Prédiction des Séries Temporelles Larges

Youssef Hmamouche

AIX-MARSEILLE UNIVERSITÉ
ÉCOLE DOCTORALE DE MATHÉMATIQUE ET INFORMATIQUE DE MARSEILLE

LABORATOIRE D'INFORMATIQUE ET DES SYSTÈMES / UMR 7020

Thèse présentée pour obtenir le grade universitaire de docteur

Discipline : Science
Spécialité : Informatique

Youssef HMAMOUCHE

Prédiction des Séries Temporelles Larges

Prediction of Large Time Series

Soutenue le 13/12/2018 devant le jury composé de :

Nadine HILGERT	INRA	Rapporteure
Pietro MICHIARDI	Eurecom	Rapporteur
Rosine CICCHETTI	Aix Marseille Université	Examinatrice
Cyril De RUNZ	Université de Reims Champagne-Ardenne	Examinateur
Lotfi LAKHAL	Aix Marseille Université	Directeur de thèse
Alain CASALI	Aix Marseille Université	Co-Directeur de thèse

Résumé

De nos jours, les systèmes de gestion et de traitement des données sont censés stocker et traiter de grandes séries temporelles. Comme le nombre de variables observées et liées augmente, leur prédiction est devenue de plus en plus compliquée, et l'utilisation de toutes les variables pose des problèmes pour les modèles de prédiction classiques.

Dans le domaine de la prédiction de séries temporelles massives, l'un des principaux problèmes qui se posent dans les applications réelles, est comment déterminer l'ensemble de variables prédictives les plus pertinentes, qui vont être utilisées dans un modèle de prédiction multi-varié en vue de prédire une série temporelle cible.

Les modèles de prédiction sans facteurs externes étaient les premières modèles de prédiction des données. En vue d'améliorer la precision des prédictions de ces modèles, l'utilisation de multiples variables est devenue commune. Ainsi, les modèles qui tiennent en compte des facteurs externes, ou bien les modèles multivariés, apparaissent et deviennent de plus en plus utilisés car ils prennent en compte plus d'informations que les modèles uni-variés.

Avec l'augmentation des données liées entre eux, l'application des modèles multi-variés devient aussi discutable. Parce que l'utilisation de toutes les informations existantes n'amène pas forcément aux meilleures prédictions. Par conséquent, le challenge dans cette situation est de trouver les facteurs les plus pertinents à partir de l'ensemble des données originales.

Dans ce travail, nous étudions ce problème en présentant une analyse détaillée des approches proposées dans la littérature. Nous remarquons que ce qui intéressant dans cette problématique est l'aspect temporel lors de l'analyse des dépendances entre les variables, et cela est absent dans beaucoup de méthodes bien connues. Nous proposons donc une nouvelle démarche pour approcher ce problème basée sur la sélection des variables en utilisant les graphes de causalité, avec des techniques d'exploration de ces graphes en vue de détecter les variables « *Hubs* » qui transfèrent l'information par rapports aux autres variables prédictives et aux variables à prédire. Finalement, nous présentons une implémentation d'un processus complet pour la prédiction des séries temporelles larges, et comme perspectives, nous présentons une première proposition pour distribuer ce système de prédiction.

Mots clés : Prédiction des Séries temporelles, Apprentissage automatique, Causalité prédictive, Réduction de dimension, Sélection des variables

Abstract

Nowadays, data management and processing systems are supposed to store and process large time series. As the number of observed and related variables increases, their prediction becomes more and more complicated, and the use of all variables poses problems for conventional prediction models.

In the field of massive time series prediction, one of the main problems in real applications is how to determine the most relevant set of predictive variables, which will be used in a multivariate prediction model, to predict a target time series.

Prediction models without external factors were the first data prediction models. In order to improve the accuracy of prediction, the use of multiple variables has become common. So, models that take external factors into account, or multivariate models, appear, and become more and more used because they take into account more information than univariate models.

With the increase in interrelated data, the application of multivariate models is also becoming questionable. Because the use of all existing information does not necessarily lead to the best predictions. Therefore, the challenge in this situation is to find the most relevant factors among the available data set, in order to predict a given time series.

In this work, we study this problem by presenting a detailed analysis of the proposed approaches in the literature. We note that what is interesting in this issue is the temporal aspect when analyzing the dependencies between the variables, and this is absent in many well-known methods. We propose a new approach to tacle this problem based on the selection of the variables by using the causality graphs, with techniques of exploration of these graphs in order to detect the "Hubs" variables which transfer the information relative to other predictors and variables to predict. Finally, we present an implementation of a complete process for the prediction of large time series, and as perspectives, we present a first proposition to distribute this prediction system.

Keywords: Time Series Prediction, Machine learning, Predictive causality, Variable Selection, Dimension Reduction

Remerciements

J'exprime mes sincères remerciements à mon directeur de thèse M. Lotfi Lakhal et mon co-directeur de thèse M. Alain Casali, pour l'aide compétente qu'ils m'ont apporté, pour la patience et l'encouragement à accomplir ce travail. Leurs remarques ont été toujours critiques et précieuses pour améliorer la qualité des différents aspects du travail. Je leurs remercie particulièrement pour les discussions intéressantes et enrichissantes que nous avons mené sur les différents aspects de l'informatique à part du sujet de la thèse, et aussi pour leur aide en ce qui concerne l'aspect administratif de la thèse.

J'adresse mes remerciements à Mme. Nadine HILGERT et M. Pietro MICHIARDI pour avoir accepter d'être rapporteurs de cette thèse. Je tiens à remercier aussi Mme. Rosine CICCHETTI et M. Cyril De RUNZ qui ont bien voulu être examinateurs.

Je remercie la directrice de l'École Doctorale en Mathématiques et Informatique de Marseille, Mme. Nadia Creignou et les anciens directeurs du Laboratoire d'Informatique Fondamentale (LIF), M. Jean Marc Talbot et M. Liva Ralaivola pour le côté administratif de la thèse et le suivi du déroulement des travaux de la thèse.

Pour finir, merci à tous les membres du département d'informatique de l'IUT d'Aix en Provence, l'équipe des Bases de Données Avancées (BDA) du Laboratoire d'Informatique et des Systèmes (LIS) de l'Université d'Aix Marseille pour m'avoir accueilli pendant cette thèse.

Table des matières

Liste des figures

Liste des tableaux

Introduction

Contexte et Objectif

De nos jours, des quantités énormes de séries temporelles sont générées par l'industrie et les systèmes de recherche, pour diverses applications, comme la biologie, la médecine, les finances, l'industrie et bien d'autres. Les systèmes modernes d'analyse de données sont censés traiter et stocker des millions de séries temporelles de grande dimension, générant des téraoctets de données. En conséquence, le nombre d'attributs générés augmente très rapidement et relève de la catégorie *Big Data* ; c'est-à-dire que cette situation crée des problèmes dans lesquels la dimension des données est un problème lui-même.

Parmi les nombreuses applications, la prédiction des séries temporelles prend une place particulière car elle est cruciale pour la prise de décision. Sans surprise, c'est souvent une partie importante des systèmes de *Business Intelligence*. L'histoire des modèles de prédiction commence dans les années 1930, avec l'apparition de l'application des premiers modèles uni-variés sur des données. L'idée de ces modèles est simple : l'historique d'une variable est utilisé pour faire des prévisions, en se basant sur des modélisations qui permettent d'adapter les données à un modèle auto-régressif, par exemple les modèles *Auto-Regressive* (AR) et *Auto-Regressive Integrated Moving Average* (ARIMA) BOX 2013. L'approche univariée présente des inconvénients évidents, car elle ignore les informations potentiellement exploitables d'autres séries temporelles. Les modèles de prédiction multivariés supposent que la valeur d'une variable dépend de valeurs précédentes d'elle-même et des autres variables, par exemple le modèle Vector Auto-Regressive (VAR), Vector Error Correcion (VEC) JOHANSEN 1988. Ces modèles sont largement utilisés, seuls ou combinés avec d'autres techniques comme la modélisation dynamique avec les réseaux de neurones artificiels, par exemple le modèle Vector Auto-Regressive Neural Network TRAPLETTI, LEISCH et HORNIK 2000.

Ce travail porte sur l'étude et le développement des modèles de prédiction pour des séries temporelles caractérisées par un large nombre de variables. Il est réalisé dans le contexte du projet *e-Business Optimization with Big Data* (eBOB), qui est un projet d' Investissements d' Avenir et du Développement de l'Économie Numérique. Il a pour objectif de révolutionner la gestion des achats en entreprises en créant de nouvelles technologies d'analyse des données mêlant les données structurées et non structurées permettant la modification fonctionnelle dynamique de l'outil selon le contexte résultant des analyses. Les données structurées (bases de données utilisateurs) permettront d'avoir des informations sur la gestion interne des achats de l'entreprise, combinées aux données non structurées (internet) et à la plateforme business. L'étude de ces données permettra une analyse des tendances en prenant en compte la situation de la société mais éga-

lement la réalité du marché. Cette analyse, optimisée par les multiples sources d'informations, a pour objectif d'apporter une aide à la décision pour les utilisateurs en les guidant dans leurs choix (modification dynamique).

Étant donné des bases de données massives, contenant des valeurs de :

— prix articles (fournisseurs)
— matières premières
— indices internationaux

L'objectif final du projet est de :

— déterminer la meilleure période d'achat en se basant sur les prédictions des articles.

Les principaux travaux de recherche associés à ce travail se focalisent sur les processus de réduction et prédiction des séries temporelles. Notre premier objectif est d'étudier le fonctionnement de différents modèles de prédiction (modèles statistiques et réseaux de neurones artificiels), ainsi que les approches de la littérature qui abordent le sujet de la prédiction des séries temporelles larges. Le deuxième objectif est de proposer des algorithmes spécifiques à la problématique industrielle du projet eBOB pour améliorer les performances des méthodes existantes. Finalement, l'aspect pratique concerne le développement d'un système de réduction et de prédiction des séries temporelles larges basé sur ces algorithmes.

Liste des notations

Nous présentons ici les notations mathématiques utilisées dans de ce manuscrit. Pour les fonctions et les opérations matricielles, nous utilisons les notations classiques :

— $A^\intercal$ est la matrice transposée de la matrice A.
— A^{-1} est la matrice inverse de la matrice A.
— AB est le produit matricielle des de A et B.
— $f'(x)$ est la dérivée de la fonction $f(x)$.

Puisque nous allons traiter beaucoup les modèles des séries temporelles, nous considérons une notation spécifique pour ce type de données. Nous désignons une série temporelles multivariée comme un ensemble de séries uni-variées. Et chaque série uni-varié est un vecteur d'observations indexés par le temps.

Soit X une séries temporelle multi-varié de dimension k, (elle contient k variables) Et chaque série X_i contient n observations.

— On écrit la série temporelle X comme suit : $X = \{X_1, X_2, \ldots, X_k\}$. On peut la considérer aussi comme une matrice de taille $(k \times n)$, c-à-d., un vecteur de séries uni-variées.
— $X_i = (X_i(1), \ldots, X_i(n))^\intercal$, pour tout $i \in \{1, \ldots, k\}$.
— $X_i(t)$ est l'observation à l'instant t de la variable X_i, pour tout $i \in \{1, \ldots, k\}$, et pour tout $t \in \{1, \ldots, n\}$.

— $X(t)$ est l'ensemble des observations de toutes les variables de la séries temporelles multivariée X à l'instant t.

$$X(t) = (X_1(t), X_2(t), \ldots, X_k(t)).$$

— $\triangle X_i(t) = X_i(t) - X_i(t-1)$.
— $\triangle^d X_i(t) = \triangle(\triangle^{d-1} X_i(t))$.
— Dans le cas où X comme une série uni-varié, alors les notations utilisées pour les X_i sont applicables à X dans ce cas.
— Nous écrivons aussi une série temporelle (uni-variée ou multivariée) qui contient n observations comme suit : $\{X, t \in [1, n]\}$.

La table 0.1 montre quelques abréviations qu'on va utiliser également dans ce manuscrit.

Abréviation	Signification
AR	Auto-Régressif
$ARIMA$	Auto-Régressif Intégré à Moyenne mobile
Σ	matrice de covariance
VAR	Vecteur Auto-Régressif
$VECM$	Modèle Vecteur à Correction d'Erreur
ACP	Analyse en Composantes Principales
PCA	Analyse en Composantes Principales
AF	Analyse Factorielle
$KernelPCA$	Analyse en Composantes Principales à noyaux
$FCBF$	*Fast Correlation-Based Filter*
RNA	Réseau de Neurone Artificiel
$LSTM$	Réseau récurrent à Mémoire Court et Long terme
$RMSE$	Erreur quadratique moyenne
$MASE$	Erreur moyenne absolue mise à l'échelle
Ridge	Régression d'arête (*ridge regression*)
$LARS$	Régression par les moindres angles
Lasso	*Least Absolute Shrinkage and Selection Operator*
$GFSM$	*Granger Feature Selection Method*
PAM	*Partitionning Arround Medoids*
$PEHAR$	*Predictors Extraction Based On Hubs and Authorities Ranking*
PV	Photovoltaique

Table 0.1 – Abréviations.

Organisation

Dans le chapitre 1, nous présentons le contexte théorique ; des généralités sur les séries temporelles et les modèles de prédiction. Dans le chapitre 2, nous présentons les travaux de la littérature qui traitent le sujet de prédiction des séries temporelles multivariées contenant plusieurs variables. Bien que ces approches sont souvent des méthodes composées de plusieurs techniques, nous discutions toutes ces composantes, à savoir, les méthodes de réduction de dimension, les méthodes de sélection des variables, et d'autres modèles de prédiction des données larges. Dans le chapitre 3, nous présentons nos contributions, les algorithmes proposés, et les expérimentations réalisées, et nous discutons ces résultats. Et finalement, dans le dernier chapitre, nous présentons une conclusion générale de nos contributions, et des perspectives de ces travaux.

1 Contexte théorique

Ce chapitre est consacré au contexte théorique en ce qui concerne la prédiction des séries temporelles multi-variées. Nous présentons d'abord des généralités sur les séries temporelles, puis nous présentons, de manière séparée, les modèles de prédiction et les méthodes de réduction de volume de données.

Nous étudions dans un premier temps le principe de quelques modèles de prédiction parmi les plus utilisés dans le cadre des séries temporelles de type numériques. Nous présentons ainsi les modèles statistiques et les modèles basés sur l'apprentissage par les réseaux de neurones artificiels (RNAs). Nous discutons aussi leurs limitations et leurs avantages. Et dans un deuxième temps, nous étudions les méthodes de réduction des données, à savoir les méthodes de dimension de réduction et les méthodes de sélection de variables

1.1 Généralités sur les Séries Temporelles

Dans cette section, nous présentons une introduction générale sur les séries temporelles. Une série temporelle est un ensemble d'observations où chacune est associée à un instant de temps unique t. Une série temporelle peut aussi être construit en rassemblant un nombre fini d'observations. Dans ce cas, on peut la représenter par un vecteur de n variables aléatoires où les réalisations sont les observations réelles. Il est possible de considérer le nombre d'observations comme un nombre infini, dans ce cas, on parle des processus stochastiques discrets. Dans la pratique, la première représentation est la plus utilisée, où les séries temporelles sont représentées par des vecteurs (indexés par le temps) d'observations successives.

Il est aussi très important de considérer multiples séries temporelles en même temps, vue que dans la pratique, les variables modélisées par les séries temporelles sont souvent en liaison avec d'autres variables. Une série temporelle multivariée $X = \{X_1, \ldots, X_n\}$ est un ensemble fini de séries uni-variées ayant des valeurs qui évoluent en même temps, c'est à dire qu'on peut les indexer par un seul vecteur de temps. Les séries temporelles multi-variées sont très utiles dans la modélisation des systèmes, surtout dans le cas où les données sont captées simultanément, par exemple les données boursières (les prix de plusieurs articles et indices), et les données de neurosciences pour mesurer les activités électriques du cerveau (électroencéphalogramme), *etc*.

1.1.1 Les Processus stochastiques discrets

Un processus stochastique (aléatoire) discret est une séquence de variables aléatoires $\{X(t), t \in T\}$ définit sur un espace de probabilité. Dans ce cas, la série

temporelles uni-varié $X = \{X(1), \ldots, X(|T|)\}^\intercal$, peut être vue comme la réalisation de ce processus stochastique. Mais en général, une série temporelle est considérée en même temps comme un processus stochastique et sa réalisation.

Nous présentons quelques processus discrets classiques, qu'on va utiliser dans suite, notamment dans les modèles de prédiction. Commençons par le plus simple ; le processus Gaussien.

Définition 1 (processus Gaussien)**.** *Un processus $(X, t \in \mathbb{Z})$ Gaussien est une séquence de variables aléatoires suivant une loi normal.*

Définition 2 (Bruit blanc)**.** *Un processus bruit blanc est une séquence de variables aléatoires non-corrélés entre eux, de moyenne nulle et de variance constante. Un processus $(X, t \in \mathbb{Z})$ est dit bruit si :*

$$\begin{aligned} E[X(t)] &= 0 \\ VAR[X] &= \sigma^2 \\ E[X(r,s)] &= 0 \quad \forall r \neq s \end{aligned}$$

où $E[X]$ la fonction moyenne, VAR est la variance de X_t, et $E[X(r,s)]$ est l'auto-covariance, avec :

$$E[X(r,s)] = E[(X(r))E(X(s))]. \tag{1.1}$$

Définition 3 (Auto-regressif)**.** *Un processus $(X, t \in \mathbb{Z})$ Auto-régressif est une séquence de variables aléatoires stationnaires, où chaque valeur dépend de la valeur précédente et d'un terme d'erreur qui est un processus de type bruit blanc.*

$$X(t) = \alpha X(t-1) + U(t)$$

où α est le paramètre du processus, et U est un processus bruit blanc. Ce processus est stationnaire ssi $|\alpha < 1$.

Définition 4 (Auto-régressif d'ordre p)**.** *Un processus $(X, t \in \mathbb{Z})$ Auto-régressive d'ordre p (AR (p)) est une séquence de variables aléatoires, où chaque valeur dépend des p valeurs précédentes et d'un terme d'erreur suivant un bruit blanc.*

$$X(t) = \alpha_1 X(t-1) + \cdots + \alpha_p X(t-p) + U(t)$$

où $\{\alpha_1, \ldots, \alpha_p\}$ sont les paramètres du processus, et $U(t)$ est un processus bruit blanc. Ce processus est stationnaire si les racines de l'équation $1 - \alpha_1 x - \cdots - \alpha_p x^p = 0$ sont strictement supérieures (en valeur absolue) à 1.

1.1.2 La Stationnarité

La stationnarité est une caractéristique très intéressante dans l'analyse des séries temporelles. Elle reflète en quelque sorte la stabilité des observations par

rapport au temps. Un processus stochastique est stationnaire, si ses propriétés statistiques ne changent pas dans le temps. Mathématiquement parlant, il y a deux types de définition de la stationnarité. La stationnarité stricte, et la stationnarité faible. Nous discutions ici la différence entre ces deux définitions.

Définition 5 (stationnarité stricte)**.** *Une série temporelle $(X, t \in \mathbb{Z})$ suit un processus strictement stationnaire si et seulement si :*
$(X(t_1), \ldots, X(t_k))$ *et* $(X(t_1 + h), \ldots, X(t_k + h))$ *ont des distributions identiques pour tout* $h, t_1, \ldots t_k, \in \mathbb{Z}$.

Définition 6 (stationnarité faible)**.** *Une série temporelle $(X, t \in \mathbb{Z})$ est faiblement stationnaire si*

$$Var(X(t)) < \infty \quad \forall t \in \mathbb{Z} \tag{1.2}$$
$$\mu(t) = \mu(t + r) \quad \forall t, r \in \mathbb{Z} \tag{1.3}$$
$$\lambda(r, s) = \lambda(r + t, s + t) \quad \forall r, s, t \in \mathbb{Z} \tag{1.4}$$

où $\mu(t) = E[X(t)]$ est la fonction moyenne, Var est la fonction variance, et $\lambda(r, s)$ est la fonction auto-covariance :

$$\lambda(r, s) = E[(X(r) - \mu(r))(X(s) - \mu(s))]. \tag{1.5}$$

La stationnarité stricte est généralement difficile à considérer dans la réalité, car parfois on connais pas exactement la distributions des observations. Pratiquement, les propriétés statistiques des variables prisent ont compte sont la moyenne et la covariance. Notons que pour les séries temporelles Gaussiennes, la stationnarité est équivalente à la stationnarité faible, car dans ce cas, la distribution peut être définie par sa moyenne et sa covariance.

La définition de stationnarité faible implique que les observations d'une série temporelles varient avec une variance finie autour d'une seule moyenne et il n y a pas d'auto-corrélation entre les observations (*cf.* Définition 6). Par exemple, on peut remarquer que le processus bruit blanc est stationnaire, en plus il a une moyenne nulle. L'effet de non-stationnarité se voit aussi graphiquement sur les courbes l'évolution des observations. Par exemple, même si la variance soit finie, si la moyenne des observations change, ou si les observations sont auto-corrélées, alors le processus est non-stationnaire.

Une méthode pour rendre les séries temporelles stationnaires consiste les différentier. Supposons une série uni-varié $\{X, t \in [1, n]\}$. La différentiation de X d'ordre 1 est définit comme suit :

$$\triangle X(t) = X(t) - X(t - 1). \tag{1.6}$$

Et La différentiation de X d'ordre d est définit comme suit :

$$\triangle^d X(t) = \triangle(\triangle^d X(t)). \tag{1.7}$$

Définition 7. *Soit X une série temporelle uni-variée. X est dite intégré d'ordre d si elle est non stationnaire et devient stationnaire juste après d différentiations, c'est-à-dire lorsque $\triangle^d X$ est stationnaire alors que $\triangle^{d-1} X$ ne l'est pas.*

De même, la stationnarité s'applique aussi au séries temporelles multi-variées. Une série temporelle multi-variée est stationnaire si toutes ses variables sont stationnaires, et dite non-stationnaire d'ordre d, si elle contient au moins une variable non-stationnaire d'ordre d.

1.2 Les modèles de prédiction

Les modèles de prédiction des séries temporelles se diffèrent des modèles de prédiction classiques, car ils permettent d'utiliser l'historique des séries pour estimer les valeurs futures, en utilisant des variables retardées. Autrement dit, pour faire des prévisions à l'instant t, on suppose qu'on connait juste les valeurs passées $(t-1, \dots)$. Cette approche est généralement adoptée pour modéliser les systèmes dynamiques, dont le temps et l'ordre des observations entre en jeux.

Il existe d'autres type de modèles de prédiction, comme la régression multiple par exemple, les arbres de décision et les RNAs classiques, qui utilisent les valeurs présentes des variables prédictives pour prédire la cible. C'est à dire, ils utilisent les valeurs des variables prédictives de l'instant t pour prédire la cible au même instant. Ces modèles sont généralement adaptés à des données qui ne sont pas forcément des séries temporelles, ou lorsque l'ordre des observations n'a pas d'impact sur les relations entre les variables cibles et les variables prédictives.

Les premiers modèles développés pour la prédiction des séries temporelles sont des modèles uni-variés basés sur le principe d'auto-régression. Dans tels modèles, les observations historiques sont utilisées pour faire des prévisions futures. Les modèles les plus populaires de ce type sont ; AR (*Auto-Regressive*), ARMA (*Auto-Regressive Moving Average*), et ARIMA (*Auto-Regressive Integrated Moving Average*) BOX 2013. Malgré leurs fondements mathématiques solides, les modèles uni-variés s'accompagnent de certains inconvénients ; ils ne tiennent pas compte des données potentiellement exploitables d'autres séries temporelles dans le même ensemble de données.

C'est pour cela que les modèles de prédiction multi-variés, qui intègrent multiple séries dans leurs analyses, ont été développés. Par exemple, la version étendue du modèle AR ; le modèle Auto-Regressive Vectoriel (VAR), et le modèle VAR cointégré ; le modèle de correction d'erreur vectorielle (VECM) JOHANSEN 1988 ; JOHANSEN 1991. Le principe qui sous-tend les modèles multi-variés est que la valeur d'une variable donnée dépend souvent des valeurs passées d'elle-même et

des autres variables. De tels modèles sont encore populaires aujourd'hui et sont utilisés indépendamment ou en combinaison avec d'autres techniques nouvelles, par exemple, la modélisation avec les réseaux de neuornes artificiels THIELBAR et D. A. DICKEY 2011.

Malgré des développements importants dans la modélisation multi-variée des prédictions, deux problèmes principaux échappent encore à des données hautement dimensionnelles : (i) comment sélectionner des variables prédictives pour une variable cible ; (ii) comment évaluer si la quantité de variables utilisées a affecté la précision des prédictions. Ces deux problèmes ont été examinés par des chercheurs ces dernières années, ce qui a donné des résultats originaux STOCK et M. W. WATSON 2012 ; JIANG, ATHANASOPOULOS, HYNDMAN et al. 2017. Nous discutons en détail quelques unes de ces approches dans 2.

1.2.1 Les modèles Auto-Régressifs

Nous discutons dans cette partie quelques modèles Auto-Régressifs, en les classifiant en deux catégories, les modèles uni-variés et les modèles multi-variés.

1.2.1.1 Les modèles uni-variés

Les modèles uni-variés permettent de prédire une seule série temporelle, en regardent les dernières valeurs de la série pour estimer les valeurs futures.

Le modèle AR : La plupart des modèles actuels sont inspirés du principe d'auto-regression, en particulier, le modèle AR (p) BOX 2013. Ce modèle considère une série temporelle uni-variée stationnaire comme une fonction linéaire de ces p valeurs précédentes. Formellement, l'équation du modèle AR (p) est la suivante :

$$Y(t) = \alpha_0 + \alpha_1 Y(t-1) + \cdots + \alpha_p Y(t-p) + U(t),$$

où p est l'ordre du modèle, $\alpha_0 \ldots \alpha_p$ sont les paramètres du modèle, et $U(t)$ est le terme d'erreur suivant un processus bruit blanc.

Le modèle MA : Le modèle Moyennes Mobiles (*Moving Average*) a la même structure que le modèle AR, mais en considérant les termes d'erreurs au lieu des valeurs précédentes de la série. Le modèle MA (q) peut être exprimé comme suit :

$$Y(t) = \theta_0 + \theta_1 U(t-1) + \cdots + \theta_p U(t-q) + U(t).$$

Le modèle ARIMA : Avant de voir le modèle ARIMA, il est important de comprendre le modèle ARMA. Le modèle ARMA (p, q) BOX 2013, combine les deux

processus AR(p) et MA(p) en considérant à la fois les termes d'erreurs et les valeurs précédentes de la série :

$$Y(t) = y_0 + \sum_{i=1}^{p} \alpha_i Y(t-i) + \sum_{i=1}^{q} \theta_i U(t-i) + U(t)$$

Où y_0, $\alpha_0 \ldots \alpha_p$, et $\theta_0 \ldots \theta_q$ sont les paramètres du modèle. Pour les séries temporelles non-stationnaires, le modèle ARIMA (p, d, q) est plus adéquat. Il consiste à appliquer le modèle ARMA(p, q) sur la série transformée par différenciation d'ordre d, c-à-d, en calculant d fois les différences entre les observations consécutives.

1.2.1.2 Exemple

Nous présentons ici un exemple montrant comment estimer le modèle AR par la méthode des moindres carrées. Pour se faire, nous allons l'appliquer sur une petite série temporelle mensuelle $\{Y, t \in [1, 12]\}$ sur une période d'une année :

$$Y = [0.2635, -0.1159, -0.8038, 1.0657, -0.4655, -0.6870, -0.7459, \\ 0.6477, -1.6496, 0.8874, 0.8349, -2.0139]$$

Nous avons vérifié que cette série est stationnaire juste pour l' ordre $p = 1$, en utilisant le test de *Dickey-Fuller* Augmenté de la bibliothèque NlinTS du logiciel R que nous avons développé (voir Chapitre 3.6 pour plus de détails). Alors on peut appliquer le modèle AR (1). A partir de $p = 2$, le processus AR (p) n'est plus stationnaire, et les erreurs ne sont pas des bruits blancs.

Construction du modèle : Considérons le modèle AR (1) :

$$Y(t) = \alpha_0 + \alpha_1 Y(t-1) + U(t) \tag{1.8}$$

Le cas idéal est lorsqu'il existe des paramètres $[\hat{\alpha_0}, \hat{\alpha}_1, \hat{\alpha}_2]$ qui modélisent exactement la variable Y. C'est-à-dire lorsque U est un vecteur nul. Dans ce cas, le modèle s'écrit comme suit :

$$Y(t) = \hat{\alpha_0} + \hat{\alpha_1} Y(t-1) + \quad \forall t \in [2, n]. \tag{1.9}$$

Dans cette situation, on dit que la variable y suit exactement un processus AR (1). Dans la réalité, les données ne suivant pas exactement les modèles théoriques. L'idée est d'aborder une approche approximative, en séparant la série en question en deux parties, une partie purement auto-régressif (dans notre cas AR (1)), et la deuxième partie représentant un terme d'erreur. Le choix des paramètres

$\alpha = [\alpha_0, \alpha_1]$ est effectué en résumant le plus possible les informations dans la série temporelle, tout en minimisant l'erreur. Ce problème peut se reformuler alors comme suit :

$$\begin{pmatrix} Y(2) \\ Y(3) \\ Y(4) \\ \dots \\ Y(n) \end{pmatrix} = \begin{pmatrix} 1 & Y(1) \\ 1 & Y(2) \\ \dots & \\ 1 & Y(n-1) \end{pmatrix} \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} + \begin{pmatrix} U(2) \\ U(3) \\ \dots \\ U(n) \end{pmatrix} \tag{1.10}$$

Ou d'une manière plus compacte :

$$Y = X\alpha + U \tag{1.11}$$

En se basant sur l'équation 1.11, on peut remarquer clairement qu'on s'est ramené à un problème de régression multiple. Dans se cas, la méthode des moindres carrée peut être appliquée. Elle permet de trouver les paramètres α qui décrivent le mieux possible les observations réelles en minimisant la somme quadratique des erreurs :

$$f(\alpha) = \sum_{t=3}^{n} \epsilon_t^2 = ||Y - X\alpha||^2 \tag{1.12}$$

$$= |Y - X\alpha|^\intercal |Y - X\alpha| \tag{1.13}$$

$$= Y^\intercal Y - 2\alpha X^\intercal y + \alpha^\intercal X^\intercal X \alpha. \tag{1.14}$$

L'estimation des moindres carrées du vecteur $\alpha = [\alpha_0, \alpha_1]$ est calculée par dérivation matricielle de la fonction f par rapport à α. Ce qui donne les équations suivantes :

$$X^\intercal y - \hat{\alpha} X^\intercal X = 0$$
$$\hat{\alpha} = (X^\intercal X)^{-1} X^\intercal y.$$

Prédiction : L'étape de la prédiction est la plus simple. Il suffit d'appliquer les paramètres trouvés sur les données d'entrée pour générer les prédictions. Dans l'exemple, on a : $\hat{\alpha} = [-0.3189 - 0.6023]$. Par conséquent, le modèle final s'écrit comme suit :

$$\hat{Y(t)} = -0.3189 + -0.6023 Y(t-1) \tag{1.15}$$

Pour simuler le modèle sur la série y, il faut commencer de $t = 2$, car on a besoin d'une valeur précédente. La figure 1.1 montre les courbes des valeurs réelles et des prédictions du modèle AR, pour des valeurs du paramètre p dans $\{1, \dots, 5\}$ (à partir de $p = 6$, le modèle ne peut pas être résolu). Notons que

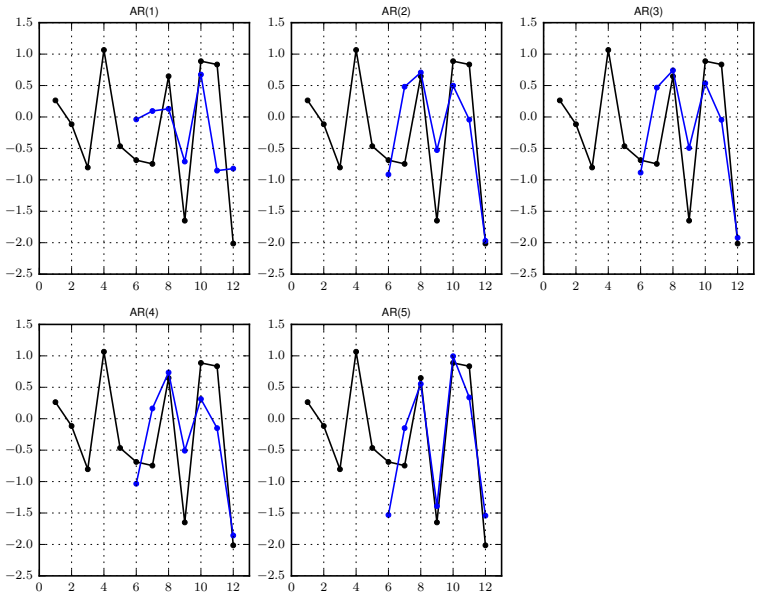


Figure 1.1 – Comparaison des prédictions du modèle AR avec différentes valeurs de l'ordre p.

même si les les modèles AR (p) est non-stationnaire à partir de $p = 2$, on peut estimer parfois les paramètres, mais les erreurs ne sont pas des bruits blancs. Dans cet exemple, nous avons une seule valeur de p pour la quelle le modèle AR est stationnaire. En général, si nous avons plusieurs modèles possibles, alors il faut se baser sur un critère pour choisir la valeur appropriée de p. Pour se faire, on peut utiliser le critère d'information d'Akaike (AIC), qui basé sur la théorie d'information AKAIKE 1974. On peut aussi se baser sur les prédictions à partir d'un ensemble d'expériences est choisir la valeur de p qui permet d'avoir l'erreur de prédiction minimum.

1.2.1.3 Les modèles multi-variés

La motivation derrière les modèles multi-variés est que pour prédire une série temporelle, d'autres données peuvent être utilisées pour améliorer la qualité des prédictions par rapport aux modèles uni-variés. Par conséquent, ils visent à exploiter d'autres sources d'informations, tout en suivant la même logique des modèles uni-variés. L'hypothèse majeur de ces modèles est que les valeurs futures d'une série cible dépendent non seulement de ses valeurs précédentes, mais aussi d'autres séries qui peuvent la causer.

Le modèle VAR (*Vector Auto-Regressive*) est une extension du modèle AR pour les séries temporelles multi-variées. Considérons une série temporelle $Y = \{Y_1, \ldots, Y_k\}$

de dimension k contenant n observations. Le modèle VAR (p) exprime chaque variable (retardée avec le paramètre p) de Y comme une fonction linéaire de ses p valeurs précédentes et des autres variables :

$$Y(t) = A_0 + \sum_{i=1}^{p} A_i Y(t-i) + U(t),$$

où U est le vecteur des erreurs qui suivent des processus bruit blanc, A_0 est un vecteur constant de taille k, et $A_1, \ldots, A_p$ sont les matrices paramètres du modèle de taille $(k \times k)$.

Si toutes les variables sont stationnaires, alors le modèle VAR est stationnaire, puisque la combinaison linéaire d'un ensemble de variable stationnaires est stationnaire. L'inverse n'est pas toujours vrai, c'est à dire qu'il se peut qu'une combinaison linéaire de variables non-stationnaire soit stationnaire. Ce phénomène s'appelle la cointegration JOHANSEN 1988, et il reflète l'existence des dépendances long terme entre les séries.

Dans le cas où le modèle VAR est non-stationnaire, où de manière formelle, si $det(\Pi = (I_k - A_1 - \cdots - A_p)) = 0$), alors le modèle VECM (*Vector Error Correction*) est plus approprié. Le modèle VECM, appelé aussi modèle VAR avec cointegration, introduit dans JOHANSEN 1991, tient compte de la non-stationnarité des variables. D'autre part, sa particularité principale est qu'il intègre les relations de cointegration (relations à long terme entre les variables). Considérons une série temporelle multivariée Y d'ordre de stationnarité 1, c-à-d., toutes les variables sont aux maximum non-stationnaire d'ordre 1 (autrement dit, il existe au moins une variable non-stationnaire d'ordre 1). Le modèle VECM peut être écrit comme suit :

$$\triangle Y(t) = \Pi Y(t-1) + \sum_{i=1}^{p-1} \Gamma_i \triangle Y(t-i) + U(t),$$

où Π est dite la matrice de cointegration (qui peut être générée par le modèle VAR). Γ_i sont les matrices coefficients du modèle. Si $rk(\Pi) = 0$, alors il n'y a pas de relation de cointegration entre les variables, même s'il ne suivent pas des processus stationnaires. Dans ce cas, le VECM est réduit à un modèle VAR sur les variables différenciés.

1.2.1.4 Méthode d'estimation

Il y a plusieurs méthodes de résolution des modèles auto-régressifs linéaires. Nous nous concentrons ici sur la technique des moindres carrées, mais nous discutons aussi brièvement les méthodes à base des méthodes de régularisation.

Soit $y = \{Y_1, \ldots, Y_n\}$ une série temporelle multi-variée contenant k variables et n observations. Pour simplifier les notations, on suppose que les p premiers

échantillons $(Y_{-p+1}, \ldots, Y_0)$ existent déjà. Cette supposition est facultative, puisqu'on peut toujours considérer la valeurs initiale $t = p+1$ si les premières valeurs des séries temporelles commencent à $t = 1$, et ensuite faire un changement de variables. Prenons le cas du modèle VAR :

$$Y(t) = A_0 + \sum_{i=1}^{p} A_i Y(t-i) + U(t), \tag{1.16}$$

Où U est le vecteur des erreurs. Le processus VAR(p) est stationnaire, ou stable, si et seulement si :

$$\det(I_k - A_1 - \cdots - A_p) \neq 0 \tag{1.17}$$

Considérons les notations suivantes :

$$\begin{aligned} X_t &= \Big[1, Y(t), \ldots, Y(t-p+1)\Big]^\intercal \\ X &= \Big[X_0, \ldots, X_{n-1}\Big] \\ B &= [A_0, A_1, \ldots, A_p] \end{aligned}$$

U est de dimension $k \times n$. B est la matrice des coefficients de dimension $(k \times (kp+1))$, c'est celle qu'on souhaite estimer. Et X représente les valeurs des variables, de dimension $((kp+1) \times n)$. Avec ces notations, on peut reformuler l'équation 1.16 comme suit :

$$Y = BX + U \tag{1.18}$$

Le but est de trouver les paramètres B qui permettent d'avoir le minimum erreur possible U.. La méthode des moindres carrées consiste à minimiser l'erreur quadratique moyenne, qu'on peut exprimer comme suit :

$$\begin{aligned} f(B) &= UU^\intercal && (1.19)\\ &= (Y - BX)(Y - BX)^T && (1.20)\\ &= YY^\intercal - Y(BX)^\intercal - BXY^\intercal + (BX)(BX)^\intercal && (1.21)\\ &= YY^\intercal - 2BXY^\intercal - BXX^\intercal B^\intercal. && (1.22) \end{aligned}$$

Par conséquent, l'estimation des paramètres B est déterminée en résoudant l'équation $\frac{\delta f(B)}{\delta B} = 0$: Le minimum de la fonction f peut être calculée par dérivation matricielle.

$$\frac{\delta f(B)}{\delta B} = 0, \tag{1.23}$$
$$-2XY' + 2XX^{\intercal}B^{\intercal} = 0, \tag{1.24}$$
$$B^{\intercal} = (XX^{\intercal})^{-1}(XY^{\intercal}), \tag{1.25}$$
$$B = (YX^{\intercal})(XX^{\intercal})^{-1}. \tag{1.26}$$

Les méthodes dites de régularisation (ou de rétrécissement (*Shrinkage*)) représentent une autre manière de déterminer les coefficients des modèles de régression. Ils cherchent à minimiser l'impact des variables non pertinentes en poussant leurs coefficients vers ou près de zéro.

Ces méthodes sont particulièrement adaptées aux problèmes où le nombre de variables prédictives est grand, ce qui pose un problème avec la résolution classique en utilisant par exemple la technique OLS, en raison de contraintes d'opérations matricielles.

Parmi les méthodes communes de régularisation , on cite la méthode RIDGE (dite aussi la régularisation de Tikhonov) HOERL et KENNARD 1970 et la méthode LASSO (*The Least Absolute Shrinkage and Selection Operator*) TIBSHIRANI 1994. La méthode Ridge minimise le terme suivant :

$$\sum_{t=1}^{n} (Y(t) - \beta_0 - \sum_{i=1}^{k} \beta_i X_i(t))^2 + \lambda \sum_{j=1}^{k} \beta_j^2, \tag{1.27}$$

où $\lambda \sum_{j=1}^{k} \beta_j^2$ est le terme qui représente la pénalité de rétrécissement. Ce mécanisme entraîne une réduction des coefficients estimés vers zéro. La méthode du Lasso est similaire à Ridge, mais utilise $\lambda \sum_{j=1}^{k} |\beta_j|$ comme terme de pénalité, afin d'augmenter la probabilité que les coefficients des variables non importantes soient égaux à zéro. Ces représentations sont en effet dédiées au problème de la régression. Cependant, ils peuvent facilement s'adapter au problème de prédiction des séries temporelles avec les modèles auto-régressifs.

1.2.2 Le modèle Auto-Régressif avec arbres de décision

Les arbres de décision représentent une technique très commune dans la modélisation des données. La prédiction des séries temporelles n'est pas une exception, et il y plusieurs façon de coupler les modèles de prédiction des séries temporelles avec les arbres de décision. La méthode classique consiste à transformer le problème sous une forme de régression multiple, après les arbres de régression peuvent être appliquer directement. Dans cette section, nous nous concentrons sur une forme particulière de l'utilisation des arbres de décision avec les modèles auto-régressifs. Cette approche à été proposée dans MEEK, CHICKERING et HECKERMAN 2002. Le but est de généraliser le modèle AR en permettant la

possibilité de tester des modèles avec différents paramètres dans une arbre de décision selon leurs adéquations aux séries temporelles. Cette approche peut être aussi utiliser pour évaluer des modèles avec différentes valeurs de l'ordre p.

A chaque arc est associé une condition du nœud parent. Les feuilles de l'arbre contiennent les modèles linéaires AR (p). A chaque feuille (modèle AR) est associée une fonction ϕ_j qui retourne 1 si toutes les condition depuis le nœud source sont vraie, et 0 sinon. Sur la Figure 1.2, on montre un exemple d'un modèle ART avec un ordre $p = 1$.

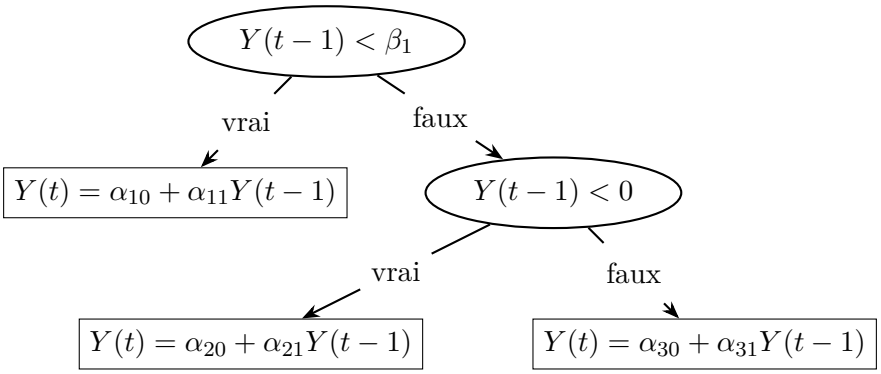


Figure 1.2 – Illustration du modèle ART avec $p = 1$.

Soit $Y(t)$ une séries temporelle uni-variée. De manière générale, le modèle ART (p) peut être exprimé comme suit :

$$Y(t) = \Pi_{j=1}^{l}(\alpha_{j0} + \sum_{i=1}^{p} \alpha_{ji}Y(t-i))_l^{\phi} \tag{1.28}$$

où l est le nombre de feuilles de l'arbre, et $(\alpha_{j0}, \ldots, \alpha_{jp})$ sont les paramètre des sous-modèles linéaires AR pour tout $j \quad \in \{1, \ldots, l\}$. Pour simplifier la représentation, on a exprimé juste le modèle uni-varié ART, mais sa version multi-variée peut être modélisée de la même manière. Avec cette stratégie, ce modèle n'est pas tout à fait linéaire, et il est plus adapté aux séries temporelles representant des dépendanes entre les variables MEEK, CHICKERING et HECKERMAN 2002. On peut voir clairement que le modèle AR est un cas particulier de ce modèle, et correspond à un arbre contenant un seul nœud.

1.2.3 Le filtre de Kalman

Le Filtre de Kalman, originalement proposé dans KALMAN 1960, permet de modéliser les systèmes dynamiques qui évoluent linéairement dans le temps en fonction de ses états précédents et d'autres facteurs aléatoires. Il à été utilisé dans plusieurs applications, notamment dans les domaine de contrôle des systèmes automatiques, la prédiction des séries temporelles, les systèmes mécaniques et physiques, et bien d'autres. L'un des avantages de Filtre de Kalman est, (i) qu'il

est un modèle rapidement adaptatif. Une fois qu'une nouvelle observation est faite, le modèle prend juste cette donnée pour mettre à jour ses paramètres. Le deuxième avantage (ii), est que le filtre de Kalman ne nécessite pas forcément des séries temporelles stationnaires, contrairement au modèles auto-régressifs.

L'idée centrale du Filtre de Kalman est principalement basée sur la notion d'état du système étudié. IL est fortement lié au chaines de Markov cachées, dans le sens où il distingue entre les états (qui sont généralement continus, c'est la différence avec les chaines de Markov cachées (CMC), ou l'ensemble des états est discret) et les observations du modèle. De manière similaire aux CMC, les observations dépendent de l'état précédent du modèle et des erreurs (qui représentent les phénomènes aléatoires non-mesurés par le modèle).

Nous présentons ici juste l'idée centrale de Filtre de Kalman et le modèle basique utilisé pour la prédiction des séries temporelles CARMONA 2014. Soit Y une série temporelle uni-variée. Les équations du modèle (dites aussi équations d'espace d'états) peuvent s'exprimer comme suit :

$$X(t) = A(t)X(t-1) + W(t),$$
$$Y(t) = F(t)X(t) + V(t),$$

où $X(t)$ représente l'état caché du modèle à l'instant t. Concrètement, c'est la fonction moyenne de la série temporelle — dans le cas d'une séries stationnaire, la moyenne de la série est stable, c'est pour cette raison que ce modèle traite aussi les séries non-stationnaires —. $A(t)$ est la matrice de transition des états. W et V sont deux bruits blancs (moyenne nulle, covariance constante, et auto-corrélation nulle) qui représentent les termes d'erreur du modèle.

On peut remarquer que l'équation $X(t) = A(t)X(t-1) + X(t)$ est similaire à l'équation d'un modèle auto-régressif AR (1). La différence est que la matrice $A(t)$, contrairement à celle du modèle AR (1), dépend du temps. Par conséquent, le Filtre de Kalman adapte ses paramètres de manière réactive après chaque itération.

1.2.4 Les Réseaux de Neurones Artificiels

Les réseaux de neurones artificiels (RNAs) sont de plus en plus utilisés dans la prédiction des séries temporelles. Avec ce type de modèles, la relation entre la variable cible et les variables prédictives est modélisée de manière non-linéaire. Ces modèles sont caractérisés principalement par la propriété d'apprentissage, vue qu'il sont conçue pour cela en imitant la structure des cerveaux biologiques. Dans le contexte des séries temporelles, ils sont capables de modéliser dynamiquement les interactions entre les variables par rapport aux modèles classiques. Le principe de l'utilisation des RNAs dans les prédictions de séries temporelles (ou des données séquentielles en général) commence par transformer la structure des données pour obtenir un problème d'apprentissage supervisé, où la sor-

tie représente les valeurs retardées des prédictives (y compris les valeurs précédentes des cibles). Puis, il y a l'étape de l'initialisation du réseau, il s'agit de déterminer la structure de réseau selon les données d'apprentissage et d'autres paramètres.

Considérons une série temporelle k-dimensionnelle $Y = \{Y_1, ..., Y_k\}$. Et supposons que nous somme intéressés à prédire toutes les variable de Y. Considérons aussi un RNA définis par la fonction fnn (c'est fonction globale du réseau, qui permet de calculer les sorties finales du réseau). Alors c'est un réseau qui prend $(k \times p)$ entrés (p valeurs de chaque variable de Y), et un vecteur de sortie de dimension k (k prédictions relatives aux k variables de Y). Un modèle de prédiction relatif à ce réseau prend en entrée les valeurs passées de chaque $(Y_i(t-1), \ldots, Y_i(t-p)$ pour $i \in [1, k])$:

$$Y(t) = f_{nn}(Y_1(t-1:t-p), \ldots, Y_k(t-1:t-p)) + U(t),$$

où $Y_i(t-1:t-p) = \{Y_i(t-1), \ldots, Y_i(t-p)\}$ pour $i \in [1, k]$.

1.2.4.1 L'algorithme de rétro-propagation

L'algorithme de rétro-propagation est parmi les principaux concepts de base de l'apprentissage des RNAs. C'est l'algorithme qui permet de mettre à jour, ou de réajuster les coefficients du réseau, en lançant les erreurs dans le sens inverse du réseau et modifier les paramètres de chaque neurone, de tel façon à minimiser une fonction d'erreur. Le concept de la rétro-propagation est le fruit des anciens travaux mathématiques de Cauchy sur la minimisation des erreurs des fonctions non linéaires. Mais le contexte était différent par rapport au RNAs. Les premièrs travaux de la rétro-propagation sur les RNAs aparaissent à partir des années 60 BRYSON et HO 1969, PARKER 1986. Puis, ce principe est devenu très populaire après une démonstration faite dans KŮRKOVÁ 1992 basée sur le théorème de Kolmogorov, qui affirme que pour toute fonction continue $f : [0, 1]^n \to \mathbb{R}$, il existe un RNA de type multi-couches, composé de 3 couches, dont la première couche contient n neurones (le nombre d'entrées) avec une couche cachée de $2n + 1$ neurones et une couche de sortie de m neurones (le nombre de sorties), qui peut implémenter la fonction f exactement. Après cette découverte pertinente, les RNAs multi-couches basées sur la rétro-propagation sont devenus des modèles d'approximation universels.

Nous ne montrons pas ici la démonstration, que l'on pourra trouver avec plus de détails dans HECHT-NIELSEN 1989 et HECHT-NIELSEN 1989. Aussi, nous ne détaillons l'algorithme de rétro-propagation, qui fait pas l'objet de ce travail, mais nous montrons brièvement les principaux étapes de cet algorithme.

Définir la structure du réseau : Le nombre de neurones de la première couche et de la dernière couche est déterminé en fonction des données d'apprentissage. Par contre, il y a plusieurs choix pour le nombre de couches cachées et de neurones de chaque couche. C'est l'un des inconvénients des RNAs, car généralement on sait par la meilleures structures à utiliser pour avoir les meilleures prédictions.

Initialisation : Cette étape consiste à initialiser les paramètres du réseau, c-à-d., les coefficients de chaque neurone à partir de la deuxième couche. Généralement, ces paramètres sont générés de manière aléatoire.

Propagation des donnée d'entrée : Cette étape consiste à calculer les sorties de chaque neurone et la sortie finale du réseau à partir des données d'entrée. C'est une sorte de simulation du réseau.

Rétro-propagation des erreurs : C'est l'étape la plus importante et qui constitue le cœur de l'apprentissage supervisé du réseau. L'apprentissage est un processus itératif, contrôlé soit par le nombre d'itérations, soit par un seuil minimum d'erreur. Á chaque itération, le réseau modifie ses paramètres à partir d'un tuple des données d'apprentissage, en minimisant une fonction objective, par exemple l'erreur quadratique moyen. Une méthode pour minimiser de telles fonctions consiste à appliquer l'algorithme du gradient. C'est un algorithme de minimisation des fonctions à plusieurs variables. Il consiste à minimiser la fonction d'erreur de manière itérative, et à chaque itération l'erreur se diminue jusqu'à ce qu'il trouve un minimum local. Mais cet algorithm ne garantit pas de trouver le minimum globale.

Soulignons aussi que lorsque l'historique des séries temporelles est long, l'algorithme du gradient stochastique peut être utilisé pour économiser du temps le temps d'apprentissage du réseau. Mais dans se cas, l'entrainement des RNAs sur les séries temporelles est légèrement différent des données non temporelles. Elle doit être fait de manière spécifique, lors de la sélection des sous-ensembles stochastiques des entrées, afin de respecter la séquence chronologique des données.

Au niveau des algorithmes de rétro-propagation, d'autres méta-heuristiques d'optimisation peuvent être utilisés pour l'adaptation des paramètres du réseau, en vue d'éviter les minimums locaux. Par exemple, les algorithmes génétiques ont été utilisés dans GUPTA et SEXTON 1999, comme alternative de l'algorithme du gradient pour l'apprentissage des RNAs multi-couches. Dans KHAN et SAHAI p.d., aussi pour améliorer l'apprentissage des es RNAs multi-couches, une comparaison à été expérimentée entre l'algorithme de gradient descent, l'optimisation par essaims particulaires, où l'auteur propose un nouveau algorithme, dit BAT YANG 2010, qui permet d'avoir de meilleures performances.

1.2.4.2 Le modèle VAR avec réseau de neurone multi-couches

Le réseau multi-couche (MLP) est celui que nous avons discuté dans la section précédente, est beaucoup utilisé dans la prédiction des séries temporelles. Mais il nécessite une étape de préparation, en l'adaptant pour qu'il puisse prendre en compte des variables retardées (*cf*. Figure 1.3). Par exemple, dans Dhoriva U. WUTSQA 2008, le modèle Réseau de Neurones Vecteur Auto-Regressif (VARNN) à été étudié, comme extension du modèle classique VAR.

Dans AYDIN et CAVDAR 2015, une comparaison entre le modèle VAR et un réseau de neurones multi-couches a été appliqué pour la prédiction des variables macroéconomiques. Et il est démontré que le réseau de neurones a des résultats supérieurs à celles du modèle VAR. Dans Dhoriva Urwatul WUTSQA, SUBANAR et SUJUTI 2006, le modèle VARNN également montre de bonnes performances lors de la prédiction de fonctions non linéaires en le comparant au modèle VARMA (Vector Auto-Regressive Moving Average).

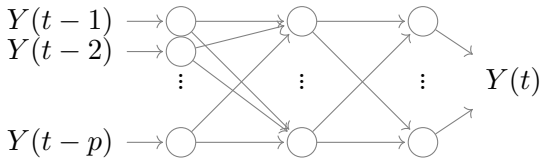


Figure 1.3 – Illustration du modèle ARNN (p).

1.2.4.3 Les réseaux de neurones récurrents

Le principal inconvénient du réseau de neurones multicouche est que leurs neurones ne sont pas capables de se souvenir des informations précédentes passées dans le réseau. Parce que chaque neurone fournit une sortie basée directement sur la fonction d'activation et les valeurs d'entrée. Dans les séries temporelles et les données séquences en général, le maintien d'informations à l'intérieur du réseau est une propriété très importante et susceptible d'améliorer les performances du réseau. Les réseaux neuronaux récurrents (RNN) ont été conçus pour gérer ce problème. Leurs structures permettent aux couches cachées d'être auto-connectées, de manière à avoir une sortie qui dépend de l'entrée courante et de son état précédent. Mathématiquement, la fonction des neurones d'un réseau de neurones récurrents RNRs peut être exprimée comme suit :

$$h(t) = f(Wx(t) + Uh(t-1) + b)$$

Le modèle RNN présente aussi quelques inconvénients. La mémoire maintenue dans le réseau est courte, car l'état actuel du réseau dépend seulement de l'état précédent. Le réseau à mémoire court et long terme LSTM (*Long Short Term*

Memory) HOCHREITER et SCHMIDHUBER 1997 a été conçu pour résoudre ce problème. Les auteurs proposent une structure spécifique, en ajoutant des options supplémentaires transformant les neurones traditionnels en blocs, capables de modéliser un mécanisme d'oubli et de mémorisation. Ainsi, il apprend comment utiliser les informations à long terme transmises dans le réseau. La formulation mathématique d'un bloc LSTM peut s'écrire comme suit :

$$
\begin{aligned}
f(t) &= \sigma(W_f x(t) + U_f h(t-1) + b_f) \\
i(t) &= \sigma(W_i x(t) + U_i h(t-1) + b_i) \\
c(t) &= f(t) \odot c(t-1) + i(t) \odot tanh(W_c x(t) + U_c h(t-1) + c_0) \\
o(t) &= \sigma(W_o x(t) + U_o (t-1) + b_o) \\
h(t) &= o(t) \odot tang(c(t)).
\end{aligned}
$$

Où :
— x_t est le vecteur d'entrée
— $\odot$ représente le produit matriciel de Hadamard
— (W, U, b) sont les paramètres du modèle,
— σ et $tanh$ sont respectivement la fonction sigmoïde et la fonction tangente, qui représentent les fonctions d'activation,
— f_t : porte d'oubli (*forget gate layer*), responsable de mettre à jour les poids associés à la capacité de l'unité LSTM à se souvenir des informations précédentes,
— i_t : porte d'entrée (input gate layer), contrôle la mise à jour à partir nouvelles informations,
— c_t : est l'état de l'unité LSTM à l'instant t,
— o_t : porte de sortie (output gate layer),
— h_t est la sortie de l'unité.

1.2.5 Discussion sur les modèles de prédiction étudiés

Nous avons présenté quelques modèles de prédiction parmi les plus utilisés dans le domaine des séries temporelles. Ces modèles peuvent être classés en plusieurs classes, selon le type du modèle (linéaire ou non-linéaire), la structure (uni-varié ou multi-varié) et la manière d'adapter ses paramètres (adaptatif ou statique). La première classe des modèles présentés concerne les modèles autorégressifs. Ces modèles ont beaucoup d'avantages. Leurs fonctionnement est basé sur des principes solides développés depuis des décennies. Il y a deux types de modèles, les modèles qui permettent de traiter les séries temporelles uni-variés (ex., AR, ARIMA, ...), et les modèles qui prennent en compte plusieurs séries en même temps (ex., modèle VAR, VECM).

Nous avons vu aussi brièvement le Filtre de Kalman. L'avantage de ce modèle est qu'il est adaptatif, est caractérisé par sa rapidité à mettre à jour ses

paramètres, et à générer les prédictions, ainsi que sa capacité à traiter les séries non-stationnaires de manière directe, qui est un inconvénient de beaucoup de modèles. Par exemple, les modèles auto-régressifs, comme les modèles ARIMA et VECM considèrent aussi la non stationnarité des variables, mais en deux étapes, ils transforment d'abord les variables via la différentiation, puis appliquent le modèle sur les nouvelles données stationnaires. Le filtre de Kalman à quelques limitations. Premièrement c'est un modèle linéaire. Deuxièmement, il est basé sur le fait que l'état actuel ne dépend que son état précédent (principe similaires à celui des chaines de Markov Cachées). Or, on sait que les séries temporelles peuvent dépendre des valeurs plus anciennes d'elle mêmes, qui représentent des relations long terme. Notamment, d'autres modèles plus récents, non-linéaires basés sur les RNAs, prennent en compte des valeurs plus anciennes des séries temporelles pour réaliser des prédictions, questionnent les modèles linaires et le Filtre de Kalman.

Les RNAs ont une caractéristique très importante. Il est possible de transformer tous les modèles Auto-Régressifs vers des modèles non-linéaires via les RNAs, et ceci donne de nouveaux modèles qui héritent à la fois des avantages des modèles auto-régressifs, et des points forts des RNAs, c.à.d., le principe d'apprentissage et la non-linéarité. Leurs inconvénients sont la difficulté à trouver la meilleur structure du réseau, et le temps d'apprentissage, qui est généralement trop lent par rapport aux modèles linéaires sur les grandes séries temporelles.

1.3 La Notion de Pertinence des Variables

Dans cette section nous abordons la question de pertinence des variables. La pertinence est un concept très important, il est commun dans l'apprentissage automatique, la théorie des ensemble approximatifs, et d'autre domaines. Dans notre problématique, elle va nous permettre d'évaluer l'importance des variables prédictives par rapport à une variable cible dans un modèle de prédiction multivarié. Dans ce qui suit, nous formalisons la pertinence des variables dans le cadre de la prédiction des séries temporelles, et nous discutons quelques définitions de la littérature.

1.3.1 Définitions

Soit $X = \{X_1, \ldots, X_n\}$ un ensemble de variables prédictives, et une variable cible Y. Et supposons que les données d'apprentissage sont sous la forme de tuples $\{X, Y\}$.

La question qu'on veut répondre via la notion de pertinence est la suivante :

— Quel est le sous ensemble de X qui contient les variables les plus pertinentes par rapport à Y en terme de prédiction ?

Dans la littérature, il y a plusieurs définitions de la pertinence des variables (*features relevance*). Dans ALMUALLIM et DIETTERICH 1991, il est définit qu'une variable X_i est pertinente à un concept C si et seulement si X_i apparait dans toute formule booléenne qui représente C. Une autre alternatiave plus pratique été proposée dans JOHN, KOHAVI et PFLEGER 1994. Elle est basée sur l'effet de filtrage. En ajoutant ou bien en enlevant une variable de l'ensemble des variables prédictives, est ce que cela engendrera un changement dans le comportement de la variable cible. Formellement, cette définition est la suivante :

Définition 8 (Pertience des variables JOHN, KOHAVI et PFLEGER 1994)**.** *La variable X_i est pertinente si et seulement si il existe x_i et y tel que : $P(X_i = x_i) > 0$) et :*

$$P(Y = y|X_i = x_i) \neq P(Y = y) \tag{1.29}$$

Cette définition veut dire que X_i est pertinente par rapport à Y, si le fait de savoir et de prendre en compte l'information de X_i dans le modèle change l'estimation de Y par rapport à la situation où X_i n'est pas prise en compte. Dans le même travail (JOHN, KOHAVI et PFLEGER 1994), d'autres définitions similaires ont été présentées, où plus de deux variables sont considérées, mais elles ont le même principe.

Soulignons que cette définition est assez proche de la définition de causalité prédictive, que nous discutons dans la prochaine sous-section, dans la mesure où les deux portent sur l'idée de tester la variable prédictive avec la cible, puis la retirer et évaluer la différence entre ces deux situations pour mesurer son effet sur la cible.

1.3.2 La causalité prédictive

L'étude des dépendances entre les variables est une étape importante dans l'analyse des séries temporelles multi-variées. Sans surprise, elle peut être exploitée pour optimiser les modèles de prédiction, et par la suite améliorer la qualité des prédictions.

La causalité est une notion générale, mais il est beaucoup utilisée dans la modélisation des données complexes, notamment, pour représentes les dépendances entre les variables. De même que la pertinence, la causalité à également plusieurs définitions, et il n'y a pas une définition universelle.

Les mesures standards, comme la corrélation et l'information mutuelle, sont largement utilisées dans l'analyse des liens entre les données. Mais à notre avis, elles ne sont pas forcément adéquates pour le problème de prédiction des séries temporelles avec l'utilisation les variables retardées. Parce que ces mesures sont symétriques, elles ne fournissent donc pas suffisamment d'informations sur le sens de dépendances entre les variables, c'est-à-dire quelle variable influence l'autre.

De ce point de vue, la causalité est plus précise que la corrélation dans le sens ou elle fournit plus d'informations sur les interactions entre les variables.

Un problème de la causalité se manifeste dans la difficulté à la définir mathématiquement. Contrairement à la corrélation, qui est basée globalement sur des concepts bien étudiés en mathématique et en statistiques, comme la covariance et la similitude. La causalité reste une abstraction difficile à définir de manière objective. Parmi les définitions les plus populaires dans la littérature, on peut citer celle de Granger proposée en GRANGER 1980. En supposant que c'est un point de vue sur la définition de la causalité, l'auteur propose une définition basée sur la prédiction, et a pris le prix Nobel d'économie sur ces travaux. Une autre modélisation basée sur l'entropie est introduite en SCHREIBER 2000 qui repose sur le théorie d'information, et proposée comme extension de l'information mutuelle.

Dans le reste de cette paragraphe, nous allons présenter deux mesures de causalité beaucoup utilisées dans le domaine des séries temporelles : la causalité de Granger GRANGER 1980, et l'Entropie de Transfert SCHREIBER 2000.

1.3.2.1 La causalité de Granger

La définition de causalité de Granger est basée sur le fait qu'une variable X cause une variable Y s'il contient des informations utiles pour prédire Y. En d'autres termes, si en supprimant X des informations disponibles utilisées pour prédire X, les résultats de prédiction pour Y seront affectés.

Définition 9 (Causalité de Granger GRANGER 1980)**.** *Considérons deux séries temporelles uni-variées X, Y contenant t observations, et $X(t+1)$ est inconnu à l'instant t, et peut donc être exprimé en utilisant la probabilité dans un ensemble B. Considérons I l'ensemble des informations disponibles à l'instant t y compris $Y(t)$. Y cause X à l'instant $t+1$ si :*

$$P(X(t+1) \in B|I) \neq P(X(t+1) \in B|I \setminus y_t) \tag{1.30}$$

Cette formulation présente le concept générale de la causalité de Granger. En pratique, pour un test statistique associé à la causalité de Granger est utiliser pour vérifier la causalité entre deux variables. Ce test utilise le modèle VAR, et il suit la même logique de la définition et compare deux modèles, (i) le premier ne prend en compte que les valeurs précédentes de Y et (ii) la seconde utilise à la fois X et Y en vue de prédire Y. S'il y a une différence significative entre les résultats des deux modèles, alors le test considère que x cause y :

$$\begin{aligned}
\text{Modèle}_1 : \quad Y(t) &= \alpha_0 + \alpha t + \sum_{i=1}^{p} \alpha_i Y(t-i) + U(t), \\
\text{Modèle}_2 : \quad Y(t) &= \alpha_0 + \alpha t + \sum_{i=1}^{p} \alpha_i Y(t-i) + \sum_{i=1}^{p} \beta_i Y(t-i) + U(t),
\end{aligned}$$

où $\{\alpha_0, \alpha, \alpha_1, \ldots, \alpha_p, \beta, \beta_0, \beta_1, \ldots, \beta_p\}$ sont des paramètres, et U est un vecteur d'erreur.

L'étape suivante du test consiste simplement trouver le modèle (parmi Modèle$_1$ et Modèle$_2$) qui explique mieux Y, c-à-d, qui minimise plus le vecteur U. Pour cela, le test de causalité de Granger compare la somme résiduelle des carrés (RSS) de ces deux modèles en utilisant le test de *Fisher*. La statistique du test est exprimée comme suit :

$$F = \frac{(\text{RSS}_1 - \text{RSS}_2)/p)}{(\text{RSS}_2/n - 2p - 1)},$$

où RSS$_1$ et RSS$_2$ sont les somme résiduelles des carrés des Modèle$_1$ et Modèle$_2$ respectivement, n est la taille du vecteur prédit par le modèle VAR. Deux hypothèses sont testées :

— $H_0 : \forall i \in \{1, \ldots, p\}, \beta_i = 0$
— $H_1 : \exists i \in \{1, \ldots, p\}, \beta_i \neq 0$.

Selon l'hypothèse nulle H_0, x ne cause pas y. Dans ce cas, F suit la distribution de Fisher avec $(p, n - 2p - 1)$ degrés de liberté, puis le test est effectué à un niveau α afin d'examiner l'hypothèse nulle de non-causalité. La valeur de α est généralement fixée à 5% ou 1%.

1.3.2.2 L'Entropie de Transfert

L 'Entropie de Transfert est une mesure de causalité basée sur la théorie de l'information. Avant de discuter son principe, il est utile de rappeler la notion d'information mutuelle entre deux processus (ou deux séries temporelles univariées) X et Y. Il mesure les dépendances mutuelles (symétriques) entre les deux variables. L'Information Mutuelle entre deux processus peut être exprimée en utilisant l'entropie de *Kullback* :

$$M_{XY} = \sum_{x \in X, y \in Y} P(x, y) log(\frac{P(x|y)}{P(x)P(y)}).$$

Le principal inconvénient de cette mesure est qu'elle est symétrique et ne modélise pas le transfert d'informations d'un processus à un autre. L'Entropie de Transfert a été proposée comme une extension de l'information mutuelle SCHREIBER 2000, avec l'idée d'ajouter un paramètre de décalage temporel à l'Information Mutuelle, ce qui permet de modéliser le transfert d'information entre deux processus. La formulation mathématique de l'Entropie de Transfert de J à I peut être exprimée comme suit :

Définition 10 (Entropie de Transfert SCHREIBER 2000)**.** *L'Entropie de Transfert entre deux séries temporelles, ou deux processus X et Y, mesure le flux d'information*

de X vers Y :

$$T_{X\to Y} = \sum P(Y(t+1), Y^{(l)}(t), X^{(m)}(t)) log(\frac{P(Y(t+1) \mid Y^{l}(t), X^{m}(t))}{P(Y(t+1) \mid Y^{(l)}(t))}), \quad (1.31)$$

où $Z^{l}(t) = (Z(t), \ldots, Z(t-l+1))$ pour $Z = X, Y$, l, m sont les paramètres de décalage, et P représente la probabilité.

Même si la causalité de Granger et l'Entropie de Transfert semblent être des concepts différents, une découverte intéressante a été présentée dans BARNETT, BARRETT et SETH 2009, montrant qu'elles sont équivalentes pour les variables qui suivent une loi normale. De plus, l'entropie de transfert est considérée comme une extension non-linéaire de la causalité de Granger, et une alternative lorsque le test de Granger ne peut pas être effectué au niveau de la détermination des paramètres des modèles VAR du test.

1.4 Réduction du Nombre de Variables

Dans cette section, nous présentons quelques méthodes de réduction de dimension de données multi-dimensionnelles. Plusieurs types de méthodes peuvent être appliqués dans ce contexte. Les méthodes classiques sont les méthodes dites de réduction de dimension, comme l'Analyse en Composantes Principales (ACP), l'analyse factorielle, et bien d'autres. Ces méthodes réduisent la dimension par construction de nouvelles données comme combinaison des données originales. Une autre manière d'aborder ce problème consiste à appliquer les méthodes de sélection de variables. Dans ce cas, on filtre parmi les données existantes quelques variables (les plus pertinentes) qui vont les représenter.

Une autre méthodes non classique consisté à appliquer les techniques de l'analyse des liens (*Link Analysis*). Ces méthodes sont typiquement conçues pour détecter les pages les plus pertinentes à partir d'un graphe du web, qui représente les pages et les liens entre eux. Nous pensons que c'est méthodes peuvent être appliquées pour sélectionner les variables les plus importantes, puisque souvent, on représentent les séries temporelles multidimensionnelles sous forme de graphe de dépendances. Et c'est l'une des propositions de ce travail. Nous détaillons cette idée dans le Chapitre 3.

Dans la suite de cette section, nous allons présenter le principe des approches de réduction existantes dans la littérature, et discutons leurs avantages et inconvénients dans le contexte des séries temporelles.

1.4.1 Sélection des variables

Dans cette première partie, nous commençons par une description générale des techniques de sélection des variables — parfois aussi appelées techniques

d'extraction des variables —. Nous discutons également quelques méthodes classiques, parmi les plus utilisées dans les domaines de l'apprentissage et l'analyse des données multi-dimensionnelles. Ensuite, nous nous concentrons sur leurs utilisation dans la prédiction. Finalement, nous présentons quelques approches proposées dans la littérature, qui les abordent pour la prédiction de séries temporelles.

Considérons un ensemble de variables X de taille n. La sélection de variables est le processus qui consiste à extraire un sous-ensemble de X contenant les variables pertinentes, tout en éliminant les informations redondantes.

En général, dans les modèles d'apprentissage multi-variés, l'objectif est d'expliquer des variables cibles (ou classes) en fonction d'un ensemble de facteurs (variables prédictives). Avec l'augmentation des données disponibles, utiliser tous les données existantes d'apprentissage peut d'un coté complexifier le problème, c'est à dire, augmenter le temps de calcul, et dans certains cas, il peut rendrel'estimation des paramètres de quelques modèles statistiques impossible STOCK et M. W. WATSON 2006. D'un autre coté, utiliser toutes les variables peut engendrer le problème de sur-apprentissage, qui peut dégrader la qualité des prédictions JIANG, ATHANASOPOULOS, HYNDMAN et al. 2017.

L'idée est de construire des modèles simples en utilisant un sous-ensemble de variables prédictives, dans le but d'avoir à des modèles qu'on peut estimer à un temps raisonnable, et aboutissant à de bonnes performances de la prédiction.

Les méthodes de sélection des variables peuvent être classées en deux principales catégories, les approches basées sur le filtrage des variables, et les approches dites d'enveloppe, qui sélectionnent les variables au fur est à mesure par l'algorithme d'apprentissage lui même.

1.4.1.1 Approches de type *Wrapper*

Afin de sélectionner un sous-ensemble de variables prédictives pour un modèle d'apprentissage (dans notre un modèle de prédiction), les méthodes de type *Wrapper* exploitent le modèle lui même pour sélectionner les variables. Ces méthodes fonctionnent itérativement en raffinant à chaque itération les variables sélectionnées par rapport au performance du modèle. Sur la Figure 1.4, on présente une illustration qui montre le principe général de ces approches.

Ces méthodes présentent quelques inconvénients. Premièrement, le temps de calcul peut être très important, surtout si les données d'entrée sont massives, parce qu'on répète plusieurs fois les étapes de sélection et de prédiction. Deuxièmement, le sous-ensemble de variables choisit est relatif au modèle d'apprentissage utilisé. Ceci dit que si on change le modèle utilisé, il faut refaire tout le processus.

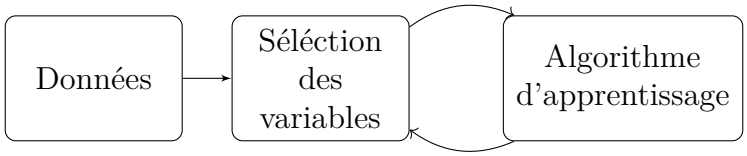


Figure 1.4 – Schéma des approches de sélection de type *Wrapper*.

1.4.1.2 Approches de Filtrage

Les méthodes basées sur le filtrage fonctionnent indépendamment du modèle d'apprentissage utilisé. Elles consistent à filtrer l'ensemble initial des variables en se basant sur leurs caractéristiques, les dépendances entre eux, et leurs dépendances avec les variables cibles. Dans la suite, nous détaillons quelques approches de cette catégorie.

Figure 1.5 – Schéma des approches de sélection des variables basées sur le filtrage.

Les approches de classement : Elle consistent à classer les variables par rapport à une mesure d'importance, comme les mesures de similarité, la corrélation, les distances, ... Les variables finaux sont ensuite sélectionnées en fixant un seuil limite sur la mesure utilisée, ou bien selon le nombre de variables souhaité. L'avantage de ces méthodes et qu'elles sont simples à implémenter et rapides. Leurs principal inconvénient se manifeste dans le fait qu'elles ne prennent pas en compte les dépendances entre les variables.

Les méthodes basées sur la corrélation : Les méthodes basées sur la corrélation utilisent la corrélation comme mesure de similarité entre les variables prédictives et la variable cible. La méthode naïve consiste à classer les variables prédictives selon leurs corrélation avec la cible. Pour cela, il suffit de calculer la corrélation de chaque variable avec la variable cible via la corrélation de *Pearson* par exemple.

Une autre approche cherche à extraire le sous-ensemble qui contient les variables les plus corrélés avec la variable cible et les moins corrélées entre elles. Dans cette perspective, une heuristique à été proposée dans HALL 1998 basée sur l'évaluation du sous-ensemble de variables en utilisant une formule qui prend en compte les corrélations des variables par rapport à la cible, et l'inter-corrélation entre eux.

Considérons un ensemble S contenant k variables prédictives. La qualité de l'ensemble S est exprimée comme suit :

$$Q_s = \frac{k\bar{r}_c}{\sqrt{k + (k-1)\bar{r}_v}}. \tag{1.32}$$

Où kr_c est la corrélation moyenne des pairs variables-cible, et kr_v est la corrélation moyenne des paires variable-variable de l'ensemble S.

Méthode basée sur la causalité : Une méthode qui utilise la causalité consiste à classer les variables selon leurs causalités vers la variable cible. Cette approche a déjà été étudiée dans SUN, Jiuyong LI, J. LIU et al. 2014 et ZHANG, HU, XIE et al. 2014 en utilisant le test de causalité de Granger. Par exemple l'algorithme proposé dans SUN, Jiuyong LI, J. LIU et al. 2014 sélectionne les variables qui causent la variable cible avec la contrainte que la cible ne les causent pas. Pour se faire, un seuil de causalité minimal min_caus doit être fixé. Soit deux variables x et y, et considérons la notation $caus(x \to y)$ comme la causalité de la variable x vers la variable y. Cet méthode est développée par l'Algorithme 1.

Algorithm 1 Classement par causalité vers la cible

Input : X : l'ensemble de n variables ;
Y : la variable cible ;
min_caus : seuil minimum de causalité ;

$P = \emptyset$ // L'ensemble de variables sélectionnées
$k = 0$ /* le nombre de variables sélectionnées */
for $\{X_i \in X\}$ **do**
  if $(causality(X_i \to Y) \geq min\_caus)$ **then**
    if $(causality(Y \to X_i) \leq min\_caus)$ **then**
      $P = P \cup \{X_i\}$
      $k = k + 1$
    end if
  end if
end for
return P, k ;

1.4.2 Les techniques de réduction de dimension

Les méthodes de réduction de dimension génèrent, à partir d'un ensemble de variables de départ, de nouvelles variables de dimension réduite, dites facteurs, comme combinaison des variables originales. Ces facteurs sont construis de telle façon qu'ils soient indépendants, et qu'ils contiennent autant que possible les caractéristiques et les informations contenues dans les variables originales. Afin

de détailler le fonctionnement de ces approches, nous détaillons deux méthodes très connues, et qui représentent la base des méthodes de réduction de dimension. Il s'agit de l'analyse en composante principale (ACP) et l'analyse factorielle (AF).

1.4.2.1 L'Analyse en Composantes Principales

Nous nous concentrons ici sur la formulation standard de l'ACP JOLLIFFE 1986. Elle est très utilisée dans le domaine de l'apprentissage non-supervisé, d'analyse des données, de traitement d'image, et bien d'autres. Nous allons se limiter sur son application dans la réduction de dimension des séries temporelles (voir Section 2.1).

Le principe fondamental de cette méthode est de trouver un nombre minimum de facteurs sur lesquelles les données peuvent être mieux expliquées. Considérons un vecteur de n variables $Y = \{Y_1, \ldots, Y_n\}$ normalisées, et $\Sigma = Y^\intercal Y$ est proportionnelle à la matrice de covariance de Y. L'ACP permet de générer $k \ll n$ facteurs $[P_1, \ldots, P_k]$, dites composantes principales, qui sont des combinaisons linéaires des variables de Y, de telle façon à ce qu'elles soient non-corrélées entre elles, et ayant les plus grandes variances possibles.

Les composantes principales sont définies comme combinaisons linéaires des variables originales. On peut les exprimer comme suit :

$$P_j = A_j^\intercal Y \tag{1.33}$$

$$= \sum_{i=1}^{n} a_{ji} Y_i \quad \forall j \in [1, k], \tag{1.34}$$

Les vecteurs coefficients $[A_1, \ldots, A_k]$, où $A_j^\intercal = [a_{j1}, \ldots, a_{jn}]$, pour tous $j \in [1, k]$, doivent être estimés de manière à obtenir des composantes principales $[P_1, \ldots, P_k]$ non-corrélées entre elles, et ayant des variances maximales. Une approche pour trouver ces coefficients consiste à utiliser la décomposition de la matrice Σ en éléments propres. Σ est une matrice carrée, symétrique, et positive, alors, elle est diagonalisable. Par conséquent elle peut être écrite sous la forme :

$$\Sigma = UDU^\intercal \tag{1.35}$$

où D est une matrice diagonale, U est la matrice des vecteur propres tel que $U^\intercal U = I$.

En se basant sur cette propriété, les composantes principales $[A_1^\intercal Y, \ldots, A_k^\intercal Y]$) sont déterminées de telle façon à ce que $[A_1, \ldots, A_k]$ sont tout simplement les vecteurs propres de Σ dans l'ordre croissant par rapport aux valeurs propres, c-à-d, A_k est associé à la $k^{\text{ème}}$ valeur propre de Σ. C'est-à-dire, il suffit de trouver la matrice U ($U = [A_1, \ldots, A_k]$).

Démonstration

Ce résultat peut être démontré en utilisant la technique de Lagrange. Pour ce faire, on va se limiter à déterminer les deux premiers vecteurs A_1 et A_2, car la méthode pour déterminer les autres se fait de la même façon. Le problème revient à trouver un vecteur A_1 qui maximisent la variance $Var[A_1 \intercal Y] = A_1^\intercal \Sigma A_1$, et tel que $A_1^\intercal A_1 = 1$. Alors c'est un problème où il faut maximiser un terme avec une contrainte supplémentaire ($A_1^\intercal A_1 = 1$). Ce type de problème peut être résolut avec les techniques de recherche opérationnelle. La technique de Lagrange consiste à rendre ce problème à un problème de maximisation (dans ce cas) d'une seule fonction objective, c-à-d, il intègre la contrainte dans le terme à maximiser :

$$\max A_1^\intercal \Sigma A_1 - \lambda(A_1^\intercal A_1 - 1) \tag{1.36}$$

où λ est le multiplicateur de Lagrange. Par dérivation par rapport à A_1, on obtient :

$$(\Sigma - \lambda I_k)A_1 = 0. \tag{1.37}$$

Par conséquent, λ est une valeur propre de Σ. Or, Σ peut avoir plusieurs valeurs propres. Le choix se fait en maximisant $A_1^\intercal \Sigma A_1 = \lambda$, alors A_1 est le vecteur propre associé qui correspond à la valeur propre maximale de Σ.

En suivant la même démarche, A_2 doit aussi être un vecteur propose de Σ, et doit être non-corrélé avec A_1 , c-à-d., $A_1 A_2 = 0$. Avec ces deux contraintes, la fonction objective par la méthode de Lagrange consiste à maximiser le terme suivant :

$$A_1^\intercal \Sigma A_1 - \lambda(A_1^\intercal A_1 - 1) - \beta(A_1^\intercal A_2), \tag{1.38}$$

où λ et β sont des multiplicateur de Lagrange. Par dérivation par rapport à A_2, on obtient :

$$(\Sigma - \lambda I_k)A_2 - \beta A_1 = 0. \tag{1.39}$$

En multipliant cette équation par $A_1^\intercal$, on obtient :

$$A_1^\intercal \Sigma A_2 - A_1^\intercal \lambda A_2 - \beta A_1^\intercal A_1 = 0.$$

Or, $A_1^\intercal \Sigma A_2 = 0$ et $A_1^\intercal \lambda A_2 = 0$ car $A_1^\intercal A_2 = 0$. Et puisque $A_1^\intercal A_1 = 1$, alors $\beta = 0$. En remplaçant ceci dans l'équation 1.39, et en tenant le fait que $A_2^\intercal A_2 = 1$, on obtient :

$$A_2^\intercal \Sigma A_2 = \lambda. \tag{1.40}$$

Par conséquent, en choisissant la valeur maximale possible, λ correspond à la deuxième valeur propre de Σ. Pour les autres valeurs propres, le même principe

s'applique.

Réduction de dimension

Finalement, les composantes principales sont calculées en se basant sur l'équation 1.34 :

$$[P_1, \ldots, P_k] = [A_1^\intercal Y, \ldots, A_k^\intercal Y] \tag{1.41}$$

Notons qu'on peut choisir un nombre de vecteur propres inférieur ou égal à k, si on veut réduire les données à une taille inférieure à k. Généralement, le choix de k

Il existe bien d'autres méthodes de détermination des composantes principales, que l'on ne va pas détailler ici. Par exemple, la méthode basé sur la décomposition en valeurs singulières (WALL, RECHTSTEINER et ROCHA 2003), qui permettent d'estimer les vecteurs propres de la matrice des covariance sous la forme : $\Sigma = UAV^\intercal$. Cette méthode est une généralisation de décomposition des matrices carrées (matrice de covariance dans le cas de l'ACP), car il permet de décomposer les matrices rectangulaires en général. Dans YATA et AOSHIMA 2010, l'auteur (s) montre que cette approche est plus robuste pour la manipulation des données à grande dimension. La méthode ACP avec noyau (Kernel PCA) est une analyse en composantes principales non linéaire proposée comme une extension de l'ACP, en considérant la corrélation non linéaire entre les variables SCHÖLKOPF, SMOLA et MÜLLER 1998. l'ACP probabiliste introduit dans M. E. TIPPING et C. BISHOP 1999, est une autre variante qui considère un modèle statistique pour déterminer les composantes principales. Cette technique est proche de la formulation du modèle d'Analyse Factorielle (AF) qu'on va détailler dans la partie suivante.

1.4.2.2 L'Analyse Factorielle

De manière similaire à l'ACP, l'Analyse Factorielle (AF) est une méthode de réduction de dimension qui permet de réduire la dimension d'un vecteur de n variables, en les résumant de manière linéaire par un vecteur de facteurs (variables non-observables) de dimension $p \leq n$. Bien que l'AF et l'ACP semblent similaires, car les deux consistent à déterminer des facteurs qui résument mieux les variables, la différence entre eux est fondamentale, notamment dans la manière de déterminer les facteurs.

Considérons un vecteur de n variables $Y = \{Y_1, \ldots, Y_n\}$. Rappelons que l'objectif est de réduire la dimension de Y par un vecteur de variables, appelés facteurs, $[F_1, \ldots, F_k]$ de dimension $k \ll n$. La determination des facteurs (composantes principales dans le cas de l'ACP) pour l'AF est basée sur un modèle statistique multi-varié JOLLIFFE 1986 SCHNEEWEISS et MATHES 1995, ce qui n'est pas le cas pour l'ACP, ou les facteurs dépendent des vecteurs propres de $Y^\intercal Y$. L'idée de

l'AF consiste à construire un modèle ou les variables originales sont exprimées comme des fonctions linéaires des facteurs et des termes d'erreurs. Les équations de ce modèle peuvent s'exprimer comme suit :

$$\begin{aligned} Y_1 &= a_{1,0} + a_{1,1}F_1 + \cdots + a_{1,k}F_k + U_1, \\ &\vdots \\ Y_n &= a_{n,0} + a_{n,1}F_1 + \cdots + a_{n,k}F_k + U_n, \end{aligned}$$

où $a_{i,j}$ sont les paramètres du modèle, et $U^\intercal = [U_1, \ldots, U_n]$ sont les vecteurs des erreurs des modèles associés à chaque variable de Y.

Soit $F^\intercal = [I_n, F_1, \ldots, F_p]$, et A la matrice des paramètres $(a_{i,j})$ de dimension $(n \times (k+1))$, et I_n est le vecteur dont tous les éléments sont égaux à 1 de dimension n. Une représentation matricielle compacte du modèle AF, à partir des équations précédentes est la suivante :

$$Y = AF + U \tag{1.42}$$

1.4.3 Discussion

Les méthodes de sélection des variables et les méthodes de réduction de dimension peuvent jouer le même rôle pour réduire l'ensemble des variables prédictives dans un modèle de prédiction multi-varié. Mais leurs principes sont totalement différents, puisque les méthodes de réduction de dimension génèrent de nouvelles variables artificielles, dites facteurs, qui sont des combinaisons des variables originales. Par contre, les méthode de sélection extraient quelques variables parmi les originales. De ce point de vue, ces deux techniques ont des avantages et inconvénients par rapport au problème de prédiction des séries temporelles avec plusieurs variables. D'un coté, dans la pratique, il est important de savoir exactement les variables réelles les plus important, notamment dans l'analyse descriptive. Puisque c'est une information utile, qui peut être utilisée pour d'autres objectifs que la prédiction.

D'un autre coté, parfois on s'intéresse juste à la construction des entrées des modèles de prédiction sans avoir une signification réelle des données d'entrée des modèles de prédiction. Dans tous les cas, le choix de la meilleure méthode doit être multi-critère. Il dépend non seulement des précisions des prédictions obtenues, mais aussi de la problématique courante. D'où l'importance de tester plusieurs méthodes en vue de détecter la meilleure.

2 Vue générale des méthodes existantes

Dans le chapitre précédent, nous avons présenté les méthodes de réduction et les modèles de prédiction des séries temporelles multi-variées. Dans ce chapitre, nous allons voir comment les combiner pour traiter notre problématique principale, qui est la prédiction des données temporelles massives.

Dans un premier temps, nous formulons la problématique , puis nous rassemblons les travaux existants qui adressent cette dernière. Les approches existantes appartiennent à deux catégories : les modèles qui utilisent les méthodes de réduction de dimension ou les méthodes d'extraction de variables, comme une première étape, puis une étape de prédiction multi-variée sur les variables réduites. La deuxième catégorie contient les modèles de prédiction basés sur les méthodes de régularisation. Dans se cas, la réduction et la prédiction se font au fur et à mesure par le modèle de prédiction lui même.

En étudiant ces travaux, nous avons remarqué que chacun traite la problématique abordée en utilisant des approches de même type. Dans notre cas, nous présentons une vue d'ensemble du problème, et discutons les différentes approches proposées.

Même si dans ce manuscrit nous traitons les approches qui s'exécutent de manière algorithmique ou automatique, dans la pratique, l'existence de beaucoup de variables prédictives est très commune, et les techniques de « connaissance de domaine » devrait toujours être la première étape du pré-sélection des variables, c-à-d., l'utilisation de la connaissance humaine avant l'apprentissage automatique. Par exemple, étant donné les données macroéconomiques dans lesquelles les séries chronologiques sont des observations de différents aspects de l'économie, on peut regrouper les variables de manière cohérente en utilisant la connaissance du domaine. De même, puisque le prix d'un produit peut fortement dépendre du prix des sous-composants utilisés pour construire ce produit, il peut être judicieux de limiter le nombre de variables prédictives utilisées uniquement aux prix des sous-composants. Bien qu'il soit difficile de surestimer son importance, la connaissance du domaine peut être insuffisante pour pré-grouper des variables, en particulier dans les cas de séries temporelles multi-variées avec de nombreuses variables, d'où le besoin d'approches automatiques.

Dans un souci de clarté, nous traitons les modèles existants comme des processus en trois étapes, consistant en la pré-prédiction, la prédiction et la post-prédiction. Et nous discutons aussi les modèles qui n'entraîne que l'étape de prédiction, que nous désignerons comme « l'approche en une seule étape ». Des exemples de telles approches incluent les méthodes de régularisation et les RNAs.

Dans les cas où l'étape de pré-prédiction est requise, le nombre de variables sera réduit soit par les méthodes dites de sélection de variables, soit par la réduction de dimension. Dans le premier cas, un sous-ensemble de variables est sélectionné, et dans le deuxième, de nouveaux variables de dimension réduite sont créés comme combinaisons des originaux. Dans les deux cas, l'étape de prédiction sera traitée sur un espace de variables de dimension inférieure à l'original. Tout modèle de prédiction multi-variée peut être utilisé dans cette étape, ce qui implique que les approches en une étape mentionnées ci-dessus conviennent également.

L'étape de post-prédiction est utilisée pour construire une prédiction pour une variable basée sur les prédiction d'autres variables. Considérons un exemple où il existe n différents modèles de prédiction possibles pour différentes variables : la prédiction de la variable cible peut être construite en moyennant les prédiction des modèles utilisés. Certains modèles utiliseront les étapes pré et post-prédiction. La famille des modèles de facteurs dynamiques est un exemple ; dans l'étape de pré-prédiction, nous recherchons des facteurs communs qui conduisent toutes les séries temporelles observées. Nous prévoyons ensuite les facteurs communs et les utilisons dans la post-prédiction pour reconstruire les variables cibles.

2.1 Formulation du problème

Considérons une série temporelle multidimensionnelle contenant un grand nombre de variables. La question que nous abordons est comment utiliser les informations existantes (l'historique) pour une meilleure prédiction des séries temporelles ? Nous partons logiquement avec les modèles multi-varié puisqu'ils permettent l'utilisation de plusieurs variables. Par conséquent, l'objectif est d'optimiser les modèles de prédiction en présence de plusieurs variables prédictives. D'un coté, ceci à pour objectif de diminuer la complexité du problème, qui augmente avec l'augmentation du nombre de variables. Et d'un autre coté, pour améliorer la précision même des prédictions. Étant donné une variable cible à prédire en utilisant plusieurs variables prédictives, qui ne sont pas forcément liées avec la cible. Autrement dit, dans les séries à grande dimension, on peut souligner deux principales problématiques, (i) l'utilisation de toutes les séries n'améliore pas forcément les performances des prédictions, parfois même impossible à cause des problèmes de résolution des modèles multi-variés, (i) les relations de causalité entre les séries ne sont ni symétriques ni monotones.

Ceci nous mène automatiquement à la formulation suivante : Considérons $Y = \{Y_1, Y_2, \ldots, Y_n\}$ une série temporelle multi-varié de dimension n. L'idée est de sélectionner un sous-ensemble ou bien construire des facteurs de Y de dimension $\leq k$ une qui permettent d'obtenir les meilleures prédictions d'une ou plusieurs variables de Y. D'un point de vue combinatoire, pour prédire une seule variable de Y, il y a $\sum_{i=0}^{k} \binom{n-1}{i}$ sous-ensembles possibles de taille $\leq k$, et de manière

générale, il y a 2^{n-1} sous-ensembles possibles, car :

$$\begin{aligned}\sum_{i=0}^{n-1}\binom{n-1}{i} &= \sum_{i=0}^{n-1}\binom{n-1}{i}a^i b^{n-1-i} \quad \text{où a = 1 et b = 1}\\ &= 2^{n-1} \quad \text{(par formule du binôme.)}\end{aligned}$$

2.2 Modèle factoriels dynamiques

Les modèles factoriels dynamiques (MFD) ont reçu une attention considérable pendant les deux dernières décennies en raison de leur capacité à modéliser des séries temporelles dans lesquelles le nombre de séries est important ou dépasse le nombre d'observations, et ils ont été proposés comme extension des modèles factoriels statiques FORNI, HALLIN, LIPPI et al. 2000. Ils ont été massivement appliqués dans la prédiction des données financières et économiques. Par exemple, on peut se référer à STOCK et M. W. WATSON 2006, STOCK et M. WATSON 2011, et STOCK et M. W. WATSON 2012. Considérons une série temporelle n-dimensionnelle Y_t. Le modèle factoriel classique (AF), comme vu dans la section 1.4.2.2, suppose que chaque variable de Y est une combinaison linéaire d'un vecteur de facteurs de dimension $k < n$, plus un terme d'erreur :

$$Y_1 = a_{1,0} + a_{1,1}F_1 + \cdots + a_{1,k}F_k + U_1, \tag{2.1}$$

$$\vdots \tag{2.2}$$

$$Y_n = a_{n,0} + a_{n,1}F_1 + \cdots + a_{n,k}F_k + U_n, \tag{2.3}$$

ou de manière compacte :

$$Y = AF + U \tag{2.4}$$

où A sont les paramètres du modèle $(a_{i,j})$, $F = [F_1, ..., F_k]^\intercal$ sont les facteurs, et U représente les termes d'erreur. Le MFD à été proposé dans GEWEKE 1977 comme une extension du modèle AF, dans le but d'adapter à des fins de prédiction, de telle sorte que Y puisse être conduit dynamiquement en fonction des facteurs (facteurs dynamiques). Le modèle factoriel dynamique peut être exprimé comme suit :

$$Y(t) = A_p F(t) + U(t), \tag{2.5}$$

$$F(t) = B_1 F(t-1) + ... + B_p F(t-p) + E(t), \tag{2.6}$$

où A_p est les paramètres du modèle factoriel (elle dépend du paramètre p, contrairement au modèle factoriel statique), $[B_1, ..., B_p]$ et E sont *resp*. les paramètres et les termes d'erreurs du modèle VAR.

Le MFD peut être vu comme un processus de prédiction en trois étapes. Pre-

mièrement, les facteurs initiaux sont déterminés en utilisant l'analyse factorielle, ou l'ACP, ensuite, il évoluent en suivant un processus auto-régressif. Dans cette étape, un modèle de prédiction (le plus commun est le modèle VAR ou VAR bayésien) est appliqué à ces facteurs. Autrement dit, les prédictions sont effectuées sur les facteurs, ensuite ces prédictions sont remplacés dans les équations 2.3 pour avoir les prédictions des variables originales.

Cette approche est légèrement différente des méthodes basées sur des techniques de réduction de dimension, pour lesquelles les variables réduites sont déterminées, puis, utilisées dans le modèle multi-varié à côté de la variable cible. Par contre, le MFD détermine la variable cible en fonction des facteurs, puis applique le modèle de prédiction sue ces facteurs (sans variable cible), puis les prédictions de la variable cible sont obtenues selon les équations linéaires avec les facteurs. Dans PEÑA 2009, une étude comparative montre les similitudes entre ces deux approches.

Notons que l'application de l'AF dans ce modèle n'est pas la seule méthode possible pour trouver les facteurs. Par exemple dans STOCK et M. W. WATSON 2002, l'ACP à été appliqué pour réduire le nombre de variables prédictives, puis le modèle VAR a ensuite été utilisé pour prédire les facteurs trouvés par l'ACP, et enfin, la prédiction de la variable cible a été construite en utilisant les relations linéaires entre les variables d'origine et les facteurs. Comme l'approche précédente, il s'agit également d'une approche en trois étapes.

Plus tard, dans STOCK et M. W. WATSON 2012, des expérimentations détaillées ont été réalisées pour prédire des ensembles de données de séries temporelles macroéconomiques des États Unis avec un grand nombre de variables. Diverses méthodes de régularisation ont été comparées au MFD. Les résultats illustrent que ces approches surpassent le MFD pour un nombre considérable de variables (voir la sous-section Méthodes de régularisation 2.4).

2.3 Approches à deux étapes

Cette section est consacrée aux approches qui utilisent la sélection ou la réduction de dimension comme première étape pour la prédiction de séries temporelles. Le traitement de séries temporelles de haute dimension rend inapproprié les modèles de prédiction statistique traditionnels FAN, HAN et Han LIU 2014. Dans le cadre de la prédiction de grandes séries temporelles, une manière de traiter ce problème est d'extraire des relations cachées entre des variables, puis de construire un modèle optimisé multi-varié sur le sous-ensemble réduit des variables prédictives pour chaque variable cible. Ainsi, ces approches pourraient être répertoriés en deux étapes, une étape de réduction, et une étape de prédiction (voir Figure 2.1). Les méthodes de sélection extraient les variables pertinentes, tandis que les méthodes de réduction de dimension génèrent de nouvelles variables (Voir paragraphe 1.4 pour plus de détails).

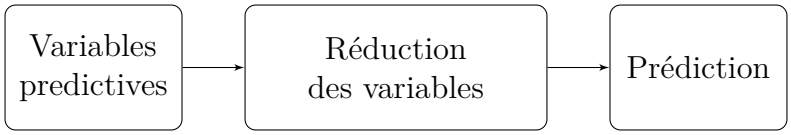


Figure 2.1 – Schéma générale des approches à deux étapes.

2.3.1 Travaux de littérature

Une méthode de réduction de dimension basée sur l'ACP a été utilisée dans ZHONG et ENKE 2017, et a été évaluée à l'aide de données financières. Les auteurs ont proposé un processus de prédiction, dans lequel ils ont premièrement testé trois versions de l'ACP. Ensuite, ils ont utilisé un réseau neuronal artificiel (RNA) entièrement connectée pour la phase de prédiction. Dans ABRAHAM, NATH et MAHANTI 2001, les auteurs proposent un processus hybride pour la prédiction des marchés boursiers et l'analyse des tendances. Une étape de pré-prédiction est réalisée en utilisant l'ACP pour réduire la dimension des variables d'entrée. De même, un RNA multi-couches est utilisé pour prévoir les sorties de stock. Enfin, un système basé sur la logique floue est utilisé pour analyser les tendances des prédictions.

Un algorithme de sélection de variables pour les séries temporelles basées sur l'ACP, l'élimination récursive, et la machine à vecteurs de support a été proposé dans YOON et SHAHABI 2006. L'expérimentation portait sur des ensembles de données du cerveau humain avec un nombre de variables supérieur au nombre d'observations. Dans ce genre de situations, c-à-d, dans le cas où le nombre de variables excède le nombre d'observations, quelques modèles de prédiction sont impossibles à appliquer, notamment les modèles linéaires, d'où l'intérêt de l'étape de réduction.

Dans KOPRINSKA, RANA et AGELIDIS 2015, une approche de prédiction en deux étapes a été introduite pour prévoir deux années de séries temporelles australiennes de l'énergie électrique. Premièrement, des méthodes de corrélation, d'information mutuelle et de sélection de variables ont été appliquées afin d'extraire les variables pertinentes. Ensuite, un RNA multi-couches combiné à des modèles auto-régressifs a été utilisé pour prédire les variables cibles à l'aide des variables prédictives sélectionnés.. Les modèles utilisant le test de causalité de Granger GRANGER 1980 ont également été étudiés dans le contexte de la prédiction des séries temporelles SUN, Jiuyong LI, J. LIU et al. 2014. Les variables ont été sélectionnées si elles avaient une causalité significative sur la cible. Les auteurs ont rapporté de bonnes performances, comparées aux méthodes de réduction des dimensions de sélection. Une approche différente basée sur la causalité de Granger a été proposée dans Youssef HMAMOUCHE, Alain CASALI et LAKHAL 2017. Tout d'abord, un graphe de causalité de Granger a été construit sur la base de la causalité entre les séries temporelles. Ensuite, le graphe a été mis en grappe et la sélection des caractéristiques a été effectuée en fonction du regroupement. Enfin,

le modèle vectoriel de correction d'erreur (VECM) a été utilisé pour prédire la variable cible. L'évaluation est effectuée par rapport à un benchmark composé du modèle ARIMA pour tester les préditions sans prédicteurs, du modèle VECM avec tous les prédicteurs, d'une méthode de sélection de variables basée sur la causalité SUN, Jiuyong LI, J. LIU et al. 2014, et de l'ACP. Le modèle proposé surpasse les autres méthodes sur les ensembles de données évaluées.

2.4 Méthodes de Régularisation

Généralement, les méthodes de régularisation représentent une autre approche pour estimer les coefficients des modèles linéaires qui contiennent de nombreuses variables. Comme nous avons déjà discuté, beaucoup de ces variables peuvent avoir un effet mineur ou nul sur la série cible, et par conséquent, peuvent fausser l'effet des variables importantes.

Ces méthodes consistent à minimiser au fur est à mesure l'impact des variables non pertinentes. Ainsi, un modèle de prédiction après estimation avec de les méthodes de régularisation, peut contenir beaucoup de variables prédictives, avec une régularisation sur les coefficients non importants, par exemple en les poussant près de 0.

2.4.1 Représentation classique

La méthode de régression Ridge à été proposée dans HOERL et KENNARD 1970, dans le but d'étendre la méthode des moindres carrées pour la résolution des systèmes linéaires. Considérons un vecteur de variables prédictives $\{X_1, \ldots, X_k\}$ et une variable cible Y, et pour simplifier, supposons que toutes les variables sont standardisées et contiennent n observations. Notons que la dernière hypothèse est facultative et que le modèle peut être étendu pour les variables non standardisées en ajoutant un coefficient constant.

$$\begin{aligned} &\underset{\beta_1,\ldots,\beta_n}{\text{minimize}} && \sum_{t=1}^{n} (Y(t) - \sum_{i=1}^{k} \beta_i X_i(t))^2 \\ &\text{sous la condition} && \sum_{j=1}^{k} \beta_j^2 \leq \lambda \end{aligned} \tag{2.7}$$

où $\lambda \sum_{j=1}^{k} \beta_j^2$ est le terme de pénalité.

Logiquement, lorsque $\lambda = 0$, le terme de pénalité n'a aucun effet. La sélection d'une valeur appropriée pour λ est importante. Elle peut être effectuée en utilisant, par exemple une validation croisée. Un problème important de la régression avec la méthode Ridge est que les coefficients de variables non pertinentes sont rarement égaux à 0. Ainsi, de nombreuses variables prédictives resterons inclus dans le modèle final. La méthode Lasso TIBSHIRANI 1994 est une autre

méthode de régularisation qui permet résoudre relativement ce problème. Elle fonctionne de la même manière que la régression Ridge, à une exception près, il utilise un terme de pénalité différent, dans le but d'obtenir des coefficients exactement égaux à 0. Pour ce faire, la méthode Lasso est exprimée par les équations suivantes :

$$\begin{aligned} &\underset{\beta_1,\ldots,\beta_n}{\text{minimize}} && \sum_{t=1}^{n} (Y(t) - \sum_{i=1}^{k} \beta_i X_i(t))^2 \\ &\text{sous la condition} && \sum_{j=1}^{k} |\beta_j| \leq \lambda \end{aligned} \tag{2.8}$$

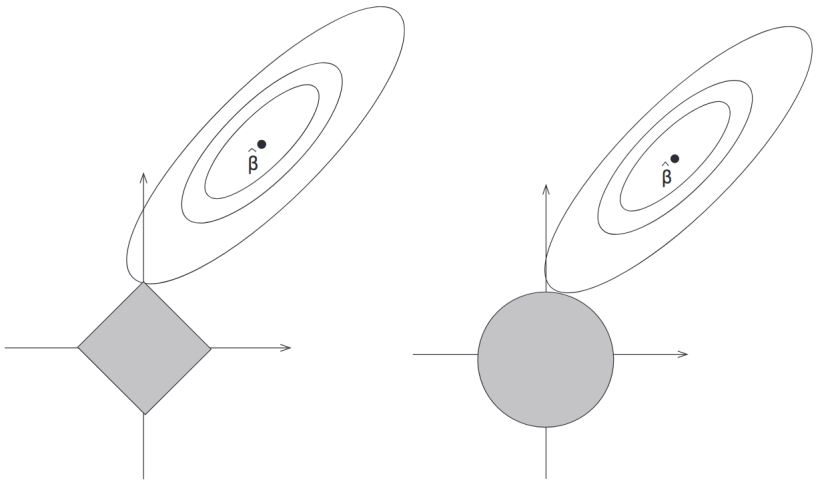


Figure 2.2 – Comparaison géométrique entre la méthode Lasso (à gauche) et la méthode Ridge (figure extraite de Tibshirani 1994).

La Figure 2.2, montre la différence entre l'estimation des coefficients par la méthode Ridge et Lasso. Pour simplicité, focalisons sur le plan 2D, c-à-d., $k = 2$. Dans le cas de la méthode Ridge, la contrainte $\sum_{j=1}^{k} \beta_j^2 \leq \lambda$ est sous forme d'un ellipse, et l'estimation de β est la première l'intersection entre cet ellipse et l'ensemble des solutions selon le critère $\sum_{t=1}^{n} (Y(t) - \sum_{i=1}^{k} \beta_i X_i(t))^2$, qui sont centrées autour de l'estimation des moindres carrées. Tenant en compte cette topologie, il est rare d'avoir des coefficient nuls. Par contre, pour la méthode lasso, la contrainte $\sum_{j=1}^{k} |\beta_j| \leq \lambda$ est un carré, ce qui facilite l'intersection avec ces coins.

Comme nous avons vue dans 1.4.2 pour déterminer les composantes principales de la méthode ACP, lorsque nous voulons optimiser un terme sous une condition, une méthode classique consiste à appliquer le technique de Lagrange. Si on l'applique pour trouver les coefficients des modèles Ridge et Lasso à partir

des deux équations 2.7 et 2.8, on obtient les deux fonctions objectives suivantes :

$$\text{Rgide} : \min_{\beta} \sum_{t=1}^{n} (Y(t) - \sum_{i=1}^{k} \beta_i X_i(t))^2 + \lambda \sum_{j=1}^{p} (\beta_j)^2 \tag{2.9}$$

$$\text{Lasso} : \min_{\beta} \sum_{t=1}^{n} (Y(t) - \sum_{i=1}^{k} \beta_i X_i(t))^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.10}$$

2.4.2 Application aux prédiction des séries temporelles

Comme les méthodes Lasso et Ridge ont été initialement dédiées au problème de régression multiple, elles peuvent être facilement étendue pour fonctionner avec les modèle tel que Vecteur Auto-Régressif (VAR). Considérons une série temporelle k-dimensionnelle $(y_1, \ldots, y_k)$ contenant n observations, et supposons que nous sommes intéressés par la prédiction de la variable y_k. Le modèle VAR dans ce cas peut être exprimé comme suit :

$$Y_i(t) = a_0 + \sum_{i=1}^{p} (a_{1,i} Y_1(t-i) + \cdots + a_{k,i} Y_k(t-i)) + U(t) \quad \forall i \in [1, k] \tag{2.11}$$

où $A = [a_0, a_{1,1}, \ldots, a_{k,p}]$ sont les paramètres du modèle et U est un vecteur de taille $n - p$ représentant les termes d'erreur.

L'estimation classique des paramètres A est généralement effectuée en utilisant la méthode des moindres carrés (LS), qui cherche à minimiser la somme des carrés de U. Cette méthode a quelques inconvénients dans le cas de présence de plusieurs de variables. Si on veut régulariser les coefficients du modèle VAR avec la méthode Lasso par exemple, avec l'équation 2.11, il faut minimiser le terme :

$$\min_{\beta} \sum_{t=1}^{n} (Y(t) - a_0 - \sum_{i=1}^{k} a_i X_i(t))^2 + \lambda \sum_{j=1}^{p} |\beta_j| \tag{2.12}$$

De nombreuses recherches ont été consacrées à l'exploration des méthodes de régularisation dans le contexte de la prédiction des séries temporelles. Dans BAI et NG 2008, une méthode à été proposée basée sur Lasso pour sélectionner un sous-ensemble de variables prédictives. Le sous-ensemble obtenu a été utilisé dans un modèle factoriel dynamique. Différentes variantes de Lasso ont été proposées dans Jiahan LI et CHEN 2014. L'approche proposée permet d'éliminer les variables non pertinents du modèle prédictif. La précision des prédictions résultantes sur les données utilisées était meilleures que la précision du modèle factoriel dynamique. La régression par les moindres angles (LARS) est une autre méthode de régularisation proposée dans EFRON, HASTIE, JOHNSTONE et al. 2004. Le processus de sélection de l'algorithme LARS produit un classe-

ment des variables, où le taux de rétrécissement est contrôlé par le nombre de variables sélectionnées. Une variante de la méthode LARS, proposée dans GELPER et CROUX 2008, a été utilisée pour construire un modèle de prédiction de la croissance de la production industrielle. Le modèle a été évalué dans deux applications. La première consistait à sélectionner des variables pour prédire la croissance de la production industrielle globale aux États-Unis, sur la base d'un total de 131 variables prédictives. Les auteurs ont suivi la procédure décrite dans STOCK et M. W. WATSON 2012 pour prédire la croissance de la production industrielle aux États-Unis. La deuxième application concernait la prédiction de la croissance de la production industrielle en Belgique sur la base de 75 séries temporelles mesurant les indicateurs économiques des pays européens.

3 Travaux de Recherche et Résultats

Comme nous avons vue dans le chapitre précédent, il existe plusieurs approches pour la prédiction des séries temporelles larges. Elles sont généralement basées sur des méthodes de régularisation, ou bien sur des processus à deux étapes visant dans un premier temps à réduire le nombre de variables prédictives avec des méthodes de réduction de dimension ou bien de sélection de variables, ensuite, des modèles de prédiction sont construits à partir de ces variables. Dans ce chapitre, nous allons présenter les travaux réalisés et les contributions amenées dans cette thèse pour l'amélioration des modèles de prédiction multi-variés avec beaucoup de séries temporelles.

Notre contribution se concentre sur une représentation multi-variée et spécifique des séries temporelles à les graphes de causalités, et sur la sélection des variables prédictives.

Dans la première section de ce chapitre, nous présentons la procédure utilisée et le processus de prédiction sur lequel nous avons comparé les différentes méthodes de réduction et les modèles de prédiction y compris nos propositions. Ensuite, dans les Sections 3.2 et 3.3, nous présentons les algorithmes proposés. La quatrième section est dédiée aux expérimentations, où nous présentons les données utilisées et les résultats obtenues. Et finalement, nous discutons ces résultats.

3.1 Notre Procédure de Prédiction

Dans cette section, nous commençons par décrire brièvement le système de prédiction sur lequel nous avons mis en œuvre l'ensemble des méthodes utilisées pour la prédiction des séries temporelles. L'intérêt de cette structure est de pouvoir comparer de manière générique, flexible, et multi-critère, différentes méthodes de réduction et modèles de prédictions. Il s'agit d'un processus de prédiction complet (analyse des dépendances entre les variables, réduction de l'ensemble des variables prédictives pour chaque série temporelle cible, l'application des modèles de prédiction, et finalement, l'analyse des erreurs).

Ce processus est composé de quatre principales étapes comme le monte la Figure 3.1.

— Calcul des graphes de causalités entre les variables (cette étape est utilisée juste par nos méthodes de réduction).

— Réduction de variables : dans cette étape, on exécute toutes les méthodes de réduction, et on génère les données contenant les variables cibles avec les prédicteurs obtenues par chaque méthode.

— Prédiction : exécution des modèles de prédiction.
— Évaluation : évaluation des erreurs de prédiction.

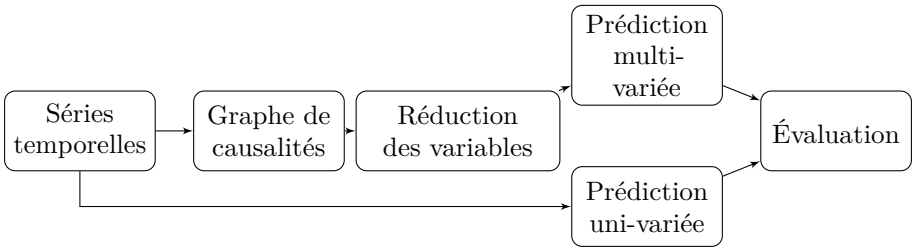


Figure 3.1 – Processus de prédiction.

3.1.1 L'étape de description : calcul des graphes de causalités

L'exploration et l'analyse des réseaux complexes à partir des données numériques est très utilisée dans beaucoup de domaines. Dans le cas des séries temporelles, les relations de dépendances entre les variables sont généralement représentées à l'aide des mesures de similarité.

Les mesures de similarités sont souvent symétriques, et ne modélisent pas les dépendances entre les variables. Par conséquent, elles engendrent des graphes non-orientés et ne contenant pas beaucoup d'information.

Nous représentons les relations entre les séries temporelles en utilisant le graphe de dépendances, où les nœuds sont des séries temporelles et les arcs représentent les valeurs des causalités par paires.

Nous utilisons les mesures de causalités présentées dans 1.3.2 ; la causalité de Granger et le Transfert Entropie, car elles nous permettent de connaitre le sens du transfert d'information d'une variable à une autre. L'idée de cette modélisation est de schématiser les dépendances entre les variables de manière graphique et de résumer les informations cachées entre les variables. Une modélisation similaire a été proposé dans EICHLER 2001 basée sur la causalité de Granger. Dans notre cas, nous exploitons cette représentation pour des objectifs de prédiction.

L'inconvénient de cette représentation est que les graphes engendrés peuvent être de grande dimension dans le cas ou il y beaucoup de variables. Et cela pose quelques problèmes de stockage (temps de transfert de données). Nous discutons brièvement cette problématique et proposons quelques solutions dans la conclusion (3.6.3.3). Un autre point important à souligner est que les causalités entre les variables peuvent changer dans le temps. Cela implique qu'il faut recalculer les graphes de causalités périodiquement ou bien après chaque changement. Aussi, les mesures de causalité utilisées ne permettent pas de mettre à jour les nouveaux valeurs en utilisant les anciennes . Pour remédier à ce problème, nous avons proposer ne version non-linéaire de la causalité de Granger basée sur les

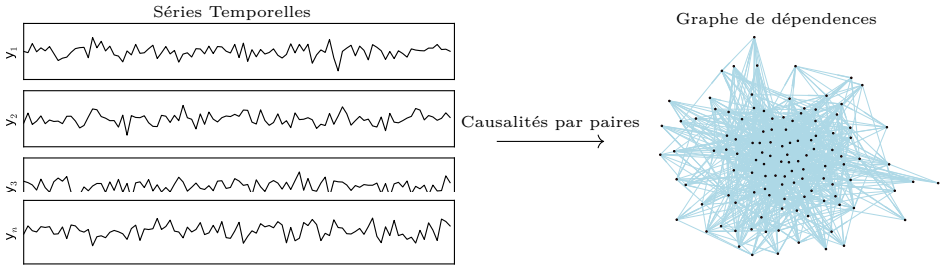


Figure 3.2 – Illustration de la transformation des séries temporelles au graphe de dépendances par des causalités bivariées .

RNAs (voir Section 3.6). Avec cette approche, le premier calcul du graphe sera lent par rapport au test classique de Granger ou bien le Transfert d'entropie, mais si on a de nouvelles observations dans les séries temporelles, le modèle s'adapte plus rapidement grâce au propriétés d'apprentissage des RNAs.

3.1.2 L'étape de réduction et de prédiction

Après le calcul du graphe de causalité, la deuxième étape consiste à réduire la taille des variables prédictives pour chaque variable cible. Ensuite, les modèles de prédiction sont appliqués sur chaque variable cible. On peut résumer ces étapes comme suit :

— Génération des données d'entrée des modèles de prédiction multi-variés (variables prédictives + variable cible), en réduisant l'ensemble des variables disponibles par toutes méthodes de réduction disponibles.
— Application de tous les modèles de prédictions, y compris les modèles univariés.
— Évaluation de la précision des prédictions.

3.2 GFSM : Algorithme d'Extraction de Variables Basé sur le *Clustering*

Dans la littérature, il y a peu de méthodes de sélection de variables qui sont basées sur la causalité ; or cette notion est très important dans le domaine de la prédiction des séries temporelles. Dans cette section, nous présentons notre algorithme proposé GFSM (*Granger Feature Selection Method*) (Youssef HMAMOUCHE, Alain CASALI et LAKHAL 2017, Youssef HMAMOUCHE, Piotr PRZYMUS, Alain CASALI et al. 2017), qui est une nouvelle méthode de sélection des variables spécifique à la prédiction des séries temporelles. Dans ce qui suit, nous décrivons le principe de cette méthode, et justifions son principe dans le cadre du problème de sélection de variables de manière générale.

3.2.1 Principe de la méthode

Considérons $X = \{X_1, X_2, \ldots, X_n\}$ une série temporelle multi-variée et une variable cible Y. L'objectif est de sélectionner un sous-ensemble de X constituant les variables les pertinentes à Y en terme de prédiction, c-à-d., qui permettent d'avoir les meilleurs prédictions de Y dans un modèle multi-varié.

Notre approche se concentre sur la sélection des meilleures variables prédictives en décrivant les relations entre les variables en utilisant la causalité. Nous supposons que la causalité est plus importante que la corrélation lors de la prévision des séries temporelles parce que, contrairement à la corrélation, la causalité modélise les dépendances non symétriques entre les variables. En plus, le fait que deux variables soient corrélées n'identifie pas quelle variable a un impact sur l'autre. Une approche naïve consiste à classer les variables par rapport à leurs causalités vers la cible y. Cette approche a déjà été étudiée dans SUN, Jiuyong LI, J. LIU et al. 2014 et ZHANG, HU, XIE et al. 2014, en utilisant la causalité de Granger. Le principal inconvénient de cette approche consiste à ignorer les relation cachées entre les variables prédictives. Car elles classent les variables directement selon les causalités vers la cible, en ignorant les redondances qui peuvent exister entre elles. Et dans ce cas, il est possible d'utiliser la même source d'informations même avec beaucoup de variables.

Rappelons il existe $\sum_{i=0}^{n} \binom{n}{i}$ parties possibles de taille inférieure ou égale à k, et dans le cas général, c'est-à-dire, sans fixer le nombre de variables prédictives, il y a 2^n parties possibles, ce qui signifie 2^n modèles possibles (STOCK et M. W. WATSON 2006). De plus, la causalité n'est généralement pas une fonction monotone. Par conséquent, trouver le meilleur sous-ensemble de variables qui maximise la causalité est un problème NP-difficile.

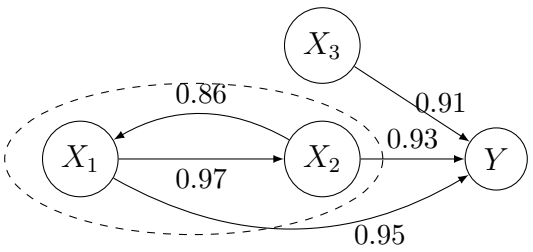


Figure 3.3 – Illustration des dépendances entre séries temporelles à l'aide du graphe de causalité de Granger

Pour bien illustrer ce problème, dans la figure 3.3, nous montrons un petit graphe de causalités décrivant les dépendances entre 4 variables. Essayons de sélectionner deux variables comme prédicteurs pour la variable cible Y. Sélectionner des variables en les classant en fonction de la causalité conduit à obtenir X_1 et X_2. Cependant, X_1 et X_2 pourraient fournir la même information parce que X_1 cause X_2. Dans ce cas, il vaut mieux diversifier les sources d'information pour prédire Y en choisissant soit X_1 soit X_2.

Pour adresser plus précisément ce problème, nous proposons une nouvelle méthode pour adresser ce problème, basée sur l'idée de regroupement des variables dans le graphe de causalités.

3.2.2 L'algorithme GFSM

L'algorithme GFSM se compose de trois étapes :
— Construction de la matrice des causalités entre les variables.
— Élimination des variables ayant une faible causalité sur la cible.
— Regroupement les variables prédictives restantes, en minimisant les causalités entre les *clusters*, et en maximisant la causalité au sein des *cluster*. Dans cette étape, nous utilisons la méthode PAM (*Partitionning arround medoids*), qui exige une matrice symétrique (matrice de distances) pour regrouper les variables. Pour cela, nous symétrisons la matrice de causalité temporairement, en prenant la causalité maximale entre deux variables. Puis, une autre remarque, la méthode PAM minimise les distances entre les groupes. Mais dans notre cas, on veut maximiser les causalités dans les groupes, c'est pour ça on considère la distance comme $1 - max(causalité))$ (line 11 de l'algorithme 2).
— Choisir un élément de chaque groupe, celui qui maximise la causalité sur la variable cible.

Soulignons que la méthode proposée est une généralisation des méthodes de classement, dans le sens où si le nombre de groupes est égal au nombre de variables, alors l'algorithme sélectionnera les k premières variables en les classant selon à la causalité sur la cible. Dans la figure 2, l'algorithme GFSM résume notre approche.

3.3 PEHAR : Algorithme d'Éxtraction de Variables Basée sur le Classement

Dans cette section, nous présentons une nouvelle méthode de sélection de variables. Cette fois-ci, nous utilisons un autre concept pour analysez le graphe de causalités. Il s'agit d'une application personélisée de l'algorithme Hits (*Hyperlink-Induced Topic Search*) (KLEINBERG 1999).

L'algorithme PEHAR est une adaptation particulière et une nouvelle application de l'algorithme Hits. Une idée similaire a été étudiée dans WANG et SU 2002. Il s'agit de classer les éléments qui maximisent un profit dans un modèle de règles d'association en utilisant l'algorithme Hits. Mais à notre connaissance, ce concept n'a pas utilisé auparavant pour sélectionner les variables prédictives pour la prédiction de séries temporelles multiples.

Dan un premier temps, nous présentons le principe général de l'algorithme

Algorithm 2 GFSM.

Input : $X = \{X_1, X_2, \ldots, X_n\}$: l'ensemble des variables prédictives ;
y : la variable cible ;
MinCaus : un seuil minimum de causalité ;
k : la taille de réduction ;
Output : Gfsm-Cl l'ensemble des variables sélectionnées

1: **for** $i = 1$ to n **do**
2: **if** $causality(X_i \rightarrow y) \leq$ MinCaus **then**
3: $X = X \setminus \{X_i\}$
4: **end if**
5: **if** $X.size() \leqslant k$ **then**
6: **return** Gfsm-Cl $= Y$
7: **end if**
8: **end for**
{Construction de la matrice de causalités. }
9: $MC = [[\quad]]$
10: **for all** X_i, X_j in Y, tel que $i \neq j$ **do**
11: $MC[i,j] = MC[j,i] = 1 - max(causality(X_i \rightarrow X_j), causality(X_j \rightarrow X_i))$
12: **end for**
{Regroupement et sélection.}
13: $Clusters = PAM(MC, k)$
14: **for all** Cluster Cl in $Clusters$ **do**
15: Gfsm-Cl $=$ Gfsm-Cl $\cup \arg\max_{cl_j \in Cl} (causality(cl_j \rightarrow y)$
16: **end for**
17: **return** Gfsm-Cl

Hits. Puis, nous décrivons l'idée de notre approche et détaillons l'algorithme PEHAR.

3.3.1 Généralités sur l'algorithme Hits

L'algorithme Hits proposé dans Kleinberg 1999 était à l'origine utilisé pour l'analyse de pages Web. Il a été développé pour détecter les pages les plus pertinentes du graphe des pages Web en analysant les dépendances entre eux. Il considère deux notions de pertinence des pages ; les *Hubs* et les *Authorités*. Et ceci est important dans notre problématique, parce que nous ne sommes pas seulement intéressés à extraire les variables les plus pertinentes, mais nous distinguons entre les variables qui transfèrent l'information, et les variables qui reçoivent l'information. Chaque page à un score *Autorité* et un score *Hub*. Les pages qui ont un bon score *Hub* sont celles qui pointent vers de nombreuses pages (transfert d'information), alors que les pages qui ont un bon score *Autorité* sont celles qui sont les plus pointées.

Le but final de la première application de l'algorithme Hits est de classer les pages en fonction de leurs scores d'*Autorités*. Il calcul les scores *Autorités* et *Hubs* de manière itérative, jusqu'à la convergence. Les équations pour mettre à jour les scores *Authorités* et *Hubs* de chaque page sont les suivantes :

$$a(p) = \sum_{q:q\to p} h(q) \tag{3.1}$$

$$h(p) = \sum_{q:p\to q} a(q). \tag{3.2}$$

Où $a(p)$ et $h(p)$ sont respectivement les scores *Authorités* et *Hubs* de la page p. Notons que l'on peut aussi calculer ces scores en utilisant l'algèbre linéaire, chose que nous allons détailler dans la suite.

3.3.2 Notre approche

Rappelons que notre objectif est de sélectionner les variables les plus pertinentes qui causent la cible tout en minimisant les dépendances entre elles. Ainsi, nous essayons de trouver des variables indépendantes ayant une forte causalité à la cible.

Considérons un ensemble de prédicteurs $P = \{X_1, \ldots, X_n\}$ et une série temporelle cible Y. On calcule d'abord le graphe des dépendances $G = \{V, E\}$ (ou une matrice M) associée à l'ensemble P en utilisant une mesure de causalité prédictive. En adoptant une représentation matricielle, $M[i, j]$ représente la causalité de la variable i vers la variable j. Nous considérons aussi V comme le vecteur des causalités de variables de P vers la cible, c-à-d., $V[i]$ est la causalité de la variable i vers Y. Pour avoir une idée de cette représentation, le graphe illustré dans la figure 3.3 est en fait le vrai graphe des causalités de Granger des jeux de données utilisés dans les expériences (*cf.* Section 3.6.3), qui contiennent 117 séries temporelles.

Notre idée est d'intégrer les intra-dépendances entre les prédicteurs et les dépendances entre les prédicteurs et la cible. Pour ce faire, nous pondérons les scores *Hubs* et *Authorités* avec les causalités *predicteur-prédicteur* et *predicteur-cible* comme indiqué dans les équations suivantes :

$$h(p) = \sum_{q:p\to q} a(q) \times M[p, q] \times V[p], \tag{3.3}$$

$$a(p) = \sum_{q:q\to p} h(q) \times M[q, p] \times V[q]. \tag{3.4}$$

Ces équations peuvent être exprimées de manière compacte comme suit :

$$h = Ga, \tag{3.5}$$

$$a = G^{\intercal}h, \tag{3.6}$$

où : $G[i,j] = M[i,j] \times V[i]$. Avec l'utilisation de cette méthode de pondération, l'algorithme calcule pour chaque variable les scores *Hubs* et *Authorités* en incluant la causalité à la cible. En fin de compte, nous nous intéressons aux variables supérieures en fonction de leurs scores *Hubs*, car ce sont les variables qui transmettent plus information dans le graphe en tenant en compte la cible.

3.3.3 L'algorithme PEHAR

Dans cette partie, nous décrivons l'algorithme proposé ; (*cf.* Algorithme 3) *PEHAR*(*Predictors Extraction using Hubs and Authorités Ranking*) (Youssef HMAMOUCHE, LAKHAL et Alain CASALI (accepted) 2018). La méthode exacte pour trouver les *Hubs* et les *Authorités* est basée sur la résolution matricielle en utilisant l'algèbre linéaire BENZI, ESTRADA et KLYMKO 2013. Plus précisément, il s'agit de la décomposition en vecteurs propres des matrices diagonalisables. En remplaçant les vecteurs *Hubs* et *Authorités* dans les équations 3.5 et 3.6, on obtient les équations suivantes :

$$a = G^{\intercal}Ga, \tag{3.7}$$

$$h = GG^{\intercal}h. \tag{3.8}$$

On peut remarquer que le vecteur *Authorités* converge vers le vecteur propre de la matrice $G^{\intercal}G$, et le vecteur *Hubs* converge vers le vecteur propre de la matrice $GG^{\intercal}$. Dans notre cas, nous nous intéressons seulement au vecteur propre de $GG^{\intercal}$, celui qui correspond à la première valeur propre.

Cette approche peut être coûteuse en termes de temps de calcul, car elle calcule tous les vecteurs propres, surtout si la matrice G est de grande dimension et dense. Par conséquent, nous adoptons une résolution itérative basée sur la méthode de puissance itérée, car l'algorithme converge généralement en un nombre raisonnable d'itérations.

Cette technique consiste à exécuter les équations 3.3 et 3.4 de façon itérative avec une normalisation à chaque étape jusqu'à la stabilité. Ainsi, l'utilisateur peut contrôler le temps de calcul à travers le nombre d'itérations, ou à travers la précision de l'algorithme. Nous considérons que les valeurs des dépendances entre variables sont calculées séparément, de sorte que l'algorithme prend en entrée la matrice des causalités *predictor-predictor* et le vecteur des causalités *predictor-target*.

Algorithm 3 *PEHAR.*

Input : M : Matrice de causalities *prédicteur-prédicteur* ;
V : Vecteur de causalités *prédicteur-cible* ;
n : le nombre d'itérations ;
min_error : erreur minimum ;
k : la taille de réduction ;

1: $h(p) \leftarrow \frac{1}{k}; \quad p = 1, \ldots, k$
2: **while** $i < n$ **do**
3: $hlast \leftarrow h$;
4: $h(p) \leftarrow 0; \quad p = 1, \ldots, k$
5: $a(p) \leftarrow 0; \quad p = 1, \ldots, k$
6: **for all** $p \in [1, k]$ **do**
7: **for all** $q \in \{[1, k], p \neq q\}$ **do**
8: $a(q) \leftarrow a(q) + hlast(p) \times M[p, q] \times V[p]$;
9: **end for**
10: **end for**
11: **for all** $p \in [1, k]$ **do**
12: **for all** $q \in \{[1, k], p \neq q\}$ **do**
13: $h(p) \leftarrow h(p) + a(q) \times M[p, q] \times V[p]$;
14: **end for**
15: **end for**
16: $s_a \leftarrow max(h)$;
17: $s_h \leftarrow max(a)$;
18: $h(p) \leftarrow h(p)/s_h; \quad p = 1, \ldots, k$
19: $a(p) \leftarrow a(p)/s_a; \quad p = 1, \ldots, k$
20: $error \leftarrow 0$;
21: **for all** $p \in [1, k]$ **do**
22: $error \leftarrow error + abs(h(p) - hlast(p))$;
23: **end for**
24: **if** $error > min\_error$ **then**
25: **return** a ;
26: **end if**
27: **end while**
28: **return** a ;

3.3.4 Exemple

Dans cet exemple, nous appliquons l'algorithme PEHAR sur un très petit ensemble de variables. Considérons un ensemble de 5 prédicteurs ; $P = \{X_1, \ldots, X_5\}$ et une variable cible Y. Considérons aussi M comme la matrice des causalités de P, et le V le vecteur des causalités des variables de P vers Y (les mêmes notations utilisées dans la sous-section 3.3.2) :

$$M = \begin{pmatrix} 0 & 0.38 & 0.52 & 0.51 & 0.70 \\ 0.88 & 0 & 0.91 & 0.401 & 0.89 \\ 0.89 & 0.34 & 0 & 0.96 & 0.71 \\ 0.95 & 0.62 & 0.56 & 0 & 0.67 \\ 0.92 & 0.96 & 0.99 & 0.77 & 0 \end{pmatrix}, \quad V = \begin{pmatrix} 0.07 \\ 0.90 \\ 0.65 \\ 0.16 \\ 0.35 \end{pmatrix}.$$

Nous calculons la matrice G comme indiqué dans la section 3.3.2 en utilisant la formule ; $G[i,j] = M[i,j] \times V[i]$. On obtient :

$$G = \begin{pmatrix} 0 & 0.026 & 0.036 & 0.035 & 0.049 \\ 0.792 & 0 & 0.819 & 0.360 & 0.801 \\ 0.578 & 0.221 & 0 & 0.624 & 0.461 \\ 0.152 & 0.099 & 0.0896 & 0 & 0.107 \\ 0.322 & 0.336 & 0.3465 & 0.2695 & 0 \end{pmatrix}.$$

Comme discuté dans 3.3.3, la méthode de résolution algébrique pour trouver le vecteur *Hub* est basée sur les vecteurs propres de la matrice $GG^\intercal$. En calculant tout les vecteurs propres normalisés (norme l_1) de $G^\intercal G$, et en les classant selon des valeurs propres, le premier vecteur propre obtenu est le suivant :

$$V_{alg}^\intercal = \begin{pmatrix} 0.0196 & 0.4639 & 0.2853 & 0.0661 & 0.1651 \end{pmatrix}.$$

Si nous appliquons la méthode d'itération de puissance comme détaillé dans l'algorithme 3, alors nous obtenons le vecteur *Hub* suivant en seulement 3 itérations :

$$V_{iter}^\intercal = \begin{pmatrix} 0.0196 & 0.4638 & 0.2854 & 0.0661 & 0.1651 \end{pmatrix}.$$

La dernière étape consiste simplement à classer les variables en fonction des scores *Hub* obtenues. Dans l'exemple, malgré la faible différence entre les deux vecteurs obtenus, ils conduisent au même classement.

3.4 Expérimentations

Cette section est consacrée au expérimentations. Nous commençons par décrire les ensembles de données utilisées. Ensuite nous montrons les méthodes et modèles mises en œuvre, la méthodologie de prédiction adoptée, et finalement, nous discutons les résultats obtenus.

3.4.1 Les données utilisées

Nous utilisons deux ensembles de données, qui sont déjà utilisées dans la littérature, ce qui va nous permettre de faire des comparaisons avec d'autres approches :

— Données macroéconomiques trimestrielles des États-Unis : 143 variables et 200 observations, années $1960 - 2008$ utilisées par J. H. Stock et M. W. Watson STOCK et M. W. WATSON 2012.
— Données macroéconomiques trimestrielles d'Australie : 117 variables et 119 observations, années $1984-2015$ JIANG, ATHANASOPOULOS, HYNDMAN et al. 2017.

Ces ensembles de données ont été utilisés dans le contexte de la prévision avec de nombreux prédicteurs dans STOCK et M. W. WATSON 2012 ; JIANG, ATHANASOPOULOS, HYNDMAN et al. 2017. Dans ces deux travaux, différents modèles ont été évalués en utilisant ces ensembles de données, le modèle AR (4) comme modèle de référence, le modèle de facteurs dynamiques, qui utilise l'analyse des composantes principales et le modèle VAR, et d'autres modèles basés sur les méthodes de régularisation.

3.4.2 Les méthodes de réduction évaluées

Nous présentons dans cette partie les techniques de réduction de dimension et de sélection des variables évaluées. Nous utilisons le module *Python* Scikit-learn d'apprentissage automatique PEDREGOSA, VAROQUAUX, GRAMFORT et al. 2011, pour l'implémentation des techniques de réduction de dimension suivantes :
— L'analyse en composante principales (PCA) Michael E. TIPPING et C. M. BISHOP 1999.
— L'analyse en composante principales à noyaux (KernelPCA) SCHÖLKOPF, SMOLA et MÜLLER 1998.
— L'analyse factorielle (FA) JOLLIFFE 1986.
— L'algorithme FCBF (Fast Correlation-Based Filter) YU et Huan LIU 2003.

Et nous avons implémenté les deux méthodes de sélection de variables suivantes :
— Deux versions de chaque algorithme proposé sont implémentés, en utilisant deux mesures de causalité prédictives, la causalité de Granger et l'Entropie de Transfert ; GFSM-gc, GFSM-te, PEHAR-gc, et PEHAR-te.

3.4.3 Les modèles de prédiction utilisés

Les modèles de prédiction utilisés peuvent être classés en trois catégories ; les modèles statistiques, les modèles de régularisation, et les RNAs. La formulation mathématique de ces modèles est détaillée dans la Section 1.2.
— Modèles statistiques : nous utilisons le modèle VECM. Il utilise les relations cointégration de séries temporelles non stationnaires, ce qui constitue un inconvénient majeur de certains modèles auto-régressifs, pour exploiter les dépendances à long terme entre des variables. Nous utilisons également également le modèle ARIMA $(4, d, q)$ (avec une détermination automatique des paramètres d et q à l'aide du critère d'information d'Akaike (AIC)). Ce

modèle est utilisé comme un modèle de référence, pour analyser les prédictions sans l'utilisation de prédicteurs. L'ordre des modèles (p) est fixé à une valeur maximale $p = 4$ pour pourvoir comparer les résultats avec les travaux de STOCK et M. W. WATSON 2012 et JIANG, ATHANASOPOULOS, HYNDMAN et al. 2017, où ils ont utilisé cette valeur qui correspond à un historique d'une année (données trimestrielles).

— Modèles de régularisation : Nous utilisons dans cette catégorie les modèles suivant ; Ridge bayesian (BayeRidge), Ridge et Lasso. Et nous utilisons différentes valeurs du paramètre de régularisation ($\alpha \in \{0.1, 0.2, 0.3\}$).

— RNAs : la stratégie adoptée pour les RNAs est un peu différente de celle des modèles statistiques car elle nécessite une étape de préparation de la structure des données d'entrées, en transformant le problème en un problème d'apprentissage supervisé via le paramètre de temporisation, qui est 4 dans notre cas, comme utilisé dans STOCK et M. W. WATSON 2012 et JIANG, ATHANASOPOULOS, HYNDMAN et al. 2017, pour pouvoir comparé avec leurs résultats. Nous utilisons le modèle LSTM CHOLLET et al. 2015. C'est un RNR particulier qui supporte les décalages temporels des variables d'entrée, et il devient très utile pour la prédiction de séries chronologiques (*cf.* voir 1.2.4 pour plus de détails). La structure adoptée est basée sur une couche cachée en utilisant l'algorithme de gradient stochastique pour entrainer le réseau.

3.4.4 Procédure de prédiction

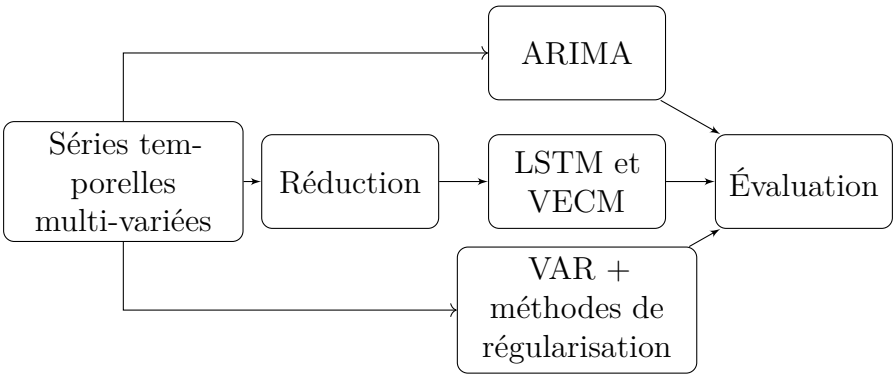


Figure 3.4 – Le processus de prédiction.

Nous avons implémenté un processus de prédiction automatique comme décrit dans la Section 3.1. Il consiste d'abord à réduire le nombre de prédicteurs, ensuite à appliquer des modèles de prédiction, et enfin à évaluer la qualité des prédiction obtenues (*cf*. Figure 3.4).

L'étape de réduction n'est pas requise pour les modèles uni-variés, car ils considèrent uniquement la variable cible pour effectuer des prédictions. Pour les mé-

thodes de réduction, nous avons testé différentes valeurs de nombre de prédicteurs k (de 1 à 20), car nous avons remarquer que l'utilisation de plus de 20 de variables diminuait la précision de la prédiction. L'étape de préparation consiste à transformer des séries temporelles en stationnaires par différenciation, et cette étape n'est pas obligatoire pour tous les modèles tels que VECM et ARIMA car ils l'intègre automatiquement.

3.4.5 Procédure d'évaluation des prédictions

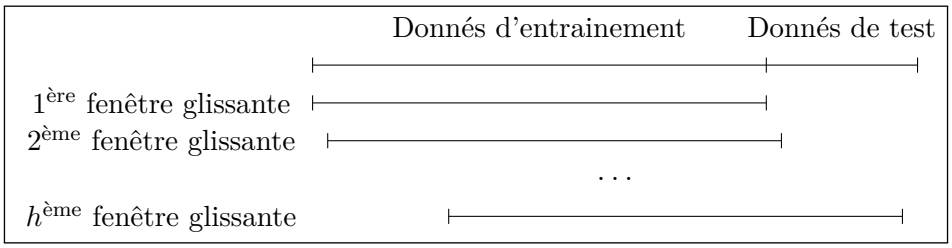


Figure 3.5 – Méthode d'évaluation.

Nous adoptons les mêmes procédures d'évaluation des prédiction que celles utilisées dans les deux travaux STOCK et M. W. WATSON 2012 et JIANG, ATHANASOPOULOS, HYNDMAN et al. 2017, qui ont été effectuées sur les deux jeux de données *Stock-and-Watson* et *Ausmacrodata resp*. Nous considérons un paramètre de temporisation (p) équivalent à 4 pour les modèles de prédiction. Le nombre de prédictions utilisé pour la construction des modèles est 100 pour les données US, et 75 pour les données d'Australie. Le paramètre de temporisation pour le calcul de la causalité de Granger est déterminé automatiquement en utilisant le critère d'information Akaikes (AIC) avec une valeur maximale de 4. Enfin, les prédictions sont effectuées en utilisant une procédure de fenêtres glissantes (*cf.* Figure 3.5), en avançant un pas après chaque prédiction.

Pour chaque sous-échantillon de fenêtre glissante :

1. Estimer chaque modèle.
2. Estimer les prédictions.
3. Calculer les erreurs pour chaque prévision.

Deux mesures de précision des prédiction sont utilisées ; l 'erreur moyen quadratique RMSE, et MASE (mean absolute scaled error) HYNDMAN p.d., qui est une nouvelle mesure basée sur les erreurs des prédiction et l'erreur absolue moyenne de la méthode naïve. Les expressions de ces deux mesures sont les

suivantes :

$$\text{RMSE} = \sqrt{\frac{\sum_{t=1}^{h}(Y(t+n) - \hat{Y}(t+n))^2}{h}},$$
$$\text{MASE} = \frac{\frac{1}{h}\sum_{t=n+1}^{n+h} \mid Y(t) - \hat{Y}(t) \mid}{\frac{1}{n-1}\sum_{t=2}^{n} \mid Y(t) - Y(t-1) \mid},$$

où $(\hat{Y}(n+1), \ldots, \hat{Y}(n+h))$ sont les prédictions, $(Y(n+1), \ldots, Y(n+h))$ sont les valeurs réelles.

3.4.6 Résultats

Nous présentons dans cette partie les résultats obtenus. Pour des contraintes de lisibilité et de volume des résultats, nous montrons juste les meilleures prédictions pour chaque variable. C'est-à-dire, après avoir exécuté toutes les méthodes de réduction et modèles de prédiction sur toutes les variables, nous montrons juste celles qui permettent d'obtenir les meilleures scores de prédiction. La première colonne de chaque de chaque table contient les noms des variables. Pour chaque variable, on montre les meilleures performances obtenues, la meilleure méthode de réduction, le meilleure modèle de prédiction, et le nombre de prédicteurs., La Table 3.1 représente une forme générale des résultats obtenues, et qui montre comment lire les résultats à partir des tables qu'on va représenter dans la suite, où :

— val_i est la meilleure valeur obtenue de la mesure de précision utilisée pour variable i,
— méthode_i et modèle_i sont respectivement la méthode de réduction et le modèle de prédiction qui permettent d'aboutir à val_i,
— nb_i est le nombre de variables prédictives obtenues par méthode_i pour variable_i.

Pour plus de détails sur toutes les résultats des prédictions obtenues, nous referrons le lecteur à Y. HMAMOUCHE 2018.

Table 3.1 – Modèle des tables des expérimentations.

Séries temporelles	**Paramètres de la meilleure prédiction**			
	Mesure de précision	**Méthodes de réduction**	**Modèles de prédiction**	**Nombres de prédicteurs**
variable$_1$	val_1	méthode_1	modèle_1	nb_1
variable$_2$	val_2	méthode_2	modèle_2	nb_2
...				
variable$_n$	val_n	méthode_n	modèle_n	nb_n

3.4.6.1 Résultats détaillés des données des État Unis

Nous montrons ici les résultats des variables des données *Stock-and-Watson*.

Table 3.2 – Évaluations par rapport à RMSE des prédictions des données des États Unis (1).

Séries temporelles	Paramètres de la meilleure prédiction			
	RMSE	Méthodes de réduction	Modèles de prédiction	Nombres de prédicteurs
BUSLOANS	0.0064	KernelPCA	Vecm	6
CCINRV	0.0037	Arima	Arima	0
CES002	0.0028	FCBF	Vecm	1
CES003	0.0130	PEHAR-gc	Vecm	1
CES006	0.0176	PEHAR-te	Vecm	6
CES011	0.0081	PEHAR-te	Vecm	4
CES015	0.0093	PEHAR-gc	Vecm	1
CES017	0.0117	PEHAR-gc	Vecm	1
CES033	0.0077	PCA	Vecm	1
CES046	0.0019	FA	Vecm	2
CES048	0.0040	KernelPCA	Vecm	2
CES049	0.0040	KernelPCA	Vecm	1
CES053	0.0046	FCBF	Vecm	1
CES088	0.0030	FA	Vecm	9
CES140	0.0042	PEHAR-gc	Lstm	1
CES151	0.0685	PEHAR-te	Lstm	1
CES155	0.0421	KernelPCA	Vecm	2
CES275R	0.0107	PEHAR-gc	Lstm	1
CES277R	0.0119	Arima	Arima	0
CES278.R	0.0117	PEHAR-gc	Lstm	1
CPIAUCSL	0.0045	PEHAR-gc	Vecm	2
CPILFESL	0.0013	Arima	Arima	0
EXRCAN	0.0515	PEHAR-gc	Lstm	1
EXRJAN	0.0253	Arima	Arima	0
EXRSW	0.0247	Arima	Arima	0
EXRUK	0.0453	Arima	Arima	0
EXRUS	0.0476	PEHAR-te	Lstm	1
FM1	0.0087	FA	Vecm	1
FM2	0.0033	PEHAR-gc	Vecm	3
FMFBA	0.0369	FA	Vecm	10
FMRRA	0.0889	PEHAR-gc	Vecm	5
FSDJ	0.0322	PEHAR-gc	Lstm	5
FSDXP	0.0322	PEHAR-gc	Vecm	1
FSPCOM	0.0371	PEHAR-gc	Lstm	4
FSPIN	0.0378	GFSM-gc	Lstm	3
FSPXE	0.0647	PEHAR-te	Lstm	2
FYAAAC	0.0315	Arima	Arima	0
FYBAAC	0.0315	PEHAR-gc	Lstm	3
FYFF	0.0294	Arima	Arima	0
FYGM3	0.0316	Arima	Arima	0
FYGM6	0.0322	FA	Vecm	1
FYGT1	0.0345	Arima	Arima	0
FYGT10	0.0379	Arima	Arima	0
FYGT5	0.0376	PEHAR-te	Lstm	1
GDP251	0.0051	FCBF	Vecm	1
GDP252	0.0040	GFSM-gc	Lstm	1
GDP253	0.0155	PEHAR-gc	Vecm	1
GDP254	0.0071	FCBF	Vecm	1
GDP255	0.0028	GFSM-gc	Lstm	4
GDP256	0.0206	KernelPCA	Vecm	1
GDP257	0.0114	PEHAR-te	Vecm	2

Table 3.3 – Évaluations par rapport à RMSE des prédictions des données des États Unis (2).

Séries temporelles	Paramètres du meilleure prédiction			
	RMSE	Méthodes de réduction	Modèles de prédiction	Nombres de prédicteurs
GDP258	0.0116	FA	Vecm	1
GDP259	0.0307	GFSM-gc	Lstm	4
GDP260	0.0123	GFSM-gc	Lstm	4
GDP261	0.0170	FCBF	Vecm	1
GDP263	0.0131	PEHAR-gc	Lstm	6
GDP264	0.0103	FA	Vecm	7
GDP265	0.0093	PCA	Lstm	2
GDP266	0.0247	Arima	Arima	0
GDP267	0.0061	Arima	Arima	0
GDP272A	0.0019	PEHAR-gc	Lstm	5
GDP273A	0.0033	PEHAR-gc	Lstm	5
GDP274A	0.0050	Arima	Arima	0
GDP274_1	0.0069	Arima	Arima	0
GDP274_2	0.0070	PEHAR-gc	Vecm	3
GDP274_3	0.0066	GFSM-gc	Lstm	1
GDP275A	0.0089	PEHAR-gc	Vecm	3
GDP275_1	0.0028	PEHAR-gc	Vecm	1
GDP275_2	0.0115	Arima	Arima	0
GDP275_3	0.0430	PEHAR-gc	Lstm	4
GDP275_4	0.0037	PEHAR-gc	Lstm	3
GDP276A	0.0015	PEHAR-gc	Lstm	2
GDP276_1	0.0014	GFSM-gc	Lstm	1
GDP276_2	0.0080	PEHAR-gc	Lstm	7
GDP276_3	0.0149	GFSM-gc	Lstm	9
GDP276_4	0.0044	PEHAR-gc	Lstm	5
GDP276_5	0.0067	PEHAR-gc	Lstm	2
GDP276_6	0.0017	GFSM-gc	Lstm	5
GDP276_7	0.0028	PEHAR-te	Lstm	1
GDP276_8	0.0041	PEHAR-gc	Lstm	2
GDP277A	0.0033	PEHAR-gc	Vecm	1
GDP278A	0.0032	PEHAR-te	Vecm	1
GDP279A	0.0041	PEHAR-gc	Lstm	1
GDP280A	0.0036	Arima	Arima	0
GDP281A	0.0059	PEHAR-gc	Vecm	1
GDP282A	0.0046	KernelPCA	Vecm	7
GDP284A	0.0107	PCA	Vecm	8
GDP285A	0.0194	PEHAR-gc	Vecm	3
GDP286A	0.0041	FA	Vecm	3
GDP287A	0.0051	PEHAR-te	Lstm	2
GDP288A	0.0044	FA	Vecm	10
HHSNTN	0.1030	Arima	Arima	0
HSBR	0.0546	Arima	Arima	0
HSFR	0.0525	PEHAR-gc	Lstm	1
HSMW	0.0647	Arima	Arima	0
HSNE	0.0620	Arima	Arima	0
HSSOU	0.0617	PEHAR-te	Vecm	1
HSWST	0.0714	GFSM-gc	Lstm	10
IPS10	0.0077	PEHAR-gc	Vecm	1
IPS11	0.0071	FA	Vecm	1
IPS12	0.0094	GFSM-te	Lstm	10
IPS13	0.0182	FCBF	Lstm	4
IPS18	0.0097	PEHAR-te	Lstm	4
IPS25	0.0096	FA	Vecm	4
IPS299	0.0071	FA	Vecm	1
IPS306	0.0249	Arima	Arima	0
IPS32	0.0097	PEHAR-gc	Vecm	2
IPS34	0.0113	FA	Vecm	3
IPS38	0.0136	FA	Vecm	1
IPS43	0.0084	FCBF	Vecm	2

Table 3.4 – Évaluations par rapport à RMSE des prédictions des données des États Unis (3).

Séries temporelles	Paramètres du meilleure prédiction			
	RMSE	**Méthodes de réduction**	**Modèles de prédiction**	**Nombres de prédicteurs**
LBLCPU	0.0080	GFSM-gc	Lstm	4
LBMNU	0.0086	FA	Vecm	1
LBOUT	0.0076	FA	Lstm	3
LBPUR7	0.0131	PEHAR-gc	Lstm	10
LHEL	0.0296	Arima	Arima	0
LHELX	0.0200	Arima	Arima	0
LHEM	0.0049	KernelPCA	Lstm	1
LHNAG	0.0049	PEHAR-te	Vecm	2
LHU14	0.0396	Arima	Arima	0
LHU15	0.0301	FA	Vecm	1
LHU26	0.0371	FA	Lstm	1
LHU27	0.0296	GFSM-te	Lstm	2
LHU5	0.0423	PCA	Lstm	2
LHU680	0.0365	PEHAR-gc	Lstm	9
LHUR	0.0240	PEHAR-te	Vecm	1
MOCMQ	0.0196	PCA	Vecm	5
MSONDQ	0.0396	FA	Vecm	1
MZMSL	0.0049	PCA	Lstm	6
PCEPILFE	0.0014	PEHAR-te	Lstm	1
PMCP	0.1134	FA	Vecm	3
PMDEL	0.0315	KernelPCA	Lstm	1
PMI	0.0670	KernelPCA	Vecm	2
PMNO	0.0926	KernelPCA	Vecm	2
PMNV	0.0512	KernelPCA	Lstm	2
PMP	0.0898	KernelPCA	Vecm	2
PSCCOMR	0.0273	PEHAR-gc	Lstm	1
PW561R	0.0711	FA	Vecm	4
Sfygm6	0.0640	PEHAR-gc	Lstm	2
Sfygt1	0.0757	Arima	Arima	0
Sfygt10	0.0867	PEHAR-gc	Lstm	1
UTL11	0.0316	KernelPCA	Lstm	1
sFYAAAC	0.0802	PEHAR-gc	Lstm	3
sFYBAAC	0.0605	PEHAR-te	Vecm	1

Table 3.5 – Évaluations par rapport à MASE des prédictions des données des États Unis (1).

Séries temporelles	Paramètres du meilleure prédiction			
	MASE	**Méthodes de réduction**	**Modèles de prédiction**	**Nombres de prédicteurs**
BUSLOANS	3.6188	KernelPCA	Vecm	7
CCINRV	3.4517	Arima	Arima	0
CES002	0.6656	FCBF	Vecm	1
CES003	0.4100	PCA	Vecm	3
CES006	0.4392	PEHAR-gc	Vecm	1
CES011	0.6440	PEHAR-te	Vecm	4
CES015	0.3474	PEHAR-gc	Vecm	1
CES017	0.3482	Arima	Arima	0
CES033	0.4045	PCA	Vecm	1
CES046	1.1360	FA	Vecm	2
CES048	1.0001	PEHAR-gc	Lstm	6
CES049	0.9688	KernelPCA	Vecm	2
CES053	1.0700	PEHAR-gc	Lstm	7
CES088	2.1276	Arima	Arima	0
CES140	0.9052	PEHAR-gc	Lstm	1
CES151	0.4666	PEHAR-te	Lstm	1
CES155	0.6228	PCA	Vecm	2
CES275R	0.6595	Arima	Arima	0
CES277R	0.5677	Arima	Arima	0
CES278.R	0.5925	PEHAR-te	Lstm	1
CPIAUCSL	2.2936	PEHAR-gc	Lstm	1
CPILFESL	0.9563	Arima	Arima	0
EXRCAN	2.0668	PEHAR-gc	Lstm	1
EXRJAN	0.8980	Arima	Arima	0
EXRSW	0.9382	Arima	Arima	0
EXRUK	0.9595	Arima	Arima	0
EXRUS	1.6600	PEHAR-te	Lstm	1
FM1	4.2302	FA	Vecm	1
FM2	3.4094	PCA	Vecm	5
FMFBA	24.7211	Arima	Arima	0
FMRRA	31.9910	Arima	Arima	0
FSDJ	6.0728	PEHAR-gc	Lstm	5
FSDXP	0.4758	PEHAR-gc	Lstm	2
FSPCOM	6.6417	PEHAR-gc	Lstm	4
FSPIN	7.1873	GFSM-gc	Lstm	1
FSPXE	1.8015	KernelPCA	Lstm	2
FYAAAC	0.9034	FA	Vecm	1
FYBAAC	1.2145	Arima	Arima	0
FYFF	0.3779	Arima	Arima	0
FYGM3	0.4454	FCBF	Vecm	1
FYGM6	0.4648	Arima	Arima	0
FYGT1	0.5046	FA	Lstm	1
FYGT10	0.8699	Arima	Arima	0
FYGT5	0.7553	PEHAR-te	Lstm	1
GDP251	0.8428	PEHAR-gc	Lstm	6
GDP252	1.0305	GFSM-gc	Lstm	1
GDP253	1.7668	PEHAR-gc	Vecm	1
GDP254	1.2633	PEHAR-gc	Lstm	9
GDP255	1.3298	GFSM-gc	Lstm	4
GDP256	0.9900	KernelPCA	Vecm	1
GDP257	1.3719	PEHAR-te	Vecm	2
GDP258	2.1443	PCA	Vecm	1
GDP259	1.1546	PCA	Lstm	1
GDP260	2.7255	GFSM-gc	Lstm	4

Table 3.6 – Évaluations par rapport à MASE des prédictions des données des États Unis (2).

Séries temporelles	Paramètres du meilleure prédiction			
	MASE	**Méthodes de réduction**	**Modèles de prédiction**	**Nombres de prédicteurs**
GDP261	0.6247	GFSM-te	Lstm	2
GDP263	1.4157	GFSM-te	Lstm	9
GDP264	1.4659	FA	Vecm	7
GDP265	0.9163	PEHAR-gc	Vecm	1
GDP266	0.9762	Arima	Arima	0
GDP267	0.8978	Arima	Arima	0
GDP272A	1.7920	PEHAR-gc	Lstm	2
GDP273A	2.4272	PEHAR-gc	Lstm	1
GDP274A	1.1307	GFSM-gc	Lstm	10
GDP274_1	1.2555	PCA	Vecm	1
GDP274_2	1.6578	FCBF	Vecm	2
GDP274_3	1.8808	GFSM-gc	Lstm	1
GDP275A	2.9338	PEHAR-te	Lstm	1
GDP275_1	1.0077	PEHAR-te	Lstm	1
GDP275_2	1.7652	Arima	Arima	0
GDP275_3	4.9881	PEHAR-gc	Lstm	2
GDP275_4	2.5394	GFSM-gc	Lstm	8
GDP276A	1.8032	PEHAR-gc	Lstm	2
GDP276_1	1.7120	GFSM-gc	Lstm	1
GDP276_2	3.1734	PEHAR-gc	Lstm	7
GDP276_3	4.6699	PEHAR-gc	Lstm	3
GDP276_4	1.4524	GFSM-te	Lstm	4
GDP276_5	0.8370	PEHAR-gc	Lstm	2
GDP276_6	1.8824	GFSM-gc	Lstm	5
GDP276_7	1.7637	PCA	Lstm	1
GDP276_8	2.7843	PEHAR-gc	Lstm	2
GDP277A	1.1092	PEHAR-gc	Vecm	1
GDP278A	1.1144	PEHAR-te	Vecm	1
GDP279A	1.3038	PEHAR-gc	Lstm	1
GDP280A	1.8424	Arima	Arima	0
GDP281A	1.4317	PEHAR-gc	Vecm	1
GDP282A	1.6610	KernelPCA	Vecm	7
GDP284A	1.0928	PCA	Vecm	8
GDP285A	2.3979	PEHAR-gc	Lstm	2
GDP286A	2.2945	FA	Vecm	3
GDP287A	1.3926	GFSM-gc	Lstm	2
GDP288A	3.3092	Arima	Arima	0
HHSNTN	0.9409	Arima	Arima	0
HSBR	0.6570	PEHAR-gc	Lstm	1
HSFR	0.5360	FCBF	Vecm	2
HSMW	0.3843	Arima	Arima	0
HSNE	0.3480	Arima	Arima	0
HSSOU	0.5953	PEHAR-te	Vecm	1
HSWST	0.5711	Arima	Arima	0
IPS10	0.7663	PCA	Lstm	2
IPS11	0.9316	FA	Vecm	1

Table 3.7 – Évaluations par rapport à MASE des prédictions des données des États Unis (3).

Séries temporelles	Paramètres du meilleure prédiction			
	MASE	**Méthodes de réduction**	**Modèles de prédiction**	**Nombres de prédicteurs**
IPS12	0.7323	PEHAR-gc	Lstm	10
IPS13	0.9170	FCBF	Lstm	4
IPS18	1.0287	PEHAR-te	Lstm	4
IPS25	1.9803	FA	Vecm	3
IPS299	0.8891	FA	Vecm	1
IPS306	0.6865	Arima	Arima	0
IPS32	0.6887	Arima	Arima	0
IPS34	0.9098	FA	Vecm	3
IPS38	0.8050	FA	Lstm	5
IPS43	0.8747	PEHAR-gc	Vecm	1
LBLCPU	1.3706	GFSM-gc	Lstm	4
LBMNU	0.8304	PEHAR-te	Vecm	1
LBOUT	0.7204	PEHAR-gc	Vecm	2
LBPUR7	1.4192	FA	Lstm	2
LHEL	0.8020	Arima	Arima	0
LHELX	0.4422	Arima	Arima	0
LHEM	0.8056	KernelPCA	Lstm	1
LHNAG	0.8301	PEHAR-te	Vecm	2
LHU14	0.7114	PEHAR-te	Lstm	3
LHU15	0.9781	FA	Vecm	1
LHU26	0.8284	FA	Lstm	1
LHU27	0.9127	GFSM-te	Lstm	2
LHU5	0.6428	FCBF	Lstm	4
LHU680	0.7785	GFSM-gc	Lstm	5
LHUR	0.5249	PEHAR-te	Vecm	1
MOCMQ	0.6226	PCA	Vecm	5
MSONDQ	0.8929	FA	Vecm	1
MZMSL	3.3108	PEHAR-te	Vecm	10
PCEPILFE	1.4966	FA	Vecm	1
PMCP	0.8321	PEHAR-te	Lstm	1
PMDEL	0.3057	KernelPCA	Lstm	1
PMI	0.4040	PCA	Vecm	2
PMNO	0.5186	PEHAR-te	Lstm	1
PMNV	0.2928	KernelPCA	Lstm	2
PMP	0.4636	PCA	Lstm	2
PSCCOMR	0.4578	KernelPCA	Vecm	1
PW561R	3.7283	Arima	Arima	0
Sfygm6	0.4351	PEHAR-gc	Lstm	2
Sfygt1	0.5096	Arima	Arima	0
Sfygt10	0.6560	PEHAR-gc	Lstm	1
UTL11	0.3817	KernelPCA	Lstm	1
sFYAAAC	0.8806	PEHAR-gc	Lstm	3
sFYBAAC	0.7346	Arima	Arima	0

3.4.6.2 Résultats détaillées des données d'Australie

Nous montrons ici les résultats de chaque variable à part des données d'Australie, par rapport au mesures utilisées.

Table 3.8 – Évaluations par rapport à RMSE des prédictions des données d'Australie (1).

Séries temporelles	Paramètres de la meilleure prédiction			
	RMSE	**Méthodes de réduction**	**Modèles de prédiction**	**Nombres de prédicteurs**
10yrNSWT-bond	0.3785	PCA	Lstm	1
10yrT-bond	0.3918	PEHAR-gc	Lstm	13
5yrT-bond	0.4221	PEHAR-gc	Lstm	13
90daysBkbills	0.4220	PEHAR-gc	Lstm	11
AveComp	3.2555	Arima	Arima	0
BM	4.3590	Arima	Arima	0
BusInv	19.1427	Arima	Arima	0
COMMP	25.2060	FA	Vecm	2
CPI-ALL	2.9907	Arima	Arima	0
Cap-BioRes	38.2372	PEHAR-te	Vecm	1
Cap-IntPro	8.4156	GFSM-te	Lstm	13
Cap-MacEqu	19.9539	GFSM-gc	Lstm	1
Cons-Alc	5.4533	Arima	Arima	0
Cons-CigTob	9.6739	GFSM-gc	Lstm	10
Cons-CloFoo	9.9472	Arima	Arima	0
Cons-Com	5.8299	PEHAR-gc	Lstm	4
Cons-EGF	10.1184	FCBF	Vecm	1
Cons-Edu	4.5686	Arima	Arima	0
Cons-Final	2.1495	PEHAR-gc	Lstm	14
Cons-Food	3.9090	Arima	Arima	0
Cons-FurEqu	7.2541	GFSM-gc	Lstm	9
Cons-HCR	7.1895	Arima	Arima	0
Cons-Hea	9.4737	Arima	Arima	0
Cons-InsFin	5.9195	KernelPCA	Lstm	8
Cons-OpeVeh	6.3023	GFSM-gc	Lstm	6
Cons-Other	3.7091	GFSM-gc	Lstm	1
Cons-PurVeh	22.0620	Arima	Arima	0
Cons-RecCul	5.2994	PEHAR-gc	Lstm	4
Cons-RenDwe	0.3954	PCA	Lstm	15
Cons-TraSer	12.0438	Arima	Arima	0
Credit-Bus	6.2100	GFSM-gc	Lstm	12
Credit-Nar	2.5600	PEHAR-gc	Lstm	2
Credit-Per	10.1235	GFSM-te	Lstm	1
Credit-Total	2.4405	KernelPCA	Lstm	1
Currency	4.1217	Arima	Arima	0
DomSales	6.3797	GFSM-te	Lstm	1
EXrateIDR	52.3694	KernelPCA	Lstm	14
EXrateKRW	22.4462	PEHAR-gc	Lstm	8
Emp-FtPer	1.7592	PEHAR-gc	Lstm	2
Emp-HoursFt	2.9836	PEHAR-gc	Lstm	8
Emp-HoursPt	4.2395	PCA	Lstm	2

Table 3.9 – Évaluations par rapport à RMSE des prédictions des données d'Australie (2).

Séries temporelles	Paramètres de la meilleure prédiction			
	RMSE	**Méthodes de réduction**	**Modèles de prédiction**	**Nombres de prédicteurs**
Emp-JobVac	7.5567	GFSM-te	Vecm	1
Emp-PtPer	3.1822	PCA	Lstm	3
Emp-TotalPer	1.2674	PEHAR-gc	Lstm	1
Emp.HoursPer	2.7246	Arima	Arima	0
Exports	10.2160	GFSM-gc	Lstm	3
ExrateCNY	22.7113	PEHAR-gc	Lstm	1
ExrateGBP	15.6017	Arima	Arima	0
ExrateHKD	21.5192	GFSM-gc	Lstm	3
ExrateJPY	26.5681	Arima	Arima	0
ExrateMYR	18.6472	GFSM-gc	Lstm	3
ExrateNZD	11.1951	GFSM-gc	Lstm	2
ExrateSGD	17.1865	PEHAR-gc	Lstm	1
ExrateUSD	21.9062	Arima	Arima	0
Exrateavg	16.6358	Arima	Arima	0
FedFunds	0.3682	Arima	Arima	0
Gov	3.6372	FA	Lstm	1
Gov-Nat	7.2068	Arima	Arima	0
Gov-NatLoc	3.5380	FA	Lstm	1
HCE-AVCE	8.4179	Arima	Arima	0
HCE-DTA	10.9295	FCBF	Vecm	1
HCE-Fuel	26.0255	PCA	Vecm	1
HCE-ITA	12.4713	FA	Lstm	8
HCE-MV	5.0543	PEHAR-gc	Vecm	1
HCE-MaiRep	5.2216	GFSM-gc	Vecm	1
HCE-OthMot	4.3157	PEHAR-gc	Vecm	1
HCE-ParAcc	4.9910	FA	Lstm	2
HCE-Pet	9.9564	PEHAR-gc	Vecm	1
HCE-Post	6.7828	PEHAR-gc	Vecm	1
HCE-TelEqu	5.7096	Arima	Arima	0
HCE-UTF	5.3940	GFSM-gc	Vecm	1
HSexfinNum	22.1135	Arima	Arima	0
HSexfinVal	23.7569	Arima	Arima	0
HStartsACT	32.2451	Arima	Arima	0
HStartsNSW	21.6614	GFSM-gc	Vecm	1
HStartsNT	38.6941	Arima	Arima	0
HStartsQLD	28.5647	Arima	Arima	0
HStartsSA	21.4597	Arima	Arima	0
HStartsTAS	33.5064	Arima	Arima	0
HStartsTotal	20.2773	Arima	Arima	0
HStartsVIC	20.3476	Arima	Arima	0
HStartsWA	22.9792	PCA	Lstm	4

Table 3.10 – Évaluations par rapport à RMSE des prédictions des données d'Australie (3).

Séries temporelles	Paramètres de la meilleure prédiction			
	RMSE	**Méthodes de réduction**	**Modèles de prédiction**	**Nombres de prédicteurs**
Hstarts-PDA	31.5409	Arima	Arima	0
IP-EGWWS	6.6560	PEHAR-gc	Vecm	2
IP-FBT	11.1784	PEHAR-gc	Vecm	1
IP-ME	16.2014	GFSM-te	Lstm	2
IP-MP	17.6873	PEHAR-gc	Lstm	2
IP-Man	7.4847	PEHAR-gc	Lstm	3
IP-Min	9.9457	Arima	Arima	0
IP-NMM	25.0742	GFSM-gc	Vecm	1
IP-PCCR	18.4193	GFSM-gc	Vecm	1
IP-PRM	23.0443	GFSM-gc	Lstm	3
IP-TCO	22.3538	PEHAR-te	Lstm	6
IP-Total	5.0464	PCA	Lstm	2
IP-WPP	19.1196	GFSM-gc	Lstm	1
Imports	10.3986	PEHAR-gc	Lstm	9
InvLevel	4.8337	GFSM-gc	Lstm	9
InvtoSales	0.0157	Arima	Arima	0
LabForce	1.1266	FCBF	Lstm	3
LoaAdv	3.3679	Arima	Arima	0
M1	10.8296	Arima	Arima	0
M3	4.4772	Arima	Arima	0
Mbase	23.0123	Arima	Arima	0
PEXP	17.0815	PEHAR-gc	Lstm	4
PIMP	16.6785	GFSM-gc	Lstm	11
RGDP	2.2029	GFSM-gc	Lstm	10
RHDI	6.1697	FCBF	Lstm	13
ResInv	16.3870	GFSM-gc	Vecm	1
RetSales	4.0069	GFSM-gc	Lstm	12
SPASXAllOrds	25.3721	Arima	Arima	0
TotComp	2.9637	PEHAR-te	Lstm	3
Unum-LFT	14.5714	GFSM-gc	Vecm	9
Unum-LPT	20.2005	Arima	Arima	0
Unum-Total	14.1751	PCA	Lstm	3
Urate	0.2075	GFSM-gc	Lstm	13
Urate-LFT	0.2258	GFSM-gc	Lstm	3
UrateLabForce	31.2047	Arima	Arima	0

Table 3.11 – Évaluations par rapport à MASE des prédictions des données d'Australie (1).

Séries temporelles	Paramètres de la meilleure prédiction			
	MASE	**Méthodes de réduction**	**Modèles de prédiction**	**Nombres de prédicteurs**
10yrNSWT-bond	0.4568	PCA	Lstm	1
10yrT-bond	0.4044	GFSM-gc	Lstm	13
5yrT-bond	0.3812	PEHAR-gc	Lstm	9
90daysBkbills	0.3282	Arima	Arima	0
AveComp	0.5619	GFSM-te	Lstm	3
BM	0.6876	Arima	Arima	0
BusInv	0.5870	Arima	Arima	0
COMMP	0.7553	FA	Vecm	2
CPI-ALL	0.6378	Arima	Arima	0
Cap-BioRes	1.7590	PEHAR-te	Vecm	1
Cap-IntPro	0.3598	GFSM-te	Lstm	13
Cap-MacEqu	0.4493	GFSM-gc	Lstm	1
Cons-Alc	0.7205	Arima	Arima	0
Cons-CigTob	1.5034	PEHAR-te	Lstm	14
Cons-CloFoo	0.7522	Arima	Arima	0
Cons-Com	0.2815	FA	Lstm	2
Cons-EGF	0.6929	Arima	Arima	0
Cons-Edu	0.0937	Arima	Arima	0
Cons-Final	0.5098	KernelPCA	Lstm	15
Cons-Food	0.5582	Arima	Arima	0
Cons-FurEqu	0.5081	Arima	Arima	0
Cons-HCR	1.0404	Arima	Arima	0
Cons-Hea	0.5558	PEHAR-te	Lstm	1
Cons-InsFin	0.8489	PCA	Vecm	1
Cons-OpeVeh	0.7519	Arima	Arima	0
Cons-Other	0.4054	KernelPCA	Lstm	1
Cons-PurVeh	0.4510	Arima	Arima	0
Cons-RecCul	0.5948	Arima	Arima	0
Cons-RenDwe	0.7639	PEHAR-te	Vecm	1
Cons-TraSer	0.6902	Arima	Arima	0
Credit-Bus	0.4333	FCBF	Lstm	1
Credit-Nar	0.4805	PEHAR-gc	Lstm	2
Credit-Per	0.2303	PCA	Lstm	3
Credit-Total	0.4634	KernelPCA	Lstm	1
Currency	0.7553	Arima	Arima	0
DomSales	0.6456	GFSM-te	Lstm	1
EXrateIDR	1.2870	GFSM-gc	Lstm	1
EXrateKRW	0.7601	GFSM-te	Lstm	9
Emp-FtPer	0.7434	FCBF	Lstm	6
Emp-HoursFt	0.7907	Arima	Arima	0
Emp-HoursPt	0.8566	GFSM-te	Lstm	4

Table 3.12 – Évaluations par rapport à MASE des prédictions des données d'Australie (2).

Séries temporelles	Paramètres de la meilleure prédiction			
	MASE	**Méthodes de réduction**	**Modèles de prédiction**	**Nombres de prédicteurs**
Emp-JobVac	1.0012	PEHAR-gc	Lstm	12
Emp-PtPer	0.5548	PCA	Lstm	3
Emp-TotalPer	0.5910	PEHAR-gc	Lstm	2
Emp.HoursPer	0.7372	PEHAR-gc	Lstm	1
Exports	0.5247	Arima	Arima	0
ExrateCNY	0.6553	PEHAR-gc	Lstm	11
ExrateGBP	0.4571	Arima	Arima	0
ExrateHKD	0.7631	Arima	Arima	0
ExrateJPY	0.6143	Arima	Arima	0
ExrateMYR	0.5691	GFSM-gc	Lstm	3
ExrateNZD	0.5180	GFSM-te	Lstm	3
ExrateSGD	0.5465	Arima	Arima	0
ExrateUSD	0.7710	PEHAR-gc	Lstm	5
Exrateavg	0.5250	Arima	Arima	0
FedFunds	0.2620	Arima	Arima	0
Gov	0.2923	PEHAR-gc	Lstm	1
Gov-Nat	0.3393	Arima	Arima	0
Gov-NatLoc	0.2776	FA	Lstm	1
HCE-AVCE	0.8955	PEHAR-gc	Lstm	12
HCE-DTA	0.6137	FCBF	Vecm	1
HCE-Fuel	0.5180	PCA	Vecm	1
HCE-ITA	0.6185	Arima	Arima	0
HCE-MV	0.4406	PEHAR-gc	Vecm	1
HCE-MaiRep	0.9230	Arima	Arima	0
HCE-OthMot	0.2564	Arima	Arima	0
HCE-ParAcc	0.4449	Arima	Arima	0
HCE-Pet	0.5474	PEHAR-gc	Vecm	1
HCE-Post	0.5256	PEHAR-gc	Vecm	1
HCE-TelEqu	0.3776	Arima	Arima	0
HCE-UTF	0.5599	GFSM-gc	Vecm	1
HSexfinNum	0.5574	Arima	Arima	0
HSexfinVal	0.5437	Arima	Arima	0
HStartsACT	0.4060	Arima	Arima	0
HStartsNSW	0.3988	GFSM-gc	Vecm	1
HStartsNT	0.6547	Arima	Arima	0
HStartsQLD	0.5499	Arima	Arima	0
HStartsSA	0.4251	Arima	Arima	0
HStartsTAS	0.7680	Arima	Arima	0
HStartsTotal	0.5060	Arima	Arima	0
HStartsVIC	0.4275	Arima	Arima	0
HStartsWA	0.4490	PCA	Lstm	4

Table 3.13 – Évaluations par rapport à MASE des prédictions des données d'Australie (3).

Séries temporelles	Paramètres de la meilleure prédiction			
	MASE	Méthodes de réduction	Modèles de prédiction	Nombres de prédicteurs
Hstarts-PDA	1.2595	Arima	Arima	0
IP-EGWWS	0.7609	GFSM-gc	Lstm	1
IP-FBT	0.5589	PEHAR-gc	Vecm	1
IP-ME	0.6124	GFSM-gc	Lstm	1
IP-MP	1.0241	PEHAR-gc	Lstm	2
IP-Man	0.7783	PEHAR-gc	Lstm	13
IP-Min	0.4267	PEHAR-te	Lstm	2
IP-NMM	0.7225	GFSM-gc	Lstm	1
IP-PCCR	0.6806	PEHAR-te	Vecm	1
IP-PRM	0.6142	GFSM-gc	Lstm	3
IP-TCO	0.8886	PEHAR-gc	Lstm	8
IP-Total	0.5662	PCA	Lstm	2
IP-WPP	0.8346	GFSM-gc	Lstm	2
Imports	0.5945	PEHAR-gc	Lstm	9
InvLevel	0.9326	GFSM-gc	Lstm	9
InvtoSales	0.5446	Arima	Arima	0
LabForce	0.6135	FA	Lstm	1
LoaAdv	0.3146	Arima	Arima	0
M1	0.3696	Arima	Arima	0
M3	0.5272	Arima	Arima	0
Mbase	1.4963	Arima	Arima	0
PEXP	0.6199	Arima	Arima	0
PIMP	0.6013	Arima	Arima	0
RGDP	0.5569	FA	Lstm	1
RHDI	0.6394	Arima	Arima	0
ResInv	0.9074	GFSM-gc	Vecm	1
RetSales	0.6320	FA	Lstm	1
SPASXAllOrds	0.4624	Arima	Arima	0
TotComp	0.5075	PEHAR-gc	Lstm	15
Unum-LFT	1.0040	GFSM-gc	Vecm	5
Unum-LPT	0.5685	Arima	Arima	0
Unum-Total	0.9051	Arima	Arima	0
Urate	0.3802	GFSM-te	Lstm	9
Urate-LFT	0.4765	GFSM-gc	Lstm	3
UrateLabForce	0.5091	Arima	Arima	0

3.4.6.3 Résultats globales

Les meilleures prédictions obtenues des données des États-Unis de chaque variable selon la mesure RMSE sont montrés dans les Tables 3.2, 3.3, et 3.4. Et les résultats obtenues selon la mesure MASE sont montrés dans les Tables 3.5, 3.6, et 3.7.

Les meilleures prédictions obtenues des données d'Austraile de chaque variable selon la mesure RMSE sont montrés dans les Tables 3.8, 3.9, et 3.10. Et les résultats obtenues selon la mesure MASE sont montrés dans les Tables 3.11, 3.12, et 3.13.

À partir de ces résultats, nous comparons les méthodes utilisées en terme de classement ; on calcule pour chaque méthode le nombre de variables sur lesquels cette elle permet (avec l'un des modèles de prédiction utilisé) d'avoir les

meilleures prédictions. Ces comparaisons sont illustrées dans les Figures 3.6a, 3.6b, 3.7a et 3.7b.

La première colonne de chaque graphique montre le nombre de variables sur lesquelles chaque méthode (sur la ligne) conduit aux prédictions les plus précises. La deuxième colonne montre le nombre de variables où chaque méthode est la deuxième meilleure méthode, et ainsi de suite. On peut donc se concentrer sur la première colonne de chaque figure, car ce qui nous intéresse c'est le nombre de variables où chaque méthode donne les meilleurs résultats. Nous discutons ces résultats dans 3.4.7.

3.4.6.4 Résultats par modèles de prédiction

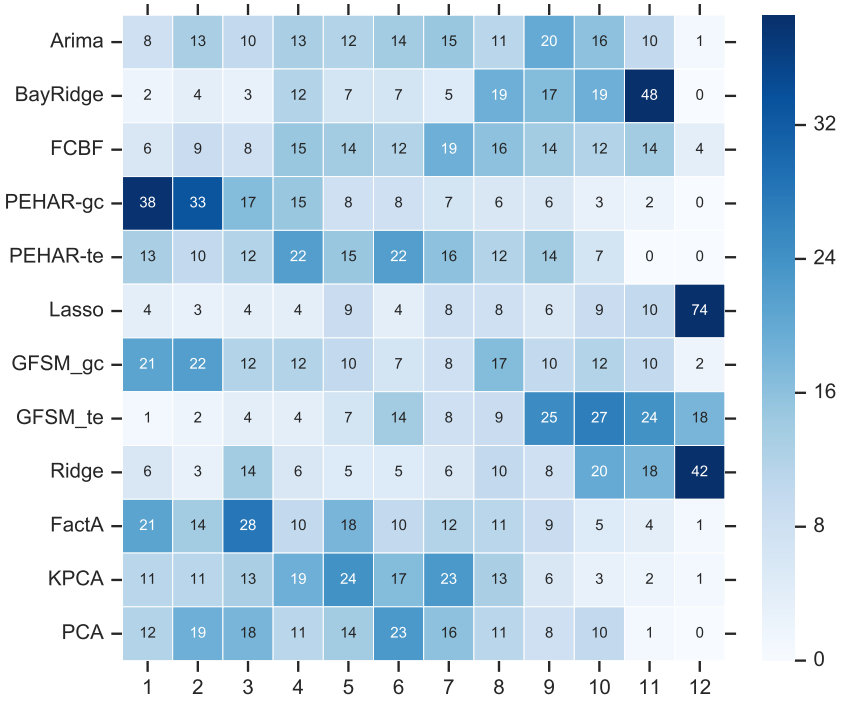
Nous montrons dans la Table 3.14 l'applicabilité des modèles de prédiction, c.à.d., le nombre de variables qui sont mieux prédites par chaque modèle. Soulignons que les méthodes de régularisation sont couplées avec le modèle de prédiction VAR, c'est à dire que le régularisation se fait sur les coefficients du modèles VAR. Pour ne pas confondre avec le modèle classique VAR, nous avons mis juste le nom de la méthode de régularisation.

Données	Modèles	Nombre d'applications	
		Rmse	Mase
US	Arima	8	17
	Var+BayesianRidge	2	2
	Var+Lasso	4	3
	Var+Ridge	6	5
	Vecm	50	40
	Lstm	73	76
Australie	Arima	39	49
	Var+BayesianRidge	0	1
	Var+Lasso	3	3
	Vecm	19	14
	Lstm	56	50

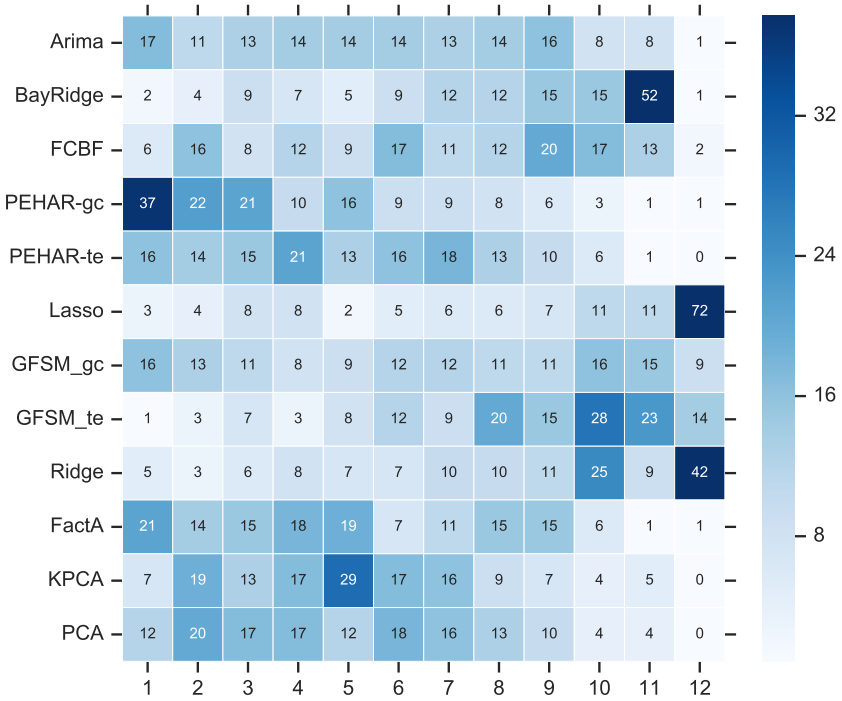
Table 3.14 – Applicabilité des modèles de prédiction.

3.4.7 Discussion

Au niveau des modèles de prédiction, chaque modèle de prédiction surpasse les autres pour un sous-ensemble de variables (Table 3.14). Mais globalement, le modèle LSTM est le plus sélectionné, et les méthodes de régularisation (Lasso

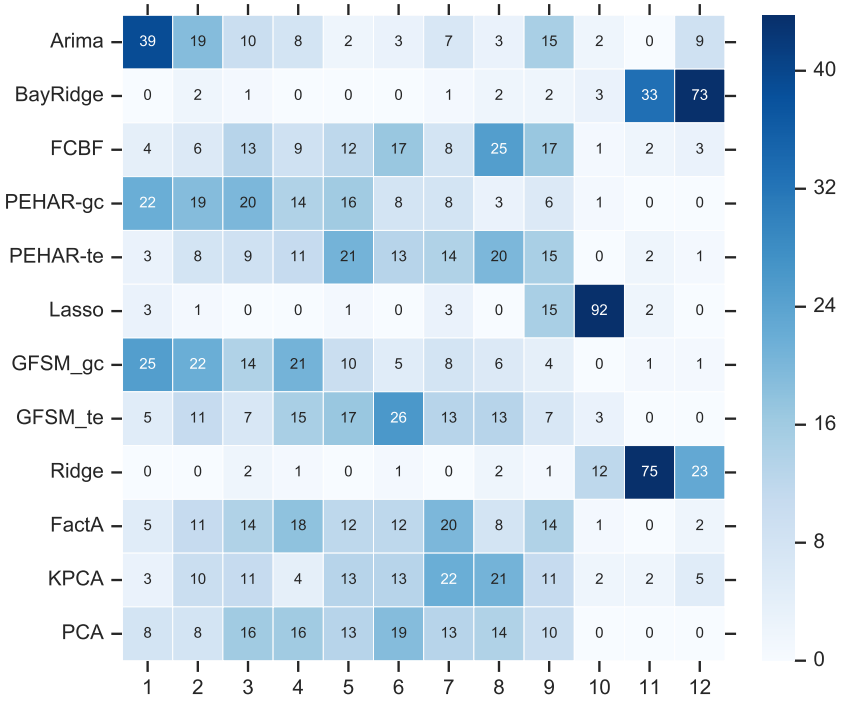


(a) Comparaison en terme de RMSE

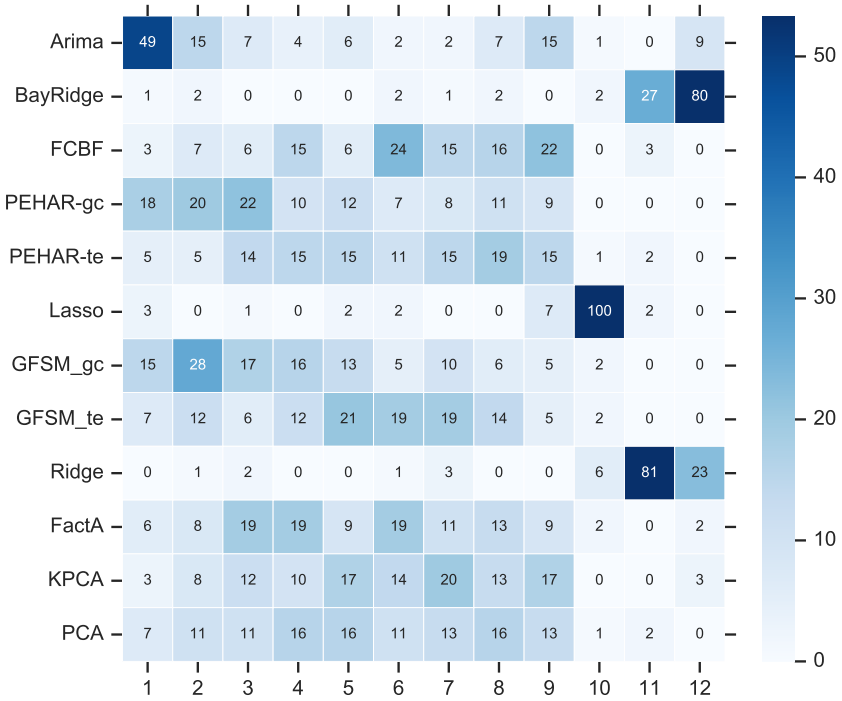


(b) Comparaison en terme de MASE

Figure 3.6 – Comparaison de l'applicabilité des méthodes sur les données des États Unis.



(a) Results based on RMSE



(b) Results based on MASE

Figure 3.7 – Comparaison de l'applicabilité des méthodes sur les données d'Australie.

et Ridge) sont les moins utilisées. Ceci est probablement dû au fait que LSTM exploite davantage les dépendances non-linéaires entre les prédicteurs et les cibles que le modèle VECM. Cependant, le modèle ARIMA fournit également de bons résultats, surtout pour les données d'Australie, où il permet de mieux prédire 52 variables selon MASE. On peut expliquer ce dernier résultat par deux explications :

1. les variables qui sont mieux prédites par ARIMA sont considérées comme indépendantes, ou, ne nécessitent pas d'informations externes provenant d'autres variables
2. les VECM et LSTM n'exploitent pas bien les dépendances si elles existent entre ces variables et les autres..

À notre avis, la première explication est la plus logique, car il est tout à fait normale d'avoir des variables indépendantes, ceux qui se trouvent au sommet du graphe de causalité.

Au niveau des méthodes de réduction, les résultats obtenues montrent que chaque méthode est plus appropriée à un ensemble de variables. Mais par rapport au nombre de variables, la distribution de l'applicabilité des méthodes sur les variables n'est pas uniforme ; les résultats montrent que la méthode PEHAR permet de mieux prédire plus de variables pour les données des États unis, et la méthode GFSM pour les données d'Australie.

Contrairement aux résultats des deux travaux STOCK et M. W. WATSON 2012 et JIANG, ATHANASOPOULOS, HYNDMAN et al. 2017, qui ont été effectuées sur les mêmes jeux de données, nous avons trouvé que les méthodes de régularisation ne fournissent pas de bonnes prédictions. Par exemple, pour les donnée d'Australie, la méthode Lasso est la seule sélectionnée et aide de mieux prédire juste 3 variables, et pour la données des états unies, c'est la méthode Ridge avec 6 variables. A notre avis, l'explication de ce phénomène revient au principe de ces méthodes. Elles permettent d'éliminer l'impact des variables non importants, mais ces variables parfois restent dans le modèle. Par conséquent, la présence des ces variables dans le modèles (avec des coefficients faibles) ne résout pas radicalement le problème de prédiction avec beaucoup de variables.

Soulignons une différence majeure entre les méthodes de réduction utilisées. Les méthodes de réduction la dimension telles que l'ACP et l'analyse factorielle appartiennent à la catégorie d'apprentissage non-supervisé, car elles réduisent la dimension des variables simplement en fonction des caractéristiques statistiques des prédicteurs, sans tenir compte de la variable cible. En général, l'historique des données contient des valeurs passées de variables cibles. Dans ce cas, il est utile de l'exploiter. L'idée est simplement de tirer parti de ces informations. Ainsi, les deux algorithmes proposés (GFSM et PEHAR) appartiennent à la catégorie des méthodes d'apprentissage supervisé, et ses avantages consistent à l'exploitation des dépendances entre les prédicteurs et les variables cibles.

Concernant le nombre de variables conduisant aux meilleures prédictions, la

moyenne du nombre de variables sélectionnées est proche de 6. Ce qui signifie que dans la plupart des cas, les meilleurs résultats sont obtenus en utilisant un nombre relativement petit de variables. Cette réduction et l'un des résultats importants obtenues, car elle est très utile en termes de temps de calcul, surtout si nous utilisons des réseaux de neurones comme modèles de prédiction.

Un problème que nous avons rencontré consiste à comment améliorer les méthodes de réduction pour pouvoir générer automatiquement un nombre de variables, c'est à dire, sans le fixer à l'avance, car dans les expérimentations, nous avons tester plusieurs valeurs pour chaque méthode (*cf.* 3.4.4) . L'utilisation des méthode de réduction qui trouvent le nombre de prédicteurs ne permet pas pas nécessairement de trouver l'ensemble optimal de variables en terme de précision de prédiction, mais cela est parfois utile dans la pratique. Dans discutons maintenant comment étendre les algorithmes proposés pour pouvoir générer automatiquement un nombre de variables sans qu'il soit une entrée de l'algorithme. Pour la méthode PEAHR, une idée consiste à fixer un seuil sur le vecteur des *Hubs*, en utilisant un test statistique. Cette méthode est pratique si nous utilisons la causalité de Granger comme mesure de pertinence, puisqu'elle est naturellement calculée en utilisant un test statistique. Cependant, pour Transfer Entropy, il n'est pas clair comment effectuer un test de signification pour les valeurs obtenues.

Pour la méthode GFSM, le nombre de variable à sélectionner est égal au groupes de *clusters* générés par la méthode de *clustering* utilisée dans l'algorithme. Par conséquent, cet algorithme, peut être facilement étendu pour fournir un nombre automatique de variables en utilisant certaines méthodes de sélection du nombre optimal de groupes de la méthode PAM (Partitionning Arround Medoid) par example KAUFMAN et ROUSSEEUW 2009, TIBSHIRANI, WALTHER et HASTIE 2001.

3.5 Concours de Prédiction de l'Énergie Photovoltaïque Multi-Plantes

Dans cette section, nous présentons nos travaux effectués lors d'un concours internationale de prédiction, dans lequel nous avons eu la quatrième place parmi 11 équipes au niveau internationale. Il s'agit de la prédiction de l'énergie photovoltaïque multi-plantes.

3.5.1 Introduction

La prédiction de la production d'énergie photovoltaïque (PV) a attiré beaucoup d'attention récemment avec l'intérêt croissant pour l'utilisation du PV comme source d'énergie renouvelable.

En général, l'énergie PV peut être mesurée en tant que série temporelle qui change en fonction de l'état du système et des conditions externes. La prédic-

tion de la production de tels systèmes a un impact direct sur l'optimisation et le contrôle de l'énergie utilisée. Il vise à prédire la production d'énergie sur la base des données captées sur des panneaux solaires (puissance générée, température, humidité) et des capteurs de station météo (température, couverture nuageuse, humidité, ...). Ce problème est très similaire à celui du problème de prédiction de séries temporelles multiples. Avec la présence de multiples variables, elles ne contribueront pas de la même façon à la prédiction de l'énergie produite.

Dans cette partie, nous présentons un processus qui englobe plusieurs méthodes de réduction et modèles de prédiction. L'idée est de trouver la meilleure méthode et modèle qui permettent de prédire l'énergie PV à une heur donnée de la journée. Nous utilisons diverses approches de sélection basées sur la causalité et des techniques de réduction de dimension.

L'organisation de cette partie est la suivante. Nous présentons et discutons quelques travaux liés au problème abordé. Ensuite nous détaillons notre proposition. Nous décrivons également le processus de prédiction et la méthodologie utilisés pour effectuer les expériences. Finalement, nous montrons et discutons les résultats, et résumons notre approche.

3.5.2 Travaux antérieurs

Dans la littérature, de nombreuses approches ont été proposées pour traiter le problème de la prédiction de la production d'énergie photovoltaïque. Dans CECI, CORIZZO, FUMAROLA et al. 2017, une comparaison entre les RNAs, les arbres de régression et les méthodes basées sur l'auto-corrélation spatio-temporelle a été expérimentée. Les auteurs montrent que les arbres de régression fournissent de meilleurs résultats que les RNAs.

Dans DUMITRU, GLIGOR et ENACHESCU 2016, les RNAs sont aussi utilisés pour prédire la production d'énergie PV, profitant de leurs capacité à apprendre les changements dans l'historique des données, en intégrant plusieurs facteurs (internes et externes) qui influencent la production d'énergie. La même problématique a été étudiée dans GANDELLI, GRIMACCIA, LEVA et al. 2014. Une approche hybride a été utilisée, en ajoutant les contraintes physiques de l'installation des PV à l'entrée d'un RNA. Les résultats montrent une amélioration de la précision de la prédiction par rapport au modèle sans ces contraintes.

Le problème de la prévision de l'énergie PV peut être modélisé comme une prédiction de séries temporelles multi-variées. Même si ce problème apparait similaire aux approches discutés dans ce manuscrit, mais il y une différence basique dans la structure des données utilisées, de qui influence aussi la structure des modèles de prédiction à utiliser. Par conséquent, nous n'allons essayer d'adresser ce problème indépendamment de ce que nous avons déjà proposé dans ce manuscrit, surtout dans la phase de prédiction.

Dans la suite, nous reformulons ce problème et discutons des principales approches utilisées. Considérons un ensemble de séries temporelles $X = [X_1, \ldots, X_k]$

et une variable cible Y, avec n observations. Il existe plusieurs stratégies pour prédire Y en utilisant X. Une façon consiste à utiliser des modèles qui exploitent les valeurs précédentes de Y et X, par exemple les modèles vectoriels auto-régressifs JOHANSEN 1991. Dans ce travail, nous nous concentrons sur les modèles de prédiction qui prédisent Y à l'instant t basé sur les valeurs des variables de X en même temps t, car pour ce problème, les informations des variables prédictives à l'instant t sont disponibles. Par conséquent, le modèle général peut être exprimé comme suit,

$$Y(t) = f(X_1(t) + \cdots + X_k(t)) + E(t), \tag{3.9}$$

où E est un terme d'erreur.

Les modèles linéaires supposent que Y s'exprime comme une combinaison linéaire des variables de X L'estimation de leurs paramètres peut être réalisée par différentes méthodes. Le plus commun est la technique du moindre carré, qui consiste à minimiser la somme des erreurs au carré, et la résolution est effectuée sur la base d'une dérivation simple. Les méthodes de régularisation visent à minimiser l'impact des variables non pertinentes en le poussant vers zéro. Cette technique est pratique lorsque le nombre de prédicteurs est grand, ou lorsque la résolution classique n'est pas possible en raison des contraintes algébriques. Les modèles linéaires sont généralement basés sur des RNAs ou les arbres de décision, ce qui est intéressant pour ce type de problématiques.

3.5.3 Méthodologie

Pour cette compétition, nous avons proposé une nouvelle méthode de sélection de variables. Considérons une variable cible Y et un ensemble de prédicteurs P. L'objectif est d'extraire les variables pertinentes de P. Premièrement, nous calculons le graphe des causalités, puis nous le réduisons en éliminant les dépendances à l'aide de la technique de réduction transitive. Enfin, nous les classons en fonction de la causalité sur la variable cible.

Nous résumons dans l'algorithme 4 cette méthode, où nous supposons que le graphe de causalité est une entrée de l'algorithme, et nous utilisons les notations suivantes : $X \to Y$ exprime le fait que X cause Y, et *caus*$(X \to Y)$ est la valeur de cette causalité.

Algorithm 4 *Transitive Reduction on Causality Graph* (TRCG)

Input : G : le graphe de causalités, Y : la variable cible,k : la taille de réduction
Output : S : l'ensemble des variables prédictives.

/* Réduction du graphe par élimination des dépendances */
for all node $ts_1 \in G.nodes \setminus \{Y\}$ **do**
  for all node ts_2 in $G.nodes \setminus \{ts_1, Y\}$ **do**
    if $ts_1 \rightarrow ts_2$, $ts_2 \rightarrow Y$ and $ts_1 \rightarrow Y$ **then**
      Supprimer l'arrêt entre ts_1 et Y
    end if
  end for
end for/* Sélection des top k variables qui causent Y */
P = {ts ∈ G.nodes, ts → Y}
P_s = P.sort (key=lambda x : $caus(x \rightarrow Y)$)
S = top_k (P_s)
return S

3.5.4 Expérimentations

Les ensembles de données expérimentés sont des séries horaires multiples (de l'heure 2 à l'heure20, chaque jour), représentant des données de 3 plantes, s'étendant sur une période de 12 mois (année 2012). L'objectif est de prédire 3 mois de la variable de production, de janvier à mars 2013 (où les valeurs des variables cibles ne sont pas connues), en fonction de facteurs internes (température et irradiance), et facteurs externes (couverture nuageuse, point de rosée, humidité, pression, température, portance, vitesse du vent). Les données sont organisées de manière à prévoir chaque heure séparément, c'est-à-dire, pour chaque plante, nous avons des 19 variables cibles à à prévoir. Nous avons adopté cette manière de structurer les données car nous pensons que la production d'énergie dans chaque heure est différente de l'autre, par conséquent chaque heure nécessite un modèle de prédiction spécifique. C'est possible aussi d'utiliser l'approche classique, et considérer le production d'énérgie dans toute la journée comme une seule variable, dans ce cas, c'est plus simple

La méthodologie adoptée est basée sur la sélection du meilleure modèle pour prédire la production d'énergie à chaque chaque heure de 2 du matin à 20 soir. Tout d'abord, une expérience de référence est effectuée sur les données d'apprentissage (année 2012) en utilisant la validation croisée avec 8 expériences en prévoyant 3 mois dans chaque expérience. Nous exécutons tous les modèles sur les sous-ensembles générés par toutes les méthodes de réduction. Ensuite, nous sélectionnons pour chaque variable cible une paire {méthode, modèle } qui sera utilisée dans l'étape de test.

Dans l'étape de réduction, nous utilisons deux méthodes existantes, la méthode proposée dans P. PRZYMUS, Y. HMAMOUCHE, A. CASALI et al. 2017, qui

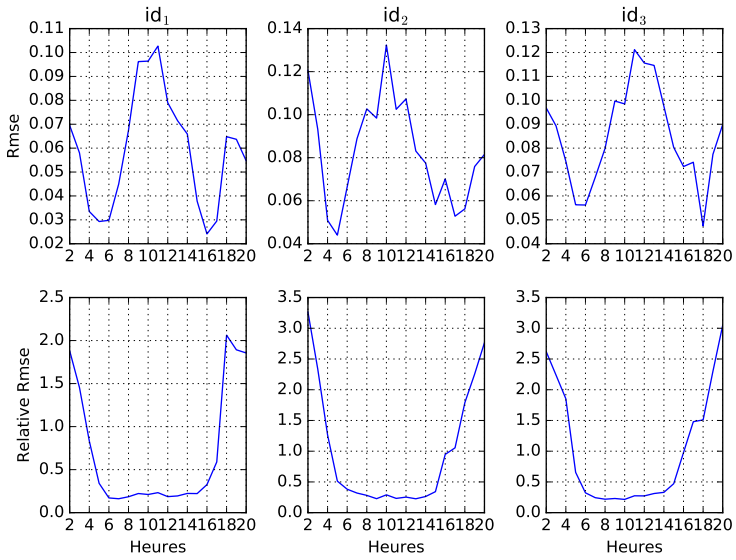


Figure 3.8 – Forecast accuracy analysis using RMSE

est basée sur l'algorithme de marche aléatoire dans les graphes de causalités, avec une variante GRWR en utilisant le graphe de causalité de Granger, et TRWR en utilisant le graphe de l'Entropie de Transfert. Nous utilisons également la méthode ACP, et deux versions de l'algorithme GFSM (4), qu'on le nomme ici TTRCG et GTRCG, en utilisant respectivement l'Entropie de Transfert et la causalité de Granger.

Concernant les modèles de prédictions, nous utilisons 4 types de modèles :

— Modèles de régression multiple : régression linéaire, RANSAC Regressor (RR), Orthogonal Matching Pursuit (OMP), Theil Sen Regressor (TSR), Hiber Regressor (HB).
— Régression avec des méthode de régularisation : Ridge, Bayesian Ridge, SVM, Lasso.
— Arbre de régression : Decision Tree Regressor (DTR), Gradient Boosting Regressor (GBR)
— RNAs : un réseau multicouche (MLP) avec une couche cachée et un algorithme de descente de gradient stochastique pour mettre à jour les paramètres du réseau.

3.5.5 Résultats et discussion

Dans cette section, nous présentons les résultats obtenus, et nous fournissons une discussion. Comme nous l'avons décrit dans la section précédente, nous avons utilisé 3 méthode (PCA, RWR et TRCG) dans l'étape de réduction. Nous avons obtenu une erreur moyenne quadratique RMSE$= 0,177$ pour 10% de données de test et $0,253$ pour toutes les données de test.

Globalement, les approches basées sur l'analyse des graphes de causalités ; RWR P. PRZYMUS, Y. HMAMOUCHE, A. CASALI et al. 2017 et l'algorithme proposé, sont les plus sélectionnés. Mais en général, il n'y a pas de modèle qui obtient de meilleurs résultats dans tous les cas. Les résultats détaillées de la prédiction de chaque plante se trouvent dans Y. HMAMOUCHE 2018.

La Figure 3.8 montre les RMSE et les RMSE relatifs (RMSE divisé par la moyenne des données) des prédiction de l'énergie des 3 plantes. Nous remarquons qu'il existe quelques heures au début et à la fin de la journée (de 2 à 5 et de 18 à 20) lorsque la production d'énergie est faible et qu'il est très difficile à fournir de bonnes performances de prédiction. Malheureusement, cela diminue la précision des prédictions globales. Par conséquent, il fallait peut être de se concentrer plus sur la prédiction pendant ces heures pour avoir des meilleures prédictions. En terminant, les modèles de prédiction sélectionnés pour ces heures sont les modèles MLP et Gradient Boosting. Ce qui signifie que lorsque la production d'énergie est faible, les données sont susceptibles d'avoir des valeurs aberrantes et des changements non-linéaires. Dans ce cas, les RNAs sont plus précis que les autres modèles utilisés.

3.6 Une bibliothèque en R pour l'analyse non linéaire des séries temporelles

Dans cet section, nous décrivons la bibliothèque **NlinTS** du logiciel R que nous avons développer pour l'analyse non linéaire des séries temporelles, dans lequel nous avons implémenté des modèles non-linéaires utilisant des réseaux neurones artificiels. Nous proposons également une implémentation d'un test de causalité de Granger non-linéaire, basé essentiellement sur deux modèles de prédiction. Cette version peut être une alternative au test standard, qui dans certains cas présente quelques limites. Des expérimentations sur la prédiction des séries temporelles multi-variées sont effectuées sur de nombreux jeux de données. Les résultats montrent que les prédictions de nombreux ensembles de données sont améliorées à l'aide des modèles fournis.

3.6.1 Les modèles implémentés

Son code source de cette librarire et le manuel de fonctionalités peuvent être trouvés dans Youssef HMAMOUCHE 2017. Nous décrivons dans cette section les principaux modèles implémentés. Le premier modèle est un VAR non linéaire. Puis nous avons utilisé ce modèle pour développer aussi une version non linéaire de la causalité de Granger. Nous détaillons dans ce qui suit ce modèle de prédiction. Appelons-le VARMLP (Multi-Layer Perceptron VAR). Le réseau de neurones MLP utilisé est basé sur une version modifiée d'une bibliothèque C ++ BARTHELEMY 2017.

3.6.1.1 Le modèle VARMLP

Dans cette partie, nous détaillons le modèle implémenté VARMLP . Considérons un ensemble de données d'apprentissage composé d'une série temporelle multi-variée $\{Y, Y_1, \ldots, Y_k\}$, qui contient une variable cible Y et k variables prédictives $\{Y_1, \ldots, Y_k\}$. Nous considérons une structure basée sur un perceptron multi-couche avec des neurones de polarisation avec une sortie, c'est-à-dire un neurone dans la dernière couche. Nous avons fait ce choix pour permettre la prédiction de chaque variable cible avec une série de prédicteurs spécifiques. Parce que toutes les variables cibles n'ont pas nécessairement les mêmes prédicteurs. Le modèle VARMLP (p) est un modèle de réseau neuronal qui prend en compte les valeurs antérieures de p de toutes les variables pour prédire Y. D'abord, il réorganise les données sous la forme d'une tâche d'apprentissage supervisée, puis détermine les coefficients du modèle. La fonction globale de VARMLP (p) peut s'écrire comme suit :

$$Y(t) = \Psi_{nn}(Y(t-1), \ldots, Y(t-p), \ldots, Y_k(t-1), \ldots, Y_k(t-p)) + E(t), \quad (3.10)$$

où Ψ_{NN} est la fonction réseau et E sont les termes d'erreur.

Par exemple, pour $k = 1$ (2 variables) et $p = 1$, l'équation 3.10 s'écrit comme suit :

$$Y(t) = \Psi_{nn}(Y(t-1), Y_1(t-1), Y_2(t-1)) + E(t). \quad (3.11)$$

L'algorithme implémenté dans notre bibliothèque pour l'entrainement du réseau est l'algorithme de gradient descend stochastique.

3.6.1.2 Le modèle de la causalité non-linéaire

L'une des principales caractéristiques des RNA est la non-linéarité et le mécanisme d'apprentissage. Cela peut être très important lors du calcul des causalités entre les variables, en particulier lorsqu'elles changent de façon non linéaire avec le temps. Nous proposons une implémentation d'une version étendue de la causalité de Granger en utilisant le modèle VARMLP. La structure adoptée est illustrée sur la figure 3.9.

Considérons deux séries chronologiques X et Y, et nous voulons tester si X provoque Y. Basé sur le principe général de la causalité de Granger, qui est une causalité prédictive, une série temporelle en provoque une autre si elle contient des informations utiles qui peuvent être exploitées pour avoir de meilleures prédictions GRANGER 1980. Ainsi, dans notre cas, deux modèles de prédiction (réseaux de neurones) sont construits, le premier ne considère que la série temporelle cible, et le second prend en compte à la fois les valeurs précédentes de la cible et les séries temporelles prédictives.

$$\text{Model}_1 : \quad Y(t) = \Psi_{1nn}(Y(t-1), \ldots, Y(t-p)) + E(t), \quad (3.12)$$

$$\text{Model}_2 : \quad Y(t) = \Psi_{2nn}(Y(t-1), \ldots, Y(t-p), X(t-1), \ldots, X(t-p)) + E(t), \quad (3.13)$$

où Ψ_{1nn} et Ψ_{2nn} sont les fonctions réseau des modèles 1 et 2, respectivement. Ensuite, la différence entre ces deux modèles est évaluée en comparant la somme résiduelle des carrés des deux modèles. Enfin, le test est effectué à un niveau α pour examiner l'hypothèse nulle (l'hypothèse que x ne cause pas y) à l'aide du test de Fisher. La différence avec le test classique de causalité, c'est que dans ce cas, les deux modèles VARNN contiennent plusieurs paramètres (coefficients des neurones) qu'ils faut considérer lors du calcul des degrés de liberté du test de Fisher. Le nombre de ces coefficients dépendent de la structure choisie des

deux modèles, c-à-d., le nombre de couches et le nombre de neurones de chaque couche.

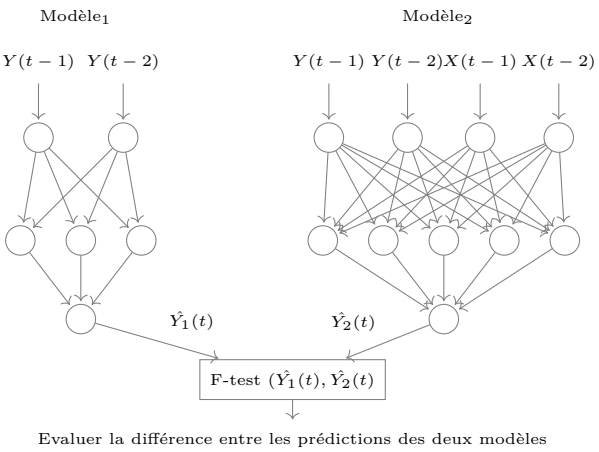


Figure 3.9 – Illustration du modèle proposé pour la causalité de Granger non-linéaire.

3.6.2 Exemples

Le modèle VARMLP prend en entrée une *Dataframe* numérique, où chaque colonne représente une variable, et la première colonne est la variable cible (la Dataframe peut contenir une colonne, dans ce cas, le modèle sera uni-varié). Le deuxième argument est le paramètre de retard, puis un vecteur numérique représentant la taille des couches cachées du réseau, le nombre d'itérations, et enfin, une valeur logique pour l'utilisation du neurone de polarisation dans le réseau. Nous représentons ici un exemple d'utilisation du modèle VARMLP :

```
1
2  library (timeSeries)
3  library (NlinTS)
4
5  # Charger les donnees
6  data = LPP2005REC
7  train_data = head (data, nrow (data) - 1)
8  test_data = tail (data, 1)
9
10 # Construire le modele avec deux couche cachees
11 model = varmlp (train_data, 8, c(16), 500, TRUE)
12
13 # Calculer les predictions
14 predictions = model$forecast (train_data)
15
16 # Afficher les prediction
17 print (tail (predictions,1))
18
```

```
# Mise a jour du modele (apprentissage a partir des donnees de test)
model$train (test_data)
```

Le modèle de causalité prend en entrée deux vecteurs numériques (le but étant de tester si le second provoque le premier), le paramètre de retard ; une valeur entière, deux vecteurs numériques représentant la taille des couches cachées utilisées dans les modèles 1 et 2, respectivement, le nombre des itérations ; former les réseaux, et enfin une valeur booléenne pour l'utilisation du neurone de polarisation dans les réseaux (avec vrai comme valeur par défaut). Un exemple d'utilisation du test de causalité non linéaire est le suivant :

```
library (timeSeries)
library (NlinTS)

# Charger les donnees
data = LPP2005REC

# Calculer la causalite non lineaire
model = nlin_causality.test (data[,1], data[,2], 2, c(2), c(4), 500,
    TRUE)
model $ summary ()
```

3.6.3 Expérimentations

Dans cette partie, nous montrons la procédure utilisée et les résultats obtenus des expériences de prévision de séries temporelles multi-variées.

Les jeux de données utilisés sont des séries temporelles multi-variées financières extraites de la bibliothèque **FinTS** du logiciel R. Nous évaluons deux modèles linéaires, les modèles VEC et ARIMA, et deux modèles de réseaux neuronaux artificiels, les modèles VARMLP et ARMLP (version uni-variée). Pour les modèles uni-variés (ARIMA et ARMLP), chaque variable de chaque ensemble de données est prédite séparément. Pour les modèles multi-variés, chaque variable cible est prédite en utilisant elle-même et d'autres variables appartenant au même jeu de données comme prédicteurs. La structure des ANNs utilisée est basée sur deux couches cachées, et nous utilisons une formule de *Thumb* : $2/3 \times (n+1)$, pour déterminer le nombre de neurones dans chaque couche, où n est le nombre d'entrées .

3.6.3.1 La procédure de prédiction

Les prédictions calculées sont associées aux dernières 20% observations des données. La procédure de prédiction adoptée est basée sur une stratégie de fenêtre glissante (on calcule une prédiction puis on avance d'un pas). Le choix de

l'ordre p des modèles de prédiction est une problématique. Dans notre cas, pour limiter le temps de calcul, nous fixons l'ordre de décalage maximal à 10, puis nous sélectionnons l'ordre approprié pour chaque série en utilisant le critère AIC.

Les modèles de prédiction utilisés sont les modèles ARIMA et VEC (de la bibliothèque **forecast**), et le modèle VARMLP (ARMLP pour prédiction uni-variée) utilisant le notre bibliothèque. Notons que pour les séries stationnaires, le modèle ARIMA est réduit au modèle ARMA. Pour les modèles VEC et ARIMA, les paramètres sont recalculés après chaque prédiction, alors que pour VARMLP et ARMLP, ils sont construits une fois, puis leurs paramètres sont mis à jour à partir des dernières valeurs réelles après chaque prédiction.

Enfin, nous évaluons la prédiction en utilisant deux mesures de précision : l'erreur quadratique moyenne (RMSE) et l'erreur moyenne absolue (MASE).

3.6.3.2 Préparation des données

Premièrement, les données sont normalisées entre 0 et 1 en utilisant la technique min-max. Deuxièmement, nous testons la stationnarité des séries temporelles en utilisant le test Dickey Fuller augmenté David A. DICKEY et FULLER 1979. Nous montrons dans Table 3.15 les ordres d'intégration de toutes les variables, regroupées par jeux de données. Cette analyse montre que l'ordre d'intégration maximum est 1, c'est-à-dire que chaque variable est stationnaire ou nécessite qu'une transformation de différenciation soit stationnaire.

Données	L'ordre d'intégration des variables													
d.ibmvwewsp6203	0	0	0	0										
m.ibmvwewsp2603	0	0	0	0										
m.fama.bond5203	1	0	0	0										
m.ibm3dx2603	0	0	0	0										
m.decile1510	0	0	0											
w.gs1n36299	1	1												
m.ibmspln	0	0												
ibm	1	0	1	1	1									
m.bnd	0	0	0	0	0									
m.gs1n3.5301	1	1												
w.tb3n6ms	1	1												
m.fac9003	0	0	0	0	0	0	0	0	0	0	0	0	0	0
m.cpice16.dp7503	0	0												
m.barra.9003	0	0	0	0	0	0	0	0	0	0				
m.5cln	0	0	0	0	0									
d.hkja	0	0												
d.spcscointc	0	0	0											
aa.3rv	0	0	0											

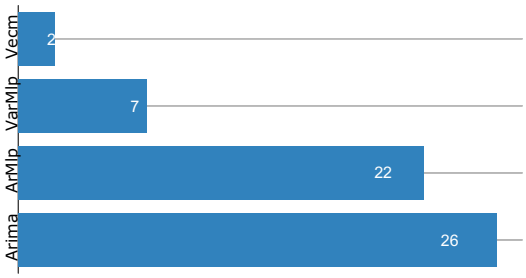
Table 3.15 – Analyse de stationnarité des données utilisés.

Données	Séries	RMSE				MASE			
		Arima	Vecm	ArMlp	VarMlp	Arima	Vecm	ArMlp	VarMlp
aa.3rv	X5m	0.025018	0.042553	0.025251	0.025213	0.608524	1.159667	0.617039	0.613195
aa.3rv	X10m	0.133399	0.153748	0.133923	0.134038	0.843168	1.143277	0.885151	0.854824
aa.3rv	X20m	0.125987	0.132101	0.124878	0.125137	0.865872	0.991161	0.826906	0.873854
d.hkja	HongKong	0.121422	0.120516	0.115683	0.119142	1.956501	1.969712	1.892793	1.934809
d.hkja	Japan	0.140389	0.164397	0.141187	0.143571	1.133964	1.302253	1.145903	1.164159
d.ibmvwewsp6203	IBM	0.065166	0.068115	0.065756	0.065599	1.138429	1.206342	1.14817	1.146705
d.ibmvwewsp6203	VW	0.047665	0.050772	0.047838	0.047914	1.259212	1.337042	1.26348	1.269427
d.ibmvwewsp6203	EW	0.051035	0.055352	0.051101	0.05124	1.226809	1.350394	1.235683	1.242456
d.ibmvwewsp6203	SP	0.044176	0.044962	0.042676	0.042822	1.211486	1.230671	1.165819	1.174269
d.spcscointc	SP500	0.10229	0.11953	0.10311	0.103172	1.199172	1.437745	1.197358	1.197946
d.spcscointc	Cisco	0.075734	0.082183	0.075493	0.076873	0.725141	0.80097	0.727496	0.737413
d.spcscointc	Intel	0.101261	0.118588	0.102315	0.102647	0.893181	1.058649	0.904561	0.906317
ibm	price	0.00298	0.002977	0.003732	0.00382	0.987232	0.998113	1.338241	1.372234
m.5cln	IBM	0.188075	0.247094	0.184932	0.194502	0.840763	1.074577	0.82014	0.828603
m.5cln	HPQ	0.261131	0.339563	0.258696	0.259071	0.995182	1.310625	1.004035	1.007419
m.5cln	INTC	0.210038	0.273591	0.208287	0.227213	0.826199	1.047992	0.819392	0.907168
m.5cln	MER	0.258418	0.316216	0.246272	0.250106	1.318579	1.447121	1.231447	1.210081
m.5cln	MWD	0.200097	0.263525	0.199238	0.206634	1.106883	1.462892	1.122782	1.219215
m.barra.9003	AGE	0.12682	0.166658	0.125925	0.133388	0.601784	0.780115	0.60111	0.633704
m.barra.9003	C	0.14523	0.197488	0.141562	0.15754	0.614233	0.782455	0.605188	0.690783
m.barra.9003	MWD	0.121005	0.181067	0.120944	0.122728	0.680515	0.996834	0.683806	0.696084
m.barra.9003	MER	0.178282	0.261361	0.17794	0.193879	0.786837	1.181808	0.787239	0.877548
m.barra.9003	DELL	0.123172	0.172253	0.119661	0.120806	0.385719	0.505738	0.366183	0.358425
m.barra.9003	HPQ	0.213384	0.269663	0.203956	0.209619	0.831212	1.016144	0.792208	0.788334
m.barra.9003	IBM	0.177285	0.251654	0.178425	0.188127	0.768331	1.063836	0.768904	0.83654
m.barra.9003	AA	0.156415	0.19538	0.153844	0.164443	0.957278	1.174494	0.946801	1.015842
m.barra.9003	CAT	0.143265	0.180631	0.144374	0.152673	0.772765	0.938364	0.779958	0.813915
m.barra.9003	PG	0.062337	0.091181	0.064481	0.063323	0.381208	0.550372	0.387347	0.394892
m.bnd	mature30year	0.137327	0.140966	0.137302	0.149323	1.017001	1.015294	1.025849	1.115901
m.bnd	mature20year	0.102011	0.106707	0.103465	0.108739	0.872867	0.896913	0.895977	0.930947
m.bnd	mature10year	0.111227	0.114902	0.113685	0.119584	0.814182	0.817779	0.834738	0.883196
m.bnd	mature5year	0.073043	0.076136	0.07561	0.083633	0.808718	0.85293	0.836505	0.917948
m.bnd	mature1year	0.037331	0.039802	0.040313	0.046121	0.543527	0.624228	0.586303	0.686845
m.cpice16.dp7503	CPI	0.120204	0.123036	0.115536	0.116131	1.151257	1.186614	1.135233	1.156678
m.cpice16.dp7503	CE16	0.130046	0.139531	0.134168	0.132465	0.724044	0.77052	0.739522	0.727509
m.decile1510	Decile1	0.113265	0.113932	0.112456	0.106455	1.015693	1.040614	1.020689	0.960883
m.decile1510	Decile5	0.098294	0.106671	0.100657	0.097623	0.837985	0.906497	0.864247	0.823829
m.decile1510	Decile10	0.127992	0.12699	0.125281	0.124801	0.8829	0.895538	0.866738	0.859225
m.fama.bond5203	m1.12	0.033871	0.036537	0.039215	0.032974	0.440281	0.467038	0.513667	0.4085
m.fama.bond5203	m24.36	0.048988	0.051925	0.050923	0.048007	0.609282	0.656064	0.635063	0.586611
m.fama.bond5203	m48.60	0.077537	0.082269	0.080911	0.077286	0.729969	0.775448	0.765103	0.719169
m.fama.bond5203	m61.120	0.076109	0.084175	0.080263	0.077515	0.668113	0.743663	0.713456	0.684398
m.gs1n3.5301	Treasury1year	0.014489	0.015863	0.015048	0.017536	0.575183	0.633253	0.579157	0.700819
m.gs1n3.5301	treasury3year	0.017253	0.018308	0.018942	0.020406	0.783489	0.854146	0.839184	0.917007
m.ibm3dx2603	IBM	0.151501	0.155382	0.151774	0.15211	1.106274	1.132193	1.112627	1.115107
m.ibm3dx2603	VW	0.065962	0.074524	0.065501	0.067114	0.650392	0.714168	0.642841	0.657765
m.ibm3dx2603	EW	0.053081	0.064402	0.056657	0.056188	0.574327	0.68032	0.616467	0.609914
m.ibm3dx2603	SP	0.060432	0.068676	0.060611	0.064226	0.61815	0.691526	0.61638	0.656774
m.ibmspln	IBM	0.132878	0.132064	0.130813	0.130816	1.008606	0.999219	0.985834	0.979656
m.ibmspln	SP	0.063627	0.069558	0.062167	0.063742	0.595338	0.659436	0.566031	0.58944
m.ibmvwewsp2603	IBM	0.151501	0.155382	0.152036	0.150805	1.106274	1.132193	1.115315	1.11179
m.ibmvwewsp2603	VW	0.065962	0.074524	0.066134	0.068736	0.650392	0.714168	0.65255	0.670005
m.ibmvwewsp2603	EW	0.053081	0.064402	0.056833	0.056863	0.574327	0.68032	0.617305	0.619975
m.ibmvwewsp2603	SP	0.060432	0.068676	0.060073	0.063546	0.61815	0.691526	0.608522	0.643655
w.gs1n36299	gs1	0.008964	0.00917	0.010681	0.010559	0.478006	0.50139	0.769482	0.783878
w.gs1n36299	gs3	0.009833	0.009955	0.012604	0.011228	0.544616	0.556852	0.953896	0.786979
w.tb3n6ms	w.tb3	0.004838	0.004783	0.010893	0.009231	0.388846	0.399066	0.952108	0.825241
w.tb3n6ms	w.tb6	0.004608	0.004737	0.010044	0.009715	0.378939	0.392207	0.866705	0.865585

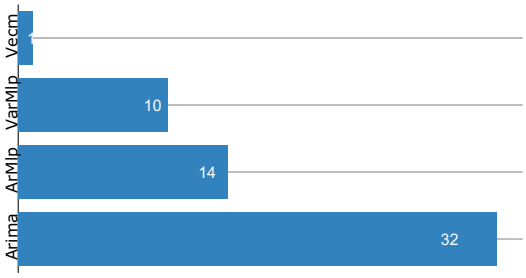
Table 3.16 – Les précisions de prédiction obtenues pour chaque variable.

3.6.3.3 Résultats et discussion

Dans la Table 3.16, nous présentons la précision de prédiction obtenue des modèles de prédiction utilisés pour chaque variable. Dans la figure 3.10, nous montrons le nombre de variables sur lesquelles chaque modèle fournit les meilleures prédictions.



(a) Comparaison en terme de RMSE



(b) Comparaison en terme de MASE

Figure 3.10 – Comparaison statistique des modèles étudiés.

Globalement, la comparaison entre les modèles montre que chacun d'entre eux surpasse les autres sur un ensemble de séries temporelles. Mais elle montre aussi l'intérêt de l'utilisation de modèles non linéaires, car ils améliorent la qualité des prédiction de nombreuses variables.

Les modèles ARIMA et ARMLP uni-variés sont les plus sélectionnés.

Nous remarquons également que le modèle VARMLP est plus utilisé que le modèle VEC, qui est sélectionné au plus 2 fois. Ainsi, le modèle non linéaire multi-varié détecte davantage les dépendances entre les séries temporelles.

Ainsi, le modèle fournit dans la bibliothèque **NlinTS** et compétitif avec les modèles existants, et peut être utilisé par les utilisateur du langage **R** pour la prédiction des séries temporelles.

Conclusions et Perspectives

Conclusions

Dans ce travail, nous avons étudié différents modèles de prédiction des séries temporelles. Nous avons discuté mathématiquement leurs principes, puis nous avons distingué leurs avantages et inconvénients. Nous avons commencé par les modèles linéaires auto-régressifs uni-variés et multi-variés, qui sont parmi les plus utilisés dans la littérature, et quelques modèles basés sur les réseaux de neurones artificiels, qui sont de plus en plus utilisés dans la prédiction des séries temporelles. Nous avons aussi étudié les modèles linéaires basés sur les méthodes de régularisation. Ces modèles ont la caractéristique de supporter des données avec beaucoup de variables.

Comme notre problématique porte sur la prédiction des séries temporelles larges, l'étude des modèles de prédiction seule n'est pas suffisante, car il faut réduire aussi le nombre de variables prédictives avant d'appliquer ces modèles. Cela nous a amené à étudier les méthodes de réduction de dimension et de sélection de variables. En effet, ces deux techniques permettent de réduire la taille de l'ensemble des variables prédictives dans un modèle de prédiction mutlivarié. Mais leurs principes sont très différents. D'un coté, les méthodes de réduction génèrent de nouvelles variables, généralement comme combinaison des variables originales, tout en représentant le mieux possible leurs propriétés statistiques. D'un autre coté, les méthodes de sélection de variables choisissent un sous-ensemble de l'ensemble des variables originales, en tenant en compte les dépendances entre les variables, et éventuellement, la notion de pertinence par rapport à une variable cible.

Nous avons constaté une différence entre les méthodes de réduction étudiées. Les méthodes de réduction la dimension comme l'Analyse en Composantes Principales et l'Analyse Factorielle appartiennent à la catégorie d'apprentissage non-supervisé, car elles réduisent la dimension des variables simplement en fonction des caractéristiques statistiques des variables prédictives. Dans la plupart des problèmes de prédiction des séries temporelles numériques, et en particulier dans notre problématique, les données d'apprentissage contiennent les valeurs passées de variables cibles, et c'est utile de les exploiter. En se basant sur cette remarque, nous nous sommes focaliser sur les méthodes de sélection de variables supervisées. Nous avons proposé une démarche pour la prédiction des séries temporelles larges, en représentant les dépendances entre les variables sous forme de graphe en utilisant la causalité prédictive comme mesure de pertinence des variables. Cette approche permet de résoudre quelques problèmes trouvés dans des méthodes de sélection de variable basées sur la causalité, comme SUN, Jiuyong LI, J. LIU et al. 2014 et ZHANG, HU, XIE et al. 2014 (*cf.* 3.2.1). Cet analyse nous

à conduit à proposer deux algorithmes différents de sélection de variables, qui sont spécifiques à la prédiction des séries temporelles. Le premier est basé sur le regroupement, et le deuxième est basé sur la classement.

Pour terminer, des résultats sur deux données réelles bien étudiées dans la littérature, STOCK et M. W. WATSON 2012 ; JIANG, ATHANASOPOULOS, HYNDMAN et al. 2017, confirment l'utilité des deux algorithmes proposés par rapport aux méthodes de réduction de dimension évaluées, et aussi par rapport au méthodes de régularisation utilisées dans STOCK et M. W. WATSON 2012 ; JIANG, ATHANASOPOULOS, HYNDMAN et al. 2017.

Comme extension de ces travaux, dans le futur, nous souhaitons envisager des méthodes de réduction temporelles, car les causalités entre les variables peuvent changer d'une période à une autre. De ce point de vue, pour chaque variable cible, il y aura différentes sélections de variables, chacune est spécifique à une période donnée. Nous envisagerons aussi évaluer d'autres algorithmes d'analyse des données complexe sur les graphes de causalités.

Perspectives

Il est important de noter que lorsqu'on traite les séries temporelles massives, la distribution du stockage des données et du calcul sont nécessaires. La procédure adoptée dans ce manuscrit est spécifique à ce type de données, mais les expérimentations ont été effectuées sur des données de dimension moyenne en utilisant des calculs parallèles classiques Y. HMAMOUCHE 2018.

Le parallélisme est différent du calcul distribué, puisque il consiste à réaliser des taches en parallèles, et son but principale est de minimiser le temps de calcul, par exemple, en utilisant plusieurs cœurs d'un processeur en même temps.

Or, le calcul distribué est plus générique et plus précis. Son premier objectif est la faisabilité, c-à-d., pouvoir exécuter les algorithmes standards sur des données massives stockées de manière distribuée sur une plateforme composée de plusieurs machines.

Le calcul distribué est basé sur deux notions principales, le stockage des données, et le traitement distribué de ces données. Le stockage est généralement réalisé par partitions sur plusieurs nœuds. L'idée de partitionner les données rend la phase de développement des algorithmes délicate, en plus, ce n'est pas toujours possible d'adapter des algorithmes standards vers des plateformes distribuées sans perte d'information.

De ce point de vue, notre future projet consiste à rendre notre processus de prédiction distribué. Nous avons déjà commencé une partie de cela, à savoir la phase de stockage des données, le calcul des dépendances entre les séries temporelles, et la sélection des variables. Actuellement, nous utilisons la plateforme *Hadoop / Spark*. Par défaut les données distribuée dans Spark (*Resilient Distributed Datasets*), sont partitionnées sur les différentes nœuds par lignes. Dans

notre cas, l'objet principale que nous traitions est les variables, par conséquent nous stockons les données par variables, pour rendre l'étape de calcul des dépendances entre les séries temporelles plus pratique.

Essayons de discuter la scalabilité des deux algorithmes proposés GFSM et PEHAR. La causalité de Granger est calculée en comparant la précision des prédictions entre le modèle uni-varié et le modèle bi-varié (habituellement suivi d'un test de signification). Cela nécessite de construire deux modèles de prédictions pour calculer la causalité de Granger, un modèle uni-varié sur la variable cible et un modèle bi-varié VAR avec une variable cible et un prédicteur supplémentaire.

L'algorithme présenté utilise le graphe de causalités. Il nécessite le calcul d'un ensemble de tuples $G = (P \times P) \setminus \Delta$, où P est l'ensemble des variables prédictives et $\Delta = (p_1, p_2) \in P \times P : p = p$. Par conséquent, nous devons calculer des modèles uni-variés pour tous les éléments dans P et des modèles bi-variés pour des tuples dans G. Ceci est trivialement scalable car tous les modèles peuvent être calculés indépendamment. Enfin, les résultats doivent être regroupés par variable cible, et les tests statistiques de la causalité de Granger sont calculés à partir d'un modèle uni-varié sur les variables $p \in P$, et d'un modèle bi-varié $(p_1, p_2) \in G$ (tâche simple dans l'approche *MapReduce*). En résultat, nous construisons la matrice d'adjacence, qui peut être stocké par block de sous matrices dans les nœuds de calcul. De même pour les calcul des Entropies de Transfert qui sont plus simples et plus rapides par rapport à la causalité de Granger.

Ensuite, en suivant les étapes de l'algorithme GFSM, il faut appliquer une méthode de regroupement sur la matrice résultante. Pour les séries temporelles financières, il est raisonnable de supposer que la matrice résultante aura une taille réduite et que le nœud de calcul sera unique. Dans de rares cas, lorsque la matrice est très grande, des algorithmes de classification évolutifs pourraient être utilisés, comme k-medoids et d'autres méthodes présentées dans BABU p.d. et WU, ZHU, HUANG et al. 2015. Un avantage de l'algorithme PEHAR est qu'il est facilement scalable, surtout avec la méthode itérative (cf. Algorithme 3), car les étapes de mise à jour et de normalisation des vecteurs *Hubs* et *Authorités* s'effectuent par des simples opérations *MapReduce*.

De ce point de vue, nous proposons une version distribuée du processus de prédiction présenté, capable de traiter de grandes séries temporelles multidimensionnelles. La particularité de ce système réside dans le fait qu'il repose sur une étape de sélection distribuée afin de réduire les modèles de prédiction multi-variés en de nombreux petits modèles pouvant être exécutés en parallèle. La mise en œuvre de ce processus est effectuée dans l'environnement distribué Spark/Haddop Y. HMAMOUCHE 2018.

Structure du system proposé

Nous proposons un système qui se résume aux étapes suivantes :

— Stockage de séries temporelles dans un ensemble de partitions de variables.
— Préparation de donnés nous appliquons dans cette étape la méthode de

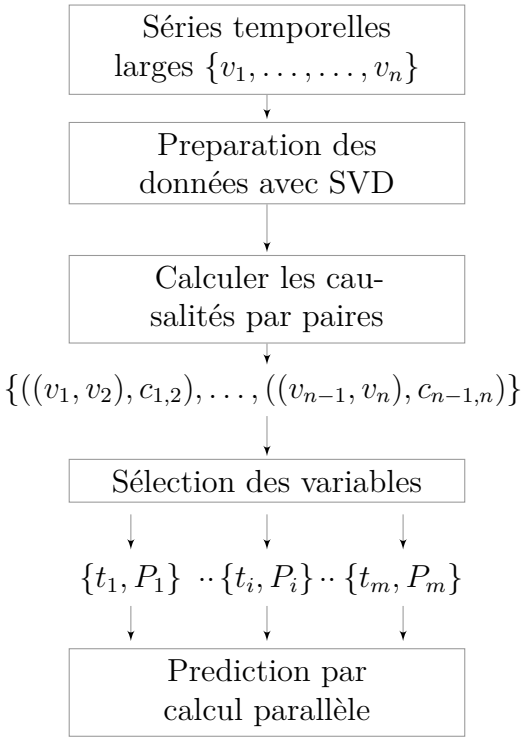


Figure 3.11 – Illustration du processus proposé.

décomposition en valeurs singulières (SVD) pour approximer les données brutes afin de diminuer les bruits.

— Calculer les dépendances entre les séries.

— Application d'une version distribuée de notre l'algorithme PEHAR pour déterminer les prédicteurs de chaque variable cible.

— Exécution de modèles de prédiction en parallèle (calcul multicœur).

Le schéma de ce système est montré dans la figure 3.11, où $\{v_1, \ldots, v_n\}$ sont les variables d'origine, $c_{i,j}$ représente la causalité de v_i à v_j, et $\{t_i, P_i\}$ constitue une entrée du modèle de prédiction (variable cible et ensemble de prédicteurs associés), $\{t_1, \ldots, t_m\}$ sont les variables cible et $\{P_1, \ldots, P_i\}$ are est l'ensemble des prédicteurs sélectionnés. La taille de chaque ensemble P_i est $k << n$. Soulignons que différentes valeurs de k peuvent être testées, comme dans les expériences présentées dans la section précédente. Dans ce cas, les résultats doivent être regroupés par variables cibles pour trouver le nombre approprié de variables.

Dans le futur, nous souhaitons optimiser la manière du stockage des séries temporelles. Pour ce faire, au lieu d'utiliser la méthode par défaut de *Spark*, qui gère cette phase selon les ressources existantes, le nombre de cœurs exécuteurs, ..., nous planifions contrôler les partitions stockées dans chaque nœud, en tenant en compte les dépendances entre les variables pour minimiser le temps transfert d'information entre les nœuds de calcul lors de la sélection des variables prédictives dans un modèle de prédiction.

Bibliographie

[ANM01] Ajith ABRAHAM, Baikunth NATH et P. K. MAHANTI. « Hybrid Intelligent Systems for Stock Market Analysis ». en. In : *Computational Science - ICCS 2001*. Springer, Berlin, Heidelberg, mai 2001, p. 337–345. DOI : 10.1007/3-540-45718-6_38 (cf. p. 47).

[Aka74] H. AKAIKE. « A New Look at the Statistical Model Identification ». In : *IEEE Transactions on Automatic Control* 19.6 (déc. 1974), p. 716–723. ISSN : 0018-9286. DOI : 10.1109/TAC.1974.1100705 (cf. p. 21).

[AD91] Hussein ALMUALLIM et Thomas G. DIETTERICH. « Learning with Many Irrelevant Features ». In : *Proceedings of the Ninth National Conference on Artificial Intelligence - Volume 2*. AAAI'91. Anaheim, California : AAAI Press, 1991, p. 547–552. ISBN : 978-0-262-51059-2 (cf. p. 32).

[AC15] Alev Dilek AYDIN et Seyma Caliskan CAVDAR. « Comparison of Prediction Performances of Artificial Neural Network (ANN) and Vector Autoregressive (VAR) Models by Using the Macroeconomic Variables of Gold Prices, Borsa Istanbul (BIST) 100 Index and US Dollar-Turkish Lira (USD/TRY) Exchange Rates ». In : *Procedia Economics and Finance*. IISES 3rd and 4th Economics and Finance Conference 30 (jan. 2015), p. 3–14. ISSN : 2212-5671. DOI : 10.1016/S2212-5671(15)01249-6 (cf. p. 29).

[Bab] M. Chaitanya Kumari BABU P. Nagendra. « IJETT - Survey on Clustering on the Cloud by UsingMap Reduce in Large Data Applications ». English. In : *International Journal of Engineering Trends and Technology* () (cf. p. 97).

[BN08] Jushan BAI et Serena NG. « Forecasting Economic Time Series Using Targeted Predictors ». In : *Journal of Econometrics*. Honoring the research contributions of Charles R. Nelson 146.2 (oct. 2008), p. 304–317. ISSN : 0304-4076. DOI : 10.1016/j.jeconom.2008.08.010 (cf. p. 50).

[BBS09] Lionel BARNETT, Adam B. BARRETT et Anil K. SETH. « Granger Causality and Transfer Entropy Are Equivalent for Gaussian Variables ». In : *Physical Review Letters* 103.23 (déc. 2009). ISSN : 0031-9007, 1079-7114. DOI : 10.1103/PhysRevLett.103.238701. arXiv : 0910.4514 (cf. p. 35).

[Bar17] Sylvain BARTHELEMY. *Mlp : Neural Network Library (Multi-Layer Perceptron)*. Oct. 2017 (cf. p. 88).

[BEK13] Michele BENZI, Ernesto ESTRADA et Christine KLYMKO. « Ranking Hubs and Authorities Using Matrix Functions ». In : *Linear Algebra and its Applications* 438.5 (mar. 2013), p. 2447–2474. ISSN : 0024-3795. DOI : 10.1016/j.laa.2012.10.022 (cf. p. 59).

[Box13] George BOX. « Box and Jenkins : Time Series Analysis, Forecasting and Control ». en. In : *A Very British Affair*. Palgrave Advanced Texts in Econometrics. Palgrave Macmillan UK, 2013, p. 161–215. ISBN : 978-1-349-35027-8 978-1-137-29126-4. DOI : 10.1057/9781137291264_6 (cf. p. 10, 17, 18).

[BH69] Arthur Earl BRYSON et joint author.) HO Yu-Chi. *Applied Optimal Control : Optimization, Estimation, and Control*. English. Waltham, Mass. : Blaisdell Pub. Co, 1969 (cf. p. 27).

[Car14] René CARMONA. « Multivariate Time Series, Linear Systems and Kalman Filtering ». en. In : *Statistical Analysis of Financial Data in R*. Springer Texts in Statistics. Springer, New York, NY, 2014, p. 423–472. ISBN : 978-1-4614-8787-6 978-1-4614-8788-3. DOI : 10.1007/978-1-4614-8788-3_7 (cf. p. 26).

[Cec+17] M. CECI, R. CORIZZO, F. FUMAROLA et al. « Predictive Modeling of PV Energy Production : How to Set Up the Learning Task for a Better Prediction ? » In : *IEEE Transactions on Industrial Informatics* 13.3 (juin 2017), p. 956–966. ISSN : 1551-3203. DOI : 10.1109/TII.2016.2604758 (cf. p. 83).

[Cho+15] François CHOLLET et al. « Keras : Deep Learning Library for Theano and Tensorflow ». In : *URL : https ://keras. io/k* (2015) (cf. p. 63).

[DF79] David A. DICKEY et Wayne A. FULLER. « Distribution of the Estimators for Autoregressive Time Series with a Unit Root ». In : *Journal of the American Statistical Association* 74.366a (juin 1979), p. 427–431. ISSN : 0162-1459. DOI : 10.1080/01621459.1979.10482531 (cf. p. 92).

[DGE16] Cristian-Dragos DUMITRU, Adrian GLIGOR et Calin ENACHESCU. « Solar Photovoltaic Energy Production Forecast Using Neural Networks ». In : *Procedia Technology*. 9th International Conference Interdisciplinarity in Engineering, INTER-ENG 2015, 8-9 October 2015, Tirgu Mures, Romania 22 (jan. 2016), p. 808–815. ISSN : 2212-0173. DOI : 10.1016/j.protcy.2016.01.053 (cf. p. 83).

[Efr+04] Bradley EFRON, Trevor HASTIE, Iain JOHNSTONE et al. « Least Angle Regression ». In : *The Annals of Statistics* 32.2 (2004), p. 407–499. ISSN : 0090-5364, 2168-8966. DOI : 10.1214/009053604000000067 (cf. p. 50).

[Eic01] Michael EICHLER. *Granger Causality Graphs for Multivariate Time Series*. eng. Working Paper 10.11588/heidok.00020749. Heidelberg, juin 2001. DOI : DOI:10.11588/heidok.00020749 (cf. p. 53).

[FHL14] Jianqing FAN, Fang HAN et Han LIU. « Challenges of Big Data Analysis ». In : *National Science Review* 1.2 (juin 2014), p. 293–314. ISSN : 2095-5138. DOI : 10.1093/nsr/nwt032 (cf. p. 46).

[For+00] Mario FORNI, Marc HALLIN, Marco LIPPI et al. « The Generalized Dynamic-Factor Model : Identification and Estimation ». In : *The Review of Economics and Statistics* 82.4 (nov. 2000), p. 540–554. ISSN : 0034-6535. DOI : 10.1162/003465300559037 (cf. p. 45).

[Gan+14] A. GANDELLI, F. GRIMACCIA, S. LEVA et al. « Hybrid Model Analysis and Validation for PV Energy Production Forecasting ». In : *2014 International Joint Conference on Neural Networks (IJCNN)*. Juil. 2014, p. 1957–1962. DOI : 10.1109/IJCNN.2014.6889786 (cf. p. 83).

[GC08] Sarah GELPER et Christophe CROUX. « Least Angle Regression for Time Series Forecasting with Many Predictors ». en. In : (jan. 2008) (cf. p. 51).

[GEW77] J. GEWEKE. « The Dynamic Factor Analysis of Economic Time Series ». In : *Latent Variables in Socio-Economic Models* (1977) (cf. p. 45).

[Gra80] C. W. J. GRANGER. « Testing for Causality ». In : *Journal of Economic Dynamics and Control* 2 (jan. 1980), p. 329–352. ISSN : 0165-1889. DOI : 10.1016/0165-1889(80)90069-X (cf. p. 33, 47, 89).

[Gra14] Spencer GRAVES. *FinTS : Companion to Tsay (2005) Analysis of Financial Time Series*. R package version 0.4-5. 2014. URL : https://CRAN.R-project.org/package=FinTS.

[GS99] Jatinder N. D GUPTA et Randall S SEXTON. « Comparing Backpropagation with a Genetic Algorithm for Neural Network Training ». In : *Omega* 27.6 (déc. 1999), p. 679–684. ISSN : 0305-0483. DOI : 10.1016/S0305-0483(99)00027-4 (cf. p. 28).

[Hal98] Mark A. HALL. *Correlation-Based Feature Selection for Machine Learning*. Rapp. tech. 1998 (cf. p. 37).

[HEC89] R. HECHT-NIELSEN. « Kolmogorov's Mapping Neural Network Existence Theorem ». In : *Proceedings of the International Joint Conference in Neural Networks* 3 (1989), p. 11–14 (cf. p. 27).

[Hec89] R. HECHT-NIELSEN. « Theory of the Backpropagation Neural Network ». In : *International 1989 Joint Conference on Neural Networks*. 1989, 593–605 vol.1. DOI : 10.1109/IJCNN.1989.118638 (cf. p. 27).

[Hma18] Y. HMAMOUCHE. *Large-Time-Series-Prediction*. https://github.com/Hmamouche/Large-Time-Series-Prediction. 2018 (cf. p. 65, 87, 96, 97).

[Hma17] Youssef HMAMOUCHE. *NlinTS : Non Linear Time Series Analysis- An Rcpp Package and Benchmark Experiments*. Nov. 2017 (cf. p. 88).

[HCL17] Youssef HMAMOUCHE, Alain CASALI et Lotfi LAKHAL. « A Causality-Based Feature Selection Approach for Multivariate Time Series Forecasting ». In : *DBKDA*. 2017, p. 97–102. ISBN : 978-1-61208-558-6 (cf. p. 47, 54).

[HLC18] Youssef HMAMOUCHE, Lotfi LAKHAL et Alain CASALI. « Predictors Extraction in Time Series using Authorities-Hubs Ranking ». In : *2018 IEEE International Conference on Data Mining Workshops (ICDMW)*. IEEE, (accepted) 2018 (cf. p. 59).

[Hma+17] Youssef HMAMOUCHE, Piotr PRZYMUS, Alain CASALI et al. « GFSM : A Feature Selection Method For Improving Time Series Forecasting. » In : *International Journal On Advances in Systems and Measurements* 10.3 and 4 (déc. 2017), p. 255–264. ISSN : 1942-261x (cf. p. 54).

[HS97] Sepp HOCHREITER et Jürgen SCHMIDHUBER. « Long Short-Term Memory ». In : *Neural Computation* 9.8 (nov. 1997), p. 1735–1780. ISSN : 0899-7667. DOI : 10.1162/neco.1997.9.8.1735 (cf. p. 30).

[HK70] Arthur E. HOERL et Robert W. KENNARD. « Ridge Regression : Biased Estimation for Nonorthogonal Problems ». In : *Technometrics* 12.1 (1970), p. 55–67. ISSN : 0040-1706. DOI : 10.2307/1267351 (cf. p. 24, 48).

[Hyn] Rob J. HYNDMAN. « Another Look at Forecast-Accuracy Metrics for Intermittent Demand ». In : *Foresight, International Journal of Applied Forecasting* (), p. 43–46 (cf. p. 64).

[Jia+17] Bin JIANG, George ATHANASOPOULOS, Rob J. HYNDMAN et al. *Macroeconomic Forecasting for Australia Using a Large Number of Predictors*. Monash Econometrics and Business Statistics Working Paper 2/17. Monash University, Department of Econometrics and Business Statistics, 2017 (cf. p. 18, 36, 62–64, 81, 96).

[Joh88] Søren JOHANSEN. « Statistical Analysis of Cointegration Vectors ». In : *Journal of Economic Dynamics and Control* 12.2 (juin 1988), p. 231–254. ISSN : 0165-1889. DOI : 10.1016/0165-1889(88)90041-3 (cf. p. 10, 17, 22).

[Joh91] Søren JOHANSEN. « Estimation and Hypothesis Testing of Cointegration Vectors in Gaussian Vector Autoregressive Models ». In : *Econometrica* 59.6 (1991), p. 1551–1580. ISSN : 0012-9682. DOI : 10.2307/2938278 (cf. p. 17, 22, 84).

[JKP94] George H. JOHN, Ron KOHAVI et Karl PFLEGER. « Irrelevant Features and the Subset Selection Problem ». In : *Machine Learning : Proceedings of the Eleventh International*. Morgan Kaufmann, 1994, p. 121–129 (cf. p. 32).

[Jol86] I. T. JOLLIFFE. « Principal Component Analysis and Factor Analysis ». en. In : *Principal Component Analysis*. Springer Series in Statistics. Springer, New York, NY, 1986, p. 115–128. ISBN : 978-1-4757-1906-2 978-1-4757-1904-8. DOI : 10.1007/978-1-4757-1904-8_7 (cf. p. 39, 41, 62).

[Kal60] R. E. KALMAN. « A New Approach to Linear Filtering and Prediction Problems ». In : *Journal of Basic Engineering* 82.1 (mar. 1960), p. 35–45. ISSN : 0098-2202. DOI : 10.1115/1.3662552 (cf. p. 25).

[KR09] Leonard KAUFMAN et Peter J. ROUSSEEUW. *Finding Groups in Data : An Introduction to Cluster Analysis*. en. John Wiley & Sons, sept. 2009. ISBN : 978-0-470-31748-8 (cf. p. 82).

[KS] Koffka KHAN et Ashok SAHAI. « A Comparison of BA, GA, PSO, BP and LM for Training Feed Forward Neural Networks in e-Learning Context ». In : *International Journal of Intelligent Systems and Applications* 4.7 (), p. 23–29. ISSN : 2074-904X (cf. p. 28).

[Kle99] Jon M. KLEINBERG. « Authoritative Sources in a Hyperlinked Environment ». In : *J. ACM* 46.5 (sept. 1999), p. 604–632. ISSN : 0004-5411. DOI : 10.1145/324133.324140 (cf. p. 56, 57).

[KRA15] Irena KOPRINSKA, Mashud RANA et Vassilios G. AGELIDIS. « Correlation and Instance Based Feature Selection for Electricity Load Forecasting ». In : *Knowledge-Based Systems* 82 (juil. 2015), p. 29–40. ISSN : 0950-7051. DOI : 10.1016/j.knosys.2015.02.017 (cf. p. 47).

[Kůr92] Věra KŮRKOVÁ. « Kolmogorov's Theorem and Multilayer Neural Networks ». In : *Neural Networks* 5.3 (jan. 1992), p. 501–506. ISSN : 0893-6080. DOI : 10.1016/0893-6080(92)90012-8 (cf. p. 27).

[LC14] Jiahan LI et Weiye CHEN. « Forecasting Macroeconomic Time Series : LASSO-Based Approaches and Their Forecast Combinations with Dynamic Factor Models ». In : *International Journal of Forecasting* 30.4 (oct. 2014), p. 996–1015. ISSN : 0169-2070. DOI : 10.1016/j.ijforecast.2014.03.016 (cf. p. 50).

[MCH02] C. MEEK, D. CHICKERING et D. HECKERMAN. « Autoregressive Tree Models for Time-Series Analysis ». In : *Proceedings of the 2002 SIAM International Conference on Data Mining*. Proceedings. Society for Industrial and Applied Mathematics, avr. 2002, p. 229–244. ISBN : 978-0-89871-517-0. DOI : 10.1137/1.9781611972726.14 (cf. p. 24, 25).

[Par86] David B. PARKER. « A Comparison of Algorithms for Neuron-like Cells ». In : *AIP Conference Proceedings* 151.1 (août 1986), p. 327–332. ISSN : 0094-243X. DOI : 10.1063/1.36233 (cf. p. 27).

[Ped+11] Fabian PEDREGOSA, Gaël VAROQUAUX, Alexandre GRAMFORT et al. « Scikit-Learn : Machine Learning in Python ». In : *Journal of Machine Learning Research* 12.Oct (2011), p. 2825–2830. ISSN : ISSN 1533-7928 (cf. p. 62).

[PEÑ09] DANIEL PEÑA. « Dimension Reduction in Time Series and the Dynamic Factor Model ». In : *Biometrika* 96.2 (2009), p. 494–496. ISSN : 0006-3444 (cf. p. 46).

[Prz+17] P. PRZYMUS, Y. HMAMOUCHE, A. CASALI et al. « Improving Multivariate Time Series Forecasting with Random Walks with Restarts on Causality Graphs ». In : *2017 IEEE International Conference on Data Mining Workshops (ICDMW)*. Nov. 2017, p. 924–931. DOI : 10.1109/ICDMW.2017.127 (cf. p. 85, 87).

[SM95] H. SCHNEEWEISS et H. MATHES. « Factor Analysis and Principal Components ». In : *Journal of Multivariate Analysis* 55.1 (oct. 1995), p. 105–124. ISSN : 0047-259X. DOI : 10.1006/jmva.1995.1069 (cf. p. 41).

[SSM98] Bernhard SCHÖLKOPF, Alexander SMOLA et Klaus-Robert MÜLLER. « Nonlinear Component Analysis as a Kernel Eigenvalue Problem ». In : *Neural Computation* 10.5 (juil. 1998), p. 1299–1319. ISSN : 0899-7667. DOI : 10.1162/089976698300017467 (cf. p. 41, 62).

[Sch00] Thomas SCHREIBER. « Measuring Information Transfer ». In : *Physical Review Letters* 85.2 (juil. 2000), p. 461–464. DOI : 10.1103/PhysRevLett.85.461 (cf. p. 33, 34).

[SW11] James H. STOCK et Mark WATSON. « Dynamic Factor Models ». In : *Oxford Handbook on Economic Forecasting*. Oxford University Press, 2011 (cf. p. 45).

[SW02] James H. STOCK et Mark W. WATSON. « Forecasting Using Principal Components From a Large Number of Predictors ». In : *Journal of the American Statistical Association* 97.460 (déc. 2002), p. 1167–1179. ISSN : 0162-1459. DOI : 10.1198/016214502388618960 (cf. p. 46).

[SW06] James H. STOCK et Mark W. WATSON. « Chapter 10 Forecasting with Many Predictors ». In : *Handbook of Economic Forecasting*. Sous la dir. de C. W. J. Granger G. ELLIOTT et A. TIMMERMANN. T. 1. Elsevier, 2006, p. 515–554. DOI : 10.1016/S1574-0706(05)01010-4 (cf. p. 36, 45, 55).

[SW12] James H. STOCK et Mark W. WATSON. « Generalized Shrinkage Methods for Forecasting Using Many Predictors ». In : *Journal of Business & Economic Statistics* 30.4 (oct. 2012), p. 481–493. ISSN : 0735-0015. DOI : 10.1080/07350015.2012.715956 (cf. p. 18, 45, 46, 51, 62–64, 81, 96).

[Sun+14] Youqiang SUN, Jiuyong LI, Jixue LIU et al. « Using Causal Discovery for Feature Selection in Multivariate Numerical Time Series ». en. In : *Machine Learning* 101.1-3 (juil. 2014), p. 377–395. ISSN : 0885-6125, 1573-0565. DOI : 10.1007/s10994-014-5460-1 (cf. p. 38, 47, 48, 55, 95).

[TD11] Melinda THIELBAR et D. A. DICKEY. *Neural Networks for Time Series Forecasting : Practical Implications of Theoretical Results*. 2011 (cf. p. 18).

[Tib94] Robert TIBSHIRANI. « Regression Shrinkage and Selection Via the Lasso ». In : *Journal of the Royal Statistical Society, Series B* 58 (1994), p. 267–288 (cf. p. 24, 48, 49).

[TWH01] Robert TIBSHIRANI, Guenther WALTHER et Trevor HASTIE. « Estimating the Number of Clusters in a Data Set via the Gap Statistic ». en. In : *Journal of the Royal Statistical Society : Series B (Statistical Methodology)* 63.2 (jan. 2001), p. 411–423. ISSN : 1467-9868. DOI : 10.1111/1467-9868.00293 (cf. p. 82).

[TB99a] M. E. TIPPING et Christopher BISHOP. « Probabilistic Principal Component Analysis ». In : *Journal of the Royal Statistical Society, Series B* 21/3 (jan. 1999) (cf. p. 41).

[TB99b] Michael E. TIPPING et Christopher M. BISHOP. « Mixtures of Probabilistic Principal Component Analyzers ». In : *Neural Computation* 11.2 (fév. 1999), p. 443–482. ISSN : 0899-7667. DOI : 10.1162/089976699300016728 (cf. p. 62).

[TLH00] Adrian TRAPLETTI, Friedrich LEISCH et Kurt HORNIK. « Stationary and Integrated Autoregressive Neural Network Processes ». In : *Neural Computation* 12.10 (oct. 2000), p. 2427–2450. ISSN : 0899-7667. DOI : 10.1162/089976600300015006 (cf. p. 10).

[WRR03] Michael E. WALL, Andreas RECHTSTEINER et Luis M. ROCHA. « Singular Value Decomposition and Principal Component Analysis ». en. In : *A Practical Approach to Microarray Data Analysis*. Springer, Boston, MA, 2003, p. 91–109. ISBN : 978-1-4020-7260-4 978-0-306-47815-4. DOI : 10.1007/0-306-47815-3_5 (cf. p. 41).

[WS02] Ke WANG et Ming-Yen Thomas SU. « Item Selection by "Hub-Authority" Profit Ranking ». In : *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '02. New York, NY, USA : ACM, 2002, p. 652–657. ISBN : 978-1-58113-567-1. DOI : 10.1145/775047.775144 (cf. p. 56).

[Wu+15] Y. WU, Y. ZHU, T. HUANG et al. « Distributed Discord Discovery : Spark Based Anomaly Detection in Time Series ». In : *2015 IEEE 17th International Conference on High Performance Computing and Communications, 2015 IEEE 7th International Symposium on Cyberspace Safety and Security, and 2015 IEEE 12th International Conference on Embedded Software and Systems*. Août 2015, p. 154–159. DOI : 10.1109/HPCC-CSS-ICESS.2015.228 (cf. p. 97).

[Wut08] Dhoriva U. WUTSQA. « The Var-NN Model for Multivariate Time Series Forecasting ». In : *MatStat* 8.1 (jan. 2008), p. 35–43 (cf. p. 29).

[WSS06] Dhoriva Urwatul WUTSQA, Suryo Guritno SUBANAR et Zanzawi SUJUTI. « Forecasting Performance of VAR-NN and VARMA Models ». In : *Proceedings of the 2nd IMT-GT Regional Conference on Mathematics*. 2006 (cf. p. 29).

[Yan10] Xin-She YANG. « A New Metaheuristic Bat-Inspired Algorithm ». en. In : *Nature Inspired Cooperative Strategies for Optimization (NICSO 2010)*. Studies in Computational Intelligence. Springer, Berlin, Heidelberg, 2010, p. 65–74. ISBN : 978-3-642-12537-9 978-3-642-12538-6. DOI : 10.1007/978-3-642-12538-6_6 (cf. p. 28).

[YA10] Kazuyoshi YATA et Makoto AOSHIMA. « Effective PCA for High-Dimension, Low-Sample-Size Data with Singular Value Decomposition of Cross Data Matrix ». In : *Journal of Multivariate Analysis* 101.9 (oct. 2010), p. 2060–2077. ISSN : 0047-259X. DOI : 10.1016/j.jmva.2010.04.006 (cf. p. 41).

[YS06] Hyunjin YOON et Cyrus SHAHABI. « Shahabi : Feature Subset Selection on Multivariate Time Series with Extremely Large Spatial Features Data Mining Workshops ». In : *ICDM Workshops 2006. Sixth IEEE International Conference on Volume , Issue , Dec. 2006 Page(s) :337 - 342 Digital Object Identifier 10.1109/ICDMW.2006.81*. 2006 (cf. p. 47).

[YL03] Lei YU et Huan LIU. « Feature Selection for High-Dimensional Data : A Fast Correlation-Based Filter Solution ». English (US). In : 2003 (cf. p. 62).

[Zha+14] Xiangzhou ZHANG, Yong HU, Kang XIE et al. « A Causal Feature Selection Algorithm for Stock Prediction Modeling ». In : *Neurocomputing*. SI Computational Intelligence Techniques for New Product Development 142 (oct. 2014), p. 48–59. ISSN : 0925-2312. DOI : 10.1016/j.neucom.2014.01.057 (cf. p. 38, 55, 95).

[ZE17] Xiao ZHONG et David ENKE. « Forecasting Daily Stock Market Return Using Dimensionality Reduction ». In : *Expert Systems with Applications* 67 (2017), p. 126–139 (cf. p. 47).