

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES

VERS L'OPTIMISATION DE L'OPÉRATION D'UN RÉSEAU DE DISTRIBUTION :
PRÉVISION DE LA CONSOMMATION DU SYSTÈME ÉLECTRIQUE DE LA VILLE
DE BAIE-COMEAU

MÉMOIRE PRÉSENTÉ
COMME EXIGENCE PARTIELLE DE LA
MAÎTRISE EN GÉNIE ÉLECTRIQUE

PAR
AYOUB HAOUFADI

SEPTEMBRE 2022

Université du Québec à Trois-Rivières

Service de la bibliothèque

Avertissement

L'auteur de ce mémoire, de cette thèse ou de cet essai a autorisé l'Université du Québec à Trois-Rivières à diffuser, à des fins non lucratives, une copie de son mémoire, de sa thèse ou de son essai.

Cette diffusion n'entraîne pas une renonciation de la part de l'auteur à ses droits de propriété intellectuelle, incluant le droit d'auteur, sur ce mémoire, cette thèse ou cet essai. Notamment, la reproduction ou la publication de la totalité ou d'une partie importante de ce mémoire, de cette thèse et de son essai requiert son autorisation.

UNIVERSITÉ DU QUÉBEC À TROIS-RIVIÈRES
MAÎTRISE EN GÉNIE ÉLECTRIQUE (M. Sc. A.)

Direction de recherche :

Prof. Alben Cardenas, Ph. D., directeur de recherche

Prof. Sergio Raul Rivera, Ph. D., codirecteur de recherche

Jury d'évaluation

Prof. Ahmed Chériti, Ph. D., Évaluateur interne

Prof. Tahar Tafticht, Ph. D., Évaluateur externe

Prof. Alben Cardenas, Ph. D., Directeur de recherche

Résumé

Canada connaît des températures basses en hiver avec une température annuelle moyenne d'environ 1°C et des températures minimales inférieures à -30°C, ce qui pose un problème pour sécuriser l'énergie électrique propre nécessaire au chauffage sans préjudice des sources d'énergie fossiles. Il faut souligner que le Canada est signataire de plusieurs accords environnementaux, dont le plus célèbre est l'Accord de Paris sur le climat avec la participation de la plupart des pays menés par les membres du G-20. Grâce aux nombreuses recherches qui ont été faites dans le domaine sur le lien entre la température et la consommation, « la commande intelligente des ressources énergétiques » dans le réseau de distribution en conditions climatiques extrêmement froides est apparue comme nouvelle approche. Celle-ci permet la commande adaptative par intelligence artificielle de la charge flexible, de la production distribuée et du stockage dans les réseaux de distribution.

À ce sujet et en partenariat avec la Ville de la Baie-Comeau, il a été décidé de mener une étude de recherche pour améliorer les performances et l'efficacité du réseau électrique, en permettant aux gestionnaires de réseau d'équilibrer l'approvisionnement énergétique de la demande. L'étude est motivée par des raisons économiques telles que réduire la facture énergétique, y compris pour des raisons techniques telles que la prévision des périodes de pointe de consommation en hiver pour réduire le facteur d'utilisation. Nous avons utilisé des informations sur le réseau de la ville comme le nombre de lignes de distribution et le type de consommateurs dans chaque ligne, en plus des données de consommation d'électricité comme la puissance d'apparente, la puissance active, la puissance réactive, la tension et le courant. Nous avons utilisé également des données météorologiques de la ville afin de faire

le processus de corrélation entre la puissance et la température pour choisir la méthode d'intelligence artificielle à utiliser. Le choix s'est porté sur les méthodes d'analyse de séries chronologiques, spécifiquement les modèles ARMAX, ARX et Box-Jenkins. Nous avons comparé les performances en calculant l'erreur absolue moyenne du résultat de la prévision dynamique de 24 heures avec la consommation réelle pour chaque modèle. Cette étude de comparaison de modèles a été réalisée grâce à MATLAB. Nous avons employé également Python pour réaliser des validations des algorithmes de prévision et pour proposer une interface pour la prévision en temps réel à partir des informations météorologiques disponibles en ligne. Bien entendu la ville attend une interface graphique pratique et simple d'utilisation au niveau de l'affichage de prévision de l'énergie qui sera consommée et de l'heure de pointe pour bien interagir, la validation expérimentale a été effectuée en utilisant les moyennes disponibles comme l'utilisation d'une API gratuite pour extraire la température future ainsi que le meilleur modèle de prédiction identifié dans la comparaison effectuée, cette fois en utilisant Python avec des bibliothèques certifiées.

Ce mémoire nous a permis d'explorer en simulation et en implémentation temps réel la prévision de la consommation, et ouvre les portes pour des travaux plus poussés dans le domaine de la gestion intelligente de la charge flexible dans les réseaux de distribution.

Remerciement

Par ce mémoire, qui clôt deux années d'études, je tiens tout d'abord à présenter mes vifs remerciements à tous ceux qui ont contribué de près ou de loin à l'élaboration de ce modeste travail de recherche.

Surtout les professeurs du Département de génie électrique et informatique de l'Université du Québec à Trois-Rivières (UQTR) et en particulier le Pr Alben Cardenas, membre du Groupe de Recherche en Électronique Industrielle et de l'Institut de recherche sur l'hydrogène qui n'a ménagé aucun effort pour s'assurer que cette recherche est à la hauteur, mes remerciements à monsieur le co-directeur Sergio Raul Rivera est professeur au département de génie électrique et génie électronique à Université Nationale de Colombie (UNAL) et membre du groupe EMCUN; et aussi expert dans le domaine des systèmes de puissance et de l'intelligence computationnelle.

Je tiens à remercier l'UQTR qui a financé ce projet grâce au Programme d'aide à l'internationalisation de la recherche (PAIR), et la Ville de Baie-Comeau en la personne du directeur adjoint Stephen Caron pour nous avoir donné son accord pour nous fournir des données techniques nécessaires pour la réalisation de cette recherche.

Un merci tout particulier au Groupe de Recherche en Électronique Industrielle et à l'Institut de Recherche sur l'Hydrogène et à tous les personnels qui veillent sur la recherche et les jeunes chercheurs.

Sans oublier pour autant de remercier tout le personnel de l'Université de Québec à Trois-Rivières.

Table des matières

Résumé.....	i
Remerciement	iii
Table des matières.....	v
Liste des tableaux.....	ix
Liste des figures	x
Liste des symboles	xii
Chapitre 1 - Introduction.....	1
1.1 Introduction	1
1.2 Contexte et Problématique	3
1.3 Objectif du mémoire.....	9
1.4 Méthodologie.....	10
1.5 Organisation du mémoire	11
Chapitre 2 - Optimiser le réseau de distribution électrique grâce à l'intelligence artificielle : l'état de l'art	12
2.1 Introduction	12
2.2 C'est quoi l'optimisation de réseau de distribution?	13

2.3	L'état de l'art sur les méthodes de l'optimisation électrique	13
2.4	Les horizons de la prévision.....	15
2.5	L'état de l'art sur les méthodes de prévision.....	18
2.6	Synthèse du chapitre.....	24
Chapitre 3 - Analyse préliminaire et développement des modèles.....		26
3.1	Introduction	26
3.2	Le coefficient de Hurst et la prévisibilité	27
3.3	Modélisation des séries chronologiques.....	29
3.3.1	Processus stochastique	29
3.3.2	Autocovariance et autocorrélation	30
3.3.3	Stationnarité	32
3.3.4	Bruit blanc.....	33
3.3.5	Opérations sur les séries.....	34
3.4	Processus aléatoires stationnaires.....	37
3.4.1	Modèle autorégressif: processus AR	37
3.4.2	Modèle à moyenne mobile processus MA.....	37
3.5	Identification du système.....	39
3.6	Analyse théorique des méthodes de prévision.....	40
3.6.1	Les modèles ARX et ARMAX	40

3.6.2	Modèle Box Jenkins.....	42
3.7	Analyse d'erreur de prévision	44
3.7.1	Calcul du pourcentage d'erreur	45
3.7.2	Erreur absolue moyenne MAE.....	46
3.8	Synthèse du chapitre.....	46
Chapitre 4 - Collecte et analyse des données		48
4.1	Introduction	48
4.2	Description de réseau de la ville Baie-Comeau.....	48
4.2.1	Les types de consommateurs	49
4.2.2	Données relatives aux clients.....	51
4.2.3	Données relatives à la consommation par la ligne.....	51
4.3	Analyse des données.....	52
4.3.1	Les données de la consommation électrique.....	52
4.3.2	Les données météorologiques	54
4.3.3	Corrélation entre les variables	55
4.3.4	Analyse des pics de consommation hivernale	62
4.3.5	Étude de la prédictibilité des données.....	64
4.4	Synthèse du chapitre.....	65
Chapitre 5 - Proposition et analyse des résultats.....		67

5.1	Introduction	67
5.2	Proposition de méthode de prévision en vue de l'optimisation.....	67
5.3	Prévision de la consommation.....	69
5.3.1	Méthodologie et base de données	69
5.3.2	Résultats et discussion	73
5.4	Validation expérimentale.....	80
5.4.1	Implémentation de modèle.....	81
5.4.2	Interface de programmation d'application (API).....	85
5.4.3	Interface graphique	87
5.5	Synthèse du chapitre.....	88
Chapitre 6 - Conclusion et recommandations		89
6.1	Conclusion générale	89
6.2	Recommandation.....	90
RÉFÉRENCES.....		92
Annexe A – Boucle de prévision de consommation d'électricité pour toutes les lignes de distribution.....		98
Annexe B – Résultats de la prévision obtenus lors de l'exécution sous Python.....		111
Annexe C – Code de programmation sous MATLAB et Python		117

Liste des tableaux

Tableau 4-1: Facteur de crête pour le jour et le soir.	64
Tableau 4-2 : Le coefficient de Hurst	65
Tableau 5-1: Les ordres polynomiaux des modèles	73
Tableau 5-2: MAE obtenue avec le modèle ARMAX pour toutes les lignes.	78
Tableau 5-3: MAE de modèle ARX pour toutes les lignes.....	79
Tableau 5-4: MAE de modèle Box-Jenkins pour toutes les lignes.	80
Tableau 5-5 : MAE pour les trois méthodes en Python.	84

Liste des figures

Figure 1-1 : Comparaison de la consommation d'énergie par habitant au Québec avec d'autres pays, 2016 [16].	5
Figure 1-2: Consommation de différentes formes d'énergie par secteur d'activité au Québec, 2016 [17].	7
Figure 1-3 : Évolution des appareils électriques par ménage entre 1990 et 2013 [18].	8
Figure 2-1: Les horizons de prévision.....	17
Figure 3-1: Approche d'identification du système [52].	39
Figure 3-2: Système à entrées multiples et à sortie unique (MISO) [53].....	40
Figure 3-3: Modèle paramétrique de ARX (a), ARMAX (b) et Box Jenkins (c) [58].	44
Figure 4-1: Consommation de la Ville Baie-Comeau par ligne et par client.....	50
Figure 4-2: Répartition de la demande en puissance de la Ville Baie-Comeau par client ou secteur.....	51
Figure 4-3: Consommation de la Ville Baie Comeau par ligne.	52
Figure 4-4: Données réelles de la consommation électrique.	53
Figure 4-5: Courbes de la consommation électrique après l'élimination de données aberrantes.....	54
Figure 4-6: Données météorologique de la ville Baie-Comeau.	55
Figure 4-7: Interprétation du coefficient de corrélation [65].	55
Figure 4-8: Corrélation entre la puissance et les variables pour la ligne 1.	56
Figure 4-9: Corrélation entre la puissance et les variables pour la ligne 2.	57
Figure 4-10: Corrélation entre la puissance et les variables pour la ligne 3.	58
Figure 4-11: Corrélation entre la puissance et les variables pour la ligne 4.	58

Figure 4-12: Corrélation entre la puissance et les variables pour la ligne 5.	59
Figure 4-13: Corrélation entre la puissance et les variables pour la ligne 6.	60
Figure 4-14 : Corrélation entre la puissance et les variables pour la ligne 7.	61
Figure 4-15: Corrélation entre la puissance et les variables pour la ligne 8.	62
Figure 4-16: Analyse des pics de la consommation.....	63
Figure 5-1: L’approche générale du projet.....	68
Figure 5-2 : Réseau de distribution de la ville Baie-Comeau.	69
Figure 5-3: Schéma approximatif de programmation.	70
Figure 5-4: Fonctionnement de programmation.	71
Figure 5-5: Schéma d'entrée et de sortie du modèle.	72
Figure 5-6: Prévion de la consommation pour toutes les lignes.....	74
Figure 5-7: Plan de programmation pour la validation expérimentale.	81
Figure 5-8: Prévion de la consommation pour le premier février 2020.....	84
Figure 5-9: Utilisation d'API pour la prévion de la consommation.	85
Figure 5-10 : Prévion de consommation avec API.....	86
Figure 5-11: Interface graphique.....	87

Liste des symboles

AR : Modèle autorégressif

ARMA : Modèles autorégressifs et moyenne mobile

ARIMA : Moyenne mobile intégrée autorégressive.

ARMAX : Modèle de moyenne mobile autorégressive avec entrées exogènes.

ARX : Autorégressive exogène.

ANN : Réseaux de neurones artificiels.

BJ : Box et Jenkins.

IA : Intelligence artificielle.

LTLF : long-term load forecasting.

LM : Modèle régression linière.

LNREG : La régression linière multiple.

MA : Modèle à moyenne mobile.

MAE : Erreur moyenne absolue.

MAPE : Erreur moyenne absolue en pourcentage.

MaxAPE : Erreur maximale absolue en pourcentage.

MSE : Erreur quadratique moyenne.

MTLF : Medium term load forecasting.

PSO : Optimisation par essaims particulaires.

RBF : Fonctions de base radiales

RMSE : La racine de l'erreur quadratique moyenne.

RSREG : La régression de surface de réponse.

SVM : Régression de machine vectorielle du support.

STLF : Short term load forecasting.

VSTLF : Very short term load forecasting.

Chapitre 1 - Introduction

1.1 Introduction

De nos jours, s'appuyer sur l'intelligence artificielle est devenu une nécessité incontournable. De ce point de vue, les industries de production d'énergie et les villes sont impliquées dans ce domaine notamment dans la gestion et la planification, à travers l'anticipation de la demande pour minimiser et optimiser la consommation énergétique. La prévision de la demande en électricité est un domaine fertile et vital pour les scientifiques, d'autant plus que les acteurs économiques et politiques soutiennent le domaine en raison de ses bénéfices directs et indirects sur l'environnement et l'économie.

Environnementalement, une gestion adéquate de la consommation contribue à réduire les émissions gazeuses nocives pour l'homme et les animaux, et ces émissions proviennent de la combustion de pétrole et d'autres sources fossiles. Il s'agit à la base de prévoir la consommation pour connaître le moment approprié pour faire fonctionner les centrales électriques et pour mettre en action des stratégies pour optimiser l'utilisation des ressources.

Sur le plan économique, les points positifs sont plus nombreux que les points négatifs, car on réduit le coût de production d'énergie électrique et on contribue à créer des opportunités d'emploi pour les jeunes; sans négliger l'impact sur l'économie générale, ce qui augmente le taux de croissance économique.

Dans ce contexte, la ville de Baie-Comeau n'est pas à l'abri, étant elle-même l'opérateur de son réseau électrique sur le secteur Marquette. Elle a donc intérêt à rationaliser les dépenses en utilisant cette technologie prometteuse.

Cela nous amène à parler de l'évolution de la consommation d'énergie électrique, c'est-à-dire de l'évolution des budgets financiers de la ville, car ces dernières années, nous avons assisté à une augmentation significative de la facture d'électricité, à l'aide de rapports financiers [1]–[8] pour la ville de Baie-Comeau avec une facture comprise entre 11,3 et 18,8 milliard de dollar pour les années 2013 à 2020, soit une augmentation de 65%. À noter qu'elle sécurise ses besoins énergétiques grâce aux centrales hydroélectriques du bassin hydrographique de la rivière Manicouagan.

La rivière Manicouagan contient quatre barrages hydroélectriques. Ceux de McCormick et Manic-1 partagent le même réservoir à l'entrée de la Baie-Comeau, ont respectivement des capacités installées de 235 MW et 184 MW. La centrale McCormick est détenue conjointement par Hydro-Québec et Alcoa. Dans le même secteur, il y a la centrale Jean-Lesage (Manic-2) au nord de Baie-Comeau avec 1 229 MW, la centrale René Lévesque (Manic-3) d'une capacité de 1 326 MW et le complexe Manic-5 avec le barrage Daniel-Johnson, d'une capacité totale de 2 660 MW [9].

On sait aussi que la ville est responsable de la distribution et de la facturation de l'électricité du secteur Marquette, tandis que le secteur Mingan est desservi par Hydro-Québec [10]. Étant un opérateur de réseau électrique, la Ville de Baie-Comeau est membre de l'Association québécoise de redistribution de l'électricité (AREQ) [11].

1.2 Contexte et Problématique

Comme mentionné plus tôt, la consommation d'énergie électrique est devenue un enjeu pour les producteurs et les distributeurs surtout dans les conditions de température extrêmes. À partir de là, nous discuterons qui sont les plus importants distributeurs et producteurs d'électricité dans la province de Québec. Hydro-Québec est un acteur majeur dans la région, fournissant de l'électricité à une population de 8,3 millions de personnes selon le dernier recensement de 2016, ce qui représente environ 23 % des Canadiens et contribue ainsi à 4,1 % de l'électricité produite au Canada. Son rôle se limite à approvisionner le marché local et commercialise ses surplus sur les marchés de gros [12], [13].

Cela nous amène à parler des sources d'énergie produites par Hydro-Québec d'après l'étiquette, il est clair que la région dépend principalement de l'hydroélectricité, car la région de Québec possède d'énormes ressources hydrauliques. D'ailleurs, plus de 40 % des réserves d'eau du Canada résident au Québec [12].

Malgré sa dépendance totale vis-à-vis de l'hydroélectricité, au cours des deux dernières décennies, l'énergie éolienne a commencé à émerger comme le confirment les données. Environ 4 % de l'énergie produite au Québec est de source éolienne, avec une capacité de production de 4 096 MW, un pourcentage qui continue d'augmenter, car Hydro-Québec a des ententes avec des producteurs indépendants [14],[12].

Il y a un problème dont souffre la région, qui se reflète dans la demande de pointe pendant la saison hivernale, cette contrainte oblige Hydro-Québec à utiliser l'énergie provenant d'autres sources dont la biomasse, le gaz naturel et le diesel [14].

Hydro-Québec est responsable de la majeure partie de l'énergie produite au Québec, puisque 62 centrales hydroélectriques sont maintenues par la société d'État, sans oublier les producteurs indépendants [14]. C'est le cas de l'Association de redistribution de l'électricité du Québec (AREQ), une association composée de neuf (9) municipalités en plus de la Coopérative St-Jean-Baptiste, les municipalités fondatrices de cette association se nomment Alma, Amos, Baie-Comeau, Coaticook, Joliette, Magog, Saguenay, Sherbrooke et Westmount. Cette association a été fondée en 1990 afin d'offrir de bons services à ses clients et de concurrencer Hydro-Québec dans la qualité de la production et de la distribution, cela n'empêche pas les villes de faire affaire avec eux. Pour rappel, l'association mise aussi sur l'hydroélectricité [11].

À notre époque, la consommation d'énergie électrique est devenue associée au comportement humain et aux équipements de consommation, car l'homme est par nature curieux et veut expérimenter tout ce qui se reflète dans tout ce qui l'entoure.

Au Québec, comme dans d'autres villes et pays du monde la consommation d'énergie électrique a augmenté d'une façon significative. Entre début 2012 et fin 2013, une augmentation de 2.9% a été observée. En 2012, les Québécois ont consommé une énergie équivalente à 172,5 milliard de kWh [15]. C'est pourquoi nous jetterons un rapide coup d'œil sur les résultats obtenus dans le reste du monde, qui se caractérise par un climat et une nature similaire à celui du Québec.

Le diagramme de la figure 1-1 fourni par l'association des véhicules électriques du Québec (AVEQ) montre que la consommation au Québec dépasse celle des pays comme la Norvège et la Suède (des pays scandinaves). Fait remarquable, la consommation moyenne par habitant au Québec se rapproche de celle des États-Unis et du Canada par une

consommation de 193 GJ. La Figure 1.1 expose la comparaison de la consommation d'énergie par habitant au Québec avec d'autres pays, en 2016.

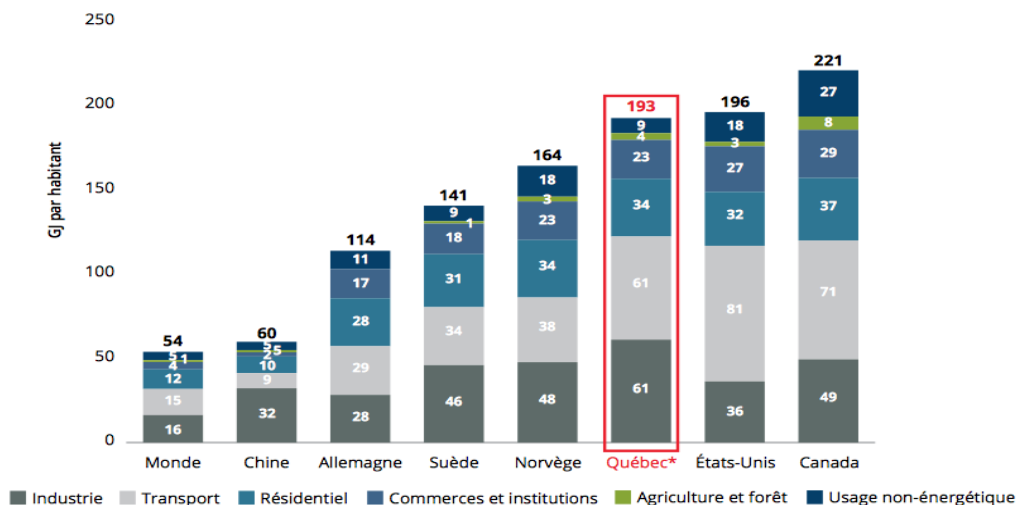


Figure 1-1 : Comparaison de la consommation d'énergie par habitant au Québec avec d'autres pays, 2016 [16].

Le graphique montre la réalité des secteurs qui se chevauchent dans la consommation d'énergie de diverses natures, comme le charbon, le gaz naturel, les produits pétroliers raffinés et les biocarburants, etc., qui à leur tour sont touchés par l'augmentation de la consommation pendant le pic de l'hiver. À la figure 1.2, nous présentons la consommation des différentes formes d'énergie par secteur d'activité au Québec, 2016 [17].

Au Québec, le secteur industriel occupe la plus grande part de la consommation d'énergie, car l'énergie consommée, la plus grande partie va à l'énergie électrique avec 48%, 26% correspondent au gaz naturel, 14% aux biocarburants, 2% au charbon et 8 % aux produits pétroliers raffinés.

Quant au transport, il se classe au deuxième rang après le secteur industriel pour la consommation d'énergie, à 97 % pour les produits pétroliers raffinés et à 3 % pour les biocarburants, le gaz naturel, l'électricité et le gaz naturel liquéfié.

Le secteur résidentiel occupe le troisième rang en termes de consommation d'énergie avec 73 % d'électricité, 13 % de biocombustible, 8 % de gaz naturel et 6 % de produits pétroliers raffinés.

Les centres commerciaux et les institutions occupent le quatrième rang en termes de consommation : 42% d'électricité, 37% de gaz naturel, 17% de produits pétroliers raffinés et 3% de liquide de gaz naturel.

Le graphique montre l'utilisation non énergétique d'un monopole complet sur les produits pétroliers raffinés.

Le secteur agricole est un secteur vital qu'il ne faut pas négliger. Il consomme également des énergies de toutes sortes. La part la plus grande revient aux produits pétroliers raffinés, puisqu'elle est estimée à 63 %, tandis que le reste se répartit comme suit : 19 % de l'électricité, 14% de gaz naturel liquéfié et 3% de gaz naturel.

Donc nous comprenons que la consommation d'énergie électrique domine dans les industries, au niveau résidentiel, commercial et institutionnel.

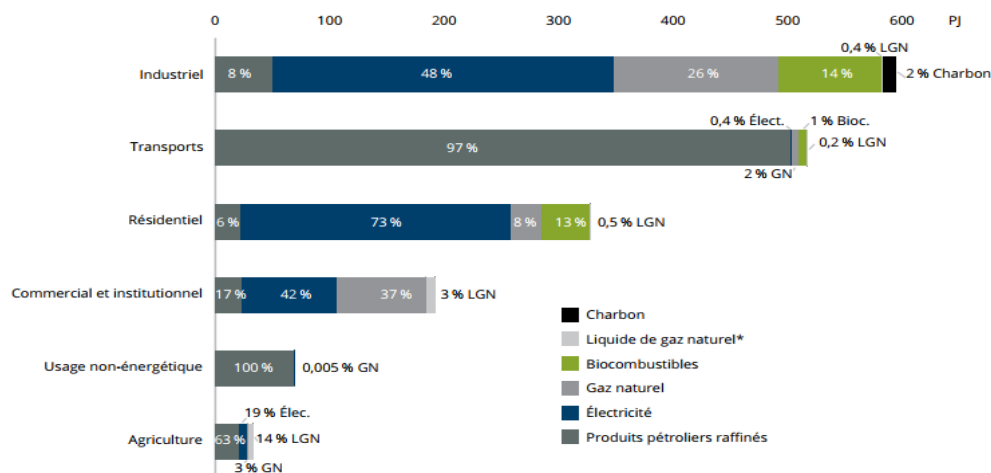


Figure 1-2: Consommation de différentes formes d'énergie par secteur d'activité au Québec, 2016 [17].

Récemment, on assiste à un développement incomparable en termes d'innovation technologique, les fabricants d'appareils électriques augmentent leur production et leurs profits en attirant de nouveaux clients ce qui a de répercussions directes sur l'environnement. Ce qui est confirmé par le rapport sur le développement de l'efficacité énergétique au Canada 1990-2013 [18]. Selon le rapport, le Canada a connu une croissance en nombre de familles passant de 9,9 millions à 13,8 millions, ce qui a entraîné aussi un changement quant au nombre de membres des familles, c'est-à-dire une famille composée de 2.8 membres en 1990, et en 2013 les familles sont devenues composées de 2.6 membres. Mise à part l'augmentation du nombre de ménages, le nombre d'équipements électriques a passé de 15 à 22 appareils par foyer. La Figure 1.3 présente pour plus détails un graphique illustrant ce changement démographique qui peut être corrélé d'une certaine manière avec les changements dans les profils de consommation.

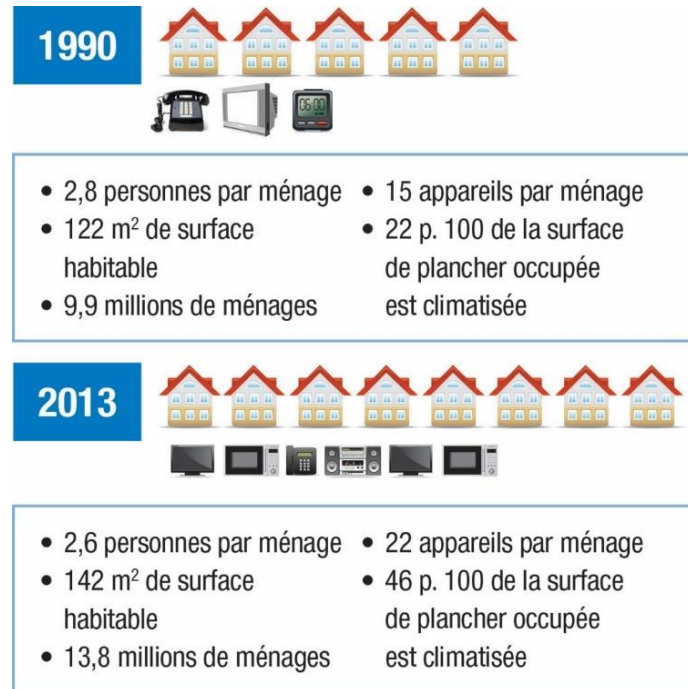


Figure 1-3 : Évolution des appareils électriques par ménage entre 1990 et 2013 [18].

Cela nous conduit à connaître les acteurs impliqués dans cette consommation électrique pour le Québec, qui est divisé en 17 régions administratives : Bas-Saint-Laurent, Saguenay-Lac-Saint-Jean, Capitale-nationale, Mauricie, Estrie, Montréal, Outaouais, Abitibi-Témiscamingue, Côte-Nord, Nord-du-Québec, Gaspésie, les Îles-de-la-Madeleine, Chaudière-Appalaches, Laval, Lanaudière, Laurentides, Montérégie et Centre-du-Québec. Toutes ces régions administratives constituent les principales artères qui alimentent la province en termes de stimulation de la production et de la consommation d'énergie. Tout ce qui compte, c'est de connaître les caractéristiques énergétiques de la Côte-Nord, région à laquelle appartient la ville de Baie-Comeau. Nous discuterons donc de l'évolution de la consommation électrique dans cette région.

En 2008, un rapport a été préparé par le Conseil régional de l'environnement de la Côte-Nord indiquant que l'électricité se classe au premier rang en termes de consommation

d'énergie dans les secteurs suivants : 68,9 % du secteur résidentiel, 55 % du secteur industriel et 48,2 % du secteur commercial. Quant au secteur des transports, les hydrocarbures représentent 98,7 % de la part [19].

Entre 1994 et 2011, le nombre d'abonnés a considérablement augmenté, ce qui a produit une augmentation de la consommation, passant de 244 mégawattheures à 377 mégawattheures, avec un taux de croissance annuel de 2,8 %. Notez que ces statistiques excluent la ville Baie-Comeau, car elle est membre de l'association AREQ [19].

L'AREQ comprend 156 000 clients répartis entre neuf communes et coopératives électriques, consommant de l'énergie électrique estimée à 1 000 mégawatts, en haute saison hivernale, obtenue à partir de 13 petites centrales électriques [11].

Ainsi, le problème général de cette mémoire est de trouver un modèle pour anticiper la demande future d'électricité dans des conditions extrêmes. Ceci dans le but de faciliter la mise en œuvre des stratégies pour améliorer le réseau de distribution d'électricité.

1.3 Objectif du mémoire

Le but de ce projet de recherche est de proposer une méthode de prévision de la consommation pour un réseau de distribution et de l'appliquer au cas spécifique du réseau opéré par la Ville de Baie-Comeau.

Les objectifs spécifiques sont les suivants :

1. Analyser les données de consommation d'électricité du réseau de distribution opéré par la Ville de Baie-Comeau afin de modéliser les profils de puissance;
2. Proposer des méthodes de prévision de la consommation pour un réseau de distribution, et les adapter au cas spécifique de la Ville de Baie-Comeau;

3. Implémenter et valider les méthodes proposées;

À long terme, cette prévision, utilisant des méthodes d'intelligence artificielle, permettrait de mettre en place une stratégie d'optimisation du facteur d'utilisation des ressources énergétiques du réseau.

1.4 Méthodologie

Pour atteindre l'objectif de ce projet, nous devons commencer par revoir la bibliographie de l'utilisation de l'intelligence artificielle pour améliorer le fonctionnement du réseau de distribution électrique, ce qui permettra d'identifier des directions de recherche dans ce domaine. En plus de définir le contexte du problème et d'analyser les caractéristiques du réseau et de la consommation d'électricité, nous allons procéder à une analyse des données afin de modéliser le comportement selon les différentes lignes (ou artères) du réseau de distribution. Cette étude permettra de cibler des méthodes de prévision adaptées au cas spécifique de la Ville de Baie-Comeau. Par la suite, le choix final d'une méthode dépendra du compromis entre la performance et la complexité pour son implémentation. Nous allons encore valider les propositions utilisant des données réelles fournies par la Ville de Baie-Comeau et des modèles qui seront développés sous MATLAB et/ou Python.

Dans cette étude nous allons considérer les conditions particulières du réseau de la Ville de Baie-Comeau et des conditions météorologiques propres de la région pour les intégrer dans la formulation des méthodes de prévision de la demande d'électricité.

Cette étude est réalisée dans le cadre du projet « *Commande adaptative par intelligence artificielle de la charge flexible, de la production et du stockage décentralisés dans les réseaux de distribution* » financé en partie par le Programme d'aide à l'internationalisation

de la recherche (PAIR) de l'UQTR et réalisé en coopération avec UNAL en Colombie (<https://unal.edu.co/>), NERGICA (<https://nergica.com/>) et la Ville de Baie-Comeau (<https://www.ville.baie-comeau.qc.ca/accueil/>).

1.5 Organisation du mémoire

L'étude de recherche comprend cinq chapitres organisés comme suit :

Le deuxième chapitre présente une recherche bibliographique sur l'optimisation du réseau de distribution à l'aide de l'intelligence artificielle notamment dans la prévision de la consommation électrique.

Le chapitre trois fournit une analyse préliminaire des méthodes de prévision classiques utilisant les modèles de séries chronologiques.

Le chapitre quatre traite de la description du réseau électrique de la ville de Baie-Comeau; il présente également une explication détaillée des données liées à la consommation électrique de chaque ligne de distribution.

Le cinquième chapitre tend à faire une comparaison des résultats de la prévision de la consommation susceptible d'être obtenue en utilisant l'analyse de séries temporelles; le chapitre comprend également la vérification expérimentale des modèles sélectionnés en utilisant la température prévue à l'aide d'une API (Application Programming Interface) gratuite.

Enfin, au chapitre six, une conclusion générale de la recherche, et quelques recommandations sont proposées.

Chapitre 2 - Optimiser le réseau de distribution électrique grâce à l'intelligence artificielle : l'état de l'art

2.1 Introduction

À notre époque, nous assistons à un développement sans précédent en termes de développement technologique, notamment la révolution de l'intelligence artificielle (IA). À cet égard, le secteur de l'énergie électrique peut s'appuyer sur cette technologie pour l'aider à développer de nouvelles méthodes et améliorer la gestion des ressources énergétiques. À noter que l'intelligence artificielle (IA) est à la base des théories et des techniques utilisées pour produire des machines capables de simuler l'intelligence humaine [20]. L'intelligence artificielle s'appuie sur des méthodes mathématiques et statistiques pour permettre aux ordinateurs d'apprendre à partir de données précédemment collectées [21]. Il existe un champ d'études appelé apprentissage automatique ou apprentissage statique (en anglais : Machine Learning) qui utilise la puissance des ordinateurs disponibles pour mettre en œuvre des mécanismes d'apprentissage. Les algorithmes d'apprentissage automatique sont généralement classés en trois types : apprentissage supervisé ; apprentissage non supervisé ; et l'apprentissage par renforcement [22]. L'apprentissage profond à son tour, il appartient à l'apprentissage automatique, il est largement appliqué à la fois à la régression et à la classification dans les bâtiments et les systèmes énergétiques [23]. L'étude que nous menons vise à améliorer et optimiser la consommation d'énergie électrique dans des conditions extrêmes. Il faut d'abord utiliser les données collectées les années précédentes pour prévoir

la consommation pour cette période de l'année en fonction de la météo et du calendrier. Comme nous l'avons mentionné dans le premier chapitre, les pays nordiques souffrent d'une baisse de température qui atteint parfois -30°C , plus particulièrement le Canada et la province de Québec. De ce point de vue, nous devons trouver des moyens pratiques pour résoudre le problème à partir d'une revue de la littérature scientifique.

2.2 C'est quoi l'optimisation de réseau de distribution?

L'optimisation des réseaux de distribution est pertinente pour de nombreux responsables logistiques, car elle peut conduire à des réductions significatives de coûts et à une augmentation de la qualité du service fourni. Une autre façon d'améliorer le réseau de formation radiale est de changer l'état des machines et la manière de les interconnecter. La reconfiguration permet d'économiser de l'énergie et peut donner au réseau une marge de manœuvre pour faire face aux urgences sur le réseau [24].

2.3 L'état de l'art sur les méthodes de l'optimisation électrique

La communauté scientifique est plus que jamais convaincue que l'avenir de l'énergie électrique réside dans l'exploitation de l'intelligence artificielle dans le domaine de l'énergie. À cet égard, de nombreuses études et recherches se concentrent sur l'optimisation électrique pour réduire la consommation, les coûts et d'éviter la consommation aux heures de pointe. C'est pourquoi nous avons soigneusement sélectionné quelques articles qui traitaient du sujet.

Harold Salazar et ses collaborateurs nous présentent un article intitulé "Réseaux de neurones artificiels et techniques de clustering appliquées à la reconfiguration des systèmes de distribution" [25]. Dans cet article, plusieurs stratégies sont suggérées qui peuvent être utilisées pour optimiser les pertes de puissance dans le réseau de distribution, notamment

l'inclusion de batteries de condensateurs, l'équilibrage de phase et la reconfiguration des alimentations de distribution. En changeant la topologie du système en ouvrant et fermant des interrupteurs afin de réduire les pertes du système ou de réduire le temps de récupération du service en cas d'urgence. L'article a également abordé deux types de modèles d'optimisation, qui sont les suivants :

- Algorithmes heuristiques et/ou techniques classiques d'optimisation mathématique qui déterminent généralement une seule solution pour une condition de charge spécifique qui est normalement atteinte par un processus interactif.
- Algorithmes basés sur l'intelligence artificielle (IA) dans lesquels certaines méthodes métaheuristiques peuvent être incorporées.

Alors que l'article se concentre sur l'optimisation basée sur l'intelligence artificielle, en particulier les réseaux de neurones artificiels. ANN utilise un seul réseau de neurones et offre la possibilité de sélectionner les structures les plus appropriées. Afin d'améliorer les performances et l'architecture de l'ANN, il est suggéré d'utiliser des techniques de clustering (création de clusters ou groupes) pour réduire le nombre de données d'entrée pour l'ANN. Les meilleures performances sont obtenues en utilisant des techniques de clustering pour l'apprentissage des clusters, résultant en une source d'informations plus efficace pour l'ANN. Le résultat est un ANN avec une capacité de généralisation améliorée et la capacité de définir des topologies de haute qualité avec des pertes plus faibles [25].

En 2005, Chao-Ming Huang et ses collaborateurs [26] ont effectué une revue de la littérature sur l'optimisation par essaims de particules afin de déterminer le modèle ARMAX pour prédire la charge à court terme. L'article faisait également référence à de nombreuses

méthodes de prévision telles que les réseaux de neurones artificiels et les modèles de séries chronologiques stochastiques. Notons que la prévision est l'une des bases de l'intelligence artificielle.

Cet article propose une nouvelle approche pour PSO (PSO pour Particle Swarm Optimization) pour définir une moyenne mobile autorégressive avec un modèle à variable exogène (ARMAX) pour la prévision à court terme (STLF pour Short Term Load Forecasting). Une technique alternative est fournie pour la détermination de l'ordre du modèle et l'estimation des paramètres afin d'obtenir une erreur de prévision minimale globale et un temps de calcul court [26]. De même l'approche PSO est subsumée sous la bannière d'optimisation heuristique.

À travers ces différentes études, nous avons constaté que l'optimisation basée sur l'intelligence artificielle donne des résultats prometteurs, notamment sous l'aspect de la prévision. Ainsi, la recherche bibliographique se limitera aux méthodes de prévision.

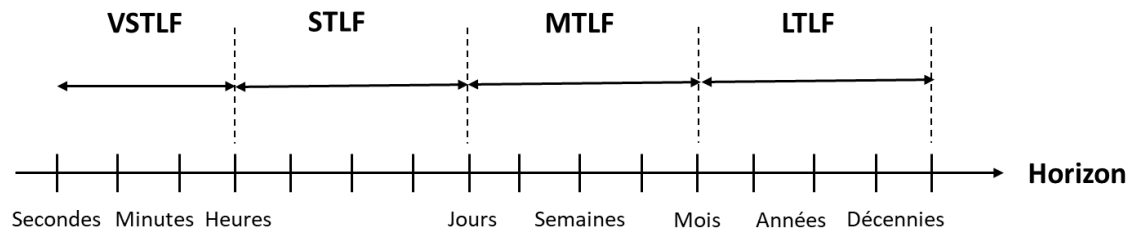
2.4 Les horizons de la prévision

Afin de stabiliser le réseau électrique en évitant les coupures de courant pendant les périodes de pointe en hiver dans les régions du nord et en été dans les régions du sud, il est nécessaire d'améliorer la gestion par la technologie de prévision, voire en la divisant en plusieurs classes :

- La prévision de charge à très court terme nommée en anglais comme Very Short Term Load Forecasting (VSTLF). L'objectif est de prévenir la charge d'énergie consommée d'un quart d'heure à une heure [27], [28].

- La prévision de charge à court terme nommée en anglais comme Short Term Load Forecasting (STLF). La conception du marché de l'électricité nécessite une planification préalable à la demande pour les différentes divisions énergétiques, c'est-à-dire la production, le transport et la distribution. Alors que la STLF est particulièrement essentielle pour le fonctionnement ponctuel du marché de l'électricité. Des prévisions de demande inexactes coûteront à l'installation une charge financière énorme, et cette méthode traditionnelle prend principalement en compte les effets météorologiques ainsi que les effets de calendrier. La durée prévue varie de quelques heures à une journée entière [29], [28].
- La prévision de charge à moyen terme connue en anglais comme Medium Term Load Forecasting (MTLF) couvre une période de (1 à 12 mois). Ce type de prévision considère principalement des facteurs de croissance, c'est-à-dire des facteurs qui affectent la demande tels que des événements majeurs, l'ajout de nouvelles charges, les changements saisonniers et les modèles de combinaison de la demande les besoins importants et d'entretien des gros consommateurs. De plus, ce type de prévision utilise des délais pour anticiper le pic de demande pour les jours ou semaines à venir. Avec ces informations, il est possible de déterminer si certaines installations/usines doivent être maintenues ou non pendant une certaine période. Cette prévision aide également à planifier les principaux tests et événements de fonctionnement et à déterminer les temps d'arrêt des principales installations et équipements [30], [28].
- La prévision de charge à long terme connue en anglais comme Long Term Load Forecasting (LTLF) couvre des horizons de prévision d'un à dix ans, et parfois jusqu'à plusieurs décennies. La prévision de la demande de charge à long terme est nécessaire pour le bon fonctionnement des services publics d'électricité. Les prévisions à long

terme sont généralement utilisées pour la planification, comme la détermination des futurs sites de générateurs. Une prévision précise des prix de l'électricité permettrait aux commerçants et aux entreprises de prendre des décisions commerciales éclairées dans un environnement volatile. La prévision de charge à long terme joue un rôle important dans les systèmes électriques pour la planification et la gestion des actifs du système, la planification de la construction de nouveaux sites ou installations de production et l'achat d'électricité des unités de production [31],[32],[28].



VSTLF: prévision de charge à très court terme

STLF: prévision de charge à court terme

MTLF: prévision de charge à moyen terme

LTLF: prévision de charge à long terme

Figure 2-1: Les horizons de prévision.

Par conséquent, nous concluons de ce qui précède que toute prévision de charge est principalement basée sur les informations collectées précédemment comme les données météorologiques, calendaires, économiques, démographiques et l'historique de la demande électrique [33].

2.5 L'état de l'art sur les méthodes de prévision

Après avoir discuté de la durée optimale de chaque prévision, nous apporterons un petit éclairage sur les modèles proposés jusqu'à présent. Dans ce contexte, de nombreuses études et recherches dans ce domaine ont prouvé son efficacité.

Une étude a été réalisée dans la région de Sverdlovsk en Russie, plus précisément sur la ville d'Ekaterinbourg. Dans cette étude, les auteurs ont utilisé l'information de comptage d'énergie automatisé pour la période de juillet 2012 jusqu'à janvier 2016. Dans ce projet, les données obtenues étaient complètes et présentées en kilowattheures, c'est pourquoi aucune interpolation ou filtrage des données n'a été effectué. À noter que, selon l'étude, les données collectées pendant 24 semaines permettent d'anticiper les 4 prochaines semaines de consommation électrique. Dans cet article, les auteurs ont utilisé les méthodes suivantes : suite des réseaux de neurones artificiels à mémoire longue à court terme (LSTM-ANN pour Long Short Term Memory model - ANN), régression de machine vectorielle de support (SVM pour Support Vector Machine) basée sur des fonctions de base radiales (RBF pour Radial Basis Function), régression SVM linéaire et moyenne mobile intégrée autorégressive (ARIMA pour AutoRegressive Integrated Moving Average) [34].

- Le réseau LSTM est une sous-espèce de l'architecture récurrente des réseaux de neurones basés sur des modules LSTM. Chaque module dispose de trois ports : port d'entrée, de sortie et d'oubli. Chaque module LSTM est capable de stocker des valeurs pour des périodes courtes et longues. Le temps de stockage des valeurs en mémoire est déterminé au niveau du port d'oubli. Le nombre de modules LSTM du réseau de neurones varie pour obtenir la plus petite erreur de prévision au niveau de l'échantillon d'apprentissage [34].

- La méthode de régression machine vectorielle basée sur SVM a été utilisée pour trouver la fonction de régression basée sur des échantillons de données qui ne sont pas à plus d'une certaine distance de la fonction. La méthode de régression basée sur RBF est différente de la méthode de régression linéaire. Elle transforme les échantillons de données en un espace de grande dimension. Ce n'est que lorsque l'algorithme entre dans la deuxième étape de l'espace de grande dimension que le modèle de régression linéaire est construit [34].
- Le modèle ARIMA appartient à la classe des modèles statistiques d'analyse et de prévision des séries temporelles généralement appelés ARIMA (p, d, q), où les paramètres sont des entiers non négatifs qui caractérisent l'ordre des parties du modèle [34].

Pour calculer les prévisions de consommation d'énergie, le programme a été écrit sous Python 3.6 à l'aide des bibliothèques de modèles Keras [35], Scikit-learn [36] et Statsmodels [37] contenant des implémentations des algorithmes de prévision utilisés.

Lors de la programmation, les étapes décrites ci-dessous ont été suivies:

- Étape 1 : Les données pour chaque semaine sont segmentées de manière à obtenir une série chronologique de 182 échantillons.
- Étape 2 : Cette étape est considérée comme la base du travail, puisque les 182 échantillons obtenus précédemment sont divisés en deux parties, la première pour l'entraînement 66%, et la seconde pour l'expérimentation de 34%.
- Étape 3 : Enfin, les modèles de prévision considérés précédemment sont utilisés, et après la prévision, le résultat obtenu est comparé aux données réelles pour découvrir l'erreur. Deux méthodes ont été utilisées pour estimer l'ampleur de l'erreur : erreur

moyenne absolue en pourcentage (MAPE) et erreur maximale absolue en pourcentage (MaxAPE) [34].

Les résultats présentés au cours de cette étude montrent que SVM a les valeurs minimales de MAPE dans le problème de prévision de puissance électrique basé sur les méthodes RBF et ARIMA. Ces méthodes montrent une erreur moyenne attendue de 12% dans la plage de 1 à 4 semaines. La méthode de régression SVM basée sur RBF a un MaxAPE inférieur de 21% par rapport à ARIMA pour prévoir un intervalle de 4 semaines. La méthode de régression SVM basée sur RBF a une meilleure précision dans le problème de prévision de la consommation d'énergie [34].

Une étude similaire a été menée en Estonie où les données horaires pour 2019 ont été utilisées comme données d'entraînement pour prévoir les 20 premiers jours de 2020. Dans cette étude [38], les trois modèles de prévision suivantes ont été utilisés :

- Le modèle régression linéaire (LM) est un modèle série chronologique qui utilise 24h de l'entraînement pour prédire 1h d'avance à l'aide de la fonction fitlm de MATLAB. Quant à cette expérience, les données des dernières 24 heures de 2019 ont été utilisées pour prévoir la première heure de 2020. Après cela, en créant une boucle de prévision de la première heure à l'heure suivante jusqu'à la fin des 24 heures, l'expérience a été répétée pendant 20 jours, ensuite l'erreur quadratique moyenne RMSE est calculée.
- Le réseau de mémoire à long court terme (LSTM) d'après le nom, il est clair que ce modèle dépend principalement du long terme de prévision. Contrairement au premier modèle, celui-ci permet de prévoir les 20 premiers jours de l'année 2020 à l'aide des données complètes de l'année 2019. La fonction trainNetwork de MATLAB a été

utilisée pour l'entraînement. En ce qui concerne la RMSE, la même opération a été répétée.

- Dans le modèle de réseau de neurones (NN), l'ensemble des données 2019 a été utilisé pour l'entraînement (70 %) et les tests (30 %) et les 20 premiers jours de 2020 ont été prévus sur la base de la consommation réelle des 24 heures précédentes pour chaque jour. Le calcul de RMSE a été également réalisé pour comparer les résultats.

Ainsi, le modèle de régression linéaire (ML) donne de bons résultats en ce qui concerne le RMSE. Le modèle LSTM donne des résultats différents entre le prévu et le réel selon les jours, et donne une RMSE important. Et pour le cas NN, lui aussi donne une RMSE considérable [38].

A. Sahebalam et ses collaborateurs font une étude [39] pour la prévision de la consommation d'électricité en combinant des séries chronologiques et des données météorologiques. Par conséquent, les modèles de Box et Jenkins et les modèles de régression sont utilisés pour prévoir la consommation d'énergie pour les résidences. L'étude a été menée sur 31 ménages, les données obtenues sont donc des données journalières et saisonnières équivalentes à 618 189 échantillons pour une consommation toutes les 30 minutes. Ces données correspondent à la période définie entre avril 2013 et mars 2014. Cette date est importante, car elle correspond aux dates des données météorologiques. Les données météorologiques contiennent 335 enregistrements avec 6 variables composées de jours, mois, années, température maximale, température minimale et température moyenne. Après avoir préparé les données, les consommateurs d'énergie sont regroupés en deux groupes et chaque groupe est divisé pour entraîner et tester l'ensemble de données. On choisit au hasard 75 % des utilisateurs dans les deux groupes pour l'entraînement et les 25 % restants pour les tests.

Deux modèles Box Jenkins (BJ) et un modèle de régression sont utilisés pour prévoir la consommation. Lorsque le modèle de régression est utilisé, les données de météo doivent être corrélés avec les données de consommation. En revanche, si le modèle Box Jenkins est utilisé, il n'est pas nécessaire de corrélés les données, car il utilise des séries chronologiques. Notons que la segmentation des données permet d'améliorer la prévision.

Le réseau européen a organisé un concours pour résoudre le problème de la prévision de la consommation d'électricité, un cas d'une compagnie d'électricité en Slovaquie orientale organisé sous le slogan « *Technologies intelligentes pour les systèmes intelligents adaptatifs (EUNITE)* ». Dans cette étude [40], la société fournit les données de charges journalières à une demi-heure et la température quotidienne moyenne pour la période de 1997-1998. La société fournit aussi la liste des jours fériés pour la même période. Pour vérifier l'exactitude de la prévision, l'erreur maximale (M) et l'ampleur de l'erreur (MAPE) ont été calculées. La prévision a été faite à l'aide d'un modèle ARX pour prévoir le mois suivant en réalisant l'entraînement par le mois précédent. Dans ce cas, on peut utiliser la prévision basée sur les données météorologiques fiables (à court terme). Le problème d'identification du système à l'aide du modèle ARX consiste à estimer un modèle d'un système basé sur des données d'entrée-sortie observées. Le modèle ARX utilise la méthode des moindres carrés pour modifier les paramètres d'un modèle polynomial linéaire. Dans cette expérience, les chercheurs ont entraîné le modèle en fonction de la météo de 1997 à 1998, et utilisé la météo comme entrée dans ce système pour estimer la consommation électrique du mois de janvier 1999. Puis les résultats de la puissance estimée sont comparés à la puissance réelle sachant que les données de janvier 1999 sont déjà connues. La comparaison a été faite à l'aide de l'erreur maximal et MAPE.

Une étude [41] a été menée à Taïwan pour explorer la possibilité d'une relation entre les quatre facteurs économiques, à savoir le revenu national (NI), la population (POP), la production intérieure brute (PIB) et l'indice des prix à la consommation (IPC) sur la consommation d'électricité. Ensuite, l'étude a permis de développer un modèle de prévision de la consommation d'électricité grâce à l'enregistrement des données linéaires et non linéaires (NI, POP, PIB et IPC) fournies par TEDC (Taiwan Economic Data Center) supervisée par le ministère de l'Éducation de Taïwan pour la période de janvier 1990 à décembre 2002. De plus, l'enregistrement de la température mensuelle (TEMP) est collecté à partir de 25 stations météorologiques sans oublier l'historique de consommation électrique de janvier 1990 à décembre 2002. La consommation diminue en hiver et augmente progressivement pour atteindre son pic en été. Une corrélation a été établie entre les variables afin de confirmer mathématiquement l'existence d'une relation entre les variables. Pour trouver l'effet des variables économiques sur la consommation d'électricité, la consommation est estimée à l'aide de l'ANN non linéaire et de trois types de modèles linéaires, la régression linéaire multiple (LNREG), la régression de la surface de réponse (RSREG) et la régression avec le modèle d'erreur ARMA (ARMAX). Ensuite, les potentiels de prévision hors échantillon pour tous les modèles sont comparés. Puis pour construire un modèle de prévision à Taïwan. Dans la première étape, seules les 11 premières années de données ont été utilisées pour l'entraînement, et l'année suivante a été estimée et comparée aux résultats réels. La deuxième expérience utilise uniquement un ensemble de données de 4 ans de janvier 1997 à décembre 2000 comme données d'apprentissage et prévisions pour le même horizon temporel. Cela a permis de construire des modèles linéaires LNREG, ARMAX et RSREG ce qui a donné des résultats importants lorsqu'on utilise POP et NI pour la prévision.

2.6 Synthèse du chapitre

Par notre lecture attentive des dernières tendances en matière d'optimisation liée à l'IA, nous avons approfondi nos connaissances sur les méthodes de prévisions de consommation d'électricité. Nous pouvons déduire entre autres la multiplicité des facteurs qui peuvent être utilisés pour la prévision de la consommation électrique, et elles sont les suivantes : historique de consommation électrique, température, calendrier des jours fériés et les quatre facteurs économiques, à savoir le revenu national (NI), la population (POP), la production intérieure brute (PIB) et l'indice des prix à la consommation électrique (IPC). Il a été confirmé qu'il existe une corrélation entre les facteurs indiqués avec la consommation électrique. Nous segmentons uniquement les données obtenues à partir de la source confiée afin que la segmentation soit adaptée au modèle utilisé. Cette répartition contient une partie de l'entraînement et une partie d'essai. Nous avons aussi établi qu'il y a plusieurs horizons pour la prévision (VSTLF, STLF, MTLF et LTLF), et que nous pouvons faire une prévision en boucle, par exemple de la première heure à l'heure suivante jusqu'à la fin des 24 heures. Cette prévision peut être réalisée par des modèles comme: ANN, ARX, SVM, RBF, ARIMA, LM, LNREG, RSREG, ARMAX et Box Jenkins. Pour que les résultats obtenus puissent être comparés aux données réelles à l'aide des outils de calcul de l'erreur RMSE, MAPE, MaxAPE et MAE.

Dans ce travail, nous allons concentrer notre attention sur les modèles ARMAX, ARX et Box Jenkins, car ils donnent des résultats prometteurs en termes de précision, et de faible taux d'erreur par rapport aux autres modèles selon les articles précités. Une autre raison pour laquelle nous avons choisi les modèles ARMAX, ARX et Box Jenkins est la qualité des

données obtenues par la ville pour faire une application de prévision de la consommation électrique dans des conditions météorologiques extrême.

Chapitre 3 - Analyse préliminaire et développement des modèles

3.1 Introduction

Comme nous l'évoquions précédemment, il est devenu nécessaire d'améliorer la consommation d'énergie et d'accélérer la transition énergétique, ici l'intelligence artificielle apparaît comme l'un des moyens d'atteindre ces objectifs. Ainsi, en analysant de grandes quantités de données provenant de producteurs d'énergies fossiles ou renouvelables ou de consommateurs individuels, l'IA et l'apprentissage automatique peuvent fournir des éléments pour optimiser la distribution, équilibrer les charges de consommation d'énergie et gérer les fluctuations de la production d'énergie renouvelable [42].

En apprentissage automatique, de nombreux modèles sont utilisés pour la prévision. Deux grandes catégories de méthodes de prévision (processus d'apprentissage) pour l'énergie électrique sont proposées. Dans une première catégorie, nous trouvons la prévision supervisée de l'énergie qui est basée sur des algorithmes tels que les algorithmes de logique (arbre de décision, apprentissage avec un ensemble de règles), les méthodes basées sur la cognition (réseaux de neurones), les algorithmes d'apprentissage statistique (réseaux bayésiens, apprentissage basé sur les instances), les machines à vecteurs de support et les séries chronologiques. Dans une deuxième catégorie nous trouvons la prévision non supervisée (ou clustering) d'énergie. Cette catégorie inclut des algorithmes de regroupement,

l'analyse en composante principale, la décomposition en valeur singulière, l'analyse en composantes indépendantes, le clustering hiérarchique, et le clustering k-means [43].

3.2 Le coefficient de Hurst et la prévisibilité

Pour choisir ou développer une méthode de prévision, il est très important d'analyser et de scruter les possibilités des données utilisées pour prévoir correctement en définissant une méthode pour son évaluation. La méthode courante d'évaluation des données consiste à calculer l'exposant ou le coefficient de Hurst. Ce dernier est utilisé pour classer les séries chronologiques et est un indicateur utilisé dans l'analyse fractale, et a été démontré dans de nombreux domaines de recherche tels que le traitement du signal biomédical, les fluctuations de température, les études sur le changement climatique, les études financières et études de consommation prédictive. À la lumière de cela, l'exposant de Hurst fournit une mesure pour la quantification de la mémoire à long terme et une rupture de série chronologique [44],[45].

Les valeurs obtenues à partir du coefficient de Hurst sont comprises entre 0 et 1. Les séries chronologiques sont classées en trois catégories selon la valeur du coefficient H [46]:

- Si la valeur de H égale 0,5 alors la série est aléatoire;
- Si la valeur varie entre $0 < H < 0,5$ indique une série anti-persistante ; c'est-à-dire une série a une caractéristique de « retour à la moyenne », ce qui signifie qu'une valeur à la hausse est plus susceptible d'être suivie d'une valeur à la baisse, et vice versa. La force du "retour à la moyenne" augmente à mesure que H approche de 0.
- La troisième valeur représente la série persistante qui varie entre $0,5 < H < 1$. Ce dernier renforce la tendance, ce qui signifie que la direction (vers le haut ou vers le bas par rapport à la dernière valeur) de la valeur suivante est plus probablement la

même que la valeur actuelle. La force de la tendance augmente à mesure que H approche 1.

Pour estimer l'exposant de Hurst, il faut d'abord estimer la dépendance de l'étendue remise à l'échelle sur l'espace de temps d'observation n. Une série temporelle de longueur N est divisée en séries temporelles plus courtes de longueur $n = N, N/2, N/4, \dots$ puis la plage de remise à l'échelle moyenne est calculée pour chaque valeur de n [44].

L'exposant de Hurst peut être calculé en analysant la plage remise à l'échelle (analyse R/S). Pour une série chronologique (partielle) de longueur, $X = X_1, X_2, \dots, X_n$. La méthode d'analyse R/S est calculée comme suite [46] :

$$\text{Calculer la valeur moyenne } m : m = \frac{1}{n} \sum_{i=1}^n X_i \quad 3-1)$$

$$\text{Calculer la série moyenne ajustée } Y : Y_t = X_t - m, \quad t = 1, 2, \dots, n \quad 3-2)$$

$$\text{Calculer la série d'écart cumulés } Z : Z_t = \sum_{i=1}^t Y_i, \quad t = 1, 2, \dots, n \quad 3-3)$$

$$\text{Calculer la série de plages } R : R_t = \max(Z_1, Z_2, \dots, Z_t) - \min(Z_1, Z_2, \dots, Z_t), \\ t = 1, 2, \dots, n \quad 3-4)$$

$$\text{Calculer la série d'écart type } S : S_t = \sqrt{\frac{1}{t} \sum_{i=1}^t (X_i - U)^2}, t = 1, 2, \dots, n \quad 3-5)$$

- Calculer la série rééchelonnée $R(n)/S(n)$ et la moyenne de toutes les séries temporelles partielles de longueur n.

L'estimation de l'exposant de Hurst est faite en ajustant une loi de puissance aux données

$$E \left[\frac{R(n)}{S(n)} \right] = C_{n^H}. \text{ Cela peut être fait en traçant la régression logarithmique et linéaire de}$$

$$E \left[\frac{R(n)}{S(n)} \right] \text{ par rapport à } \log n ; \text{ la pente de la droite donne } H \text{ [44].}$$

3.3 Modélisation des séries chronologiques

Une série temporelle est une série de nombres qui correspondent à l'évolution d'une quantité donnée dans le temps. Ces variables aléatoires peuvent être la cible d'analyses mathématiques pour comprendre leur évolution passée et prédire leur comportement futur. Les données sur la vitesse du vent, par exemple, sont considérées comme des séries temporelles [44].

Pour un examen plus approfondi, il existe deux catégories de modèles pour le calcul des séries chronologiques. La première considère les données comme une fonction du temps ($y = q(v)$). Cette classe de modèle peut être construite en utilisant la méthode des moindres carrés ou d'autres méthodes itératives. L'analyse du modèle par transformation de Fourier est une version avancée de ce type de modèle. Alors qu'il est considéré comme une seconde classe de modèles qui cherche à déterminer chaque valeur de la série en fonction des valeurs qui la précèdent ($y_t = f(y_{t-1}, y_{t-2}, \dots)$). C'est le cas des modèles ARIMA ("AutoRegressive - Integrated - Moving Average") [47].

Dans ce chapitre, plusieurs notions importantes liées à l'analyse des séries chronologiques seront discutées. Parmi celles-ci, on retrouve les notions d'autocorrélation, de stationnarité et de bruit blanc [48].

3.3.1 *Processus stochastique*

Les séries chronologiques sont un processus aléatoire ou stochastique, c'est-à-dire une série d'événements ou de faits qui ne sont pas liés au temps ou aux événements eux-mêmes. C'est pourquoi on l'appelle aléatoire, par exemple, le prix hebdomadaire du stock, la

consommation d'électricité, précipitations quotidiennes, taux d'inflation mensuel ou prix du baril de pétrole.

Définition 3.1 : Nous appelons un processus stochastique, ou un processus aléatoire, une famille $(X_t)_{t \in T}$ de variables aléatoires à valeurs dans E . Autrement dit, pour tout $t \in T$, l'application $\omega \rightarrow X_t(\omega)$ est une application mesurable de (Ω, \mathcal{F}) dans (E, \mathcal{E}) . On appelle E l'espace d'états du processus [49]. En pratique, la modélisation dynamique est souvent utilisée : X_t est la valeur d'une variable d'intérêt à l'instant t . L'ensemble T représente alors l'ensemble des instants possibles.

Définition 3.2 : Lorsque $T = \mathbb{N}$ ou $T = \mathbb{Z}$ on dit que $(X_t)_{t \in T}$ est un processus à temps discret. Lorsque $T = \mathbb{R}$, ou un intervalle de \mathbb{R} , on parle de processus à temps continu [49].

3.3.2 Autocovariance et autocorrélation

Dans le processus $(X_t)_{t \in \mathbb{Z}}$, la variable aléatoire à l'instant t : dépend généralement, d'une certaine manière, des observations précédentes $X_s, s < t$. Une façon de comprendre les interrelations entre des termes tels que cette série est de considérer les fonctions d'autocovariance et d'autocorrélation [48]. Ces fonctions mesurent respectivement la covariance et la corrélation entre les termes d'une série chronologique. De façon générale, la covariance entre deux variables aléatoires V_1 et V_2 est définie par

$$\text{cov}(V_1, V_2) = E\{(V_1 - \mu_1)(V_2 - \mu_2)\} \quad (3-6)$$

Où $\mu_1 = E(V_1)$ et $\mu_2 = E(V_2)$. On peut montrer facilement que $\text{cov}(V_1, V_2) = E(V_1 V_2) - \mu_1 \mu_2$. Pour sa part, la corrélation est une version standardisée de la covariance. En particulier [48],

$$\text{cov}(V_1, V_2) = \frac{\text{cov}(V_1, V_2)}{\sigma_1 \sigma_2} \quad (3-7)$$

Où σ_1 et σ_2 sont les écarts types de V_1 et V_2 , respectivement. Certaines propriétés de covariance et de corrélation, pour lesquelles des preuves peuvent être obtenues facilement, sont données ci-dessous [48].

1. Si V_1 et V_2 sont stochastiquement indépendantes, alors $\text{cov}(V_1, V_2) = 0$;
2. Si le lien entre V_1 et V_2 est de la forme $V_2 = AV_1 + B$, où $A > 0$, alors $\text{cov}(V_1, V_2) = +1$; on parle alors de lien positif linéaire très fort entre V_1 et V_2 ;
3. Si le lien entre V_1 et V_2 est de la forme $V_2 = AV_1 + B$, où $A < 0$, alors $\text{cov}(V_1, V_2) = -1$; parle alors de lien négatif linéaire très fort entre V_1 et V_2 ;

Notez que l'inverse de la propriété (1) n'est pas nécessairement vrai. En effet, il est possible que la corrélation entre deux variables aléatoires soit nulle, sans impliquer qu'elles soient indépendantes [48].

Pour une série chronologique $(X_t)_{t \in \mathbb{Z}}$, la fonction d'autocovariance de délai l , pour tout $l \in \mathbb{Z}$, est définie par [48],

$$Y(l) = \text{cov}(X_t, X_{t-l}) \quad (3-8)$$

En particulier, on a $Y(0) = \text{var}(X_t)$. La fonction d'autocorrélation de délai l est définie à partir de la fonction d'autocovariance, à savoir [48],

$$\rho(l) = \text{cov}(X_t, X_{t-l}) = \frac{Y(l)}{Y(0)} \quad (3-9)$$

À noter que les fonctions d'autocovariance et d'autocorrélation sont symétriques au sens

Où

$$Y(-l) = Y(l) \text{ et } \rho(-l) = \rho(l). \quad (3-10)$$

L'autocorrélation partielle mesure la corrélation entre X_t et X_{t-h} sans toutefois prendre en considération l'influence des valeurs antérieures à X_{t-h} . Ainsi, on peut montrer que la fonction d'autocorrélation partielle d'un processus $(X_t)_{t \in \mathbb{Z}}$ est donnée par [48],

$$\rho(l) = \frac{|P(l)^*|}{|P(l)|}, l \in \mathbb{N} \quad (3-11)$$

Où

$$\rho(l)^* = \begin{pmatrix} 1 & \rho(1) & \rho(2) & \dots & \rho(l-2) & \rho(1) \\ \rho(1) & 1 & \rho(1) & \dots & \rho(l-3) & \rho(2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho(l-1) & \rho(l-2) & \rho(l-3) & \dots & \rho(1) & \rho(l) \end{pmatrix} \quad (3-12)$$

Et

$$\rho(l) = \begin{pmatrix} 1 & \rho(1) & \rho(2) & \dots & \rho(l-2) & \rho(l-1) \\ \rho(1) & 1 & \rho(1) & \dots & \rho(l-3) & \rho(l-2) \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ \rho(l-1) & \rho(l-2) & \rho(l-3) & \dots & \rho(1) & 1 \end{pmatrix} \quad (3-13)$$

Ici, $|A|$ est le déterminant d'une matrice carrée A. Ainsi, les trois premières autocorrélations partielles sont définies par les relations [48].

$$\rho(1) = \rho(1) \quad (3-14)$$

$$\rho(2) = \frac{\rho(2) - \rho(1)^2}{1 - \rho(1)^2} \quad (3-15)$$

$$\rho(3) = \frac{\rho(1)^3 - \rho(1)\rho(2)(2 - \rho(2)) + \rho(3)(1 - \rho(1)^2)}{1 - \rho(2)^2 - 2\rho(1)^2(1 - \rho(2))} \quad (3-16)$$

3.3.3 Stationnarité

La modélisation du comportement stochastique d'une série temporelle est beaucoup plus simple lorsque certains aspects de la loi $(X_t)_{t \in \mathbb{Z}}$ ne changent pas avec une différence de t.

Ensuite, nous parlons de processus ou d'une chaîne qui ont certaines propriétés fixes. Plus formellement, il existe deux types d'impermanence : celle au sens strict, et celle au sens faible [50].

Définition 3.3: L'opération $(X(t), t \in \mathbb{Z})$ est dite stationnaire au sens strict si les lois des vecteurs de dimension finie sont $(X_{t_1}, X_{t_2}, \dots, X_{t_k})$ et $(X_{t_1+h}, X_{t_2+h}, \dots, X_{t_k+h})$ correspond à $\forall k \in \mathbb{N}^*, \forall t_1, \dots, t_k \in \mathbb{Z}$ et $\forall h \in \mathbb{N}$.

Définition 3.4: L'opération $(X(t), t \in \mathbb{Z})$ est dite stationnaire au sens faible si :

- Son espérance est constante au cours du temps, c'est-à-dire, $E[X_t] = \mu, \forall t \in \mathbb{Z}$;
- Sa fonction d'autocovariance dépend seulement de l'écart de temps $t_2 - t_1$, c'est-à-dire, $Cov(X_{t_1}, X_{t_2}) = Cov(X_{t_1+c}, X_{t_2+c}) = \gamma(t_2 - t_1)$ avec c , une constante quelconque. Dans ce cas, $E(X_t^2) = \gamma(0) + \mu^2$ est une constante, $\forall t \in \mathbb{Z}$.

Pour la suite, $(X(t), t \in \mathbb{Z})$ sera dit stationnaire s'il l'est au sens faible.

La stationnarité signifie non seulement l'absence de direction, mais aussi la variation ou la symétrie constante. La cohérence au sens strict du terme signifie cela au sens faible. Ces deux concepts ne sont identiques que dans le cas du processus gaussien [50].

3.3.4 Bruit blanc

Définition 3.5 (Bruit blanc faible) Le bruit blanc faible est centré ($E(\varepsilon_t) = 0$) sur le processus $\varepsilon_t, t \in \mathbb{Z}$, ses variables non corrélées, c'est-à-dire $Cov(\varepsilon_{t_1}, \varepsilon_{t_2}) = 0, \forall t_1 \neq t_2$.

Le bruit blanc est dit fort si les variables sont indépendants [50].

Définition 3.6 (Bruit blanc gaussien) Un processus $\varepsilon_t, t \in \mathbb{Z}$ est un bruit blanc gaussien s'il s'agit d'un bruit blanc dont les variables sont indépendants et distribués de manière similaire avec une distribution normale de moyenne nulle [50].

3.3.5 Opérations sur les séries

Les opérations séquentielles ou l'opérations délai c'est une série permettent d'écrire simplement un processus à partir d'une perception passée. Ce facteur sera très utile, car il nous permettra de définir plusieurs types de modèles en utilisant une notation compacte et efficace [50],[48].

Définition 3.7 (facteur de retard) Supposons que B , le facteur de retard, $(X(t), t \in \mathbb{Z})$, soit une séquence de valeurs et k l'ordre du retard [50]. Puis,

$$B^k X_t = X_{t-k}, \forall t \in \mathbb{Z}, \forall k \in \mathbb{N}. \quad (3-17)$$

Ainsi, l'opérateur B permet de reculer de k pas depuis la position t .

Bien sûr, si $X_t = C, \forall t \in \mathbb{Z}$, puis

$$B^k X_t = B^k C = C, \forall k \in \mathbb{N}. \quad (3-18)$$

Parce que la constante ne peut pas être modifiée.

En observant des corrélations automatiques simples (ACF) et un diagramme en chaîne, il est possible de détecter la présence ou l'absence de problèmes statiques telle que la direction. Ceci, lorsqu'il est présent, indique un lien plus ou moins important entre les réalisations de la série. Il est détecté lorsque les autocorrélations simples ne descendent pas vers zéro de manière exponentielle (très rapidement). D'autre part, il peut être détecté lorsqu'un graphique en série affiche des valeurs qui suivent un certain schéma général

(courbe, augmentation ou diminution, etc.) au fil du temps. Dans ce cas, il y a une forte corrélation automatique alors que l'on veut travailler avec une chaîne statique. La différenciation est destinée à être une solution pour masquer l'orientation, réduisant ainsi les ACF vers zéro de manière exponentielle. Une autre façon de vérifier la nécessité de la différenciation est d'adapter la série au processus autorégressif d'ordre 1 visé à AR et de vérifier si la valeur absolue du module associé au modèle existant est proche de 1, mais inférieure à 1 [50].

Définition 3.8 (Différenciation) Soit ∇ , le facteur de différenciation, $(X(t), t \in T)$, une série de valeurs n et d l'ordre de différenciation [50].

On a

$$\nabla^d X_t = \sum_{k=0}^d (-1)^k C_k^d X_{t-k}, \forall d \geq 1, d \leq n - 1. \quad (3-19)$$

Donc l'ordre d représente le nombre de différentiels appliqués à la série; et le nombre de valeurs dans la série obtenue sera égal à n - d. Lorsqu'il est d'ordre 1, le différentiel annule le sens linéaire. Il peut être nécessaire de distinguer d'ordre 2 dans le cas de la direction quadratique. La différenciation est utile, mais la sur-différenciation doit être évitée, car elle crée plus de problèmes qu'elle n'en résout. En fait, on ajoute la dépendance si elle n'existe pas, alors que le but est d'avoir une chaîne statique. Il est donc nécessaire de s'assurer qu'un nombre suffisant de différentiels est appliqué, obtenant ainsi l'agencement optimal. Une façon de déterminer l'hyperparamètre d'une série consiste à vérifier si la valeur de l'estimateur $\hat{\rho}(1)$ est d'environ -0,5 alors que les autres valeurs $\hat{\rho}(h)$ sont petites [50].

En se référant à I, l'opérateur d'identité, nous pouvons voir que $\nabla = I - B$, car,

$$\nabla X_t = X_t - X_{t-1} = X_t - B X_t = (I - B) X_t \quad (3-20)$$

Une autre forme de différenciation est observée lorsque l'on observe une tendance saisonnière dans la série, c'est-à-dire la présence de pics à intervalles réguliers. La saisonnalité peut être observée à l'aide du graphique à chaînes et d'un simple programme d'autocorrélation. Il est à noter que l'ACF ne décroît pas très rapidement vers zéro avec les multiples de saison. Par exemple, les températures sont saisonnières [50].

Définition 3.9 (Facteur de différenciation saisonnière) L'opérateur ∇_s et la session s , l'équation de l'opérateur est définie par [50]

$$\nabla_s X_t = X_t - X_{t-s} \quad (3-21)$$

Où

$$\nabla_s = I - B^s \quad (3-22)$$

Et la différenciation saisonnière pour l'ordre D ,

$$\nabla_s^D = (1 - B^s)^D \quad (3-23)$$

Nous devons différencier si nécessaire tout en évitant la sur-différenciation. De plus, le nombre de valeurs dans les séries divergentes sera réduit dans l'ordre multiplié par la saison. Par exemple, si on observe 50 valeurs quartiles pour une série à saisonnalité annuelle ($s = 4$), alors la série obtenue après différenciation saisonnière aura 46 valeurs. Plus d'une tendance peut être observée. Une ou plusieurs distinctions saisonnières et simples s'avéreront nécessaires [50].

3.4 Processus aléatoires stationnaires

3.4.1 Modèle autorégressif: processus AR

Les premières opérations autorégressives ont été introduites par George Udny Yule. Le processus autorégressif est un processus dans lequel l'observation est écrite à l'instant t sous la forme d'un ensemble linéaire d'observations précédentes plus un peu de bruit blanc [51].

Définition 3.10 : La suite $\{X_t : t \geq 0\}$ est une opération autorégressive d'ordre p ($p > 0$) si elle s'écrit :

$$X_t = \sum_{k=1}^p \phi_k X_{t-k} + \varepsilon_t \text{ où } \{\varepsilon_t\} \sim WN(0, \sigma_\varepsilon^2) \quad (3-24)$$

Les ϕ_k ($k = 1, 2, \dots, p$) constituent les paramètres du modèle,

Dans ce cas, on se réfère à $\{X_t\} \sim AR(p)$, et de la même manière, on peut réécrire un processus AR(p) avec un polynôme $\phi(B)$ qui doublera X_t cette fois :

$$\phi(B)X_t = \varepsilon_t \text{ avec } \phi(B) = 1 - \phi_1 B - \phi_2 B^2 - \dots - \phi_p B^p \quad (3-25)$$

3.4.2 Modèle à moyenne mobile processus MA

Eugene Slutsky a été le premier à introduire des opérations de moyenne mobile en 1927. La définition suivante présente ce processus [51].

Définition 3.11 : On dit que la suite $\{X_t : t \geq 0\}$ est une opération de moyenne mobile d'ordre q ($q > 0$) si elle peut s'écrire [51] :

$$X_t = \sum_{k=1}^q \theta_k \varepsilon_{t-k} + \varepsilon_t, \text{ où } \{\varepsilon_t\} \sim WN(0, \sigma_\varepsilon^2), \quad (3-26)$$

Où les θ_k ($k = 1, 2, \dots, q$) sont les paramètres du modèle.

Dans ce cas, notons $\{X_t\} \sim MA(q)$

Nous pouvons utiliser l'opérateur de retard B pour écrire ce processus sous une autre forme [51]. On aura donc un polynôme dans B qui se multipliera par ε_t :

$$X_t = \theta(B)\varepsilon_t \text{ avec } \theta(B) = 1 - \theta_1 B - \theta_2 B^2 - \dots - \theta_p B^p \quad (3-27)$$

Définition 3.12 : Un processus est dit causal s'il existe une suite réelle $\{a_k\}$ telle que $\sum_{k=0}^{\infty} |a_k| < \infty$ et que [51]:

$$X_t = \sum_{k=0}^{\infty} a_k \varepsilon_{t-k} \quad (3-28)$$

Parfois, quand on parle d'un processus causal, on dit qu'il a une représentation de MA(∞).

Définition 3.13 : Le processus est dit réversible s'il existe une suite réelle $\{b_k\}$ telle que $\sum_{k=0}^{\infty} |b_k| < \infty$ et

$$\varepsilon_t = \sum_{k=0}^{\infty} b_k X_{t-k} \quad (3-29)$$

Une autre façon de dire que le processus est réversible et de dire qu'il a une représentation AR(∞) [51].

Théorème 3.1 : Processus autorégressif AR (p) est causal et constant si et seulement si c'est un polynôme $\emptyset(Z)$ donc

$$\emptyset(Z) \neq 0 \text{ avec } z \in \mathbb{C} \text{ tel que } |z| \leq 1. \quad (3-30)$$

En d'autres termes, toutes les racines de $\emptyset(z)$ ont un module supérieur à 1 [51].

Théorème 3.2 : Un processus à moyenne mobile MA(q) est réversible si et seulement si le polynôme $\theta(z)$ est ainsi [51]

$$\theta(z) \neq 0 \text{ avec } z \in \mathbb{C} \text{ tel que } |z| \leq 1. \quad (3-31)$$

Notez la similitude de cette déclaration avec le théorème de stabilité et de causalité pour les processus autorégressifs [51].

3.5 Identification du système

Pour déterminer le modèle approprié pour la prévision, l'action que doit être suivie est de définir le système. C'est le processus qui couvre le problème de la construction de modèles d'un système dynamique à partir des signaux d'entrée et de sortie observés. La figure 3-1 montre une procédure de modélisation de système dynamique basée sur l'approche d'identification [52].

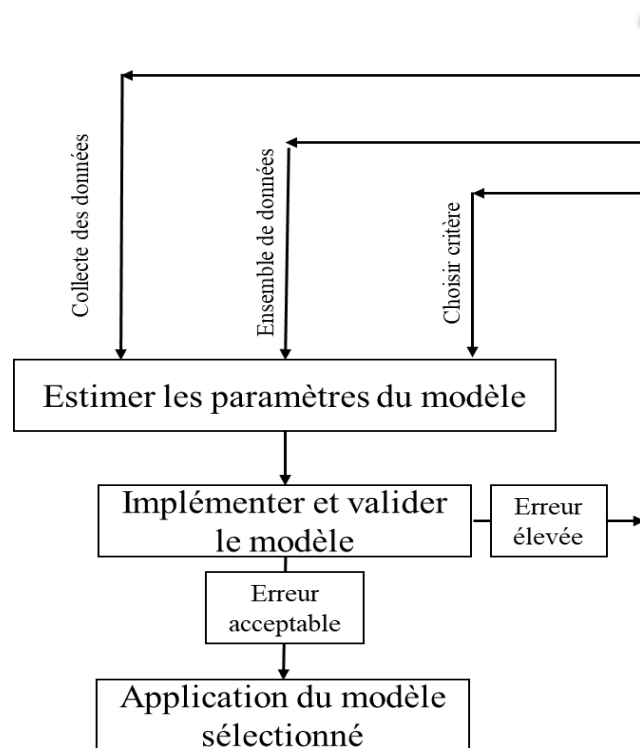


Figure 3-1: Approche d'identification du système [52].

La construction du modèle à partir des données extraites implique quatre étapes de base [52]:

- L'acquisition de données (Dataset);
- Proposition d'un ensemble de modèles candidats et détermination du meilleur modèle de l'ensemble;

- Estimation des paramètres;
- Validation du modèle.

Le diagramme schématique du modèle de système à entrées multiples et à sortie unique est illustré à la Figure 3-2. En plus des perturbations du système causées par la non-corrélation (ou faible corrélation) de certaines valeurs dans les entrées du système [53].

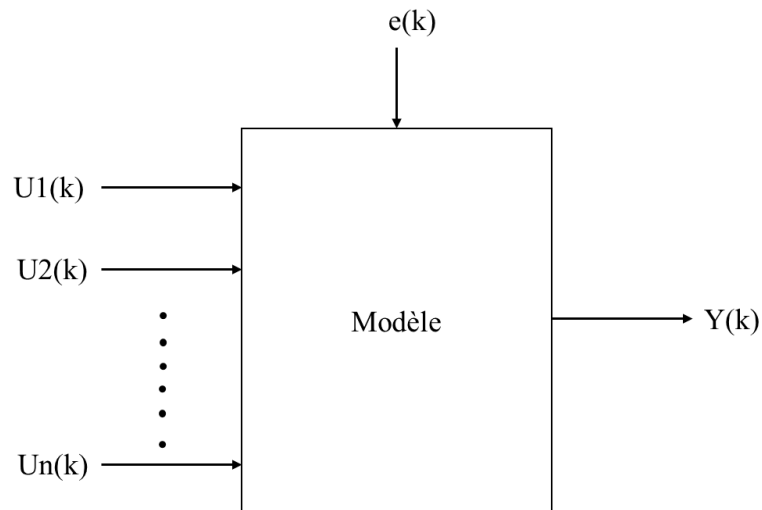


Figure 3-2: Système à entrées multiples et à sortie unique (MISO) [53].

$U1(k), U2(k) \dots Un(k)$: sont les entrées de système.

$Y(k)$: est la sortie du système.

$e(k)$: est un signal de bruit blanc.

3.6 Analyse théorique des méthodes de prévision

3.6.1 Les modèles ARX et ARMAX

Dans cette étude, nous considérerons deux modèles non-linéaires simples avec une seule sortie et multiples entrées. Ces modèles sont respectivement appelés modèle ARX

(Autorégressif avec entrée externe) et ARMAX (Moyenne mobile autorégressive avec entrée externe) [52].

ARX, un processus autorégressif exogène, est un modèle de séries chronologiques autorégressif avec des paramètres d'entrée exogènes. Les paramètres connus et le filtre généré par la corrélation entre les points de données de sortie précédents permettront de décrire le modèle [54]. La structure du modèle ARX peut être écrite sous la forme de l'équation :

$$A(q^{-1})y(k-n) = q^{-d}B(q^{-1})u(k-n) + e(k) \quad (3-32)$$

Ce qui peut être exprimé comme :

$$A(q^{-1})y(k-n) = q^{-d}B_1(q^{-1})u_1(k-n) + q^{-d}B_2(q^{-1})u_2(k-n) + e(k) \quad (3-33)$$

ARMAX, processus exogène de moyenne mobile autorégressive, est une extension d'un ARX où des paramètres de moyenne mobile sont ajoutés [54]. La structure du modèle ARMAX peut être écrite sous la forme de l'équation :

$$A(q^{-1})y(k-n) = q^{-d}B(q^{-1})u(k-n) + C(q^{-1})e(k) \quad (3-34)$$

Ce qui peut être exprimé comme :

$$A(q^{-1})y(k-n) = q^{-d}B_1(q^{-1})u_1(k-n) + q^{-d}B_2(q^{-1})u_2(k-n) + C(q^{-1})e(k) \quad (3-35)$$

Que ce soit le modèle ARX ou ARMAX $A(q^{-1}), B(q^{-1})$, et $C(q^{-1})$, il y a des polynômes à estimer. Pour le modèle ARX, le polynôme $C(q^{-1}) = 1$.

Ces polynômes représentent la dynamique globale du système où $A(q^{-1}), B(q^{-1})$, et $C(q^{-1})$ sont définis par :

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{na}q^{-na} \quad (3-36)$$

$$B(q^{-1}) = b_0 + \dots + b_{nb}q^{-nb} \quad (3-37)$$

$$C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_{nc}q^{-nc} \quad (3-38)$$

$u(k)$ et $y(k)$ sont respectivement l'entrée et la sortie du système et $e(k)$ est un signal de bruit blanc. k est l'unité de temps et q^{-1} représente l'opérateur de retard.

Les variables a , b et c sont les paramètres du modèle à estimer, avec $i = 1, \dots, n_a, j=1, \dots, n_b, l=1, \dots, n_c$. n est le modèle à décalage pur. La valeur minimale de n est supposée égale à 0 [52].

3.6.2 Modèle Box Jenkins

George Box et Gwilym Jenkins (1976) ont développé une méthodologie pour déterminer un modèle de série chronologique approprié. Leur méthode Box-Jenkins fait référence à une méthode systématique d'identification, d'ajustement, de vérification et d'utilisation de modèles de séries chronologiques intégrés autorégressifs à moyenne mobile (ARIMA). Cette méthode convient aux séries chronologiques de durée moyenne à longue [55]. La méthode Box-Jenkins utilise l'opérateur de rétrogradation pour faciliter l'écriture de ces modèles. L'opérateur de rétrogradation B a pour effet de changer la période t en période $t - 1$. Ainsi $BX_t = X_{t-1}$ et $B^2X_t = X_{t-2}$. En utilisant cette notation de rétrogradation, le modèle ci-dessus peut être réécrit comme :

$$(1 - \phi_1B - \dots - \phi_pB^p)X_t = (1 - \theta_1B - \dots - \theta_qB^q)a_t \quad (3-39)$$

Cela peut être abrégé encore plus en écrivant :

$$\phi_p(B)X_t = \theta_q(B)a_t \quad (3-40)$$

$$\text{Où : } \theta_p = (1 - \theta_1B - \dots - \theta_qB^q)a_t \quad (3-41)$$

$$\text{Et : } \phi_p = (1 - \phi_1B - \dots - \phi_pB^p)a_t \quad (3-42)$$

Ces formules montrent que les opérateurs $p(B)$ et $q(B)$ sont des polynômes dans B pour les ordres p et q respectivement. L'un des avantages d'écrire des modèles de cette manière est que l'on voit pourquoi plusieurs modèles sont équivalents. Autrement dit, la structure du modèle Box Jenkins [56] peut être écrite sous la forme de l'équation :

$$(1 - \phi_1 B)(y_t - u) = (1 - \theta_1 B)a_t \quad (3-43)$$

Ce type de réarrangement de modèle est utilisé par les prévisionnistes expérimentés de Box-Jenkins pour obtenir les modèles les plus simples possible. Le programme ARIMA théorique affiche les racines des deux polynômes, $p(B)$ et $\theta q(B)$, afin de montrer les simplifications possibles du modèle.

Généralement la structure du modèle Box-Jenkins est [57] :

$$y(t) = \frac{A(q^{-1})}{D(q^{-1})} u(t - nk) + \frac{B(q^{-1})}{C(q^{-1})} \frac{e(t)}{1 - q^{-1}} \quad (3-44)$$

Les éléments du modèle Box-Jenkins sont définis comme suit :

$$A(q^{-1}) = a_1 + a_2 q^{-1} + \dots + a_{na} q^{-na+1} \quad (3-45)$$

$$B(q^{-1}) = 1 + b_1 q^{-1} + \dots + b_{nb} q^{-nb} \quad (3-46)$$

$$C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc} \quad (3-47)$$

$$D(q^{-1}) = 1 + d_1 q^{-1} + \dots + d_{nd} q^{-nd} \quad (3-48)$$

Les variables a , b , c et d sont les paramètres du modèle à estimer. Les polynômes à estimer sont définis comme $A(q^{-1})$, $B(q^{-1})$, $C(q^{-1})$ et $D(q^{-1})$.

Comme montré dans la Figure 3-3, selon la manière dont la perturbation $e(k)$ est supposée se produire, nous obtenons des structures différentes pour les modèles ARX, ARMAX et Box Jenkins [58].

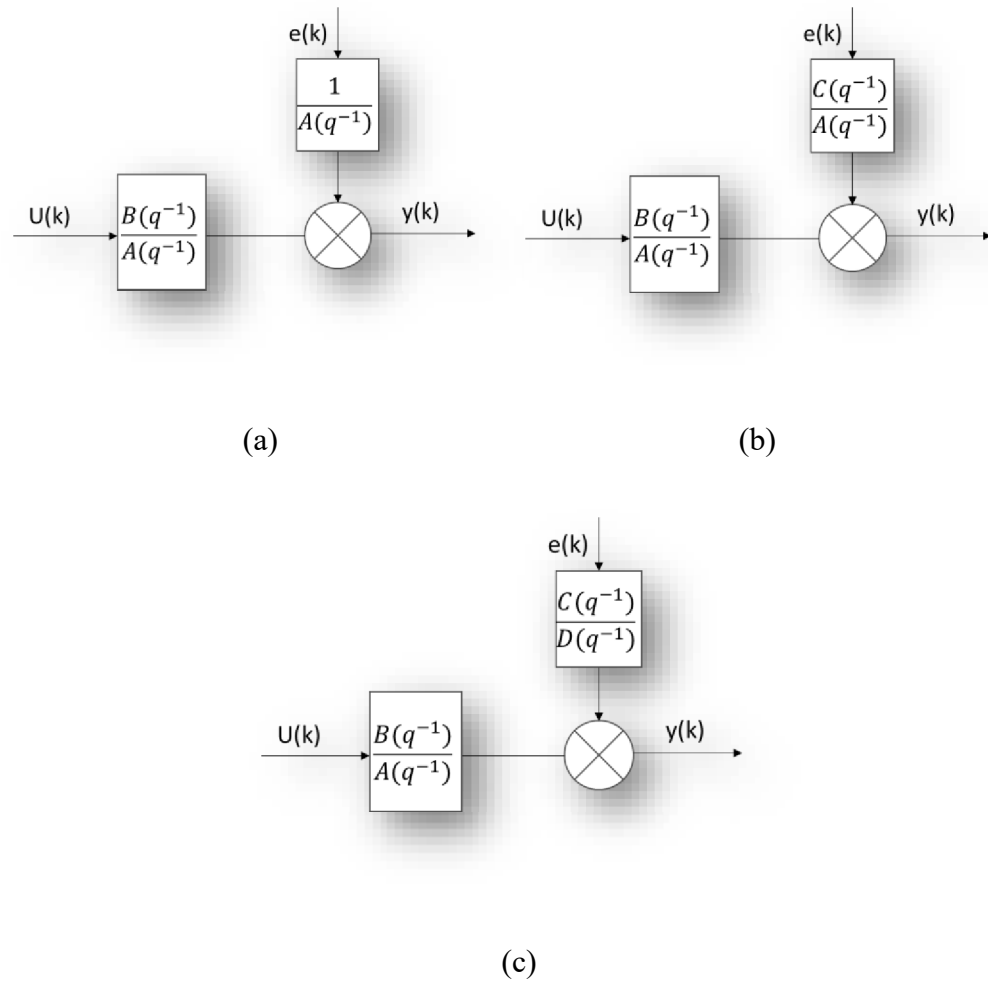


Figure 3-3: Modèle paramétrique de ARX (a), ARMAX (b) et Box Jenkins (c) [58].

3.7 Analyse d'erreur de prévision

Pour s'assurer de l'exactitude de la prévision de la consommation d'énergie, il est nécessaire de faire une comparaison en calculant l'erreur entre le signal prévu et le réel afin d'apporter une modification au modèle si nécessaire ou de l'améliorer à l'aide des outils suivants tels que la fonction de coût, la fonction de perte et fonction objectif [59]. Quelques outils pour calculer l'erreur de prévision couramment utilisés sont : l'erreur absolue moyenne symétrique (SMAPE), le pourcentage d'erreur absolue moyenne (MAPE), l'erreur absolue

moyenne (MAE), l'erreur quadratique moyenne (MSE) et la racine de l'erreur quadratique moyenne (RMSE) [44].

$$SMAPE = \frac{2}{m} \sum_{k=1}^m \frac{|X_k - Y_k|}{X_k + Y_k} \quad (3-49)$$

$$MAE = \frac{1}{m} \sum_{k=1}^m |X_k - Y_k| \quad (3-50)$$

$$RMSE = \sqrt{\frac{\sum_{k=1}^m (X_k - Y_k)^2}{m}} \times 100 \quad (3-51)$$

$$MSE = \frac{\sum_{k=1}^m (X_k - Y_k)^2}{m} \quad (3-52)$$

À partir de maintenant, nous allons nous concentrer sur l'erreur absolue moyenne MAE car c'est la métrique recommandée dans le cas de bases de données qui contiennent souvent des valeurs aberrantes [60].

3.7.1 Calcul du pourcentage d'erreur

Le pourcentage d'erreur est une simple expression quantitative de la différence entre une valeur expérimentale (x) et une valeur théorique (y). Le pourcentage d'erreur est toujours exprimé sous la forme d'une valeur positive [61]. La formule ci-dessous est utilisée pour le calculer :

$$\text{Pourcentage d'erreur} = \left| \frac{x-y}{y} \right| * 100\% \quad (3-53)$$

Cette fonction renvoie toujours la valeur positive du nombre qu'elle contient. Il est donc impossible d'avoir un pourcentage d'erreur négatif. C'est pourquoi nous devons définir des vecteurs absolus pour rendre les valeurs positives.

3.7.2 Erreur absolue moyenne MAE

L'erreur absolue moyenne (MAE) est la mesure de l'erreur de régression la plus facile à comprendre. Nous calculons le reste pour chaque point de données, en prenant la valeur absolue de chacun uniquement afin de ne pas annuler les résidus négatifs et positifs [62]. MAE est représentée par l'équation suivante [63] :

$$MAE = \frac{1}{T} \sum_{t=1}^T |y_t - \hat{y}_t| \quad (3-54)$$

ou

$$MAE = \frac{\sum_{i=1}^n \text{abs}(y_i - \lambda(x_i))}{n} \quad (3-55)$$

En plus, l'erreur absolue moyenne est utilisée pour augmenter la vitesse de traitement et pour l'ajustement du modèle pendant les tests pour minimiser l'erreur de généralisation [22].

3.8 Synthèse du chapitre

Le but de ce chapitre est d'approfondir l'interprétation théorique des modèles ainsi que les raisons et motivations de ce choix. Partant d'une courte définition de la série temporelle, et de la qualité du signal qui s'avère être un processus stochastique, nous avons présenté également une analyse théorique des processus aléatoires stationnaires des modèles autorégressifs (ARX / ARMAX) et du modèle de processus à moyenne mobile (Box-Jenkins). Nous avons aussi introduit un aperçu du modèle pour calculer le pourcentage d'erreur de prévision, disons l'erreur absolue moyenne en pourcentage.

Au chapitre suivant nous appliquerons cette analyse théorique dans un cas réel de la Ville de Baie-Comeau pour tenter de faire la prévision de la consommation d'électricité, et de cette manière anticiper les périodes de pointe.

Chapitre 4 - Collecte et analyse des données

4.1 Introduction

Comme convenu, la collecte et l'analyse de données sont l'un des fondements sur lesquels repose l'intelligence artificielle (IA). Étant donné que les données sont le carburant sur lequel fonctionne l'intelligence artificielle, plus la quantité de données est diversifiée, plus les applications d'apprentissage automatique peuvent acquérir et maîtriser ces compétences. Cela signifie que plus une IA possède de données, plus elle peut apprendre et améliorer ses capacités de reconnaissance.

Dans ce chapitre, un ensemble d'études sera présenté sur les données obtenues par la ville de Baie-Comeau, et donc cette étude est divisée en trois parties. Tout d'abord, l'analyse des données avec une description détaillée du réseau de la ville. Ensuite, nous analysons la relation entre la consommation d'électricité et le comportement des individus et des institutions en hiver et aux autres saisons en calculant le coefficient de corrélation. Enfin, une étude de la prédictibilité des données pour voir si les données collectées peuvent prédire la consommation d'électricité.

4.2 Description de réseau de la ville Baie-Comeau

Il s'avère que la ville de Baie-Comeau souhaite utiliser la technologie de l'intelligence artificiel plus précisément la commande adaptative par intelligence artificielle de la charge flexible, de la production distribuée et du stockage dans les réseaux de distribution pour

mieux gérer l'électricité. Sachant que la ville est responsable de la production et de la distribution d'énergie électrique, elle assure son énergie électrique grâce à l'énergie hydroélectrique de la rivière Manicouagan. Il appert que la ville utilise le logiciel d'analyse de réseaux électriques « CYME ». Ce logiciel capable de combiner tous les outils de modélisation et d'analyse nécessaires pour effectuer différents types de simulations liées à la planification du système de distribution électrique. Les moteurs informatiques prennent en charge les modèles de distribution équilibrés ou déséquilibrés qui ont différentes combinaisons de phases et fonctionnent dans une configuration radiale, toroïdale ou en réseau [64]. Dans ce contexte, la ville espère améliorer les outils de travail. Dans ce projet nous utilisons l'analyse de séries chronologiques pour aider à la planification et l'optimisation de la consommation électrique via une prévision de la consommation.

4.2.1 Les types de consommateurs

Une étude de la consommation requiert une analyse des types de consommateurs et de lignes de distribution utilisées dans le réseau. De ce point de vue, la ville de Baie-Comeau nous a fourni des informations riches et importantes sur la qualité des consommateurs et la quantité d'énergie électrique consommée. Rappelons-nous que les pics de consommation en hiver représentent un fardeau pour la ville et pour les autres opérateurs dans la province, y compris le géant Hydro-Québec. Concernant l'échantillonnage pour le mois de novembre, pour toutes les lignes de distribution d'électricité, les consommateurs de la ville sont répartis en quatre catégories, qui sont les suivantes : résidentiel, industriel, commercial et autres.

Le graphique de la Figure 4-1 montre le type de consommateurs selon chaque ligne de distribution.

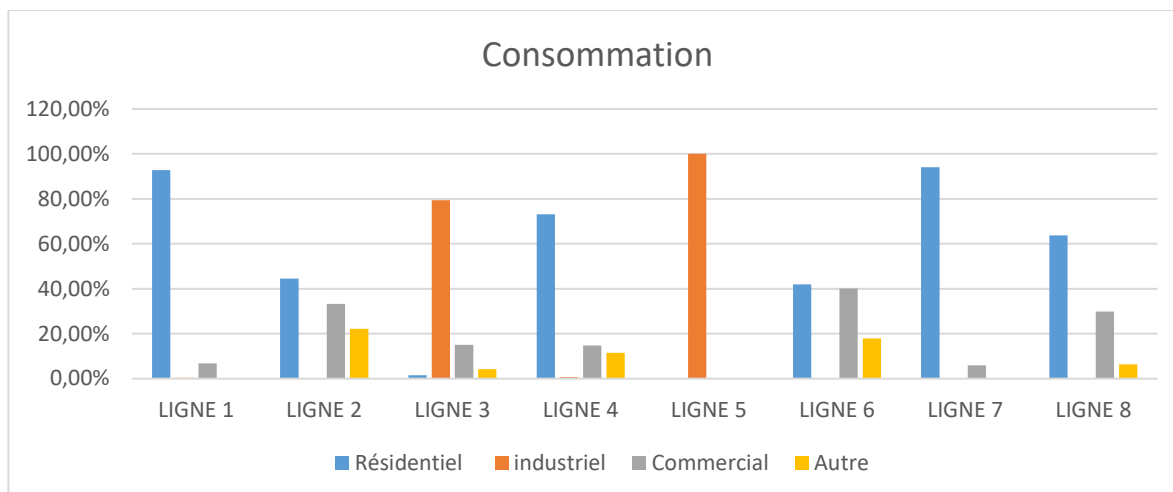


Figure 4-1: Consommation de la Ville Baie-Comeau par ligne et par client.

Bien que cet échantillon porte sur une journée entière du mois de novembre, il montre la répartition de la consommation dans les lignes de distribution d'électricité, donc la consommation d'électricité selon les graphiques varie d'une ligne de distribution à l'autre. Par exemple, la cinquième ligne est dite industrielle, la septième ligne est principalement résidentielle avec une minorité commerciale, tandis que les autres lignes sont des lignes hybrides. La première ligne de distribution contient 92,83 % résidentiel et 6,85% commercial et pour la deuxième ligne répartie sur trois types de consommation électrique 44,58 % résidentiel, 33,29 % commercial et 22,12 %. La troisième ligne est également reconnue pour la diversité des consommateurs, d'autant plus qu'elle comprend 79,36 % industriel, 15 % commercial et 1,47 % résidentiel. Alors que l'on peut dire que les lignes 4, 6 et 8, la plupart des consommateurs de ces lignes sont résidentiels et répartis comme suit : 73,09 %, 44,81 % et 63,78 %, les commerciales sont réparties comme suit : 14,72 %, 36,12 % et 29,91%, tandis que d'autres types de clients ont moins de demande. Cette portion de la consommation d'électricité est répartie comme suit : 11,53 %, 19,07 % et 6,31 %, on en conclut donc que les lignes 4, 6 et 8 sont majoritairement résidentielles.

4.2.2 Données relatives aux clients

Un échantillon prélevé un jour de novembre a montré que la ville consommait 29,40 MW de puissance électrique totale de toutes les lignes de distribution. Le graphique ci-dessous montre la répartition en pourcentage par catégorie de consommateur.

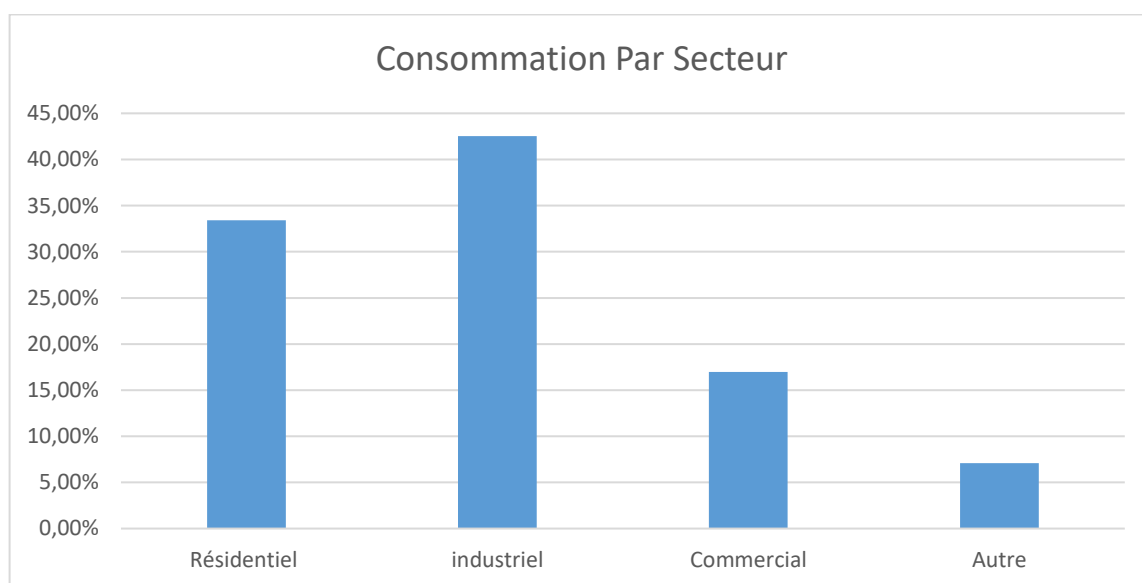


Figure 4-2: Répartition de la demande en puissance de la Ville Baie-Comeau par client ou secteur.

À travers le graphique, il est clair que la ville est une ville industrielle avec distinction, car le secteur industriel comprend 42,05 %, suivi du secteur résidentiel avec 33,03 %, tandis que le reste est occupé par le secteur commercial et d'autres types de clients divisés en ce qui suit 17,92 % et 7 %.

4.2.3 Données relatives à la consommation par la ligne.

Le graphique ci-dessous montre la distribution de la consommation électrique, en pourcentage, de chaque ligne de distribution électrique pour le même échantillon de pic de puissance pour une journée en novembre 2020.

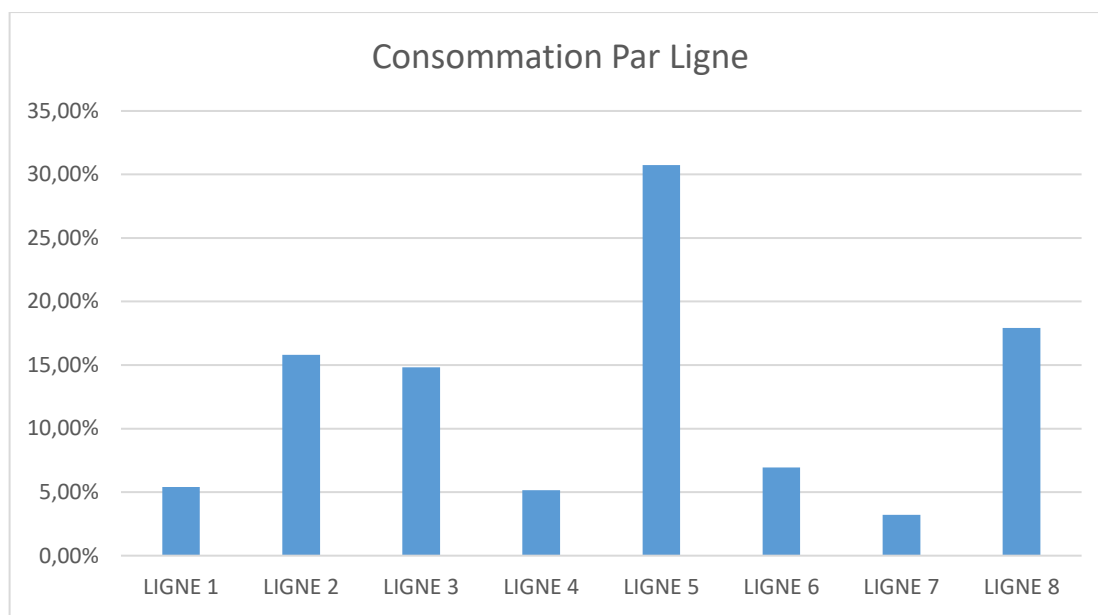


Figure 4-3: Consommation de la Ville Baie Comeau par ligne.

Comme nous l'avons dit plus tôt, la ville a consommé un total de 29,40 MW, la plus grande part est allée à la cinquième ligne avec 30,73 %, car c'est la ligne industrielle bien connue, suivie de la huitième ligne avec 17,92 % qui est une ligne hybride entre résidentiel, commercial et autres, la deuxième ligne est achevée à 15,81%, cette ligne est également une ligne hybride, tandis que les lignes qui consomment peu sont les lignes qui contiennent des logements, commerciaux et autres. La première ligne 5,41 %, la quatrième ligne 5,15% et la sixième ligne 6,95 %, à l'exception de la septième ligne 3,21 %, qui est une ligne entièrement résidentielle.

4.3 Analyse des données

4.3.1 Les données de la consommation électrique

Les données obtenues par la ville de Baie-Comeau correspondent à la consommation d'électricité pour l'année 2020. Un échantillon de chaque ligne de distribution a été prélevé

toutes les 15 minutes; cette opération a été faite pour les huit lignes, ce qui nous permet aussi d'obtenir la demande en puissance. Le prélèvement représente le courant (A), la tension (V), le facteur de puissance, la puissance réactive (KVA), la puissance d'apparente (KVA) et la puissance active (KW). Dans cette étude, nous nous intéressons à la puissance active (MW) de chaque ligne et elle est représentée par la figure suivante.

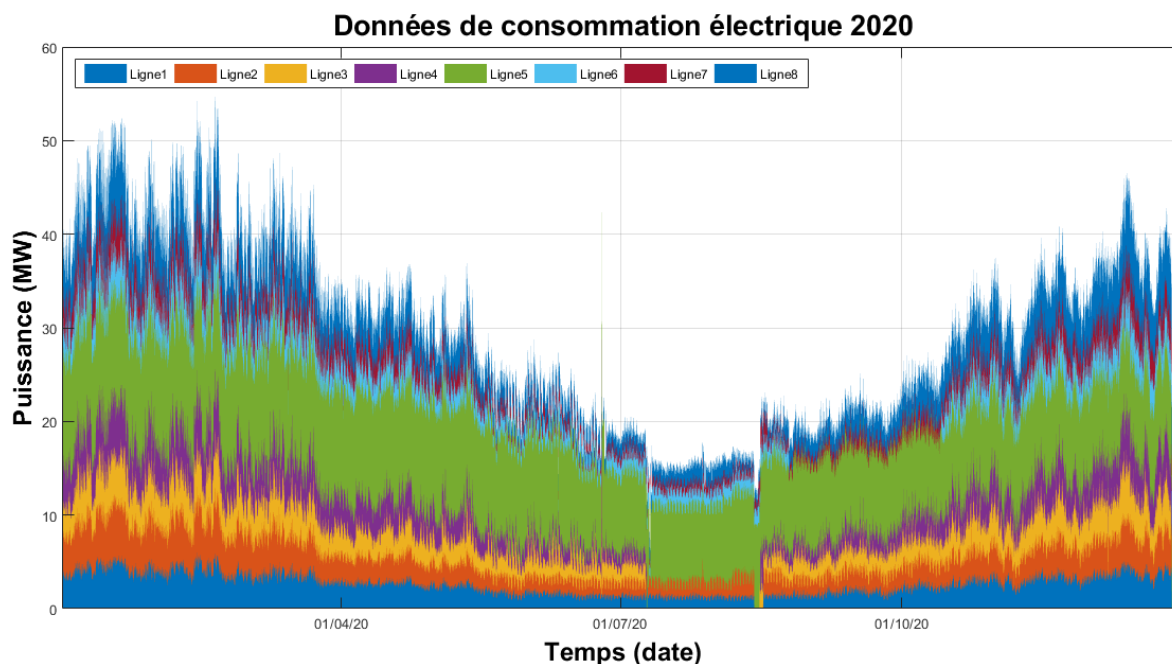


Figure 4-4: Données réelles de la consommation électrique.

Notez que ces données obtenues par la ville sont, en général, des données réelles, à l'exception de quelques échantillons manquants dans certaines lignes de distribution d'électricité. Par conséquent, nous avons dû faire quelques ajouts qui n'affectent pas négativement les données, en bref, admonester les données manquantes avec la valeur de l'échantillon précédent. La figure ci-dessous montre la courbe de consommation après le changement apporté aux données.

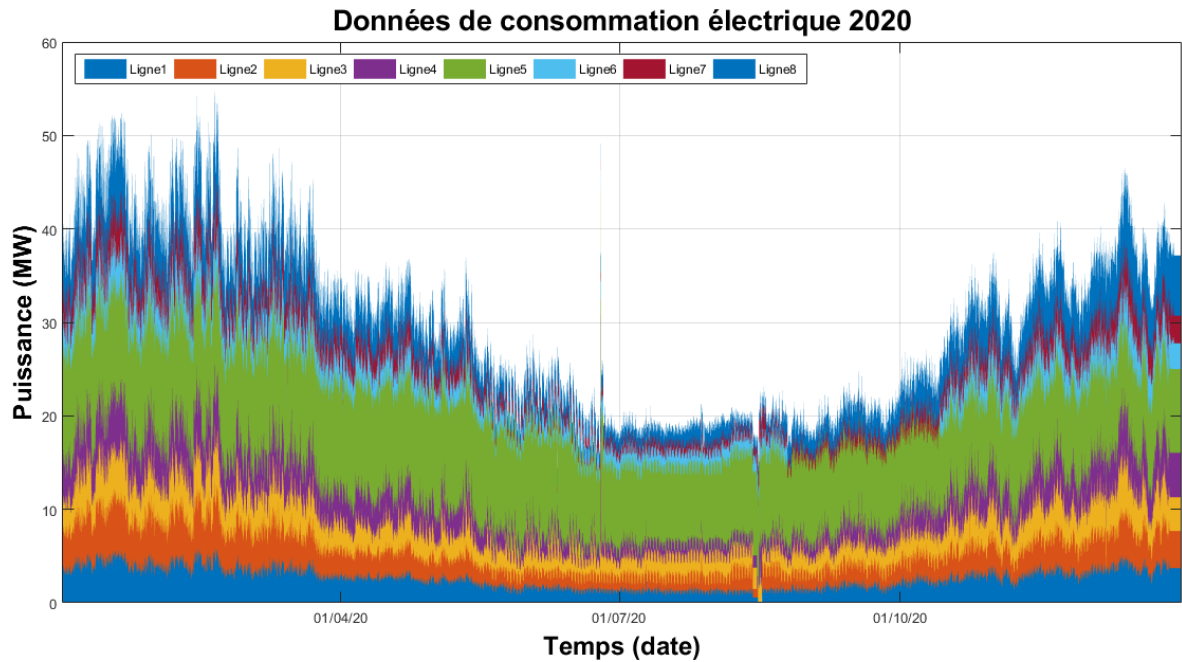


Figure 4-5: Courbes de la consommation électrique après l'élimination de données aberrantes.

4.3.2 Les données météorologiques

Dans ce cadre, nous avons téléchargé les données météorologiques associées à la ville depuis le site web ([// simeb.ca](http://simeb.ca)). Ces données donnent l'information de la température sèche, la température humide, la vitesse de vent et l'irradiation solaire. Nous essayerons d'expliquer le comportement de consommation électrique dans plusieurs cas; par exemple, dans une journée humide et dans une journée sèche; c'est pourquoi nous avons désassemblé toutes les données par mois, par type de consommateur et par ligne de distribution pour faciliter la manipulation dans MATLAB. Les courbes de données météorologiques pour l'année 2020 de la ville sont présentées dans la Figure 4-6.

Données météorologiques 2020

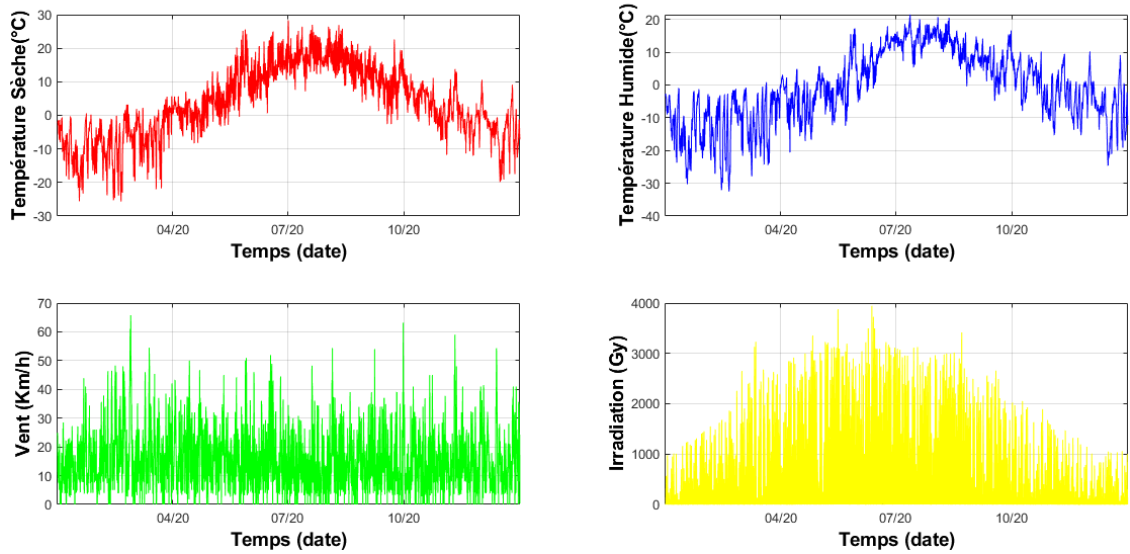


Figure 4-6: Données météorologique de la ville Baie-Comeau.

4.3.3 *Corrélation entre les variables*

Lorsque nous avons reçu les données météorologiques et de consommation liées à la ville, nous avons procédé à les corrélés pour vérifier s'il existe une relation entre la météo et la consommation électrique. Avant cela, nous avons extrapolé les données météorologiques d'un échantillon horaire à un échantillon de quart d'heure à l'aide de MATLAB. Pour interpréter le résultat, nous avons calculé le coefficient de corrélation [65]. Il faut suivre la droite de la Figure 4-7 pour connaître le type de corrélation. Selon la valeur du coefficient, la corrélation peut être parfaite, forte, moyenne, faible ou inexistante.

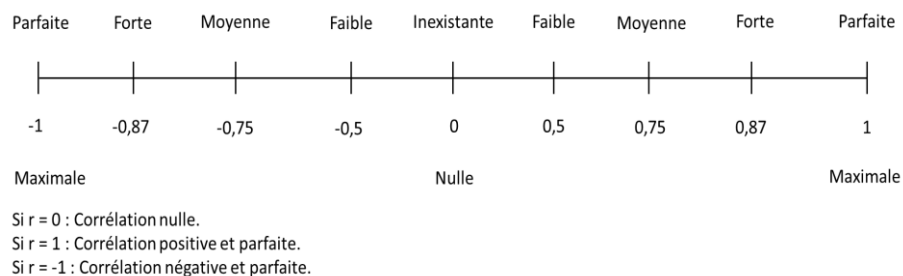


Figure 4-7: Interprétation du coefficient de corrélation [65].

Cette analyse a été présentée par le coefficient de corrélation utilisé pour évaluer la dépendance mensuelle et annuelle entre les variables météorologiques et la consommation d'électricité (Puissance en kW) pour chaque ligne de distribution comme le montrent les figures suivantes.

Dans la première ligne de distribution, les valeurs obtenues du coefficient donnent une forte corrélation pour la température sèche et humide, mais une corrélation nulle pour le rayonnement solaire et la vitesse du vent. En raison de problèmes d'exploration de données, le coefficient de corrélation pour chaque mois donne des résultats différents. Dans les premiers mois de l'année 2020, on obtient une forte corrélation de température sèche et humide, mais pour le reste de l'année une corrélation nulle. Le coefficient de corrélation du rayonnement solaire et de la vitesse du vent donne une corrélation nulle pour tous les mois.

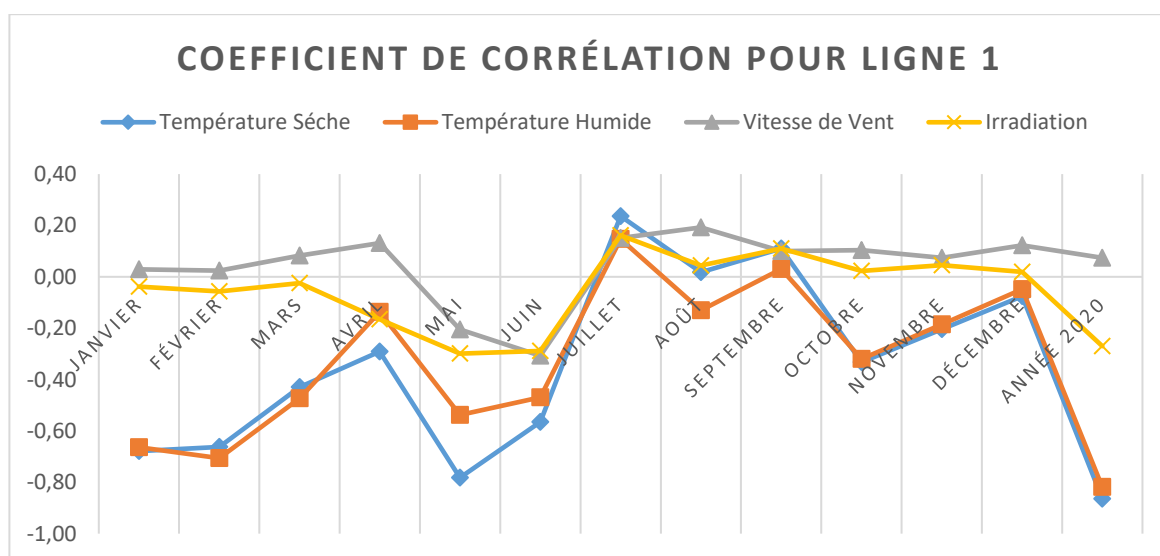


Figure 4-8: Corrélation entre la puissance et les variables pour la ligne 1.

Ce que nous avons dit sur la première ligne de distribution s'applique à la deuxième ligne (Figure 4-9). Le coefficient donne une forte corrélation pour la température sèche et humide, mais une corrélation nulle pour le rayonnement et la vitesse du vent pendant toute l'année 2020. La corrélation change mensuellement dans les températures sèches et humides. Il n'y a pas de changement concernant la vitesse du vent et le rayonnement solaire, et le coefficient de corrélation reste nul pour tous les mois sauf en septembre. Le coefficient de corrélation augmente de nulle à faible.

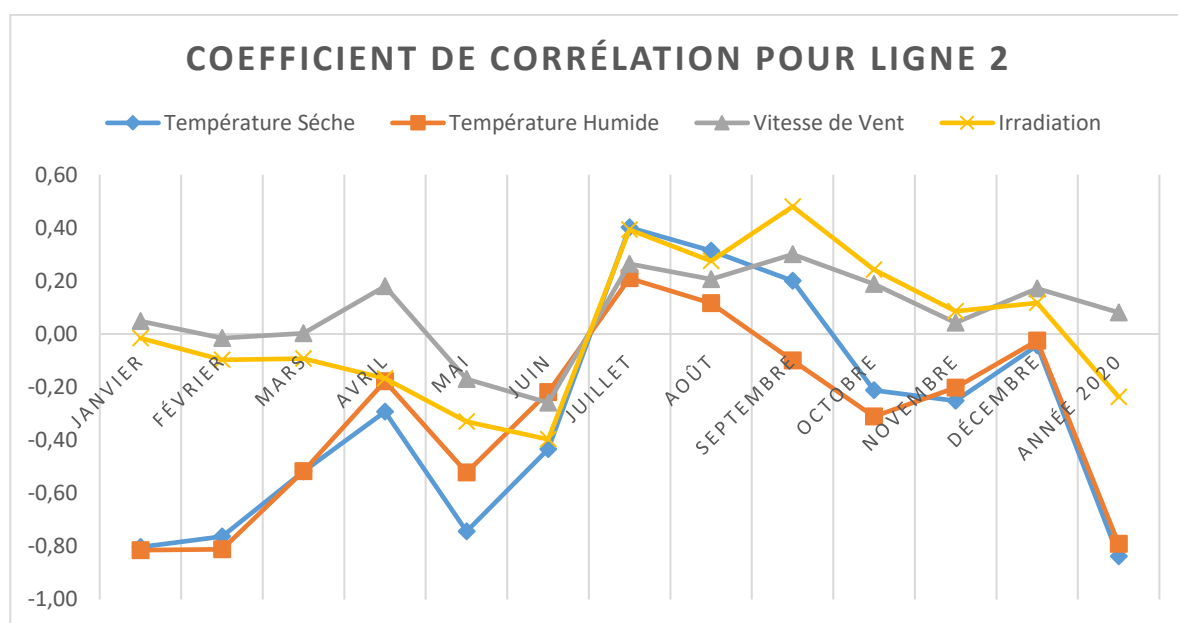


Figure 4-9: Corrélation entre la puissance et les variables pour la ligne 2.

Analysant la troisième ligne de distribution (Figure 4-10), le coefficient de corrélation varie de faible à moyen en ce qui concerne la température humide et sèche, car cette ligne alimente la plupart des usines. Ce segment de consommateurs ne se soucie pas de la météo, car sa consommation reste constante.

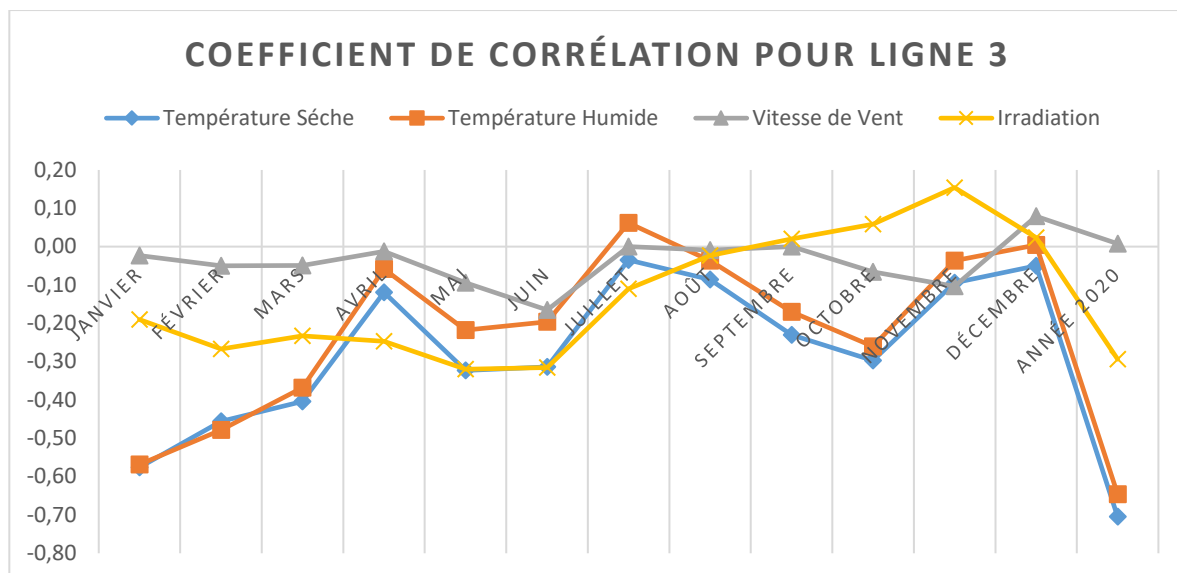


Figure 4-10: Corrélation entre la puissance et les variables pour la ligne 3.

La Figure 4-11 montre les résultats de corrélation pour la quatrième ligne ; selon ces résultats, la température sèche et humide donne une forte corrélation pour l'année 2020 en revanche l'irradiation et la vitesse du vent donnent une corrélation nulle. Similairement que dans les lignes 1 et 2, la corrélation mensuelle ne change pas beaucoup.

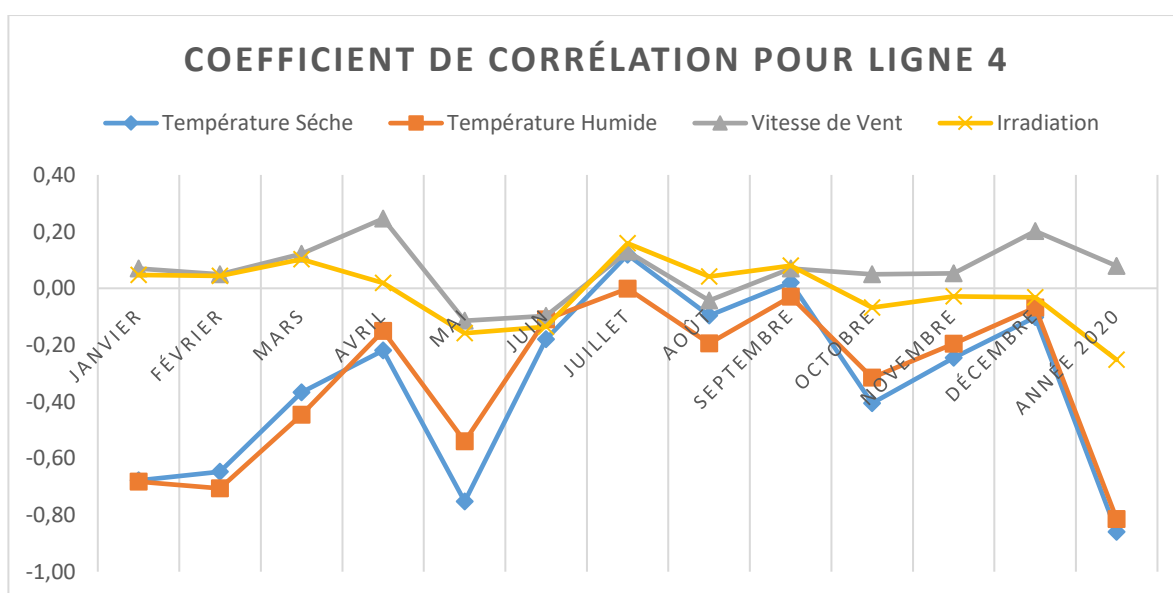


Figure 4-11: Corrélation entre la puissance et les variables pour la ligne 4.

L'analyse réalisée pour la cinquième ligne (ligne occupée par un centre des données) est présentée dans la Figure 4-12. Comme il est connu que les centres des données s'installent dans la région froide, il résulte normal que la corrélation avec la température soit nulle en période hivernale. Lorsque la température estivale augmente, les climatiseurs s'allument, donc à l'été le coefficient de corrélation donne une corrélation différente.

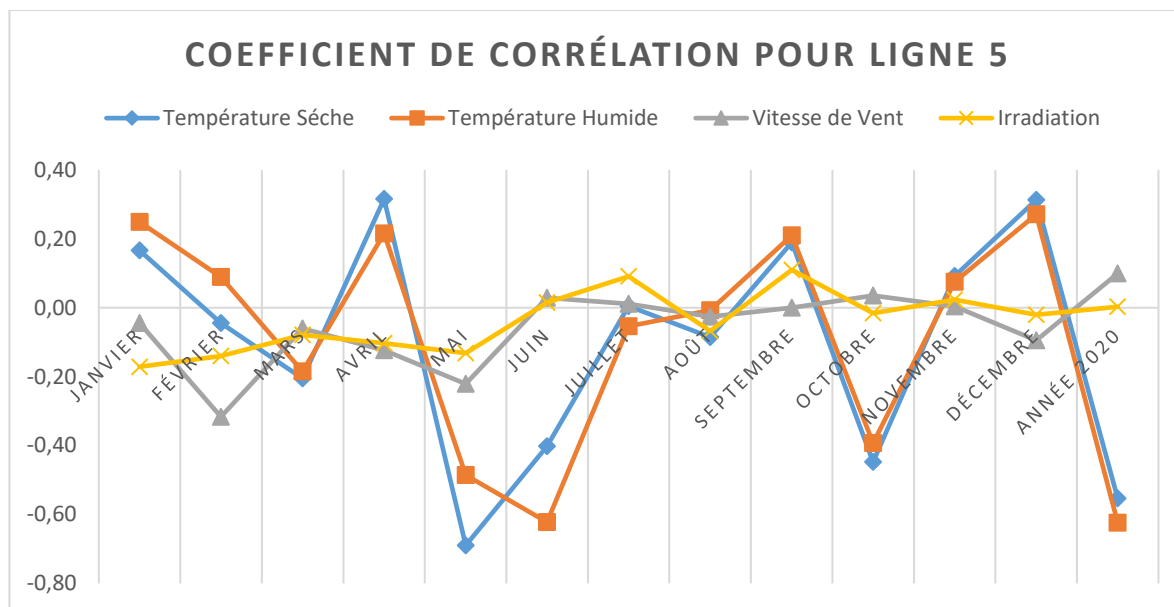


Figure 4-12: Corrélation entre la puissance et les variables pour la ligne 5.

Le coefficient de corrélation de la sixième ligne de distribution (Figure 4-13) montre qu'en hiver il existe une corrélation forte et moyenne avec la température sèche et humide. L'irradiation solaire en été donne une corrélation moyenne. En analysant l'année complète, le coefficient donne une forte corrélation avec la température.

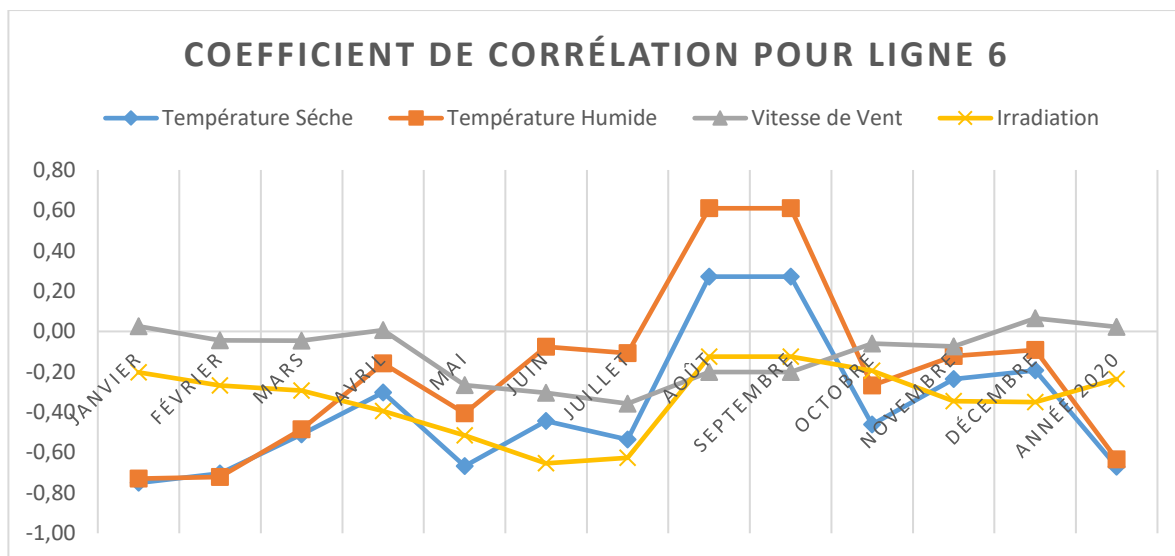


Figure 4-13: Corrélation entre la puissance et les variables pour la ligne 6.

La ligne de distribution 7 est orientée vers la consommation résidentielle, on note donc que le coefficient de corrélation est très fort aussi bien pour les températures sèches que humides par contre la corrélation nulle pour la vitesse du vent et le rayonnement solaire (Figure 4-14).

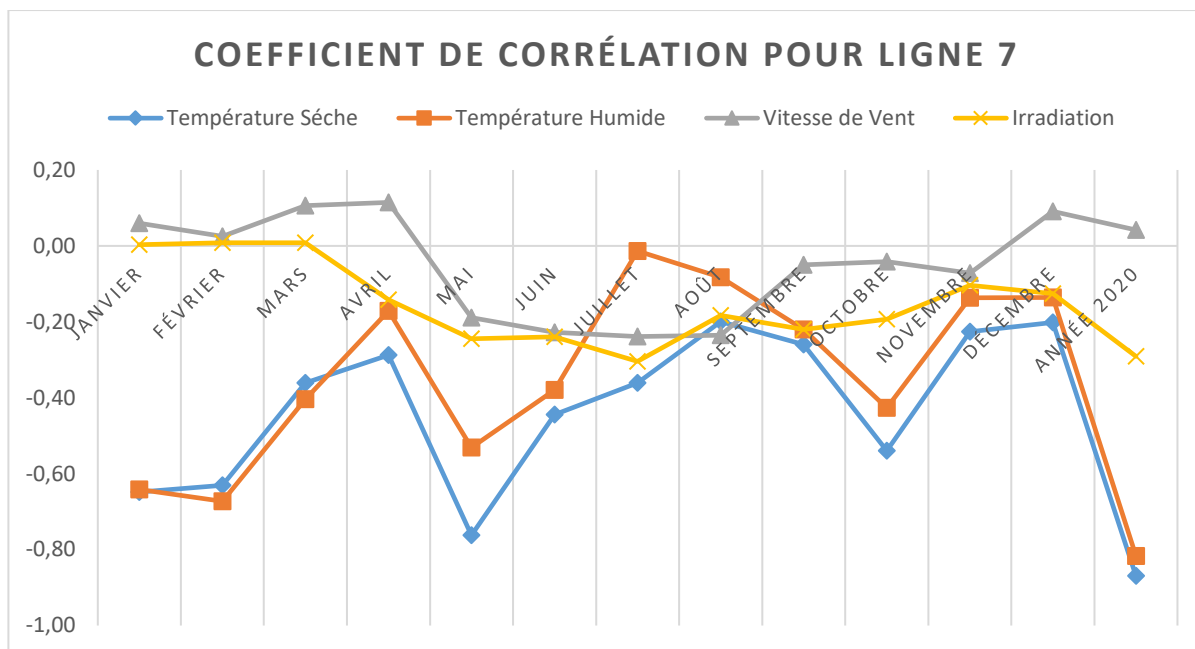


Figure 4-14 : Corrélation entre la puissance et les variables pour la ligne 7.

Les résultats obtenus pour la huitième ligne de distribution sont très similaires à ceux obtenus pour les lignes de distribution suivantes 1, 2, 4 et 6. Les valeurs obtenues du coefficient montrent une forte association en hiver avec des températures sèches et humides. L'irradiation solaire et la vitesse du vent ont une corrélation presque nulle (Figure 4-15).

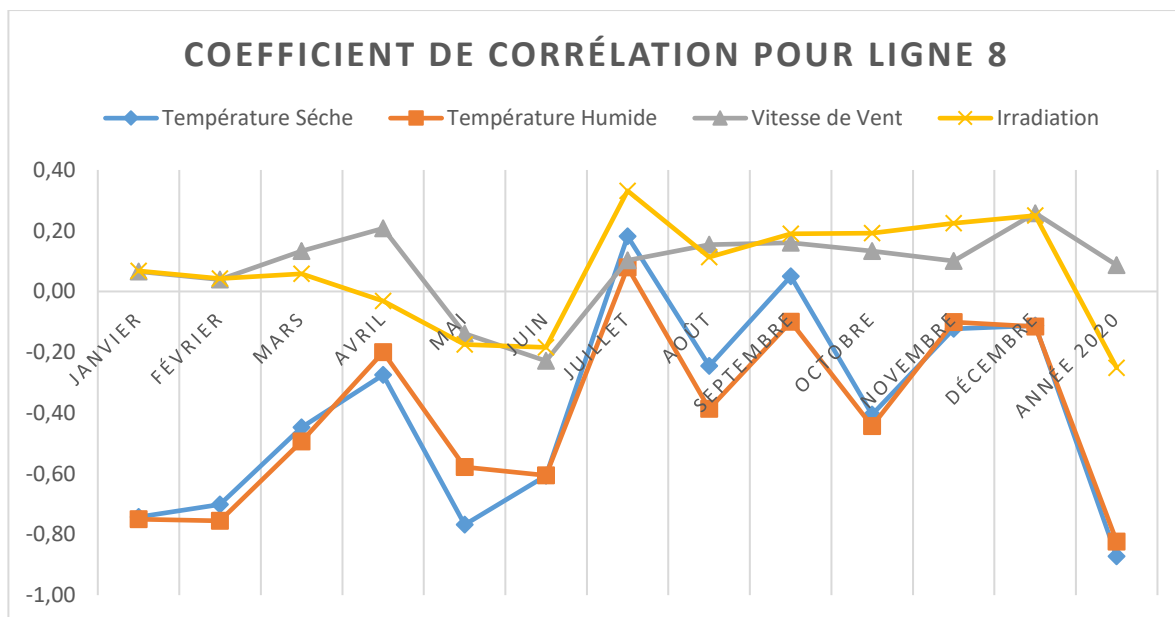


Figure 4-15: Corrélation entre la puissance et les variables pour la ligne 8.

Suite à l'analyse effectuée, différents résultats apparaissent en termes de corrélation, par exemple, la corrélation entre la puissance électrique et la vitesse du vent dans toutes les lignes est très faible ou presque nulle; c'est le même cas pour la puissance et irradiation solaire, en revanche la corrélation de la puissance avec la température sèche et humide varie entre forte et moyenne selon la ligne de distribution.

4.3.4 Analyse des pics de consommation hivernale

L'analyse des pics de consommation permet d'identifier les voies possibles pour réaliser des économies par une redistribution de la consommation d'électricité. Une fois les pics de puissance identifiés grâce au profil de charge, l'objectif est de les éviter pour réduire en même temps les factures d'électricité et améliorer le facteur d'utilisation du système de distribution.

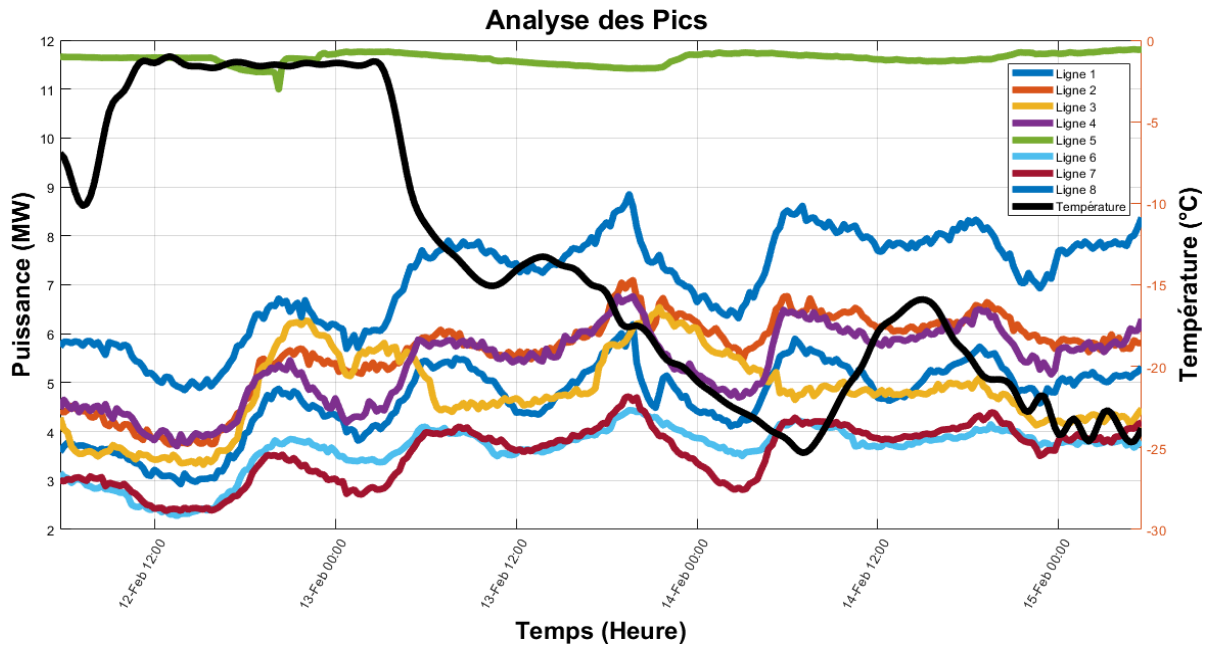


Figure 4-16: Analyse des pics de la consommation.

D'après des extraits d'une des journées de la saison d'hiver, plus précisément le 13 février 2020 (jeudi), il est clair que la pointe de consommation se produit simultanément entre toutes les lignes de distribution, à l'exception de celles à caractère industriel.

Pour vérifier la coïncidence des pics de la consommation, nous allons calculer le facteur de crête, c'est-à-dire le rapport entre l'amplitude de crête du signal et la valeur efficace du signal. Pour calculer le facteur de crête [66], nous utilisons l'équation suivante [66]:

$$C = \frac{|X|_{peak}}{X_{rms}} \quad (4-1)$$

Le tableau ci-dessous représente les résultats du facteur crête pour les pics du matin et du soir, puis leurs résultats ont été comparés au pic moyen d'hiver dans la période comprise entre janvier et avril 2020.

Tableau 4-1: Facteur de crête pour le jour et le soir.

Ligne	Pointe du matin		Pointe du soir		Pics moyens en hiver
	Temps	Facteur de crête	Temps	Facteur de crête	Facteur de crête
L1	2020-02-13 08:00	1,17	2020-02-13 19:45	1,31	2,07
L2	2020-02-13 07:00	1,07	2020-02-13 19:45	1,25	2,14
L3	2020-02-13 03:15	1,19	2020-02-13 21:30	1,34	2,11
L4	2020-02-13 05:30	1,08	2020-02-13 19:45	1,25	1,91
L5	—	—	—	—	1,09
L6	2020-02-13 06:00	1,12	2020-02-13 19:30	1,21	1,75
L7	2020-02-13 08:15	1,14	2020-02-13 19:30	1,31	2,01
L8	2020-02-13 07:30	1,10	2020-02-13 19:30	1,24	1,93

À travers les résultats présentés dans le tableau, l'analyse préliminaire réfute que la synchronisation totale des pics de consommation soit présente dans toutes les lignes de distribution. Par exemple, l'heure de pointe du matin commence de 5 h 30 à 8 h 15, mais le soir, on peut dire que le pic coïncide dans certaines lignes, comme les lignes 6, 7 et 8, qui étaient à la même heure 19h30, de même pour les lignes 1, 2 et 4, qui étaient à la même heure 19h45. On ne peut pas juger les lignes 3 et 5, car elles sont des lignes alimentant majoritairement des clients industriels.

4.3.5 Étude de la prédictibilité des données

Comme nous l'avons mentionné au chapitre trois, si nous voulons définir et étudier des séries chronologiques, nous devons tester d'abord le degré de prévisibilité. La réponse réside dans le calcul du coefficient ou de l'exposant de Hurst pour classer les données des séries chronologiques. Elles sont classées en trois catégories : séries aléatoires, séries anti-persistantes et séries persistantes. Nous avons calculé l'exposant de Hurst pour la série temporelle de la charge utilisant la bibliothèque Hurst de Python. Au Tableau 4-2, on retrouve les résultats du coefficient de Hurst pour l'ensemble des données de séries chronologiques pour les lignes de distribution d'électricité de la ville de Baie-Comeau. Dans des conditions

de temps évolutives, cela signifie tester un mois entier, janvier, février et mars et en plus les mois les plus froids d'hiver de janvier à mars et terminer en calculant l'exposant de Hurst pour l'ensemble de 2020.

Tableau 4-2 : Le coefficient de Hurst

Durée	Coefficient de Hurst							
	Ligne 1	Ligne 2	Ligne 3	Ligne 4	Ligne 5	Ligne 6	Ligne 7	Ligne 8
Janvier	0,3238	0,4396	0,4492	0,3838	0,2895	0,3948	0,3704	0,4058
Fevrier	0,3964	0,4336	0,459	0,4208	0,3352	0,3871	0,4329	0,4218
Mars	0,359	0,4816	0,4401	0,3776	0,4826	0,3669	0,3631	0,4175
Trois Mois	0,3389	0,4122	0,4104	0,3609	0,3674	0,3463	0,3583	0,3874
Un an	0,4954	0,4969	0,3557	0,4599	0,4600	0,4465	0,4328	0,5075

À travers les résultats du Tableau 4-2, il est clair que la catégorie de la série chronologique dans toutes les lignes est une série anti-persistante, en particulier dans le test de chaque mois d'hiver et des trois mois d'hiver, car le coefficient H varie entre $0 < H < 0,5$; le test de l'ensemble de données 2020 indique qu'il existe une série anti-persistence, car le coefficient H varie entre $0 < H < 0,5$ même si le type de consommateurs et la quantité de consommation par ligne de distribution diffèrent.

4.4 Synthèse du chapitre

Le quatrième chapitre, affirmant que la pointe de consommation est liée avec un degré plus ou moins important à la température extérieure en fonction de chaque ligne de distribution électrique et du type de consommateur.

Théoriquement et pratiquement, il existe une corrélation entre la consommation d'électricité et la température sèche et humide de chaque mois qui varie d'un mois à l'autre. Préalablement à cette conclusion, la qualité des consommateurs sur l'ensemble du réseau pour l'année 2020 a été analysée et auditée.

Cela nous amène au chapitre suivant de la recherche qui se penche sur une étude comparative des modèles de prévision pour choisir le modèle le plus approprié pour chaque ligne de distribution.

Chapitre 5 - Proposition et analyse des résultats

5.1 Introduction

Nous avons présenté dans le chapitre précédent la relation directe entre la météo et les heures de pointe de consommation d'électricité pendant la saison hivernale. Le travail présenté dans ce chapitre a pour objectif principal de valider les modèles de prévision sélectionnés en utilisant les données météorologiques et l'historique de consommation comme entrées, et ainsi proposer un scénario d'optimisation de la consommation d'électricité en prévoyant la demande future.

5.2 Proposition de méthode de prévision en vue de l'optimisation

La Figure 5-1 montre la perception générale de ce qui doit être fait pour optimiser les réseaux de distribution d'électricité grâce à l'intelligence artificielle. Premièrement, on a collecté des données et calculé le coefficient de corrélation entre les variables. Deuxièmement on a proposé un modèle en fonction de la variable appropriée choisie qui peut être utilisée pour prévoir la consommation électrique dans laquelle les réseaux de distribution d'électricité sont optimisés.

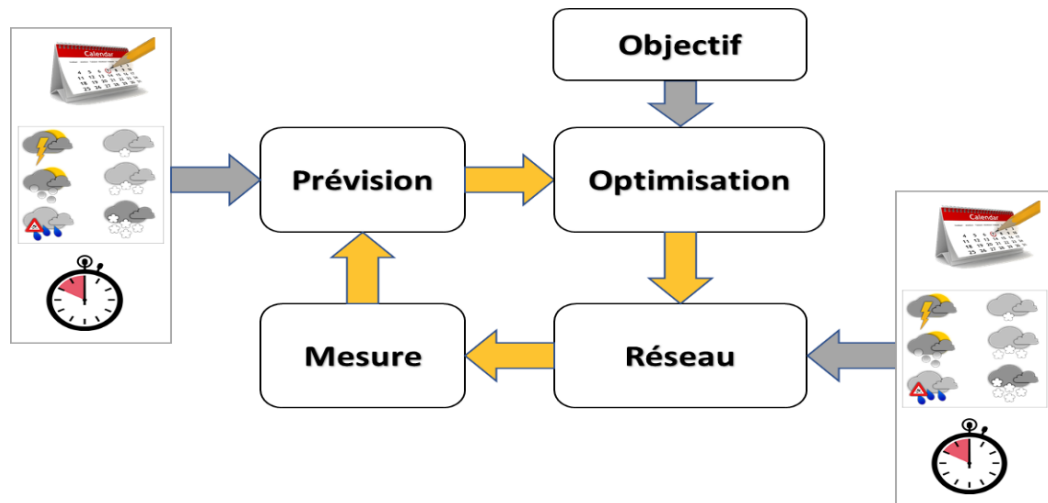


Figure 5-1: L'approche générale du projet.

Une stratégie courante pour améliorer le réseau de distribution consiste à prévoir la consommation à l'aide de l'intelligence artificielle. Certaines mesures peuvent être prises telles que le stockage de l'électricité excédentaire dans des batteries, sa conversion en hydrogène (sous forme liquide ou gazeuse) pour une utilisation aux heures de pointe et l'installation d'un système électrique hybride éolien et solaire. Le délestage électrique est un mécanisme de résolution pratique et non dispendieux pour les opérateurs des réseaux de distribution. La Figure 5-2 représente graphiquement le réseau de la ville Baie-Comeau avec une approche de gestion basée sur l'utilisation de l'intelligence artificielle.

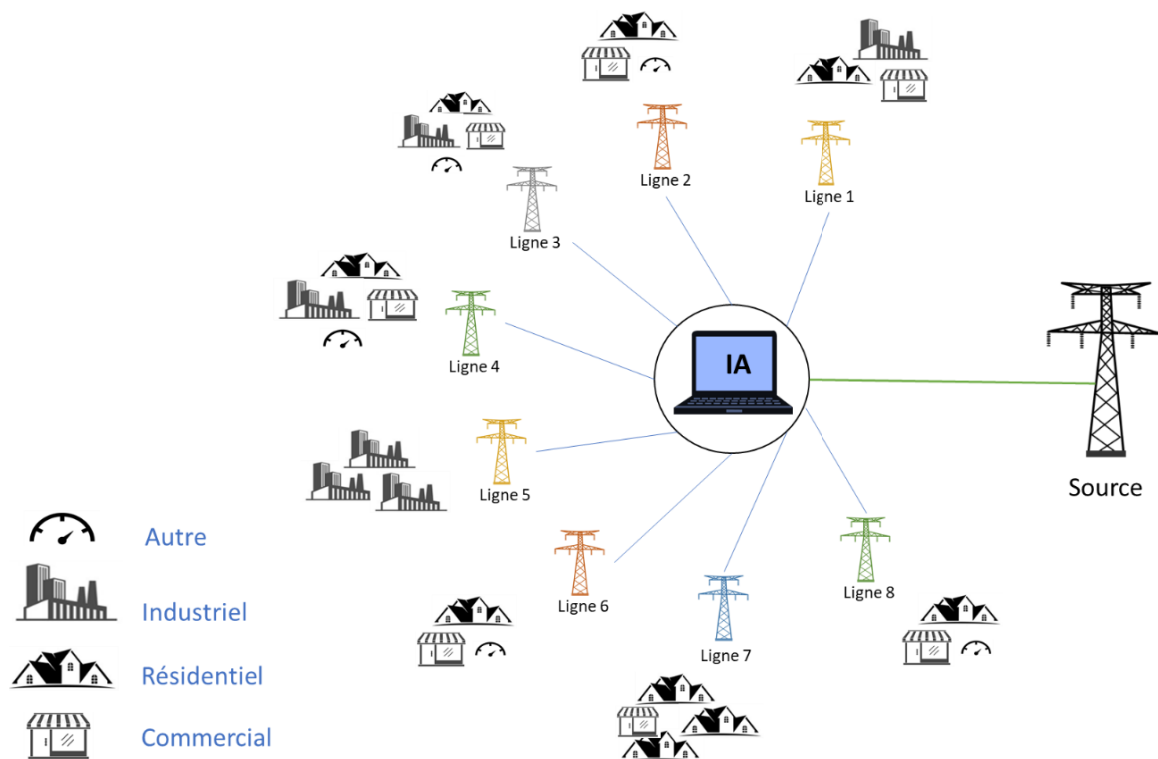


Figure 5-2 : Réseau de distribution de la ville Baie-Comeau.

5.3 Prédiction de la consommation

5.3.1 Méthodologie et base de données

La clé d'une prédiction réussie de la demande dépend de la quantité et de la qualité des données. Ainsi, dans cette étude, nous avons utilisé les données de consommation d'électricité et les informations météorologiques de la ville de Baie-Comeau pour prévoir la consommation du futur. À ce stade, les données d'origine (365 jours) sont décomposées en données de test et d'entraînement selon l'essai. Les données de test ont été utilisées comme moyen complémentaire pour analyser la performance du modèle avec la base de données. Cela se fait en calculant le taux d'erreur entre la consommation prévue et la consommation réelle. La figure 5-3 montre la méthodologie de travail.

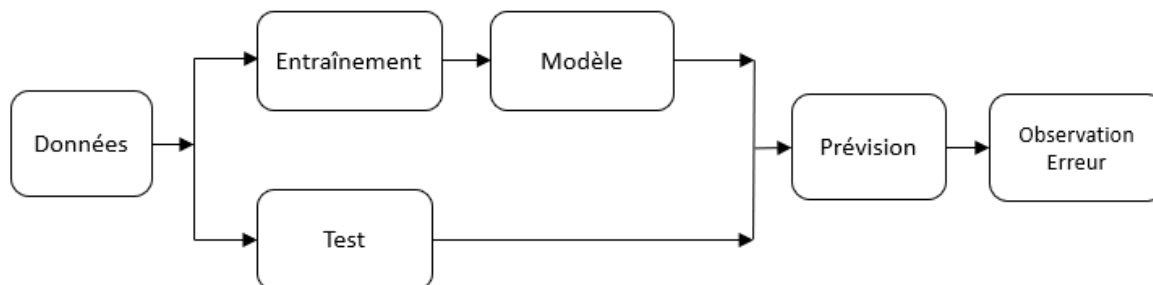


Figure 5-3: Schéma approximatif de programmation.

Il est toujours utile de rappeler la méthodologie qui a été adoptée pour entraîner cette IA à la prédiction. La méthodologie comprend les étapes suivantes :

a) Prétraitement de base de données.

Dans cette section, nous expliquons plus en détail comment gérer les données, les fonctionnalités doivent être sélectionnées pour déterminer la quantité de données pour entraîner les modèles sélectionnés.

Tout d'abord, les données doivent être adaptées pour correspondre aux spécifications du cahier de charges. Après avoir obtenu les données, nous les avons divisées en données d'entraînement et données de test. Pour obtenir une prédiction de charge à très court terme, nous avons entraîné les modèles avec une quantité de données correspondante à un mois (30 jours) pour prévoir le prochain jour. C'est-à-dire les prochaines 24 heures, mais à condition qu'affiche la prédiction progressive horaire (heure par heure) jusqu'à la fin de boucle de 24 heures. La Figure 5-4 montre le principe de fonctionnement de la prédiction avec une fenêtre glissante équivalente à 30 jours.

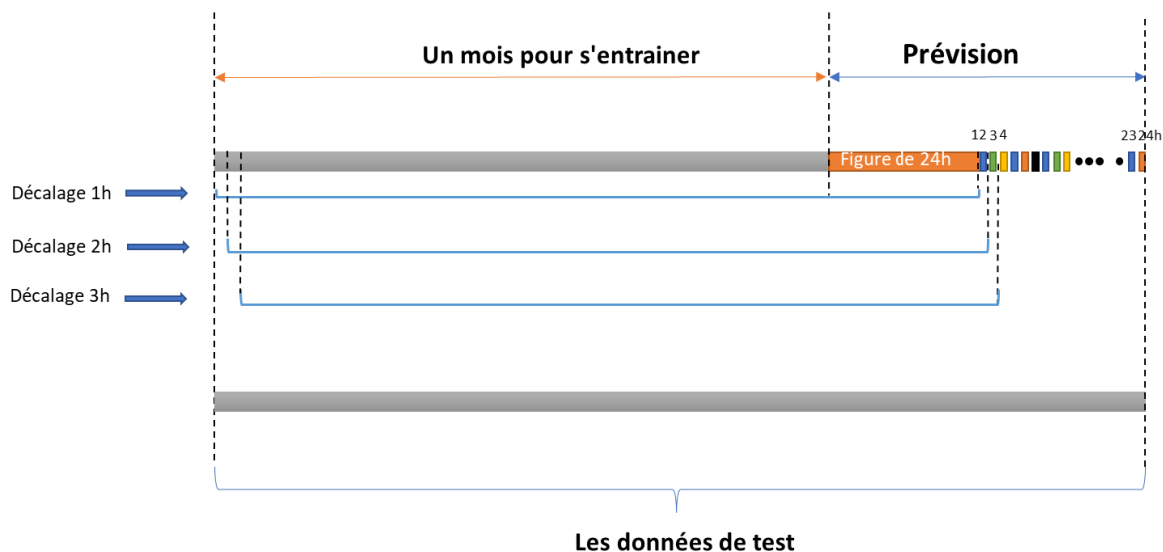


Figure 5-4: Fonctionnement de programmation.

Dans le premier essai, le résultat à obtenir était une prévision sur une période équivalente à 47 heures de consommation future après 30 jours de l'entraînement pour toutes les lignes de distribution sans exception.

Dans la première étape de la prévision après 30 jours d'entraînement, nous avons obtenu une prévision sur 24h, dont 23 heures du dernier jour de mois janvier (31/01/2020) et la première heure de février (01/02/2020).

Dans la deuxième étape de la prévision, un décalage d'une heure a été appliqué et le modèle a été réentraîné avec les données décalées. Mais ce qui nous intéresse, c'est la dernière heure de la Figure 5-4, car elle montre la prévision. L'approche proposée dans cette étude permet de réentraîner le modèle (ou les modèles) à chaque fois que les données sont décalées d'une heure.

b) Modèles utilisés.

Cette prévision a été faite grâce à des modèles de prévision comme le modèle de séries chronologiques ARMAX, ARX et Box Jenkins. Ces modèles dépendent de l'évolution des données dans le temps en plus de plusieurs paramètres. Dans ce travail, le modèle dépend de deux entrées qui sont la température sèche et l'historique de la demande de puissance, comme le montre la figure 5-5.

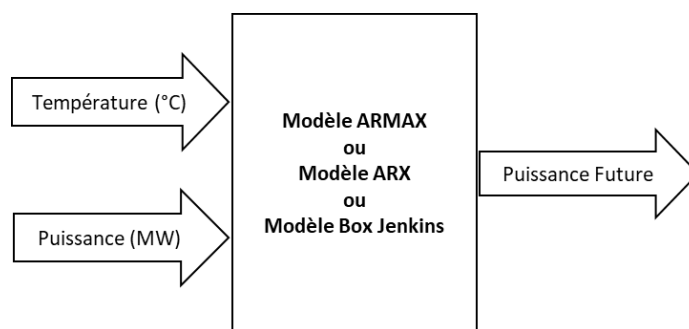


Figure 5-5: Schéma d'entrée et de sortie du modèle.

Un facteur important qui doit être abordé en ce qui concerne les modèles est l'identification des ordres de polynômes pour chaque modèle. L'ordre du polynôme est égal au nombre de coefficients à estimer dans ce polynôme autrement dit l'ordre de polynôme ou le degré de polynôme c'est la relation entre la cible (donné de test) et le prédicteur.

Le Tableau 5-1 résume les ordres polynomiaux pour chaque modèle. Avec de nombreuses expériences de choix d'ordres polynomiaux, le choix s'est porté sur l'ensemble présenté au tableau. Nous avons utilisé les mêmes caractéristiques de modèle pour toutes les lignes de distribution pour voir le comportement de prévision de la consommation électrique.

Tableau 5-1: Les ordres polynomiaux des modèles

Ordres polynomiaux		
Modèle ARX	Modèle ARMAX	Modèle Box Jenkins
na=45	na=1	nb=1
nb=50	nb=45	nc=5
nc=45	nc=50	nd=10
-	nk=1	nf=5
-	-	nk=1

A noter que le polynôme de 1 degré est une régression linéaire simple ; par conséquent, la valeur de degré doit être supérieure à 1. Avec l'augmentation du degré du polynôme, la complexité du modèle augmente également.

c) Analyse d'erreur de prévision.

Afin d'assurer l'efficacité et l'efficience des modèles implémentés pour prévoir la consommation électrique, le taux d'erreur doit être calculé. L'erreur absolue moyenne est choisie, car la base de données contient des valeurs aberrantes.

$$MAE = 100 \times \frac{\sum_{i=1}^m abs(y_i - x_i)}{m} \quad 5-1)$$

Y_i : représente la consommation mesurée

X_i : représente la consommation prévue

M : représente le nombre de caractéristiques

Par la suite, nous calculons l'erreur absolue moyenne en pourcentage pour une journée entière de prévision de la consommation électrique entre la demande et la prévision.

5.3.2 Résultats et discussion

Il est à noter que les résultats et discussions présentés dans cette section concernent l'évaluation des performances des modèles ARMAX, ARX et Box Jenkins avec la

température extérieure et les types de consommateurs. À l'aide des modèles, nous obtiendrons une prévision dynamique de la consommation d'électricité, c'est-à-dire une consommation électrique horaire estimée pour un réseau de distribution de huit lignes dans la ville de Baie-Comeau. Ensuite, les résultats sont comparés à la demande de consommation obtenue par chaque modèle.

Alors que les courbes sélectionnées pour discussion sont le résultat de la prévision pour la première heure de décalage (Figure 5-6), pour toutes les lignes de distribution d'électricité simultanément, afin de détecter les pics de consommation et de réagir pour l'éviter. Tandis que les résultats des 23 heures suivantes sont présentées en annexe.

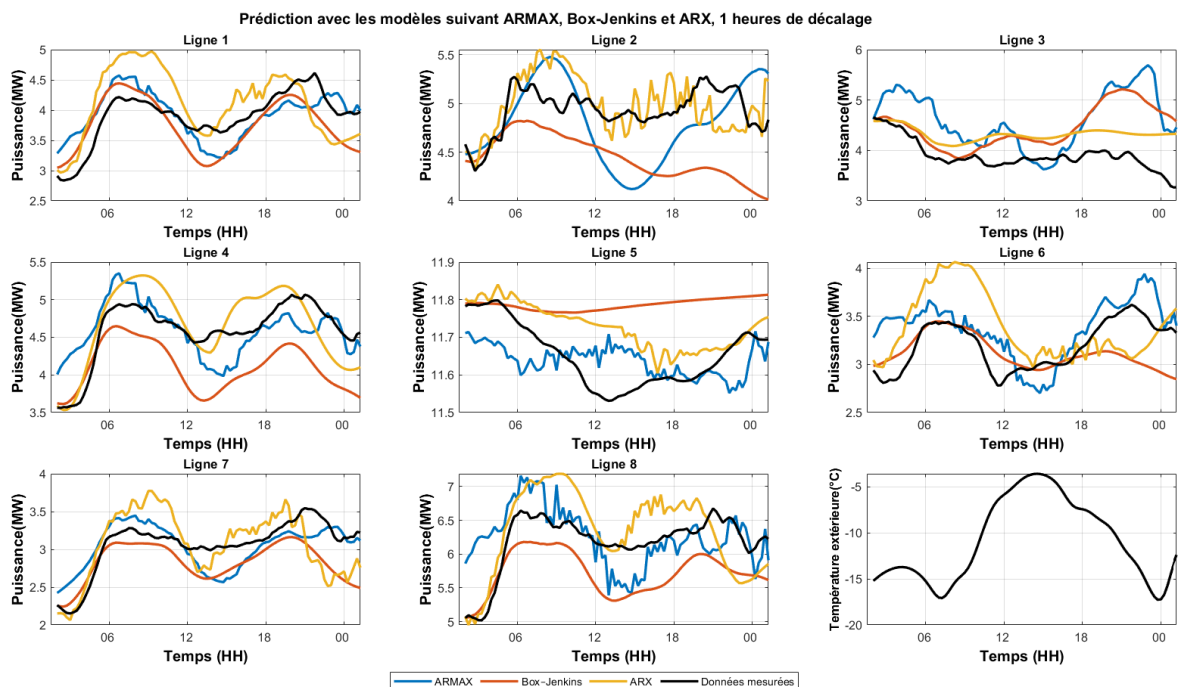


Figure 5-6: Prédiction de la consommation pour toutes les lignes.

D'après la figure 5-6 les trois modèles de prévision donnent des résultats prometteurs, toutefois, la qualité et précision du modèle varie d'une ligne à l'autre.

- Ligne 1 : dans cette ligne de distribution, on peut dire que les résultats du modèle ARMAX correspondent un peu au signal mesuré, le suit ensuite Box-Jenkins et ARX. Les modèles qui représentent mieux le lien entre la température extérieure et la demande sont les modèles de Box-Jenkins et ARMAX.
- Ligne 2 : les résultats du modèle ARX sont similaires au signal mesuré alors qu'on ne peut pas en dire autant des modèles Box Jenkins et ARMAX, car leurs résultats diffèrent du signal mesuré. Dans cette ligne le modèle ARMAX interagit avec la température extérieure, mais cela ne se reflète pas dans la précision de la prévision.
- Ligne 3 : c'est une ligne qui focalise sa consommation sur la zone industrielle en revanche on peut dire qu'aucun modèle ne donne de résultats similaires au signal mesuré.
- Ligne 4 : les modèles ARX et Box Jenkins réagissent à la température extérieure mais pas avec le signal mesuré, la courbe de prévision obtenue avec le modèle ARMAX est similaire à celle des données réelles de la consommation électrique de la ville.
- Ligne 5 : inversement à ce qu'on obtient avec la troisième ligne de distribution, on peut dire que les trois modèles alignent le signal mesuré sans tenir compte de la température extérieure, malgré la distinction de la ligne de distribution étant industrielle.
- Ligne 6 : Tous les modèles correspondent à la température extérieure, de sorte que si la température augmente, la consommation diminue et vice versa. Le modèle ARMAX donne des meilleurs résultats que les modèles Box-Jenkins et ARX.

- Lignes 7 et 8 : pour les deux lignes, le modèle ARMAX permet d'obtenir une courbe similaire à celle du signal mesuré contrairement aux modèles Box-Jenkins et ARX.

Ainsi, nous verrons en annexe que chaque nouvelle heure de boucle programmée permet de modifier les propriétés d'entrée dans le système (heures d'entraînement) et cela permet une prévision mieux adaptée et efficace grâce aux nouvelles données utilisées pour réentraîner les modèles. Ensuite, l'exercice change pour les 30 jours précédents à chaque fois que de nouvelles données sont calculées mais à condition le système garde 30 jours de l'entraînement. Cela peut montrer que les courbes d'entraînement sont uniques à chaque fois que nous entraînons le système pour chaque nouvelle heure.

Afin d'atteindre l'équité de choisir le modèle le plus efficace pour la prévision, l'erreur absolue moyenne a été utilisée pour comparer la précision de chaque modèle séparément pour toutes les lignes de distribution d'électricité. Les tableaux 5-2, 5-3 et 5-4 montrent le résultat obtenu pour chaque modèle avec l'évolution du temps.

Après avoir calculé l'erreur absolue moyenne pour toutes les lignes de distribution, le modèle ARMAX a donné des résultats variables au fil du temps. Dans les lignes hybrides comme les lignes L1, L2, L4, L6, L7 et L8, la moyenne de l'erreur (MAE) est comprise entre 7,3% et 16,7%, et nous en concluons que le taux d'erreur est acceptable. Quant aux lignes qui alimentent complètement les usines, comme la ligne 5, on peut dire que c'est un modèle idéal, mais les lignes qui alimentent les usines en plus des commerces, le discours est différent comme la ligne 3 où la méthode donne des mauvais résultats. Toutefois, on peut dire en général que le modèle ARMAX reste acceptable.

Selon les résultats obtenus et présentés dans le Tableau 5-3, le modèle ARX a une moyenne d'erreur MAE entre 4,9% et 32,1% pour le réseau de distribution, bien que les consommateurs soient différents pour chaque ligne.

Le modèle Box-Jenkins permet d'obtenir une erreur moyenne (MAE) qui varie entre 7% et 125.8%. Ces résultats nous permettent de conclure que le modèle n'est pas adapté à ce type d'application, plus précisément c'est juste pour la configuration actuelle du modèle et avec cette période de l'année hiver 2020.

Dans cette application, le modèle ARX reste le plus approprié pour la troisième ligne de distribution, tandis que pour le reste des lignes le modèle ARMAX est considéré comme celui qui offre des meilleurs résultats.

Tableau 5-2: MAE obtenue avec le modèle ARMAX pour toutes les lignes.

Erreur moyenne absolue de modèle ARMAX en (%)								
Temps	Ligne 1	Ligne 2	Ligne 3	Ligne 4	Ligne 5	Ligne 6	Ligne 7	Ligne 8
1 h	2,38	12,65	71,13	0,16	1,42	17,75	2,36	7,40
2 h	4,35	22,32	66,46	16,58	1,97	17,43	0,94	26,63
3 h	5,81	15,09	70,15	0,61	1,00	10,97	8,88	14,82
4 h	11,36	8,20	76,53	23,04	0,76	7,51	4,38	9,77
5 h	14,36	19,97	76,33	2,25	0,91	15,18	12,81	16,26
6 h	2,50	1,99	73,66	21,12	1,44	8,47	2,20	2,61
7 h	2,97	0,78	40,54	25,99	0,11	7,29	5,70	32,49
8 h	5,46	5,72	61,42	28,23	0,29	7,28	8,84	29,31
9 h	6,63	11,80	81,01	16,06	0,45	7,88	7,37	14,38
10 h	4,98	27,78	75,13	16,22	0,42	7,90	4,10	11,86
11 h	3,07	23,50	69,47	16,37	3,42	13,21	2,95	5,16
12 h	6,92	26,66	74,63	13,38	1,51	12,88	3,13	1,28
13 h	12,92	19,93	97,01	10,62	2,30	14,74	5,08	7,81
14 h	9,60	22,20	95,73	17,63	1,46	16,67	6,16	1,33
15 h	12,79	11,95	87,16	20,00	0,78	16,04	1,05	4,28
16 h	4,26	28,60	92,38	2,22	0,41	13,35	3,75	2,54
17 h	8,70	25,88	78,40	3,23	2,01	17,55	0,08	7,35
18 h	10,58	25,39	82,38	5,76	2,78	20,11	3,47	5,23
19 h	11,87	26,13	78,30	2,26	3,00	23,09	0,83	8,00
20 h	11,35	26,61	46,24	6,13	2,24	21,66	9,30	2,63
21 h	10,64	21,25	49,35	7,91	1,74	23,55	2,33	4,78
22 h	1,56	14,55	23,97	15,29	0,45	15,06	1,83	8,13
23 h	7,96	2,54	4,23	21,56	0,14	0,98	12,53	25,20
24 h	2,47	0,12	56,65	0,78	2,24	14,37	2,53	5,33
Moyenne	7,3	16,7	67,8	12,2	1,4	13,8	4,7	10,6

Tableau 5-3: MAE de modèle ARX pour toutes les lignes.

Erreur moyenne absolue de modèle ARX en (%)								
Temps	Ligne 1	Ligne 2	Ligne 3	Ligne 4	Ligne 5	Ligne 6	Ligne 7	Ligne 8
1 h	21,2	5,59	41,64	8,83	7,45	18,91	0,57	16,68
2 h	5,7	4,66	43,22	0,58	8,18	2,02	5,81	11,75
3 h	6,8	6,98	32,58	11,32	8,30	1,13	17,07	17,67
4 h	30,5	1,46	24,85	23,20	10,29	4,96	21,37	16,83
5 h	28,2	27,12	48,14	9,17	7,79	12,44	19,34	10,34
6 h	26,6	2,88	16,36	23,88	6,69	2,64	30,33	5,66
7 h	25,7	5,56	35,02	23,26	6,21	7,72	23,19	0,24
8 h	23,6	7,88	13,60	13,54	3,98	7,04	30,34	8,10
9 h	19,7	4,54	34,53	26,60	2,61	4,88	30,28	4,01
10 h	12,3	22,97	24,94	13,27	1,20	9,20	21,41	16,85
11 h	4,7	28,64	29,83	9,92	5,92	13,52	12,28	25,41
12 h	5,1	21,95	9,13	3,92	5,73	5,53	3,39	43,78
13 h	21,4	23,90	36,39	7,77	6,59	13,94	0,25	46,47
14 h	8,3	24,90	4,98	11,94	4,47	17,99	1,47	37,08
15 h	12,4	12,41	10,83	7,39	3,71	15,73	1,01	40,55
16 h	10,8	20,87	6,63	10,35	3,34	5,10	5,30	40,29
17 h	18,4	30,43	25,64	13,47	4,31	13,08	6,42	48,40
18 h	26,8	29,47	32,27	19,57	5,96	16,13	7,27	46,23
19 h	26,2	35,47	4,39	16,83	5,33	22,55	13,52	33,98
20 h	37,5	46,42	34,97	20,17	2,81	27,14	21,06	45,90
21 h	47,0	31,12	14,49	27,01	0,78	34,89	22,17	60,76
22 h	32,0	38,10	46,92	21,67	1,99	24,10	24,72	69,18
23 h	32,1	24,10	12,33	25,66	3,54	17,42	9,32	49,18
24 h	44,8	28,88	58,64	29,02	1,17	32,75	25,96	74,56
Moyenne	22,0	20,3	26,8	15,8	4,9	13,8	14,7	32,1

Tableau 5-4: MAE de modèle Box-Jenkins pour toutes les lignes.

Erreur moyenne absolue de modèle Box-Jenkins en (%)								
Temps	Ligne 1	Ligne 2	Ligne 3	Ligne 4	Ligne 5	Ligne 6	Ligne 7	Ligne 8
1 h	14,3	47,16	54,79	47,82	13,23	7,59	22,90	45,05
2 h	69,0	27,11	54,74	38,34	1,92	18,19	36,09	43,08
3 h	24,1	34,70	224,78	38,70	6,55	2,86	40,23	93,53
4 h	63,2	10,06	52,84	69,04	37,39	15,71	39,16	62,43
5 h	43,1	24,92	69,88	38,93	6,38	24,50	98,60	28,97
6 h	51,2	7,14	55,46	10,51	11,22	14,37	28,35	21,49
7 h	53,8	90,50	67,58	30,26	8,87	5,44	76,86	31,12
8 h	56,2	4,69	70,18	21,75	2,87	65,91	87,82	37,83
9 h	34,5	29,01	80,41	79,92	2,44	2,14	606,61	71,36
10 h	26,3	7,92	60,04	25,07	1,63	76,90	34,08	32,42
11 h	14,0	13,99	52,84	0,96	7,04	29,06	10,39	1,98
12 h	22,9	15,30	55,88	16,47	4,14	0,28	67,76	27,50
13 h	2,2	14,65	68,63	5,19	11,37	5,01	9,92	6,23
14 h	1,0	22,83	64,42	42,61	12,48	10,60	21,16	5,96
15 h	0,5	5,97	62,52	46,79	2,75	12,84	13,43	17,29
16 h	9,6	0,34	1759,06	12,56	3,36	1,02	17,19	0,77
17 h	8,9	41,97	51,67	9,68	0,06	14,80	1,35	24,35
18 h	12,9	1,82	45,63	36,07	11,04	5,70	6,87	73,35
19 h	6,0	10,88	24,78	0,08	7,36	6,11	3,56	30,71
20 h	15,8	26,94	10,29	14,58	5,17	24,21	1,89	19,06
21 h	10,2	7,40	20,90	10,74	0,45	12,00	6,19	24,88
22 h	8,9	24,47	4,57	11,78	1,58	0,99	23,36	33,74
23 h	0,1	43,84	3,96	28,31	7,94	21,46	44,62	14,02
24 h	17,0	16,97	3,11	28,44	1,52	15,67	10,79	100,83
Moyenne	23,6	22,1	125,8	27,7	7,0	16,4	54,5	35,3

5.4 Validation expérimentale

Cette recherche ne se limite pas à une comparaison entre le signal mesuré et le signal attendu pour le modèle à étudier. Elle va au-delà jusqu'à atteindre la procédure de validation expérimentale d'un modèle avec la future température.

La figure 5-7 présente une illustration du processus de programmation en Python pour la validation expérimentale, qui repose essentiellement sur trois étapes : l'extraction de données futures de la température extérieure (selon la prévision météorologique), l'implémentation du modèle de prévision et la proposition d'une interface graphique.

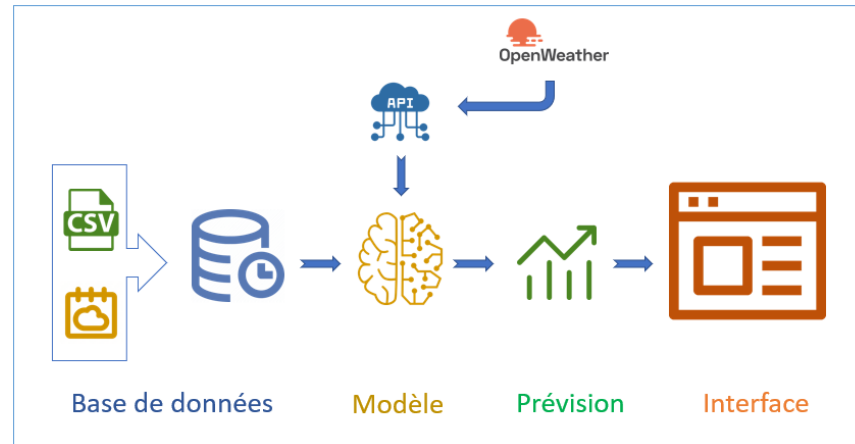


Figure 5-7: Plan de programmation pour la validation expérimentale.

5.4.1 Implémentation de modèle

Nous avons utilisé certaines bibliothèques pour l'implémentation expérimentale des méthodes de prévision. Ces bibliothèques incluent Numpy, Matplotlib, Pandas, Datetime et Statsmodels.tsa.arima_model [67]. La bibliothèque Statsmodel est spécifique au modèle de série temporelles ARMA (Autoregressive–moving-average) et sa famille, mais ce qui nous intéresse c'est le modèle ARMAX [68] et le modèle Box-Jenkins. Pour que les deux modèles soient bien entraînés, il suffit de taper les deux codes de programmation suivants.

Exemple et signification de codage pour le modèle ARMAX [69].

$$\text{Model ARMAX} = \text{ARMA}(\text{Output}, \text{exog} = \text{Input}, \text{order} = (p, q))$$

- Model ARMAX : La variable stockant les caractéristiques du modèle que nous ajusterons plus tard.
- Output : La série temporelle que nous souhaitons analyser.
- Exog : La variable exogène que nous ajoutons au modèle ARMA.

- Ordre (p,q) : L'ordre du modèle ARMAX ('il faut changer l'ordre du modèle à (p,q)').

Le code pour le modèle Box Jenkins est écrit comme suit :

Model Box Jenkins = ARIMA(Output, exog = Input, order = (p, d, q))

- Model Box-Jenkins : La variable stockant les caractéristiques du modèle que nous ajusterons plus tard.
- Output : La série temporelle que nous souhaitons étudier.
- Exog : La variable exogène que nous ajoutons au modèle ARIMA.
- Ordre (p,d,q) : L'ordre du modèle BJ ('Il faut changer l'ordre (p,d,q) jusqu'à ce que le modèle donne des bons résultats').

La bibliothèque arch.univariate [70] fournit un ensemble de modèles de séries chronologiques tels que GARCH, ZeroMean, arch_model et ARX. Dans cette librairie, nous utiliserons le modèle ARX et effectuerons la programmation pour l'entraînement du modèle comme suit :

Model ARX = ARX(Y, X, lags = [a, b, c])

- Model ARX : La variable stockant les caractéristiques du modèle que nous ajusterons plus tard.
- Y : La série temporelle que nous souhaitons étudier, cas de puissance en (MW).
- X : La variable exogène que nous ajoutons au modèle ARX (la température extérieure).

- Lags : Description de la structure des décalages du HAR. Scalaire inclus tous les décalages entre 1 et la valeur. Un tableau 1-d inclut les décalages AR lags[0], lags[1], . . .

Pour la validation expérimentale, nous avons utilisé la même méthodologie de programmation MATLAB pour Python. Autrement dit, en Python nous entraînons les modèles de prévision ARMAX, ARX et Box-Jenkins avec un mois entier de données historiques caractérisées par une forte corrélation entre la consommation électrique en (MW) et la température extérieure (°C) pour toutes les lignes de distribution afin que nous puissions obtenir une prévision pour les prochaines 24 heures.

Les données de janvier 2020 ont été retenues, car elles correspondent à un critère de corrélation forte entre la consommation électrique et la température extérieure pour toutes les lignes de distribution à l'exception lignes de distribution dites industrielles. Sachant que les méthodes choisies mathématiquement établissent une autocorrélation des données entrée la puissance et la température.

La Figure 5-8 montre les résultats obtenus en utilisant les données de température extérieure au 1er février 2020 comme entrée du modèle déjà entraîné en janvier afin d'obtenir la prévision de consommation au 1er février 2020. Nous utilisons dans ce test les trois modèles afin de comparer les résultats avec le signal.

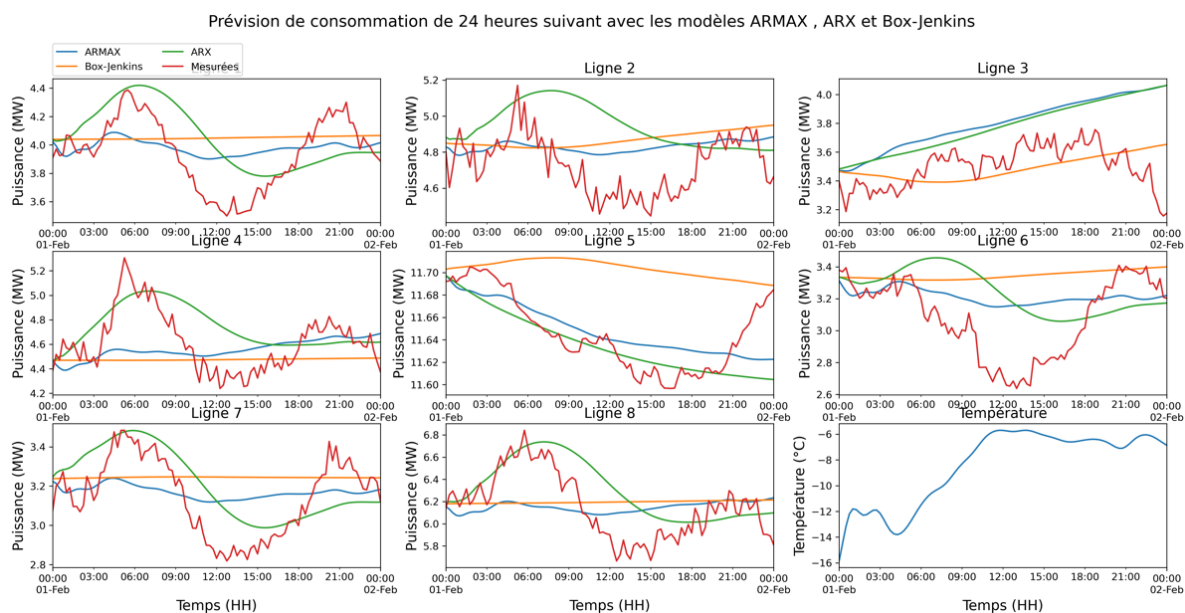


Figure 5-8: Prévision de la consommation pour le premier février 2020

À travers le tableau ci-dessous et la Figure 5-8, il est clair que les trois modèles diffèrent en termes de qualité de prévision pour chaque ligne de distribution. Par exemple, la prévision du modèle ARX suit le signal mesuré en termes de détection du pic de consommation, surtout au début du signal dans toutes les lignes de distribution à l'exception des lignes de distribution industrielles. Dans ce cas, les modèles ARMAX et Box-Jenkins donnent un signal semi-continu de sorte qu'il n'est pas possible de détecter le pic de puissance.

Tableau 5-5 : MAE pour les trois méthodes en Python.

Erreur Absolue Moyenne en (MW)								
	Ligne 1	Ligne 2	Ligne 3	Ligne 4	Ligne 5	Ligne 6	Ligne 7	Ligne 8
ARMAX	0,17	0,14	0,30	0,18	0,01	0,19	0,14	0,23
ARX	0,20	0,24	0,27	0,18	0,01	0,24	0,13	0,19
Box-Jenkins	0,21	0,17	0,13	0,20	0,05	0,27	0,17	0,24

5.4.2 Interface de programmation d'application (API)

Pour l'API, nous l'utiliserons pour extraire les futures données de température qui seront ensuite transmises aux modèles ARMAX, ARX et Box-Jenkins déjà entraînés. Pour ce projet, nous avons utilisé une API du site Web Openweathermap (<https://openweathermap.org/>) qui fournit gratuitement des données de température future sur 48 heures pour chaque appel ou exécution en Python. Si nous avons besoin d'une mise à jour périodique des données futures de la température extérieure, il existe une offre payante. En termes la prévision de 24 heures de consommation électrique, nous n'avons besoin que de 24 heures de température extérieure de l'API (gratuit). La Figure 5-9 montre un schéma simplifié de l'utilisation d'API de température d'extérieur.

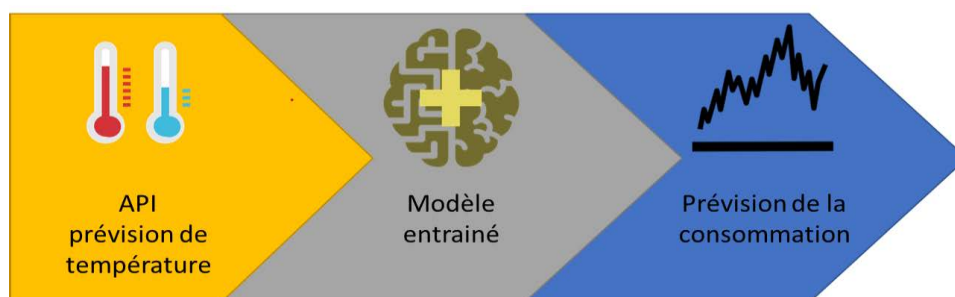


Figure 5-9: Utilisation d'API pour la prévision de la consommation.

Pratiquement pour avoir une prévision de 24 heures de consommation électrique nous avons besoin de 96 échantillons c'est-à-dire 96 prélèvements de 15 minutes de la température sachons que l'API offre 24 échantillons pour 24 heures. Alors nous avons extrait une prévision de 24 heures de la température et extrapolé pour avoir 96 échantillons. Cette opération est faite grâce à deux bibliothèques « Requests » pour importer l'API et « Scipy.interpolate » pour l'extrapolation des données.

Une expérience a été faite pour évaluer l'efficacité des prévisions de la consommation en utilisant l'API de la température future comme entrée d'un modèle pré-entraîné avec des données historiques de température et de puissance en mégawatts pour toutes les lignes de distribution, en particulier les données du mois de janvier 2020. Dans cette expérience, nous avons obtenu la prévision à partir du moment de l'expérience jusqu'aux prochaines 24 heures. La figure 5-10 présente les résultats obtenus du 02/03/2022 15h00 au 03/03/2022 15h00.

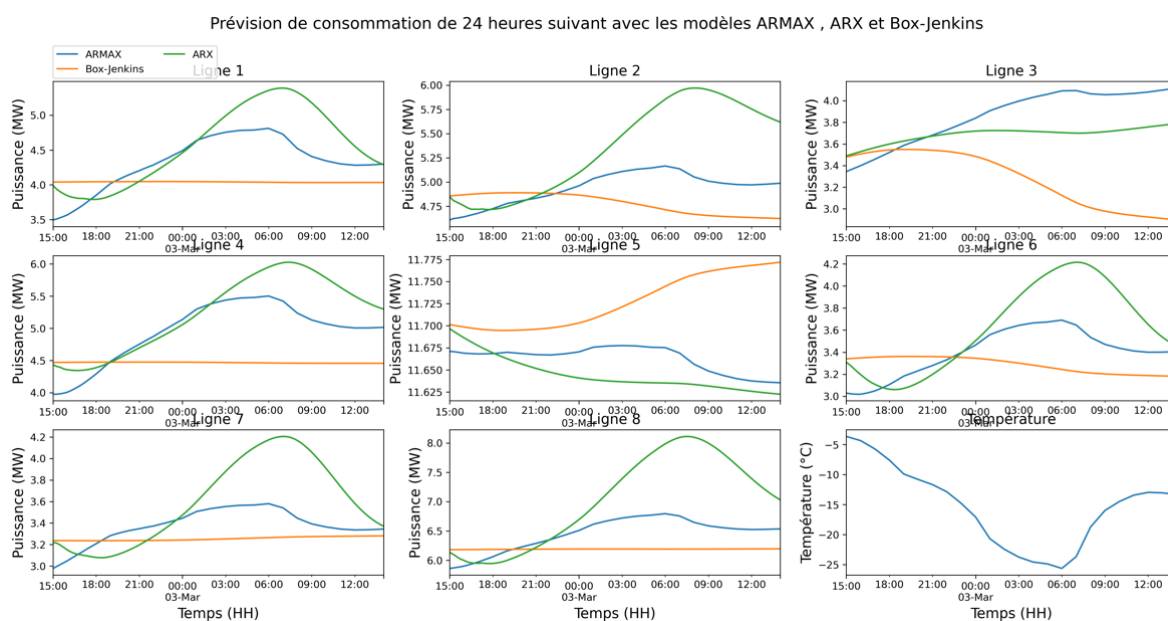


Figure 5-10 : Prévision de consommation avec API

La prévision obtenue en utilisant l'API pour la température future dans toutes les lignes de distribution démontre concrètement que les trois modèles donnent des résultats approximatifs similaires aux résultats du test effectué entre le signal mesuré et le signal prédit au 1er février 2020. D'après les résultats de l'expérience, on constate que le modèle ARX interagit bien avec la température future puis le modèle ARMAX interagit à son tour avec la température future. Mais par rapport au modèle Box-Jenkins, il n'est pas affecté par la température, car ici la prévision de la consommation reste constante.

5.4.3 Interface graphique

Nous avons implémenté une interface d'utilisateur graphique (GUI) pour présenter le programme à l'utilisateur sur l'écran d'un ordinateur ou d'un micro-ordinateur tel qu'un Raspberry Pi. Python fournit la bibliothèque « Tkinter » pour une programmation facile de l'interface.

L'utilisation de l'interface graphique open source [71] nous a permis d'implémenter simplement cette interface qui possède plusieurs boutons qui me permettent de visualiser chaque ligne de distribution individuellement. Un bouton permet aussi de voir toutes les lignes pour connaître l'électricité que la ville consomme et faire une bonne gestion des pics de consommation surtout en période hivernale.

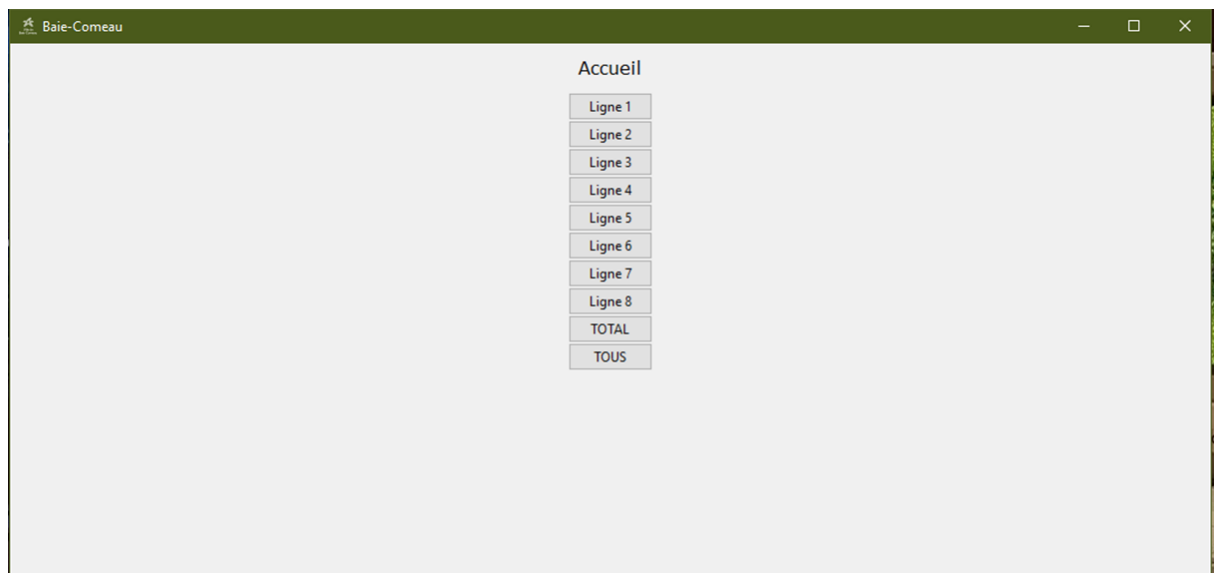


Figure 5-11: Interface graphique.

Pour surveiller les résultats obtenus à partir des prévisions de consommation d'électricité de la ville de Baie-Comeau pour toutes les lignes de distribution et la consommation totale pour toutes les lignes, en appuyant sur un bouton et nous pouvons obtenir la ligne à étudier.

Quelques résultats de la prévision de la consommation de toutes les lignes de distribution obtenue avec l'interface utilisateur graphique (GUI) sont présentés en annexe.

5.5 Synthèse du chapitre

L'objectif de ce cinquième chapitre était de proposer une stratégie pour la prévision de la puissance totale et de la puissance pour chaque ligne de distribution d'électricité séparément à partir d'une base de données réelle de la ville de Baie-Comeau. Cette information de prévision facilitera au centre de décision de la ville l'optimisation du réseau de distribution.

Cette stratégie nous a conduit à proposer des modèles de séries temporelles pour prévoir la charge à très court terme de l'ensemble des lignes de distribution d'électricité afin de voir quelle est l'efficacité du modèle pour ses différents types de consommateurs. Dans l'ensemble, les trois modèles de séries temporelles ARMAX, ARX et Box-Jenkins utilisés donnent des résultats prometteurs, car l'erreur absolue moyenne est acceptable pour ce cas de projet. Les trois modèles interagissent bien avec l'entrée exogène c'est-à-dire la température extérieure, mais avec quelques exceptions telles que la relation du modèle choisi lui-même et le type de consommateur pour chaque ligne.

Le chapitre s'est conclu par la création d'une interface utilisateur graphique pour faciliter le fonctionnement du programme afin de faire la prévision de la consommation électrique pour les prochaines 24 heures à partir du moment où le programme est activé grâce aux méthodes proposées et à l'utilisation l'API de la température future comme une entrée dans le modèle.

Chapitre 6 - Conclusion et recommandations

6.1 Conclusion générale

Nous sommes arrivés à la fin de la recherche scientifique liée à l'amélioration du fonctionnement d'un réseau de distribution électrique de la ville de Baie-Comeau ainsi que notre présentation du problème de la forte consommation d'électricité en hiver où la température atteint moins 30 degrés, qui est un problème digne d'être étudié. Malgré le fait que plusieurs chercheurs l'ont déjà abordé, nous considérons que s'il est abordé de manière systématique des contributions importantes peuvent être encore faites pour améliorer la gestion des réseaux électriques.

Nous avons essayé, autant que possible, d'analyser et de détailler dans quelle mesure le climat affecte le comportement des individus et des institutions en matière de consommation d'électricité sur des bases scientifiques. Tout d'abord, une description du réseau a été réalisée afin de déterminer le type de consommateurs et la quantité d'énergie consommée dans l'ensemble des lignes de distribution. Ceci nous amène à l'étude du facteur de corrélation entre la consommation et la météo (vitesse du vent, rayonnement solaire, température sèche et température humide) et à l'analyse du pic de consommation en fonction de la température. L'importance de cette étude est de déterminer un point de départ pour adopter des stratégies afin d'éviter les pics de consommation en prévoyant le futur pic. Ce qui se reflète positivement dans l'amélioration du réseau de distribution pour prévoir la consommation en

utilisant l'intelligence artificielle. Le choix s'est porté sur trois modèles de séries temporelles, qui sont ARMAX, ARX et Box-Jenkins. La prévision de la consommation future donne des résultats différents selon chaque modèle et chaque ligne de distribution. L'étude est complétée par la création d'une interface graphique pour faciliter la prise en main du programme de prévision.

Il y a de nombreux aspects que les chercheurs peuvent détailler concernant l'optimisation électrique via l'intelligence artificielle dans leur recherche scientifique, et nous n'avons traité qu'une petite partie du problème.

6.2 Recommandation

Dans le processus d'aborder le sujet de l'optimisation de l'opération du système électrique du réseau de distribution de la ville de Baie-Comeau, nous faisons les recommandations suivantes :

- Nous proposons de tester la prévision de la consommation d'électricité en utilisant des techniques basées sur l'apprentissage profond (Deep Learning), pour comparer les résultats à ceux obtenus dans cette étude.
- Nous suggérons une étude de prévision en manipulant la quantité de données comme une procédure de l'entraînement de modèle avec différentes quantités de données.
- Nous proposons d'améliorer les prévisions de courbe de consommation en intégrant la notion de « journée similaire » et « d'historique de consommation » dans l'étude prévisionnelle.

- Une étude sur les stratégies de gestion pour minimiser de la consommation électrique et réduire le facteur d'utilisation un exemple de stratégie qui permet de reconfigurer le réseau électrique et laisse la place à la gestion des urgences sur le réseau.

RÉFÉRENCES

- [1] J. Caron, “Attestation du trésorier ou du secrétaire-trésorier sur le rapport financier consolidé,” 2017. https://www.ville.baie-comeau.qc.ca/fileadmin/Documents/Onglet_ville/Finances_et_budget/Etats_financiers_VBC_Consolides_31_dec._2017_FINAL_signes.pdf (accessed Mar. 03, 2021).
- [2] J. Caron, “Attestation du trésorier ou du secrétaire-trésorier sur le rapport financier consolidé,” 2018. https://www.ville.baie-comeau.qc.ca/fileadmin/Documents/Onglet_ville/Finances_et_budget/2019/Etats_financiers_VBC_consolides_2018_signes.pdf (accessed Mar. 03, 2021).
- [3] J. Caron, “Attestation du trésorier ou du secrétaire-trésorier sur le rapport financier consolidé,” 2014. https://www.ville.baie-comeau.qc.ca/fileadmin/Documents/Onglet_ville/Finances_et_budget/2015/Etats_financiers_VBC_consolides_2014_FINAL_DEPOSE.pdf (accessed Mar. 03, 2021).
- [4] J. Caron, “Attestation du trésorier ou du secrétaire-trésorier sur le rapport financier consolidé,” 2019. https://www.ville.baie-comeau.qc.ca/fileadmin/Documents/Onglet_ville/Finances_et_budget/2020/Etats_financiers_2019_VBC_Consolide_FINAL_signe.pdf (accessed Mar. 03, 2021).
- [5] B. M. locale Deforge, “Attestation du trésorier ou du secrétaire-trésorier sur le rapport financier consolidé,” *Bulletin de l’Association Guillaume Budé*, 2015. https://www.ville.baie-comeau.qc.ca/fileadmin/Documents/Onglet_ville/Finances_et_budget/2016/Rapport_financier_final_2015_Depose_au_MAMOT_Juin_2016.pdf (accessed Mar. 03, 2021).
- [6] J. Caron, “Attestation du trésorier ou du secrétaire-trésorier sur le rapport financier consolidé,” 2016. https://www.ville.baie-comeau.qc.ca/fileadmin/Documents/Onglet_ville/Finances_et_budget/2017_1/Etats_financiers_VBC_Consolides_31_dec._2016.pdf (accessed Mar. 03, 2021).
- [7] J. Caron, “Attestation du trésorier ou du secrétaire-trésorier sur le rapport financier consolidé,” 2013. https://www.ville.baie-comeau.qc.ca/fileadmin/Documents/Onglet_ville/Finances_et_budget/Etats_-_rapport_financiers_consolides_2013.pdf (accessed Mar. 03, 2021).
- [8] J. Caron, “Attestation du trésorier ou du secrétaire-trésorier sur le rapport financier consolidé,” 2020. https://www.ville.baie-comeau.qc.ca/fileadmin/Documents/Onglet_ville/Finances_et_budget/2020/Etats_financiers_2020_VBC_Consolide_FINAL_signe.pdf (accessed Mar. 03, 2021).

- comeau.qc.ca/fileadmin/Documents/Onglet_ville/Finances_et_budget/2021/Etats_financiers_2020_Consolide_-_FINAL_signee.pdf (accessed Mar. 03, 2021).
- [9] Wikipidiea, “Baie-Comeau,” *Wikipedia*, 2021. <https://fr.wikipedia.org/wiki/Baie-Comeau> (accessed Mar. 12, 2021).
- [10] V. De Baie-comeau, “Site officiel de la ville,” 2021. <https://www.ville.baie-comeau.qc.ca/citoyen/electricite-baie-comeau/> (accessed Mar. 10, 2021).
- [11] AREQ, “Association québécoise de redistribution de l’électricité,” 2021. <http://www.areq.org/> (accessed Jun. 07, 2021).
- [12] J. Whitmore and P.-O. Pineau, “L’Énergie Au Québec,” 2018. http://www.canadiangeographic.com/educational_products/activities/energy_production_floormap/Energy_IQ_Quebec_FR.pdf (accessed Mar. 23, 2021).
- [13] Hydro-Quebec, “Production,” 2021. <https://www.hydroquebec.com/production/> (accessed May 15, 2021).
- [14] La Régie de l’énergie du Canada, “Profils énergétiques des provinces et territoires – Québec,” 2021. <https://www.cer-rec.gc.ca/fr/donnees-analyse/marches-energetiques/profils-energetiques-provinces-territoires/profils-energetiques-provinces-territoires-quebec.html> (accessed Jun. 21, 2021).
- [15] S. Énergétiques, “Consommation d’électricité.” <https://mern.gouv.qc.ca/energie/statistiques-energetiques/consommation-electricite/> (accessed Jun. 11, 2021).
- [16] Association des Véhicules Électriques du Québec, “Comprendre la hausse des camions légers au canada afin de renverser la tendance,” 2016. <https://www.aveq.ca/actualiteacutes/category/gaz-a-effet-de-serre-ges> (accessed Apr. 25, 2021).
- [17] HEC montréal, “État de l’énergie au québec,” 2019. http://energie.hec.ca/wp-content/uploads/2018/12/EEQ2019_WEB.pdf (accessed Aug. 19, 2021).
- [18] Ressource naturel canada, “Évolution de l’efficacité énergétique au Canada,” 2013. <https://www.nrcan.gc.ca/sites/www.nrcan.gc.ca/files/energy/pdf/evolution2013.pdf> (accessed Mar. 03, 2021).
- [19] Conseil régional de l’environnement de la Côte-Nord, “Portrait énergétique régional Côte-Nord,” p. 165 p., 2013.
- [20] Wikipidiea, “Intelligence artificielle,” 2021. https://fr.wikipedia.org/wiki/Intelligence_artificielle (accessed Jul. 02, 2021).
- [21] Wikipidiea, “Apprentissage automatique,” 2021. https://fr.wikipedia.org/wiki/Apprentissage_automatique (accessed Jul. 03, 2021).

- [22] I. Crucianu, O. Bularca, and A. M. Dumitrescu, "Modelling and forecasting of electrical consumption for demand response applications," *2019 IEEE Milan PowerTech, PowerTech 2019*, 2019, doi: 10.1109/PTC.2019.8810726.
- [23] J. Runge and R. Zmeureanu, "A review of deep learning techniques for forecasting energy use in buildings," *Energies*, vol. 14, no. 3, 2021, doi: 10.3390/en14030608.
- [24] S. Touré, "Optimisation des réseaux : réseau actif et flexible," *Univ. Grenoble, Grenoble*, no. Energie électrique, 2015.
- [25] H. Salazar, R. Gallego, and R. Romero, "Artificial neural networks and clustering techniques applied in the reconfiguration of distribution systems," *IEEE Trans. Power Deliv.*, vol. 21, no. 3, pp. 1735–1742, 2006, doi: 10.1109/TPWRD.2006.875854.
- [26] C. M. Huang, C. J. Huang, and M. L. Wang, "A particle swarm optimization to identifying the ARMAX model for short-term load forecasting," *IEEE Trans. Power Syst.*, vol. 20, no. 2, pp. 1126–1133, 2005, doi: 10.1109/TPWRS.2005.846106.
- [27] J. Walther, D. Spanier, N. Panten, and E. Abele, "Very short-term load forecasting on factory level - A machine learning approach," *Procedia CIRP*, vol. 80, pp. 705–710, 2019, doi: 10.1016/j.procir.2019.01.060.
- [28] T. Hong and S. Fan, "Probabilistic electric load forecasting: A tutorial review," *Int. J. Forecast.*, vol. 32, no. 3, pp. 914–938, 2016, doi: 10.1016/j.ijforecast.2015.11.011.
- [29] S. N. Fallah, M. Ganjkhani, S. Shamshirband, and K. wing Chau, "Computational intelligence on short-term load forecasting: A methodological overview," *Energies*, vol. 12, no. 3, 2019, doi: 10.3390/en12030393.
- [30] N. Abu-Shikhah, F. Elkarmi, and O. M. Aloquili, "Medium-Term Electric Load Forecasting Using Multivariable Linear and Non-Linear Regression," *Smart Grid Renew. Energy*, vol. 02, no. 02, pp. 126–135, 2011, doi: 10.4236/sgre.2011.22015.
- [31] D. L. Alvarez *et al.*, "Optimal decision making in electrical systems using an asset risk management framework," *Energies*, vol. 14, no. 16, pp. 1–25, 2021, doi: 10.3390/en14164987.
- [32] M. Nosrati Maraloo, A. R. Koushki, C. Lucas, and A. Kalhor, "Long term electrical load forecasting via a neurofuzzy model," *2009 14th Int. CSI Comput. Conf. CSICC 2009*, no. November, pp. 35–40, 2009, doi: 10.1109/CSICC.2009.5349440.
- [33] D. S. Hermiyanty, Wandira Ayu Bertin, "Modélisation Et Prévision De La Demande D'Électricité Résidentielle," *J. Chem. Inf. Model.*, vol. 8, no. 9, pp. 1–58, 2017.
- [34] L. Babich, D. Svalov, A. Smirnov, and M. Babich, "Industrial Power Consumption Forecasting Methods Comparison," *Proc. - 2019 Ural Symp. Biomed. Eng. Radioelectron. Inf. Technol. USBEREIT 2019*, no. 02, pp. 307–309, 2019, doi: 10.1109/USBEREIT.2019.8736640.

- [35] Keras, “Python bibliothèque keras,” 2022. <https://keras.io/> (accessed Nov. 13, 2022).
- [36] Sklearn, “Python bibliothèque scikit-learn,” 2022. <https://scikit-learn.org/stable/index.html#> (accessed Nov. 13, 2022).
- [37] statsmodels, “Python bibliothèque statsmodels,” 2022. <https://www.statsmodels.org/stable/index.html> (accessed Nov. 13, 2022).
- [38] T. Haring, R. Ahmadiyahangar, A. Rosin, T. Korotko, and H. Biechl, “Accuracy Analysis of Selected Time Series and Machine Learning Methods for Smart Cities based on Estonian Electricity Consumption Forecast,” *Proc. - 2020 IEEE 14th Int. Conf. Compat. Power Electron. Power Eng. CPE-POWERENG 2020*, pp. 425–428, 2020, doi: 10.1109/CPE-POWERENG48600.2020.9161690.
- [39] A. Sahebalam, S. Beheshti, W. Khreich, and E. W. Nidoy, “A novel approach in household electricity consumption forecasting,” *Can. Conf. Electr. Comput. Eng.*, vol. 2016-October, pp. 7–10, 2016, doi: 10.1109/CCECE.2016.7726768.
- [40] M. F. Bin Othman and M. K. B. M. Yusoff, “System identification to forecast electricity loads,” *SCORED2009 - Proc. 2009 IEEE Student Conf. Res. Dev.*, pp. 403–406, 2009, doi: 10.1109/SCORED.2009.5442986.
- [41] H. T. Pao, “Comparing linear and nonlinear forecasts for Taiwan’s electricity consumption,” *Energy*, vol. 31, no. 12, pp. 2129–2141, 2006, doi: 10.1016/j.energy.2005.08.010.
- [42] S. Jaiswal, A. Gupta, and S. K. Kanojiya, “Optimization of Energy Consumption via Artificial Intelligence: A Study,” *SAMRIDDHI A J. Phys. Sci. Eng. Technol.*, vol. 8, no. 01, pp. 26–33, 2016, doi: 10.18090/samriddhi.v8i1.11409.
- [43] F. Mountassir, “Machine Learning au service de la prédiction de la demande d’énergie dans les Smart Grid,” 2019.
- [44] BOUBACAR DOUCOURÉ, “Proposition, intégration dans un système de gestion de réseau intelligent et validation expérimentale d’une méthode de prédiction pour un système d’énergies renouvelables,” 2015.
- [45] V. KULISH and V. HORÁK, “Forecasting the Behavior of Fractal Time Series: Hurst Exponent As a Measure of Predictability,” *Rev. Air Force Acad.*, vol. 14, no. 2, pp. 61–68, 2016, doi: 10.19062/1842-9238.2016.14.2.8.
- [46] B. Qian and K. Rasheed, “Hurst exponent and financial market predictability,” *Proc. Second IASTED Int. Conf. Financ. Eng. Appl.*, pp. 203–209, 2004.
- [47] D. Delignières, “Séries temporelles – Modèles ARIMA.,” *Séminaire EA “Sport – Perform. – Santé,”* p. 19, 2000, [Online]. Available: <https://didierdelignieresblog.files.wordpress.com/2016/03/arimacomplet.pdf>.

- [48] M.-A. Rousseau, “Séries chronologique à une plusieurs variables: synthèse des méthodes classiques et modèles à base de copules,” 2011.
- [49] Nicolas Chopin, “Introduction aux processus stochastiques.” https://nchopin.github.io/files/poly_processus.pdf (accessed Sep. 04, 2021).
- [50] A. C. Chukunyere, “Les modèles VAR(p),” Quebec, 2019. [Online]. Available: <https://corpus.ulaval.ca/server/api/core/bitstreams/185caec7-863c-44d2-8a9f-ff61e43c51b1/content>.
- [51] A. Wicaksana, “Les modèles MA, AR et ARMA multidimensionnels : estimation et causalité.” vol. 4, pp. 112–136, 2016, [Online]. Available: <http://camus.math.usherbrooke.ca/revue.html>.
- [52] S. Rachad, B. Nsiri, and B. Bensassi, “System Identification of Inventory System Using ARX and ARMAX Models,” *Int. J. Control Autom.*, vol. 8, no. 12, pp. 283–294, 2015, doi: 10.14257/ijca.2015.8.12.26.
- [53] F. A. Ruslan, K. Haron, A. M. Samad, and R. Adnan, “Multiple Input Single Output (MISO) ARX and ARMAX model of flood prediction system: Case study Pahang,” *Proc. - 2017 IEEE 13th Int. Colloq. Signal Process. its Appl. CSPA 2017*, no. March, pp. 179–184, 2017, doi: 10.1109/CSPA.2017.8064947.
- [54] T. Pedersen, “Forecasting model of electricity demand in the Nordic countries,” 2014.
- [55] NCSS and LLC, “The Box-Jenkins method,” 1976. [Online]. Available: https://ncss-wpengine.netdna-ssl.com/wp-content/themes/ncss/pdf/Procedures/NCSS/The_Box-Jenkins_Method.pdf.
- [56] Office of Information Technology, “Box-Jenkins model notation,” 2022. [Online]. Available: http://faculty.smu.edu/tfomby/eco6375/BJ_Notes/BJ_notation.pdf.
- [57] MathWorks, “Estimate Box-Jenkins polynomial model using time domain data.” <https://www.mathworks.com/help/ident/ref/bj.html> (accessed May 29, 2021).
- [58] H. Garnier, “Apprentissage De Modèles Pour La Commande De La Mobilité Interne En Robotique_2010THDR1721.pdf.”
- [59] F. Jeannot and Q. Janvier, “Fonctions de coût et Machine Learning,” pp. 3–5, 2018.
- [60] Guillaume Saint-Cirgue, *Métriques de régressions en data science (Coefficient de Détermination, Erreur Quadratique, etc...)*. 2019.
- [61] F.khattabi, “Erreur expérimentale.” [Online]. Available: <http://web.dsfm.mb.ca/ecoles/clr/profs/fkhattabi/M1L2.pdf>.
- [62] Magloire Ndabagera, “Les 38 mesures de performance pour un modèle d’IA.” <https://magloire-ndabagera.com/les-38-mesures-de-performance-pour-un-modele-dia>

(accessed Aug. 13, 2021).

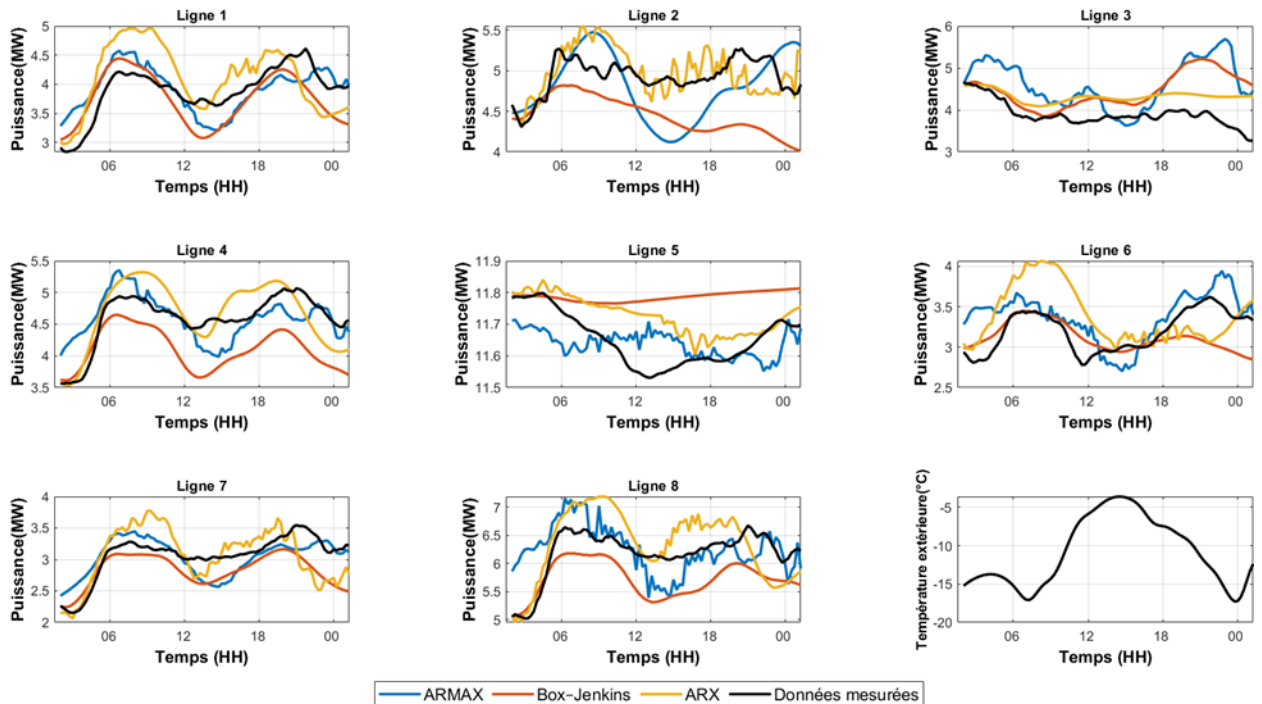
- [63] M. E. Hossain, “Application of Gaussian mixture regression model for short-term wind speed forecasting,” *2017 North Am. Power Symp. NAPS 2017*, 2017, doi: 10.1109/NAPS.2017.8107222.
- [64] EATON, “Logiciel CYME.” <https://www.eaton.com/ca/fr-ca/products/utility-grid-solutions/software-modules/distribution-system-analysis-package-CYMDIST.html> (accessed Aug. 18, 2021).
- [65] A. Diegel, “Corrélation,” *Sondage*, pp. 93–102, 2020, doi: 10.2307/j.ctv18ph5qw.15.
- [66] Wikipedia, “Facteur de crête,” 2021. https://fr.wikipedia.org/wiki/Facteur_de_crête (accessed Dec. 05, 2021).
- [67] W. McKinney, J. Perktold, and S. Seabold, “Time Series Analysis in Python with statsmodels,” *Proc. 10th Python Sci. Conf.*, no. Scipy, pp. 107–113, 2011, doi: 10.25080/majora-ebaa42b7-012.
- [68] T. S. McElroy and D. N. Politis, “Fitting Time Series Models,” *Time Ser.*, pp. 325–383, 2020, doi: 10.1201/9780429109553-10.
- [69] F. Name, M. M. Notation, T. Arima, and A. Arima, “The ARIMA Model Implementation of the Simple Model in Python :,” 2022.
- [70] K. Sheppard, “arch Documentation,” 2021.
- [71] Harrison Kinsley, “GUI,” *pythonprogramming*, 2022. <https://pythonprogramming.net/creating-main-menu-tkinter/?completed=/customizing-tkinter-matplotlib-graph/> (accessed Jan. 09, 2022).

Annexe A – Boucle de prévision de consommation d'électricité pour toutes les lignes de distribution.

Nous présentons ici, les résultats obtenus de la prévision de la consommation de la puissance électrique en (MW) pour toutes les lignes en MATLAB. Le but de cette expérience est de voir l'interaction des trois modèles dynamiquement, cela signifie la progression de la prévision pour chaque décalage horaire + 1 heure.

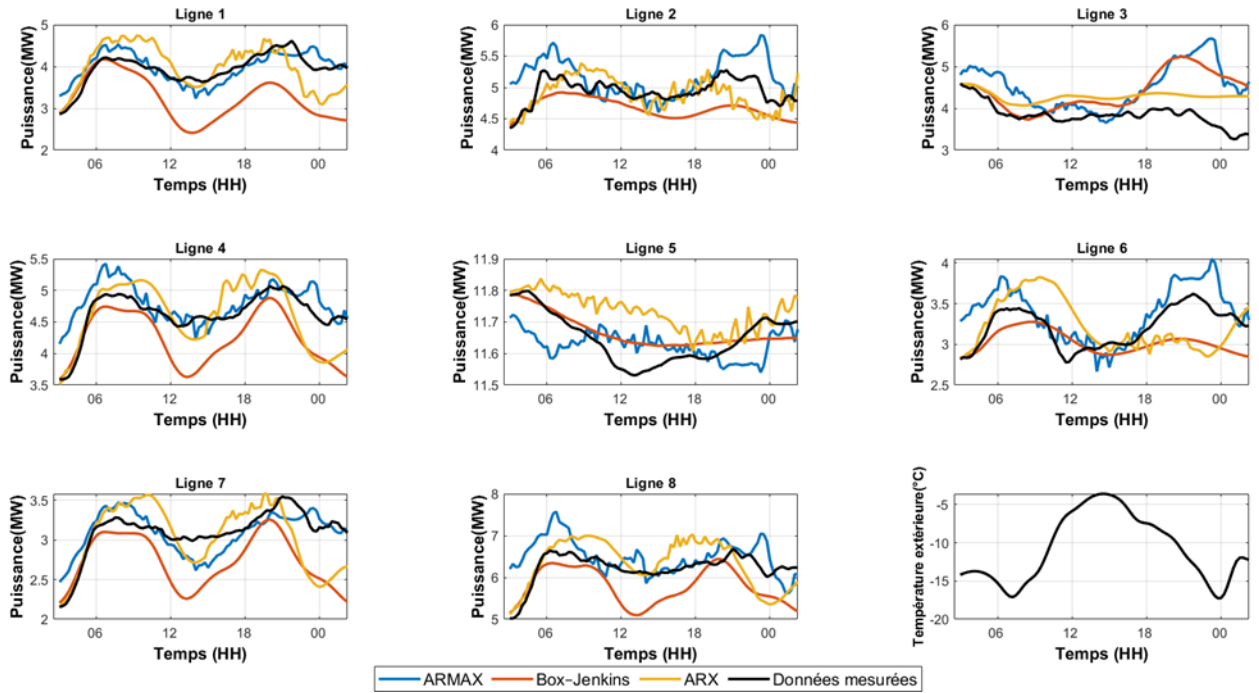
✓ Décalage 1 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 1 heure de décalage



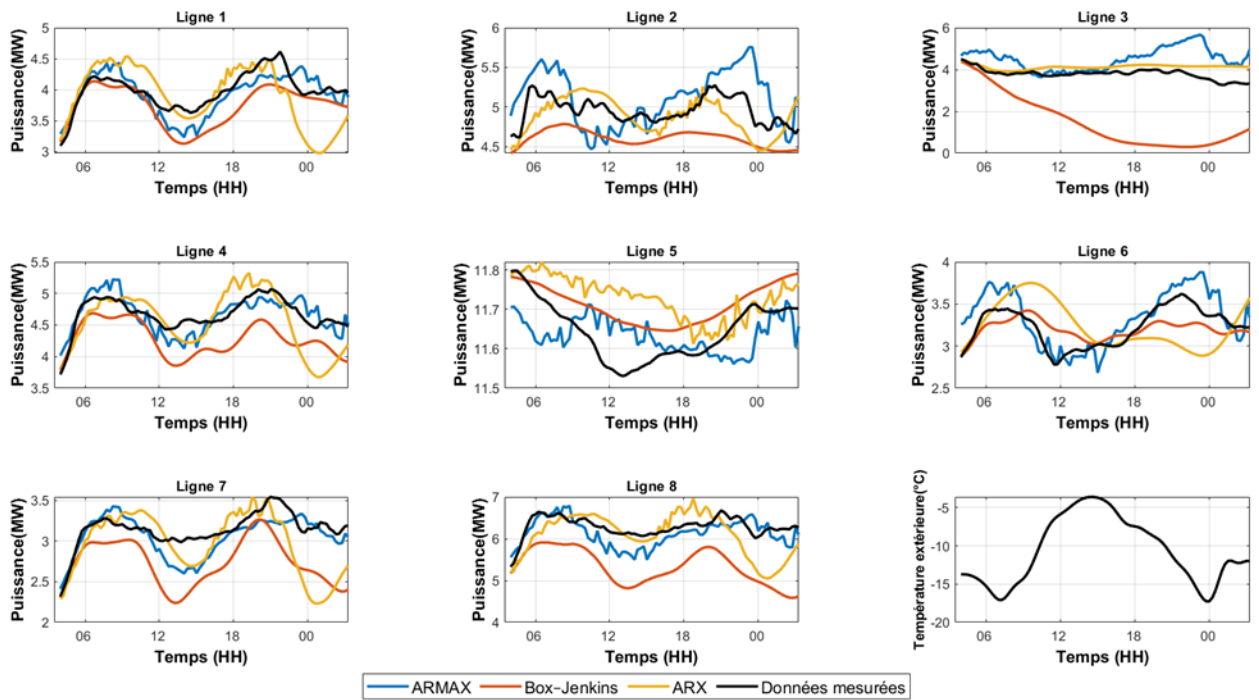
✓ Décalage 2 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 2 heure de décalage



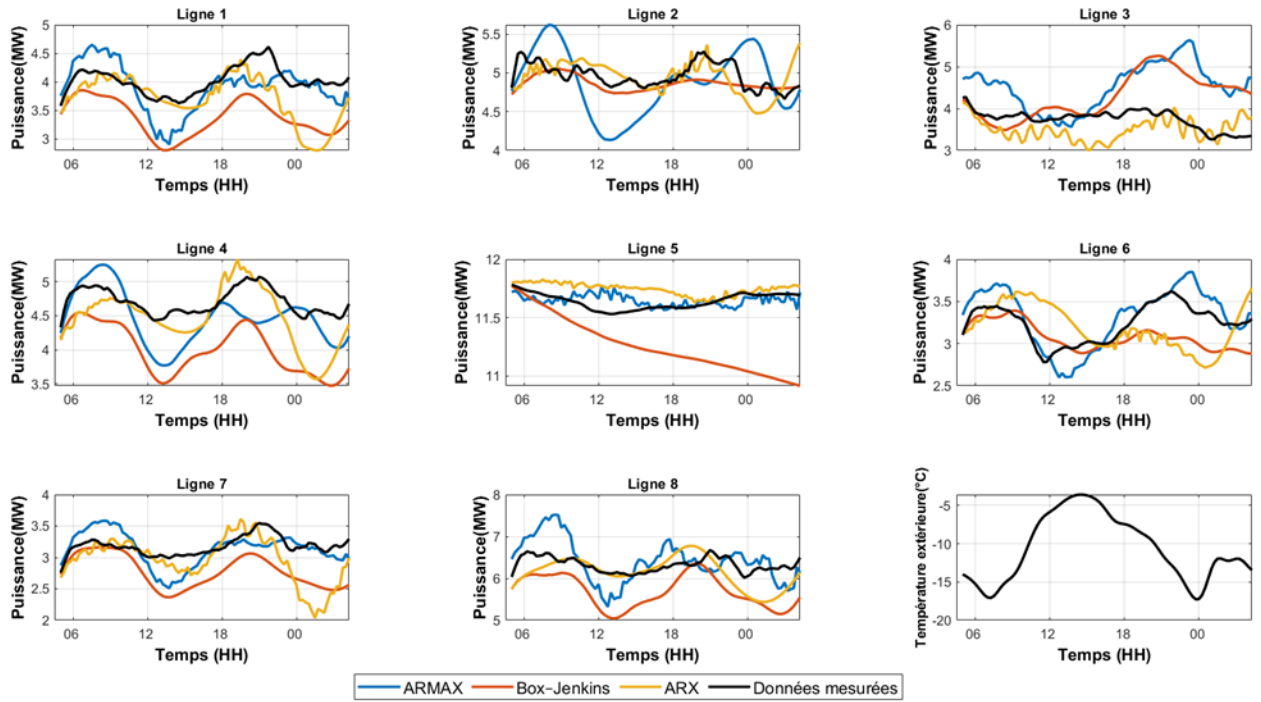
✓ Décalage 3 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 3 heure de décalage



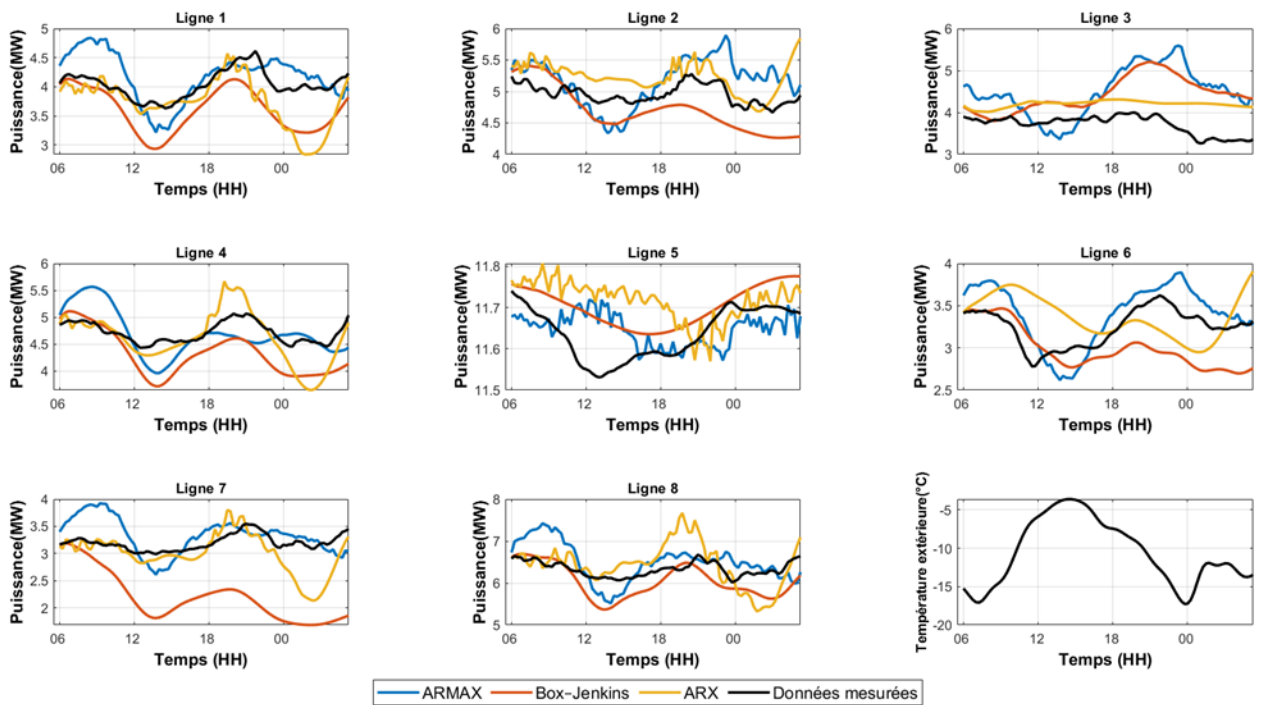
✓ Décalage 4 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 4 heure de décalage



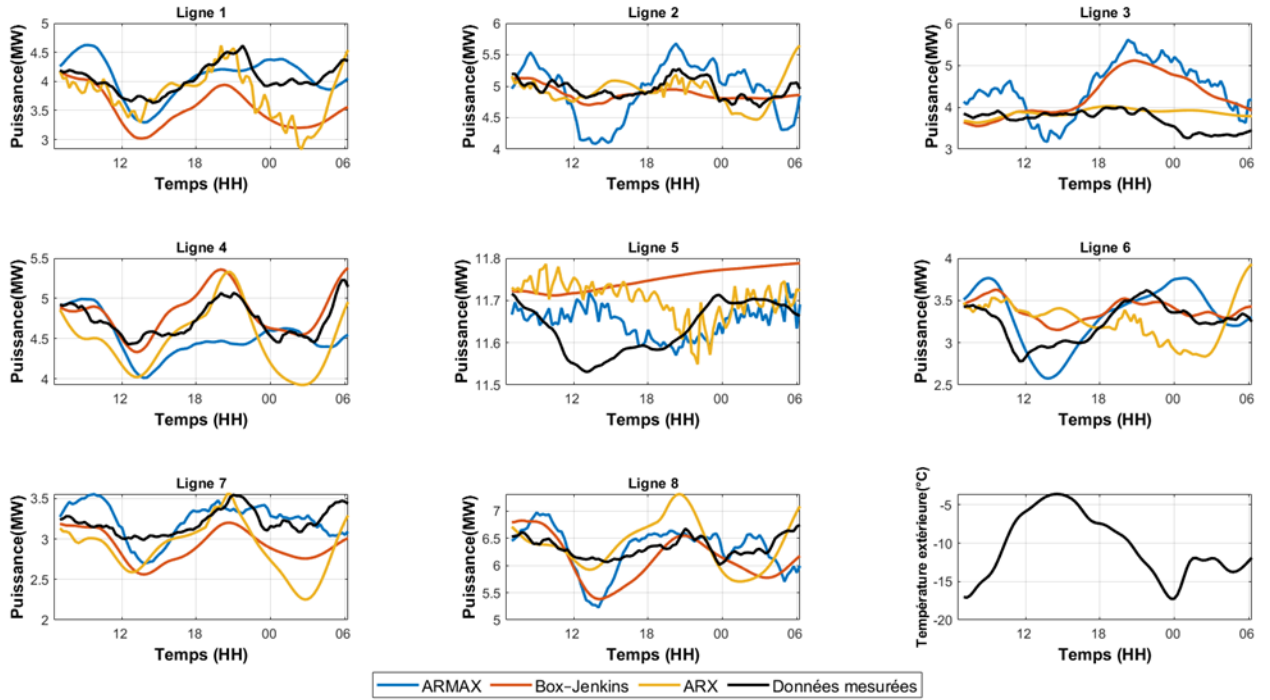
✓ Décalage 5 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 5 heure de décalage



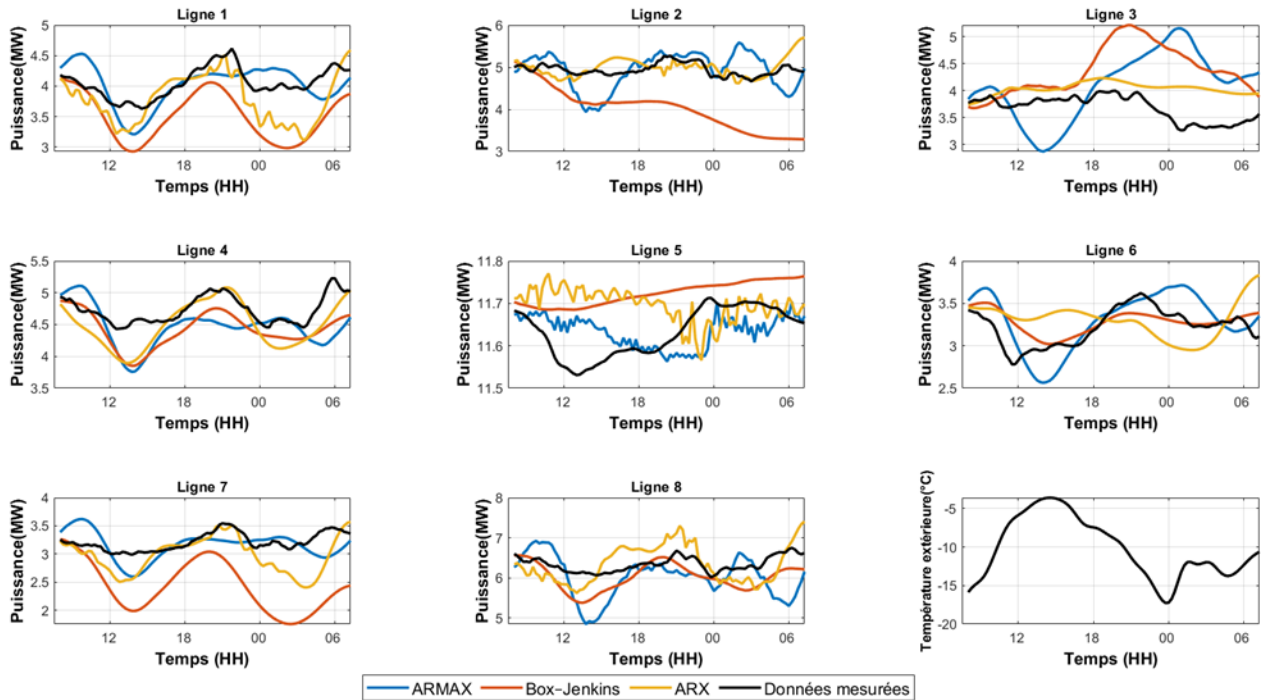
✓ Décalage 6 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 6 heure de décalage



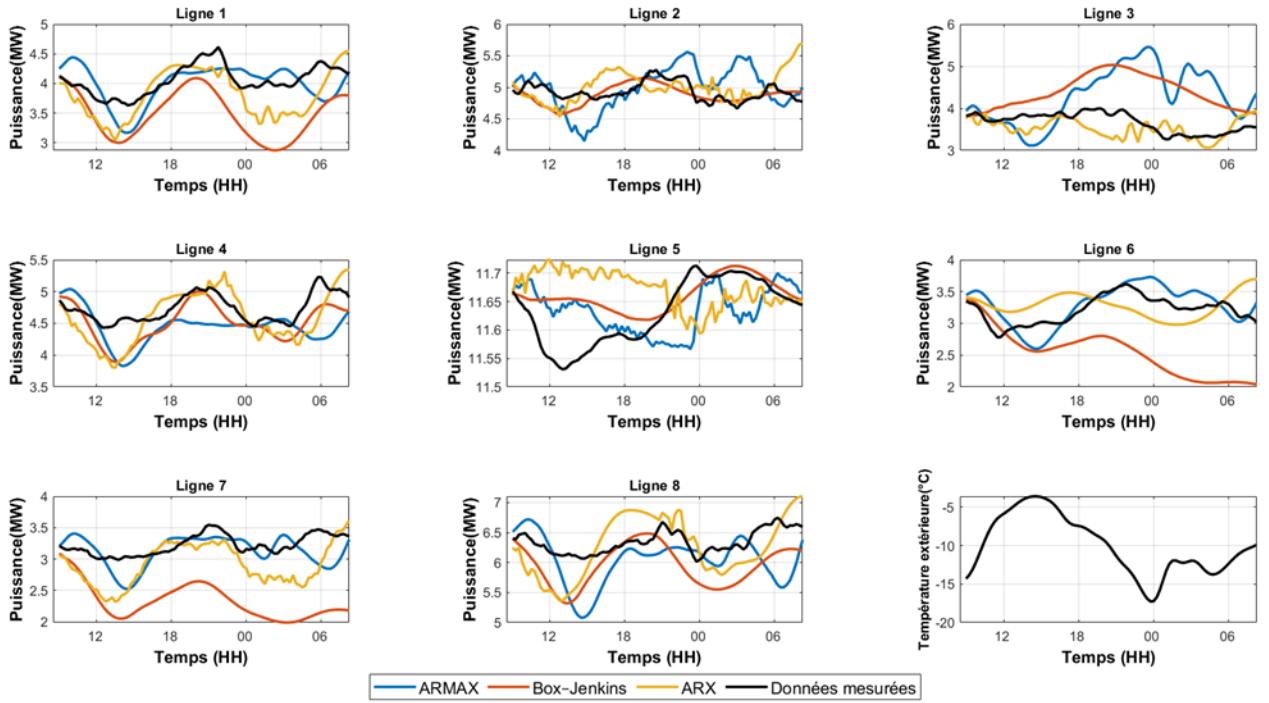
✓ Décalage 7 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 7 heure de décalage



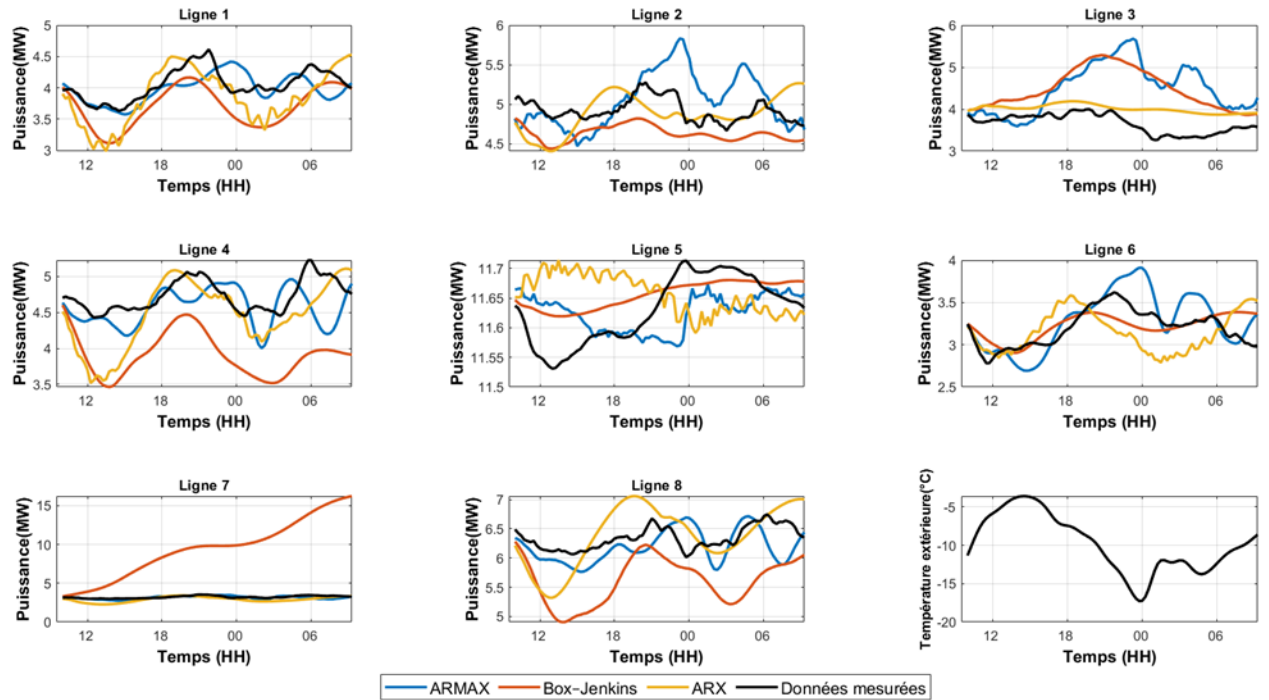
✓ Décalage 8 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 8 heure de décalage



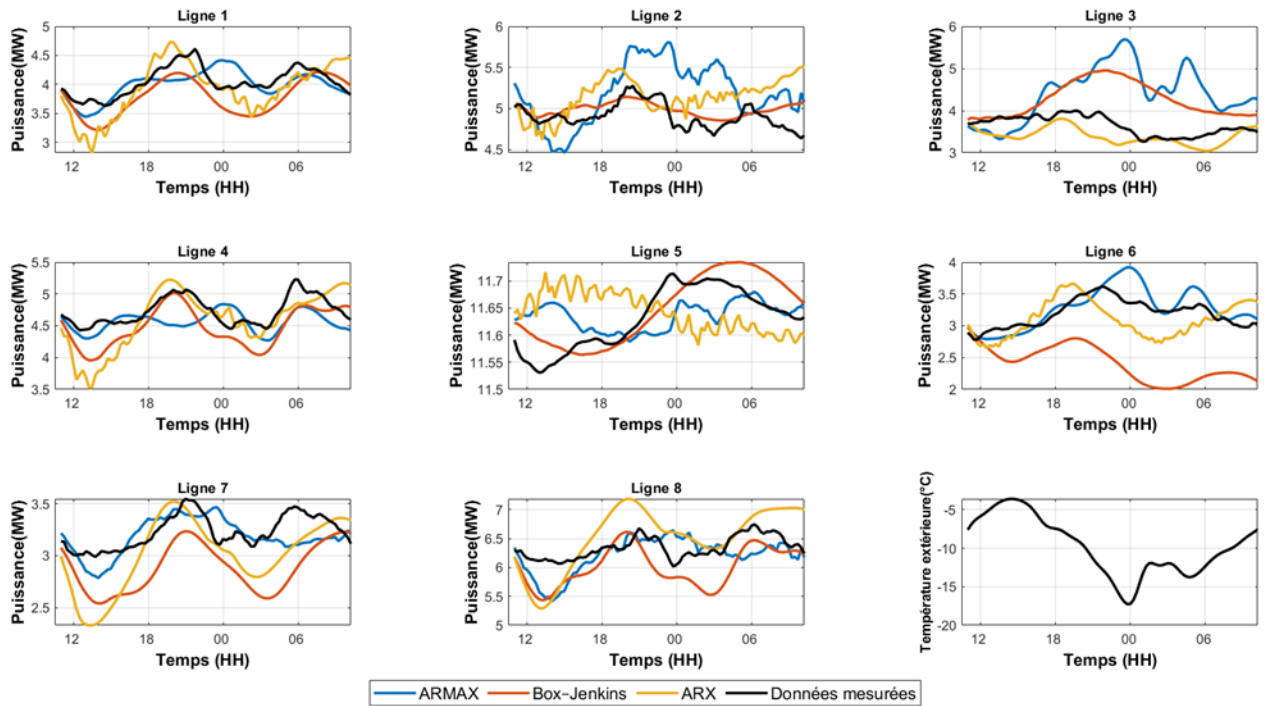
✓ Décalage 9 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 9 heure de décalage



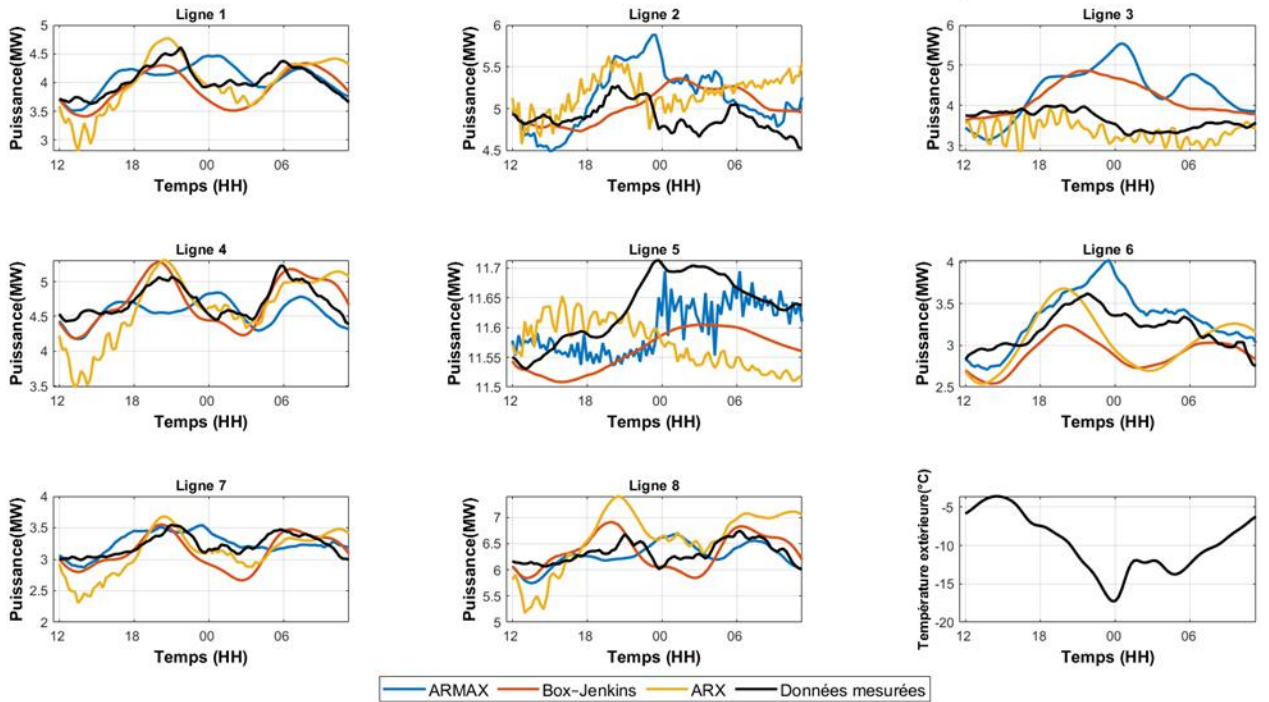
✓ Décalage 10 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 10 heure de décalage



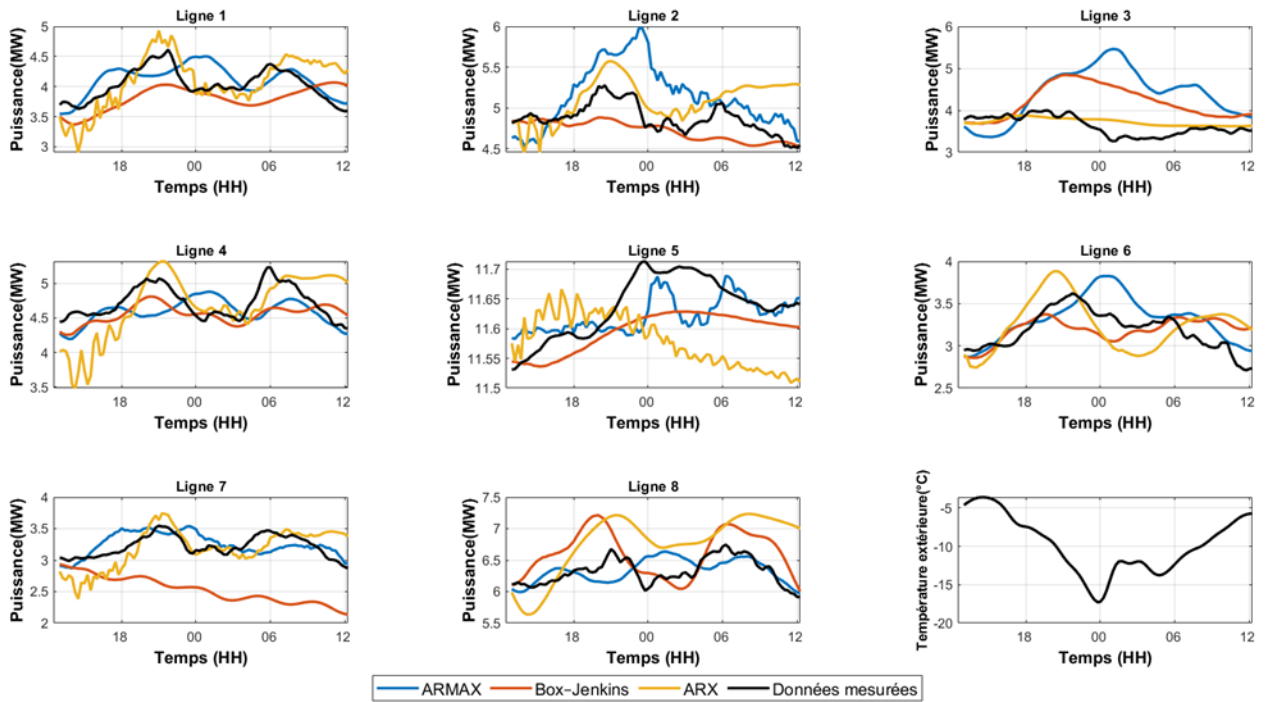
✓ Décalage 11 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 11 heure de décalage



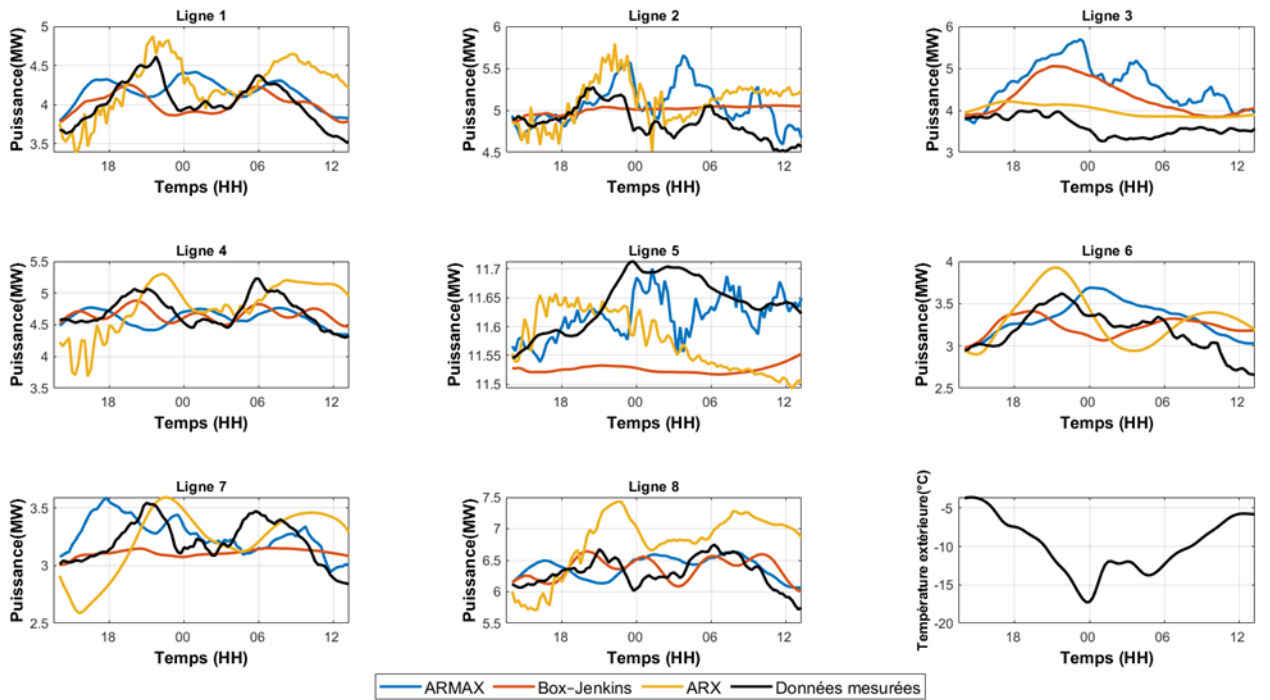
✓ Décalage 12 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 12 heure de décalage



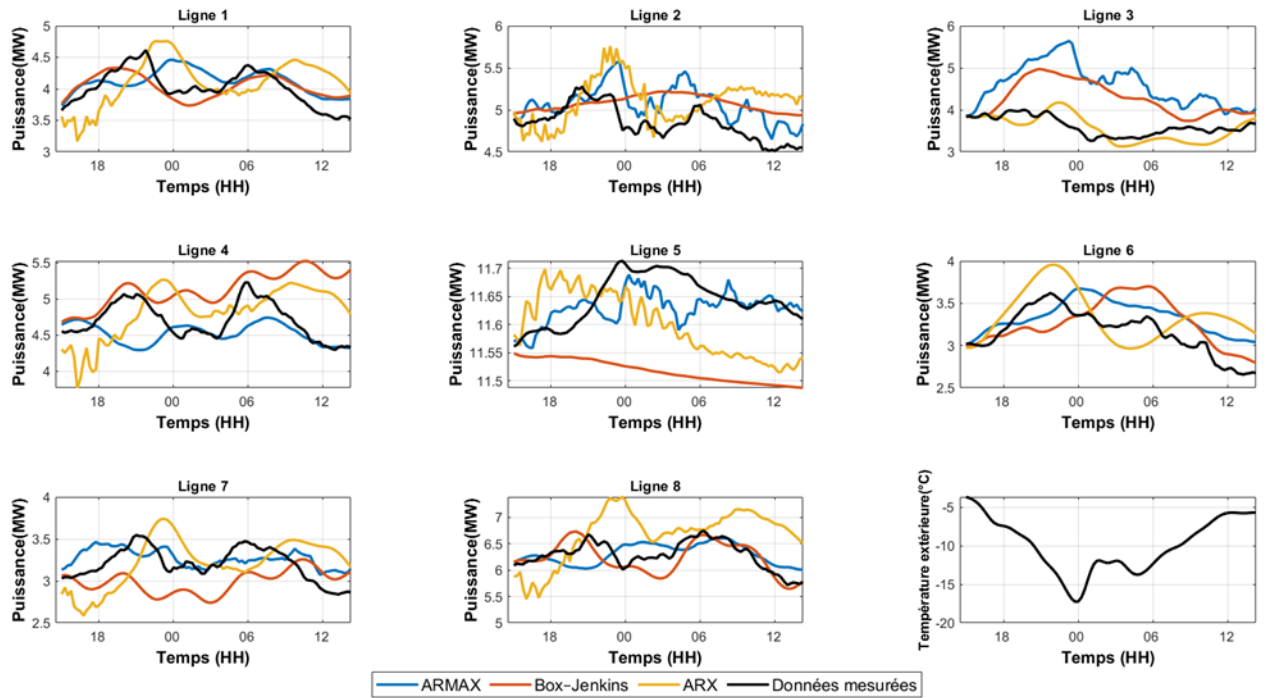
✓ Décalage 13 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 13 heure de décalage



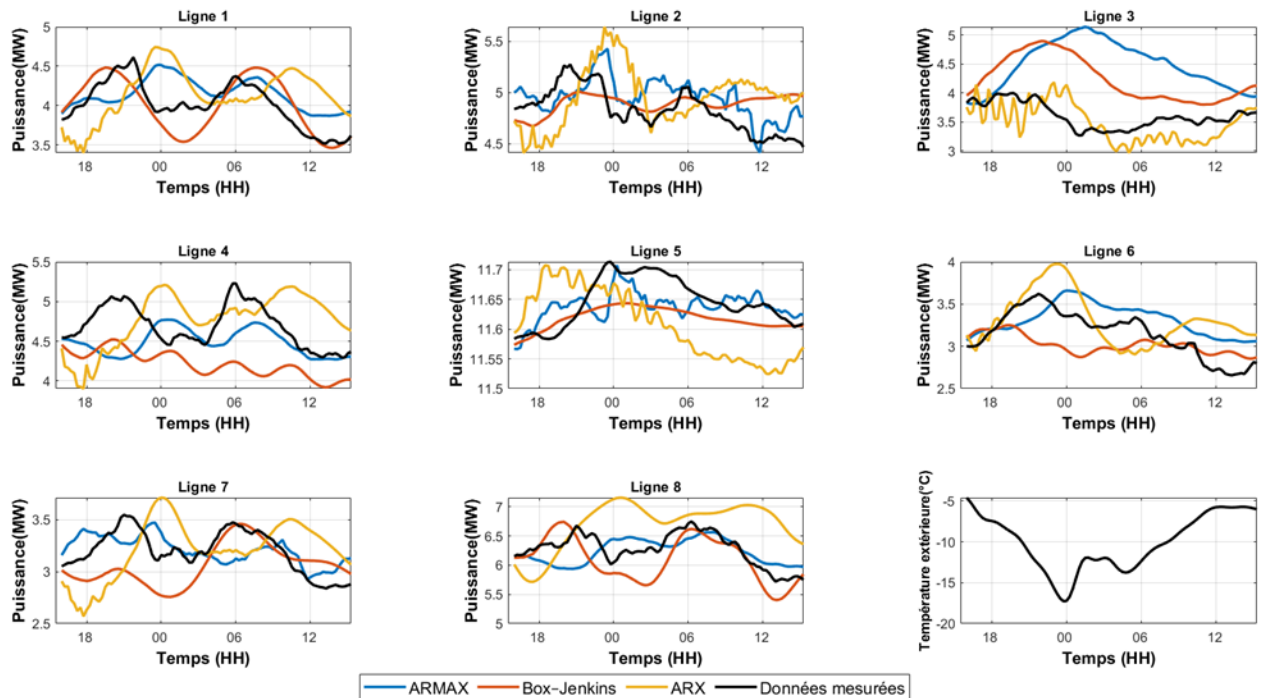
✓ Décalage 14 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 14 heure de décalage



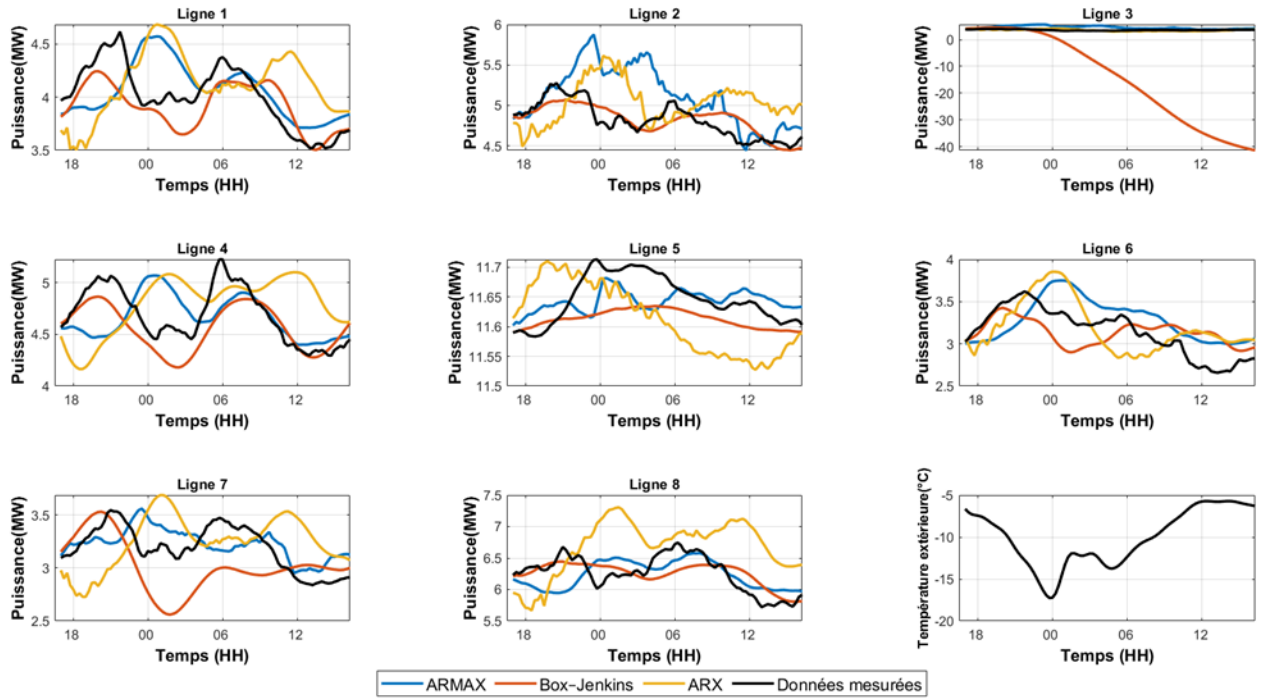
✓ Décalage 15 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 15 heure de décalage



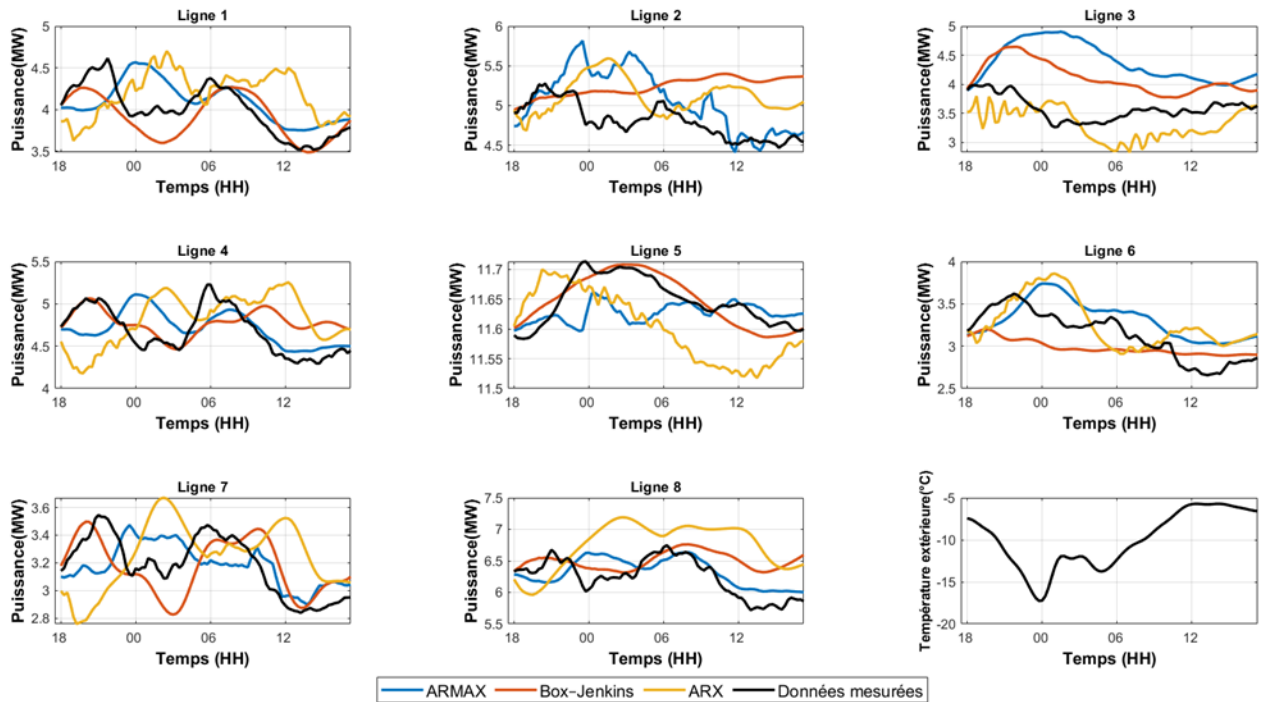
✓ Décalage 16 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 16 heure de décalage



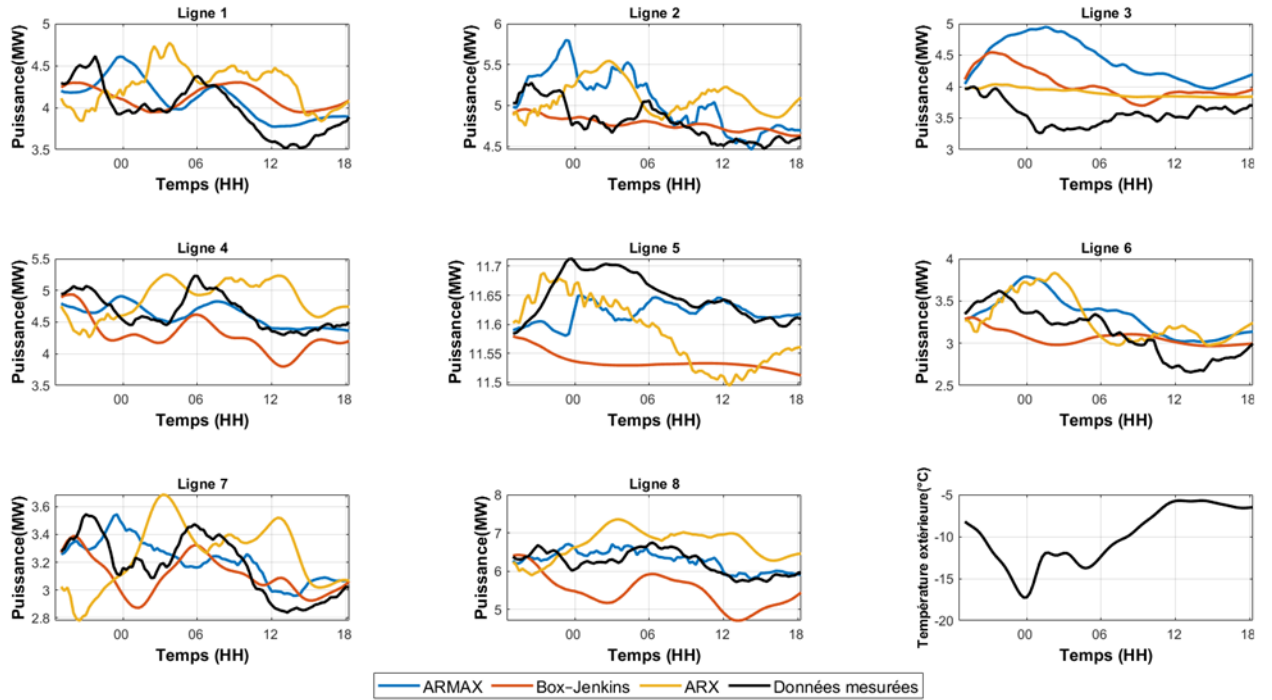
✓ Décalage 17 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 17 heure de décalage



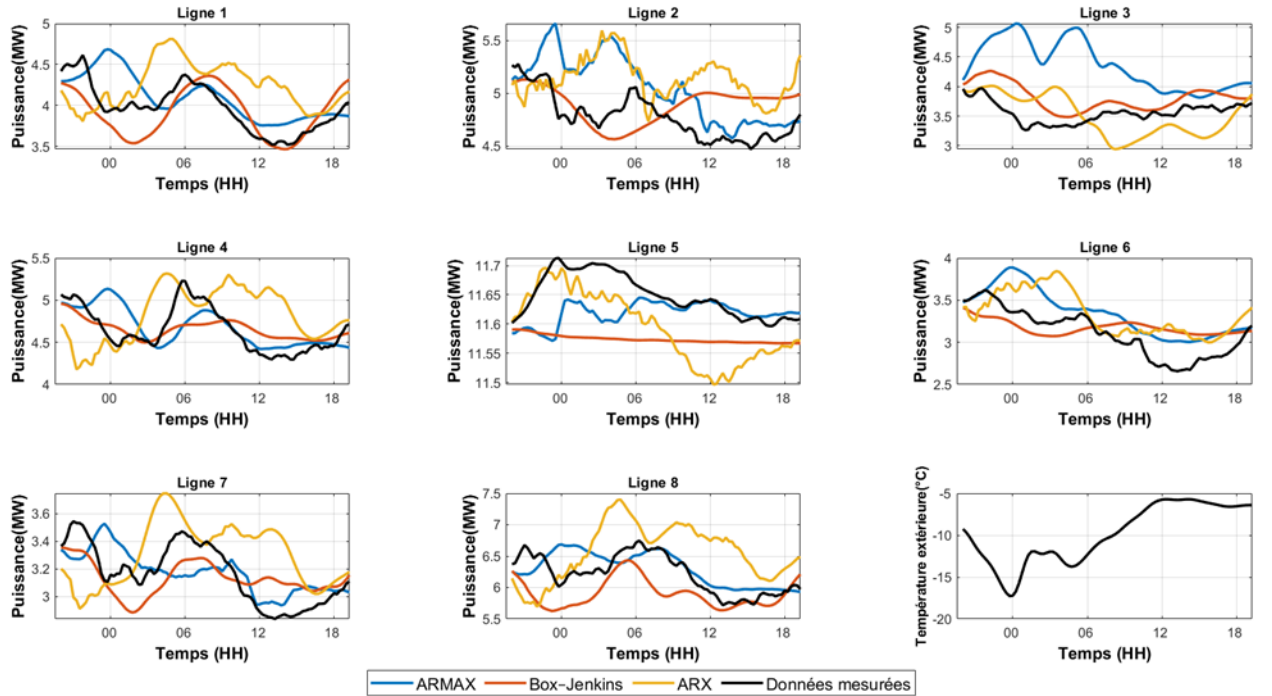
✓ Décalage 18 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 18 heure de décalage



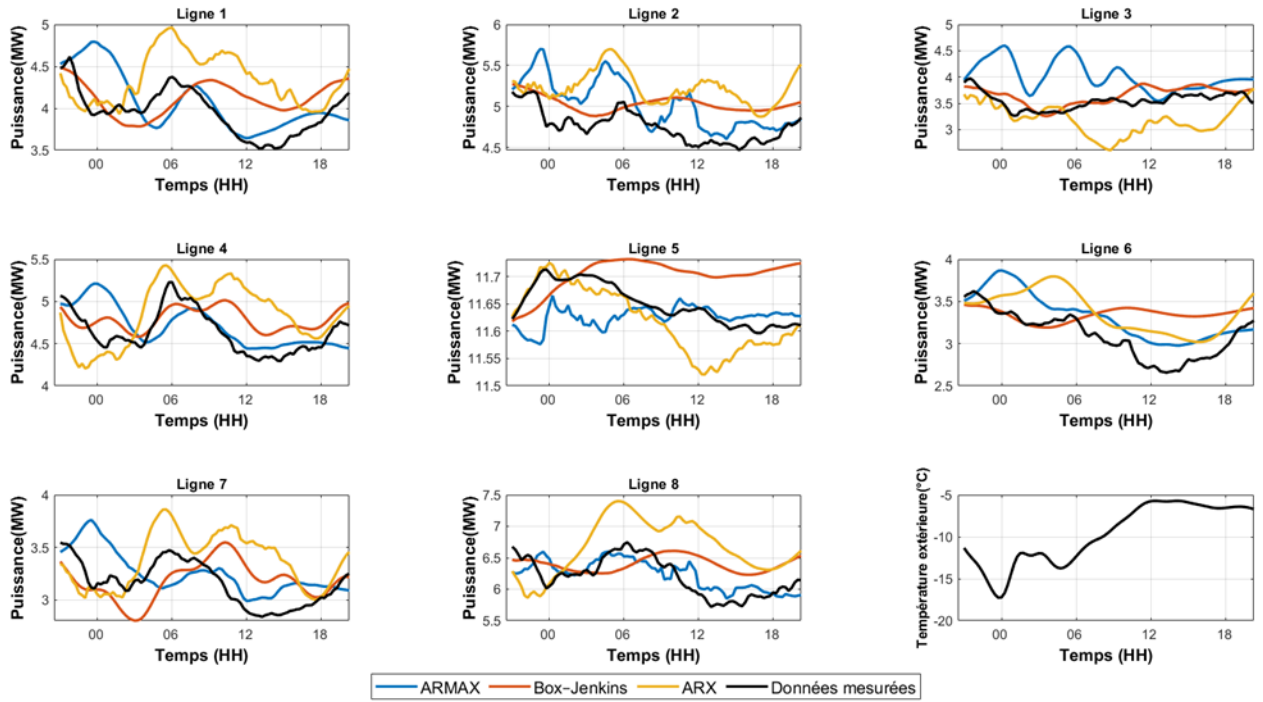
✓ Décalage 19 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 19 heure de décalage



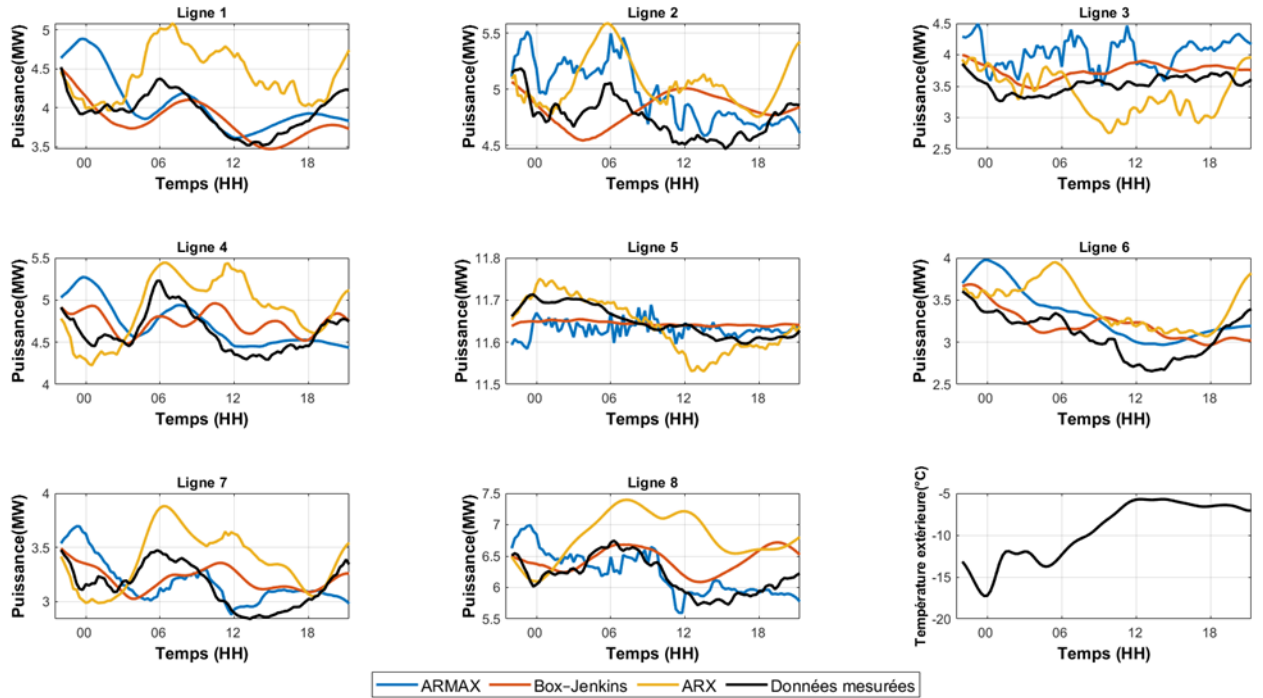
✓ Décalage 20 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 20 heure de décalage



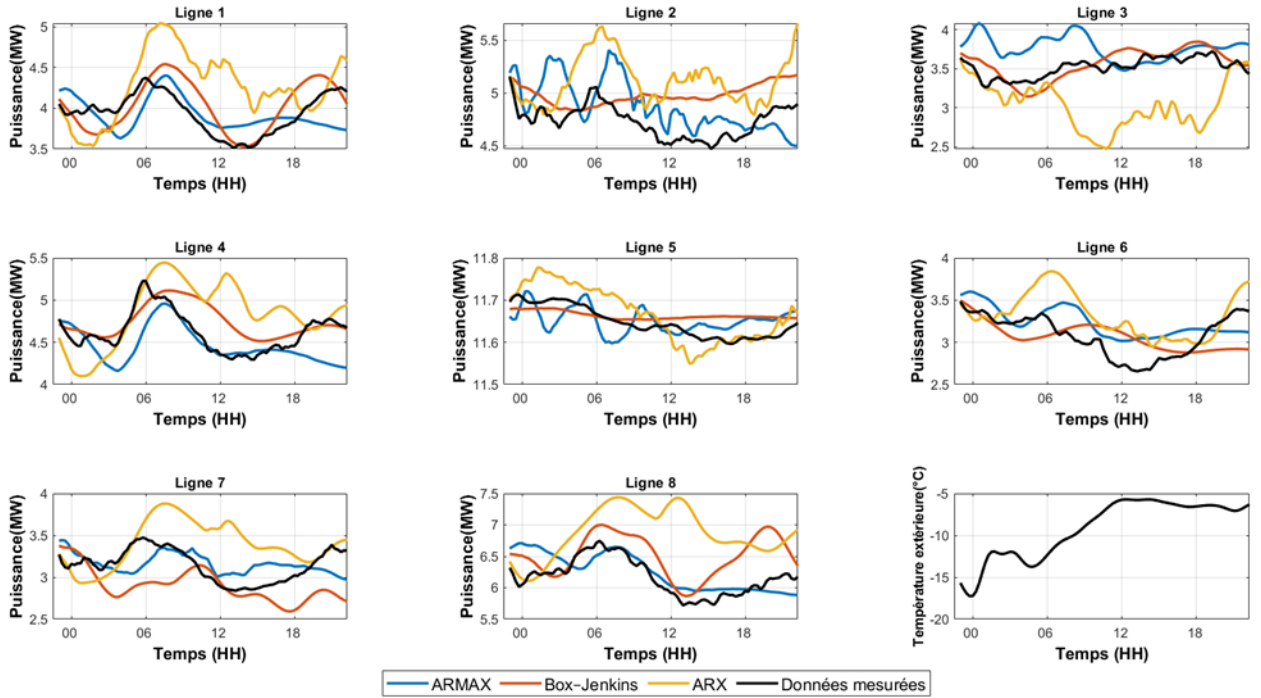
✓ Décalage 21 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 21 heure de décalage



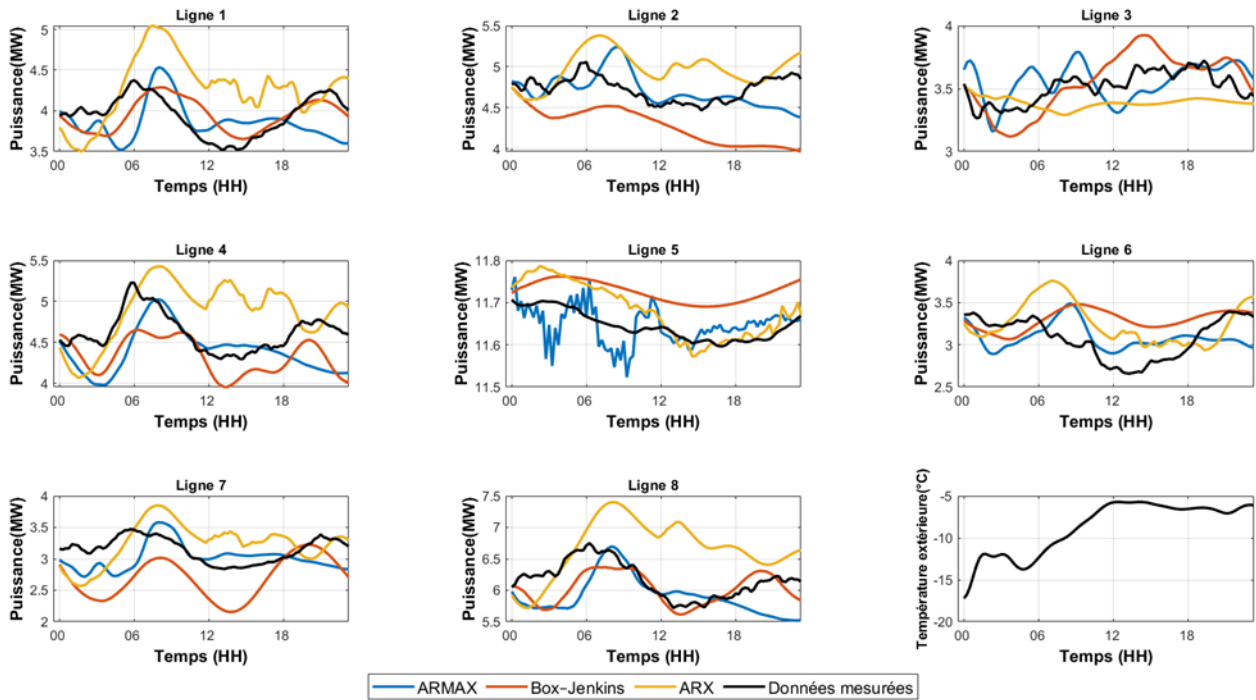
✓ Décalage 22 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 22 heure de décalage



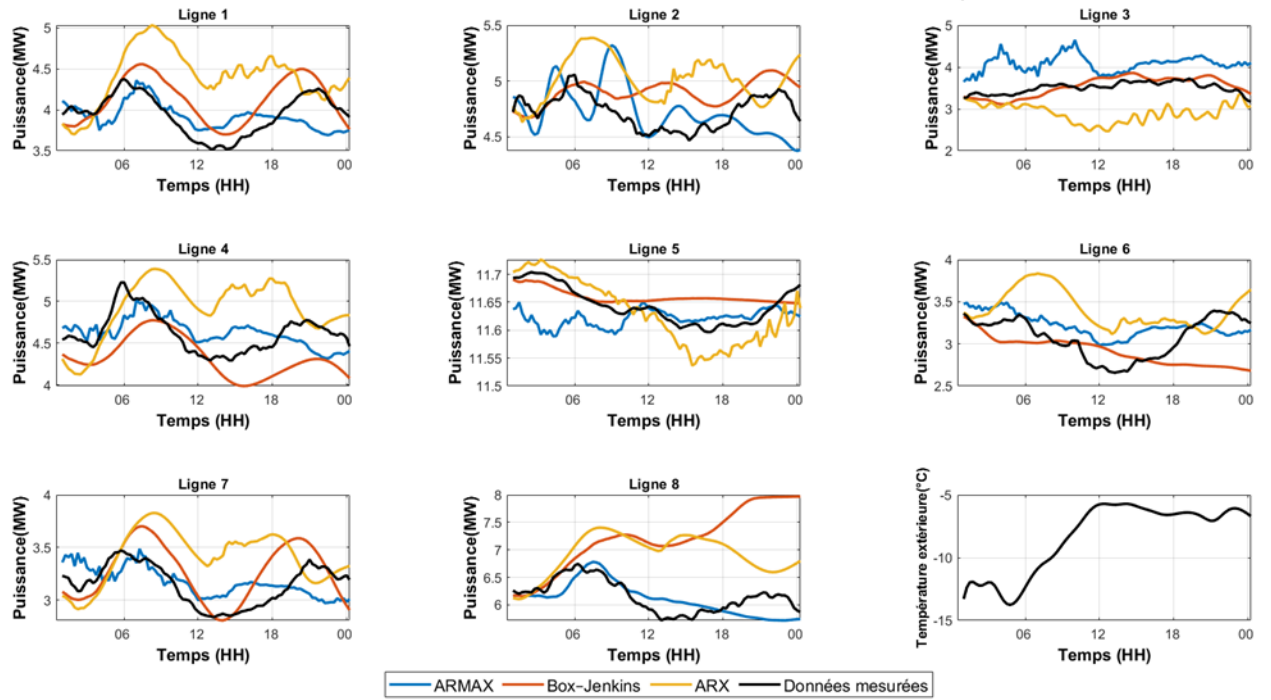
✓ Décalage 23 h

Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 23 heure de décalage



✓ Décalage 24 h

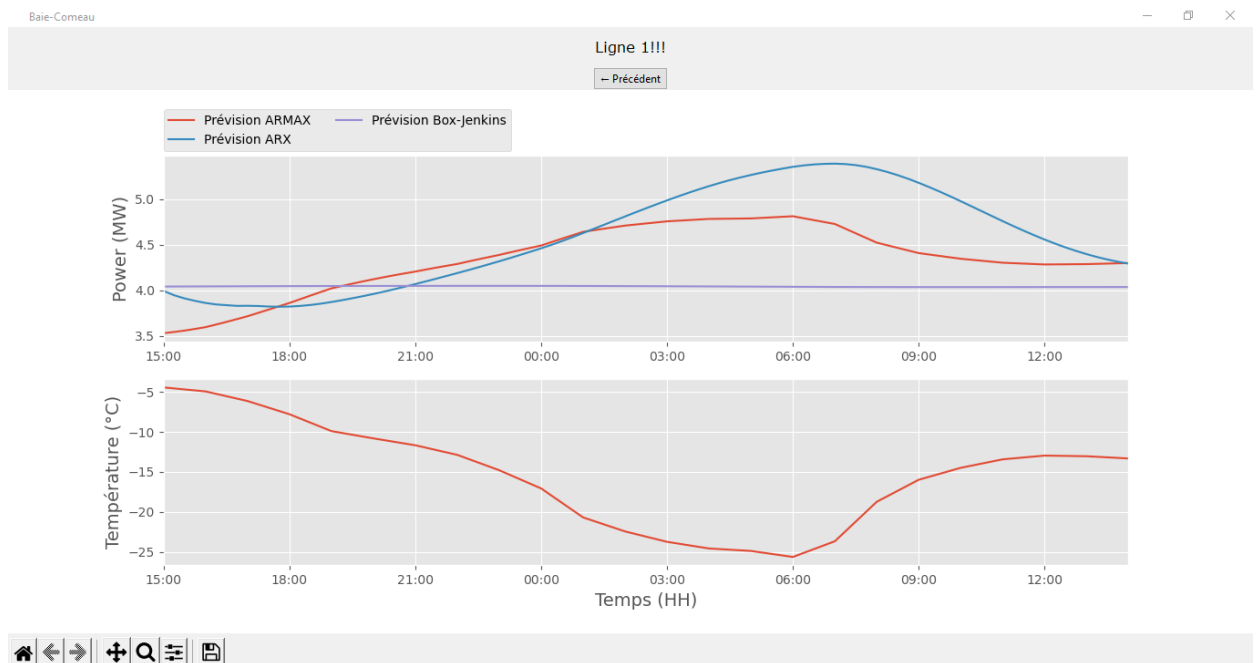
Prédiction avec les modèles suivant ARMAX, Box-Jenkins et ARX, 24 heure de décalage



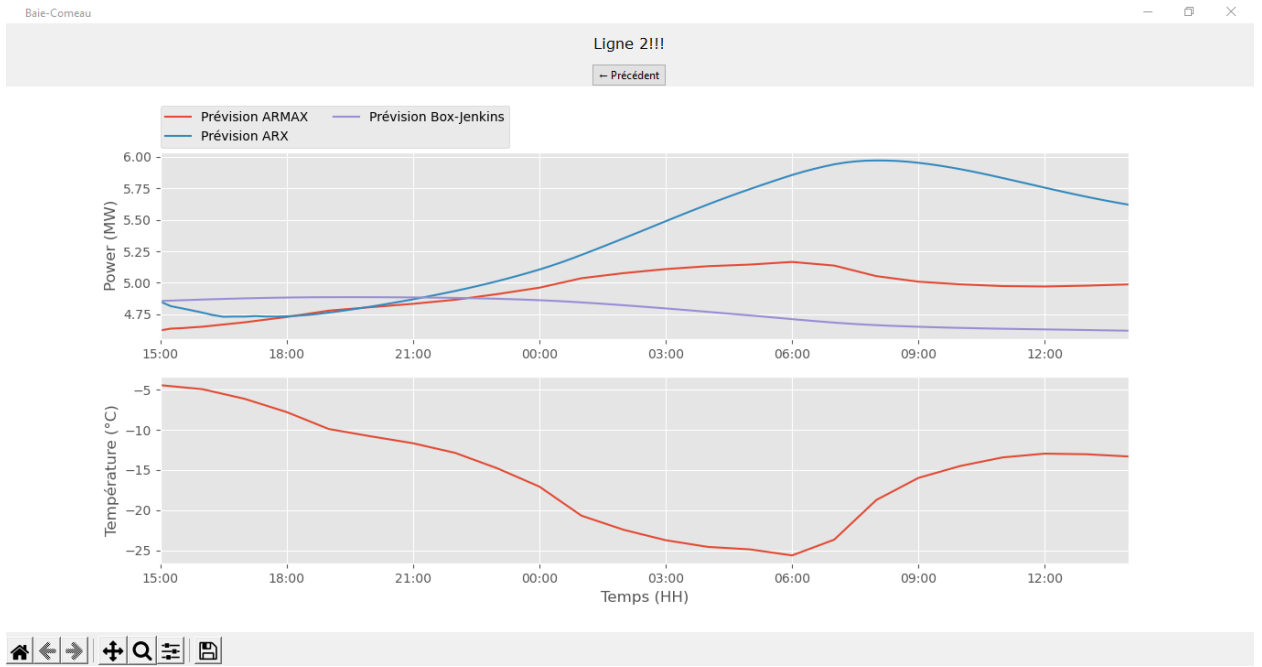
Annexe B – Résultats de la prévision obtenus lors de l'exécution sous Python

Nous présentons dans cette section, la prévision de la consommation électrique par les trois modèles ARMAX, ARX et Box-Jenkins du 2022/03/02 15h00 jusqu'à 2022/03/03 15h00.

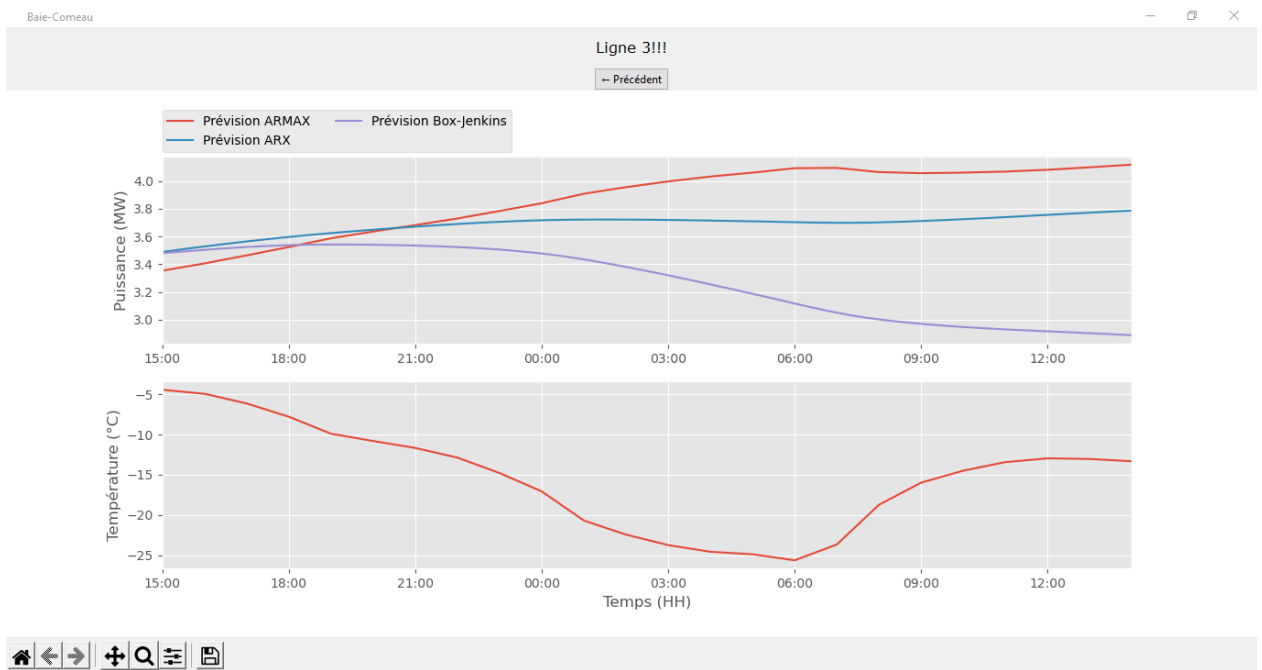
Bouton ligne 1



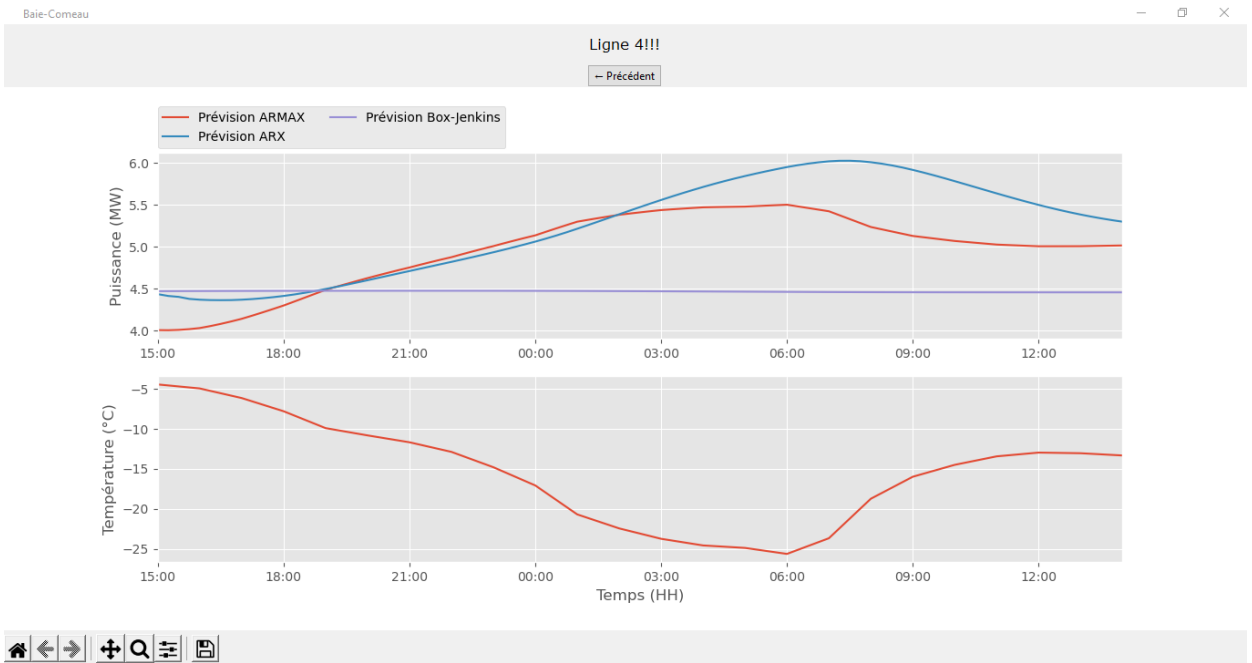
Bouton ligne 2



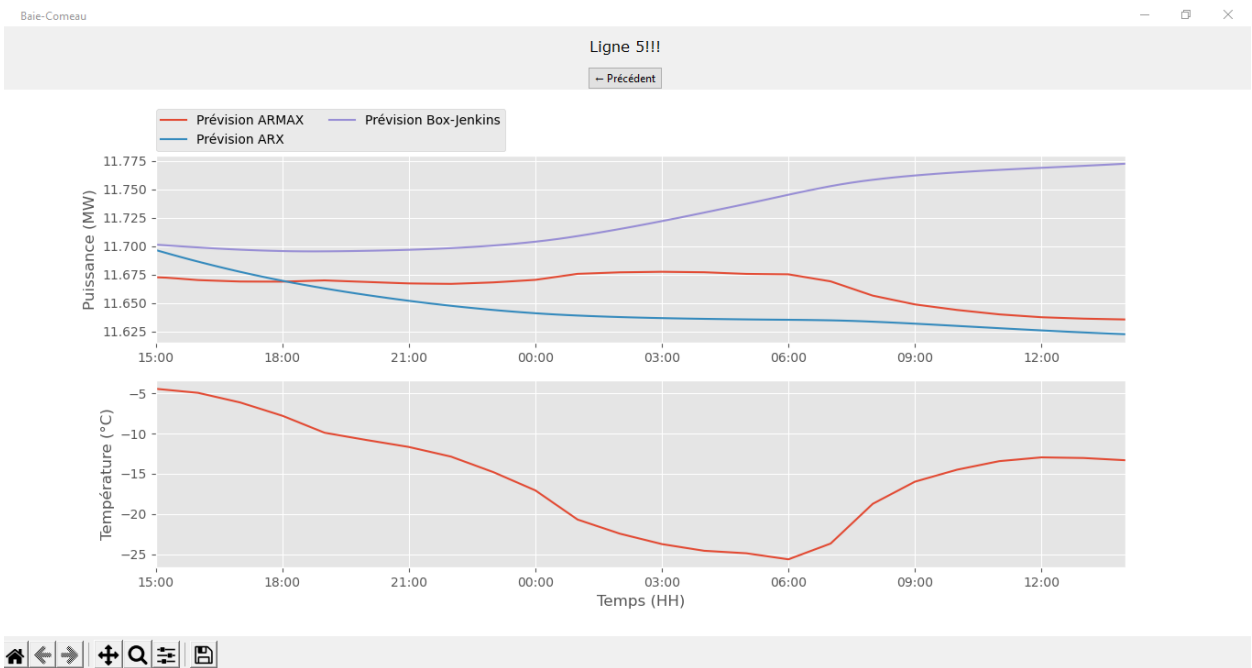
Bouton ligne 3



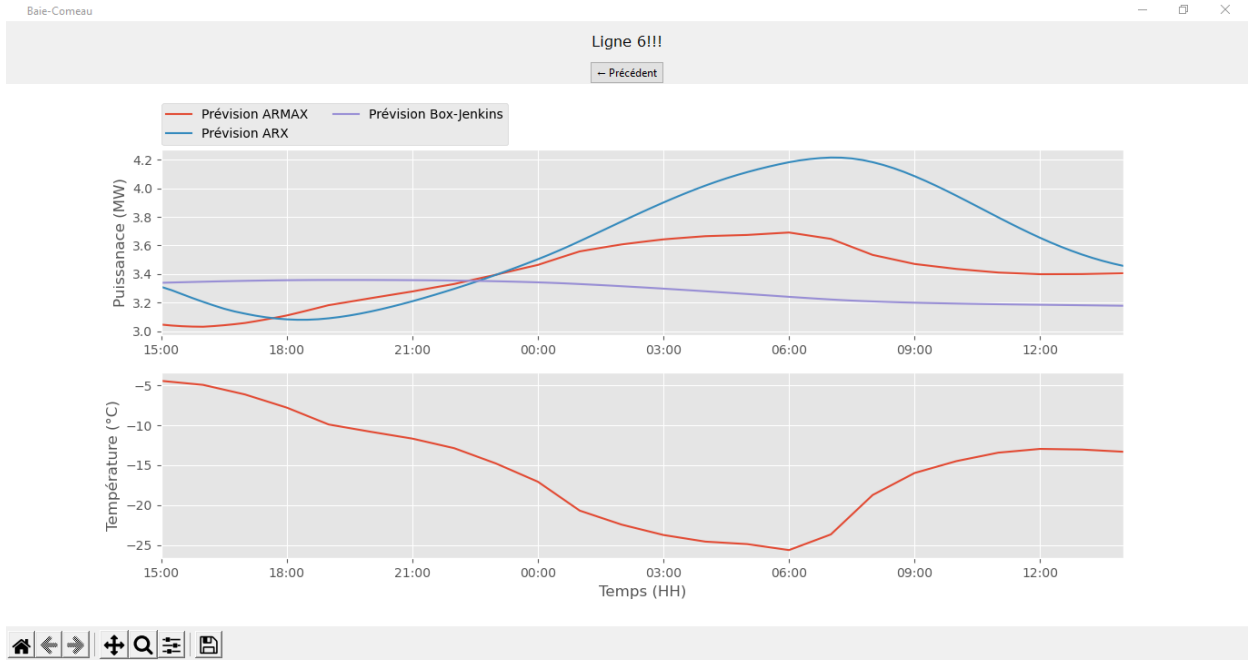
Bouton ligne 4



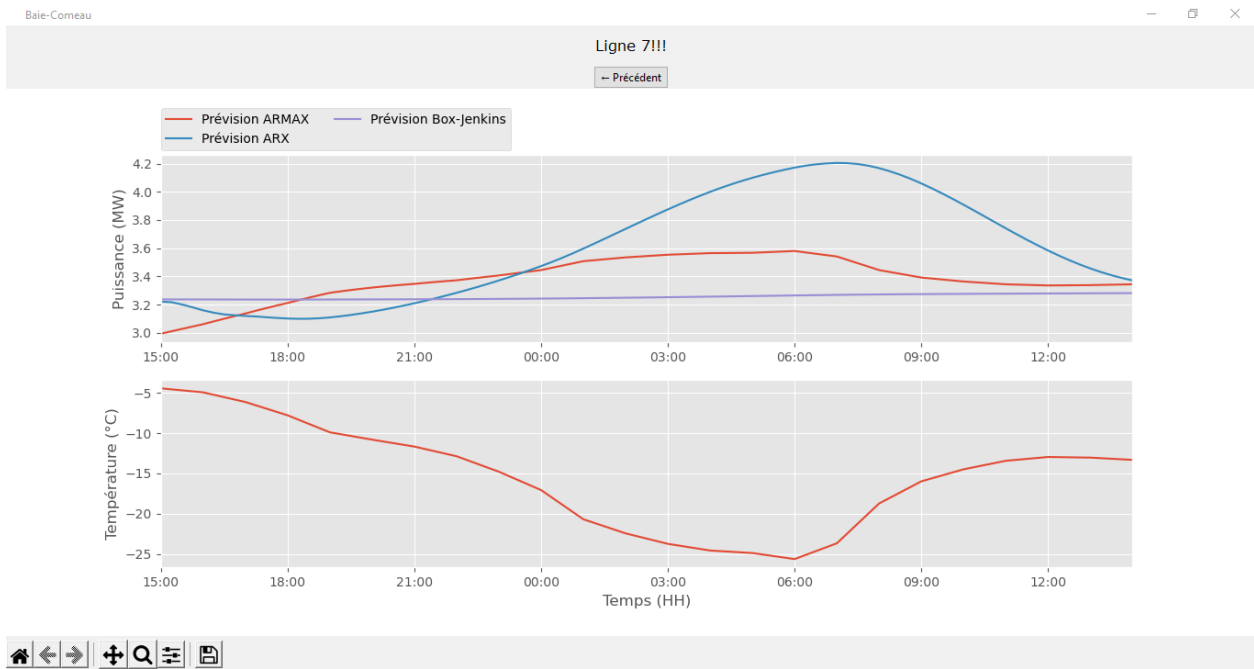
Bouton ligne 5




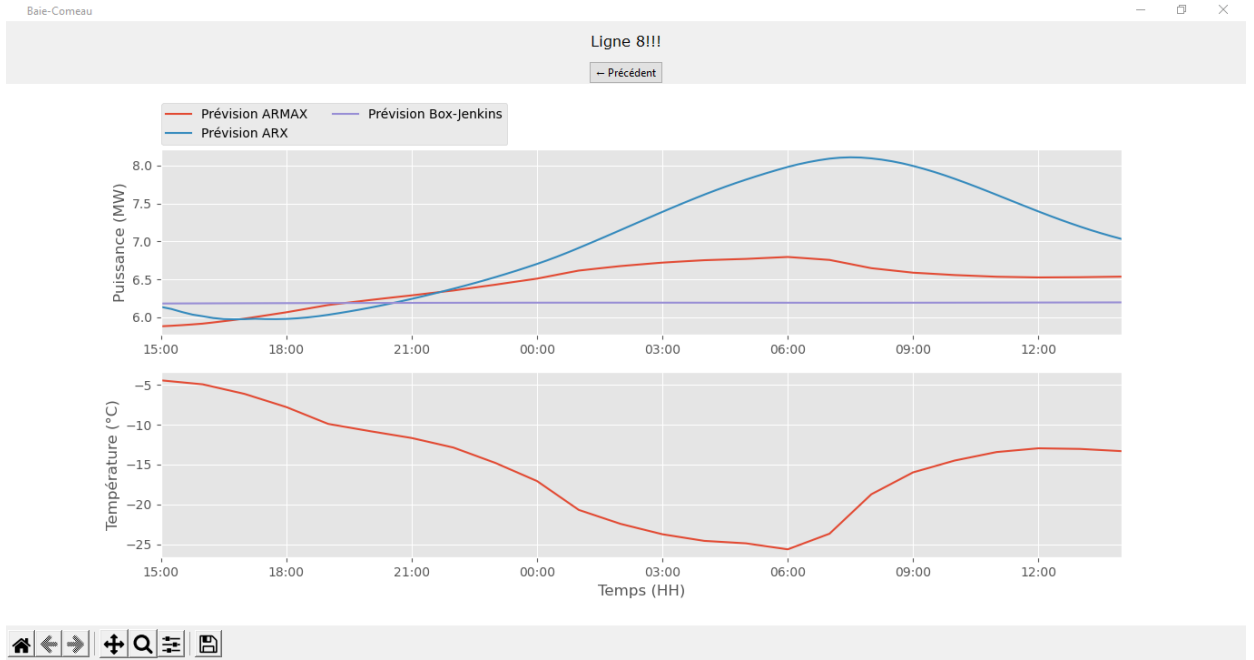
Bouton ligne 6




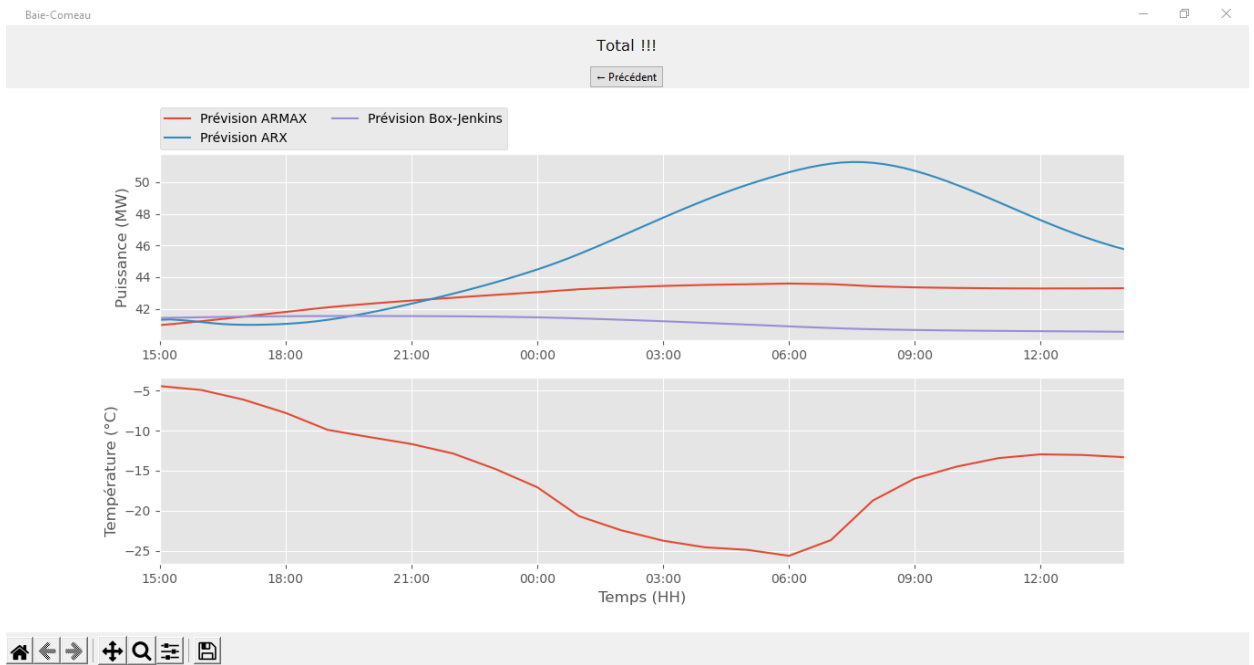
Bouton ligne 7



 Bouton ligne 8

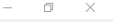


 Bouton la somme pour tous les lignes (Total)



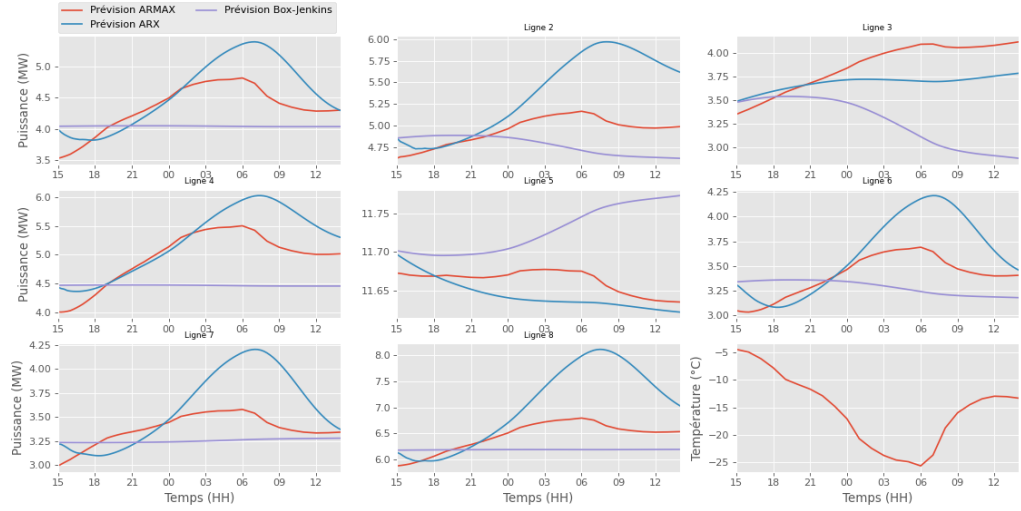
Bouton tous les lignes

Baie-Comeau



Tous les Lignes !!!

← Précédent



Annexe C – Code de programmation sous MATLAB et Python

Programme Matlab de la boucle de la prévision de la consommation électrique de la Ville Baie-Comeau; Cas de la première ligne de distribution.

```
clear all;
close all;
warning off;
load('Year2020real.mat')
% Les données météorologiques
Temperature_1h=te;
TemperatureD_1h=td;
Vent_1h=vv;
Irradiation_1h=irr;
% Extrapolation de données
Date_start = datenum('2020/01/01 01:00');
Sample_rate = 1/24;
Time_1h =
[Date_start:Sample_rate:Date_start+Sample_rate*(length(te)-1)];
Time_15m = [Date_start-
Sample_rate/2:Sample_rate/4:Date_start+Sample_rate*(length(te)-
1)+Sample_rate/4];
Temperature_15m = (interp1(Time_1h,Temperature_1h, Time_15m,
'spline','extrap')).';
TemperatureD_15m= (interp1(Time_1h,TemperatureD_1h, Time_15m,
'spline','extrap')).';
Vent_15m= (interp1(Time_1h,Vent_1h, Time_15m,
'spline','extrap')).';
Irradiation_15m= (interp1(Time_1h,Irradiation_1h, Time_15m,
'spline','extrap')).';
% Prétraitement des données
PowerkW_L1=BC_PuissancekW(:,1);
PowerkW_L2=BC_PuissancekW(:,2);
PowerkW_L3=BC_PuissancekW(:,3);
PowerkW_L4=BC_PuissancekW(:,4);
PowerkW_L5=BC_PuissancekW(:,5);
PowerkW_L6=BC_PuissancekW(:,6);
PowerkW_L7=BC_PuissancekW(:,7);
PowerkW_L8=BC_PuissancekW(:,8);
% boucle de prévision de la consommation pour 24 fois equivalent
décalage +1 jusqu'a 24 heures se termine
NTests = 24
```

```

for k=1:NTests
LD = 30*24*4; % NOMBRE DE POINTS
d=4*k;
H=24*4;
Try_PowerkW1=PowerkW_L1(1+d:LD+d);
Try_PowerkW1_test=PowerkW_L1(1+d+H:LD+d+H);
Try_PowerkW2=PowerkW_L2(1+d:LD+d);
Try_PowerkW2_test=PowerkW_L2(1+d+H:LD+d+H);
Try_PowerkW3=PowerkW_L3(1+d:LD+d);
Try_PowerkW3_test=PowerkW_L3(1+d+H:LD+d+H);
Try_PowerkW4=PowerkW_L4(1+d:LD+d);
Try_PowerkW4_test=PowerkW_L4(1+d+H:LD+d+H);
Try_PowerkW5=PowerkW_L5(1+d:LD+d);
Try_PowerkW5_test=PowerkW_L5(1+d+H:LD+d+H);
Try_PowerkW6=PowerkW_L6(1+d:LD+d);
Try_PowerkW6_test=PowerkW_L6(1+d+H:LD+d+H);
Try_PowerkW7=PowerkW_L7(1+d:LD+d);
Try_PowerkW7_test=PowerkW_L7(1+d+H:LD+d+H);
Try_PowerkW8=PowerkW_L8(1+d:LD+d);
Try_PowerkW8_test=PowerkW_L8(1+d+H:LD+d+H);
Try_Temperature=Temperature_15m(1+d:LD+d);
Try_Temperature_test=Temperature_15m(1+d+H:LD+d+H);
Try_TemperatureD=TemperatureD_15m(1+d:LD+d);
Try_TemperatureD_test=TemperatureD_15m(1+d+H:LD+d+H);
Try_Vent=Vent_15m(1+d:LD+d);
Try_Vent_test=Vent_15m(1+d+H:LD+d+H);
Try_Irradiation=Irradiation_15m(1+d:LD+d);
Try_Irradiation_test=Irradiation_15m(1+d+H:LD+d+H);
Try_time_test = Time_15m(1+d+H:LD+d+H)';
% préparation de les entrées et les sorties pour entraîner les
modèles
% les données de l'entraînement
output=Try_PowerkW1/1000;
input=Try_Temperature;
Try=iddata(output,input,15*60);
opt = armaxOptions;
opt.Focus = 'simulation';
opt.SearchMethod = 'lm';
opt.SearchOptions.MaxIterations = 10;
opt.Display = 'on';
Try_armax = armax(Try,[1 40 45 1],opt); % Modèle ARMAX
Try_bj = bj(Try,[1 10 45 30 1]); % Modèle Box-Jenkins
opt = bjOptions;
opt.Display = 'on';
opt.SearchOptions.MaxIterations = 50;
orders = [45 50 50];
[lambda,R] = arxRegul(Try,orders);
opt = arxOptions;

```

```

opt.Regularization.Lambda = lambda;
opt.Regularization.R = R;
Try_arx = arx(Try,orders,opt); % Modèle ARX
text_title=['Prédiction avec les modèles suivant ARMAX, Box-
Jenkins et ARX pour Ligne 1, ',num2str(k), ' heures de décalage' ]
% les données de test
y=Try_PowerkWl1_test/1000;
u=Try_Temperature_test;
data=iddata(y,u,15*60);
Try_pastdata=data(1:LD-H);
Try_FutureInputs = data(LD+1-H:LD);
Future_Time_test(k,:) = Try_time_test(LD+1-H:LD);
% Pour sortir les résultats de la prévision de la consommation
électrique pour 24 heures
Try_farmax=forecast(Try_armax,Try_pastdata,H,Try_FutureInputs);
Try_fbj=forecast(Try_bj,Try_pastdata,H,Try_FutureInputs);
Try_farx=forecast(Try_arx,Try_pastdata,H,Try_FutureInputs);
% convertir les valeurs de sortie (fit) vers les valeurs
numeriques
dataFilt1 = idfilt(Try_farmax,{[1/3 1/3 1/3],[1]});
yFarmax = dataFilt1.y;
dataFilt2 = idfilt(Try_fbj,{[1/3 1/3 1/3],[1]});
yFbj = dataFilt2.y;
dataFilt3 = idfilt(Try_farx,{[1/3 1/3 1/3],[1]});
yFarx = dataFilt3.y;
dataFilt4 = idfilt(Try_FutureInputs,{[1/3 1/3 1/3],[1]});
yTry_FutureInputs = dataFilt4.y;
Future_power(k,:) = dataFilt4.y;
Future_temp(k,:) = dataFilt4.u;
T_Start = Future_Time_test(k,1)
T_End = Future_Time_test(k,end)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%les
figures%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
/////
figure();
x0=100; y0=300; width=1000; height=450;
set(gcf,'position',[x0,y0,width,height])
subplot(2,1,1);
plot(Future_Time_test(k,3:end), yFarmax(3:end), 'linewidth',2);
hold;
plot(Future_Time_test(k,3:end), yFbj(3:end), 'linewidth',2);
plot(Future_Time_test(k,3:end), yFarx(3:end), 'linewidth',2);
plot(Future_Time_test(k,3:end), Future_power(k,3:end), 'k',
'linewidth',2);
xlim([T_Start T_End]);
xlabel('Temps
(HH)', 'FontSize',12,'FontWeight','bold','Color','k');
ylabel('Puissance (MW)', 'FontSize',12,'FontWeight','bold','Color','

```

```

k')
legend('ARMAX', 'Box-Jenkins', 'ARX', 'Données mesurées',
'FontSize',10)
grid on;
datetick('x','HH ', 'keeplimits')
title(text_title);
subplot(2,1,2);
plot(Future_Time_test(k,3:end), Future_temp(k,3:end), 'k',
'linewidth',2);
xlim([T_Start T_End]);
xlabel('Temps
(HH)', 'FontSize',12, 'FontWeight', 'bold', 'Color', 'k');
ylabel('Température
extérieure (°C)', 'FontSize',12, 'FontWeight', 'bold', 'Color', 'k')
grid on;
datetick('x','HH ', 'keeplimits')

Error_ARMAX(k,:) = yTry_FutureInputs-yFarmax;
Error_BJ(k,:) = yTry_FutureInputs-yFbj;
Error_ARX(k,:) = yTry_FutureInputs-yFarx;

RMSE_armax(k)=sqrt(mean((yTry_FutureInputs-yFarmax).^2)); %calcule
"Root Mean Square Error" pour prévision Armax
RMSE_bj(k)=sqrt(mean((yTry_FutureInputs-yFbj).^2)); %calcule "Root
Mean Square Error" pour prévision Box Jenkins
RMSE_arx(k)=sqrt(mean((yTry_FutureInputs-yFarx).^2)); %calcul
"Root Mean Square Error" pour prévision Arx

Percent_Error_armax(k) = mean(100 * abs(yTry_FutureInputs-yFarmax)
./ yFarmax); %Calcule "Percent Error" pour prévision Armax
Percent_Error_bj(k) = mean(100 * abs(yTry_FutureInputs-yFbj) ./
yFbj); %Calcule "Percent Error" pour prévision Box Jenkins
Percent_Error_arx(k) = mean(100 * abs(yTry_FutureInputs-yFarx) ./
yFarx); %Calcule "Percent Error" pour prévision Arx

end

%% Figures comparaison
T_Start = Future_Time_test(1,1)
T_End = Future_Time_test(end,end)
figure()
hold on
for k=1:NTests
xdata = Future_Time_test(k,:)
plot(xdata,100*Error_ARMAX(k,:) ./Future_power(k,:))
end
xlabel('Temps (HH)', 'FontSize',12, 'FontWeight', 'bold', 'Color', 'k')

```

```

ylabel('Erreur [%]', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'k')
title('Prévisions
ARMAX', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'k')
ylim([-50 50])
xlim([T_Start T_End])
x0=100; y0=300; width=1000; height=350;
set(gcf, 'position', [x0,y0,width,height])
grid on
datetick('x', 'HH', 'keplimits')

figure()
hold on
for k=1:NTests
    xdata = Future_Time_test(k,:);
    plot(xdata, 100*Error_BJ(k,:)./Future_power(k,:))
end
xlabel('Temps (HH)', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'k')
ylabel('Erreur [%]', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'k')
title('Prévisions Box-
Jenkins', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'k')
ylim([-50 50])
xlim([T_Start T_End])
x0=100; y0=300; width=1000; height=350;
set(gcf, 'position', [x0,y0,width,height])
grid on
datetick('x', 'HH', 'keplimits')

figure()
hold on
for k=1:NTests
    xdata = Future_Time_test(k,:);
    plot(xdata, 100*Error_ARX(k,:)./Future_power(k,:))
end
xlabel('Temps (HH)', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'k')
ylabel('Erreur [%]', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'k')
title('Prévisions
ARX', 'FontSize', 12, 'FontWeight', 'bold', 'Color', 'k')
ylim([-50 50])
xlim([T_Start T_End])
x0=100; y0=300; width=1000; height=350;
set(gcf, 'position', [x0,y0,width,height])
grid on
datetick('x', 'HH', 'keplimits')

% Mean absolute error as function of H

xdata = [1:1:H]/4
figure()

```

```

plot(xdata,100*abs(mean(Error_ARMAX)), 'linewidth',2)
hold on
plot(xdata,100*abs(mean(Error_BJ)), 'linewidth',2)
plot(xdata,100*abs(mean(Error_ARX)), 'linewidth',2)
x0=100; y0=300; width=1000; height=350;
set(gcf,'position',[x0,y0,width,height])
xlabel('Horizon
(Heures)', 'FontSize',12, 'FontWeight','bold', 'Color','k')
ylabel('Erreur [%]', 'FontSize',12, 'FontWeight','bold', 'Color','k')
title('MAE', 'FontSize',12, 'FontWeight','bold', 'Color','k')
ylim([ 0 50])
grid on
legend('ARMAX', 'Box-Jenkins', 'ARX')

```

Programmation de code sous Python à l'aide de la bibliothèque Tkinter pour créer une interface utilisateur graphique facilitant l'utilisation du programme.

```

1. ## Appel de la bibliothèque
2. %matplotlib inline
3. import numpy as np
4. import matplotlib.pyplot as plt
5. import pandas as pd
6. import datetime as dt
7. import requests
8. from datetime import datetime
9. from statsmodels.tsa.arima_model import ARIMA
10. from scipy.interpolate import interp1d
11. import seaborn
12. from arch.__future__ import reindexing
13. from arch.univariate import ARX, GARCH, ZeroMean, arch_model
14. import matplotlib
15. matplotlib.use("TkAgg")
16. from matplotlib.backends.backend_tkagg import NavigationToolbar2Tk as
    NavigationToolbar2TkAgg
17. #from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg,
    NavigationToolbar2TkAgg
18. from tensorflow import keras
19. from matplotlib.figure import Figure
20. import tkinter as tk
21. from tkinter import ttk
22. import matplotlib.dates as mdates
23. import matplotlib.animation as animation #
24. from matplotlib import style #
25. # Appel de données
26. data=pd.read_excel('DataSetBC.xlsx', index_col='Time',parse_dates=True)
27. # Appel API
28. url = "https://api.openweathermap.org/data/2.5/onecall?lat=49.215824721518416&lon=-
    68.1829869548623&exclude=minutely,current,daily,alerts&appid=2504b820d423b47b0bcfcb0c8
    7897f6f&units=metric"
29. response = requests.get(url).json()
30. results_perhour = []
31. results_temps = []
32. # extraire les données de température

```

```

33. i=0
34. while i<=23:
35.     #print(i)
36.     i=i+1
37.     perhour=response['hourly'][i]['temp']
38.     timeserie=response['hourly'][i]['dt']
39.     temps = datetime.fromtimestamp(timeserie)
40.     results_perhour .append(perhour)
41.     results_temps .append(temps)
42. # création d'un tableau pour faciliter la manipulation les données de la température
43. df = pd.DataFrame({'Date':(results_temps), 'temperature':(results_perhour)})
44. df.set_index('Date', inplace=True)
45. res = df.resample('s').interpolate().resample('15T').asfreq().dropna()
46. # Identification du système avec ARMAX
47. exog_fcast = res['temperature']
48. index_future_dates=res['temperature'].index
49. #ligne1
50. ARMAXDATAL1 = pd.DataFrame({'puissance11':(data['2020-
01']['PuissancekW_L1']/1000), 'temperature':(data['2020-01']['Tseche'])})
51. SARMAX_L1=ARMAXDATAL1['2020-01']['puissance11']
52. EARMAX_L1=ARMAXDATAL1['2020-01']['temperature']
53. model_ARMAXL1 = ARMA(SARMAX_L1, exog=EARMAX_L1,order=(2, 1))
54. results_ARMAXL1 = model_ARMAXL1.fit()
55. Pred_ARMAXL1 = results_ARMAXL1.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
56. dARMAXL1 =
    pd.DataFrame({'Datefuture':(index_future_dates), 'PuissanceARMAXL1':(Pred_ARMAXL1)})
57. dARMAXL1.set_index('Datefuture', inplace=True)
58. #ligne2
59. ARMAXDATAL2 = pd.DataFrame({'puissance12':(data['2020-
01']['PuissancekW_L2']/1000), 'temperature':(data['2020-01']['Tseche'])})
60. SARMAX_L2=ARMAXDATAL2['2020-01']['puissance12']
61. EARMAX_L2=ARMAXDATAL2['2020-01']['temperature']
62. model_ARMAXL2 = ARMA(SARMAX_L2, exog=EARMAX_L2,order=(2, 1))
63. results_ARMAXL2 = model_ARMAXL2.fit()
64. Pred_ARMAXL2 = results_ARMAXL2.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
65. dARMAXL2 =
    pd.DataFrame({'Datefuture':(index_future_dates), 'PuissanceARMAXL2':(Pred_ARMAXL2)})
66. dARMAXL2.set_index('Datefuture', inplace=True)
67. #ligne3
68. ARMAXDATAL3 = pd.DataFrame({'puissance13':(data['2020-
01']['PuissancekW_L3']/1000), 'temperature':(data['2020-01']['Tseche'])})
69. SARMAX_L3=ARMAXDATAL3['2020-01']['puissance13']
70. EARMAX_L3=ARMAXDATAL3['2020-01']['temperature']
71. model_ARMAXL3 = ARMA(SARMAX_L3, exog=EARMAX_L3,order=(2, 1))
72. results_ARMAXL3 = model_ARMAXL3.fit()
73. Pred_ARMAXL3 = results_ARMAXL3.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
74. dARMAXL3 =
    pd.DataFrame({'Datefuture':(index_future_dates), 'PuissanceARMAXL3':(Pred_ARMAXL3)})
75. dARMAXL3.set_index('Datefuture', inplace=True)
76. #ligne4
77. ARMAXDATAL4 = pd.DataFrame({'puissance14':(data['2020-
01']['PuissancekW_L4']/1000), 'temperature':(data['2020-01']['Tseche'])})
78. SARMAX_L4=ARMAXDATAL4['2020-01']['puissance14']
79. EARMAX_L4=ARMAXDATAL4['2020-01']['temperature']
80. model_ARMAXL4 = ARMA(SARMAX_L4, exog=EARMAX_L4,order=(2, 1))
81. results_ARMAXL4 = model_ARMAXL4.fit()
82. Pred_ARMAXL4 = results_ARMAXL4.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
83. dARMAXL4 =
    pd.DataFrame({'Datefuture':(index_future_dates), 'PuissanceARMAXL4':(Pred_ARMAXL4)})
84. dARMAXL4.set_index('Datefuture', inplace=True)
85. #ligne5
86. ARMAXDATAL5 = pd.DataFrame({'puissance15':(data['2020-
01']['PuissancekW_L5']/1000), 'temperature':(data['2020-01']['Tseche'])})
87. SARMAX_L5=ARMAXDATAL5['2020-01']['puissance15']
88. EARMAX_L5=ARMAXDATAL5['2020-01']['temperature']

```

```

89. model_ARMAXL5 = ARMA(SARMAX_L5, exog=EARMAX_L5,order=(2, 1))
90. results_ARMAXL5 = model_ARMAXL5.fit()
91. Pred_ARMAXL5 = results_ARMAXL5.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
92. dARMAXL5 =
    pd.DataFrame({'Datefuture':(index_future_dates), 'PuissanceARMAXL5':(Pred_ARMAXL5)})
93. dARMAXL5.set_index('Datefuture', inplace=True)
94. #ligne6
95. ARMAXDATAL6 = pd.DataFrame({'puissanceL6':(data['2020-
    01']['PuissancekW_L6']/1000), 'temperature':(data['2020-01']['Tseche'])})
96. SARMAX_L6=ARMAXDATAL6['2020-01']['puissanceL6']
97. EARMAX_L6=ARMAXDATAL6['2020-01']['temperature']
98. model_ARMAXL6 = ARMA(SARMAX_L6, exog=EARMAX_L6,order=(2, 1))
99. results_ARMAXL6 = model_ARMAXL6.fit()
100. Pred_ARMAXL6 = results_ARMAXL6.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
101. dARMAXL6 =
    pd.DataFrame({'Datefuture':(index_future_dates), 'PuissanceARMAXL6':(Pred_ARMAXL6)})
102. dARMAXL6.set_index('Datefuture', inplace=True)
103. #ligne7
104. ARMAXDATAL7 = pd.DataFrame({'puissanceL7':(data['2020-
    01']['PuissancekW_L7']/1000), 'temperature':(data['2020-01']['Tseche'])})
105. SARMAX_L7=ARMAXDATAL7['2020-01']['puissanceL7']
106. EARMAX_L7=ARMAXDATAL7['2020-01']['temperature']
107. model_ARMAXL7 = ARMA(SARMAX_L7, exog=EARMAX_L7,order=(2, 1))
108. results_ARMAXL7 = model_ARMAXL7.fit()
109. Pred_ARMAXL7 = results_ARMAXL7.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
110. dARMAXL7 =
    pd.DataFrame({'Datefuture':(index_future_dates), 'PuissanceARMAXL7':(Pred_ARMAXL7)})
111. dARMAXL7.set_index('Datefuture', inplace=True)
112. #ligne8
113. ARMAXDATAL8 = pd.DataFrame({'puissanceL8':(data['2020-
    01']['PuissancekW_L8']/1000), 'temperature':(data['2020-01']['Tseche'])})
114. SARMAX_L8=ARMAXDATAL8['2020-01']['puissanceL8']
115. EARMAX_L8=ARMAXDATAL8['2020-01']['temperature']
116. model_ARMAXL8 = ARMA(SARMAX_L8, exog=EARMAX_L8,order=(2, 1))
117. results_ARMAXL8 = model_ARMAXL8.fit()
118. Pred_ARMAXL8 = results_ARMAXL8.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
119. dARMAXL8 =
    pd.DataFrame({'Datefuture':(index_future_dates), 'PuissanceARMAXL8':(Pred_ARMAXL8)})
120. dARMAXL8.set_index('Datefuture', inplace=True)
121. #ligneTotal
122. SARMAX_Total=data['2020-01']['PuissancekW_L1']+data['2020-
    01']['PuissancekW_L2']+data['2020-01']['PuissancekW_L3']+data['2020-
    01']['PuissancekW_L4']+data['2020-01']['PuissancekW_L5']+data['2020-
    01']['PuissancekW_L6']+data['2020-01']['PuissancekW_L7']+data['2020-
    01']['PuissancekW_L8']
123. ARMAXDATALT =
    pd.DataFrame({'puissanceLT':(SARMAX_Total/1000), 'temperature':(data['2020-
    01']['Tseche'])})
124. SARMAX_LT=ARMAXDATALT['2020-01']['puissanceLT']
125. EARMAX_LT=ARMAXDATALT['2020-01']['temperature']
126. model_ARMAXLT = ARMA(SARMAX_LT, exog=EARMAX_LT,order=(2, 1))
127. results_ARMAXLT = model_ARMAXLT.fit()
128. Pred_ARMAXLT = results_ARMAXLT.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
129. dARMAXLT =
    pd.DataFrame({'Datefuture':(index_future_dates), 'PuissanceARMAXLT':(Pred_ARMAXLT)})
130. dARMAXLT.set_index('Datefuture', inplace=True)
131. # Identification du système avec ARX
132. exog_fcast = res['temperature']
133. index_future_dates=res['temperature'].index
134. ## ligne 1 ##
135. yL1 = data['2020-01']['PuissancekW_L1']/1000
136. uL1 = data['2020-01']['Tseche']
137. arxL1 = ARX(yL1,uL1, lags=[1, 3, 11])
138. resL1 = arxL1.fit()

```



```

139. forecastsL1 = resL1.forecast(horizon=len(exog_fcast), x=exog_fcast,
    reindex=False).mean.iloc[0]
140. dARXL1 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceARXL1':(forecastsL1)})
141. dARXL1.set_index('Datefuture',inplace=True)
142. ## ligne 2 ##
143. yL2 = data['2020-01']['PuissancekW_L2']/1000
144. uL2 = data['2020-01']['Tseche']
145. arxL2 = ARX(yL2,uL2, lags=[1, 3, 11])
146. resL2 = arxL2.fit()
147. forecastsL2 = resL2.forecast(horizon=len(exog_fcast), x=exog_fcast,
    reindex=False).mean.iloc[0]
148. dARXL2 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceARXL2':(forecastsL2)})
149. dARXL2.set_index('Datefuture',inplace=True)
150. ## ligne 3 ##
151. yL3 = data['2020-01']['PuissancekW_L3']/1000
152. uL3 = data['2020-01']['Tseche']
153. arxL3 = ARX(yL3,uL3, lags=1)
154. resL3 = arxL3.fit()
155. forecastsL3 = resL3.forecast(horizon=len(exog_fcast), x=exog_fcast,
    reindex=False).mean.iloc[0]
156. dARXL3 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceARXL3':(forecastsL3)})
157. dARXL3.set_index('Datefuture',inplace=True)
158. ## ligne 4 ##
159. yL4 = data['2020-01']['PuissancekW_L4']/1000
160. uL4 = data['2020-01']['Tseche']
161. arxL4 = ARX(yL4,uL4, lags=[1, 5, 22])
162. resL4 = arxL4.fit()
163. forecastsL4 = resL4.forecast(horizon=len(exog_fcast), x=exog_fcast,
    reindex=False).mean.iloc[0]
164. dARXL4 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceARXL4':(forecastsL4)})
165. dARXL4.set_index('Datefuture',inplace=True)
166. ## ligne 5 ##
167. yL5 = data['2020-01']['PuissancekW_L5']/1000
168. uL5 = data['2020-01']['Tseche']
169. arxL5 = ARX(yL5,uL5, lags=1)
170. resL5 = arxL5.fit()
171. forecastsL5 = resL5.forecast(horizon=len(exog_fcast), x=exog_fcast,
    reindex=False).mean.iloc[0]
172. dARXL5 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceARXL5':(forecastsL5)})
173. dARXL5.set_index('Datefuture',inplace=True)
174. ## ligne 6 ##
175. yL6 = data['2020-01']['PuissancekW_L6']/1000
176. uL6 = data['2020-01']['Tseche']
177. arxL6 = ARX(yL6,uL6, lags=[1, 5, 12])
178. resL6 = arxL6.fit()
179. forecastsL6 = resL6.forecast(horizon=len(exog_fcast), x=exog_fcast,
    reindex=False).mean.iloc[0]
180. dARXL6 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceARXL6':(forecastsL6)})
181. dARXL6.set_index('Datefuture',inplace=True)
182. ## ligne 7 ##
183. yL7 = data['2020-01']['PuissancekW_L7']/1000
184. uL7 = data['2020-01']['Tseche']
185. arxL7 = ARX(yL7,uL7, lags=[1, 3, 11])
186. resL7 = arxL7.fit()
187. forecastsL7 = resL7.forecast(horizon=len(exog_fcast), x=exog_fcast,
    reindex=False).mean.iloc[0]
188. dARXL7 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceARXL7':(forecastsL7)})
189. dARXL7.set_index('Datefuture',inplace=True)

```

```

190. ## ligne 8 ##
191. yL8 = data['2020-01']['PuissancekW_L8']/1000
192. uL8 = data['2020-01']['Tseche']
193. arxL8 = ARX(yL8,uL8, lags=[1, 3, 11])
194. resL8 = arxL8.fit()
195. forecastsL8 = resL8.forecast(horizon=len(exog_fcast), x=exog_fcast,
    reindex=False).mean.iloc[0]
196. dARXL8 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceARXL8':(forecastsL8)})
197. dARXL8.set_index('Datefuture',inplace=True)
198. ## ligne total ##
199. yLT = data['2020-01']['PuissancekW_L1']+data['2020-
    01']['PuissancekW_L2']+data['2020-01']['PuissancekW_L3']+data['2020-
    01']['PuissancekW_L4']+data['2020-01']['PuissancekW_L5']+data['2020-
    01']['PuissancekW_L6']+data['2020-01']['PuissancekW_L7']+data['2020-
    01']['PuissancekW_L8']
200. yLT=yLT/1000
201. uLT = data['2020-01']['Tseche']
202. arxLT = ARX(yLT,uLT, lags=[1, 3, 11])
203. resLT = arxLT.fit()
204. forecastsLT = resLT.forecast(horizon=len(exog_fcast), x=exog_fcast,
    reindex=False).mean.iloc[0]
205. dARXLT =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceARXLT':(forecastsLT)})
206. dARXLT.set_index('Datefuture',inplace=True)
207. # Identification du système avec Box Jenkins
208. exog_fcast = res['temperature']
209. index_future_dates=res['temperature'].index
210. #ligne1
211. DATAL1 = pd.DataFrame({'puissanceL1':(data['2020-
    01']['PuissancekW_L1']/1000),'temperature':(data['2020-01']['Tseche'])})
212. SBJ_L1=DATAL1['2020-01']['puissanceL1']
213. EBJ_L1=DATAL1['2020-01']['temperature']
214. model_BJL1 = ARIMA(SBJ_L1, exog=EBJ_L1,order=(0, 1, 1))#1,0,1
215. results_BJL1 = model_BJL1.fit()
216. Pred_BJL1 = results_BJL1.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
217. dBJL1 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceBJL1':(Pred_BJL1)})
218. dBJL1.set_index('Datefuture',inplace=True)
219. #ligne2
220. DATAL2 = pd.DataFrame({'puissanceL2':(data['2020-
    01']['PuissancekW_L2']/1000),'temperature':(data['2020-01']['Tseche'])})
221. SBJ_L2=DATAL2['2020-01']['puissanceL2']
222. EBJ_L2=DATAL2['2020-01']['temperature']
223. model_BJL2 = ARIMA(SBJ_L2, exog=EBJ_L2,order=(0, 1, 1))
224. results_BJL2 = model_BJL2.fit()
225. Pred_BJL2 = results_BJL2.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
226. dBJL2 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceBJL2':(Pred_BJL2)})
227. dBJL2.set_index('Datefuture',inplace=True)
228. #ligne3
229. DATAL3 = pd.DataFrame({'puissanceL3':(data['2020-
    01']['PuissancekW_L3']/1000),'temperature':(data['2020-01']['Tseche'])})
230. SBJ_L3=DATAL3['2020-01']['puissanceL3']
231. EBJ_L3=DATAL3['2020-01']['temperature']
232. model_BJL3 = ARIMA(SBJ_L3, exog=EBJ_L3,order=(0, 1, 1))
233. results_BJL3 = model_BJL3.fit()
234. Pred_BJL3 = results_BJL3.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
235. dBJL3 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceBJL3':(Pred_BJL3)})
236. dBJL3.set_index('Datefuture',inplace=True)
237. #ligne4
238. DATAL4 = pd.DataFrame({'puissanceL4':(data['2020-
    01']['PuissancekW_L4']/1000),'temperature':(data['2020-01']['Tseche'])})
239. SBJ_L4=DATAL4['2020-01']['puissanceL4']

```

```

240. EBJ_L4=DATAL4['2020-01']['temperature']
241. model_BJL4 = ARIMA(SBJ_L4, exog=EBJ_L4,order=(0, 1, 1))
242. results_BJL4 = model_BJL4.fit()
243. Pred_BJL4 = results_BJL4.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
244. dBJL4 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceBJL4':(Pred_BJL4)})
245. dBJL4.set_index('Datefuture',inplace=True)
246. #ligne5
247. DATAL5 = pd.DataFrame({'puissanceL5':(data['2020-
    01']['PuissancekW_L5']/1000),'temperature':(data['2020-01']['Tseche'])})
248. SBJ_L5=DATAL5['2020-01']['puissanceL5']
249. EBJ_L5=DATAL5['2020-01']['temperature']
250. model_BJL5 = ARIMA(SBJ_L5, exog=EBJ_L5,order=(0, 1, 1))
251. results_BJL5 = model_BJL5.fit()
252. Pred_BJL5 = results_BJL5.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
253. dBJL5 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceBJL5':(Pred_BJL5)})
254. dBJL5.set_index('Datefuture',inplace=True)
255. #ligne6
256. DATAL6 = pd.DataFrame({'puissanceL6':(data['2020-
    01']['PuissancekW_L6']/1000),'temperature':(data['2020-01']['Tseche'])})
257. SBJ_L6=DATAL6['2020-01']['puissanceL6']
258. EBJ_L6=DATAL6['2020-01']['temperature']
259. model_BJL6 = ARIMA(SBJ_L6, exog=EBJ_L6,order=(0, 1, 1))
260. results_BJL6 = model_BJL6.fit()
261. Pred_BJL6 = results_BJL6.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
262. dBJL6 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceBJL6':(Pred_BJL6)})
263. dBJL6.set_index('Datefuture',inplace=True)
264. #ligne7
265. DATAL7 = pd.DataFrame({'puissanceL7':(data['2020-
    01']['PuissancekW_L7']/1000),'temperature':(data['2020-01']['Tseche'])})
266. SBJ_L7=DATAL7['2020-01']['puissanceL7']
267. EBJ_L7=DATAL7['2020-01']['temperature']
268. model_BJL7 = ARIMA(SBJ_L7, exog=EBJ_L7,order=(0, 1, 1))
269. results_BJL7 = model_BJL7.fit()
270. Pred_BJL7 = results_BJL7.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
271. dBJL7 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceBJL7':(Pred_BJL7)})
272. dBJL7.set_index('Datefuture',inplace=True)
273. #ligne8
274. DATAL8 = pd.DataFrame({'puissanceL8':(data['2020-
    01']['PuissancekW_L8']/1000),'temperature':(data['2020-01']['Tseche'])})
275. SBJ_L8=DATAL8['2020-01']['puissanceL8']
276. EBJ_L8=DATAL8['2020-01']['temperature']
277. model_BJL8 = ARIMA(SBJ_L8, exog=EBJ_L8,order=(0, 1, 1))
278. results_BJL8 = model_BJL8.fit()
279. Pred_BJL8 = results_BJL8.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]
280. dBJL8 =
    pd.DataFrame({'Datefuture':(index_future_dates),'PuissanceBJL8':(Pred_BJL8)})
281. dBJL8.set_index('Datefuture',inplace=True)
282. #ligneTotal
283. SBJ_Total=data['2020-01']['PuissancekW_L1']+data['2020-
    01']['PuissancekW_L2']+data['2020-01']['PuissancekW_L3']+data['2020-
    01']['PuissancekW_L4']+data['2020-01']['PuissancekW_L5']+data['2020-
    01']['PuissancekW_L6']+data['2020-01']['PuissancekW_L7']+data['2020-
    01']['PuissancekW_L8']
284. DATALT = pd.DataFrame({'puissanceLT':(SBJ_Total/1000),'temperature':(data['2020-
    01']['Tseche'])})
285. SBJ_LT=DATALT['2020-01']['puissanceLT']
286. EBJ_LT=DATALT['2020-01']['temperature']
287. model_BJLT = ARIMA(SBJ_LT, exog=EBJ_LT,order=(0, 1, 1))
288. results_BJLT = model_BJLT.fit()
289. Pred_BJLT = results_BJLT.forecast(steps=len(exog_fcast),exog=exog_fcast)[0]

```

```

290. dBJLT =
    pd.DataFrame({'Datefuture':(index_future_dates), 'PuissanceBJLT':(Pred_BJLT)})
291. dBJLT.set_index('Datefuture', inplace=True)
292.
293. #####INTERFACE#####
294.
295. LARGE_FONT= ("Verdana", 12)
296. style.use("ggplot")
297.
298. class App(tk.Tk):
299.
300.     def __init__(self, *args, **kwargs):
301.
302.         tk.Tk.__init__(self, *args, **kwargs)
303.
304.         tk.Tk.iconbitmap(self, "logo2.ico")
305.         tk.Tk.wm_title(self, "Baie-Comeau")
306.         #tk.Tk.wm_config(self, (background='#34629E'))
307.         container = tk.Frame(self)
308.         container.pack(side="top", fill="both", expand = True)
309.         container.grid_rowconfigure(0, weight=1)
310.         container.grid_columnconfigure(0, weight=1)
311.         self.frames = {}
312.         for F in (StartPage, PageOne, PageTwo, PageThree, PageFour, PageFive,
    PageSix, PageSeven ,PageEight,PageNines,PageTen):
313.             frame = F(container, self)
314.             self.frames[F] = frame
315.             frame.grid(row=0, column=0, sticky="nsew")
316.         self.show_frame(StartPage)
317.     def show_frame(self, cont):
318.         frame = self.frames[cont]
319.         frame.tkraise()
320. class StartPage(tk.Frame):
321.     def __init__(self, parent, controller):
322.         tk.Frame.__init__(self, parent)
323.         label = tk.Label(self, text="Accueil", font=LARGE_FONT)
324.         label.pack(pady=10, padx=10)
325.
326.         button = ttk.Button(self, text="Ligne 1",
327.                             command=lambda: controller.show_frame(PageOne))
328.         button.pack()
329.
330.         button2 = ttk.Button(self, text="Ligne 2",
331.                              command=lambda: controller.show_frame(PageTwo))
332.         button2.pack()
333.
334.         button3 = ttk.Button(self, text="Ligne 3",
335.                              command=lambda: controller.show_frame(PageThree))
336.         button3.pack()
337.
338.         button3 = ttk.Button(self, text="Ligne 4",
339.                              command=lambda: controller.show_frame(PageFour))
340.         button3.pack()
341.
342.         button4 = ttk.Button(self, text="Ligne 5",
343.                              command=lambda: controller.show_frame(PageFive))
344.         button4.pack()
345.
346.         button4 = ttk.Button(self, text="Ligne 6",
347.                              command=lambda: controller.show_frame(PageSix))
348.         button4.pack()
349.
350.         button4 = ttk.Button(self, text="Ligne 7",
351.                              command=lambda: controller.show_frame(PageSeven))

```

```

352.         button4.pack()
353.
354.         button4 = ttk.Button(self, text="Ligne 8",
355.                               command=lambda: controller.show_frame(PageEight))
356.         button4.pack()
357.
358.         button4 = ttk.Button(self, text="TOTAL",
359.                               command=lambda: controller.show_frame(PageNines))
360.         button4.pack()
361.
362.         button4 = ttk.Button(self, text="TOUS",
363.                               command=lambda: controller.show_frame(PageTen))
364.         button4.pack()
365.     class PageOne(tk.Frame):
366.     def __init__(self, parent, controller):
367.         tk.Frame.__init__(self, parent)
368.         label = tk.Label(self, text="Ligne 1!!!", font=LARGE_FONT)
369.         label.pack(pady=10, padx=10)
370.         button1 = ttk.Button(self, text=" ← Précédent ",
371.                               command=lambda: controller.show_frame(StartPage))
372.         button1.pack()
373.         f = Figure(figsize=(5,5), dpi=100)
374.         a = f.add_subplot(211)
375.         a.plot(dARMAXL1['PuissanceARMAXL1'], label="Prévision ARMAX")
376.         a.plot(dARXL1['PuissanceARXL1'], label="Prévision ARX")
377.         a.plot(dBJL1['PuissanceBJL1'], label="Prévision Box-Jenkins")
378.         a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3, ncol=2, borderaxespad=0)
379.         a.autoscale(enable=True, axis='x', tight=True)
380.         b = f.add_subplot(212)
381.         myFmt = mdates.DateFormatter('%H:%M')
382.         a.xaxis.set_major_formatter(myFmt)
383.         a.set_ylabel('Power (MW)', fontsize=14)
384.         b.plot(res['temperature'])
385.         b.set_ylabel('Température (°C)', fontsize=14)
386.         b.set_xlabel('Temps (HH)', fontsize=14)
387.         b.xaxis.set_major_formatter(myFmt)
388.         b.autoscale(enable=True, axis='x', tight=True)
389.         canvas = FigureCanvasTkAgg(f, self)
390.         canvas.draw()
391.         canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)
392.         toolbar = NavigationToolbar2TkAgg(canvas, self)
393.         toolbar.update()
394.         canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
395.     class PageTwo(tk.Frame):
396.     def __init__(self, parent, controller):
397.         tk.Frame.__init__(self, parent)
398.         label = tk.Label(self, text="Ligne 2!!!", font=LARGE_FONT)
399.         label.pack(pady=10, padx=10)
400.
401.         button1 = ttk.Button(self, text=" ← Précédent ",
402.                               command=lambda: controller.show_frame(StartPage))
403.         button1.pack()
404.         f = Figure(figsize=(5,5), dpi=100)
405.         a = f.add_subplot(211)
406.         a.plot(dARMAXL2['PuissanceARMAXL2'], label="Prévision ARMAX")
407.         a.plot(dARXL2['PuissanceARXL2'], label="Prévision ARX")
408.         a.plot(dBJL2['PuissanceBJL2'], label="Prévision Box-Jenkins")
409.         #a.invert_yaxis()
410.         a.set_ylabel('Power (MW)', fontsize=12)
411.         a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3, ncol=2, borderaxespad=0)
412.         b = f.add_subplot(212)
413.         b.plot(res['temperature'])
414.         b.set_ylabel('Température (°C)', fontsize=12)
415.         b.set_xlabel('Temps (HH)', fontsize=12)
416.         myFmt = mdates.DateFormatter('%H:%M')

```

```

417.         a.xaxis.set_major_formatter(myFmt)
418.         b.xaxis.set_major_formatter(myFmt)
419.         a.autoscale(enable=True, axis='x', tight=True)
420.         b.autoscale(enable=True, axis='x', tight=True)
421.         canvas = FigureCanvasTkAgg(f, self)
422.         canvas.draw()
423.         canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)
424.         toolbar = NavigationToolbar2TkAgg(canvas, self)
425.         toolbar.update()
426.         canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
427.     class PageThree(tk.Frame):
428.     def __init__(self, parent, controller):
429.         tk.Frame.__init__(self, parent)
430.         label = tk.Label(self, text="Ligne 3!!!", font=LARGE_FONT)
431.         label.pack(pady=10, padx=10)
432.         button1 = ttk.Button(self, text=" ← Précédent ",
433.                               command=lambda: controller.show_frame(StartPage))
434.         button1.pack()
435.         f = Figure(figsize=(5,5), dpi=100)
436.         a = f.add_subplot(211)
437.         a.plot(dARMAXL3['PuissanceARMAXL3'], label="Prévision ARMAX")
438.         a.plot(dARXL3['PuissanceARXL3'], label="Prévision ARX")
439.         a.plot(dBJL3['PuissanceBJL3'], label="Prévision Box-Jenkins")
440.         #a.invert_yaxis()
441.         a.set_ylabel('Puissance (MW)', fontsize=12)
442.         a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3, ncol=2, borderaxespad=0)
443.         b = f.add_subplot(212)
444.         b.plot(res['temperature'])
445.         b.set_ylabel('Température (°C)', fontsize=12)
446.         b.set_xlabel('Temps (HH)', fontsize=12)
447.         myFmt = mdates.DateFormatter('%H:%M')
448.         a.xaxis.set_major_formatter(myFmt)
449.         b.xaxis.set_major_formatter(myFmt)
450.         a.autoscale(enable=True, axis='x', tight=True)
451.         b.autoscale(enable=True, axis='x', tight=True)
452.         canvas = FigureCanvasTkAgg(f, self)
453.         canvas.draw()
454.         canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)
455.         toolbar = NavigationToolbar2TkAgg(canvas, self)
456.         toolbar.update()
457.         canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
458.     class PageFour(tk.Frame):
459.     def __init__(self, parent, controller):
460.         tk.Frame.__init__(self, parent)
461.         label = tk.Label(self, text="Ligne 4!!!", font=LARGE_FONT)
462.         label.pack(pady=10, padx=10)
463.         button1 = ttk.Button(self, text=" ← Précédent ",
464.                               command=lambda: controller.show_frame(StartPage))
465.         button1.pack()
466.         f = Figure(figsize=(5,5), dpi=100)
467.         a = f.add_subplot(211)
468.         a.plot(dARMAXL4['PuissanceARMAXL4'], label="Prévision ARMAX")
469.         a.plot(dARXL4['PuissanceARXL4'], label="Prévision ARX")
470.         a.plot(dBJL4['PuissanceBJL4'], label="Prévision Box-Jenkins")
471.         #a.invert_yaxis()
472.         a.set_ylabel('Puissance (MW)', fontsize=12)
473.         a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3, ncol=2, borderaxespad=0)
474.         b = f.add_subplot(212)
475.         b.plot(res['temperature'])
476.         b.set_ylabel('Température (°C)', fontsize=12)
477.         b.set_xlabel('Temps (HH)', fontsize=12)
478.         myFmt = mdates.DateFormatter('%H:%M')
479.         a.xaxis.set_major_formatter(myFmt)
480.         b.xaxis.set_major_formatter(myFmt)
481.         a.autoscale(enable=True, axis='x', tight=True)

```



```

482.         b.autoscale(enable=True, axis='x', tight=True)
483.         canvas = FigureCanvasTkAgg(f, self)
484.         canvas.draw()
485.         canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)
486.
487.         toolbar = NavigationToolbar2TkAgg(canvas, self)
488.         toolbar.update()
489.         canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
490.     class PageFive(tk.Frame):
491.     def __init__(self, parent, controller):
492.         tk.Frame.__init__(self, parent)
493.         label = tk.Label(self, text="Ligne 5!!!", font=LARGE_FONT)
494.         label.pack(pady=10, padx=10)
495.         button1 = ttk.Button(self, text=" ← Précédent ",
496.                               command=lambda: controller.show_frame(StartPage))
497.         button1.pack()
498.         f = Figure(figsize=(5,5), dpi=100)
499.         a = f.add_subplot(211)
500.         a.plot(dARMAXL5['PuissanceARMAXL5'], label="Prévision ARMAX")
501.         a.plot(dARXL5['PuissanceARXL5'], label="Prévision ARX")
502.         a.plot(dBJL5['PuissanceBJL5'], label="Prévision Box-Jenkins")
503.         a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3, ncol=2, borderaxespad=0)
504.         #a.invert_yaxis()
505.         a.set_ylabel('Puissance (MW)', fontsize=12)
506.         b = f.add_subplot(212)
507.         b.plot(res['temperature'])
508.         b.set_ylabel('Température (°C)', fontsize=12)
509.         b.set_xlabel('Temps (HH)', fontsize=12)
510.         myFmt = mdates.DateFormatter('%H:%M')
511.         a.xaxis.set_major_formatter(myFmt)
512.         b.xaxis.set_major_formatter(myFmt)
513.         a.autoscale(enable=True, axis='x', tight=True)
514.         b.autoscale(enable=True, axis='x', tight=True)
515.         canvas = FigureCanvasTkAgg(f, self)
516.         canvas.draw()
517.         canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)
518.
519.         toolbar = NavigationToolbar2TkAgg(canvas, self)
520.         toolbar.update()
521.         canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
522.
523.     class PageSix(tk.Frame):
524.
525.     def __init__(self, parent, controller):
526.         tk.Frame.__init__(self, parent)
527.         label = tk.Label(self, text="Ligne 6!!!", font=LARGE_FONT)
528.         label.pack(pady=10, padx=10)
529.
530.         button1 = ttk.Button(self, text=" ← Précédent ",
531.                               command=lambda: controller.show_frame(StartPage))
532.         button1.pack()
533.         f = Figure(figsize=(5,5), dpi=100)
534.         a = f.add_subplot(211)
535.         a.plot(dARMAXL6['PuissanceARMAXL6'], label="Prévision ARMAX")
536.         a.plot(dARXL6['PuissanceARXL6'], label="Prévision ARX")
537.         a.plot(dBJL6['PuissanceBJL6'], label="Prévision Box-Jenkins")
538.         a.set_ylabel('Puissance (MW)', fontsize=12)
539.         a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3, ncol=2, borderaxespad=0)
540.         b = f.add_subplot(212)
541.         b.plot(res['temperature'])
542.         b.set_ylabel('Température (°C)', fontsize=12)
543.         b.set_xlabel('Temps (HH)', fontsize=12)
544.         myFmt = mdates.DateFormatter('%H:%M')
545.         a.xaxis.set_major_formatter(myFmt)
546.         b.xaxis.set_major_formatter(myFmt)

```

```

547.         a.autoscale(enable=True, axis='x', tight=True)
548.         b.autoscale(enable=True, axis='x', tight=True)
549.         canvas = FigureCanvasTkAgg(f, self)
550.         canvas.draw()
551.         canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)
552.
553.         toolbar = NavigationToolbar2TkAgg(canvas, self)
554.         toolbar.update()
555.         canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
556.     class PageSeven(tk.Frame):
557.         def __init__(self, parent, controller):
558.             tk.Frame.__init__(self, parent)
559.             label = tk.Label(self, text="Ligne 7!!!", font=LARGE_FONT)
560.             label.pack(pady=10, padx=10)
561.             button1 = ttk.Button(self, text=" ← Précédent ",
562.                                 command=lambda: controller.show_frame(StartPage))
563.             button1.pack()
564.             f = Figure(figsize=(5,5), dpi=100)
565.             a = f.add_subplot(211)
566.             a.plot(dARMAXL7['PuissanceARMAXL7'], label="Prévision ARMAX")
567.             a.plot(dARXL7['PuissanceARXL7'], label="Prévision ARX")
568.             a.plot(dBJL7['PuissanceBJL7'], label="Prévision Box-Jenkins")
569.             a.set_ylabel('Puissance (MW)', fontsize=12)
570.             a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3, ncol=2, borderaxespad=0)
571.             b = f.add_subplot(212)
572.             b.plot(res['temperature'])
573.             b.set_ylabel('Température (°C)', fontsize=12)
574.             b.set_xlabel('Temps (HH)', fontsize=12)
575.             myFmt = mdates.DateFormatter('%H:%M')
576.             a.xaxis.set_major_formatter(myFmt)
577.             b.xaxis.set_major_formatter(myFmt)
578.             a.autoscale(enable=True, axis='x', tight=True)
579.             b.autoscale(enable=True, axis='x', tight=True)
580.             canvas = FigureCanvasTkAgg(f, self)
581.             canvas.draw()
582.             canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)
583.             toolbar = NavigationToolbar2TkAgg(canvas, self)
584.             toolbar.update()
585.             canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
586.     class PageEight(tk.Frame):
587.
588.         def __init__(self, parent, controller):
589.             tk.Frame.__init__(self, parent)
590.             label = tk.Label(self, text="Ligne 8!!!", font=LARGE_FONT)
591.             label.pack(pady=10, padx=10)
592.
593.             button1 = ttk.Button(self, text=" ← Précédent ",
594.                                 command=lambda: controller.show_frame(StartPage))
595.             button1.pack()
596.             f = Figure(figsize=(5,5), dpi=100)
597.             a = f.add_subplot(211)
598.             a.plot(dARMAXL8['PuissanceARMAXL8'], label="Prévision ARMAX")
599.             a.plot(dARXL8['PuissanceARXL8'], label="Prévision ARX")
600.             a.plot(dBJL8['PuissanceBJL8'], label="Prévision Box-Jenkins")
601.             a.set_ylabel('Puissance (MW)', fontsize=12)
602.             a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3, ncol=2, borderaxespad=0)
603.             b = f.add_subplot(212)
604.             b.plot(res['temperature'])
605.             b.set_ylabel('Température (°C)', fontsize=12)
606.             b.set_xlabel('Temps (HH)', fontsize=12)
607.             myFmt = mdates.DateFormatter('%H:%M')
608.             a.xaxis.set_major_formatter(myFmt)
609.             b.xaxis.set_major_formatter(myFmt)
610.             a.autoscale(enable=True, axis='x', tight=True)
611.             b.autoscale(enable=True, axis='x', tight=True)

```



```

612.         canvas = FigureCanvasTkAgg(f, self)
613.         canvas.draw()
614.         canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)
615.         toolbar = NavigationToolbar2TkAgg(canvas, self)
616.         toolbar.update()
617.         canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
618.     class PageNines(tk.Frame):
619.
620.     def __init__(self, parent, controller):
621.         tk.Frame.__init__(self, parent)
622.         label = tk.Label(self, text="Total !!!", font=LARGE_FONT)
623.         label.pack(pady=10, padx=10)
624.         button1 = ttk.Button(self, text=" ← Précédent ",
625.                               command=lambda: controller.show_frame(StartPage))
626.         button1.pack()
627.         f = Figure(figsize=(5,5), dpi=100)
628.         a = f.add_subplot(211)
629.         a.plot(dARMAXLT['PuissanceARMAXLT'], label="Prévision ARMAX")
630.         a.plot(dARXL['PuissanceARXL'], label="Prévision ARX")
631.         a.plot(dBJLT['PuissanceBJLT'], label="Prévision Box-Jenkins")
632.         #a.invert_yaxis()
633.         a.set_ylabel('Puissance (MW)', fontsize=12)
634.         a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3, ncol=2, borderaxespad=0)
635.         b = f.add_subplot(212)
636.         b.plot(res['temperature'])
637.         b.set_ylabel('Température (°C)', fontsize=12)
638.         b.set_xlabel('Temps (HH)', fontsize=12)
639.         myFmt = mdates.DateFormatter('%H:%M')
640.         a.xaxis.set_major_formatter(myFmt)
641.         b.xaxis.set_major_formatter(myFmt)
642.         a.autoscale(enable=True, axis='x', tight=True)
643.         b.autoscale(enable=True, axis='x', tight=True)
644.         canvas = FigureCanvasTkAgg(f, self)
645.         canvas.draw()
646.         canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)
647.         toolbar = NavigationToolbar2TkAgg(canvas, self)
648.         toolbar.update()
649.         canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
650.     class PageTen(tk.Frame):
651.     def __init__(self, parent, controller):
652.         tk.Frame.__init__(self, parent)
653.         label = tk.Label(self, text="Tous les Lignes !!!", font=LARGE_FONT)
654.         label.pack(pady=10, padx=10)
655.
656.         button1 = ttk.Button(self, text=" ← Précédent ",
657.                               command=lambda: controller.show_frame(StartPage))
658.         button1.pack()
659.         f = Figure(figsize=(20,9), dpi=80)
660.         a = f.add_subplot(331)
661.         a.plot(dARMAXL1['PuissanceARMAXL1'], label="Prévision ARMAX")
662.         a.plot(dARXL1['PuissanceARXL1'], label="Prévision ARX")
663.         a.plot(dBJL1['PuissanceBJL1'], label="Prévision Box-Jenkins")
664.         a.set_ylabel('Puissance (MW)', fontsize=12)
665.         a.legend(bbox_to_anchor=(0, 1.02, 1, .102), loc=3, ncol=2, borderaxespad=0)
666.         b = f.add_subplot(332)
667.         b.plot(dARMAXL2['PuissanceARMAXL2'], label="Prévision ARMAX")
668.         b.plot(dARXL2['PuissanceARXL2'], label="Prévision ARX")
669.         b.plot(dBJL2['PuissanceBJL2'], label="Prévision Box-Jenkins")
670.         b.set_title('Ligne 2', fontsize=8)
671.         c = f.add_subplot(333)
672.         c.plot(dARMAXL3['PuissanceARMAXL3'], label="Prévision ARMAX")
673.         c.plot(dARXL3['PuissanceARXL3'], label="Prévision ARX")
674.         c.plot(dBJL3['PuissanceBJL3'], label="Prévision Box-Jenkins")
675.         c.set_title('Ligne 3', fontsize=8)
676.         e = f.add_subplot(334)

```

```

677.         e.plot(dARMAXL4['PuissanceARMAXL4'], label="Prévision ARMAX")
678.         e.plot(dARXL4['PuissanceARXL4'], label="Prévision ARX")
679.         e.plot(dBJL4['PuissanceBJL4'], label="Prévision Box-Jenkins")
680.         e.set_ylabel('Puissance (MW)', fontsize=12)
681.         e.set_title('Ligne 4', fontsize=8)
682.         j = f.add_subplot(335)
683.         j.plot(dARMAXL5['PuissanceARMAXL5'], label="Prévision ARMAX")
684.         j.plot(dARXL5['PuissanceARXL5'], label="Prévision ARX")
685.         j.plot(dBJL5['PuissanceBJL5'], label="Prévision Box-Jenkins")
686.         j.set_title('Ligne 5', fontsize=8)
687.         k= f.add_subplot(336)
688.         k.plot(dARMAXL6['PuissanceARMAXL6'], label="Prévision ARMAX")
689.         k.plot(dARXL6['PuissanceARXL6'], label="Prévision ARX")
690.         k.plot(dBJL6['PuissanceBJL6'], label="Prévision Box-Jenkins")
691.         k.set_title('Ligne 6', fontsize=8)
692.         l= f.add_subplot(337)
693.         l.plot(dARMAXL7['PuissanceARMAXL7'], label="Prévision ARMAX")
694.         l.plot(dARXL7['PuissanceARXL7'], label="Prévision ARX")
695.         l.plot(dBJL7['PuissanceBJL7'], label="Prévision Box-Jenkins")
696.         l.set_ylabel('Puissance (MW)', fontsize=12)
697.         l.set_xlabel('Temps (HH)', fontsize=12)
698.         l.set_title('Ligne 7', fontsize=8)
699.         u= f.add_subplot(338)
700.         u.plot(dARMAXL8['PuissanceARMAXL8'], label="Prévision ARMAX")
701.         u.plot(dARXL8['PuissanceARXL8'], label="Prévision ARX")
702.         u.plot(dBJL8['PuissanceBJL8'], label="Prévision Box-Jenkins")
703.         u.set_xlabel('Temps (HH)', fontsize=12)
704.         u.set_title('Ligne 8', fontsize=8)
705.         d = f.add_subplot(339)
706.         d.plot(res['temperature'])
707.         d.set_ylabel('Température (°C)', fontsize=12)
708.         d.set_xlabel('Temps (HH)', fontsize=12)
709.         myFmt = mdates.DateFormatter('%H')
710.         a.xaxis.set_major_formatter(myFmt)
711.         b.xaxis.set_major_formatter(myFmt)
712.         c.xaxis.set_major_formatter(myFmt)
713.         e.xaxis.set_major_formatter(myFmt)
714.         j.xaxis.set_major_formatter(myFmt)
715.         k.xaxis.set_major_formatter(myFmt)
716.         l.xaxis.set_major_formatter(myFmt)
717.         u.xaxis.set_major_formatter(myFmt)
718.         d.xaxis.set_major_formatter(myFmt)
719.         a.autoscale(enable=True, axis='x', tight=True)
720.         b.autoscale(enable=True, axis='x', tight=True)
721.         c.autoscale(enable=True, axis='x', tight=True)
722.         e.autoscale(enable=True, axis='x', tight=True)
723.         j.autoscale(enable=True, axis='x', tight=True)
724.         k.autoscale(enable=True, axis='x', tight=True)
725.         l.autoscale(enable=True, axis='x', tight=True)
726.         u.autoscale(enable=True, axis='x', tight=True)
727.         d.autoscale(enable=True, axis='x', tight=True)
728.         canvas = FigureCanvasTkAgg(f, self)
729.         canvas.draw()
730.         canvas.get_tk_widget().pack(side=tk.BOTTOM, fill=tk.BOTH, expand=True)
731.         toolbar = NavigationToolbar2TkAgg(canvas, self)
732.         toolbar.update()
733.         canvas._tkcanvas.pack(side=tk.TOP, fill=tk.BOTH, expand=True)
734.     app = App()
735.     app.geometry("1080x480")#
736.     app.mainloop()
737.
738.

```

