PolyPublie
Polytechnique Montréal

POLYTECHNIQUE MONTRÉAL
UNIVERSITÉ D'INGÉNIERIE

Titre: Title:	Détermination des dates de livraison en VMI à partir de données transactionnelles avec une approche de prévision quantile
Auteur: Author:	Félix Tremblay
Date:	2020
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Tremblay, F. (2020). Détermination des dates de livraison en VMI à partir de données transactionnelles avec une approche de prévision quantile [Mémoire de maîtrise, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/5455/

Document en libre accès dans PolyPublie
Open Access document in PolyPublie

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/5455/
Directeurs de recherche: Advisors:	Bruno Agard
Programme: Program:	Maîtrise recherche en génie industriel

Ce fichier a été téléchargé à partir de PolyPublie, le dépôt institutionnel de Polytechnique Montréal
This file has been downloaded from PolyPublie, the institutional repository of Polytechnique Montréal

https://publications.polymtl.ca

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Détermination des dates de livraison en VMI à partir de données transactionnelles avec une approche de prévision quantile

FÉLIX TREMBLAY

Département de mathématiques et de génie industriel

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

Génie industriel

Août 2020

© Félix Tremblay, 2020.

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

Détermination des dates de livraison en VMI à partir de données transactionnelles avec une approche de prévision quantile

présenté par **Félix TREMBLAY**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*

a été dûment accepté par le jury d'examen constitué de :

Robert PELLERIN, président
Bruno AGARD, membre et directeur de recherche
Michel GAMACHE, membre

REMERCIEMENTS

Avant d’entamer ce rapport, je souhaite exprimer ma reconnaissance envers plusieurs personnes qui ont, de près ou de loin, contribué à la réalisation de ce travail.

Tout d’abord, j’aimerais remercier Bruno Agard, mon directeur de recherche, pour ses judicieux conseils d’expert qui m’ont aiguillé.

Je tiens à remercier mon partenaire industriel et particulièrement le directeur de la chaine d’approvisionnement, Robert Grégoire, pour leur confiance et leur implication.

Je remercie également mes amis du Laboratoire en Intelligence des Données (LID), dans lequel j’ai tant apprécié l’esprit de fraternité et d’entraide.

Enfin, j’aimerais remercier ma famille et ma copine, pour leur soutien et leurs encouragements.

RÉSUMÉ

Dans un modèle de gestion des stocks par le fournisseur, ou « *Vendor Managed Inventory* » (VMI), les réserves de produits sont la responsabilité du fournisseur. Celui-ci gère les commandes de réapprovisionnement pour maintenir la disponibilité de l'inventaire et fait appel à des méthodes de prévision de la demande. Lorsque les données de demande ou d'inventaire du client ne sont pas disponibles, elles peuvent être substituées par les données transactionnelles. Les séries temporelles alors obtenues sont fortement bruitées et intermittentes. En conséquence, il est difficile pour le fournisseur de planifier les dates de réapprovisionnement en gardant un contrôle sur la probabilité de rupture de stock. Une planification optimale minimise les coûts de transport tout en respectant les contraintes de niveaux de service. Le niveau de service est défini comme étant la probabilité de ne pas avoir de rupture de stock avant un réapprovisionnement.

Ce projet a pour but de développer une méthode de prévision pour calculer les dates de livraisons en respectant le niveau de service et en utilisant l'historique de transactions comme unique source de données.

La revue de littérature présente des méthodes de prévision classiques ainsi que des méthodes de prévision probabilistes. Les méthodes de prévision de la littérature ne permettent pas d'exploiter directement les données transactionnelles; c'est pourquoi une nouvelle méthode doit être développée. Pour calculer le moment optimal de réapprovisionnement qui respecte un niveau de service visé, une approche stochastique (implicite ou explicite) est nécessaire. La solution traditionnelle est de calculer un stock de sécurité en modélisant les données avec une loi normale; cependant, il a été montré que cette méthode entraîne une sous-performance des niveaux de service, c'est-à-dire que les ruptures de stock surviennent plus souvent qu'attendu. La prévision quantile est un type de prévision probabiliste qui est parfaitement adapté à l'objectif. Le moment de réapprovisionnement optimal sera calculé en effectuant une prévision quantile de l'intervalle de temps menant à une rupture de stock, pour un quantile égal à 1 – le niveau de service visé. La méthodologie couvre tout le flux de données : préparation des données, choix de la méthode et évaluation de la meilleure méthode. Le cas d'étude met en application la méthodologie dans le contexte d'une logistique de collecte. Avec le modèle qui a été développé, on pourrait réduire les coûts en transport sans pénaliser le niveau de service. Si l'on vise un niveau de service de 90%

(légèrement meilleur que l'état actuel de 89%), on pourrait réduire le nombre annuel de collectes de 21%.

ABSTRACT

In vendor managed inventory (VMI) models, the supplier is responsible for inventory management and must use strategic forecasting in order to ensure the availability of products for customers. When product consumption data or inventory data are not available, transactional data can be used. Time series obtained using this substitution are noisy and intermittent. As a result, it is difficult for the supplier to plan delivery orders while respecting service level agreement (SLA). Optimal planning minimizes transportation costs while respecting service level constraints. The service level is defined as the probability of not hitting a stock-out before the next delivery.

The goal of this project is to develop a forecasting method able to optimize delivery dates while respecting the service level and using only delivery records data.

The literature review presents classical forecasting methods as well as probabilistic forecasting methods. Forecasting methods in the literature are not appropriate for transactional data; therefore, a new method must be developed. To optimize delivery dates while reaching target service levels, a stochastic approach is necessary. The traditional solution is to calculate a safety stock by fitting the data with a normal distribution; however, it has been shown that this method underachieves the target service levels (i.e., stock-outs occur more often than expected). Quantile forecast is a type of probabilistic forecast that is perfectly suited to our objective. The optimal delivery date is calculated by estimating a quantile of the time interval leading to a stock-out, for a quantile equal to 1 - the target service level. The methodology covers the entire data flow: data preparation, method selection and evaluation of the selected method. The case study applies the methodology in the context of reverse logistics. With the model that has been developed, we could reduce transportation costs without penalizing service level. If we target a service level of 90% (compared to the current state of 89%), we could reduce the annual number of collects by 21%.

TABLE DES MATIÈRES

REMERCIEMENTS III

RÉSUMÉ IV

ABSTRACT VI

TABLE DES MATIÈRES VII

LISTE DES FIGURES X

LISTE DES SIGLES ET ABRÉVIATIONS XI

CHAPITRE 1 INTRODUCTION 1

CHAPITRE 2 REVUE DE LA LITTÉRATURE 4

2.1 Prévision classique 4

2.1.1 Méthodes de prévision générales 4

2.1.2 Méthodes de prévision avec demande intermittente 5

2.1.3 Métriques d'erreur 7

2.2 Prévision probabiliste 9

2.2.1 Méthodes de prévision de distribution 10

2.2.2 Méthodes de prévision de quantile 11

2.2.3 Métriques d'erreur 14

2.2.4 Validation croisée 15

2.3 Revue critique 16

CHAPITRE 3 MÉTHODOLOGIE 18

3.1 Préparer les données 19

3.1.1 Extraire les données transactionnelles nécessaires 19

3.1.2 Enrichir les données 20

3.1.3 Détecter et traiter les données aberrantes 20

3.1.4 Retirer des données tests 21
3.2 Sélectionner la meilleure méthode 21
3.2.1 Tester les méthodes de prévision classique 21
3.2.2 Tester les méthodes de prévision quantile 26
3.3 Évaluer la meilleure méthode 33
3.3.1 Mesurer la précision des prévisions 33
3.3.2 Tracer les courbes de rendement 34
CHAPITRE 4 CAS D'ÉTUDE 38
4.1 Préparer les données 39
4.1.1 Extraire les données transactionnelles nécessaires 39
4.1.2 Enrichir les données 43
4.1.3 Détecter et traiter les données aberrantes 44
4.1.4 Retirer les données tests 46
4.2 Sélectionner la meilleure méthode 47
4.2.1 Tester les méthodes de prévision classique 47
4.2.2 Tester les méthodes de prévision quantile 54
4.3 Évaluer la meilleure méthode 60
4.3.1 Mesurer la précision des prévisions 60
4.3.2 Tracer les courbes de rendement 61
4.4 Synthèse 66
CHAPITRE 5 CONCLUSION ET RECOMMANDATIONS 68
RÉFÉRENCES 72

LISTE DES TABLEAUX

Tableau 3.1 Données transactionnelles nécessaires 19

Tableau 3.2 Enrichissement des données 20

Tableau 3.3 Résultats des prévisions probabilistes d'intervalles pour un client 35

Tableau 4.1 Données transactionnelles nécessaires en logistique de collecte 39

Tableau 4.2 Enrichissement des données en logistique de collecte 44

Tableau 4.3 Résultats pour la moyenne mobile (stratégie #1) 48

Tableau 4.4 Résultats pour le lissage exponentiel simple (stratégie #1) 48

Tableau 4.5 Résultats pour la moyenne mobile (stratégie #2) 49

Tableau 4.6 Résultats pour le lissage exponentiel simple (stratégie #2) 49

Tableau 4.7 Résultats pour Croston 50

Tableau 4.8 Résultats pour ASACT + Moyenne mobile 51

Tableau 4.9 Résultats pour ASACT + Lissage exponentiel simple 53

Tableau 4.10 Synthèse des résultats pour les méthodes classiques 54

Tableau 4.11 Résultats pour l'approche en deux étapes 55

Tableau 4.12 Résultats pour Bootstrap 1 et Bootstrap 2 55

Tableau 4.13 Quantiles observés pour Bootstrap 1 et Bootstrap 2 56

Tableau 4.14 Résultats pour Bootstrap 3 57

Tableau 4.15 Quantiles observés pour Bootstrap 3 57

Tableau 4.16 Valeurs optimales du coefficient C 58

Tableau 4.17 Résultats pour Bootstrap 4 58

Tableau 4.18 Synthèse des résultats pour les méthodes quantiles 59

Tableau 4.19 Synthèse des quantiles observés pour les méthodes quantiles 60

Tableau 4.20 Résultats et quantiles observés avec Bootstrap 4 sur des données futures 60

LISTE DES FIGURES

Figure 3.1 Étapes de la méthodologie 18

Figure 3.2 Création d'une série temporelle de demandes journalières 23

Figure 4.1 Section du code SQL représentant le choix de la source de données pour la date de réalisation d'une commande 40

Figure 4.2 Section du code SQL représentant les jointures des différentes sources de données 40

Figure 4.3 Section du code SQL montrant le calcul de la quantité collectée en Kg 41

Figure 4.4 Section du code SQL montrant l'identification des commandes de type RA 41

Figure 4.5 Section du code SQL montrant le calcul de la quantité planifiée 41

Figure 4.6 Section du code SQL montrant le calcul de la quantité collectée 41

Figure 4.7 Section du code SQL montrant le choix parmi les différentes sources pour les quantités collectées 42

Figure 4.8 Section du code SQL montrant l'identification des codes de raison de type « Cancel » au niveau de la commande et au niveau de l'action 42

Figure 4.9 Section du code SQL montrant la logique pour choisir la valeur du code de raison 43

Figure 4.10 Section du code SQL montrant la condition qui retire les commandes fantômes 43

Figure 4.11 Section du code SQL montrant le calcul de l'utilisation et de l'intervalle 44

Figure 4.12 Nombre annuel de collectes en fonction du niveau de service visé 62

Figure 4.13 Niveau de service obtenu en fonction du niveau de service visé 64

Figure 4.14 « Stock avant collecte » moyen en fonction du niveau de service visé 65

Figure 4.15 « Stock avant collecte (en %) » moyen en fonction du niveau de service visé 65

LISTE DES SIGLES ET ABRÉVIATIONS

ADIDA	Aggregate-Disaggregate Intermittent Demand Approach
ARIMA	Autoregressive Integrated Moving Average
ASACT	Aggregate, Smooth, Aggregate, Convert to Time-series
BMA	Bayesian Model Averaging
CRPS	Continuous Ranked Probability Score
EMOS	Ensemble Model Output Statistics
EPS	Ensemble Prediction System
ERP	Enterprise Resource Planning (progiciel de gestion intégrée)
GARCH	Generalized AutoRegressive Conditional Heteroskedasticity
LTD	Lead Time Demand
MAE	Mean Absolute Error (moyenne des erreurs absolues)
MAPA	Multi Aggregation Prediction Algorithm
MAPE	Mean Absolute Percentage Error (moyenne des erreurs absolues en pourcentage)
MC	Monte-Carlo
MSE	Mean Squared Error (moyenne des erreurs aux carrés)
RMSE	Root Mean Squared Error (racine carrée de la moyenne des erreurs aux carrés)
QRF	Quantile Regression Forests
SBA	Syntetos–Boylan Approximation
SLA	Service Level Agreement
SREF	Short-Range Ensemble Forecasting
TL	Tick Loss function
TSB	Teunter–Syntetos–Babai
VMI	Vendor Managed Inventory (gestion des stocks par le fournisseur)

CHAPITRE 1 INTRODUCTION

Dans une chaîne logistique VMI (Vendor Managed Inventory), c'est le fournisseur qui gère les stocks. Il est responsable de la disponibilité des produits et donc de la génération des commandes de réapprovisionnement. Pour s'assurer que les clients ne tombent pas en rupture de stock, tout en minimisant ses frais d'exploitation, le fournisseur peut intégrer des méthodes de prévision stratégique dans sa planification. La prévision de la demande est importante pour décider : 1) quand servir un client donné et 2) quelle quantité livrer.

D'abord, la prévision de la demande est nécessaire pour le choix des fenêtres de temps des commandes de réapprovisionnement. La fenêtre de temps, composée de deux bornes, sert de contraintes dans le problème d'optimisation des tournées de véhicules. Un paramétrage approprié permet de réduire les coûts de transport et d'éviter les ruptures de stock : 1) augmenter la durée de la fenêtre donne plus de flexibilité pour le routage et peut donc diminuer les coûts unitaires moyens des commandes, 2) augmenter la borne inférieure ou supérieure permet de réduire la quantité annuelle de réapprovisionnement, cependant, 3) augmenter la borne supérieure augmente les risques de ruptures de stock; c'est pourquoi elle est généralement calculée en fonction d'un stock de sécurité prédéterminée.

Ensuite, la prévision de la demande est nécessaire pour l'assignation des commandes dans l'optimisation des tournées de véhicules. Quelle que soit la stratégie de chargement du fournisseur (camions pleins ou non), il faut que la quantité totale estimée corresponde approximativement à la somme des besoins réels des clients pour assurer que les tournées planifiées soient réalisables.

Souvent, le fournisseur n'a pas accès aux données d'inventaire ou d'utilisation du client. Lorsque les informations sur le point de consommation sont absentes, l'information disponible en amont doit être utilisée. Les données transactionnelles (quantités livrées et dates) peuvent alors remplacer les données de demandes manquantes. En conséquence de cette substitution, les séries observées ont tendance à être bruyantes, irrégulières, intermittentes et donc difficiles à utiliser pour les prévisions (Ferbar Tratar, 2015; Murray et al., 2017; Petropoulos et al., 2016).

Pour un fournisseur en logistique VMI, la planification des commandes de réapprovisionnement est un problème majeur à résoudre. L'objectif principal de ce projet portera donc sur la détermination de la borne supérieure de la fenêtre de temps des commandes de

réapprovisionnement, tout en visant à satisfaire un certain niveau de service. Plus particulièrement, on s'intéressera au cas où les données de demande et d'inventaire du client ne sont pas disponibles. Bien qu'il existe plusieurs façons de définir le niveau de service, la définition adoptée pour cette étude est la probabilité de ne pas avoir de rupture de stock avant un réapprovisionnement.

La nature des données disponibles est un obstacle important à surmonter. Le fournisseur doit effectuer ses prévisions à partir uniquement des données transactionnelles passées. Avec l'historique des transactions, on connaît les dates des visites passées, la quantité livrée ainsi que le niveau du stock après (ou avant) livraison. L'information qu'on peut en tirer est une agrégation (la somme) de la demande réelle du client entre deux transactions. En conséquence, les comportements ponctuels ou cycliques, comme la saisonnalité ou les effets calendaires, peuvent être cachés ou brouillés, ce qui nuit aux prévisions. À moins d'avoir une très grande quantité de données transactionnelles, on ne peut pas tirer avantage des méthodes de prévisions complexes (pouvant traiter la saisonnalité et les effets calendaires) puisque celles-ci, plus sensibles aux bruits, causent des problèmes de suradaptation. De plus, les données utiles qu'on tire des données transactionnelles (dates, durées d'intervalle et demandes accumulées pendant les intervalles) ne sont pas directement exploitables par les méthodes de prévisions de la littérature.

Les méthodes, qui sont examinées dans la première partie de la revue de littérature (section 2.1), permettent d'estimer la demande avec des modèles déterministes. Ces approches, qui ne tiennent pas compte de l'incertitude, sont aveugles quant aux risques de rupture de stock et ne permettent donc pas un choix judicieux des fenêtres de temps. En effet, pour déterminer la borne maximale de la fenêtre de réapprovisionnement qui satisfait les exigences de fiabilité, il faut une approche stochastique. Les prévisions probabilistes (section 2.2) peuvent donner une caractérisation plus complète de la variable prédite à l'aide d'une distribution de probabilité des estimations futures.

L'objectif principal de ce mémoire est de développer une méthode pour calculer à quel moment au plus tard réapprovisionner un client, en respectant des contraintes sur la probabilité de rupture de stock et en utilisant l'historique de transactions comme source de données. La méthodologie qui sera proposée permettra aussi de :

1. Prédire un certain quantile de la quantité utilisée pendant un certain intervalle de temps après le dernier approvisionnement. (Par exemple, le 50^{e} centile représente la quantité vraisemblable et le 95^{e} centile représente un scénario extrême);

2. Calculer l’intervalle de temps maximal qui respecte un certain niveau de service (c.-à-d., qui respecte une certaine contrainte sur la probabilité de rupture de stock).

Pour atteindre ces objectifs, les méthodes utiles sont présentées dans la revue de littérature (Chapitre 2). Une méthodologie sera proposée (Chapitre 3), puis mise en application avec des données industrielles dans le cas d’étude (Chapitre 4), ce qui permettra de conclure (Chapitre 5).

CHAPITRE 2 REVUE DE LA LITTÉRATURE

Dans ce chapitre seront présentés les domaines de la littérature les plus utiles pour cette étude. Premièrement, la section Prévision classique (2.1) regroupe les méthodes générales de prévision, les méthodes de prévision avec demande intermittente, ainsi que les métriques d'erreur permettant de les évaluer. Deuxièmement, la section Prévision probabiliste (2.2) traitera des méthodes de prévision de distribution, des méthodes de prévision de quantile et des métriques d'erreur permettant de les évaluer.

2.1 Prévision classique

2.1.1 Méthodes de prévision générales

Les méthodes générales de prévisions, telles que la moyenne mobile et le lissage exponentiel, sont souvent utilisées pour le contrôle d'inventaire (Ferbar Tratar, 2015; Murray et al., 2017; Petropoulos et al., 2016). La simplicité de ces méthodes aide à éviter des problèmes de suradaptation, surtout lorsque très peu de données sont disponibles. En gestion d'inventaire, les compromis entre stocks de sécurité et niveaux de service montrent que les méthodes paramétriques simples fonctionnent bien (Syntetos et al., 2015).

Tout d'abord, il y a la **moyenne mobile**. Avec cette méthode, la prévision (ŷ) de la prochaine période (t+1) est obtenue en calculant la moyenne des (w) dernières données réelles (y) (Brown, 1959).

$$\hat{y}_{t+1} = \frac{y_t + y_{t-1} + \ldots + y_{t-w+1}}{w} \quad (1)$$

Comme on peut le voir, cette méthode traite avec la même pondération toutes les (w) dernières données et ignore complètement les données précédentes. On peut donc dire qu'il y a une transition brusque.

Si l'on souhaite diminuer la pondération des données de façon graduelle, il y a la méthode du **lissage exponentiel simple**. Avec cette méthode, le coefficient de lissage α (entre 0 et 1) contrôle la pondération. Plus on réduit la valeur du coefficient α, plus on accorde de valeur relative aux données récentes. Par exemple, avec une valeur standard de 0.2, plus la donnée est récente, plus elle a de valeur relative. Les prévisions (ŷ) sont obtenues récursivement. Pour démarrer, on suppose

que la prévision (ŷ) est égale à la demande réelle (y). Ensuite, la prévision de la période courante (t) est utilisée dans le calcul de la prévision de la prochaine période (t + 1) (Brown, 1959).

$$\hat{y}_{t+1} = \alpha\, y_t + (1-\alpha)\, \hat{y}_t \qquad (2)$$

Des variations de ces méthodes, comme la moyenne mobile avec pondération exponentielle, permettent de modéliser la tendance et une composante saisonnière (Holt, 1957; Winters, 1960). Lorsque la demande est enregistrée plusieurs fois par jour, plusieurs cycles de saisonnalité apparaissent : un cycle journalier (qui explique la ressemblance, d'une journée à l'autre, du profil de la demande) et un cycle hebdomadaire (qui explique la ressemblance, d'une semaine à l'autre, du profil de la demande) (Taylor, 2003). Le modèle de lissage exponentiel de Holt-Winters peut être adapté pour accommoder ces deux saisonnalités (Taylor, 2003). Un autre modèle permet d'effectuer des prévisions de la demande lorsque les séries temporelles subissent des effets calendaires telles que des périodes de grand solde ou des fêtes annuelles (Bouchard & Montreuil, 2011). Plus avancé et souvent utilisé en prévision, le modèle ARIMA est très général et cherche à s'ajuster aux données ainsi qu'à atténuer le bruit (Box & Jenkins, 1970). ARIMAX est une extension qui permet d'inclure des variables exogènes (Madsen, 2007). Lorsque la variance des erreurs n'est pas constante, par exemple lorsqu'elle est plus élevée pour les périodes de fortes demandes, on dit que les erreurs de la série temporelle sont hétéroscédastiques (Bollerslev, 1986). Le modèle GARCH fait l'hypothèse que la variance des erreurs suit un modèle de moyenne mobile autorégressive (Bollerslev, 1986).

2.1.2 Méthodes de prévision avec demande intermittente

Les méthodes de cette section considèrent le problème de l'intermittence de la demande. Les séries temporelles intermittentes, caractérisées par une grande proportion de périodes de demande nulle, sont généralement classifiées avec deux critères : la variabilité des demandes non nulles et la variabilité des intervalles inter-demande (Syntetos et al., 2005). Les séries temporelles dérivées de données transactionnelles sont bruyantes (Ferbar Tratar, 2015; Murray et al., 2017), c'est pourquoi la prévision d'une demande intermittente est un problème qui est difficile à résoudre (Kourentzes, 2014).

La méthode de **Croston** est la première traitant ce type de demande (Croston, 1972). Elle nécessite d'abord de décomposer les séries temporelles en deux séries: l'une formée des demandes non nulles (y) et l'autre des intervalles inter-demandes (i). Ces dernières sont ensuite extrapolées en utilisant

le lissage exponentiel simple et on obtient deux prévisions indépendantes (ŷ et î). Généralement, le même coefficient de lissage α (entre 0 et 1) est appliqué sur les deux séries et on démarre en supposant que la prévision est égale à la donnée réelle. On utilise récursivement:

$$\hat{y}_{t+1} = \alpha\, y_t + (1-\alpha)\, \hat{y}_t \quad (3)$$

$$\hat{\imath}_{t+1} = \alpha\, i_t + (1-\alpha)\, \hat{\imath}_t \quad (4)$$

On peut ensuite diviser les deux prévisions (ŷ / î) pour obtenir une prévision du taux de la demande. Finalement, on peut multiplier le taux prédit par l'intervalle voulu pour obtenir une prévision de la demande pendant cet intervalle.

La méthode de Croston demeure le standard pour les prévisions avec demande intermittente (Ghobbar & Friend, 2003; Prestwich et al., 2014), bien que plusieurs variantes et améliorations aient été proposées. L'approximation de Syntetos-Boylan (SBA) aborde un problème de biais (Syntetos & Boylan, 2005), alors que la méthode de Teunter–Syntetos–Babai (TSB) permet de gérer les problèmes d'obsolescence de produits, ce qui n'était pas possible avec Croston (Teunter et al., 2011).

Une méthode alternative, ADIDA (*Aggregate-Disaggregate Intermittent Demand Approach*), applique une agrégation temporelle sur la série, visant à réduire la variance des intervalles inter-demande et ensuite désagrège la série au niveau initial (Nikolopoulos et al., 2011). Par conséquent, cette méthode est adéquate avec les séries intermittentes dont les intervalles ont une grande variance (Petropoulos et al., 2016). Lorsque le niveau d'agrégation est approprié, la série transformée n'est plus intermittente et les méthodes de prévision de séries normales peuvent être appliquées. Une variante, ADIDA inverse (ou iADIDA), permet de réduire la variance des demandes en agrégeant sur celles-ci plutôt que sur le temps (Petropoulos et al., 2016).

Une approche similaire, MAPA (*Multi Aggregation Prediction Algorithm*), tire avantage à la fois de l'agrégation temporelle et de la combinaison de prévision (Kourentzes et al., 2014). L'application de différents niveaux d'agrégations temporelles permet de mieux capturer les composants (tendance, composante saisonnière et bruit) de la série. Aux premiers niveaux (haute fréquence), les composants périodiques sont importants, puis aux derniers niveaux (basse fréquence), la tendance apparait clairement.

Avec des données collectées en amont du point de consommation, tel que des données transactionnelles, il a été montré que la méthode **ASACT** (*Aggregate, Smooth, Aggregate, Convert to Time-series*) est efficace pour supprimer le bruit tout en préservant les comportements sous-jacents (Murray et al., 2018). L'algorithme ASACT est en trois étapes :

1. **Agréger**. Pour chaque client, les données de livraison sont d'abord agrégées avec un bac temporel dont la taille est minimale. La taille du bac est choisie de façon qu'il n'y ait pas plus d'une transaction par période, empêchant ainsi toute addition de quantités dans la première agrégation. Puisque les fréquences de livraison d'un client sont variables, cette étape crée des séries intermittentes. À partir des données de transactions, on construit un vecteur de livraisons par client. Les coordonnées du vecteur représentent la taille du bac sélectionnée. Par exemple, une taille de bac de 1 jour donne un vecteur de 365 jours pour un an de données. Pour chaque coordonnée, les données sont le volume total de produits livrés ce jour;
2. **Lisser.** En appliquant la méthode de Croston, les quantités sont réparties sur la période suivant une transaction. Cette étape élimine l'intermittence et lisse les données;
3. **Agréger.** La série lissée est ensuite agrégée au niveau désiré pour l'analyse.

2.1.3 Métriques d'erreur

La prévision classique, par rapport à la prévision probabiliste, a l'avantage de produire des résultats qui sont non seulement plus faciles à interpréter, mais dont les métriques d'erreurs sont plus intuitives. Les principales métriques sont présentées dans cette section.

La moyenne des erreurs aux carrés, ou « *Mean squared error* » (MSE), est simple à calculer, mais n'a pas la même échelle que la demande (à cause du carré), ce qui n'est pas idéal en matière d'interprétation. D'un point de vue stochastique, un modèle de prévision qui minimise la MSE est un prédicteur de la *moyenne* de la distribution (Gneiting, 2011). Une autre propriété est qu'elle est très impactée par les erreurs élevées, donc sensibles aux valeurs aberrantes. Ceci est un inconvénient majeur lorsque les données utilisées sont bruitées, comme c'est le cas avec les données transactionnelles (Murray et al., 2018). Lorsqu'on a un tableau de n lignes, avec sur chaque ligne une prévision ($\hat{y}$) et une observation (y), la MSE peut être calculée en deux étapes.

Premièrement, on calcule la colonne de l'erreur au carré : $(\hat{y} - y)^2$. Deuxièmement, on calcule la moyenne des valeurs de cette colonne.

$$MSE = \frac{1}{n}\sum_{t=1}^{n} (\hat{y}_t - y_t)^2 \quad (5)$$

La racine carrée du MSE, ou « *Root Mean squared error* » (RMSE), a l'avantage d'être plus facile à interpréter que la MSE puisque l'unité de l'erreur est la même que celle de la demande. La RMSE est aussi sensible aux valeurs extrêmes que la MSE. Pour obtenir la valeur du RMSE, il faut calculer la MSE, puis y appliquer une racine carrée.

$$RMSE = \sqrt{MSE} \quad (6)$$

La moyenne des erreurs absolues, ou « *Mean absolute error »* (**MAE**), est plus robuste que la MSE ou RMSE en présence de données aberrantes, puisqu'elle est moins affectée par les plus grosses erreurs. Tout comme la RMSE, l'unité de l'erreur est la même que l'unité de la demande. Un modèle de prévision qui minimise la MAE est un prédicteur de la *médiane* de la distribution (Gneiting, 2011). La MAE est très facile à calculer et à interpréter. Lorsqu'on a un tableau de n lignes, avec sur chaque ligne une prévision ($\hat{y}$) et une observation (y), la MAE peut être calculée en deux étapes. Premièrement, on calcule la colonne de l'erreur absolue : $|\hat{y} - y|$. Deuxièmement, on calcule la moyenne des valeurs de cette colonne.

$$MAE = \frac{1}{n}\sum_{t=1}^{n} |\hat{y}_t - y_t| \quad (7)$$

La moyenne des erreurs absolues en pourcentage, ou « *Mean absolute percentage error »* (MAPE), est souvent utilisée en prévision. MAPE donne plus d'importance aux erreurs lorsque l'observation (y) est faible, c'est pourquoi cette métrique comporte un problème de biais (Kim & Kim, 2016). En effet, un modèle de prévision qui minimise la MAPE est un prédicteur qui sous-estime la moyenne et la médiane de la distribution (Kim & Kim, 2016). Lorsqu'on a un tableau de n lignes, avec sur chaque ligne une prévision ($\hat{y}$) et une observation (y), la MAPE peut être calculée en deux

étapes. Premièrement, on calcule la colonne de l'erreur absolue en pourcentage : |ŷ - y| / y. Deuxièmement, on calcule la moyenne des valeurs de cette colonne.

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}\frac{|\hat{y}_t - y_t|}{y_t} \quad (8)$$

Somme toute, la MAE est une métrique idéale en prévision classique lorsque les séries utilisées proviennent de données transactionnelles.

2.2 Prévision probabiliste

Les prévisions probabilistes peuvent en dire beaucoup plus sur la demande future. Plutôt que de prédire un seul scénario (qui est censé être le plus vraisemblable), on peut considérer plusieurs scénarios futurs et leur attribuer des probabilités. La prévision probabiliste est particulièrement intéressante lorsque : 1) une valeur centrale unique n'est pas suffisante, 2) une seule statistique, telle que la variance, est inadéquate pour décrire l'erreur, ce qui est le cas lorsque les erreurs sont asymétriques ou 3) les erreurs sont hétéroscédastiques, c'est-à-dire qu'elles varient en fonction des caractéristiques d'entrée.

Le résultat attendu d'une prévision probabiliste peut se présenter sous différentes formes. Dans certains cas, il est possible de prédire une distribution statistique théorique de la demande (par exemple une loi normale, bêta, triangulaire ou n'importe quelle fonction de répartition). Cette approche permet de connaître la distribution prédictive entière. Lorsqu'on s'intéresse uniquement à un (ou des) quantile(s) particulier(s) de cette distribution prédictive, il est préférable d'utiliser la prévision quantile (parfois appelée régression quantile). Prédire la distribution entière comporte certains inconvénients. Premièrement, il est plus difficile de comparer différents algorithmes de prévisions, puisqu'il n'est pas évident de voir à quels quantiles les distributions prédictives diffèrent (Boylan & Syntetos, 2006). Deuxièmement, les métriques d'erreurs basées sur la distribution entière peuvent être trompeuses (Boylan & Syntetos, 2006). Par exemple, un algorithme peut prédire une distribution qui est, de façon générale, près de la vraie distribution, mais donner de très mauvais résultats pour certains quantiles qui nous intéressent particulièrement (Trapero et al., 2019). Pour ces raisons, il est recommandé de se concentrer uniquement sur les

quantiles qui sont liés au niveau de service, soit ceux entre 85% et 99% (Boylan & Syntetos, 2006; Trapero et al., 2019).

2.2.1 Méthodes de prévision de distribution

Plusieurs méthodes statistiques existent pour générer des distributions prédictives. La prévision d'ensemble, qui est la plus commune d'entre elles, est incontournable en prévision météorologique (Gneiting et al., 2005)

Pour produire une prévision d'ensemble, il faut générer une multitude de prévisions déterministes, à l'aide de différents modèles et de différentes conditions initiales (Gneiting, 2008). Le principe est donc similaire à la méthode de Monte-Carlo. Cette dernière est détaillée dans la section suivante.

Parmi les méthodes de génération d'ensembles, il y a la méthode d'élevage, ou « *breeding method* » (Toth & Kalnay, 1997), la méthode « *Ensemble Prediction System* » (EPS) (Molteni et al., 1996), la méthode des observations perturbées (Houtekamer et al., 1996) et la méthode « *short-range ensemble forecasting* » (SREF) (Frederick, 2003; Grimit & Mass, 2002). Les systèmes d'ensembles sont finis et comprennent un nombre limité de membres, c'est pourquoi ils sont avant tout utilisés pour la prévision d'événements discrets (Gneiting et al., 2005). De ce fait, pour générer des fonctions de répartition ou des fonctions de densité, il faut avoir recours à des post-traitements statistiques (Hamill & Colucci, 1998; Richardson, 2001).

Parmi les techniques de post-traitement, les plus connues sont la moyenne du modèle bayésien, ou « *bayesian model averaging* » (BMA) (Raftery et al., 2005) et les statistiques de sortie du modèle d'ensemble, ou « *ensemble model output statistics* » (EMOS) (Gneiting et al., 2005). Plus récemment, une méthode basée sur les forêts de régression quantile, ou « *quantile regression forests* » (QRF) a été développée (Taillardat et al., 2016).

BMA est un mécanisme de sélection de modèles qui est basé sur l'inférence bayésienne. Cette méthode, en tenant compte de l'incertitude associée au choix d'un modèle, procure un critère de choix simple et cohérent (Fragoso et al., 2018). L'idée de base est que dans tout ensemble de prévisions déterministes il existe un « meilleur modèle », mais nous ne savons pas lequel. BMA permet de quantifier l'incertitude associée à la détermination du « meilleur modèle ». Il génère une distribution prédictive finale (p) par une moyenne pondérée des (K) distributions prédictives

intermédiaires (h_k). $h_k(y|f_k)$ est la fonction de densité conditionnelle pour une demande (y) sachant (f_k), c'est-à-dire conditionnellement à ce que (f_k), soit le « meilleur modèle ». Les moyennes des distributions prédictives intermédiaires sont obtenues à l'aide des prévisions déterministes initiales, alors que les variances sont obtenues à l'aide des biais des prévisions déterministes initiales. Les pondérations (w_k), quant à elles, sont obtenues de façon itérative lors de multiples périodes d'entraînement : à la fin d'un entraînement, les poids sont mis à jour en fonction de la contribution relative des distributions prédictives intermédiaires à la compétence prédictive de la distribution prédictive finale (Sloughter et al., 2007). La fonction de densité de la distribution prédictive finale (p) est : (Sloughter et al., 2007)

$$p(y|f_1, \dots, f_k) = \sum_{k=1}^{K} w_k h_k(y|f_k) \tag{9}$$

EMOS est une technique générant des distributions prédictives dont la fonction de densité est paramétrique et dont les paramètres dépendent des valeurs de l'ensemble (Taillardat et al., 2016). La technique EMOS originale produit une gaussienne. La moyenne d'EMOS est calculée par moyenne pondérée des prévisions déterministes initiales, où la pondération est déterminée selon la contribution relative et où le biais est corrigé. La variance d'EMOS est une fonction linéaire de la variance des prévisions déterministes initiales où la dispersion est corrigée. Les coefficients sont ajustés de façon optimale en minimisant la métrique d'erreur CRPS (Gneiting et al., 2005). Cette métrique est détaillée dans la section 2.2.3.

2.2.2 Méthodes de prévision de quantile

En prévision quantile, le résultat est un quantile de la distribution prédictive. Il existe plusieurs méthodes. Parmi elles, il y a la régression quantile (Koenker & Bassett, 1978), la forêt de régressions quantiles (Meinshausen, 2006; Taillardat et al., 2016), la méthode de Monte-Carlo (MC) (Metropolis et al., 1953; Metropolis & Ulam, 1949) et le bootstrap (Efron, 1992).

La régression quantile peut être vue comme une extension de la régression linéaire classique. Alors que cette dernière permet d'estimer la moyenne conditionnelle de la variable réponse (en minimisant la somme des erreurs au carré), la régression quantile permet d'estimer n'importe quel quantile conditionnel de la variable réponse. Pour obtenir la médiane conditionnelle, on minimise

la somme des erreurs absolues. Pour obtenir les autres quantiles conditionnels, on minimise la somme pondérée asymétriquement des erreurs absolues (Koenker & Hallock, 2001). (La formule de cette pondération asymétrique est la même que celle de l'indicateur asymétrique de la section 2.2.3.) La régression quantile est considérée comme un estimateur robuste, c'est-à-dire qu'elle a la capacité de ne pas être perturbée par la présence de valeurs aberrantes dans les données (John, 2015). Par ailleurs, elle peut aussi servir pour la détection de valeurs aberrantes pour des variables à dimension multiples. Une nouvelle observation serait considérée comme une valeur aberrante si sa valeur est extrême selon la fonction de distribution conditionnelle prévue (Barnett & Lewis, 1994; Hodge & Austin, 2004; Meinshausen, 2006).

La forêt de régressions quantiles, ou « *quantile regression forests* » (QRF) (Meinshausen, 2006) est une généralisation de la forêt d'arbres décisionnels (« *random forest* ») (Breiman, 2001). C'est une méthode non paramétrique, robuste et non linéaire d'estimation de quantiles conditionnels (Meinshausen, 2006; Taillardat et al., 2016). Alors que la forêt d'arbres décisionnels estime la moyenne conditionnelle, la forêt de régression quantile estime n'importe quel quantile conditionnel (donc possiblement la distribution conditionnelle complète). Tout comme la forêt d'arbres décisionnels, la forêt de régressions quantiles est un ensemble d'arbres binaires. Cela dit, pour tous les derniers nœuds de chacun des arbres, on ne calcule pas une moyenne de la variable réponse, mais un quantile de celle-ci. Enfin, en calculant la moyenne du quantile conditionnel de chaque arbre, on obtient le quantile conditionnel final.

La méthode de Monte-Carlo est une famille de techniques permettant d'approximer une variable réponse (ou sa distribution) en utilisant des processus stochastiques. En utilisant des échantillons aléatoires répétés, on peut effectuer des prévisions quantiles. Voici des étapes générales : 1) on définit la fonction de répartition de chacune des (M) variables d'entrées, 2) on génère aléatoirement (N fois) toutes les variables d'entrées en suivant leur fonction de répartition respective et on enregistre les valeurs dans un tableau de N lignes et M colonnes, 3) pour chaque ligne, on calcule une variable réponse déterministe et on enregistre les valeurs dans un vecteur de taille N, 4) on calcule un quantile du vecteur N.

Le **bootstrap** est une famille de méthodes d'inférence statistique qui est similaire à la méthode de Monte-Carlo. Toutefois, les échantillons sont générés à partir de données réelles, plutôt que d'une distribution théorique. Pour les problèmes de prévision avec données intermittente, la technique du

bootstrap est largement répandue (Zhou & Viswanathan, 2011). Spécifiquement, les méthodes de la littérature sont des versions modifiées, puisqu'une application naïve du bootstrap, à un problème de prévision avec données intermittentes, ne considère pas l'autocorrélation et ne peut pas prédire de demande autre que celles présentent dans l'historique (Willemain et al., 2004). Voici les étapes de la méthode « VZ_adj » (Hasni et al., 2019), qui est une amélioration de la méthode « VZ » (Viswanathan & Zhou, 2008) :

0. Produire un histogramme pour les valeurs des demandes non nulles et un autre histogramme pour les valeurs des intervalles inter-demande;
1. Initialiser **H=1** (variable pour la progression du temps) et générer une première demande (selon l'histogramme des demandes);
2. Générer un intervalle inter-demande (**iid**) (selon l'histogramme des intervalles);
3. Mettre à jour le temps (**H = H + iid**) :
    a. Si **H ≤ L + 1**, générer une nouvelle demande (selon l'histogramme des demandes) (où L est la durée du « *Lead Time* ») et retourner à l'étape 2;
    b. Sinon, sommer les demandes pour obtenir un **LTD** (Le LTD, ou « *Lead Time Demand* » représente la demande totale pendant la durée L, ou « *Lead Time* »);
4. Après 1000 répétitions des étapes 2 et 3, générer la distribution des LTD.

La technique du bootstrap, en utilisant des échantillons, peut avoir l'inconvénient de générer des distributions dont la variance est sous-estimée, en particulier lorsque la taille de l'échantillon de départ est faible. En effet, il est connu que le maximum parmi n échantillons indépendants, en moyenne, ne représente que le centile n/(n+1) de la population (Mosteller & Rourke, 1973). Une solution est l'ajout de variation aléatoire (« *jittering* ») aux demandes (Willemain et al., 2004). Une transformation ($D_{jittered}$) de la demande (D) peut être calculée de la manière suivante : (Willemain et al., 2004)

$$D_{jittered} = MAX\left(0,\ D\ +\ Z\sqrt{D}\right) \tag{10}$$

où Z ~ N(0, 1)

Le processus de *jittering* vise à faire varier davantage les plus grandes demandes, ce qui correspond au phénomène observé empiriquement, soit que la variance augmente généralement lorsque la moyenne de la demande augmente (Willemain et al., 2004).

2.2.3 Métriques d'erreur

Les métriques telles que MSE, RMSE, MAE ou MAPE ne sont pas applicables aux prévisions probabilistes. Ces indicateurs s'appuient sur le fait qu'en prévision classique, la valeur prédite est simplement censée être le plus près possible de la valeur réelle et donc qu'une erreur de 0 est une situation idéale. En prévision probabiliste, le principe est différent, c'est-à-dire que l'objectif n'est pas d'estimer la valeur réelle, mais sa distribution statistique (ou un quantile de cette distribution). En pratique, il est impossible d'observer cette distribution, puisqu'une seule valeur est observée. En d'autres mots, la prévision est probabiliste, mais l'observation est déterministe.

En prévision quantile, lorsqu'on s'intéresse au 90^{e} centile, on voudrait qu'approximativement 90% de nos prévisions soient supérieures aux observations associées. Avec un nombre suffisant de tests, on peut effectivement vérifier si les prévisions sont « *correctement biaisées* ». Pour adéquatement évaluer ces prévisions, il faut introduire un biais dans la métrique d'erreur.

La métrique d'erreur appropriée pour évaluer une prévision quantile est une version asymétrique de la MAE connue sous les noms de « ***Tick Loss*** », « *tick loss function* », « *asymmetric piecewise linear loss* », « *linlin* », « *hinge* », « *pinball* » ou « *newsvendor loss* » (Gneiting, 2011; Trapero et al., 2019). Pour le quantile q (entre 0 et 1), l'erreur « *tick loss* » (TL_q) pour l'observation (y) et la prévision quantile ($\hat{y}_q$) se calcule avec:

$$TL_q = \begin{cases} q\left(y - \hat{y}_q\right) & si\ y \geq \hat{y}_q \\ (1-q)\left(\hat{y}_q - y\right) & sinon \end{cases} \tag{11}$$

Pour q=0.5, la formule revient à prendre la valeur absolue de l'erreur. Alors qu'un modèle de prévision qui minimise la MAE cherche à prédire la médiane, un modèle qui minimise la moyenne du « *Tick Loss* » (TL_q) cherche à prédire le q-quantile (Raiffa & Schlaifer, 1961). Cette propriété est cruciale à la prévision quantile (Gneiting, 2011; Koenker & Bassett Jr, 1978).

L'asymétrie de la fonction reflète le quantile choisi. Au 50^{e} centile, la fonction est symétrique ce qui veut dire qu'une sous-estimation est pénalisée autant qu'une surestimation. Pour le 80^{e} centile, par exemple, une sous-estimation est pénalisée 4 fois plus qu'une surestimation (0.8/(1-0.8)=4). En principe, un modèle qui minimise la moyenne des erreurs $TL_{q=0.8}$ va surestimer 80% du temps.

En prévision de distribution, les distributions prédictives sont jugées par leur netteté (« *sharpness* ») ainsi que leur étalonnage (« *calibration* »). La netteté est la concentration de la distribution prédictive alors que l'étalonnage est la comparabilité statistique de la distribution prédictive avec les observations, c'est-à-dire que les observations devraient être indiscernables de tirages au sort de la distribution prédictive (Gneiting & Katzfuss, 2014). La métrique d'erreur « *Continuous Ranked Probability Score* » (CRPS) permet d'évaluer simultanément ces deux propriétés et est définie par : (Gneiting & Katzfuss, 2014)

$$CRPS(F, y) = \int_{-\infty}^{\infty} (F(x) - \mathbb{1}\{y \leq x\})^2 \, dx \tag{12}$$

Où y est l'observation, F est la fonction de répartition de la distribution prédictive et $\mathbb{1}$ est la fonction de Heaviside (« *Heaviside step function* ») définie par :

$$\mathbb{1}(t) = \begin{cases} 1 & si\ t \geq 0 \\ 0 & sinon \end{cases} \tag{13}$$

2.2.4 Validation croisée

En prévision, pour comparer la performance de différents modèles, on a recours aux métriques d'erreur (telles que MAE et TL_q) et on cherche naturellement à minimiser celles-ci. Cependant, il faut être prudent avec la méthodologie de validation employée pour obtenir une estimation fidèle de l'erreur et pour éviter des problèmes de surajustement. Répondant à ces besoins, la validation croisée consiste à séparer les données d'apprentissage des données tests (Bergmeir & Benítez, 2012). Les données d'apprentissage permettent d'entraîner le modèle et d'effectuer des prévisions. L'erreur est ensuite calculée en comparant les données prédites aux données tests. La technique de la validation croisée apparaitra dans la méthodologie pour l'évaluation des méthodes classiques, pour l'évaluation et le choix des méthodes probabilistes ainsi que pour l'évaluation finale de la méthode sélectionnée.

2.3 Revue critique

Il existe diverses méthodes de prévision classique et probabiliste. En prévision classique, la revue de littérature a présenté des méthodes générales, des méthodes spécifiques à la demande intermittente, ainsi que des métriques d'erreurs permettant de les évaluer. Celle qui est le plus appropriée au problème est la MAE. En prévision probabiliste, des méthodes de prévision de distribution, des méthodes de prévision de quantile, ainsi que des métriques d'erreurs permettant de les évaluer ont été présentées. La prévision quantile, puisqu'elle se focalise sur un quantile en particulier, permet d'éviter plusieurs inconvénients de la prévision de distribution. C'est pourquoi il est recommandé de se limiter aux quantiles pertinents, soit ceux qui correspondent au niveau de service visé (typiquement, entre 85% et 99%).

Cependant, les méthodes de prévisions présentées ne s'appliquent pas directement à la problématique étudiée.

Premièrement, avec les méthodes classiques, les valeurs prédites sont déterministes et représentent, en principe, le scénario le plus vraisemblable. On doit donc s'attendre à ce que la demande réelle soit supérieure à la demande prédite à peu près 50% du temps (ce qui correspondrait à un niveau de service de 50%). Puisque les compagnies visent typiquement des niveaux de service entre 85% et 99% (Trapero et al., 2019), une prévision de la tendance centrale de la demande n'aide pas réellement à résoudre la problématique. Pour être en mesure de calculer le moment optimal de réapprovisionnement qui procure une certaine probabilité de rupture de stock (en fonction du niveau de service établie), une approche stochastique est préférable à une approche déterministe. Néanmoins, l'approche qui est généralement utilisée pour le contrôle d'inventaire est d'effectuer une prévision déterministe de la demande et d'y ajouter un stock de sécurité.

La solution traditionnelle en gestion d'inventaire est de calculer un « stock de sécurité » en modélisant les données avec une loi normale (Giard, 2003). Le « stock de sécurité » vise à procurer un certain niveau de service. Le problème avec cette approche est l'hypothèse que les erreurs de prévision suivent une distribution normale. Cette hypothèse risque de ne pas s'avérer adéquate avec des données industrielles pouvant présenter des valeurs aberrantes et du bruit causé par l'effet « *bullwhip* » (Murray et al., 2018). En effet, il a été montré que cette méthode « naïve » entraîne une sous-performance des niveaux de service (Trapero et al., 2019). Autrement dit, les ruptures de stock surviennent plus souvent qu'attendu.

Deuxièmement, les méthodes probabilistes de la littérature ne permettent pas non plus de répondre adéquatement à la problématique. Les méthodes se rapprochant le plus de notre problématique sont celles de la prévision probabiliste d'une demande intermittente (Viswanathan & Zhou, 2008; Hasni et al., 2019). Cependant, la problématique qui y est traitée est différente de la nôtre. En effet, on n'est pas réellement en présence d'une demande intermittente, mais d'une demande inconnue (pouvant être intermittente) agrégée de façon intermittente. Dans la littérature de la demande intermittente, la demande réelle est connue à chaque période et, pour plusieurs de ces périodes, cette demande est nulle. Dans le contexte de la problématique, la demande réelle à chaque période est inconnue alors que la demande cumulative entre deux transactions est connue et intermittente. Cette distinction devra être prise en compte.

Dans la littérature de la prévision avec demande intermittente, on traite généralement les demandes non nulles indépendamment des intervalles inter-demande. Pour notre problématique, néanmoins, on s'attend à une corrélation positive entre ces deux variables : plus l'intervalle inter-visite est long, plus la demande inter-visite devrait augmenter. Si ce n'était pas le cas, le fournisseur pourrait repousser ses réapprovisionnements indéfiniment.

Dans la méthodologie seront proposées des méthodes adaptées à la problématique. Contrairement à celles de la littérature en demande intermittente, nos méthodes (nommées Bootstrap 2, Bootstrap 3 et Bootstrap 4) considèrent la nature particulière des données disponibles, ainsi que la corrélation attendue entre les demandes et les intervalles. Puisqu'elles sont de type stochastique, elles permettent de considérer explicitement et directement le niveau de service visé. En effet, la contrainte de la satisfaction d'un niveau de service (ce que le stock de sécurité tente indirectement et inefficacement d'atteindre) est prise directement en compte lorsqu'on effectue une prévision quantile sur la durée avant rupture de stock pour un quantile égale à 1 - le niveau de service désiré. En d'autres mots, pour calculer à quel moment au plus tard réapprovisionner un client en respectant un niveau de service de 95%, il faut prédire le 5^{e} centile de l'intervalle inter-demande dont la demande est suffisante pour provoquer une rupture de stock

CHAPITRE 3 MÉTHODOLOGIE

L'objectif principal est de calculer le moment optimal de réapprovisionnement avec une certaine contrainte sur la probabilité de rupture de stock et avec comme unique source de donnée l'historique de transactions. Les algorithmes issus de cette méthodologie permettront aussi de :

1. Prédire un quantile de la **demande accumulée** pendant un intervalle de temps (par exemple, le 95^{e} centile de la demande accumulée entre le moment du dernier approvisionnement et le moment du prochain approvisionnement planifié); et
2. Prédire un quantile de **l'intervalle** de temps pour une demande accumulée (par exemple, le 5^{e} centile de l'intervalle, débutant au dernier approvisionnement, dont la demande accumulée est suffisante pour provoquer une rupture de stock).

La méthodologie présentée dans cette section permettra d'utiliser, d'évaluer, de paramétrer et de sélectionner de tels algorithmes. Les algorithmes de prévision quantile qui seront utilisés peuvent mettre en relation ces trois variables: demande, intervalle de temps et quantile.

Les étapes de la méthodologie proposée permettant d'atteindre l'objectif et les sous-objectifs sont les suivantes :

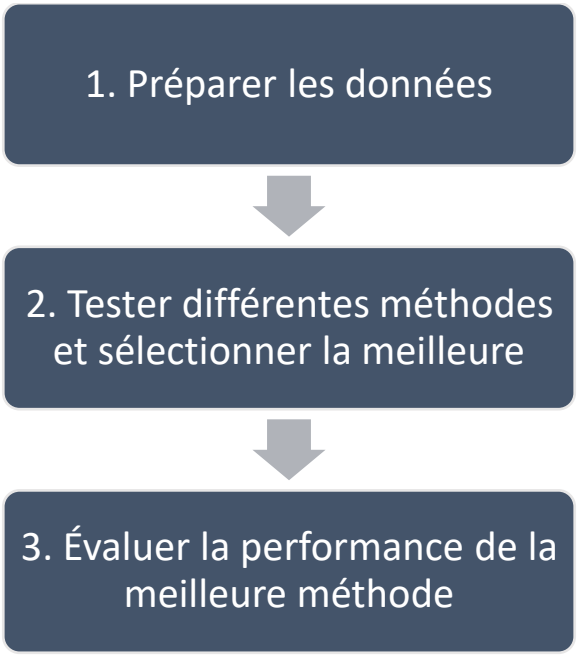

Figure 3.1 Étapes de la méthodologie

L'étape 1, soit la préparation des données, consiste à extraire les données d'entrées nécessaires, calculer des informations supplémentaires, traiter les données aberrantes et retirer les données tests pour la validation croisée. **L'étape 2** montre comment tester et sélectionner les méthodes. La prévision classique est d'abord employée, puisqu'elle permet de plus facilement se familiariser avec les données, les modèles et les paramètres. Ensuite, la prévision quantile permettra d'atteindre les objectifs. **L'étape 3,** soit l'évaluation de la méthode, permet d'estimer la précision qu'on obtiendra dans le futur avec la méthode sélectionnée. Les courbes de rendements montrent visuellement et concrètement la performance du modèle en fonction de différents niveaux de service visés.

3.1 Préparer les données

3.1.1 Extraire les données transactionnelles nécessaires

Les données qu'il faut puiser de chaque transaction sont 1) l'identifiant du client (ou du réservoir), 2) la date de livraison, 3) la quantité livrée et 4) la lecture du stock après (ou avant) la livraison. Si les données transactionnelles comprennent des lectures du « stock avant livraison », il suffit d'y additionner la quantité livrée pour obtenir le « stock après livraison ». Si aucune lecture de stock n'est disponible, c'est peut-être que les réservoirs, lors des livraisons, sont remplis jusqu'au même niveau, par exemple jusqu'à leur capacité maximale. Dans ces situations, on utilise cette information pour inférer les données du stock après livraison. Le tableau suivant montre le format des données transactionnelles à extraire.

Tableau 3.1 Données transactionnelles nécessaires

Id client	**Date de livraison**	**Quantité livrée**	**Stock après livraison**
123456	D_1	Q_1	S_1
123456	D_2	Q_2	S_2
123456	D_3	Q_3	S_3
...	...	...	...

3.1.2 Enrichir les données

Pour l'enrichissement les données, on calcule les utilisations, les intervalles et les taux d'utilisation.

L'utilisation est la quantité qui a été utilisée depuis la transaction précédente. L'intervalle est la durée entre une transaction et la précédente. Le taux est l'utilisation divisée par l'intervalle. Le tableau suivant montre les calculs à effectuer.

Tableau 3.2 Enrichissement des données

Id client	Date de livraison	Quantité livrée	Stock après livraison	**Utilisation**	**Intervalle**	**Taux**
123456	D_1	Q_1	S_1	NULL	NULL	NULL
123456	D_2	Q_2	S_2	$U_2 = Q_2 + (S_1 - S_2)$	$I_2 = D_2 - D_1$	$T_2 = U_2 / I_2$
123456	D_3	Q_3	S_3	$U_3 = Q_3 + (S_2 - S_3)$	$I_3 = D_3 - D_2$	$T_3 = U_3 / I_3$
...	...	...	...	...	...	...

3.1.3 Détecter et traiter les données aberrantes

Cette étape est très dépendante du cas d'étude. Les données doivent être interprétées et traitées adéquatement selon les règles du domaine.

Premièrement, il peut y avoir des données impossibles (par exemple, une « quantité livrée » supérieure à la capacité du camion de livraison, un « stock après livraison » supérieur à la capacité maximale du client, ou un stock après livraison inférieure à la quantité livrée). En présence de telles aberrations, on peut choisir de les ignorer, d'éliminer ces lignes, ou de tenter de corriger les données problématiques en posant des hypothèses. Ces trois options comportent toutes des risques. Le traitement doit être fait judicieusement et en fonction du cas d'étude. Des exemples seront présentés en 4.1.3.

Deuxièmement, il peut y avoir des intervalles anormalement courts (par exemple, deux transactions la même journée). Il se peut que ça soit une erreur d'entrée de données (double entrée, ou bien tentative de correction), et qu'il faille choisir quelle transaction garder. Il se peut aussi qu'il y ait

véritablement eu deux transactions anormalement rapprochées. Dans ces cas-là, il peut être préférable de les agréger. Encore une fois, le traitement doit être fait judicieusement et en fonction du cas d'étude.

3.1.4 Retirer des données tests

À cette étape, on met de côté un ensemble de données tests. Ces données tests ne seront dévoilées que lors de l'évaluation finale (étape 3). Pour que les résultats de l'évaluation finale soient représentatifs de la performance hors échantillon, il sera interdit de retourner à l'étape 2 pour ajuster les modèles en fonction des résultats de l'étape 3.

Si la quantité de transactions par client est très faible, mais qu'il y a un grand nombre de clients, il vaut mieux ne retirer qu'une seule donnée par client, ce qui permet d'en garder le plus possible pour entraîner les modèles à l'étape 2. Avec des milliers de données test, l'évaluation de la performance globale aura un haut niveau de confiance.

3.2 Sélectionner la meilleure méthode

3.2.1 Tester les méthodes de prévision classique

Bien que la solution traditionnelle pour planifier les commandes de réapprovisionnement sous la contrainte de satisfaction du niveau de service (prévision classique + stock de sécurité) est inefficace, il ne faut pas mettre de côté les méthodes de prévisions classiques. En effet, plus loin dans la méthodologie (section 3.2.2.1), une approche sera proposée pour obtenir des prévisions quantiles en utilisant des méthodes de prévisions classiques, et ce, sans poser l'hypothèse de normalité.

Cette étape a pour but d'explorer les méthodes de prévision classique et leurs paramètres ainsi que de découvrir lesquelles fonctionnent le mieux (les meilleures combinaisons de méthodes et de paramètres seront réutilisées à l'étape 3.2.2). De plus, à cette étape préliminaire, c'est le moment de vérifier que les traitements de données (de l'étape 3.1) ont été faits adéquatement et que les algorithmes ont été implémentés sans erreur.

Les méthodes de prévision classiques seront testées en suivant ces étapes :

1. N est le nombre de tests à effectuer par client;
2. Pour chaque modèle et chaque paramètre à tester:
    2.1. Pour chaque client :
        2.1.1. Prendre la série des utilisations et celle des intervalles;
        2.1.2. Pour k allant de 1 à N :
            2.1.2.1. Retirer la dernière donnée des séries (u_test et i_test);
            2.1.2.2. Entraîner le modèle sur les séries raccourcies;
            2.1.2.3. Prédire l'utilisation ($\hat{U}$) en fonction de l'intervalle (i_test);
            2.1.2.4. Prédire l'intervalle ($\hat{I}$) en fonction de l'utilisation (u_test);
    2.2. Calculer les indicateurs (MAE) (équation 7) pour ce modèle et ce paramètre :
        2.2.1. Pour la prévision de l'utilisation, les erreurs sont calculées sur $\hat{U}$ et u_test;
        2.2.2. Pour la prévision de l'intervalle, les erreurs sont calculées sur $\hat{I}$ et i_test;
3. Analyser les résultats :
    3.1. Identifier les meilleurs paramètres pour chaque modèle (ceux qui minimisent MAE);
    3.2. Identifier les meilleurs modèles (ceux qui minimisent MAE).

3.2.1.1 Méthodes de prévision générales

Tel qu'il a été discuté dans l'introduction et la revue critique, les données transactionnelles disponibles ne sont pas directement exploitables avec les méthodes de prévision générales de la littérature. Puisqu'on ne connaît que la demande totale accumulée pendant les intervalles inter-visites, on pourrait dire que la demande est sous une forme pré-agrégée. Toutefois, le niveau d'agrégation est variable (puisque les intervalles inter-visites varient). Puis, même si les intervalles étaient fixes, le niveau d'agrégation est beaucoup trop grossier pour permettre d'effectuer une prévision de la demande sur un intervalle futur.

On peut tenter de surmonter l'obstacle en posant l'hypothèse que le taux de la demande (demande/jour) est constant entre les visites. Alors, il devient possible d'appliquer l'une des deux stratégies suivantes pour obtenir des prévisions avec des méthodes de prévision générales.

La **première stratégie** est:

1. À partir de la série des taux d'utilisation, effectuer une prévision (avec une méthode générale);
2. Utiliser le taux d'utilisation prédit :
    - Pour effectuer une prévision de la demande pendant un certain intervalle, multiplier le taux prédit par l'intervalle;
    - Pour effectuer une prévision de l'intervalle pour une certaine demande, diviser la demande par le taux prédit.

La **deuxième stratégie** est:

1. Créer une série temporelle de demandes journalières en répétant les taux d'utilisation avec les intervalles (voir figure 3.2);
2. À partir de la série temporelle de demandes obtenue, effectuer une prévision de la demande sur les prochaines journées (avec une méthode générale) :
    - Pour effectuer une prévision de la demande pendant un certain intervalle, additionner les demandes prédites pour les journées de l'intervalle voulu;
    - Pour effectuer une prévision de l'intervalle pour une certaine demande, compter les prochaines journées jusqu'à ce que la somme cumulative des demandes prédites atteigne la demande voulue.

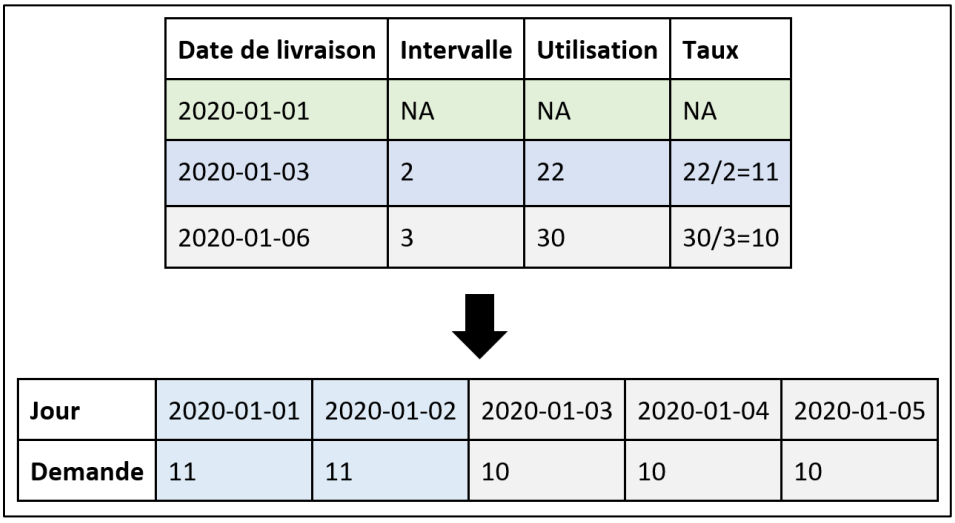

Figure 3.2 Création d'une série temporelle de demandes journalières

Ces deux stratégies exploitent l'hypothèse que la demande inter-visite est proportionnelle à la longueur de la période inter-visite et sont vraisemblablement sensibles à cette hypothèse. La deuxième stratégie, comparativement à la première, permet d'accorder plus d'importance aux demandes inter-visites provenant des plus longs intervalles (par exemple, une demande de 300 unités en 300 jours aura plus d'impact sur les prévisions qu'une demande de 10 unités en 10 jours.) Pour savoir si une stratégie est meilleure que l'autre, il faut tester avec des données. Parfois le fournisseur n'a pas une confiance absolue en la qualité de ses données et a besoin d'une méthode robuste. La première stratégie est plus sensible aux valeurs aberrantes lorsque la demande est grande et l'intervalle petit, alors que la deuxième stratégie est plus sensible aux valeurs aberrantes lorsque la demande est faible et l'intervalle est grand. Néanmoins, la deuxième stratégie permet d'obtenir l'approximation d'une série temporelle de la demande journalière, ce qui est essentiel pour traiter certains comportements, tels que la saisonnalité. En effet, si des profils saisonniers sont connus, ils peuvent être appliqués facilement au modèle. De plus, les comportements saisonniers peuvent être détectés à condition que l'historique de transactions soit suffisamment long et que les comportements ne soient pas cachés à l'intérieur des intervalles inter-visites.

Les méthodes les plus simples sont la moyenne mobile (équation 1) et le lissage exponentiel simple (équation 2). Les méthodes plus avancées de la littérature permettent de considérer la saisonnalité et les effets calendaires. Cependant, puisque la nature intermittente des données cache et brouille ces comportements, il est préférable de se limiter aux méthodes simples.

Les méthodes générales à tester sont donc: 1) la moyenne mobile avec la première stratégie, 2) le lissage exponentiel simple avec la première stratégie, 3) la moyenne mobile avec la deuxième stratégie, 4) le lissage exponentiel simple avec la deuxième stratégie.

3.2.1.2 Méthodes de prévision avec demande intermittente

Tel qu'il a été discuté dans l'introduction, la deuxième solution proposée, pour surmonter l'obstacle des données transactionnelles, est de considérer que la demande est intermittente. En faisant cette hypothèse, on peut employer les méthodes de prévision avec demande intermittentes de l'état de l'art.

Avant de poursuivre, il faut rappeler que, dans notre contexte, la demande est différente de celle de la littérature. Cela étant dit, nos séries partagent certaines irrégularités avec les « vraies »

demandes intermittentes et les méthodes associées peuvent convenir. C'est pourquoi ces méthodes ont déjà été utilisées dans le contexte de la logistique VMI (Murray et al., 2018).

La méthode de **Croston** est applicable. L'application ressemble à celle de la « première stratégie » vue dans la section précédente. Voici les étapes proposées :

1. Prendre la série des demandes inter-visites et la série des intervalles inter-visites;
2. À l'aide de la méthode de Croston vue dans la littérature (équations 3 et 4), obtenir une prévision de la demande ($\hat{y}$) et de l'intervalle ($\hat{\imath}$);
3. Diviser les deux prévisions ($\hat{y}$ / $\hat{\imath}$) pour obtenir une prévision du taux de la demande;
4. Utiliser le taux prédit :
    - Pour effectuer une prévision de la demande pendant un certain intervalle, multiplier le taux prédit par l'intervalle;
    - Pour effectuer une prévision de l'intervalle pour une certaine demande, diviser la demande par le taux prédit.

Dans le contexte de la problématique, les méthodes **ADIDA, iADIDA** et **MAPA** ne sont pas applicables. Comme mentionné précédemment, nos données sur la demande sont déjà grossièrement agrégées avec des niveaux d'agrégations hétérogènes. En conséquence, il est pratiquement impossible de réagréger de façon uniforme (à moins de tout agréger dans le même groupe, ce qui reviendra ultimement à faire une moyenne de toutes les données). De plus, puisque nos données sur la demande sont déjà sous forme agrégées, il n'est pas désirable d'appliquer une seconde agrégation, puisque ce processus amène une perte d'information utile (Nikolopoulos et al., 2011) et augmente l'erreur de prévision (Jin et al., 2015).

La méthode **ASACT** est conçue pour le même type de données que les nôtres, c'est-à-dire qu'elle est adaptée aux données transactionnelles et vise même à corriger les problèmes caractéristiques de ce type de données. En principe, pour l'application, il suffit de suivre la méthodologie proposée par l'auteur. Cependant, la première étape de celle-ci manque peut-être de clarté et de cohérence. Premièrement, l'auteur recommande de choisir la taille du bac d'agrégation *minimal* qui est déterminé par la contrainte que chaque groupe doit contenir au maximum une seule valeur de demande non nulle. Ça serait donc un problème non borné, c'est-à-dire qu'il faut minimiser la taille du bac alors que la contrainte est de type « $\leq$ ». L'auteur voulait sûrement recommander de choisir

la taille du bac d'agrégation *maximal*. Deuxièmement, bien que l'auteur mentionne à plusieurs reprises que la première étape de son algorithme évite toute sommation, dans l'explication de la première étape, il donne comme unique exemple un cas où deux livraisons sont additionnées. Alors, voici les étapes de la méthode ASACT selon ma meilleure interprétation:

1. **Agréger** les données transactionnelles. Le niveau d'agrégation temporelle est choisi pour minimiser le nombre de groupes tel qu'aucune demande n'est additionnée. Puisque les intervalles inter-demandes sont variables, cette étape crée des séries intermittentes;
2. **Lisser** la série en appliquant la méthode de Croston sur la série temporelle intermittente. Cette étape élimine l'intermittence;
3. **Agréger** la série temporelle au niveau désiré pour l'analyse;

Pour l'étape 3, le niveau désiré est une agrégation journalière. De cette manière, on peut poursuivre avec la dernière étape de la « deuxième stratégie » (vue à la section précédente) pour effectuer les prévisions à partir d'une série corrigée par ASACT.

4. À partir de la série temporelle obtenue, effectuer une prévision de la demande sur les prochaines périodes (avec une méthode générale) :
    - Pour effectuer une prévision de la demande pendant un certain intervalle, additionner les demandes prédites pour les périodes de l'intervalle voulues;
    - Pour effectuer une prévision de l'intervalle pour une certaine demande, compter les prochaines périodes jusqu'à ce que la somme cumulative des demandes atteigne la demande.

Les méthodes de prévision avec demande intermittente à tester sont donc: 1) Croston, 2) ASACT + moyenne mobile, et 3) ASACT + lissage exponentiel simple.

À la fin de cette étape, on doit avoir identifié les meilleurs paramètres pour chaque méthode, ainsi que les meilleures méthodes de prévision déterministe. Pour se faire, on choisit les paramètres et les méthodes qui minimisent la métrique d'erreur MAE (équation 7).

3.2.2 Tester les méthodes de prévision quantile

Cette étape a pour but de choisir la meilleure méthode qui résout la problématique. Pour ce faire, l'étape consiste à explorer les méthodes de prévision probabilistes et leurs paramètres ainsi que de

découvrir lesquelles minimise la métrique d'erreur (les meilleures combinaisons de méthodes et paramètres seront réutilisées pour la validation finale).

Les méthodes de prévision quantiles seront testées en suivant ces étapes :

1. N est le nombre de tests à effectuer par client;
2. Pour chaque modèle, chaque paramètre et chaque quantile à tester :
   2.1. Pour chaque client :
      2.1.1. Prendre la série des utilisations et celle des intervalles;
      2.1.2. Pour k allant de 1 à N :
         2.1.2.1. Retirer la dernière donnée des séries (u_test et i_test);
         2.1.2.2. Entraîner le modèle sur les séries raccourcies;
         2.1.2.3. Prédire le quantile de l'utilisation ($\hat{U}_q$) en fonction de l'intervalle (i_test) et du quantile (q);
         2.1.2.4. Prédire le quantile de l'intervalle ($\hat{I}_q$) en fonction de l'utilisation (u_test) et du quantile (1-q);
   2.2. Calculer les indicateurs (TL_q) (équation 11) pour ce modèle-paramètre-quantile;
3. Analyser les résultats :
   3.1. Identifier les meilleurs paramètres pour chaque modèle (ceux qui minimisent TL_q);
   3.2. Identifier les meilleurs modèles (ceux qui minimisent TL_q).

Seulement les quantiles intéressants doivent être testés. Par exemple si le niveau de service visé est 90%, on teste pour q = 0.9.

Deux approches seront proposées. La première est l'approche en deux étapes, qui cherche d'abord à prévoir la valeur centrale (avec des méthodes de prévision classique des sections 2.1.1 et 2.1.2), puis utilise les résidus pour estimer le quantile. La deuxième est l'approche directe qui consiste à prévoir directement le quantile (avec des méthodes de prévision de quantile de la section 2.2.2).

3.2.2.1 Approche en deux étapes

Cette approche permet d'effectuer des prévisions quantiles à partir de méthodes de prévision classique. Voici les étapes de cette approche. La prévision classique principale, générée à l'étape 1 et utilisée en 2.2, est notée « P », puis les prévisions classiques suivantes, générées itérativement à l'étape 2.1 pour l'estimation du quantile de l'erreur, sont notées « p » :

1. Faire une prévision (P) avec une méthode classique;
2. Ajouter un quantile de l'erreur (l'équivalent d'un « stock de sécurité ») :
   2.1. Calculer un quantile des erreurs passées :
      2.1.1. Initialiser une liste d'erreur;
      2.1.2. Répéter tant que la série a encore plus d'une donnée :
         2.1.2.1. Retirer le point le plus récent;
         2.1.2.2. Avec la série raccourcie, faire une prévision (p);
         2.1.2.3. Calculer l'erreur (e = observation - p);
         2.1.2.4. Ajouter l'erreur à la liste d'erreur;
      2.1.3. Corriger le biais de la liste d'erreur (pour que la moyenne des erreurs soit 0);
      2.1.4. Calculer un quantile de la liste d'erreur;
   2.2. Additionner le quantile à la prévision (P + quantile).

Les méthodes à tester avec cette approche sont : 1) la moyenne mobile avec la première stratégie, 2) le lissage exponentiel simple avec la première stratégie, 3) la moyenne mobile avec la deuxième stratégie, 4) le lissage exponentiel simple avec la deuxième stratégie 4) Croston, 5) ASACT + moyenne mobile, et 6) ASACT + lissage exponentiel simple.

Toutefois, ayant déjà testé toutes ces méthodes de prévision classique (à l'étape 3.2.1), il est plus facile de cibler les méthodes et les paramètres qui fonctionnent le mieux

3.2.2.2 Approche directe

L'approche directe consiste à prévoir directement le ou les quantiles en utilisant des méthodes de prévision quantile. Tel qu'on l'a vue dans la revue de littérature, la technique du bootstrap est largement répandue pour résoudre les problèmes de prévision avec demande intermittente. Plus précisément, des versions alternatives du bootstrap sont adaptées au problème de demande intermittente. Cependant, ces versions alternatives ne semblent pas réellement adaptées à notre problème, puisqu'elles traitent les demandes indépendamment des intervalles inter-demande. Ce problème a déjà été expliqué dans la section 3.2.1.2

Bien qu'elle ne soit pas parfaitement adaptée au contexte, la méthode de la littérature sera testée à des fins de comparaisons. Voici une reformulation des étapes de la méthode « VZ_adj » (Hasni et al., 2019). Pour l'identifier, cet algorithme sera surnommé **Bootstrap 1.**

1. Produire un histogramme pour les valeurs des utilisations et un autre histogramme pour les valeurs des intervalles inter-demande;
2. Pour 1000 répétitions :
   2.1. Initialiser **H = 0** (H est une variable pour la progression du temps);
   2.2. Initialiser **D = 0** (D représente la demande totale pendant la durée **L**);
   2.3. Tant que **H ≤ L** :
      2.3.1. Générer une utilisation (**u**) (selon l'histogramme des utilisations);
      2.3.2. Générer un intervalle inter-demande (**iid**) (selon l'histogramme des intervalles);
      2.3.3. **D = D + u**
      2.3.4. **H = H + iid**
   2.4. Enregistrer le D;
3. Générer la distribution des D.

Cette méthode permet de générer une distribution de la demande totale pendant la durée d'un intervalle L. Il ne reste plus qu'à calculer un quantile de cette distribution pour atteindre le premier sous-objectif (prédire un quantile de la demande accumulée pendant un intervalle de temps). Pour ce qui est du deuxième sous-objectif (prédire un quantile de l'intervalle de temps pour une demande accumulée), on peut suivre ces étapes légèrement modifiées.

1. Produire un histogramme pour les valeurs des utilisations et un autre histogramme pour les valeurs des intervalles inter-demande;
2. Pour 1000 répétitions :
   2.1. Initialiser **H = 0** (H est une variable pour la somme cumulative de la demande);
   2.2. Initialiser **L = 0** (L représente l'intervalle de temps pour une demande accumulée **D**);
   2.3. Tant que **H ≤ D** :
      2.3.1. Générer une utilisation (**u**) (selon l'histogramme des utilisations);
      2.3.2. Générer un intervalle inter-demande (**iid**) (selon l'histogramme des intervalles);
      2.3.3. **H = H + u**
      2.3.4. **L = L + iid**
   2.4. Enregistrer le L;
3. Générer la distribution des L.

Le premier problème avec Bootstrap 1 est que les utilisations sont générées indépendamment des intervalles inter-demande. Dans le contexte de notre problématique, puisque le fournisseur contrôle (et cherche à maximiser) l'intervalle inter-visite, il serait absurde et risqué de considérer que la demande n'est pas liée à l'intervalle. Pour pallier ce problème, et donc prendre en compte la corrélation supposée entre les demandes et les intervalles, la solution proposée est de générer les demandes et les intervalles en paires.

Le deuxième problème est que dans notre contexte, la période (L) sur laquelle on veut faire une prévision de la demande peut être approximativement de la même taille que les intervalles (iid) historiques. En effet, puisque les données sur la demande réelle ne sont pas disponibles et qu'on n'a accès qu'aux données des transactions passées, on doit considérer que la période (L) est la durée entre la dernière transaction et la prochaine. En conséquence, la demande simulée (D) est inutilement imprécise et biaisée à la hausse, c'est-à-dire que pour simuler des demandes (D) pour l'intervalle L, on additionne des utilisations (u) pour un intervalle H qui sera toujours significativement supérieur à L. Pour éviter d'importantes erreurs de précision, la solution proposée est de réduire le niveau d'agrégation, et donc de simuler des demandes journalières. Voici donc la méthode **Bootstrap 2**, qui vise à prendre en compte la corrélation entre les demandes et les intervalles, ainsi qu'à éviter les erreurs de troncatures :

1. Prendre la série des intervalles et celle des taux d'utilisation;
2. Initialiser une série temporelle (S) qui contiendra une demande pour chaque journée pendant la simulation;
3. Pour 1000 répétitions :
    3.1. Sélectionner aléatoirement, en paire, le taux ($\mathbf{t_i}$) et l'intervalle associé ($\mathbf{iid_i}$);
    3.2. Pour iid_i répétitions :
        3.2.1. Ajouter $\mathbf{t_i}$ à la série (S);
4. Pour effectuer une prévision de la demande pendant un certain intervalle :
    4.1. Agréger la série (S) de façon à avoir L journées par groupe, et sommer les demandes dans chaque groupe;
    4.2. Générer la distribution à partir du résultat de l'agrégation;
5. Pour effectuer une prévision de l'intervalle pour une certaine demande :
    5.1. Agréger la série (S) de façon à avoir une demande totale D dans chaque groupe, et compter les journées dans chaque groupe;

5.2. Générer la distribution à partir du résultat de l'agrégation.

Il a été montré dans la littérature que l'ajout de variation aléatoire (« *jittering* ») améliore la précision des prévisions (Willemain et al., 2004). Une transformation ($D_{jittered}$) de la demande (D) peut être calculée de la manière suivante, où Z ~ N(0, 1) :

$$D_{jittered} = MAX\left(0, D + Z\sqrt{D}\right) \quad (10)$$

Pour permettre d'ajuster l'intensité du « *jittering* », on introduira un coefficient (C):

$$D_{jittered} = MAX\left(0, \quad D + C * Z\sqrt{D}\right) \quad (14)$$

Voici la méthode **Bootstrap 3** qui vise à corriger la sous-estimation de la variance de la distribution obtenue par bootstrap :

1. Prendre la série des intervalles et celle des taux d'utilisation;
2. Initialiser une série temporelle (S) qui contiendra une demande pour chaque journée pendant la simulation;
3. Pour 1000 répétitions :
   3.1. Sélectionner aléatoirement, en paire, le taux ($\mathbf{t_i}$) et l'intervalle associé ($\mathbf{iid_i}$);
   3.2. Pour iid_i répétitions :
      3.2.1. Calculer une variation aléatoire de t_i, soit MAX(0, t_i + C * Z$\sqrt{}$ t_i);
      3.2.2. Ajouter cette valeur à la série (S);
4. Pour effectuer une prévision de la demande pendant un certain intervalle :
   4.1. Agréger la série (S) de façon à avoir L journées par groupe, et sommer les demandes dans chaque groupe;
   4.2. Générer la distribution à partir du résultat de l'agrégation;
5. Pour effectuer une prévision de l'intervalle pour une certaine demande :
   5.1. Agréger la série (S) de façon à avoir une demande totale D dans chaque groupe, et compter les journées dans chaque groupe;
   5.2. Générer la distribution à partir du résultat de l'agrégation;

Finalement, pour donner plus de poids relatif aux données plus récentes, la probabilité associée à chaque donnée d'un client sera calculée par décroissance exponentielle. Intuitivement, plus la

donnée est vieille, moins elle est probable. Pour obtenir cette propriété, on créer un histogramme à l'aide d'une équation où la probabilité (p_i) de la donnée i (indexe commençant par 0) est calculée en fonction d'un coefficient de lissage (α) et de la taille de la série (n):

$$p_i = \alpha^{n-i-1} \frac{1-\alpha}{1-\alpha^n} \tag{15}$$

Voici la méthode **Bootstrap 4** :

0. Calculer la série (P) des probabilités (p_i) associées à chaque donnée historique;
1. Prendre la série des intervalles et celle des taux d'utilisation;
2. Initialiser une série temporelle (S) qui contiendra une demande pour chaque journée pendant la simulation;
3. Pour 1000 répétitions :
   - 3.1. Sélectionner aléatoirement, en paire, le taux (**t_i**) et l'intervalle associé (**iid_i**) :
     - 3.1.1. Générer aléatoirement un index (i) (en suivant P);
     - 3.1.2. Sélectionner le i-ème taux (**t_i**) et l'intervalle associé (**iid_i**);
   - 3.2. Pour iid_i répétitions :
     - 3.2.1. Calculer une variation aléatoire de t_i, soit MAX(0, t_i + C * Z√ t_i);
     - 3.2.2. Ajouter cette valeur à la série (S);
4. Pour effectuer une prévision de la demande pendant un certain intervalle :
   - 4.1. Agréger la série (S) de façon à avoir L journées par groupe, et sommer les demandes dans chaque groupe;
   - 4.2. Générer la distribution à partir du résultat de l'agrégation;
5. Pour effectuer une prévision de l'intervalle pour une certaine demande :
   - 5.1. Agréger la série (S) de façon à avoir une demande totale D dans chaque groupe, et compter les journées dans chaque groupe;
   - 5.2. Générer la distribution à partir du résultat de l'agrégation.

Les algorithmes de cette section sont écrits de façon simplifiée pour faciliter la compréhension. Pour accélérer les calculs, il est préférable de générer tous les nombres aléatoires en même temps et de retirer les deux boucles en vectorisant l'algorithme.

Avec ces méthodes (Bootstrap 1 à 4) on obtient une distribution prédictive de la demande (en fonction de n'importe quelle durée) ou une distribution prédictive de la durée (en fonction de

n'importe quelle demande). À partir de ces distributions, on peut facilement calculer n'importe quels quantiles. Par exemple, le 90^{e} centile de la distribution de la demande représente la prévision du 90^{e} centile de la demande accumulée pendant l'intervalle (L) testé. En d'autres mots, on prévoit qu'il y a 10% de chance que la demande réelle dépasse cette valeur calculée.

La distribution de l'intervalle est particulièrement utile pour optimiser la date de réapprovisionnement. Il suffit de prendre la demande (D) égale à la demande menant à une rupture de stock (D = niveau de stock après la dernière livraison). Ensuite, si on vise un niveau de service de 95%, par exemple, on n'a qu'à calculer le 5^{e} centile de la durée de la distribution prédictive. En d'autres mots, on prévoit qu'il y a 5% de chance qu'une rupture de stock survienne plus tôt que cette durée calculée. Finalement, il ne reste qu'à additionner cette durée à la date de la dernière livraison pour obtenir la date critique pour le prochain réapprovisionnement, soit la date maximale qui respecte le niveau de service de 95%.

Les méthodes à tester avec l'approche directe sont : Bootstrap 1, Bootstrap 2, Bootstrap 3 et Bootstrap 4.

À la fin de l'étape 3.2.2, on doit avoir identifié les meilleurs paramètres, puis les meilleures méthodes de prévision quantile. Pour se faire, on sélectionne les paramètres et les méthodes qui minimisent la métrique d'erreur TL (équation 11).

3.3 Évaluer la meilleure méthode

Cette étape a pour but d'évaluer la performance de la méthode qui vient d'être sélectionnée. C'est maintenant qu'on reprend les données tests qu'on avait mises de côté lors de la préparation des données.

3.3.1 Mesurer la précision des prévisions

En utilisant la meilleure méthode (et paramètres) de prévision quantile, on calcule les métriques d'erreur en suivant les étapes suivantes :

1. N est le nombre de données tests, par client, qui avaient été mises de côté;
2. Pour chaque client :
    2.1. Prendre la série des utilisations et celle des intervalles;
    2.2. Pour k allant de 1 à N :

2.2.1. Ajouter la plus vieille donnée test aux données d'apprentissage (u_test et i_test);

2.2.2. Entraîner le modèle sur les séries allongées;

2.2.3. Prédire le quantile de l'intervalle ($\hat{I}_q$) en fonction de l'utilisation (u_test) et du quantile (1-q);

2.2.4. Prédire le quantile de l'utilisation ($\hat{U}_q$) en fonction de l'intervalle (i_test) et du quantile (q);

3. Calculer les métriques (TL_q) :

3.1. Pour la prévision du quantile de l'intervalle, les erreurs sont calculées sur $\hat{I}_q$ et i_test;

3.2. Pour la prévision du quantile de l'utilisation, les erreurs sont calculées sur $\hat{U}_q$ et u_test.

3.3.2 Tracer les courbes de rendement

Les courbes de rendement permettent, entre autres, de montrer les impacts des prévisions et du choix de niveau de service. En utilisant les données tests, pour différents niveaux de service, on peut simuler : le nombre de livraisons, le coût en transport, le niveau de service obtenu, le « stock avant livraison » moyen, etc.

Pour ce faire, il faut effectuer des prévisions quantiles des intervalles menant à rupture de stock, et ce, pour tous les quantiles qui nous intéressent. En d'autres mots, on prédit un quantile de l'intervalle ($\hat{I}_q$) en fonction d'une demande égale au « stock après livraison » de la livraison précédente. On le fait en suivant les étapes suivantes :

1. N est le nombre de données tests, par client, qui avaient été mises de côté;
2. Pour chaque client :
   2.1. Prendre la série des utilisations et celle des intervalles;
   2.2. Pour k allant de 1 à N :
      2.2.1. Entraîner le modèle sur les séries;
      2.2.2. Pour chaque niveau de service visé (q) :
         2.2.2.1. Prédire le quantile de l'intervalle ($\hat{I}_{i+1,\,q}$) en fonction d'une utilisation égale au « stock après livraison » (S_i) et du quantile (q=1-niveau de service);

2.2.3. Ajouter la plus vieille donnée test aux données d'apprentissage.

Le tableau suivant montre le format des résultats, pour un client avec N données tests.

Tableau 3.3 Résultats des prévisions probabilistes d'intervalles pour un client

					Niveaux de service visés		
Id client	**Quantité livrée**	**Stock après livraison**	**Utilisation**	**Intervalle**	**85%**	**…**	**99%**
123456	Q_1	S_1	U_1	I_1	$\hat{I}_{1,\,q=0.15}$	…	$\hat{I}_{1,\,q=0.01}$
123456	…	…	…	…	…	…	…
123456	Q_N	S_N	U_N	I_N	$\hat{I}_{N,\,q=0.15}$	…	$\hat{I}_{N,\,q=0.01}$

3.3.2.1 Nombre de livraisons

Il faut calculer le nombre annuel de livraisons pour chaque niveau de service visé. Pour ce faire, il faut d'abord calculer le nombre annuel de livraisons par client pour chaque niveau de service visé. Si on fait l'hypothèse que les livraisons seront exécutées lors des dates prescrites par les prévisions quantiles ($\hat{I}_{i,\,q}$), on peut calculer le nombre de livraisons annuelles d'un client avec ces formules :

$$Intervalle\ moyen_q = \frac{\hat{I}_{1,q} + \cdots + \hat{I}_{N,q}}{N} \tag{16}$$

$$Nombre\ de\ livraisons\ annuelles_q = \frac{365}{Intervalle\ moyen_q} \tag{17}$$

Une fois qu'on a les valeurs pour chaque niveau de service visé et pour chaque client, on fait la somme par niveau de service visé pour obtenir le nombre annuel de livraisons.

3.3.2.2 Coûts en transport

Il faut calculer les coûts variables annuels pour chaque niveau de service visé. La façon de calculer les coûts de transport varie selon l'entreprise. Ils peuvent être calculés en fonction de la distance à parcourir, de la durée du trajet à parcourir, de la durée de service, du type de client, du type de produit, du type de camion, etc. Il est donc préférable de s'informer auprès du département de

finance ou comptabilité de l'entreprise pour connaître leur méthode d'imputation des coûts. Une façon très simple est d'utiliser le coût variable moyen par livraison. Le coût variable moyen peut être calculé en divisant le coût variable total annuel par le nombre de livraisons annuel. Pour ce qui est du coût variable total, ce sont les coûts de transports qui varient en fonction du nombre de livraisons, tel que le salaire des chauffeurs, les frais en carburant, l'amortissement des camions, etc.

3.3.2.3 Niveau de service obtenu

Pour connaître le véritable niveau de service « post-mortem » en fonction d'un niveau de service visé, il faudrait mettre en application, dans l'entreprise, les dates de livraison prescrites par la méthode de prévision. La méthode suivante permet de simuler le niveau de service « post-mortem », pour tous les niveaux de service visés, avant de mettre en application la méthode de prévision.

Pour simuler l'utilisation réelle lors de l'intervalle $\hat{I}_{i,\,q}$, on doit supposer que le taux d'utilisation demeure constant à l'intérieur de l'intervalle. Cette approximation est raisonnable lorsque l'intervalle prévu ($\hat{I}_{i,\,q}$) est près de l'intervalle observé (I_i) ou lorsqu'on a un grand nombre d'observations pour réduire le bruit.

$$Utilisation_{i,q} = \frac{U_i}{I_i} * \hat{I}_{i,q} \tag{18}$$

Ensuite, on vérifie si cette utilisation simulée mènerait à une rupture de stock :

$$Rupture\ de\ stock_{i,q} = \begin{cases} 1 & si\ \ Utilisation_{i,q} \geq S_{i-1} \\ 0 & sinon \end{cases} \tag{19}$$

Une fois qu'on a les valeurs pour chaque niveau de service visé et pour chaque donnée test, on fait la moyenne par niveau de service visé pour obtenir la probabilité de rupture de stock. Finalement, on calcule le niveau de service avec 1 - la probabilité de rupture de stock.

3.3.2.4 « Stock avant livraison » moyen

Le principe est le même que pour le niveau de service obtenu. On procède par simulation :

$$« Stock\ avant\ livraison »_{i,q} = MAX(0, \qquad S_{i-1} - Utilisation_{i,q}) \tag{20}$$

Une fois qu'on a les valeurs pour chaque niveau de service visé et pour chaque donnée test, on fait la moyenne par niveau de service visé pour obtenir le « stock avant livraison » moyen.

CHAPITRE 4 CAS D'ÉTUDE

Le projet est réalisé en partenariat avec un leader en solutions durables dans l'industrie agroalimentaire en Amérique. L'entreprise récupère des sous-produits de l'industrie agroalimentaire, les renouvelle dans ses usines spécialisées puis les retourne sur le marché sous différentes formes en demande, comme de la nourriture pour animaux, des huiles de coupe ou du cuir. Les trois unités d'affaires (récupérer, renouveler et retourner) sont appelées «1er R», «2e R» et «3e R» respectivement. Au sein du 1er R, le département de transports et logistique planifie chaque semaine les tournées de véhicules pour leurs sept types de services.

Ce projet utilisera les données transactionnelles d'un des types de service, soit la collecte d'huile usée. Le modèle de gestion d'inventaire suit la logique du VMI, mais de façon inverse. Plutôt que de distribuer des produits, du fournisseur vers les clients, la flotte de véhicules de l'entreprise va collecter l'huile usée des réservoirs chez la clientèle (principalement des restaurants), puis les achemine à ses usines de transformation. Lors d'une collecte, le réservoir d'huile est vidé et la quantité ramassée est enregistrée dans les données transactionnelles. Le niveau du réservoir « après collecte », qui est normalement nulle, est aussi enregistré dans la base de données. En plus des données transactionnelles, on connaît aussi la capacité maximale des réservoirs des clients.

L'objectif est de pouvoir planifier les commandes de collectes « le plus tard possible », tout en respectant le niveau de service. Le niveau de service à atteindre est redéfini comme étant la probabilité que le réservoir n'atteigne pas la capacité maximale avant une collecte.

En moyenne, les réservoirs sont collectés lorsqu'ils sont à seulement 49% de leur capacité maximale, ce qui représente un potentiel énorme d'économie. Ce projet vise à équiper l'entreprise d'outils de prévision probabiliste qui permettront premièrement de mieux gérer et contrôler les niveaux de service. Il sera possible de viser un niveau de service par client ou par groupe de clients et le choix pourra être facilité par des courbes de rendement (coût annuel en transport vs niveau de service). Deuxièmement, ces outils permettront de maximiser les fréquences de service, donc de réduire les coûts pour le département de transport tout en respectant les niveaux de service.

Le territoire de l'entreprise couvre : le Québec, l'Alberta et l'Ontario au Canada; le Minnesota et le Wisconsin aux États-Unis; Bogota et Medellín en Colombie et Paraná au Brésil. L'âge moyen de la clientèle est de 6 ans, alors que certains clients ont plus de 10 ans de données. Les volumes

collectés varient entre 100 kg et 50 000 kg, alors que la fréquence moyenne des collectes est de quotidienne à annuel. 90% de la clientèle ont des commandes générées par l'entreprise (suivant une logistique VMI) et appellent uniquement pour un besoin urgent de collecte, alors que le reste (10%) est « sur appel », c'est-à-dire que leurs commandes sont générées si et seulement si le client appelle. Globalement, 12% des commandes proviennent d'un appel et celles-ci sont considérées avec une priorité supérieure aux commandes générées automatiquement par le système.

La méthodologie présentée au chapitre précédent sera appliquée sur les données transactionnelles de l'entreprise, puis l'évaluation montrera quels outils ont les meilleures performances et quantifiera les économies potentielles pour la compagnie.

4.1 Préparer les données

4.1.1 Extraire les données transactionnelles nécessaires

Puisque c'est une logistique de collecte, il y a quelques changements à apporter au traitement de données. Le changement le plus évident est qu'on remplace l'action de « livraison » par celle de « collecte ». Ensuite, puisque les réservoirs sont généralement complètement vidés lors des collectes, le « stock après collecte » par défaut est de 0. Ce n'est que lors de circonstances exceptionnelles (par exemple, en présence d'un problème avec la matière, le réservoir ou l'équipement de pompage), que le chauffeur donne une lecture après collecte. Le tableau suivant montre le format des données transactionnelles à extraire pour le cas d'une logistique de collecte.

Tableau 4.1 Données transactionnelles nécessaires en logistique de collecte

Id client	Date de *collecte*	Quantité *collectée*	Stock après *collecte*
123456	D_1	Q_1	S_1
123456	D_2	Q_2	S_2
123456	D_3	Q_3	S_3
...	...	...	...

L'extraction, ainsi que plusieurs transformations sont faites par l'exécution d'une requête SQL sur la base de données de l'ERP (« *enterprise resource planning* »).

La date de réalisation d'une commande est choisie par élimination parmi plusieurs sources dans la base de données, puisqu'aucune de ces sources n'est parfaitement fiable. Les figures suivantes montrent les sections pertinentes de la requête.

```
CASE
  -- main source
  WHEN u.udf_realizedfinishinstant > '1999-12-31' THEN u.udf_realizedfinishinstant
  -- 2nd source, for orders with realisedAction
      -- (r.realisedFinishInstant is more reliable than a.finishInstant)
  WHEN r.realisedFinishInstant > '1999-12-31' THEN r.realisedFinishInstant
   -- 3rd source, for orders with action
  WHEN a.finishInstant > '1999-12-31' THEN a.finishInstant
  -- 4th source, works for most orders without action
  WHEN ept.finishInstant > '1999-12-31' THEN ept.finishInstant
  -- 5th source, for orders without action
      -- (except RA orders because they don't have from&till)
  WHEN t.from_date > '1999-12-31' and t.till_date > '1999-12-31'
      THEN CAST(CAST(t.from_date + t.till_date as int) / 2 as datetime)
  -- for RA orders, the realized instant is stored as the orderDate
      -- (o.orderDate can't be NULL)
  ELSE o.orderDate
END as RealizedInstant
```

Figure 4.1 Section du code SQL représentant le choix de la source de données pour la date de réalisation d'une commande

```
FROM [order] as o
    LEFT JOIN ud_order as u on u.id_order = o.id_order
    LEFT JOIN orderAmount as oa on oa.id_order = o.id_order and oa.id_unit = 1 -- Only get Kg
    INNER JOIN contact as c on c.id_contact = o.id_contact
    INNER JOIN task as t on t.id_order = o.id_order and t.id_actionKind = 4 -- Only get pickup task
        LEFT JOIN eiderPlannedTask as ept on ept.id_task = t.id_task
        OUTER APPLY (SELECT TOP 1 *
                     FROM [action]
                     WHERE id_task = t.id_task and finished = 1 and actionKindId <> 3
                     ORDER BY id_action DESC
                    ) as a -- Only get the last finished action per order (drive_through(s) will be
            LEFT JOIN externalAction as e on e.id_action = a.id_action
                LEFT JOIN realisedAction as r on r.id_externalAction = e.id_externalAction
                    LEFT JOIN realisationKind as rk on rk.id_realisationKind = r.id_realisationKind
```

Figure 4.2 Section du code SQL représentant les jointures des différentes sources de données

La quantité collectée est la somme des quantités des lignes de commandes qui sont en unité kilogramme, tel qu'il est montré sur la figure suivante.

```
(SELECT sum(udf_realizedAmount)
 FROM subOrder as so
 INNER JOIN ud_subOrder as uso on uso.id_subOrder = so.id_subOrder
 INNER JOIN suborderAmount as sua on sua.id_subOrder = so.id_subOrder
 WHERE so.id_order = o.id_order and sua.id_unit = 1
 ) as RealizedSubAmountKg
```

Figure 4.3 Section du code SQL montrant le calcul de la quantité collectée en Kg

Excepté pour les commandes de type « *RA* » (c'est-à-dire les commandes ayant été importées d'un ancien système), pour lesquelles la quantité réalisée est enregistrée comme une quantité planifiée. Ces figures montrent les sections pertinentes du traitement de cette exception :

```
IIF(o.comment = 'Historical Order', 1, 0) as Imported_from_RA
```

Figure 4.4 Section du code SQL montrant l'identification des commandes de type RA

```
(SELECT sum(uvalue)
 FROM subOrder as so
 INNER JOIN suborderAmount as sua on sua.id_subOrder = so.id_subOrder
 WHERE so.id_order = o.id_order and sua.id_unit = 1
) as PlannedSubAmountKg
```

Figure 4.5 Section du code SQL montrant le calcul de la quantité planifiée

```
(SELECT sum(udf_realizedAmount)
 FROM subOrder as so
 INNER JOIN ud_subOrder as uso on uso.id_subOrder = so.id_subOrder
 WHERE so.id_order = o.id_order
 ) as RealizedSubAmount
```

Figure 4.6 Section du code SQL montrant le calcul de la quantité collectée

```
CASE
  -- For RA orders, the realizedAmount is stored as a planned order amount
  WHEN Imported_from_RA = 1 THEN PlannedOrderAmountKg
  WHEN RealizedSubAmountKg > 10 THEN RealizedSubAmountKg
  WHEN RealizedSubAmount > 10 THEN RealizedSubAmount
  ELSE 0
END as RealizedAmountKg
```

Figure 4.7 Section du code SQL montrant le choix parmi les différentes sources pour les quantités collectées

Lorsqu'un chauffeur est incapable d'effectuer une commande de sa route, il entre un code indiquant la raison de l'incapacité (par exemple: « réservoir inaccessible », « manque de temps », « réservoir déjà vide », etc.). Les codes de raisons peuvent être divisés en 2 catégories. 1) « *Unplan* »: signifiant que la commande doit être replanifiée (ex. « réservoir inaccessible », « manque de temps »); 2) « *Cancel* »: signifiant que la commande n'est plus nécessaire (ex., « réservoir déjà vide »). Pour bonifier les prévisions, les codes de type « *Cancel* » sont utiles. En effet, ils indiquent que la demande a été nulle (ou presque) pendant l'intervalle entre ce service et le précédent. Les codes de raison sont enregistrés à deux endroits (deux niveaux) dans la base de données: au niveau de la commande et au niveau de l'action. Puisqu'une commande peut être associée à plus d'une route (lorsqu'une commande est manquée (avec une raison de type « *Unplan* »), elle est ensuite replanifiée sur une nouvelle route, créant ainsi de nouvelles actions), il peut exister plus d'une action par commande. Le code de raison au niveau de l'action est plus intéressant, puisqu'on peut être certain de l'association entre une commande et sa date de réalisation. Cependant le code sur la commande permet de combler les trous étant donné que certaines données sont manquantes dans la base de données. Voici les sections pertinentes de la requête :

```
IIF(u.udf_unservicecode like 'U-[CA]%', 1, 0) as is_unserviceCancel
IIF(rk.realisationKindCode like '[CA]%', 1, 0) as is_realisationCancel
```

Figure 4.8 Section du code SQL montrant l'identification des codes de raison de type « Cancel » au niveau de la commande et au niveau de l'action

```
-- UnserviceCode: Normalized unserviceCode (only of type Cancel/Annuler)
, CASE
    -- ignore the UnserviceCode when there's a realisation
    WHEN RealizedSubAmount > 10 THEN NULL
    -- realisationKindCode, when available, is more reliable
    WHEN is_realisationCancel = 1 THEN realisationKindCode
    -- udf_unservicecode is at order level, so it remains available even when actions are deleted
    WHEN is_unserviceCancel = 1 THEN SUBSTRING(udf_unservicecode, 3, 99)
    ELSE NULL
  END as UnserviceCode
```

Figure 4.9 Section du code SQL montrant la logique pour choisir la valeur du code de raison

Il arrive que les processus ne soient pas respectés par les utilisateurs du système et que des commandes soient, ni réalisés proprement (avec une quantité), ni annulés proprement (avec un code de raison) et qui n'ont pas non plus de lecture du « stock après collecte ». Ceci signifie qu'une commande avait été planifiée, mais lors de la réalisation de la route, le chauffeur ne s'est pas rendu chez le client et n'a pas pris la peine de noter le code de raison (par exemple, « manque de temps », « camion brisé », etc.). Pour ces cas, on fait l'hypothèse que le service n'a jamais eu lieu et on ignore la commande. Voici la section du code qui accomplit cette tâche.

```
cte3 as (

  SELECT
        ROW_NUMBER() OVER(PARTITION BY contactCode ORDER BY RealizedInstant) as rn
      , *
  FROM cte2
  WHERE NOT (RealizedAmountKg = 0 AND UnserviceCode is NULL) -- Remove orders that were skipped
```

Figure 4.10 Section du code SQL montrant la condition qui retire les commandes fantômes

4.1.2 Enrichir les données

On doit calculer les utilisations, intervalles et taux d'utilisation. En logistique de collecte, la formule pour calculer l'utilisation diffère (dans la formule, il faut remplacer « S_2-S_1 », par « S_1-S_2»). Le tableau suivant montre les calculs à effectuer.

Tableau 4.2 Enrichissement des données en logistique de collecte

Id client	Date de collecte	Quantité collectée	Stock après collecte	Utilisation	Intervalle	Taux
123456	D_1	Q_1	S_1	NULL	NULL	NULL
123456	D_2	Q_2	S_2	$U_2 = Q_2 + (S_2 - S_1)$	$I_2 = D_2 - D_1$	$T_2 = U_2 / I_2$
123456	D_3	Q_3	S_3	$U_3 = Q_3 + (S_3 - S_2)$	$I_3 = D_3 - D_2$	$T_3 = U_3 / I_3$
...	...	...	...	...	...	...

L'utilisation (la quantité d'huile, en kg, versée dans le réservoir par le client depuis la transaction précédente) et l'intervalle (la durée, en jour, entre la transaction et la transaction précédente) sont calculés dans la requête SQL en utilisant une jointure spéciale. Voici la section du code montrant ces calculs.

```sql
SELECT
      c.contactCode
    , c.maxStock
    , c.rn
    , c.RealizedInstant as [date]
    , c.RealizedAmountKg as collected_amount
    , c.readingAfter
    , c.RealizedAmountKg + (ISNULL(c.readingAfter, 0) - ISNULL(p.readingAfter, 0)) as usage
    , DATEDIFF(HOUR, p.RealizedInstant, c.RealizedInstant) / 24.0 as interval
    , c.UnserviceCode
FROM cte3 as c -- current row
LEFT JOIN cte3 as p on c.contactCode = p.contactCode AND (c.rn - 1 = p.rn) -- previous row
```

Figure 4.11 Section du code SQL montrant le calcul de l'utilisation et de l'intervalle

Finalement, le taux (kg/jour) sera simplement calculé en divisant l'utilisation par l'intervalle.

4.1.3 Détecter et traiter les données aberrantes

Premièrement, pour certaines aberrations, le traitement approprié est de corriger les données. Par exemple, lorsque la lecture du stock après collecte n'est pas disponible, on peut faire l'hypothèse que la valeur est 0 (*figure 4.11*). Ensuite, lorsque la date de réalisation de la transaction n'est pas

disponible (ou lorsqu'elle est aberrante, par exemple on peut retrouver des dates des années 1900), on procède par élimination, de la source la plus fiable, à la source la moins fiable: 1) date de réalisation de la commande, 2) date de réalisation de l'action provenant de l'interface avec le logiciel de gestion de flotte, 3) date de l'action dans le système de routage, 4) date planifiée dans le système de prévision intégré au système de routage, 5) calcul de la date qui est au milieu de la fenêtre de temps de la commande, 6) date de la commande (valeur créée automatiquement, qui n'est jamais nulle et qui coïncide habituellement avec la date de début de la fenêtre de temps de la commande) (*figure 4.1*). Lorsque la quantité réalisée n'est pas disponible (où lorsqu'elle est inférieure à 10 kg), on procède aussi par élimination: 1) pour les commandes qui ont été importées de l'ancien ERP la valeur est stockée en tant que quantité planifiée, alors on prend la somme des quantités planifiées des lignes de commande dont l'unité est le kilogramme, 2) somme des quantités réalisées des lignes de commande dont l'unité est le kilogramme, 3) somme des quantités réalisées des lignes de commande, 4) si rien n'est trouvé (et supérieur à 10 kg) , alors la quantité retenue est 0 (*figure 4.7*). Pour le code de raison, on procède aussi par élimination: 1) si la commande possède une quantité réalisée, alors le code de raison est ignoré (en cas de contradiction, la quantité a priorité) 2) code de raison enregistrée sur l'action la plus récente liée à la commande 3) code de raison enregistré sur la commande 4) si rien n'est trouvé, alors aucun code de raison n'est retenu (*figure 4.9*).

Deuxièmement, pour certaines aberrations, le traitement approprié est de retirer la commande des données. Les commandes à partir mars 2020 ont été retirées, pour ne pas tenir compte des effets de la pandémie sur les données de demande. On retire aussi les commandes qui sont en lien avec d'autres services ou d'autres produits, par exemple les commandes de livraison de réservoirs sont ignorées. Lorsqu'il y a plus d'une commande le même jour pour le même client, celle dont la quantité est la plus grande est conservée et les autres sont ignorées.

Finalement, il existe des types d'aberrations qui rendent insignifiante la prévision. . Le traitement approprié est de retirer toutes les lignes pour ce client. Par exemple, lorsque la capacité maximale du réservoir est inconnue (ou invalide), il est impossible d'utiliser la prévision pour la planification des commandes (affecte 0.05% des clients). Ensuite, lorsque le nombre de transactions historiques est inférieur à 5 (affecte 28% des clients), on n'a pas assez de données pour effectuer des prévisions. Puisqu'au minimum 2 données doivent être réservées pour la validation croisée (une pour la

sélection du modèle et une autre pour la validation finale), on retire tous les clients qui ont moins de 7 données (affecte 44% des clients).

4.1.4 Retirer les données tests

En validation croisée, il n'est jamais évident de déterminer la proportion de données à réserver pour les tests. Lorsqu'on effectuera les prévisions finales (à l'étape 4.3), et qu'on les comparera aux données tests, on veut que les métriques d'erreur obtenues soient les plus fidèles possible (c'est-à-dire que la performance observée soit une bonne estimation de la performance qui sera obtenue dans le futur lors d'une application réelle).

Il faut bien balancer la quantité de données tests et celle de données d'entrainement. En effet, d'un côté, plus on a de données tests, plus on peut effectuer de prévisions tests, ce qui, en principe, augmente la fidélité de la métrique d'erreur. D'un autre côté, cependant, si on ne laisse pas assez de données d'entrainement, ces prévisions risquent d'être moins bonnes, donc moins fidèles aux prévisions qui seront obtenues dans le futur (en utilisant l'ensemble complet de données). Ainsi, on souhaite avoir assez de données tests, mais aussi assez de données d'entrainement.

Lorsqu'un nombre élevé de séries de données sont disponibles, il n'est pas nécessaire de prendre beaucoup de données tests par série. En effet, avec des dizaines de milliers de clients (comme c'est le cas pour ce projet), même si on ne réserve qu'une seule donnée par client, on aura suffisamment de données tests pour obtenir un haut niveau de confiance sur la fidélité des métriques d'erreurs.

Lorsque les séries de données sont très courtes, il est critique de conserver le plus de données d'entrainement possible. Pour ce cas d'étude, une grande proportion des clients sont jeunes, c'est-à-dire que leur historique de transactions ne comporte pas beaucoup de données (28% des clients ont moins de 5 données, 37% ont moins de 6 données, 44% ont moins de 7 données, 50% ont moins de 8 données, 55% ont moins de 9 données et 59% ont moins de 10 données). Par conséquent, chaque donnée réservée (par client) est très pénalisante pour la majorité des clients, et donc pénalisante pour l'estimation de la métrique d'erreur. De plus, puisqu'on veut un minimum de 5 données historiques pour effectuer des prévisions, chaque donnée réservée (par client) nous force à écarter un grand nombre de clients. En d'autres mots, plus on prend de données tests (par client), plus il y a de clients qui, se retrouvant avec moins de 5 données d'entrainement, doivent être ignorés. Naturellement, puisque le partenaire industriel cherche à réduire au maximum ses coûts

de transports, il faut que la méthode développée puisse s'appliquer sur une grande proportion de clients. C'est pourquoi il faut inclure le plus de clients possible dans les tests.

Somme toute, bien qu'on ne puisse pas en être absolument certain, le choix de réserver une seule donnée test par client est idéal pour ce cas d'étude.

4.2 Sélectionner la meilleure méthode

4.2.1 Tester les méthodes de prévision classique

Les méthodes de prévision classiques sont évaluées par validation croisée en utilisant la MAE. « $MAE_{demande}$ » représente la moyenne des erreurs absolues pour la prévision de la demande en fonction de l'intervalle, alors que « $MAE_{intervalle}$ » est pour la prévision de l'intervalle en fonction de la demande. Un modèle ou un paramétrage qui performe bien avec « $MAE_{demande}$ » ne va pas nécessairement aussi bien fonctionner avec « $MAE_{intervalle}$ ». Il est bon de savoir que « $MAE_{demande}$ » est très affectée par les gros clients qui ont des réservoirs immenses et que « $MAE_{intervalle}$ » est très affecté par les clients dont l'utilisation journalière (soit le taux d'utilisation) est faible par rapport à la capacité des réservoirs. En conséquence, les clients ayant un gros réservoir ont plus d'impact sur les métriques d'erreurs. Pour annuler cet effet, il suffirait de diviser l'erreur par la capacité maximale du client. Cependant, comme les clients ayant des réservoirs de grande capacité sont des clients qui sont de haute importance pour l'entreprise et qui méritent donc plus d'attention, il est préférable de ne pas diviser l'erreur par la capacité.

Les tableaux suivants montrent les résultats obtenus :

4.2.1.1 Moyenne mobile (stratégie #1)

Ce tableau montre les erreurs pour la prévision de la demande ($MAE_{demande}$) ainsi que pour la prévision de l'intervalle ($MAE_{intervalle}$) pour différentes valeurs du paramètre w. Le paramètre w contrôle le nombre de données à considérer dans la moyenne mobile. Avec la « stratégie #1 » (voir 3.2.1.1), ces données sont des taux d'utilisation et donc une donnée correspond à une transaction historique. Par exemple, pour w=3, on prend la moyenne des 3 derniers taux d'utilisation pour prévoir le prochain taux d'utilisation.

Tableau 4.3 Résultats pour la moyenne mobile (stratégie #1)

w	**1**	**3**	**5**	**7**	**9**	**11**	**13**	**15**
$MAE_{demande}$	140	126	124	122.2	121.6	**121.4**	121.6	121.9
$MAE_{intervalle}$	23.1	16.1	15.3	15.2	**15.08**	15.09	15.12	15.13

Sur chacune des deux lignes de résultats, le code de couleur identifie la performance relative : le meilleur MAE est vert foncé, alors que le pire est rouge foncé.

Pour la prévision de la demande, une fenêtre (w) de 11 données donne les meilleurs résultats en moyenne (MAE=121). Pour la prévision de l'intervalle, une fenêtre (w) de 9 données donne les meilleurs résultats en moyenne (MAE=15.08), suivis de près par w=11, w=13 et w=15. Il est normal que la variation de w semble avoir de moins en moins d'impact quand w augmente et c'est causé par la distribution de la taille des séries. Beaucoup de séries ont moins de 10 données, et pour ces séries, prendre la moyenne des (9, 11, 13, ou 15) dernières données est équivalent.

4.2.1.2 Lissage exponentiel simple (stratégie #1)

Ce tableau montre les erreurs pour la prévision de la demande ($MAE_{demande}$) ainsi que pour la prévision de l'intervalle ($MAE_{intervalle}$) pour différentes valeurs du paramètre α.

Tableau 4.4 Résultats pour le lissage exponentiel simple (stratégie #1)

α	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
$MAE_{demande}$	128	121	**120**	121	123	125	128	131	135
$MAE_{intervalle}$	16.5	15.2	**15.0**	15.2	15.7	16.3	17.2	18.3	20.0

Pour la prévision de la demande ainsi que la prévision de l'intervalle, une valeur pour le coefficient alpha de 0.3 offre la meilleure précision en moyenne, selon les MAE.

4.2.1.3 Moyenne mobile (stratégie #2)

Ce tableau montre les erreurs pour la prévision de la demande ($MAE_{demande}$) ainsi que pour la prévision de l'intervalle ($MAE_{intervalle}$) pour différentes valeurs du paramètre w. Le paramètre w contrôle toujours le nombre de données à considérer dans la moyenne mobile. Avec la « stratégie #2 » (voir 3.2.1.1), ces données sont des demandes journalières et donc une donnée correspond à une journée. Par exemple, pour w=400, on prend la moyenne des 400 dernières demandes journalières pour prévoir les prochaines demandes.

Tableau 4.5 Résultats pour la moyenne mobile (stratégie #2)

w	**100**	**200**	**300**	**400**	**500**	**600**	**700**	**800**
$MAE_{demande}$	117	113	109	**108**	110	111	112	112
$MAE_{intervalle}$	19.3	16.9	16.0	15.6	15.6	**15.5**	15.6	15.6

Pour la prévision de la demande, une fenêtre (w) de 400 jours donne les meilleurs résultats en moyenne (MAE=108). Pour la prévision de l'intervalle, une fenêtre (w) de 600 jours donne les meilleurs résultats en moyenne (MAE=15.5).

4.2.1.4 Lissage exponentiel simple (stratégie #2)

Ce tableau montre les erreurs pour la prévision de la demande ($MAE_{demande}$) ainsi que pour la prévision de l'intervalle ($MAE_{intervalle}$) pour différentes valeurs du paramètre α.

Tableau 4.6 Résultats pour le lissage exponentiel simple (stratégie #2)

α	**0.001**	**0.003**	**0.005**	**0.007**	**0.009**	**0.011**	**0.013**	**0.015**	**0.017**	**0.019**
$MAE_{demande}$	142	117	111	**109.6**	110.1	111	112	113	115	116
$MAE_{intervalle}$	18.6	15.5	**15.2**	15.5	15.9	16.4	16.8	17.3	17.7	18.1

Pour la prévision de la demande, un coefficient (α) de 0.007 donne les meilleurs résultats en moyenne (MAE=109.6). Pour la prévision de l'intervalle, un coefficient (α) de 0.005 donne les meilleurs résultats en moyenne (MAE=15.2).

4.2.1.5 Croston

Ce tableau montre les erreurs pour la prévision de la demande ($MAE_{demande}$) ainsi que pour la prévision de l'intervalle ($MAE_{intervalle}$) pour différentes valeurs du paramètre α.

Tableau 4.7 Résultats pour Croston

α	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.6**	**0.7**	**0.8**	**0.9**
$MAE_{demande}$	118	110	**109**	110	112	116	120	125	131
$MAE_{intervalle}$	17.9	16.0	**15.6**	15.7	16.1	16.8	17.6	18.7	20.3

Pour la prévision de la demande ainsi que la prévision de l'intervalle, une valeur pour le coefficient alpha de 0.3 offre la meilleure précision en moyenne, selon les MAE ($MAE_{demande}$ = 109 et $MAE_{intervalle}$ = 15.6).

4.2.1.6 ASACT + Moyenne mobile

Le tableau 4.8, à la page suivante, montre les erreurs pour la prévision de la demande ($MAE_{demande}$) ainsi que pour la prévision de l'intervalle ($MAE_{intervalle}$) pour une grille de paramètres. Le paramètre α est utilisé dans l'algorithme ASACT pour le lissage par Croston. Le paramètre w est utilisé pour la prévision avec moyenne mobile.

Pour la prévision de la demande, les paramètres qui minimisent la MAE sont (α=0.3, w=1) et (α=0.999, w=400). Pour (α=0.3, w=1), puisque la fenêtre de la moyenne mobile est de 1 jour, la valeur prédite est simplement la dernière valeur de la série lissée par Croston. Ça revient à faire une prévision avec Croston (α=0.3). C'est pourquoi on retrouve les mêmes résultats que ceux obtenus plus haut. Pour (α=0.999, w=400), puisque le coefficient est de 0.999 pendant le lissage par Croston, la série lissée est pratiquement identique à la série originale. Ça revient à faire une prévision avec moyenne mobile et w=400. C'est pourquoi on retrouve les mêmes résultats que ceux obtenus plus haut. On peut aussi remarquer que des MAE comparables sont obtenues le long de la diagonale qui rejoint les paramètres optimaux. Pour la prévision de l'intervalle, les paramètres qui minimisent la MAE sont (α=0.3, w=1) et (α=0.999, w=[400-600]). On peut faire les mêmes remarques que pour la prévision de la demande.

Tableau 4.8 Résultats pour ASACT + Moyenne mobile

	w \ α	0.1	0.3	0.5	0.7	0.9	0.999
$MAE_{demande}$	**1**	118	**108.6**	112	120	131	140
	100	121	110	111	113	116	117
	200	125	112	111	112	113	113
	300	128	113	110	110	109	109
	400	132	115	111	110	109	**108.4**
	500	134	118	114	112	111	111
	600	136	120	116	114	113	112
$MAE_{intervalle}$	**1**	17.9	**15.6**	16.1	17.6	20.3	23.1
	100	18.1	**15.6**	15.8	16.7	18.1	19.3
	200	18.6	15.9	**15.6**	16.0	16.6	16.9
	300	19.2	16.1	**15.6**	15.7	15.8	16.0
	400	19.7	16.5	15.8	**15.6**	**15.6**	**15.6**
	500	20.1	16.8	16.0	15.7	**15.6**	**15.6**
	600	20.5	17.2	16.3	15.9	15.7	**15.6**

4.2.1.7 ASACT + Lissage exponentiel simple

Le tableau 4.9 qui est présenté à la page suivante montre les erreurs pour la prévision de la demande ($MAE_{demande}$) ainsi que pour la prévision de l'intervalle ($MAE_{intervalle}$) pour une grille de paramètres. Le paramètre α_1 est utilisé dans l'algorithme ASACT pour le lissage par Croston. Le paramètre α_2 est utilisé pour la prévision avec lissage exponentiel simple.

Pour la prévision de la demande, les paramètres qui minimisent la MAE sont (α_1=0.3, α_2=0.999) et (α_1=0.999, α_{2S}=0.007). Pour (α_1=0.3, α_2=0.999), puisque le coefficient du lissage exponentiel simple est de 0.999, la valeur prédite est simplement la dernière valeur de la série lissée par Croston. Ça revient à faire une prévision avec Croston (α=0.3). C'est pourquoi on retrouve les mêmes résultats que ceux obtenus plus haut. Pour (α_1=0.999, α_2=0.007), puisque le coefficient de lissage de Croston est de 0.999, la série lissée est pratiquement identique à la série originale. Ça revient à faire une prévision avec lissage exponentiel simple et α=0.007. C'est pourquoi on retrouve les mêmes résultats que ceux obtenus plus haut. On peut aussi remarquer que des MAE comparables sont obtenues le long de la diagonale qui rejoint les paramètres optimaux.

Pour la prévision de l'intervalle, les paramètres qui minimisent la MAE sont (α_1=0.3, α_2=0.999) et (α_1=0.999, α_2=0.005). Pour la première de ces configurations (α_1=0.3, α_2=0.999), puisque le coefficient du lissage exponentiel simple tend vers 1, la valeur prédite est approximativement la dernière valeur de la série lissée par l'algorithme de Croston. Alors, avec ces paramètres, l'algorithme « ASACT + Lissage exponentiel simple » est équivalent à une simple prévision avec Croston pour un paramètre α de 0.3. Il est donc normal qu'on obtienne les mêmes résultats que ceux obtenus plus haut (en 4.2.1.5). Pour la deuxième de ces configurations (α_1=0.999, α_2=0.007), puisque le coefficient de lissage de Croston tend vers 1, la série lissée est pratiquement identique à la série originale. Alors, avec ces paramètres, l'algorithme revient à faire une prévision avec lissage exponentiel simple (stratégie #2) pour un paramètre α de 0.007. C'est pourquoi on retrouve les mêmes résultats que ceux obtenus plus haut (4.2.1.4). On peut aussi remarquer que des MAE comparables sont obtenues le long de la diagonale qui rejoint les paramètres optimaux. En résumé, la première configuration revient à lisser avec Croston, la deuxième configuration revient à lisser avec le lissage exponentiel simple (stratégie #2) et les configurations qui se trouvent le long de la diagonale reviennent à appliquer les deux types de lissage, avec une proportion bien balancée.

Tableau 4.9 Résultats pour ASACT + Lissage exponentiel simple

	$\alpha_2 \setminus \alpha_1$	**0.1**	**0.3**	**0.5**	**0.7**	**0.9**	**0.999**
$MAE_{demande}$	**0.001**	160	149	145	143	142	142
	0.003	142	125	120	118	117	117
	0.005	134	117	113	112	111	111
	0.007	129	114	111	110	110	**110**
	0.01	126	111	109	110	110	111
	0.02	121	109	110	112	115	116
	0.2	118	109	112	119	130	138
	0.999	118	**109**	112	120	131	140
$MAE_{intervalle}$	**0.001**	23.4	20.4	19.4	18.9	18.6	18.6
	0.003	20.5	17.1	16.1	15.8	15.6	15.5
	0.005	19.5	16.2	15.5	15.2	15.2	**15.2**
	0.007	19.0	15.9	15.3	15.2	15.4	15.5
	0.01	18.6	15.7	15.3	15.5	15.9	16.1
	0.02	18.2	15.5	15.6	16.3	17.4	18.2
	0.2	17.9	15.6	16.1	17.6	20.3	23.0
	0.999	17.9	**15.6**	16.1	17.6	20.3	23.1

4.2.1.8 Synthèse des résultats pour les méthodes classiques

Pour ce qui est d'ASACT, la méthode ne semble pas aider pour les prévisions. En effet, dans tous les cas, la stratégie optimale revient à utiliser une des méthodes plus simples (moyenne mobile 2, lissage exponentiel 2 ou Croston). Ensuite, on peut remarquer les meilleures méthodes et meilleurs paramètres pour la prévision de la demande sont différents de ceux pour la prévision de l'intervalle. Voici les résultats pour les meilleurs paramètres de chaque méthode :

Tableau 4.10 Synthèse des résultats pour les méthodes classiques

Méthode	**$MAE_{demande}$**	**$MAE_{intervalle}$**
Moyenne mobile 1	121	15.1
Lissage exponentiel simple 1	120	**15.0**
Moyenne mobile 2	**108**	15.5
Lissage exponentiel simple 2	110	15.2
Croston	109	15.6

Les buts de l'étape 2.1 ont été atteints, soit 1) se familiariser avec les données, les méthodes de prévisions classiques et l'effet des paramètres, 2) découvrir les meilleures méthodes et paramètres et 3) valider que l'implémentation des algorithmes fonctionne jusqu'ici.

4.2.2 Tester les méthodes de prévision quantile

Les méthodes de prévision quantiles sont évaluées par validation croisée en utilisant la métrique d'erreur « *Tick Loss* » (TL_q). Les niveaux de service visés sont 85%, 90%, 95% et 99%, donc on effectuera des prévisions avec les quantiles [0.85, 0.9, 0.95, 0.99] pour la prévision de la demande et les quantiles [0.15, 0.1, 0.05, 0.01] pour la prévision de l'intervalle.

4.2.2.1 Approche en deux étapes

Les paramètres (α et w) utilisés sont ceux qui ont été sélectionnés à l'étape précédente (respectivement pour chacune des méthodes). Ce tableau montre les erreurs pour les prévisions

quantiles de la demande ($TL_{q\ demande}$) ainsi que pour les prévisions quantiles de l'intervalle ($TL_{q\ intervalle}$) :

Tableau 4.11 Résultats pour l'approche en deux étapes

	$TL_{q\ demande}$				$TL_{q\ intervalle}$			
Méthode	**q=0.85**	**q=0.9**	**q=0.95**	**q=0.99**	**q=0.15**	**q=0.1**	**q=0.05**	**q=0.01**
Moyenne mobile 1	45.0	35.4	23.4	10.4	5.63	4.54	3.20	1.83
Lissage exponentiel simple 1	43.7	34.5	22.7	9.91	5.61	4.55	3.22	1.85
Moyenne mobile 2	39.1	31.3	21.2	10.2	5.86	4.75	3.41	2.10
Lissage exponentiel simple 2	**38.8**	**30.8**	**20.4**	**9.12**	**5.54**	**4.44**	**3.13**	**1.82**
Croston	39.7	31.9	21.8	10.8	6.25	5.19	3.91	2.60

On peut rapidement voir que la méthode « Lissage exponentiel simple (stratégie #2) » minimise toutes les métriques d'erreur testées. Pour ce qui est des autres méthodes, leur position dépend du type de prévision et de la valeur du quantile (q) visé.

On peut aussi remarquer que la grandeur des erreurs diminue lorsque q s'éloigne de 0.5. Ce phénomène vient de la métrique TL_q.

4.2.2.2 Approche directe

Bootstrap 1, surnom donné à la méthode « VZ_adj » (Hasni et al., 2019) et **Bootstrap 2** (première méthode développée pour ce mémoire, qui évite les erreurs de précision et qui prend en compte la corrélation entre demandes et intervalles) sont non-paramétriques. Ce tableau compare les erreurs obtenues avec ces méthodes ainsi que la méthode « Lissage exponentiel simple (stratégie #2) » :

Tableau 4.12 Résultats pour Bootstrap 1 et Bootstrap 2

	$TL_{q\ demande}$				$TL_{q\ intervalle}$			
Méthode	**q=0.85**	**q=0.9**	**q=0.95**	**q=0.99**	**q=0.15**	**q=0.1**	**q=0.05**	**q=0.01**
Lissage exponentiel simple 2	38.8	30.8	20.4	9.12	5.54	4.44	3.13	1.82
Bootstrap 1	70.7	53.1	30.9	**7.53**	8.39	6.57	5.07	3.43
Bootstrap 2	**36.8**	**29.6**	**20.1**	9.95	**3.95**	**3.04**	**2.03**	**0.74**

Ce tableau montre les quantiles observés, soit la proportion des prévisions qui sont supérieures aux données réelles. Les quantiles observés (Q) doivent être le plus près possible du quantile visé (q).

Tableau 4.13 Quantiles observés pour Bootstrap 1 et Bootstrap 2

	$Q_{demande}$				$Q_{intervalle}$			
Méthode	**q=0.85**	**q=0.9**	**q=0.95**	**q=0.99**	**q=0.15**	**q=0.1**	**q=0.05**	**q=0.01**
Lissage exponentiel simple 2	0.82	0.86	0.90	0.93	0.21	0.17	0.12	0.09
Bootstrap 1	0.98	0.99	0.99	0.99	0.46	0.36	0.25	0.17
Bootstrap 2	0.76	0.81	0.86	0.91	0.18	0.15	0.11	0.06

Sur ce tableau, le code de couleur indique l'exactitude des quantiles observés : vert foncé s'ils sont très près du quantile visé et rouge foncé s'ils sont très loin.

L'approche directe **Bootstrap 1**, comme prévu, n'offre pas de bons résultats. Cette méthode n'est pas bien adaptée au contexte de la problématique. Puisque les LTD sont simulées en additionnant des LTD historiques (qui ont le même ordre de grandeur) jusqu'à ce qu'on dépasse le Lead Time, il est normal qu'on perde de la précision. La moyenne de la distribution prédictive de la demande est surestimée alors que sa variance est sous-estimée. Par hasard, ces deux problèmes semblent s'annuler pour le 99^{e} centile de la distribution prédictive de la demande. C'est pourquoi la métrique d'erreur (TL_q) montre un très bon score pour ce centile.

L'approche directe **Bootstrap 2** a réussi à corriger les principaux problèmes de Bootstrap 1. On observe que cette méthode minimise toutes les métriques d'erreurs testées sauf une. Lorsqu'on se penche sur les quantiles observés (tableau 4.13), on remarque qu'ils sont tous trop près de 0.5 (les quantiles supérieurs à 0.5 sont sous-estimés et les quantiles inférieurs à 0.5 sont surestimés). Ceci indique que les distributions prédictives n'ont pas assez de variance. L'introduction du processus de « *jittering* » dans la méthode **Bootstrap 3** va permettre d'augmenter les variances et devrait donc faire réduire les métriques d'erreur. Le tableau 4.14 montre les résultats obtenus avec cette méthode, pour différentes valeurs du coefficient C :

Tableau 4.14 Résultats pour Bootstrap 3

	$TL_{q\ demande}$				$TL_{q\ intervalle}$			
C	**q=0.85**	**q=0.9**	**q=0.95**	**q=0.99**	**q=0.15**	**q=0.1**	**q=0.05**	**q=0.01**
0	36.79	29.62	20.06	9.95	3.95	3.04	2.03	0.740
0.5	36.94	29.61	19.97	9.47	3.92	3.05	1.99	0.717
1	36.74	29.29	19.76	9.04	3.87	2.96	1.90	0.659
1.5	36.04	28.97	19.18	8.35	3.76	2.84	1.78	0.566
2	**35.88**	28.42	18.75	7.43	3.63	2.73	1.66	0.525
2.5	35.95	**28.37**	18.23	6.90	**3.56**	2.64	1.55	0.457
3	36.52	28.37	**18.16**	6.46	3.58	**2.62**	1.50	0.419
3.5	37.74	29.18	18.32	6.16	3.62	2.64	**1.49**	0.404
4	39.39	30.36	18.68	6.10	3.71	2.69	1.50	0.387
4.5	41.54	32.04	19.56	6.05	3.84	2.74	1.53	0.384
5	44.24	33.80	20.47	**5.96**	3.96	2.81	1.55	**0.378**
5.5	47.39	35.73	21.46	6.11	4.09	2.89	1.58	0.379
6	50.73	38.39	22.74	6.30	4.22	2.98	1.61	0.384

Ce tableau montre les quantiles observés, soit la proportion des prévisions qui sont supérieures aux données réelles. Les quantiles observés (Q) doivent être le plus près possible du quantile visé (q).

Tableau 4.15 Quantiles observés pour Bootstrap 3

	$Q_{demande}$				$Q_{intervalle}$			
C	**q=0.85**	**q=0.9**	**q=0.95**	**q=0.99**	**q=0.15**	**q=0.1**	**q=0.05**	**q=0.01**
0	0.76	0.81	0.86	0.91	0.18	0.15	0.11	0.06
0.5	0.77	0.82	0.87	0.91	0.18	0.15	0.11	0.05
1	0.78	0.82	0.88	0.92	0.17	0.14	0.10	0.04
1.5	0.80	0.84	0.89	0.94	**0.16**	0.12	0.09	0.04
2	0.82	0.86	0.91	0.95	0.14	**0.10**	0.07	0.03
2.5	**0.85**	0.88	0.93	0.96	0.11	0.08	**0.05**	0.02
3	0.87	**0.91**	0.94	0.97	0.09	0.07	0.04	0.02
3.5	0.90	0.93	**0.95**	0.98	0.07	0.05	0.03	0.02
4	0.92	0.94	0.96	0.98	0.06	0.04	0.03	0.01
4.5	0.93	0.95	0.97	0.98	0.05	0.03	0.02	**0.01**
5	0.94	0.96	0.98	0.99	0.04	0.03	0.02	0.01
5.5	0.96	0.97	0.98	**0.99**	0.03	0.02	0.01	0.01
6	0.96	0.97	0.98	0.99	0.03	0.02	0.01	0.01

Lorsque le coefficient (C) pour la variation aléatoire (« jittering ») est nul, on obtient naturellement les mêmes résultats que pour Bootstrap 2. Lorsqu'on augmente la valeur de C suffisamment, la variance de la distribution prédictive augmente, ce qui fait en sorte que le quantile observé se

rapproche du quantile visé et que le score augmente. Cependant, lorsque la valeur de C est trop élevée, le quantile observé est trop extrême (trop loin de 0.5) et le score diminue. La valeur de C qui minimise la métrique d'erreur dépend du quantile visé. En effet, plus le quantile à prédire est extrême, plus il faut produire de variations aléatoires. Le tableau suivant montre les valeurs du coefficient C qui minimisent les métriques d'erreurs :

Tableau 4.16 Valeurs optimales du coefficient C

	Demande				**Intervalle**			
	q=0.85	**q=0.9**	**q=0.95**	**q=0.99**	**q=0.15**	**q=0.1**	**q=0.05**	**q=0.01**
C	2	2.5	3	5	2.5	3	3.5	5

Le dernier modèle à tester, **Bootstrap 4**, permet d'accorder plus de poids aux données plus récentes. Tel qu'on l'a observé empiriquement avec les méthodes de prévision classique (section 4.2.1), une légère décroissance exponentielle de la pondération des données permet d'obtenir de meilleures prévisions. Ce modèle contrôle la pondération avec un paramètre α ($0 < \alpha < 1$). Lorsque α tend vers 0, seulement la dernière donnée est considérée. Lorsque α=0.5, la pondération de la dernière donnée est d'approximativement 50%. Lorsque α tend vers 1, les pondérations sont distribuées uniformément (et le modèle devient équivalent à Bootstrap 3). Le modèle a été testé pour des valeurs du coefficient α entre 0.5 et 0.9999 et en prenant les valeurs optimales pour le coefficient C trouvées avec le modèle précédent (tableau 4.16). Voici les résultats :

Tableau 4.17 Résultats pour Bootstrap 4

	$TL_{q\ demande}$				**$TL_{q\ intervalle}$**			
α	**q=0.85**	**q=0.9**	**q=0.95**	**q=0.99**	**q=0.15**	**q=0.1**	**q=0.05**	**q=0.01**
0.5	37.0	29.1	18.8	6.59	3.74	2.73	1.57	0.399
0.55	36.3	28.7	18.5	6.39	3.66	2.69	1.55	0.399
0.6	35.9	28.3	18.3	6.36	3.62	2.65	1.53	0.391
0.65	35.3	28.1	18.0	6.23	3.58	2.62	1.52	0.381
0.7	34.8	27.8	17.9	6.19	3.55	2.61	1.50	0.381
0.75	**34.5**	27.6	17.7	6.06	3.54	2.59	1.49	0.379
0.8	**34.5**	27.6	17.6	6.01	3.52	2.59	1.49	**0.378**
0.85	**34.5**	**27.5**	**17.5**	**5.86**	**3.51**	**2.58**	**1.48**	**0.378**
0.9	34.6	27.6	17.6	**5.86**	3.52	**2.58**	**1.48**	**0.378**
0.95	34.8	27.7	17.6	5.89	3.54	2.59	1.49	**0.378**
0.9999	35.9	28.4	18.2	5.96	3.56	2.62	1.49	**0.378**

On peut confirmer que lorsque α tend vers 1 (α = 0.9999), les résultats obtenus sont les mêmes qu'avec le modèle précédent. Lorsque trop de poids est accordé aux données récentes (α > 0.6),

toutes les métriques d'erreurs sont plus élevées que celles de Bootstrap 3. Lorsque la pondération est bien dosée ($0.85 \leq \alpha < 1$), toutes les métriques d'erreurs sont inférieures ou égales à Bootstrap 3. Les erreurs sont minimisées avec $\alpha=0.85$.

4.2.2.3 Synthèse des résultats pour les méthodes quantiles

Avec l'approche en deux étapes, le « Lissage exponentiel simple (stratégie #2) » minimise toutes les métriques testées. La méthode Bootstrap 1, tirée de la littérature (Hasni et al., 2019) n'est pas bien adaptée à la problématique. En conséquence, sa performance est mauvaise pour toutes les métriques sauf $TL_{q=0.99\ demande}$ où la surestimation de la moyenne de la distribution prédictive et la sous-estimation de la variance s'annulent. Bootstrap 2, première méthode développée au cours de cette étude, permet de corriger la surestimation de la moyenne de la distribution prédictive en utilisant une échelle plus appropriée (c-à-d, l'agrégation est effectuée sur des demandes journalières simulées par répétition des taux d'utilisation par l'intervalle). De plus, cette méthode considère la corrélation des demandes et des intervalles. La majorité (7/8) des scores obtenus avec cette méthode sont meilleurs qu'avec les méthodes précédentes. Ensuite, l'introduction du « *jittering* » dans Bootstrap 3 permet de corriger le manque de variance dans les distributions prédictives. Un bon dosage des variations aléatoires est en fonction du quantile à prédire et permet d'obtenir des scores meilleurs que les méthodes précédentes. Finalement, avec Bootstrap 4, en accordant légèrement plus de poids aux données récentes ($\alpha=0.85$), toutes les métriques sont minimisées. Voici les résultats (TL_q) et les quantiles observés :

Tableau 4.18 Synthèse des résultats pour les méthodes quantiles

	$TL_{q\ demande}$				**$TL_{q\ intervalle}$**			
Méthode	**q=0.85**	**q=0.9**	**q=0.95**	**q=0.99**	**q=0.15**	**q=0.1**	**q=0.05**	**q=0.01**
Moyenne mobile 1	45.0	35.4	23.4	10.4	5.63	4.54	3.20	1.83
Lissage exponentiel simple 1	43.7	34.5	22.7	9.91	5.61	4.55	3.22	1.85
Moyenne mobile 2	39.1	31.3	21.2	10.2	5.86	4.75	3.41	2.10
Lissage exponentiel simple 2	38.8	30.8	20.4	9.12	5.54	4.44	3.13	1.82
Croston	39.7	31.9	21.8	10.8	6.25	5.19	3.91	2.60
Bootstrap 1	70.7	53.1	30.9	7.53	8.39	6.57	5.07	3.43
Bootstrap 2	36.8	29.6	20.1	9.95	3.95	3.04	2.03	0.740
Bootstrap 3	35.9	28.4	18.2	5.96	3.56	2.62	1.49	0.378
Bootstrap 4	**34.5**	**27.5**	**17.5**	**5.86**	**3.51**	**2.58**	**1.48**	**0.378**

Tableau 4.19 Synthèse des quantiles observés pour les méthodes quantiles

	Q demande				Q intervalle			
Méthode	**q=0.85**	**q=0.9**	**q=0.95**	**q=0.99**	**q=0.15**	**q=0.1**	**q=0.05**	**q=0.01**
Moyenne mobile 1	0.83	0.87	0.90	0.92	0.18	0.14	0.10	0.08
Lissage exponentiel simple 1	0.84	0.88	0.91	0.93	0.18	0.14	0.10	0.08
Moyenne mobile 2	0.80	0.84	0.89	0.92	0.23	0.18	0.13	0.10
Lissage exponentiel simple 2	0.82	0.86	0.90	0.93	0.21	0.17	0.12	0.09
Croston	0.81	0.85	0.89	0.92	0.21	0.17	0.12	0.09
Bootstrap 1	0.98	0.99	0.99	0.99	0.46	0.36	0.25	0.17
Bootstrap 2	0.76	0.81	0.86	0.91	0.18	0.15	0.11	0.06
Bootstrap 3	0.82	0.91	0.94	0.99	0.12	0.07	0.03	0.01
Bootstrap 4	0.83	0.91	0.94	0.99	0.12	0.07	0.03	0.01

La méthode sélectionnée est donc Bootstrap 4 avec un coefficient C tiré du tableau 4.16 et un coefficient α de 0.85.

4.3 Évaluer la meilleure méthode

À cette étape, on applique et évalue la méthode choisie sur des « données futures ». En fait, on utilise des données qui avaient été mises de côté lors de la préparation des données. Puisque cet ensemble de données tests n'a pas été utilisé lors de la sélection du modèle, les métriques ne seront pas biaisées et représenteront une bonne estimation des résultats qu'on obtiendra dans le futur. La méthode sélectionnée est Bootstrap 4, le coefficient α est 0.85 et le coefficient C est tiré du tableau 4.16.

4.3.1 Mesurer la précision des prévisions

En appliquant les étapes pour la validation croisée finale (détaillée en 3.3.1), on obtient les métriques d'erreurs et les quantiles suivants. :

Tableau 4.20 Résultats et quantiles observés avec Bootstrap 4 sur des données futures

	Demande				Intervalle			
	q=0.85	**q=0.9**	**q=0.95**	**q=0.99**	**q=0.15**	**q=0.1**	**q=0.05**	**q=0.01**
TL	37.9	30.2	19.8	7.09	3.89	2.88	1.68	0.453
Q	0.84	0.90	0.95	0.99	0.11	0.07	0.03	0.01

En prévision de la demande, pour un quantile q=0.85, l'erreur asymétrique « *Tick Loss* » (TL) est de 37.9 et le quantile observé (Q) est 0.84. En prévision de l'intervalle, pour un quantile q=0.01, l'erreur asymétrique « *Tick Loss* » (TL) est de 0.453 et le quantile observé (Q) est 0.01.

Les erreurs sont toutes légèrement plus élevées que celles obtenues avec la même méthode lors de la sélection. C'est normal. La validation croisée est une bonne méthode pour sélectionner des modèles et des paramètres; cependant, lorsqu'on sélectionne un modèle (ou des paramètres) en fonction des performances obtenues par validation croisée, ces performances sont surestimées. D'ailleurs, c'est la raison pour laquelle on a retiré des données tests avant de débuter la sélection des modèles.

Puisque les métriques sont comparables à celles vues lors de la sélection du modèle et des paramètres, on peut conclure que le modèle et les paramètres sélectionnés généralisent adéquatement, c'est-à-dire qu'ils s'adaptent aux nouvelles données. De plus, les résultats (qui ne sont pas montrés dans ce rapport) obtenus avec ces données tests, en utilisant les autres méthodes, sont tous de moins bonne qualité que ceux du tableau 4.20.

4.3.2 Tracer les courbes de rendement

Les courbes de rendement permettent de visualiser les impacts du choix du niveau de service. En logistique de collecte, certains termes et formules diffèrent par rapport à ce qui a été présenté dans la méthodologie.

D'abord, plutôt que de prédire un quantile de l'intervalle menant à une rupture de stock, on s'intéresse au quantile de l'intervalle menant à un remplissage complet du réservoir. En d'autres mots, on ne prédit pas un quantile de l'intervalle ($\hat{I}_q$) en fonction d'une demande égale au « stock après livraison » de la livraison précédente, mais on prédit un quantile de l'intervalle ($\hat{I}_q$) en fonction d'une demande égale à la capacité maximale du réservoir – le « stock après collecte » de la collecte précédente. Pour les niveaux de service qui ne sont pas inclus dans le tableau 4.16, la valeur du coefficient C est calculée par interpolation.

4.3.2.1 Nombre de collectes

Les formules utilisées ne sont pas les mêmes que celles de la méthodologie. En effet, pour ce cas d'étude, il ne serait pas raisonnable de supposer que les intervalles de collectes seraient appliqués aveuglément. De façon générale, lorsqu'un client voit que son réservoir est plein ou presque plein, il appelle l'entreprise et une commande urgente est créée. À la suite de la création d'une commande prioritaire, une collecte a lieu dans un délai maximal de 3 jours. Alors, il serait plus exact d'appliquer une borne supérieure sur l'intervalle simulé de cette façon :

$$Intervalle\ max_i = \begin{cases} \mathrm{I}_i & si\ \mathrm{S}_i + \mathrm{Q}_i \geq capacité\ max. \\ \mathrm{I}_i * \dfrac{capacité\ max. - \mathrm{S}_{i-1}}{\mathrm{U}_i} & sinon \end{cases} \quad (21)$$

$$Intervalle\ moyen_q = \frac{MIN(\hat{\mathrm{I}}_{1,q},\ Intervalle\ max_i) + \cdots + MIN(\hat{\mathrm{I}}_{N,q},\ Intervalle\ max_i)}{N} \quad (22)$$

$$Nombre\ annuel\ de\ collectes_q = \frac{365}{Intervalle\ moyen_q} \quad (23)$$

Sur la figure suivante, la courbe bleue représente le nombre annuel de collectes en fonction du niveau de service visé, alors que la droite rouge représente l'état actuel :

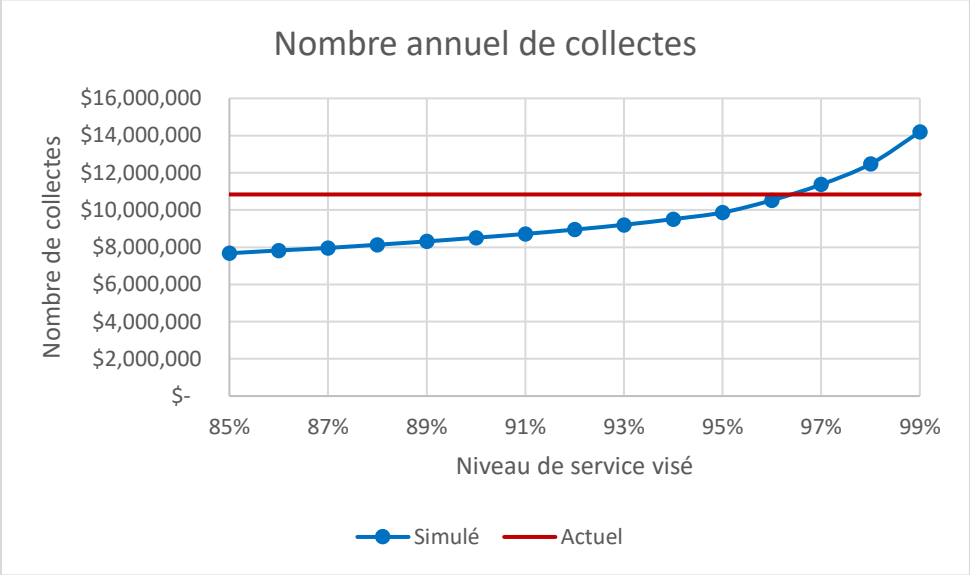

Figure 4.12 Nombre annuel de collectes en fonction du niveau de service visé

La courbe bleue montre clairement l'impact du choix du niveau de service visé : plus on veut augmenter le niveau de service, plus le nombre de collectes augmente. On remarque aussi que l'augmentation est exponentielle, c'est-à-dire que l'augmentation d'1% coute de plus en plus cher.

Pour que le point de comparaison soit adéquat, l'état actuel (droite rouge) est tiré des intervalles de l'ensemble de données tests. Si l'on appliquait les dates de collectes suggérées par l'algorithme, en

visant un niveau de service de 90%, on aurait besoin d'effectuer 46 500 collectes (21%) en moins par année.

4.3.2.2 Coûts en transport

L'information obtenue par l'entreprise est que le coût variable moyen d'une collecte (définie en 3.3.2) est de 50$. On peut donc se référer à la figure 4.12 et multiplier le nombre annuel de collectes pour obtenir le coût variable annuel en transport. Avec un niveau de service visé de 90%, l'application de l'algorithme pourrait donc représenter une économie d'environ 2 325 000$. Cela étant dit, puisque les collectes seront moins fréquentes et les quantités collectées plus grandes (en général), les répercussions sur le coût des tournées de véhicule sont difficiles à prévoir.

4.3.2.3 Niveau de service obtenu

Pour le cas d'étude, le niveau de service est défini comme étant la probabilité de ne pas atteindre la capacité maximale avant une collecte. Les formules utilisées sont :

$$Utilisation_{i,q} = \frac{\mathrm{U}_i}{\mathrm{I}_i} * \hat{\mathrm{I}}_{i,q} \tag{24}$$

$$Remplissage\ complet_{i,q} = \begin{cases} 1 & si\ \ Utilisation_{i,q} \geq Capacité\ max - \ S_{i-1} \\ 0 & sinon \end{cases} \tag{25}$$

Une fois qu'on a les valeurs pour chaque niveau de service visé et pour chaque donnée test, on fait la moyenne par niveau de service visé pour obtenir la probabilité de remplissage complet. Finalement, on calcule le niveau de service avec 1 - la probabilité de remplissage complet.

Sur la figure suivante, la courbe bleue représente le niveau de service obtenu en fonction du niveau de service visé, alors que la droite rouge représente l'état actuel et la droite grise représente le scénario optimal :

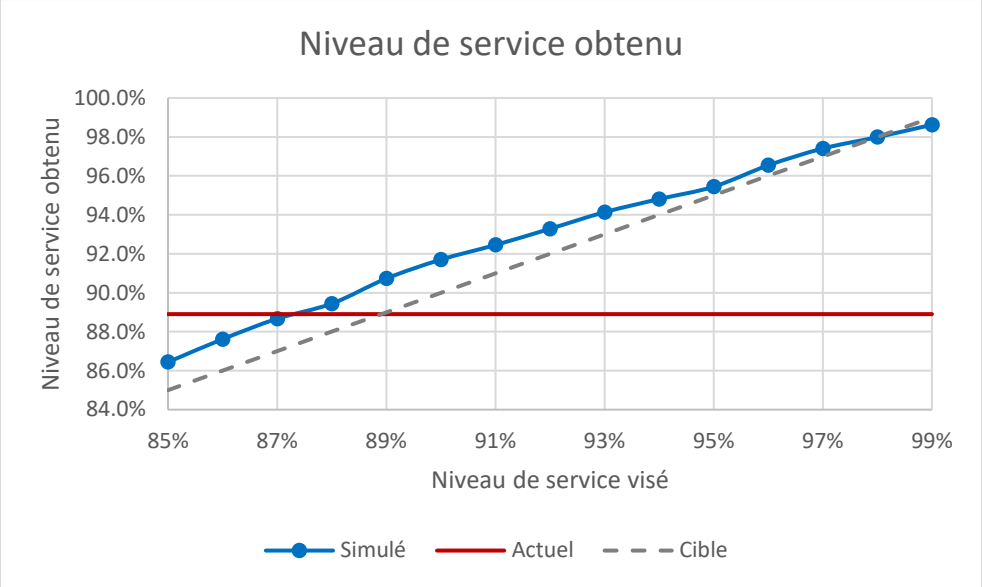

Figure 4.13 Niveau de service obtenu en fonction du niveau de service visé

La courbe bleue est très près du scénario cible, c'est-à-dire que le niveau de service obtenu est approximativement le niveau de service visé (± 1.7%). On pouvait déjà tirer cette conclusion à partir du tableau 4.20. Maintenant, on sait que la méthode développée Bootstrap 4 fonctionne aussi bien lorsqu'on l'applique pour effectuer des prévisions quantiles de l'intervalle en fonction d'une utilisation menant à un remplissage complet du réservoir. De plus, on sait que l'interpolation est une bonne méthode pour déterminer la valeur du coefficient C (pour le « *jittering* ») en fonction du niveau de service visé.

Le point de comparaison (droite rouge) est tiré de l'ensemble de données tests. Il n'aurait pas été adéquat de simplement prendre la proportion des commandes dont le « stock avant livraison » dépasse la capacité maximale du client. En effet, puisque les clients appellent généralement lorsque le réservoir est presque plein, les « débordements » sont souvent évités de justesse. Cependant, ces quasi-accidents sont mauvais pour la satisfaction des clients et causent des problèmes pour la planification (les routes doivent être modifiées à la dernière minute lorsque les clients appellent et qu'une commande urgente est ajoutée). La proportion de « débordements » (commandes dont le « stock avant livraison » dépasse la capacité maximale) est donc un indicateur biaisé. Alors, il est plus adéquat de prendre la proportion des commandes dont le « stock avant livraison » dépasse la capacité maximale du client, ou qui ont une priorité élevée. Ceci représente 11% des commandes, ce qui donne un niveau de service de 89%.

4.3.2.4 « Stock avant collecte » moyen

On utilise cette formule :

$$\text{« } Stock\ avant\ collecte \text{ »}_{i,q} = MIN(capacité\ max., \quad S_{i-1} + Utilisation_{i,q}) \qquad (26)$$

Sur les deux figures suivantes, la courbe bleue représente « stock avant collecte » moyen en fonction du niveau de service visé, alors que la droite rouge représente l'état actuel :

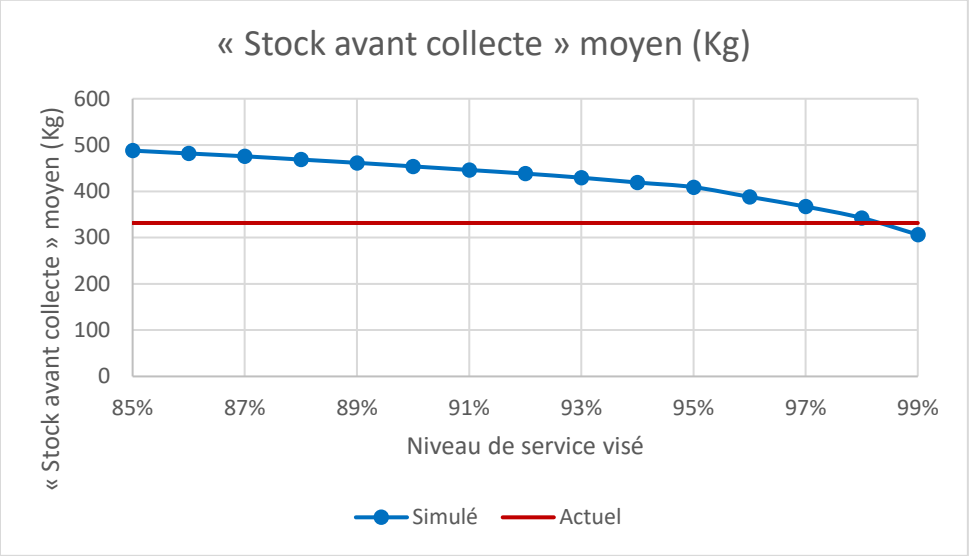

Figure 4.14 « Stock avant collecte » moyen en fonction du niveau de service visé

Sur celle-ci, le « stock avant collecte » est exprimé en pourcentage de la capacité maximale :

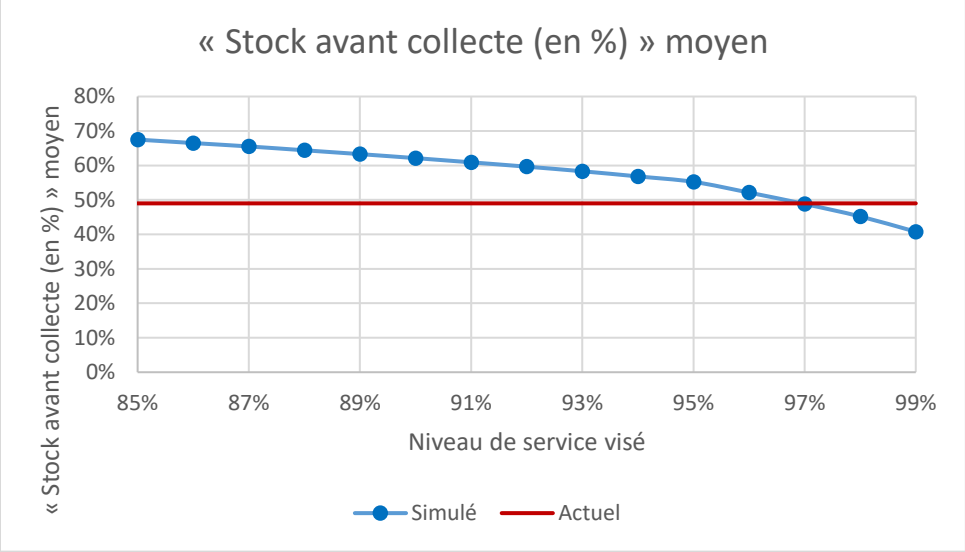

Figure 4.15 « Stock avant collecte (en %) » moyen en fonction du niveau de service visé

En logistique de collecte, le « stock avant collecte » est un indicateur de performance à maximiser. En effet, dans un scénario idéalisé, les collectes auraient lieu juste avant que la capacité maximale soit atteinte. La courbe bleue montre que plus on vise un niveau de service élevé, plus le stock à collecter diminue.

La droite rouge provient des données tests. Pour la figure 4.14, c'est simplement la moyenne du « stock avant collecte ». Pour la figure 4.15, les « stock avant collecte » sont divisés par la capacité maximale du réservoir avant de calculer la moyenne. Si l'on visait un niveau de service de 90% et qu'on appliquait les dates de collectes suggérées, les réservoirs collectés auraient 122 kg (ou 13%) de plus, en moyenne, par collecte.

Les courbes de rendements ont pu donner un aperçu des impacts de l'algorithme en fonction du niveau de service visé.

4.4 Synthèse

Pour la prévision déterministe de la demande, la « moyenne mobile (stratégie #2) », avec une fenêtre de 400 jours, minimise la MAE. Pour la prévision déterministe de l'intervalle, le « lissage exponentiel simple (stratégie #1) », avec α=0.3, minimise la MAE. Pour la prévision quantile de la demande, on emploie les quantiles [0.85, 0.9, 0.95, 0.99]; pour la prévision quantile de l'intervalle, on emploie les quantiles [0.15, 0.1, 0.05, 0.01]. Avec l'approche en deux étapes, c'est la méthode du « lissage exponentiel simple (stratégie #2) » qui minimise toutes les métriques d'erreur testées. Pour l'approche directe, on remarque, tel que prévu, que la méthode de la littérature est inadéquate. La moyenne de la distribution prédictive de la demande est surestimée alors que sa variance est sous-estimée. La première méthode développée pour ce mémoire, Bootstrap 2 procure une grande amélioration, mais la variance demeure sous-estimée. Bootstrap 3 permet de corriger le manque de variance. Bootstrap 4, en accordant légèrement plus de poids aux données récentes, est le modèle qui minimise toutes les métriques d'erreurs. Enfin, l'évaluation finale montre que le modèle et les paramètres sélectionnés s'adaptent correctement aux nouvelles données. Les courbes de rendements montrent la performance du modèle sélectionné par rapport à la planification actuelle et l'impact du choix du niveau de service. Avec le modèle sélectionné (Bootstrap 4), on pourrait réduire les coûts en transport en conservant le même niveau de service qu'actuellement. Si l'on

vise un niveau de service de 90% (comparativement à l'état actuel de 89%), on pourrait réduire le nombre annuel de collectes de 21%, soit une réduction de 46 500 collectes ou 2 325 000$ en coûts variables de transport.

CHAPITRE 5 CONCLUSION ET RECOMMANDATIONS

Ce mémoire portait sur l'optimisation des dates de livraisons en VMI à partir uniquement des données transactionnelles, c'est-à-dire sans connaître la demande ou l'utilisation des clients. Dans un modèle logistique VMI, le fournisseur doit prévoir la demande des clients pour gérer la planification des commandes de réapprovisionnement et respecter les niveaux de services visés. Pour ce travail, la définition employée du niveau de service est la probabilité de ne pas avoir de rupture de stock avant une livraison. L'objectif principal était de développer une méthode de prévision pour optimiser les dates de réapprovisionnement sous les contraintes des niveaux de service lorsque seulement les données transactionnelles sont disponibles. La méthode proposée répond aussi aux deux sous-objectifs : 1) elle permet de prédire un quantile de la demande accumulée en pendant un intervalle, en fonction de la durée de l'intervalle (par exemple, 7 jours) et du quantile considéré (par exemple, 0.5 : le 50^{e} centile, ou 0.9 : le 90^{e} centile) et 2) elle permet de prédire un quantile de la durée de l'intervalle, en fonction de la demande accumulée lors de cet intervalle (par exemple, 500 unités, ou 100% des unités restant : menant ainsi à une rupture de stock) et du quantile considéré.

La revue de littérature a présenté les domaines utiles pour ce travail, soit la prévision classique, la prévision probabiliste et la validation croisée. En prévision classique, un résultat est déterministe et vise à estimer une valeur vraisemblable. C'est pourquoi les métriques d'erreurs en prévision classique sont non biaisées, c'est-à-dire qu'une surestimation est pénalisée autant qu'une sous-estimation. La métrique d'erreur la plus appropriée au problème est la MAE. On a vu des méthodes générales (moyenne mobile, lissage exponentiel et autres variantes plus avancées) ainsi que des méthodes spécifiques à la demande intermittente (telles que Croston, ADIDA, MAPA et ASACT). En prévision probabiliste, un résultat est stochastique et la distribution prédictive permet de considérer tous les scénarios futurs envisageables. Le résultat attendu d'une prévision probabiliste peut aussi bien être une distribution prédictive complète (par exemple une loi normale ou une fonction de répartition quelconque), qu'un quantile de cette distribution prédictive. Lorsqu'on se concentre sur un (ou quelques) quantile(s), on parle alors de prévision quantile. En se focalisant sur un quantile en particulier, on peut éviter plusieurs problèmes et obtenir de meilleures prévisions là où ça compte vraiment. C'est pourquoi on prévoit seulement un quantile à la fois, soit celui qui correspond au niveau de service visé. En prévision quantile, la métrique d'erreur est

intentionnellement biaisée. Plus le quantile visé est extrême (loin de 0.5), plus la métrique est biaisée. De cette manière, un modèle qui minimise l'erreur est un prédicteur du quantile visé de la distribution prédictive.

Ce projet comportait deux obstacles à surmonter. Le premier obstacle est la nature de la source de donnée. L'information disponible est la somme de la demande réelle du client entre deux transactions. La demande réelle à chaque période est inconnue alors que la demande cumulative entre deux transactions est connue et intermittente. On peut dire que les données s'apparentent à celles d'une demande intermittente, mais avec certaines distinctions importantes. Les données disponibles n'étant pas directement exploitables par la majorité des méthodes de la littérature, il faut les adapter ou bien en créer de nouvelles. Par exemple, on peut supposer que la demande journalière est constante pendant l'intervalle inter-visite (et ainsi on peut appliquer les méthodes de prévision générales) ou on peut supposer que la demande accumulée est la demande réelle (et ainsi on peut appliquer les méthodes de prévision avec demande intermittentes). Cependant, on ne peut pas atteindre l'objectif sans franchir un autre obstacle. Le deuxième obstacle est la nature des résultats obtenus avec toutes les méthodes de prévision classiques. Avec ce type de prévision, le résultat est, en principe, la médiane de la distribution prédictive, ce qui veut dire qu'on peut s'attendre à sous-estimer la demande réelle 50% du temps. Alors, en se basant uniquement sur ces prédictions déterministes, on obtiendrait un niveau de service de 50%. Pour obtenir des niveaux de service plus élevés, la solution traditionnelle est d'utiliser un « stock de sécurité » calculé en ajustant une loi normale sur les données. Puisque les données sont bruitées, cette approche naïve donne de mauvaises performances. L'approche proposée est fondée sur la prévision quantile. Le moment de réapprovisionnement optimal est calculé en effectuant une prévision quantile de l'intervalle de temps menant à une rupture de stock, pour un quantile égal à 1 – le niveau de service.

La méthodologie couvre le flux de données. Premièrement, il y a la préparation des données. Dans cette étape, on montre 1) les données d'entrées nécessaires, 2) les calculs pour enrichir les données, 3) la détection et le traitement des données aberrantes et 4) la séparation des données d'apprentissage des données tests pour la validation croisée lors de l'évaluation finale. Deuxièmement il y a le choix de la méthode. Cette étape débute par l'exploration des méthodes de prévision classique, qui permet de découvrir quelles méthodes et quels paramètres minimisent la MAE. Cette information est utile plus tard en prévision quantile pour l'approche en deux étapes. Cette approche permet d'obtenir des prévisions quantiles en utilisant des méthodes de prévisions

classiques, et ce, sans poser d'hypothèse sur la distribution. L'approche directe, utilisant des méthodes de prévision quantile, est aussi proposée. Pour le choix de la méthode, l'indicateur asymétrique « *Tick Loss* » est recommandé. Finalement, l'évaluation de la méthode permet d'obtenir une bonne estimation de la performance qu'on obtiendra dans le futur avec la méthode sélectionnée. Les courbes de rendements montrent visuellement la performance du modèle en fonction de différents niveaux de service visés.

Le cas d'étude décrit le partenaire industriel et met en application la méthodologie dans le contexte d'une logistique de collecte. Puisque la logistique est inversée, de légères modifications sont apportées aux formules et aux termes de la méthodologie. En prévision classique, on remarque que les méthodes et paramètres minimisant la MAE pour la prévision de la demande ne sont pas les mêmes que ceux pour la prévision de l'intervalle. Dans le premier cas, la « moyenne mobile (stratégie #2) » avec une fenêtre de 400 jours et la méthode de Croston avec α=0.3 fonctionnent le mieux. Dans le deuxième cas, le « lissage exponentiel simple (stratégie #1) » avec α=0.3 et la « moyenne mobile (stratégie #1) » avec une fenêtre de 9 données fonctionnent le mieux. En prévision quantile, on a testé différentes combinaisons de méthodes et paramètres. Pour chacune, on a testé avec 4 différents quantiles, en fonction du niveau de service visé. Pour la prévision de la demande, on emploie les quantiles [0.85, 0.9, 0.95, 0.99]; pour la prévision de l'intervalle, on emploie les quantiles [0.15, 0.1, 0.05, 0.01]. Avec l'approche en deux étapes, c'est la méthode du « lissage exponentiel simple (stratégie #2) » qui minimise toutes les métriques d'erreur testées. Pour l'approche directe, on remarque, tel qu'on l'avait prévu, que la méthode de la littérature (surnommé Bootstrap 1) est inadéquate. La moyenne de la distribution prédictive de la demande est surestimée alors que sa variance est sous-estimée. La première méthode développée pour ce mémoire, Bootstrap 2, atténue ces problèmes, mais la variance est encore légèrement sous-estimée. Bootstrap 3, en introduisant le « *jittering* » , permet de corriger le manque de variance. Bootstrap 4, en accordant légèrement plus de poids aux données récentes, est le modèle qui minimise toutes les métriques d'erreurs. Enfin, l'évaluation finale montre que le modèle et les paramètres choisis généralisent bien, c'est-à-dire qu'ils s'adaptent correctement aux nouvelles données. Les courbes de rendements montrent concrètement la performance du modèle sélectionné, par rapport à la planification actuelle, et l'impact du choix du niveau de service. Actuellement, le niveau de service est de 89%. Avec le modèle sélectionné, on pourrait obtenir un niveau de service de 90% et réduire

le nombre annuel de collectes de 21%, soit une réduction de 46 500 collectes ou 2 325 000$ en coûts variables de transport.

Ce travail comporte certaines limites. Tout d'abord, les données à partir du mois de mars 2020 ont été retirées. Pour ce cas d'étude, on a préféré ne pas tenir compte des effets extrêmes de la pandémie sur la demande. Les métriques d'erreurs auraient inévitablement été plus élevées, et n'auraient pas représenté une situation normale et fidèle de la performance future à long terme. Cela étant dit, il serait intéressant, pour des travaux ultérieurs, de comparer l'efficacité de différents modèles à répondre aux fortes et soudaines variations de la demande. Par la suite, les courbes de rendement pour l'évaluation du modèle sont obtenues par simulation à partir de l'ensemble de données tests. Elles pourraient donc être sensibles aux valeurs aberrantes et aux manques de précision des données utilisées. Ensuite, les clients ayant moins de 7 données (44% des clients) ont été retirés de l'analyse. En principe, puisque 2 données par client avaient été réservées pour les validations croisées, la méthode s'applique sur les clients ayant au moins 5 données. Pour le reste (28%) des clients, on ne sait pas comment performent les méthodes. Une importante piste d'amélioration serait le démarrage à froid. On pourrait d'abord comparer la performance de différents modèles et paramètres en fonction du nombre de données disponibles. Puis, pour augmenter la précision des prévisions pour les jeunes clients, on pourrait essayer, par exemple, d'ajouter des variables exogènes, de segmenter ou bien d'utiliser les données de l'ensemble des clients pour effectuer une inférence bayésienne sur les paramètres de la distribution des jeunes clients. Une seconde piste d'amélioration serait le traitement des saisonnalités et des effets calendaires. Puisque peu de clients ont plusieurs années de données et que l'intermittence cache et brouille les comportements cycliques ou ponctuels, il serait sans doute insignifiant de chercher à traiter la saisonnalité des clients. Préférablement, on pourrait chercher la saisonnalité par groupes de clients. Par exemple : 1) Normaliser les données; 2) Segmenter les clients; 3) Agréger les données par jours, semaines ou mois; 4) Identifier les saisonnalités et les effets calendaires sur les séries agrégées (maintenant suffisamment longues et lisses); 5) Effectuer des prévisions sur les séries originales, à l'aide des saisonnalités dénormalisées; Enfin, 6) Optimiser, par validation croisée, la segmentation initiale. Une troisième piste d'amélioration serait de tester d'autres méthodes de prévisions. En effet, il existe une abondance d'outils de prévision.

RÉFÉRENCES

Barnett, V., & Lewis, T. (1994). Outliers in Statistical Data, -John Wiley and Sons. *New York*.

Bergmeir, C., & Benítez, J. M. (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences, 191*, 192-213.

Bollerslev, T. (1986). Glossary to ARCH (GARCH. in Volatility and Time Series Econometrics Essays in Honor of Robert Engle. MarkWatson, Tim Bollerslev and Jeffrey,

Bouchard, J., & Montreuil, B. (2011). *CIGI 2011 Prévisions journalières de séries temporelles saisonnières avec effets calendaires* Actes de congrès du 9e Congrès international de génie industriel, 12, 13 et 14 octobre 2011, Saint-Sauveur, Canada.

Box, G., & Jenkins, M. (1970). *Time series forecasting analysis and control*. Holden Day, San Francisco (USA).

Boylan, J. E., & Syntetos, A. A. (2006). Accuracy and accuracy-implication metrics for intermittent demand. *Foresight: The International Journal of Applied Forecasting, 4*, 39-42.

Breiman, L. (2001). Random forests. *Machine learning, 45*(1), 5-32.

Brown, R. G. (1959). *Statistical forecasting for inventory control*. McGraw/Hill.

Croston, J. D. (1972). Forecasting and stock control for intermittent demands. *Journal of the Operational Research Society, 23*(3), 289-303.

Efron, B. (1992). Bootstrap methods: another look at the jackknife. In *Breakthroughs in statistics* (pp. 569-593). Springer.

Ferbar Tratar, L. (2015). Forecasting method for noisy demand. *International Journal of Production Economics, 161*, 64-73.

Fragoso, T. M., Bertoli, W., & Louzada, F. (2018). Bayesian Model Averaging: A Systematic Review and Conceptual Classification. *International Statistical Review, 86*(1), 1-28.

Frederick, A. (2003). *Effective mesoscale, short-range ensemble forecasting* Ph. D. dissertation, University of Washington].

Ghobbar, A. A., & Friend, C. H. (2003). Evaluation of forecasting methods for intermittent parts demand in the field of aviation: a predictive model. *Computers & Operations Research, 30*(14), 2097-2114.

Giard, V. (2003). *Gestion de la production et des flux*. Economica Paris.

Gneiting, T. (2008). Editorial: Probabilistic Forecasting. *Journal of the Royal Statistical Society. Series A (Statistics in Society), 171*(2), 319-321.

Gneiting, T. (2011). Quantiles as optimal point forecasts. *International Journal of Forecasting, 27*(2), 197-207.

Gneiting, T., & Katzfuss, M. (2014). Probabilistic forecasting. *Annual Review of Statistics and Its Application, 1*, 125-151.

Gneiting, T., Raftery, A. E., Westveld, A. H., III, & Goldman, T. (2005). Calibrated Probabilistic Forecasting Using Ensemble Model Output Statistics and Minimum CRPS Estimation. *Monthly Weather Review, 133*(5), 1098-1118.

Grimit, E. P., & Mass, C. F. (2002). Initial Results of a Mesoscale Short-Range Ensemble Forecasting System over the Pacific Northwest. *Weather and Forecasting, 17*(2), 192-205.

Hamill, T. M., & Colucci, S. J. (1998). Evaluation of Eta–RSM Ensemble Probabilistic Precipitation Forecasts. *Monthly Weather Review, 126*(3), 711-724.

Hasni, M., Aguir, M. S., Babai, M. Z., & Jemai, Z. (2019). On the performance of adjusted bootstrapping methods for intermittent demand forecasting. *International Journal of Production Economics, 216*, 145-153.

Hodge, V., & Austin, J. (2004). A survey of outlier detection methodologies. *Artificial intelligence review, 22*(2), 85-126.

Holt, C. C. (1957). Forecasting trends and seasonal by exponentially weighted moving averages. *ONR Memorandum, 52*.

Houtekamer, P. L., Lefaivre, L., Derome, J., Ritchie, H., & Mitchell, H. L. (1996). A System Simulation Approach to Ensemble Prediction. *Monthly Weather Review, 124*(6), 1225-1242.

Jin, Y. H., Williams, B. D., Tokar, T., & Waller, M. A. (2015). Forecasting With Temporally Aggregated Demand Signals in a Retail Supply Chain. *Journal of Business Logistics, 36*(2), 199-211.

John, O. O. (2015). Robustness of quantile regression to outliers. *American Journal of Applied Mathematics and Statistics, 3*(2), 86-88.

Kim, S., & Kim, H. (2016). A new metric of absolute percentage error for intermittent demand forecasts. *International Journal of Forecasting, 32*(3), 669-679.

Koenker, R., & Bassett, G. (1978). Regression Quantiles. *Econometrica, 46*(1), 33-50.

Koenker, R., & Bassett Jr, G. (1978). Regression quantiles. *Econometrica: journal of the Econometric Society*, 33-50.

Koenker, R., & Hallock, K. F. (2001). Quantile regression. *Journal of economic perspectives, 15*(4), 143-156.

Kourentzes, N. (2014). On intermittent demand model optimisation and selection. *International Journal of Production Economics, 156*, 180-190.

Kourentzes, N., Petropoulos, F., & Trapero, J. R. (2014). Improving forecasting by estimating time series structural components across multiple frequencies. *International Journal of Forecasting, 30*(2), 291-302.

Madsen, H. (2007). *Time series analysis*. CRC Press.

Meinshausen, N. (2006). Quantile regression forests. *Journal of Machine Learning Research, 7*(Jun), 983-999.

Metropolis, N., Rosenbluth, A. W., Rosenbluth, M. N., Teller, A. H., & Teller, E. (1953). Equation of state calculations by fast computing machines. *The journal of chemical physics, 21*(6), 1087-1092.

Metropolis, N., & Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association, 44*(247), 335-341.

Molteni, F., Buizza, R., Palmer, T. N., & Petroliagis, T. (1996). The ECMWF ensemble prediction system: Methodology and validation. *Quarterly journal of the royal meteorological society, 122*(529), 73-119.

Mosteller, F., & Rourke, R. E. (1973). Sturdy statistics: nonparametrics and order statistics.

Murray, P. W., Agard, B., & Barajas, M. A. (2017). Market segmentation through data mining: A method to extract behaviors from a noisy data set. *Computers & Industrial Engineering, 109*, 233-252.

Murray, P. W., Agard, B., & Barajas, M. A. (2018). ASACT - Data preparation for forecasting: A method to substitute transaction data for unavailable product consumption data. *International Journal of Production Economics, 203*, 264-275.

Nikolopoulos, K., Syntetos, A. A., Boylan, J. E., Petropoulos, F., & Assimakopoulos, V. (2011). An aggregate–disaggregate intermittent demand approach (ADIDA) to forecasting: an empirical proposition and analysis. *Journal of the Operational Research Society, 62*(3), 544-554.

Petropoulos, F., Kourentzes, N., & Nikolopoulos, K. (2016). Another look at estimators for intermittent demand. *International Journal of Production Economics, 181*, 154-161.

Prestwich, S. D., Tarim, S. A., Rossi, R., & Hnich, B. (2014). Forecasting intermittent demand by hyperbolic-exponential smoothing. *International Journal of Forecasting, 30*(4), 928-933.

Raftery, A. E., Gneiting, T., Balabdaoui, F., & Polakowski, M. (2005). Using Bayesian Model Averaging to Calibrate Forecast Ensembles. *Monthly Weather Review, 133*(5), 1155-1174.

Raiffa, H., & Schlaifer, R. (1961). Applied statistical decision theory.

Richardson, D. S. (2001). Measures of skill and value of ensemble prediction systems, their interrelationship and the effect of ensemble size. *Quarterly journal of the royal meteorological society, 127*(577), 2473-2489.

Sloughter, J. M. L., Raftery, A. E., Gneiting, T., & Fraley, C. (2007). Probabilistic Quantitative Precipitation Forecasting Using Bayesian Model Averaging. *Monthly Weather Review, 135*(9), 3209-3220.

Syntetos, A. A., & Boylan, J. E. (2005). The accuracy of intermittent demand estimates. *International Journal of Forecasting, 21*(2), 303-314.

Syntetos, A. A., Boylan, J. E., & Croston, J. D. (2005). On the categorization of demand patterns. *Journal of the Operational Research Society, 56*(5), 495-503.

Syntetos, A. A., Zied Babai, M., & Gardner, E. S. (2015). Forecasting intermittent inventory demands: simple parametric methods vs. bootstrapping. *Journal of Business Research, 68*(8), 1746-1752.

Taillardat, M., Mestre, O., Zamo, M., & Naveau, P. (2016). Calibrated Ensemble Forecasts Using Quantile Regression Forests and Ensemble Model Output Statistics. *Monthly Weather Review, 144*(6), 2375-2393.

Taylor, J. W. (2003). Short-term electricity demand forecasting using double seasonal exponential smoothing. *Journal of the Operational Research Society, 54*(8), 799-805.

Teunter, R. H., Syntetos, A. A., & Babai, M. Z. (2011). Intermittent demand: Linking forecasting to inventory obsolescence. *European Journal of Operational Research, 214*(3), 606-615.

Toth, Z., & Kalnay, E. (1997). Ensemble Forecasting at NCEP and the Breeding Method. *Monthly Weather Review, 125*(12), 3297-3319.

Trapero, J. R., Cardós, M., & Kourentzes, N. (2019). Quantile forecast optimal combination to enhance safety stock estimation. *International Journal of Forecasting, 35*(1), 239-250.

Viswanathan, S., & Zhou, C. (2008). A new bootstrapping based method for forecasting and safety stock determination for intermittent demand items. In *Nanyang Business School, Nanyang Technological University Singapore*.

Willemain, T. R., Smart, C. N., & Schwarz, H. F. (2004). A new approach to forecasting intermittent demand for service parts inventories. *International Journal of Forecasting, 20*(3), 375-387.

Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management science, 6*(3), 324-342.

Zhou, C., & Viswanathan, S. (2011). Comparison of a new bootstrapping method with parametric approaches for safety stock determination in service parts inventory systems. *International Journal of Production Economics, 133*(1), 481-485.