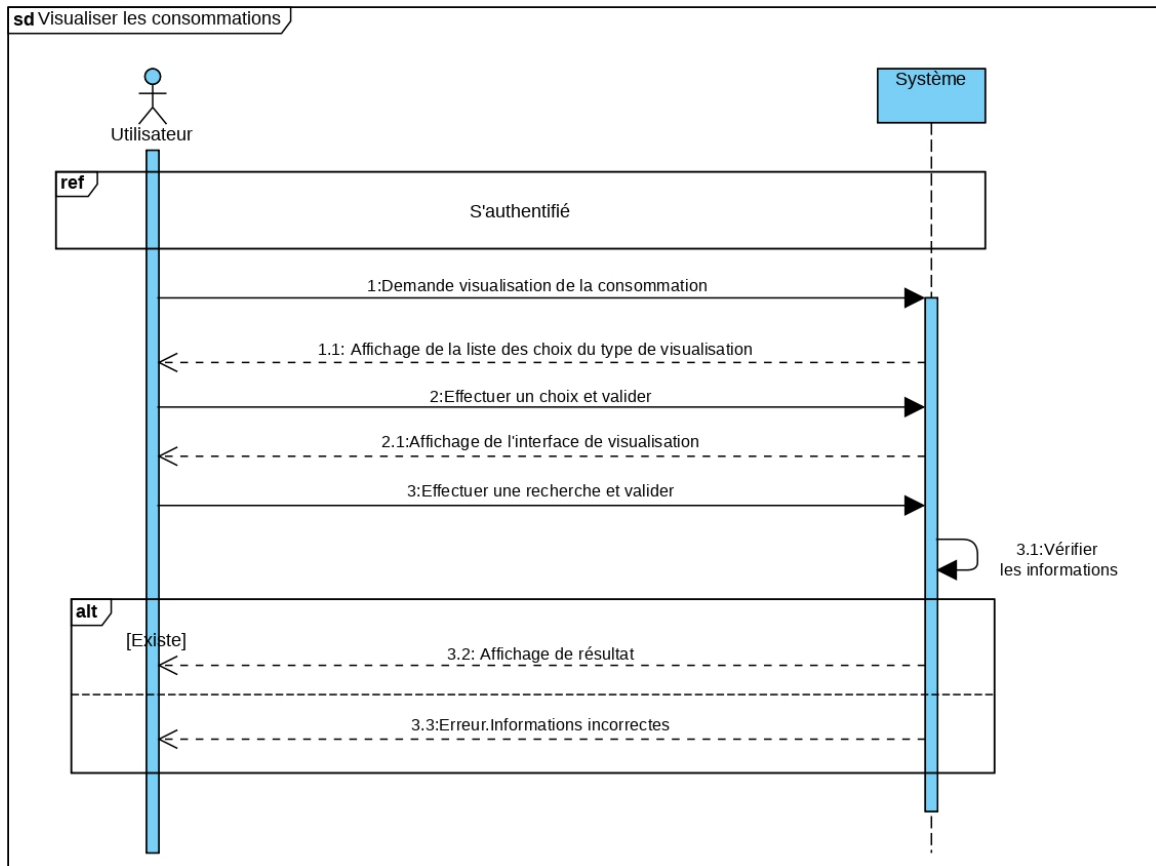


FIGURE 2.5 – Diagramme de séquence système du cas d’utilisation « Visualiser les consommations ».

2.10.3 Cas d’utilisation «Prédire les ventes»

La description textuelle de la prédiction des ventes est donnée dans le tableau 2.7

Prédire les ventes	
Acteurs	Utilisateur.
Description	Permet à l'utilisateur de prédire les ventes à travers deux types de prédiction.
Auteurs	ALOUANE Kahina. AIROUCHE Kafia.
Date	26/05/2022.
Pré-condition	L'utilisateur doit être authentifié.
Description des scénarios	
Scénario nominal	<ol style="list-style-type: none"> 1. L'utilisateur demande l'accès à l'espace de prédiction des ventes. 2. Le système lui affiche un choix de prediction. 3. L'utilisateur effectue un choix. 4. Le système lui affiche l'interface de prédiction. 5. L'utilisateur remplit et valide les champs du formulaire. 6. Le système affiche le résultat.
Scénario alternatif	<ol style="list-style-type: none"> 5.1. L'utilisateur a mal rempli le formulaire. 5.2. Le système renvoie un message d'erreur et demande de saisir des informations valide.
Post-condition	<ul style="list-style-type: none"> • Les prédictions s'affichent.

TABLE 2.7 – Description du cas d'utilisation «Prédire les ventes».

Le diagramme de la figure 2.6 représente le scénario lorsque l'utilisateur essaye de prédire les ventes.

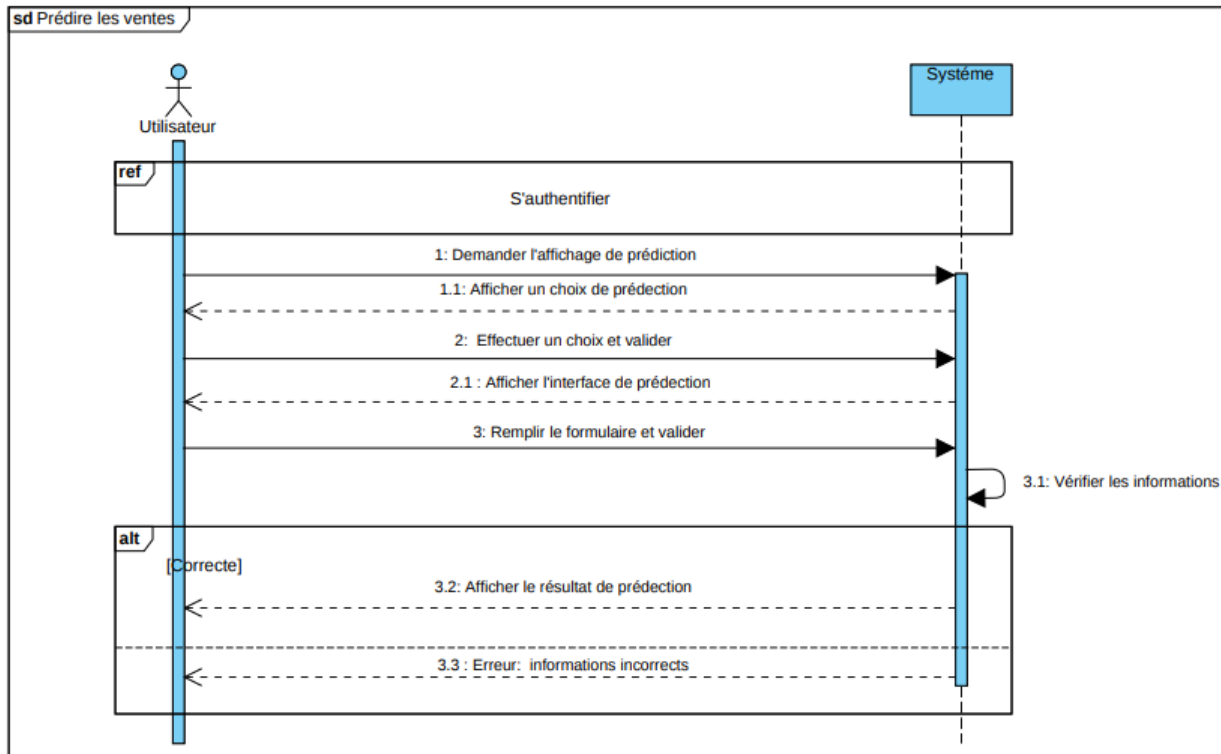


FIGURE 2.6 – Diagramme de séquence système du cas d'utilisation « Prédire les ventes ».

2.11 Diagramme séquence détaillé

Dans ce genre de diagramme, les objets communiquent en s'envoyant des messages qui invoquent des opérations (ou méthodes) sur les objets récepteurs. Il est ainsi possible de suivre visuellement les interactions dynamiques entre objets et les traitements réalisés par chacun. Ces objets sont des instances des trois types de classes d'analyse (interfaces, contrôles et entités), ils remplacent le système qui est présenté comme une boîte noire.

Pour chaque diagramme de séquence système définit précédemment nous allons établir un diagramme d'interaction.

le tableau 2.8 représente les trois types de classe utilisés dans le diagramme de séquence détaillé.

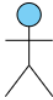
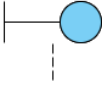

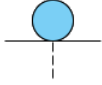
Rôle	Représentation	Signification
Acteur		une entité qui définit le rôle joué par un utilisateur ou par un système qui interagit avec le système modélisé.
Interface		servent à modéliser les interactions entre le système et ses acteurs.
Contrôle		utilisées pour représenter la coordination, l'enchaînement et le contrôle d'autres objets.
Entité		servent à modéliser des informations durables et souvent persistantes.

TABLE 2.8 – Description UML du diagramme de séquence détaillé.

2.11.1 Diagramme de séquence détaillé du cas d'utilisation «S'authentifier »

L'utilisateur doit remplir le formulaire d'authentification, ensuite le système fait un contrôle d'existence de compte, si le compte existe la page d'accueil s'affiche sinon l'utilisateur se retrouve sur le formulaire d'authentification avec un message d'erreur. La figure 2.7 illustre le diagramme de séquence détaillé qui représente ce scénario :

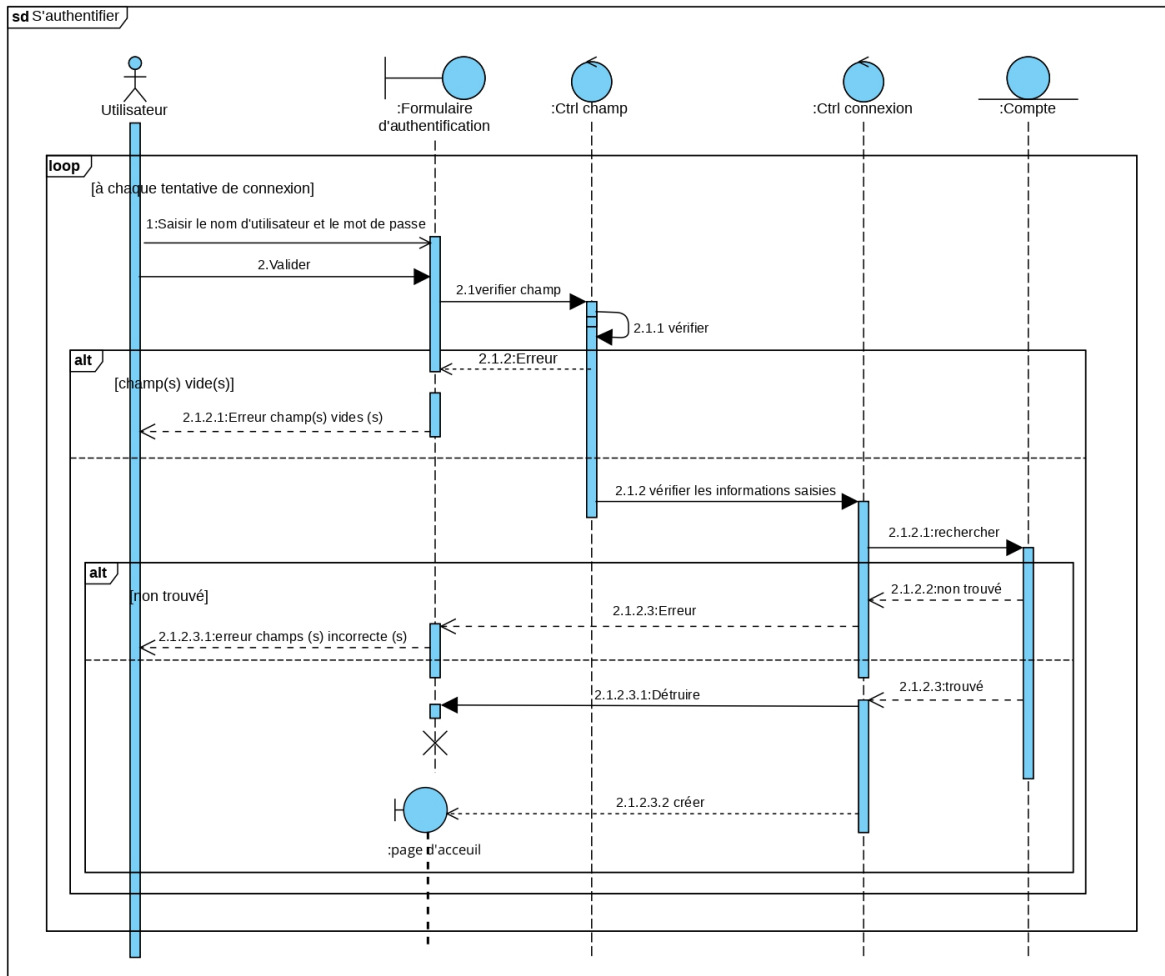


FIGURE 2.7 – Diagramme de séquence détaillée du cas d'utilisation « S’authentifier ».

2.11.2 Diagramme de séquence détaillé du cas d’utilisation « Visualiser les consommations »

Après l’authentification, l’utilisateur se retrouve dans la page d’accueil. Il accédera à la page de visualisation des consommations pour faire les analyses, il pourra visualiser sous format tableau ou graphe en effectuant une recherche sur les clients et les années. Le scénario qui s’exécute est représenté dans la figure 2.8 :

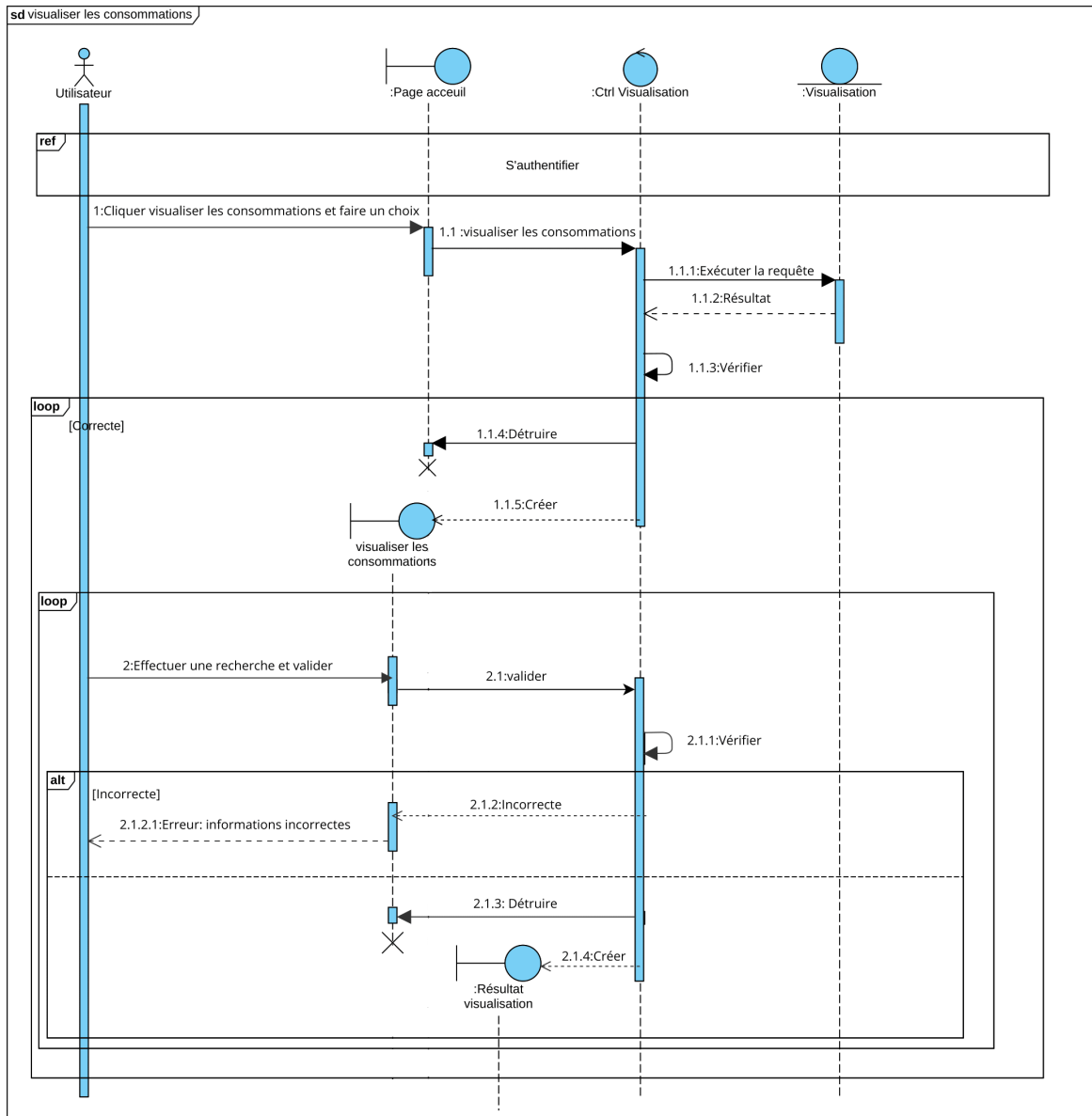


FIGURE 2.8 – Diagramme séquence détaillé du cas d'utilisation « Visualiser les consommations ».

2.11.3 Diagramme séquence détaillé du cas d'utilisation « Prédire les ventes »

Après l'authentification, l'utilisateur se retrouve dans la page d'accueil. Il accédera à la page de prédiction des ventes, il aura le résultat de prédiction après avoir fait un choix de type de prédiction et aussi avoir remplis le formulaire correctement. Le scénario qui s'exécute est représenté dans la figure 2.9

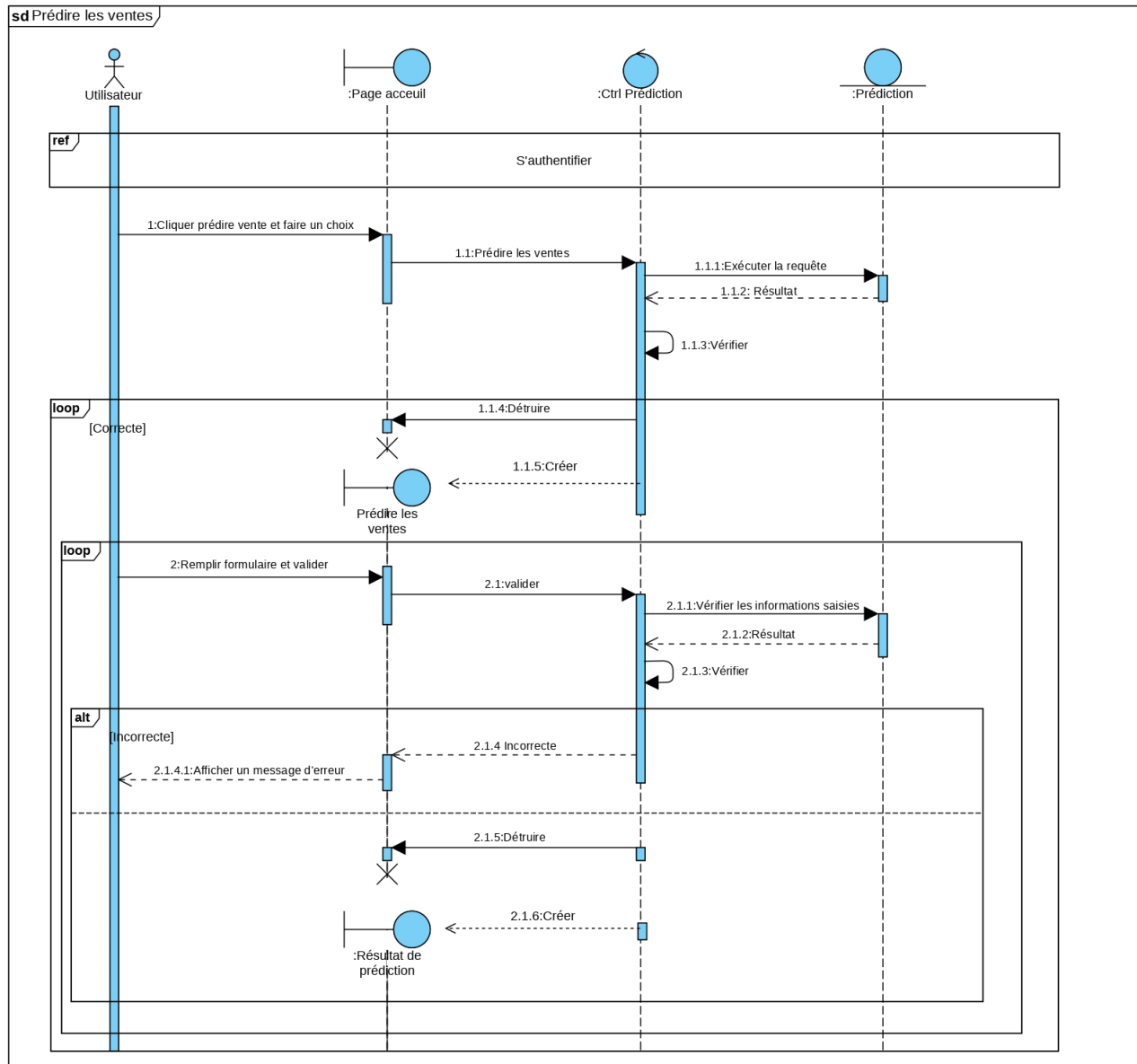


FIGURE 2.9 – Diagramme séquence détaillé du cas d'utilisation «Prédire les ventes».

2.12 Diagramme de navigation

Le diagramme de navigation est un diagramme dynamique représentant de manière formelle l'ensemble des navigations possibles entre les principales interfaces. Les diagrammes de navigation sont utilisés pour naviguer au sein des éléments et des relations dans un projet. La figure 2.10 représente le diagramme de navigation de notre système

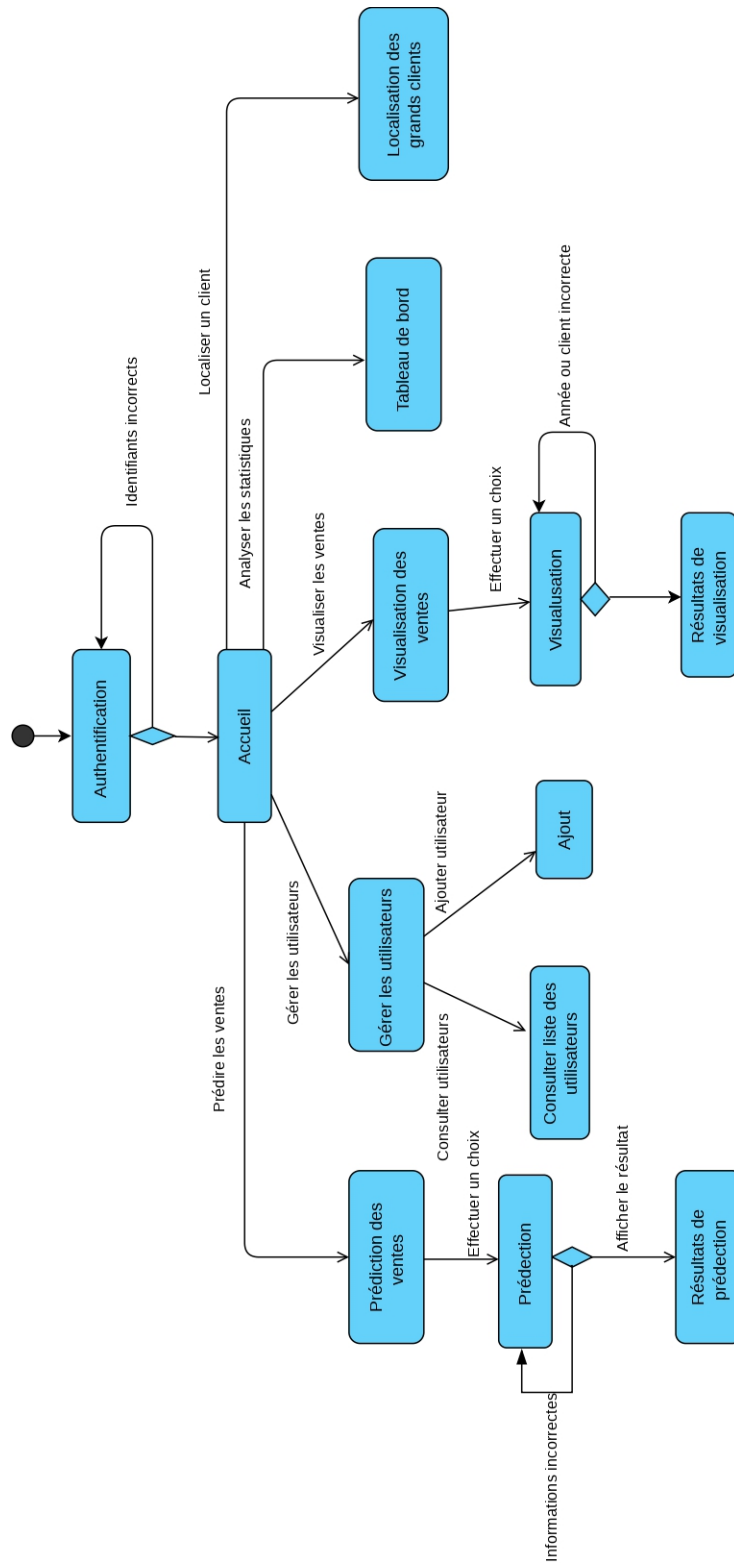


FIGURE 2.10 – Diagramme de navigation.

2.13 Diagramme de classe

Le diagramme de classe constitue l'un des hubs important pour la modélisation UML. En effet, ce diagramme permet de donner la représentation statique du système à développer. Cette représentation est centrée sur les concepts de classe et d'association. Chaque classe se décrit par les données et les traitements dont elle est responsable pour elle-même et vis-à-vis des autres classes. Les traitements sont matérialisés par des opérations [23]. le diagramme de classe qui représente la base de données de l'entreprise est défini dans figure 2.11 :

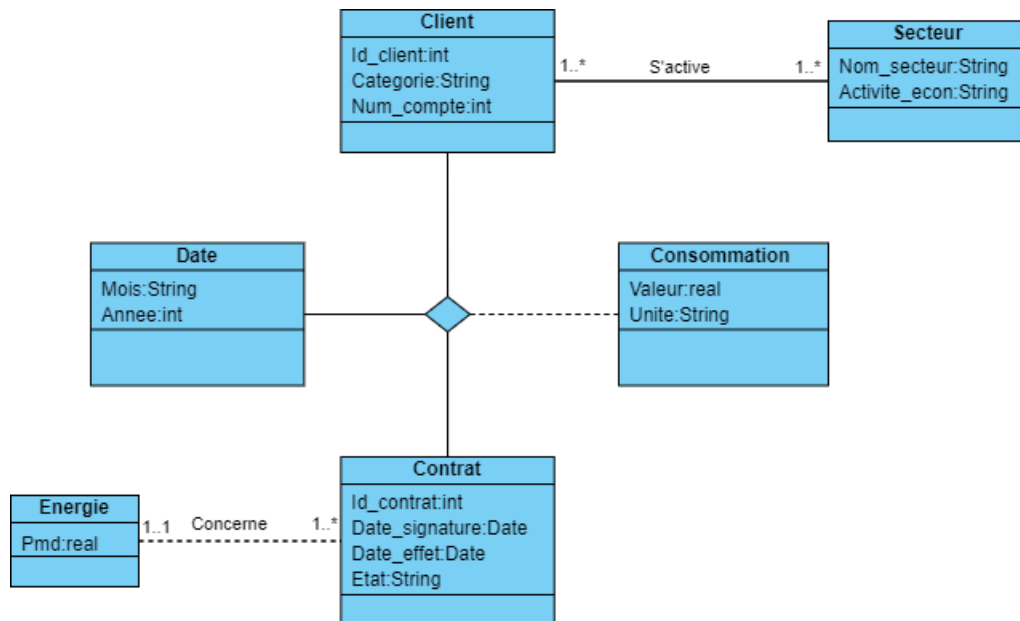


FIGURE 2.11 – Diagramme de classes.

Pour des raisons de confidentialité et de sécurité, ils nous ont communiqué uniquement les données et information nécessaire dont nous aurons besoin dans le cadre de notre stage. Donc nous nous sommes contentées uniquement des informations qu'ils nous ont communiquer qui sont représenté dans le diagramme dans la figure 2.12.

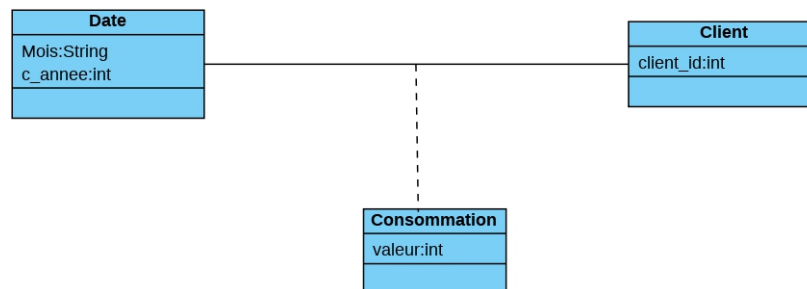


FIGURE 2.12 – Diagramme de classes.

Classe	Attribut	Description	Type
Client	Id_client	L'identifiant du client	Entier
	Catégorie	Catégorie du client	Texte
	Num_compte	Numéro du compte	Entier
Secteur	Nom_secteur	Le nom du secteur	Texte
	Activité_écon	L'activité économique du secteur	Texte
Consommation	Valeur	Valeur de consommation	Réel
	Unité	Unité de consommation	Texte
Date	Mois	Le mois de la vente	Texte
	Annee	l'année de la vente	Entier
Contrat	Id_contrat	L'identifiant de contart	Entier
	Date_signature	Date de signature du contrat	Date
	Dateeffet	Date effet du contrat	Date
	Etat	L'état du contrat	Texte
Energie	Pmd	Puissance de l'énergie mise a disposition	Réel

TABLE 2.9 – Dictionnaire de données.

2.14 Schéma de Base de donnée NoSQL

Nous avons utilisé une base de données NoSQL de type orientée document (MongoDB) pour implémenter le jeu de données. Nous avons choisi un serveur de données NoSQL, car on traite un jeu de données volumineux (la consommation mensuelle de 1699 clients durant 14 ans).

Nos données sont stockées sous forme de collections et de documents, constitués de paire clé-valeur. Chaque collection contient des documents, chaque document est composé d'un " _id " qui représente une valeur unique, c'est-à-dire la clé principale de chaque document ainsi que des champs de nom et valeurs.

La figure 2.13 représente le schéma de notre base données.

```
_id: ObjectId('6248557d4f1e81f941f9cdb6')
client_id: 1600
c_annee: 2015
Janvier: 1561
Fevrier: 1122
Mars: 816
Avril: 512
Mai: 1112
Juin: 1375
Juillet: 974
Aout: 517
Septembre: 493
Octobre: 705
Novembre: 596
Decembre: 757
```

FIGURE 2.13 – schéma BDD

2.15 Conclusion

Ce chapitre a permis d'identifier les principales fonctionnalités de notre système et de mettre l'accent sur l'application de la méthode scrum sur notre projet. Il a permis aussi de présenter en détail les différentes fonctionnalités de notre système à travers les diagrammes de séquence détaillé, ainsi que de concevoir les diagrammes de navigation et classes. Ce chapitre est une ébauche de l'étape de la réalisation de notre système que nous allons présenter dans le chapitre suivant.

CHAPITRE 3

RÉALISATION

3.1 Introduction

Dans ce dernier chapitre, nous allons exposer le travail achevé. Nous allons présenter en premier lieu l'environnement matériel et logiciel dans lesquels notre application a été développée en indiquant les technologies et les langages de programmation utilisés. Nous allons également présenter les étapes suivies pour la modélisation des prédictions à l'aide de l'apprentissage automatique et approfondie.

Enfin, nous clôturons ce chapitre par un aperçu du projet et sur quelques captures d'écrans traduisant le déroulement de l'application.

3.2 Environnement de développement de l'application

3.2.1 Architectures

L'architecture MVT (Model-View-Template) : L'architecture utilisée pour développer notre application web est une architecture 3 tiers plus exactement l'architecture MVT, elle sera basée sur la communication entre le client (navigateur) et le serveur Flask en utilisant des requêtes HTTP.

Model : Il aide à gérer la base de données. Il s'agit d'une couche d'accès aux données, qui contient les champs obligatoires et le comportement des données que vous stockez.

View : Il est utilisé pour exécuter la logique métier et interagir avec un modèle pour transporter des données et restituer un modèle. Il accepte les requêtes HTTP, puis renvoie des réponses HTTP.

Template : Il s'agit d'une couche de présentation qui gère complètement la partie interface utilisateur. Un fichier HTML dans lequel il est possible d'interpréter des objets python, c'est la page web qui s'affiche aux utilisateurs.

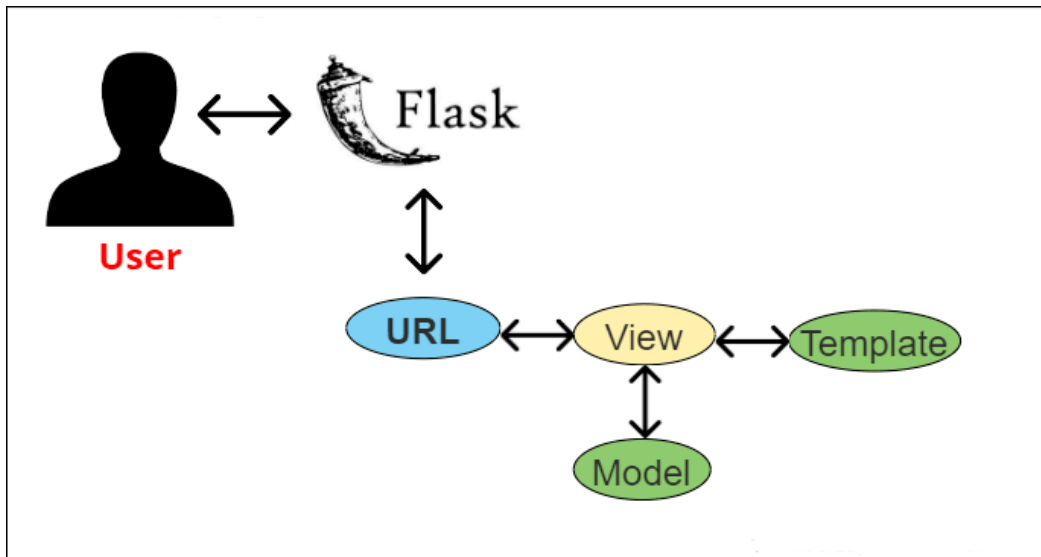


FIGURE 3.1 – L'architecture MVT.

- un utilisateur saisit une URL dans un navigateur, cette URL sera envoyé dans la vue.
 - La vue (route) reçoit une requête HTTP et renvoie une réponse HTTP convenable (par exemple si la requête est une interaction avec une base de données, la vue appelle un modèle pour récupérer les items demandés). Les vues (routes) se trouvent dans le fichier route.py .
 - Le modèle interagit avec une base de données via un ODM dans notre cas nous avons utilisé Mongoengine. Les modèles se trouvent dans des fichiers python models.py.
 - Le template est un fichier HTML récupéré par la vue et envoyé au visiteur avec les données des modèles.
- **ODM** : (Object Document Mapper) Très fréquemment utilisées dans plusieurs types de données stockées dans différents SGBD, comme les bases de données relationnelles ou les bases de données orientées documents. Un ODM fait le lien entre un modèle d'objet et une base de données orientée documents. MongoEngine est un exemple de mappeur objet-document, écrit en Python pour travailler avec MongoDB [41].

3.2.2 Matériel

Les principales caractéristiques des machines utilisées pour implémenter et tester notre application sont :

- PC 1
 - Type de la machine : DELL.
 - Processeur : Intel(R) Core(TM) i3-4005U CPU @ 1.70GHz 1.70 GHz.
 - RAM : 4 Go.
 - Système d'exploitation : Windows 10 (64 bits).

- PC 2
 - Type de la machine : HP.
 - Processeur : Intel(R) Core(TM) i3-6006U CPU @ 2.00GHz 2.00 GHz.
 - RAM : 4 Go.
 - Système d'exploitation : Windows 10 (64 bits).

3.2.3 Langages utilisés

HTML

HyperText Markup Language (HTML) est un langage de balisage hypertexte utilisé pour structurer une page web et son contenu.

L'« hypertexte » désigne les liens qui relient les pages web entre elles, que ce soit au sein d'un même site web ou entre différents sites web. Il permet entre autres, d'écrire de l'hypertexte, mais aussi d'introduire des ressources multimédias dans un contenu.

Le HTML est utilisé par l'initiative mondiale d'information World Wide Web (WWW) depuis 1990 [16].

CSS

Cascading Style Sheets (CSS) est un langage de règles de style utilisé pour mettre en forme le contenu HTML et une technologie fondamentale du World Wide Web, aux côtés de HTML et JavaScript. CSS décrit comment les éléments HTML doivent être affichés. Les feuilles de style en cascade (CSS) sont un langage de feuille de style utilisé pour décrire la présentation d'un document écrit dans un langage de balisage tel que le HTML [28].

Java Script

JavaScript est un langage de programmation informatique, une forme de code qui permet de créer un contenu plus dynamique, plus animé ou encore de réaliser des animations complexes (images, vidéos) sur une page web. Dans le développement web, le JavaScript permet aux pages web de disposer d'une meilleure réactivité et interactivité [37].

Python

Python est un langage de programmation de haut niveau interprété pour la programmation à usage général. Créé par Guido van Rossum, et publié pour la première fois en 1991 [18]. Ses principales caractéristiques sont les suivantes :

- «open-source» : son utilisation est gratuite et les fichiers sources sont disponibles et modifiables.
- doté d'une bibliothèque de base très fournie.
- importante quantité de bibliothèques disponibles : pour le calcul scientifique, les statistiques, les bases de données, la visualisation.
- grande portabilité : indépendant vis à vis du système d'exploitation (linux, windows, MacOS)
- .

3.2.4 Frameworks

Bootstrap

Bootstrap est le populaire HTML, CSS et JavaScript. Framework gratuit et open-source pour le développement Web frontal réactif et mobile. Il offre des outils avec des styles déjà en place pour des typographies, des boutons, des interfaces de navigation et bien d'autres encore [44].

Flask

Flask est un framework web écrit en python et basé sur la boîte à outils WSGI et jinja2. Il fournit un modèle simple pour le développement web, il n'a pas de couche d'abstraction de base de données, de validation de formulaires ou d'autres composants [33].

3.2.5 Logiciels

Anaconda

C'est une distribution Python particulièrement populaire pour l'analyse de données et le calcul scientifique.

- Projet open source développé par Continuum Analytics, Inc ;
- Disponible pour Windows, Mac OS et Linux ;
- Comprends de nombreux packages populaires : NumPy, Matplotlib, Pandas, Cython ;
- Inclut Spyder, un environnement de développement Python ;
- Conda, un gestionnaire de packages indépendant de la plateforme ;
- Inclut Jupyter Notebook [20].

Jupyter Nootbook

Jupyter Notebook est un outil open source permettant d'écrire du code informatique, s'exécute via un navigateur web. Il permet de combiner le code, les commentaires, le contenu multimédia et les visualisations dans un document interactif : le notebook. Il supporte de nombreux langages de programmation. Il est abondamment utilisé dans la data science [43].

Visual paradigm

Visual est un outil CASE UML, multiplateforme conçu pour un large éventail d'utilisateurs, y compris les ingénieurs logiciels, les analystes pour modéliser les systèmes d'information. Il permet de créer des diagrammes UML ainsi que d'autre type de diagramme, fournit des capacités de génération de rapports de code [12].

MongoDB Compass

MongoDB est l'interface utilisateur graphique de Mongo DB. Il nous permet d'examiner visuellement nos données et exécutez des requêtes. Les données sont stockées sous forme de documents, ces documents sont stockés sous format JSON. Il utilise la fonctionnalité CRUD complète pour interagir avec nos données [25].

PyCharm

PyCharm est un outil développé par la société JetBrains, un environnement de développement intégré permettant d'écrire et d'exécuter des programmes ou des scripts en langage Python. Il a été développé en Python et en Java. PyCharm est publié sous une licence propriété pour les entreprises et administrations et sous la licence Apache (licence de logiciel open source et libre) pour la communauté des développeurs Python [6].

3.2.6 Bibliothèques Python

Pandas

Pandas est la bibliothèque la plus complète en ce qui concerne la manipulation de données. Elle permet de travailler avec des tableaux de données en deux dimensions (lignes et colonnes)

appelés DataFrames [9].

NumPy

Cette bibliothèque est très utile pour effectuer des opérations mathématiques et statistiques en Python. Numpy contient des modules de gestion de données et de calcul. Elle permet de gérer facilement des bases de données [32].

Matplotlib

Est une librairie puissante, elle permet de générer directement des graphiques à partir de Python. Matplotlib offre une large variété de types de graphes qui s'adaptent à tous les besoins : histogrammes, courbes, scatter plots, camembert, etc [7].

Scikit learn

Scikit learn, aussi appelée sklearn, est une bibliothèque open source de machine learning la plus populaire. Cette bibliothèque permet d'effectuer du supervisé ou du non-supervisé, elle fournit une sélection d'outils efficaces pour l'apprentissage automatique et la modélisation statistique, notamment la classification, la régression et le clustering [8].

Keras

Écrit en Python, Keras est une bibliothèque open source de prototypage rapide de modèles de deep learning. A la portée des débutants en IA, elle s'articule autour d'une API de haut niveau supportant différentes librairies de réseaux de neurones artificiels récurrents ou convolutifs, comme TensorFlow [31].

TensorFlow

Est une bibliothèque Open Source, de calcul numérique et de Machine Learning compatible avec le langage Python. Elle regroupe des modèles et des algorithmes de Machine Learning et de Deep Learning. Tensorflow simplifie le processus de génération de prédictions des résultats futurs [14].

3.3 Modélisation de la prédiction de vente

L'objectif principal de notre application est de développer un système pour la prédiction des ventes (prévision consommation électrique des clients HTA).

Pour ce faire, nous nous intéressons au modèle SARIMA évoquée dans la sous section 5 de la section 3 du chapitre 1. Ainsi que, le modèle LSTM évoquée dans la sous section 4 de la section 5 du chapitre 1. Ces deux modèles sont utilisés pour la prévision des séries temporelles.

Dans ce qui suit, nous allons expliquer en détail ces deux méthodes que nous avons utilisées pour

la prédiction.

3.3.1 Modélisation de la méthode SARIMA

Nous avons utilisé cette méthode pour prédire la consommation des 5 grands client HTA cette méthode nécessite un modèle pour chaque client.

La modélisation du modèle SARIMA se fait suivant les différentes étapes expliquées dans ce qui suit :

1. **Collecte et récupération des données** : La première étape a été la constitution d'un jeu de données sur lesquels nous pourrions tester et vérifier les algorithmes. Nous avons exploré toutes les pistes possibles pour récupérer les données nécessaires qui constituent l'expérience et les exemples que nous allons fournir aux algorithmes afin qu'ils puissent apprendre.
2. **Nettoyage de données** : Dans cette étape, nous avons employé la librairie *Pandas*. En premier, nous avons importé le jeu de données qui contient notre Dataset. Ensuite, nous avons nettoyé les données en remplissant les cases vides par des zéros, éliminer les valeurs aberrantes¹, la même librairie est utilisée pour visualiser les données.
3. **Découpage des données** : nous avons découpé le jeu de données en deux parties avec la librairie *pandas* en utilisant la méthode *dataframe.iloc()* :
 - **Données d'entraînement (Train set)** : représentent 90% de données destinées à l'apprentissage du modèle.
 - **Données de test (Test set)** : représentent 10% des données destinées à l'évaluation du modèle. Ces données ne doivent en aucun cas être utilisées lors de la conception du modèle.
4. **Modélisation** : Le processus de construction du modèle SARIMA divisée en plusieurs étapes : D'abord, nous avons observé le graphique des composantes de la série temporelle avec la librairie *statsmodels.tsa.seasonals* via la fonction *seasonal_decompose* pour avoir une vision globale de la chronique à étudier et à réaliser.
Ensuite, nous avons testé la stationnarité de cette série en utilisant le (test dickey fuller), un test statistique qui permet d'évaluer si une série donnée est stationnaire ou non, nous l'avons utilisé avec la librairie *statsmodels.tsa.stattools* via la fonction *adfuller*. La figure 3.2 représente les composantes de la série temporelle du client 967 :

1. valeurs aberrantes : En statistique, une donnée aberrante est une valeur ou une observation qui est « distante » des autres observations effectuées sur le même phénomène.

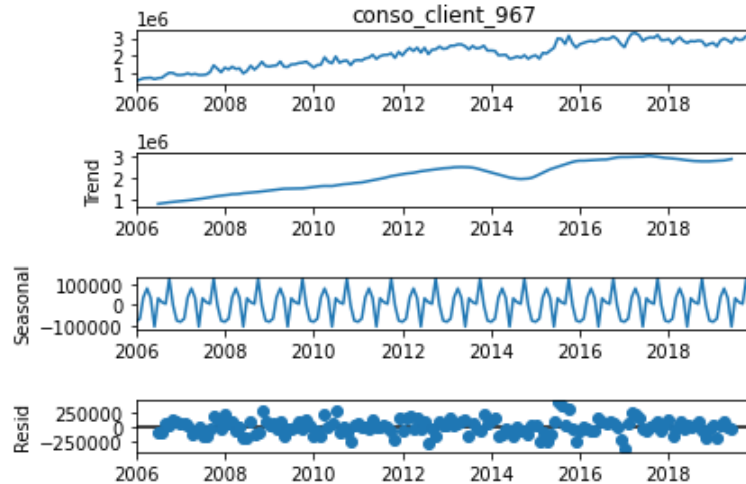


FIGURE 3.2 – Les composantes de la série temporelle du client 967.

En analysant les graphes de la figure 3.2, nous avons conclu que la série temporelle possède une tendance et une saisonnalité et dans notre cas la série est non stationnaire (la valeur de p -value est proche de 1) .

On entend par la non-stationnarité le fait que la structure d'un processus stochastique évolue avec le temps. SARIMA est un processus non stationnaire qui traite la non stationnarité et la saisonnalité.

La somme de toutes les informations citées ci-dessus, nous amène à la conclusion que, pour notre cas, SARIMA est la méthode la plus adaptée aux traitement de nos données ou de notre série.

Par la suite, nous avons appliqué sur les données d'entraînement le modèle SARIMA intégré dans *statsmodels.api*, après avoir choisi les ordres p, d, q, P, D et Q du modèle selon l'AIC¹ le plus bas.

5. **Evaluation :** Pour évaluer la performance du modèle obtenu, on doit s'assurer que les résidus de notre modèle ne sont pas corrélés et normalement distribués. Pour cela, nous avons affiché les diagnostics qui sont présenté ci-dessous avec la fonction *plot_diagnostic* de la librairie *matplotlib* qui nous a permet de générer rapidement des diagnostics de modèle et d'enquêter sur tout comportement inhabituel.

Comme nous pouvons le constater dans les figures 3.3, 3.4, 3.5, 3.6 la bibliothèque *matplotlib*

1. AIC : utilisé pour la selection du modèle. Il détermine la valeur d'information relative du modèle à l'aide de l'estimation du maximum de vraisemblance et nombre de paramtres du modèle.

nous affiche 4 différents graphes soit :

- **Histogram plus estimated density** : nous voyons que la KDE¹ ligne rouge suit presque de près la $N(0,1)$ ligne (où $N(0,1)$ est la notation standard pour une distribution normale avec une moyenne 0 et un écart type de 1). C'est une bonne indication que les résidus sont normalement distribués.

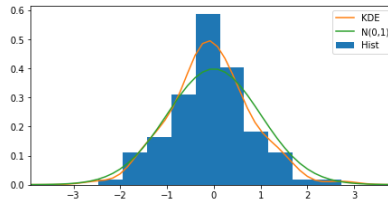


FIGURE 3.3 – Les diagnostics du modèle sarima-Histogram plus estimated density.

- **Graphique Q-Q** : Montre que la distribution ordonnée des résidus (points bleus) suit la tendance linéaire des échantillons tirés d'une distribution normale standard avec $N(0,1)$. Encore une fois, cela indique fortement que les résidus sont normalement distribués.

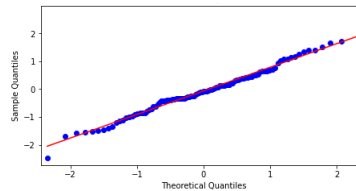


FIGURE 3.4 – Les diagnostics du modèle sarima-Graphique Q-Q.

- **Standardized residual** : Les résidus au fil du temps n'affichent aucune saisonnalité évidente et semblent être un bruit blanc. Ceci est confirmé par le graphique d'auto-corrélation (c'est-à-dire le corrélogramme).



FIGURE 3.5 – Les diagnostics du modèle sarima-Standardized residual.

- **Correlogram** : Montre que les résidus de la série chronologique ont une faible corrélation.

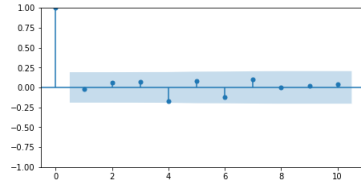


FIGURE 3.6 – Les diagnostics du modèle sarima-Correlogram.

Ces observations nous amènent à conclure que notre modèle produit un ajustement satisfaisant qui pourrait nous aider à comprendre nos données de séries chronologiques et à prévoir les valeurs futures.

6. **Exportation des résultats :** Nous avons sauvegardé les résultats des prédictions dans un fichier excel avec la librairie *pandas* en utilisant la méthode *dataframe.to excel()* .

Nous avons suivi la même démarche décrite en haut pour construire 5 modèles pour les 5 grands clients (chaque client a son propre modèle). Les résultats de prédiction ont été sauvegardés dans un fichier Excel que nous avons intégré dans la BDD que nous interrogeons grâce au serveur web Flask .

3.3.2 Modélisation de la méthode LSTM

Nous avons utilisé cette méthode pour prédire la consommation de tous les clients HTA, cette méthode nécessite un seul modèle pour tous les clients contrairement à SARIMA qui nécessite un modèle pour chaque client. Nous avons choisi LSTM car nous voulons faire des prédictions basées sur le temps (des longues séquences) et nous avons un ensemble de données avec des échantillons (chaque client a sa consommation) et la prédiction va dépendre de plusieurs items en entrée. Le processus de construction du modèle LSTM se fait suivant les différentes étapes expliquées dans ce qui suit :

les 2 premières étapes **préparation des données et nettoyage des données** sont les mêmes que celles citées dans la section précédente (SARIMA).

3. **Découpage des données :** Nous avons découpé le jeu de données par année avec la librairie *pandas* en utilisant la méthode *dataframe.loc()* en deux parties :

- a. **Données entraînement :** Les données à partir de l'année 2006 jusqu'à 2018 sont destinées à l'entraînement du modèle.

1. KDE : est une méthode non-paramétrique d'estimation de la densité d'une variable aléatoire

- b. **Données de test** : les données de l'année 2018 et 2019. l'année 2019 pour faire le test et l'année 2018 qui est indispensable pour prédire 2019.
4. **Supervision des données** : Cette étape consiste à superviser l'apprentissage de la machine en lui montrant des exemples (des données) de la tâche qu'elle doit réaliser. Nous avons supervisé de façon que pour chaque 12 valeurs en entrée (12 mois) nous aurons une seule valeur en sortie avec un chevauchement pour au final avoir un résultat de prédiction annuel par client en utilisant la technique appelée sliding window.
5. **Normalisation des données** : Pour modéliser correctement les données ainsi que les neurones de notre réseau apprendront plus rapidement nous avons normalisé les données c'est à dire les mettre sur une même échelle avec la fonction "*MinMaxScaler*" dans le package *preprocessing* de la librairie *sklearn* pour rendre nos données entre 0 et 1.
6. **Modélisation** : On a construit notre modèle lstm en utilisant les deux approches suivantes :
- **conception du modèle** :
 Nous avons utilisé l'API *Keras sequential* pour construire notre variable de modèle. Nous avons construit un réseau de neurones avec 3 couches : la première contient 512 unités lstm ou chaque unité prends 12 valeurs en entrée et une seule valeur en sortie, la deuxième avec 256 unités, la troisième avec 128 unités. Ensuite, nous avons désactivé 20% d'unité pour chaque couche pour éviter le overfitting (le risque pour un modèle d'apprendre par cœur) et une couche de sortie avec une seule unité.
 Ensuite, nous avons configuré notre modèle avec différents paramètres :
 - nous avons utiliser l'**optimiseur Adam**¹ avec la librairie *keras* pour implémenter la descente de gradient pour minimiser la fonction de perte.
 - Pour la fonction de perte on a utilisé la mse (mean squared error)
 - **Entraînement du modèle** : Nous avons entraîné notre modèle avec les hyperparamètres suivants :
 - * **nombre d'échantillons** : 1570 clients.
 - * **taille de lots (batch size)** : 64.
 - * **époques** : 100.
 - **Époque** : Le nombre de fois que l'algorithme d'apprentissage fonctionnera (le nombre de fois que le modèle va passer) sur l'ensemble de données d'apprentissage.
 - **Descente de gradient** : Est un algorithme d'optimisation qui permet de trouver le minimum de n'importe quelle fonction [43].

1. Adam : Est un algorithme d'optimisation de la technique de descente du gradient. Elle est efficace lorsqu'on travaille sur un problème de grande taille impliquant un grand nombre de données ou de paramètres.

- **Fonction de perte** : Est une fonction qui évalue l'écart entre les prédictions réalisées par le réseau de neurones et les valeurs réelles des observations utilisées pendant l'apprentissage.
- **BatchSize** : Taille du lot ou le nombre d'exemples d'apprentissage en une seule époque.

7. **Evaluation** : Nous avons utilisé les metrics accuracy et MSE pour évaluer la performance du notre modèle et MAPE pour évaluer les résultats du modèle obtenu après l'entraînement.

- **accuracy** : Métrique de performance qui se calcule en divisant les résultats de prédiction sur les données réelle .
- **MSE (Mean Squared Error)** : C'est la fonction de perte la plus couramment utilisée pour la régression. La perte est la moyenne des données supervisées des différences au carré entre les valeurs vraies et prédites [11]. définie comme suit :

$$MSE = \frac{1}{n} * \sum (réel-prévu)^2 \quad (3.1)$$

Où :

n : Taille de l'échantillon qui est le nombre de client.

réel : La valeur réelle des données.

prévu : La valeur de données prédite.

- **MAPE (Mean Absolute Percentage Error)** : Est l'une des mesures les plus populaires de la précision des prévisions, elle mesure cette précision en pourcentage [27] définie comme suit :

$$MAPE = \frac{1}{N} \sum_{T=1}^N \left| \frac{A_t - F_t}{A_t} \right| \quad (3.2)$$

Où :

N : Nombre d'échantillons (nombre clients).

At : La valeur réelle des données.

Ft : La valeur de données prédite.

Métrique	Résultat
Accuracy	0.0002
MSE	0.002
Loss	0.002
MAPE (Janvier)	1.532655
MAPE (Février)	1.352949
MAPE (Mars)	4.955473
MAPE (Avril)	4.433453
MAPE (Mai)	3.198503
MAPE (Juin)	0.418555
MAPE (Juillet)	4.52921
MAPE (Août)	2.605849
MAPE (Septembre)	1.022349
MAPE (Octobre)	5.666635
MAPE (Novembre)	0.529647
MAPE (Décembre)	0.019882

TABLE 3.1 – Tableau des résultats LSTM.

8. **Sauvegarde du modèle** : Pour sauvegarder notre model nous avons utilisé le format Save-Model de la librairie *keras* avec l'Api `model.save()`. SavedModel est le format de sauvegarde le plus complet qui enregistre l'architecture du modèle et les poids.
9. **Déploiement du modèle** : Nous avons intégré le modèle sauvegardé précédemment dans notre projet que nous avons interrogé dans le serveur Flask avec une fonction que nous avons codés nous même.

3.4 Présentation de notre application

3.4.1 Application web

L'architecture de l'application web développé est une architecture client/serveur basée sur le concept d'API REST, elle même autogéré par Flask. Le client étant le navigateur effectue la

demande et le serveur (micro serveur Flask), traite et renvoie la réponse en utilisant les requêtes HTTP.

Les interfaces graphiques de la partie client de l'application web sont réalisées et développées par le langage de balisage HTML et les feuilles de style CSS ainsi que javascript et bootstrap, pour la partie serveur elle est réalisée à l'aide du framework web Flask qui utilise l'architecture modèle vue template décrite au-dessus.

3.4.2 Quelques interfaces graphiques

Nous allons présenter les différentes interfaces de notre application et expliquer l'utilité de chacune d'entre elles :

3.4.2.1 Interface d'authentification

La figure 3.7 représente l'interface d'authentification où l'utilisateur saisi son nom et son mot de passe pour pouvoir accéder aux fonctionnalités du système.

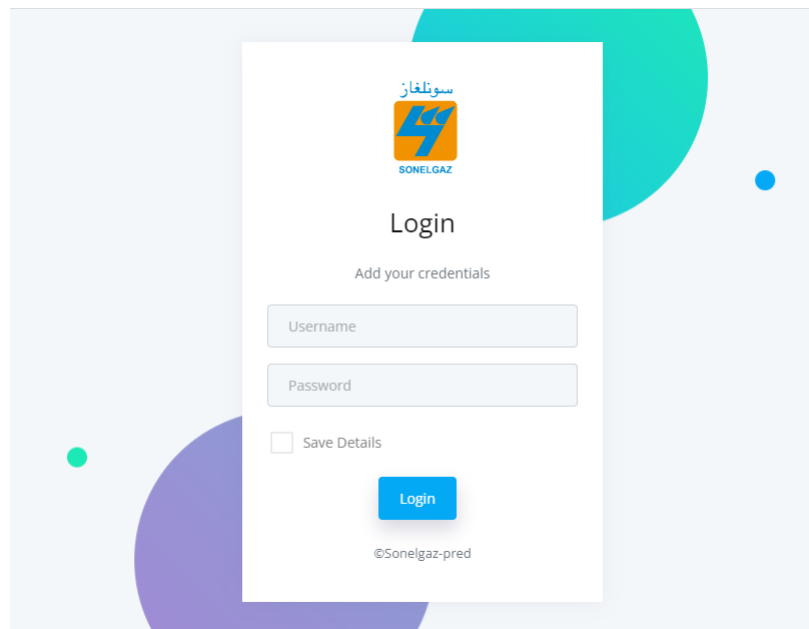


FIGURE 3.7 – Interface Authentification.

3.4.2.2 Interface de tableau de bord

la figure 3.8 correspond à l'interface tableau de bord qui est la première interface qui s'affiche après l'authentification de l'utilisateur, cette interface permet de visualiser les statistiques concernant la consommation annuelle de l'électricité, le nombre de clients etc.

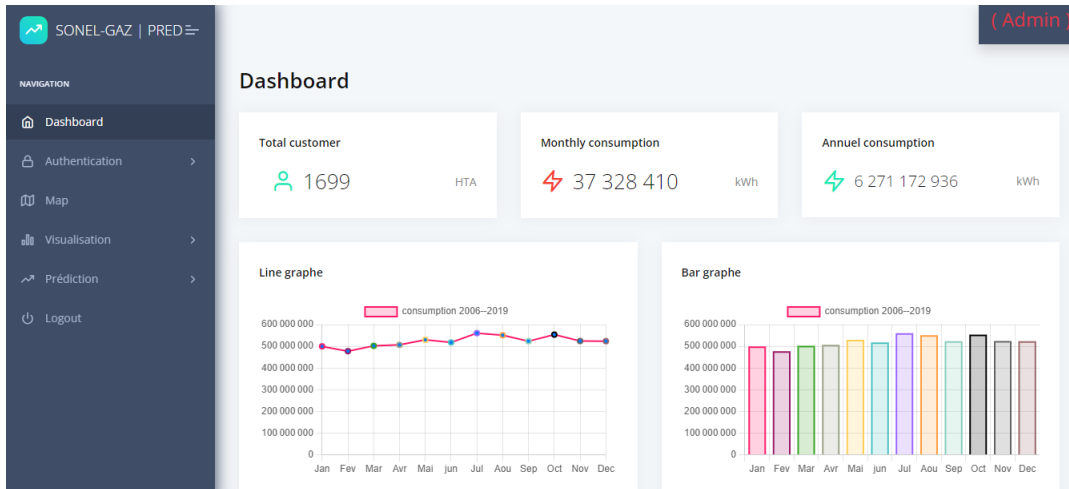


FIGURE 3.8 – Interface Tableau de bord.

3.4.2.3 Interface Utilisateurs

Cette interface est accordée qu'à l'administrateur de l'application, elle lui permet de créer des comptes utilisateurs pour qu'ils puissent accéder aux différentes fonctionnalités du système. Cette interface est illustrée dans la figure 3.9

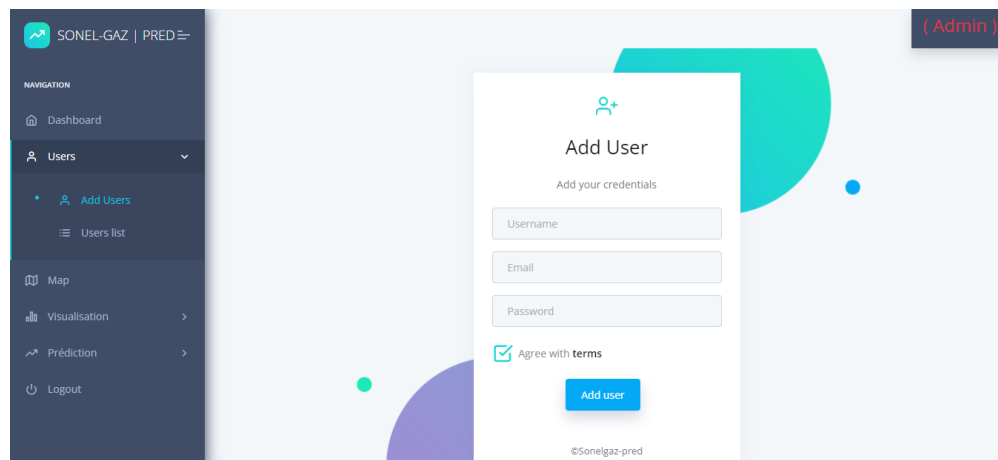


FIGURE 3.9 – Interface Utilisateurs.

3.4.2.4 Interface visualisation des ventes

Cette interface permet de visualiser les ventes des différents clients. L'utilisateur pourra visualiser les ventes dans différents formats :

- * Format tableau après avoir saisi l'id du client, illustré dans figure 3.10

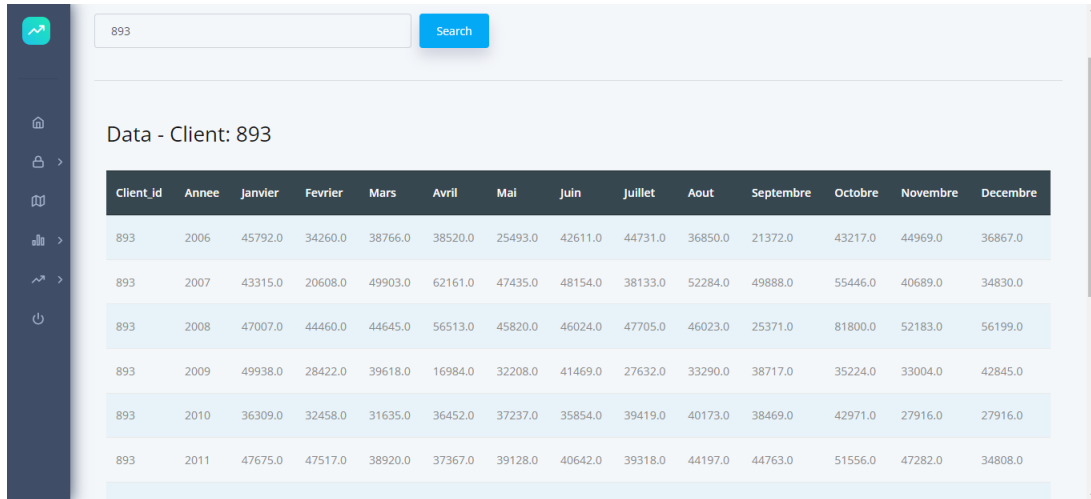


FIGURE 3.10 – Interface visualisation des ventes-Tableau.

- * Format graphique après avoir saisi l'id du client et l'année, illustré dans figure 3.11

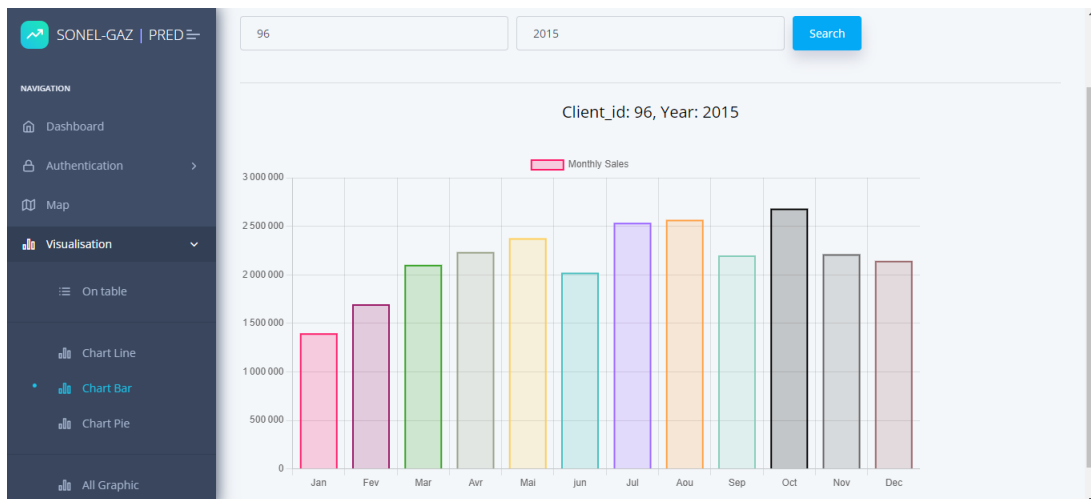


FIGURE 3.11 – Interface visualisation des ventes-Graphe.

3.4.2.5 Interface localisation des clients

La figure 3.12 représente l'interface qui permet de visualiser la localisation de quelques grands clients sur la carte géographique de la wilaya de Bejaïa :

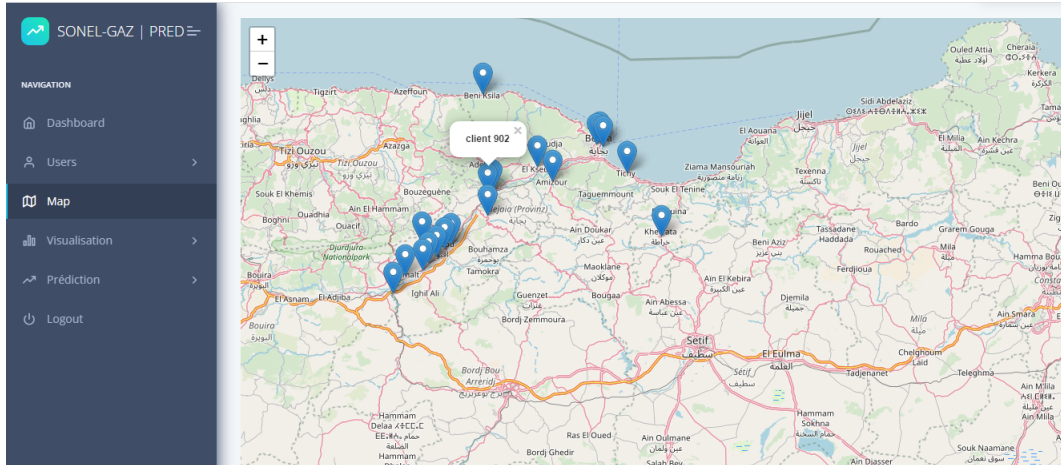


FIGURE 3.12 – Interface localisation des clients.

3.4.2.6 Interface prédiction des ventes

Cette interface permet de visualiser la prédiction de la consommation d'électricité des différents clients. L'utilisateur peut faire deux choix de prédiction :

- La prédiction SARIMA : à travers cette interface l'utilisateur pourra afficher la prédiction de la consommation des cinq grands clients HTA après avoir saisi l'id du client. cette interface est donnée dans figure 3.13

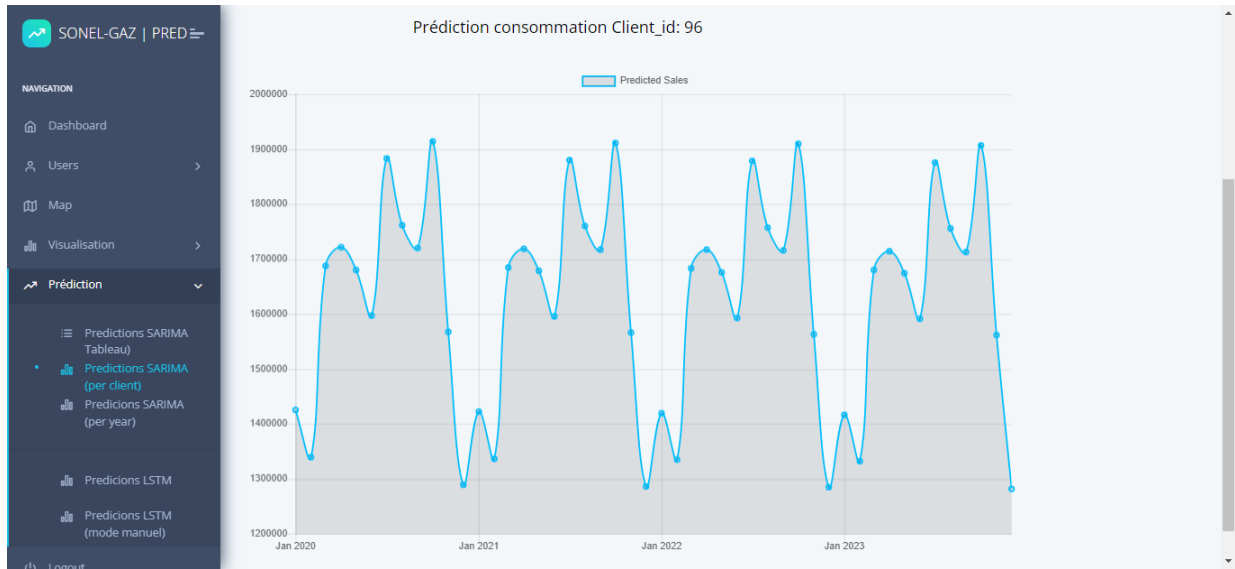


FIGURE 3.13 – Interface prédiction des ventes -SARIMA.

- La prédiction LSTM : dans cette interface l'utilisateur pourra afficher la consommation future de tous les clients HTA après avoir saisi l'id du client, comme illustré dans la figure 3.14

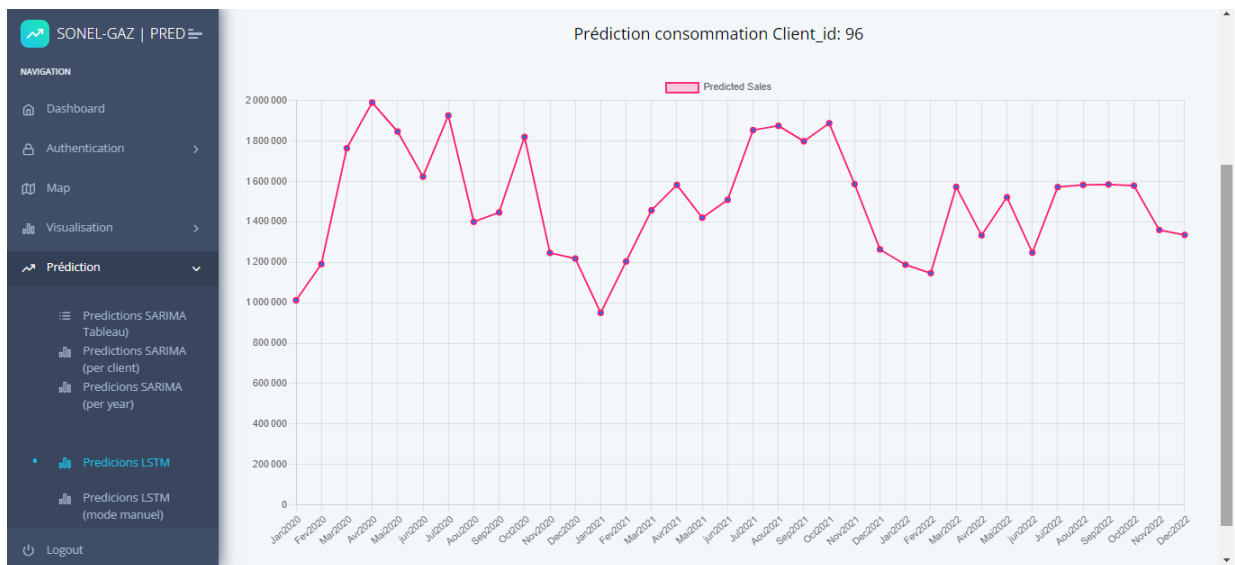


FIGURE 3.14 – Interface prédiction des ventes -LSTM.

3.5 Conclusion

Dans ce chapitre, nous avons décrit l'aspect pratique de notre projet, tout d'abord, nous avons donné les environnements matériels et logiciels qui nous ont permis de mener à bien ce travail et à le réaliser.

Ensuite, nous avons présenté les différentes interfaces de notre application ainsi que leurs comportements à travers cette réalisation.

Enfin, nous avons expliqué en détail les étapes de construction et modélisation des prédictions de ventes en utilisant les modèles de prévision de série temporelle en occurrence SARIMA et LSTM.

Conclusion générale et perspectives

Au cours de notre projet, nous avons conçu et mis en œuvre une application web pour la prévision et l'analyse des ventes électriques des clients HTA de Sonelgaz .

En effet, afin d'apporter une bonne aide à la prise de décision au sein du service des grands comptes, la Sonelgaz Distribution Direction de BEJAIA nous a proposé ce projet qui a pour thème : Conception et réalisation d'une application web d'analyse et de prédiction de ventes. Nous avons commencé par recenser l'objectif principal de résoudre la problématique que rencontre la Sonelgaz Distribution Direction de BEJAIA qui est la perte d'énergie ou la non satisfaction de client et le manque d'outil automatique qui permet de faire des prévisions sur les ventes autre que les méthodes manuelles et traditionnelles coûteuses en effort et en temps.

Par la suite, nous avons opté pour la méthode scrum pour la gestion de notre projet, les besoins fonctionnels sont définis par le product owner sous forme d'un product backlog que nous avons réparti en sprint. Nous nous sommes aussi appuyés sur le langage UML pour la modélisation des différents aspects et fonctionnalités de l'application.

Aussi, nous avons proposé deux modèles d'apprentissage automatique et profond pour pouvoir faire la prédiction de la consommation future de l'électricité des clients HTA que nous avons intégré dans notre système.

Enfin, en vue de valider notre système nous nous sommes réunies avec le product owner ou nous avons exposé notre travail et nous avons tester les fonctionnalités du système décrites dans le chapitre 1. La plupart de ces fonctionnalités correspondent à ces attentes.

Ce projet a fait l'objet d'une expérience très intéressante et enrichissante, car il nous a permis d'appliquer nos connaissances en informatique plus particulièrement dans le domaine de Génie Logiciel et d'acquérir de nouvelles connaissances, notamment dans le domaine de l'intelligence artificielle et la science des données. En outre, il nous a permis de nous familiariser avec certains outils de développement, ainsi qu'avec l'environnement de la vie professionnelle.

En perspective nous souhaitons développer de nouvelles fonctionnalités et les intégrer dans cette solution telle que la détection d'anomalie ,nous souhaitons aussi élargir notre système pour analyser, prédire les ventes des autres catégories de clients et classer les clients selon leur consommation.

BIBLIOGRAPHIE

- [1] <https://www.techno-science.net/glossaire-definition/Processus-autoregressif.html> consulter le 25/04/2022. 6
- [2] <https://www.zendesk.fr/blog/analyse-des-ventes> onsulter le 01/05/2022. 16
- [3] <https://www.1min30.com/dictionnaire-du-web/approche-systemique/> consulter le 30/06/2022. 17
- [4] <https://www.syloe.com/glossaire/application-web/> consulter le 08/05/2022. 21
- [5] <https://www.sonelgaz.dz> consulter le 25/05/2022. 24
- [6] <https://fr.vikidia.org/wiki/PyCharm/> consulter le 07/06/2022. 54
- [7] <https://openclassrooms.com/fr/courses/4452741-decouvrez-les-librairies-python-pour-la-data-science/> consulter le 30/06/2022. 55
- [8] <https://scikit-learn.org/> consulter le 30/06/2022. 55
- [9] R Adriano. Pandas : la bibliothèque python dédiée à la data science. 2022. 55
- [10] Anin Arthus. Séries temporelles. *SARIMA pour initiation aux prédictions temporelles*, 01 février 2020. 7
- [11] Gupta Aryan. Mean squared error. *Overview, Examples, Concepts and More*, 2021. <https://www.simplilearn.com/tutorials/statistics-tutorial/mean-squared-error>. 61
- [12] Simone Autiero et al. Computer science and informatics. 2021. 54
- [13] Chloé-Agathe Azencott. *Introduction au machine learning*. Dunod, 2019. 8, 9, 10
- [14] L Bastien. Tensorflow : le framework de machine learning de google. 2021. 55
- [15] Younes Benzaki. Introduction à l'algorithme k nearest neighbors (knn). *Mr. Mint : Apprendre le Machine Learning de A à Z*, 2, 2018. 10

-
- [16] Tim Berners-Lee and Dan Connolly. Rfc1866 : Hypertext markup language-2.0, 1995. 52
- [17] Samah Dendouga. Séries temporelles : Théorie et application. 2020. 4, 5
- [18] Younes Derfoufi. Programmation en langage python. 2019. 53
- [19] Pascal Roques et Franck Vallée. “UML 2 en action”. In : De l’analyse des besoins à la conception J2EE, 4ème édition Eyrolles, 2004. 29
- [20] Stephen Weston et Robert Bjornson. *Introduction to Anaconda*, 2016. 54
- [21] Benabbas Farouk. *méthodes heuristiques pour la prédiction des séries temporelles*. PhD thesis, PhD thesis, Universté Badji Mokhtar Annaba UBMA, 2012. 4
- [22] Alain Fernandez. Principe du client serveur. 22/03/2017. <https://www.piloter.org/techno/support/client-serveur.htm>. 21
- [23] Joseph Gabay and David Gabay. *UML 2 Analyse et conception : Mise en œuvre guidée avec études de cas*. Dunod, 2008. 47
- [24] Yannig Goude. Séries chronologique. *Les processus AR et MA*, 2020-2021. 6
- [25] Abbas Kh Ibrahim, Mohammad H Abdulwahab, Maiwan B Abdulrazzaq, and Mayyadah R Mahmood. A tree method for managing documents in mongodb. *vol*, 83 :18351–18359, 2020. 54
- [26] Paquet Judicael. Manifeste agile. *Valeurs agiles*, 12 aout 2018. 18
- [27] Sungil Kim and Heeyoung Kim. A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting*, 32(3) :669–679, 2016. 61
- [28] Jörg Krause. Css : Cascading style sheets. In *Introducing Web Development*, pages 65–86. Springer, 2016. 52
- [29] G Kurzawa and G Farges. Les clés pour démarrer avec l’agilité. 18, 19, 20
- [30] Vancappel Kévan. Machine learning. *comprendre c’est qu’est un réseau de neurone et en créer un*, 10 novembre 2020. 13
- [31] P Laura. Keras : tout savoir sur l’api de deep learning. 2021. 55
- [32] P Laura. Numpy : la bibliothèque python la plus utilisée en data science. 2021. 55
- [33] PS Lokhande, Fankar Aslam, Nabeel Hawa, Jumal Munir, and Murade Gulamgaus. Efficient way of web development using python and flask. 2015. 53
- [34] Renaud DE Maricourt. prévision des ventes. *8 V S5273 (10)*. <https://excerpts.numilog.com/books/9782130390138.pdf>. 17
- [35] Ghalia Merzougui. Apprentissage profond. *université Mustaphe Benboulaid de Batna*, 2020-2021. 11
- [36] Orpheo. Zeste de savoir. *Deep learning, c’est quoi ?*, 2019. <https://zestedesavoir.com/articles/pdf/1654/deep-learning-cest-quoi.pdf>. 11, 12, 13

-
- [37] Addy Osmani. *Learning JavaScript Design Patterns : A JavaScript and jQuery Developer's Guide*. " O'Reilly Media, Inc.", 2012. 53
- [38] Jaufert Philippe. Introduction aux réseau de neurone. *CNRS-Monpellier*, 28/12/2002. 12
- [39] Planzone. *quest-ce-que-la-methodologie-extreme-programming*. 2017. 18
- [40] Gilles Roy. *Conception de bases de données avec UML*. PUQ, 2007. 33, 35
- [41] Diego Sevilla Ruiz, Severino Feliciano Morales, and Jesús Garcia-Molina. An mde approach to generate schemas for object-document mappers. In *Proceedings of the 5th International Conference on Model-Driven Engineering and Software Development (MODELSWARD)*, pages 220–228, 2017. 51
- [42] Courdier Rémy. Analyse et conception objet du logiciel. *Approche Orientée Objet et UML*. 17
- [43] Guillaume Saint-Cirgue. Apprendre le machine learning en une semaine. *Machine Learnia, Londres*, 2019. 54, 60
- [44] Jake Spurlock. *Bootstrap : responsive web development*. " O'Reilly Media, Inc.", 2013. 53
- [45] José F Torres, Dalil Hadjout, Abderrazak Sebaa, Francisco Martínez-Álvarez, and Alicia Troncoso. Deep learning for time series forecasting : a survey. *Big Data*, 9(1) :3–21, 2021. 13, 14, 15
- [46] Thomas Vincent. A guide to time series forecasting with arima in python 3. *ARIMA Time Series Data Forecasting and Visualization in Python— DigitalOcean*, 2017. 7
- [47] Junping Wang, Quanshi Chen, and Yong Chen. Rbf kernel based support vector machine with universal approximation and its application. In *International symposium on neural networks*, pages 512–517. Springer, 2004. 10
- [48] Tony Yiu. Understanding random forest. Aug. 2019. 11

Résumé

L'émergence de l'intelligence artificielle ces dernières années notamment le Machine Learning (ML) et le Deep Learning (DL) motive les entreprises à intégrer ces nouvelles technologies dans le cadre de l'amélioration de la gestion et de faire face aux différentes problématiques pratiques telles que répondre aux besoins des clients en électricité. C'est dans ce contexte que s'introduit le travail présenté dans ce mémoire qui consiste à la conception et à la réalisation d'une application web pour l'analyse et la prédiction des ventes au sein de la Sonelgaz Distribution Direction de BEJAIA en utilisant les techniques d'Intelligence Artificielle. Notre application permet de faire principalement la visualisation et la prédiction de ventes avec deux modèles de prévision des séries temporelles qui ont fait leurs preuves dans le domaine, à savoir : Les réseaux neuronaux à mémoire à long court terme et la méthode statistique Seasonal AutoRegressive Integrated Moving Average. Ce travail a été réalisé en suivant la méthode Scrum, une méthode Agile de développement et de gestion de projet. Il a été expérimenté avec près de 1699 clients et 14 ans de consommation mensuelle d'électricité de BEJAIA. Concernant les résultats, nous avons obtenu un taux d'erreur de 26% pour modèle LSTM et 45% pour modèle SARIMA.

Mots clés : Intelligence Artificiel, Machine Learning, Deep Learning, Analyse et Prédiction des ventes énergies, série temporelles, Méthode agile.

Abstract

The emergence of artificial intelligence in recent years, particularly Machine Learning (ML) and Deep Learning (DL) motivates companies to integrate these new technologies in the technologies in the context of improving management and dealing with various practical practical problems such as meeting the needs of customers in electricity. It is in this context that is introduced the work presented in this thesis which consists of the design and implementation of a web application for the analysis and prediction of sales within the Sonelgaz Distribution Direction of BEJAIA using the techniques of Artificial Intelligence. Our application makes it possible to make mainly the visualization and the prediction of sales with two models of prediction of the temporal series which were proved in the field, namely : Long Short Term Memory Neural Networks and the Seasonal AutoRegressive Integrated Moving Average statistical method. This work was carried out following the Scrum method, an Agile method of development and project management. It has been experimented with nearly 1699 customers and 14 years of monthly electricity consumption in BEJAIA. Concerning the results, we obtained an error rate of 26% for the LSTM model and 45% for the SARIMA model.

Key words : Artificial Intelligence, Machine Learning, Deep Learning, Energy sales analysis and prediction, Time series, Agile method.