Sous la direction scientifique de
Nathalie de Marcellis-Warin – Benoit Dostie
Sous la coordination de
Genevieve Dufour

Le Québec économique 9

Perspectives et défis de la transformation numérique

Chapitre 12

PRÉVISION MACROÉCONOMIQUE DANS L'ÈRE DES DONNÉES MASSIVES ET DE L'APPRENTISSAGE AUTOMATIQUE

DALIBOR STEVANOVIC

CIRANO

2020CH-12
CHAPITRE DE LIVRE

CH

Chapitre 12

PRÉVISION MACROÉCONOMIQUE DANS L'ÈRE DES DONNÉES MASSIVES ET DE L'APPRENTISSAGE AUTOMATIQUE

Dalibor Stevanovic

Professeur titulaire à l'École des sciences de la gestion de l'Université du Québec à Montréal, chercheur et fellow au CIRANO

Avec la collaboration de Hugo Couture et de Maxime Leroux

Résumé

L'accessibilité des données massives et l'avancement des techniques d'apprentissage automatique ont considérablement changé la façon d'approcher le problème de prévision macroéconomique. L'utilisation des techniques d'apprentissage automatique et de l'inclusion des composantes principales dans les modèles standards à l'application des réseaux de neurones est cependant une succession de boîtes noires. Ce chapitre dresse le portrait des avancées récentes dans le domaine de la prévision macroéconomique dans un contexte de données massives et de l'apprentissage automatique. Nous y présentons différents groupes de modèles de prévision basés sur la réduction de dimension, la régularisation, les ensembles et la non-linéarité. Nous discutons des défis qu'ils présentent pour les praticiens. Finalement, nous appliquons ces méthodes pour la prévision de la croissance de l'économie québécoise.

Introduction

Prévision macroéconomique et apprentissage automatique

Traditionnellement, la modélisation macroéconométrique a été guidée par le principe de parcimonie, et les modèles simples avec peu de paramètres à estimer étaient souvent préférés à des modèles non linéaires comprenant un nombre élevé de paramètres (Swanson et White, 1997 ; Stock et Watson, 1999 ; Teräsvirta, 2006). Or, l'accessibilité des données massives et l'avancement des techniques d'apprentissage automatique (machine learning ou ML ; aussi « apprentissage machine ») ont considérablement changé la façon d'approcher le problème de prévision macroéconomique (Kotchoni, Leroux et Stevanovic, 2019 ; Goulet Coulombe, Leroux, Stevanovic et Surprenant, 2019). Par contre, ces changements causent aussi deux problèmes. Premièrement, les données massives accessibles se résument à un très grand nombre de séries temporelles, et non à une hausse des périodes observées. Dans une régression prédictive linéaire, ceci implique plusieurs paramètres inconnus, ce qui rend leur estimation par les moindres carrés ordinaires (MCO) impossible[1]. Deuxièmement, l'accessibilité des algorithmes d'apprentissage automatique hautement complexes augmente le risque d'une utilisation naïve et d'une interprétation erronée de leurs résultats (Mullainathan et Spiess, 2017).

Les méthodes d'apprentissage automatique s'avèrent particulièrement utiles lorsque nous n'avons pas beaucoup d'information sur la forme et la complexité du vrai modèle. Pour illustrer cette situation, prenons y_{t+h}, la variable macroéconomique, à prévoir *h* périodes en avance. L'indice *t* représente la fréquence d'observation, qui peut être mensuelle ou trimestrielle. L'ensemble d'information disponible en période *t* est donné par X_t, un vecteur contenant *N* prédicteurs. Dénotons par $g^*(X_t)$ le vrai modèle (inconnu par l'utilisateur) et par $g(X_t;\theta)$ la forme fonctionnelle spécifiée par l'économètre. À noter que la fonction $g(\cdot)$ peut être paramétrique ou non, et que le vecteur θ contient tous les paramètres du modèle. De plus, définissons le modèle estimé et sa prévision par $\hat{g}(X_t;\hat{\theta})$ et $\hat{y}_{t+h}$ respectivement, où $\hat{\theta}$ est l'estimateur de θ. Par conséquent, l'erreur de prévision se décompose comme suit :

$$y_{t+h} - \hat{y}_{t+h} = \underbrace{g^*(X_t) - g(X_t;\theta)}_{erreur\ d'approximation} + \underbrace{g(X_t;\theta) - \hat{g}\left(X_t;\hat{\theta}\right)}_{erreur\ d'estimation} + \varepsilon_{t+h} \quad (1).$$

L'erreur intrinsèque à la prévision, ε_{t+h}, n'est pas réductible, c'est-à-dire qu'elle est due aux mouvements stochastiques imprévisibles. L'erreur d'estimation peut être contrôlée par l'économètre en ajoutant plus d'observations et en choisissant des estimateurs plus efficaces. La contribution des méthodes de ML se manifeste dans la réduction de l'erreur d'approximation en permettant des formes fonctionnelles $g(\cdot)$ très flexibles. Par contre, cette flexibilité vient habituellement avec le risque élevé de surajustement[2] et une régularisation performante doit accompagner le choix de $g(\cdot)$.

L'impact du choix d'une fonction $g(\cdot)$ peut être illustré à travers le cadre d'analyse suivant, où le choix des valeurs des paramètres θ vise à minimiser la fonction de coût *L* définissant la prédiction optimale (Hastie, Tibshirani et Friedman, 2017, p. 168) :

$$\min_{g \in G} \left\{ L\left(y_{t+h}, g(X_t;\theta)\right) + pen(g;\tau) \right\} \quad (2).$$

Ce cadre d'analyse est constitué de quatre caractéristiques importantes :

- *G* : l'espace de fonctions possibles $g(\cdot)$ combinant les données pour construire la prévision. En particulier, $g(\cdot)$ peut être linéaire ou non linéaire, paramétrique ou non paramétrique ;
- $pen(\cdot)$: la pénalité associée à la fonction $g(\cdot)$. Plusieurs méthodes nécessitent une forme de régularisation pour contrôler le risque de surajustement dû à la prolifération des paramètres et/ou aux formes complexes de nonlinéarité. En général, il s'agit de la pénalité de type Bridge ou de la réduction de dimension par composantes principales (modèle à facteurs) ;
- τ : les formes spécifiques de la fonction $g(\cdot)$ et de la pénalité $pen(\cdot)$ sont déterminées par un ensemble d'hyperparamètres τ. Le problème consiste à choisir une méthode efficace pour fixer τ.
- L : la fonction de coût qui définit la prévision optimale. La majorité du temps, elle est quadratique, mais certains modèles d'apprentissage automatique utilisent d'autres formes.

L'intelligence artificielle (IA) est constituée de combinaisons de ces ingrédients, qui composent les algorithmes d'apprentissage automatique, où chacun résout un problème de prévision spécifique.

Puisque la formulation (2) est très générale et abstraite, voyons sous quelles conditions nous pouvons obtenir un simple modèle prédictif. Supposons que *L* est quadratique, et que, par conséquent, la prévision optimale est l'espérance conditionnelle $E(y_{t+h}|X_t)$. Supposons que $g(\cdot)$ est une fonction linéaire et paramétrique, de sorte que $E(y_{t+h}|X_t)$ soit approximée par $X_t\theta$. Si le nombre de coefficients dans θ n'est pas très grand, la pénalité peut être ignorée et le modèle devient simplement $y_{t+h} = X_t\theta + e_{t+h}$. Les coefficients sont ensuite estimés efficacement (sous conditions additionnelles) par MCO et la prévision est obtenue par $\hat{y}_{t+h|t} = X_t\hat{\theta}$. Si X_t contient les valeurs retardées de y_t, alors le modèle devient une autorégression $y_{t+h} = \rho(L)y_t + e_{t+h}$, où $\rho(L)$ est un polynôme d'ordre fini *p*. Manifestement, les coefficients autorégressifs s'estiment par MCO pour une valeur de *p* donnée. Donc, *p* est un hyperparamètre qui est habituellement déterminé par un critère d'information.

Dans ce qui suit, nous dressons le portrait des avancées récentes dans le domaine de la prévision macroéconomique. Les contributions importantes seront présentées à travers le prisme prédictif dans l'équation (2), en particulier du choix de la fonction $g(\cdot)$.

Revue des développements récents

C'est l'apprentissage statistique non supervisé qui a été le plus utilisé et étudié sur le plan de la prévision macroéconomique, et ce, depuis les travaux de Stock et Watson (2002a, b). Cette analyse, basée sur la réduction de dimension de l'espace des prédicteurs à l'aide du modèle à facteurs, a été proposée comme première solution au problème de malédiction de dimensionnalité[3]. En ce qui concerne la problématique présentée dans l'équation (2), la fonction $g(\cdot)$ est linéaire en prédicteurs et θ, la régularisation $pen(\cdot)$, est un modèle à facteurs, les hyperparamètres sont typiquement choisis par un critère d'information, et la fonction de coût est quadratique. Cette forme de régularisation est présentée dans la sous-section intitulée « Autorégressif augmenté par les indices de diffusion ».

Une autre solution au problème de dimensionnalité consiste à choisir une pénalité permettant une sélection de variables parmi tous les *N* éléments de X_t, tout en estimant le modèle de régression prédictive linéaire sous *L* quadratique. Les hyperparamètres sont souvent sélectionnés par la validation croisée. Cette famille de modèles apparaît sous l'appellation « Elastic net » et sera présentée en détail plus loin dans le chapitre. Plusieurs travaux ont comparé ces techniques dans différents contextes de prévision macroéconomique et, de manière générale, la modélisation factorielle est préférable. Les exercices récents les plus complets sont disponibles dans Giannone, Lenza et Primiceri (2018) et dans Kotchoni *et al.* (2019).

Les deux méthodes précédentes ont utilisé seulement un aspect des grands changements survenus depuis les années 2000, soit l'accessibilité à des données massives. La fonction $g(\cdot)$ est toujours supposée linéaire, tandis que les méthodes d'apprentissage automatique servent justement à réduire l'erreur d'approximation dans l'équation (1) en choisissant les formes fonctionnelles pour $g(\cdot)$ les plus flexibles. La littérature sur ces tentatives est beaucoup moins volumineuse et plus récente. Les réseaux de neurones (*neural networks*), l'approche d'apprentissage automatique la plus prometteuse dans plusieurs domaines d'application de l'IA, ont été peu utilisés pour la prévision macroéconomique, puisqu'ils nécessitent l'estimation d'une quantité phénoménale de (hyper) paramètres. De plus, les premières tentatives se sont avérées peu fructueuses (voir, par exemple, Swanson et White, 1997). Néanmoins, les réseaux de neurones ont servi à améliorer la prévision de l'inflation et du taux de chômage dans les travaux de Nakamura (2005), de Cook et Hall (2017) et de Joseph (2019), et celle des rendements d'actifs dans Gu, Kelly et Xiu (2019). Plusieurs formes de régularisation sont nécessaires, non seulement pour pénaliser l'estimation des paramètres du modèle, mais aussi pour l'optimisation numérique (Gu *et al.,* 2019).

Une autre méthode populaire pour choisir $g(\cdot)$ est celle des forêts aléatoires (*random forests*). Cette méthode consiste en une analyse non linéaire et non paramétrique basée sur la régression par arbre. Medeiros, Vasconcelos, Veiga et Zilberman (2019) ont utilisé cette approche pour prévoir l'inflation, tandis que Ng (2014) et Döpke, Fritsche et Pierdzioch (2017) modélisent les probabilités de récession. Goulet Coulombe (2020) propose, quant à lui, de modéliser les changements structurels à l'aide des forêts aléatoires. Comme c'est une méthode qui tend naturellement vers le surajustement, elle est accompagnée par plusieurs approches de régularisation.

Finalement, la régression par vecteurs supports (*support vector regression,* ou SVR) a été populaire surtout dans les cas de problèmes de classification, mais elle est maintenant de plus en plus utilisée pour la prévision des séries temporelles. La SVR utilise habituellement une fonction $g(\cdot)$ non linéaire et la pénalité de type Ridge. Mais la plus grande différence par rapport aux autres modèles est le changement de la fonction de perte *L,* qui est insensible aux erreurs de prévision dans un intervalle particulier. Sermpinis, Stasinakis, Theolatos et Karathanasopoulos (2014) et Joseph (2019) ont appliqué cette méthode à la prévision de l'inflation et du taux de chômage, tandis que Colombo et Pelagatti (2020) l'ont utilisée pour améliorer grandement la prédiction des taux de change.

La plupart de ces travaux ne comparent qu'un nombre restreint de modèles et considèrent la prévision d'un petit groupe de variables à peu d'horizons. Quelques comparaisons de modèles à plus grande échelle ont été réalisées (Kim et Swanson, 2018 ; Milunovich, 2019 ; Chen, Dunn, Hood, Driessen et Batch, 2019). Une autre exception est le cas de Goulet Coulombe et ses collaborateurs (2019), qui ont étudié l'importance des quatre propriétés énumérées dans (2) pour la prévision macroéconomique. Ils ont conclu que la non-linéarité de la fonction $g(\cdot)$, combinée avec la réduction de dimension par le modèle à facteurs, améliorent nettement la capacité des modèles à prévoir les fluctuations cycliques. Ces auteurs ont montré que la meilleure pratique de sélection des hyperparamètres est la validation croisée standard (de type K-fold), et la fonction de perte quadratique est préférée à celle utilisée par le modèle de régression par vecteurs supports.

Que peut faire l'apprentissage machine ?

Il est important de souligner que l'apprentissage machine est un outil de prévision, tel que nous l'avons décrit dans (2). Il sert à gérer les données massives et à approximer des modèles inconnus par des formes fonctionnelles très flexibles, tout en contrôlant le risque de surajustement grâce aux techniques de régularisation. C'est donc un puissant outil d'analyse de données qui ne demande pas beaucoup de connaissances *a priori* sur leur structure ni sur la forme du modèle.

Par contre, cette grande flexibilité ne permet pas (à ce jour) à l'apprentissage machine de reconnaître les mécanismes fondamentaux derrière les fluctuations macroéconomiques. Pour le faire, il sera nécessaire d'adapter le ML à l'analyse causale en incorporant des restrictions économiques et en permettant la modélisation en équilibre général. Des avancées en analyse causale ont été récemment réalisées en microéconomie par Chernozhukov et ses collaborateurs (2018) et par Athey, Tibshirani et Wager (2019). Il est donc évident que l'apprentissage automatique causal est la prochaine grande étape dans l'analyse empirique des phénomènes économiques.

Application à la prévision de l'activité économique

Afin d'illustrer les concepts plutôt abstraits discutés précédemment, nous allons appliquer la modélisation basée sur l'apprentissage automatique à la prévision de l'activité macroéconomique québécoise. Pour ce faire, il faut avant tout préciser la cible y_{t+h} et l'ensemble de prédicteurs X_t. Ensuite, il faut choisir les modèles de prévision et spécifier les paramètres de l'exercice permettant d'évaluer leurs capacités prédictives.

Variable d'intérêt

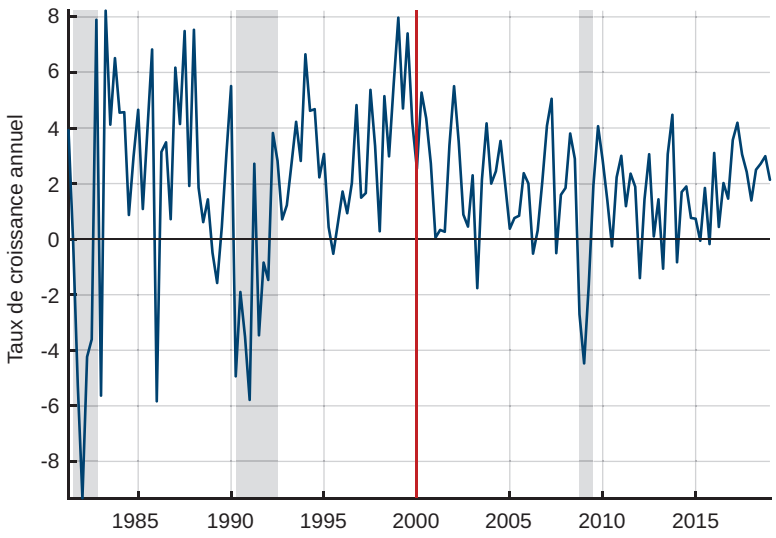
La variable d'intérêt sera simplement le plus important agrégat macroéconomique : le PIB. Avant de poursuivre, il faut déterminer plus précisément la cible. Soit Y_t, le PIB réel trimestriel désaisonnalisé. La cible est définie comme suit :

$$y_t = \log\left({Y_t}/{Y_{t-1}}\right).$$

Donc, l'objectif est de prévoir le taux de croissance *h* trimestres en avance, y_{t+h}. La série temporelle de la cible, en taux annualisé, est illustrée sur le graphique 12-1. Les zones ombragées représentent les périodes de récession au Canada, telles que les déclare l'Institut C.D. Howe[4]. On remarque que le PIB réel du Québec plonge durant les récessions, et que les expansions affichaient des taux de croissance plus élevés au début de l'échantillon par rapport à ce qui a été noté à compter de 2005. Depuis la dernière récession, la croissance stagne, mais il y a une tendance nette à la hausse à partir de 2015. Le taux de croissance du PIB est habituellement

difficile à prévoir, et c'est le modèle autorégressif simple qui est la référence, du moins pour le cas des États-Unis. Cet exercice sera aussi une occasion de documenter la prévisibilité du taux de croissance de l'économie québécoise[5].

Taux de croissance annualisé du PIB du Québec



Graphique g/2020-c**12-1**

Source : Calculs de l'auteur.

Ensemble de prédicteurs

Les données, pour cette application, proviennent de deux sources. Premièrement, les variables canadiennes sont disponibles dans la grande base de données construite par Fortin-Gagnon, Leroux, Stevanovic et Surprenant (2018) contenant 349 séries temporelles trimestrielles observées entre 1981Q1 et 2019Q1[6]. Différents secteurs d'activités et variables économiques sont représentés : production, marché immobilier, fabrication, ventes, agrégats monétaires, marché de crédit, taux d'intérêt, commerce international, prix et marché financier. La deuxième base de

données contient 248 variables macroéconomiques américaines et provient de McCracken et Ng (2016)[7]. La composition sectorielle est semblable à celle de la base de données canadienne et la même période est couverte.

Au total, en enlevant les doublons, il y a 590 candidats prédicteurs exogènes. En plus, lorsque quatre retards de la cible et des prédicteurs sont ajoutés dans l'ensemble d'informations, X_t contiendra jusqu'à 2 364 éléments. Il est donc évident que nous sommes dans le contexte de données massives et que la régularisation est nécessaire, puisque seulement 149 périodes temporelles sont disponibles. Certains modèles utiliseront les composantes principales extraites à partir des bases de données canadiennes et américaines séparément, dénotées par X_t^{CA} et X_t^{US} respectivement.

Exercice de prévision

La période de test, autrement dit hors échantillon (*pseudo-out-of-sample*) est 2000Q1 – 2019Q1 (voir la ligne rouge sur le graphique 12-1). Les horizons de prévision sont 1, 2, 4, 6 et 8 trimestres en avance. Donc, il y a 77 périodes d'évaluation pour chaque horizon. Les modèles sont estimés récursivement avec la fenêtre grandissante. Il est important de retenir ici l'esprit de l'exercice, tel qu'il est présenté à la figure 12-1. L'objectif est de vérifier la capacité des modèles à faire la prévision hors échantillon. En temps réel, le modèle est utilisé pour prévoir le futur, et l'erreur de prévision sera observée seulement une fois que la cible est réalisée. Pour approximer cette situation, l'échantillon est séparé en deux parties : période d'entraînement et période d'évaluation. Toutes les 77 cibles de la période d'évaluation sont prévues récursivement en ajoutant la dernière observation à la période d'entraînement lorsqu'elle devient disponible. Par exemple, pour h=1, la première cible est y_{2000Q1} étant donné l'information en 1999Q4. Un modèle est spécifié et estimé sur la période d'entraînement et, ensuite, la prévision $\hat{y}_{2000Q1|1999Q4}$ est produite. La prochaine cible est y_{2000Q2}, et l'ensemble d'information s'agrandit en ajoutant y_{2000Q1} dans la période d'entraînement puisque cette observation est maintenant supposée connue. Le modèle peut, en principe, être spécifié et estimé au cours de la nouvelle période d'entraînement, et la prévision $\hat{y}_{2000Q2|2000Q1}$ est obtenue. Ceci est répété jusqu'à la fin de l'échantillon, et ce, pour tous les modèles et pour tous les horizons de prévision.

Illustration de l'exercice d'évaluation

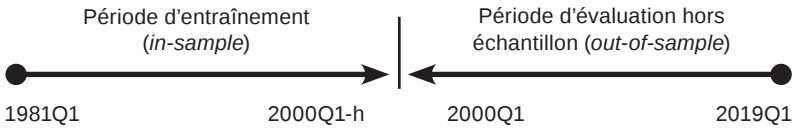


Figure f/2020-c**12-1**

Source : Auteur.

Comme résultat, on a une erreur de prévision pour chaque période *t*, horizon *h* et modèle *j* :

$$\hat{e}_{t,h,j} = y_{t+h} - \hat{y}_{t+h|t,j}$$

avec $t = 2000Q1, \ldots 2019Q1$, $h \in \{1,2,4,6,8\}$ et $j = 1, \ldots J$, où J, est le nombre de modèles.

Métriques d'évaluation prédictive

La métrique standard dans la littérature est l'erreur quadratique moyenne (EQM), qui est le choix logique lorsque la fonction de perte est quadratique :

$$EQM_{h,j} = \frac{1}{T}\sum_{t=1}^{T} \hat{e}_{t,h,j}^2.$$

Pour tester la précision prédictive entre deux modèles, la méthode de Diebold et Mariano (1995) est habituellement utilisée[8]. Finalement, il est intéressant de vérifier la capacité des modèles à bien prévoir la direction que prendra la cible dans l'avenir : est-ce que le PIB québécois augmentera ou diminuera dans h trimestres ? S'il peut être difficile d'obtenir une bonne performance en ce qui a trait à l'EQM, les modèles peuvent avoir une capacité significative à prévoir le signe de la cible (Satchell et Timmermann, 1995). Le test de signe prédictif est proposé par Pesaran et Timmermann (1992). La statistique du test s'écrit :

$$S_n = \frac{\hat{p} - \hat{p}^*}{\sqrt{Var(\hat{p}) - Var(\hat{p}^*)}},$$

où $\hat{p}$ est la proportion des prévisions ayant le bon signe et $\hat{p}^*$ est la valeur estimée de son espérance. Sous l'hypothèse nulle, le signe de la prévision est indépendant de celui de la cible, et on n'a que $S_n \to N(0,1)$. Autrement dit, si un modèle n'a pas de pouvoir prédictif sur la direction de la cible, alors son taux de succès sera proche de 50 %, comme si on lançait une pièce de monnaie.

Optimisation d'hyperparamètres

Avant de montrer formellement les modèles de prévision, il est opportun de détailler comment les hyperparamètres seront sélectionnés. Rappelons l'exemple du modèle autorégressif AR de l'introduction, $y_{t+h} = \rho(L)y_t + e_t$, où l'ordre $\rho(L)$, p, est le seul hyperparamètre. Pour fixer p, l'approche standard est le critère de sélection. Dans cette application, nous utiliserons le critère d'information bayésien (*Bayesian Information Criterion*, ou BIC) :

$$\log\left(\frac{SCR_{p_j}}{T}\right) + p_j \cdot \frac{\log(T)}{T} ,$$

où SCR_{p_j} est la somme des carrés des résidus et où p_j dénote le choix de l'ordre autorégressif. Comme le premier terme est décroissant en p_j, le deuxième permet au BIC de régulariser le surajustement en pénalisant avec le nombre de paramètres à estimer.

Pour illustration, supposons que la cible est y_{2000Q1}, soit un trimestre en avance. L'estimation du modèle AR requiert la sélection de son ordre p. La valeur estimée de p peut s'obtenir en appliquant le BIC sur *toute* la période d'entraînement de la figure 12-1 en testant de façon séquentielle tous les points de la grille $p_j = \{0,1, \ldots p_{max}\}$. Le point de la grille produisant la plus petite valeur du BIC sera l'estimation finale de l'ordre autorégressif. Il faut noter que cette procédure cache un hyperparamètre de second ordre, p_{max}[9].

Une autre façon de faire, surtout populaire en apprentissage automatique, est la validation croisée (*cross-validation,* ou CV). Tout comme le BIC, la CV choisit aussi l'ordre optimal p, mais en régularisant au moyen de la performance de prévision hors échantillon, tandis que la sélection par le BIC est basée uniquement sur la performance dans l'échantillon. La popularité de la CV tient aussi à sa simplicité, puisqu'elle peut être pratiquée même lorsque le critère d'information n'est pas disponible. Il existe plusieurs approches de CV, mais la plus populaire est basée sur un rééchantillonnage

aléatoire (K-fold) dans la période d'entraînement. Supposons que le nombre de plis (*fold*) est fixé à cinq (un autre hyperparamètre de second ordre). Ceci revient à séparer la période *in-sample,* de la figure 12-1, en cinq sous-échantillons de tailles égales. Ensuite, quatre sous-échantillons sont utilisés tour à tour pour estimer le modèle *j* de la grille présentée précédemment (formant la période d'entraînement, ou *training set,* en langage de ML), et un sous-échantillon sert à évaluer la performance hors échantillon avec, habituellement, l'EQM comme métrique. L'élément *j* de la grille produisant l'EQM minimale sera l'estimation de l'ordre optimal *p*.

Bien que la méthode K-fold soit très répandue dans les applications microéconomiques, elle paraît mal adaptée pour la prévision temporelle, comme dans cette application, et ce, pour deux raisons. Premièrement, l'ordre temporel n'est pas respecté lors du choix de *K* plis, et donc il est probable que les données futures soient utilisées pour prévoir le passé. Deuxièmement, les séries temporelles macroéconomiques sont souvent très persistantes, et K-fold brisera cette structure, impliquant un potentiel biais dans la sélection de *p.* Bergmeir, Hyndman et Koo (2018) montrent que la méthode K-fold CV peut être utilisée dans le cadre de l'autorégression tant que les résidus sont non corrélés. Sinon, la solution de rechange à la méthode K-fold est d'imiter l'exercice de prévision hors échantillon à l'intérieur même de la période d'entraînement (*pseudo-out-of-sample,* ou POOS CV). Ainsi, l'ordre temporel est respecté et la structure d'autocorrélation reste intacte. Par contre, le nombre d'observations pour évaluer la performance est grandement diminué relativement à la méthode K-fold, et ce, surtout en cas de petits échantillons dans les applications typiques en macroéconomie. Goulet Coulombe et ses collaborateurs (2019) suggèrent que, dans ces situations, la méthode K-fold est préférable justement à cause de la réduction de variance grâce à la plus grande taille des périodes d'évaluation. Ici, nous utilisons la méthode K-fold CV.

Modèles de prévision

Le modèle de référence est l'autorégressif direct (ARD) :

$$y_{t+h} = c + \rho(L)y_t + e_{t+h},$$

et l'ordre de $\rho(L)$ est choisi par le BIC. Un lecteur averti remarquera que la variable indépendante est retardée de *h* périodes par rapport à la cible, ce qui ne correspond pas au modèle autorégressif traditionnel. En fait, si *h=1,* les deux approches coïncident. Pour *h>1,* l'approche prédictive adoptée ici est dite *directe,* puisque la cible est projetée sur l'ensemble d'information et la prévision est faite directement à partir des observations les plus récentes. Ceci a comme avantage une plus grande robustesse aux changements structurels au coût de plus petite efficacité si le modèle est inutilement estimé pour chaque choix de *h* (Chevillon, 2007). Dans la majorité des modèles de ML, l'approche directe est la seule option.

Autorégressif augmenté par les indices de diffusion

Le premier modèle utilisant la régularisation comme dans l'équation (2) est l'autorégressif augmenté par les indices de diffusion (*Autoregressive Diffusion Indexes,* ou ARDI) de Stock et Watson (2002b) :

$$y_{t+h} = c + \rho(L)y_t + \beta(L)F_t + e_{t+h} \quad (3),$$

$$X_t = \Lambda F_t + u_t \quad (4),$$

où F_t contient *K* facteurs latents, et où $\rho(L)$ et $\beta(L)$ sont des polynômes d'ordre p^y et p^f respectivement. L'idée est d'estimer d'abord les facteurs par les composantes principales de X_t, pour ensuite les inclure comme prédicteurs dans l'équation (3). La réduction de dimension opère donc à l'équation (4), puisque l'ensemble des 590 prédicteurs contenus dans X_t sera réduit à un petit nombre de facteurs. Les hyperparamètres K, p^y et p^f seront déterminés par le BIC, bien que la validation croisée puisse aussi être utilisée.

Régressions pénalisées

La régularisation alternative est la régression linéaire pénalisée. Le cadre général est la régression Bridge, que voici :

$$\hat{\theta}_{Bridge} = arg\min_{\theta}(y_{t+h} - X_t\theta)^2 + \lambda\sum_{j=1}^{N}|\theta_j|^{\eta}, \ \ \eta > 0 \quad (5),$$

où $\lambda > 0$ est un hyperparamètre contrôlant la force de la régularisation. Deux cas spéciaux sont considérés. Si $\eta = 2$, l'estimateur Ridge (Hoerl, Kennard et Baldwin, 1975) est obtenu :

$$\hat{\theta}_{Ridge} = (X'X + \lambda I_N)^{-1} X'Y \qquad (6).$$

Dans le cas où $\eta = 1$, c'est l'estimateur Lasso (pour *Least Absolute Shrinkage Selection Operator,* de Tibshirani, 1996) qui est obtenu :

$$\hat{\theta}_{Lasso} = arg\min_{\theta}(y_{t+h} - X_t\theta)^2 + \lambda \sum_{j=1}^{N} |\theta_j| \qquad (7).$$

La particularité de Ridge est d'avoir une solution intérieure et analytique étant donné λ. Si $\lambda = 0$, (7) se réduit à l'estimateur MCO. Si $\lambda > 0$, Ridge diminuera la valeur des coefficients des variables moins importantes vers zéro. À l'opposé, Lasso, simultanément, estime la régression prédictive et sélectionne les variables importantes, puisque la norme L_1 est employée comme pénalité. Autrement dit, le vecteur de coefficients estimés, $\hat{\theta}_{Lasso}$, contiendra des zéros. Comme Lasso ne gère pas très bien les données corrélées, deux alternatives ont été proposées[10].

La première est Elastic net (EN, due à Zou et Hastie, 2004), qui combine Ridge et Lasso :

$$\hat{\theta}_{EN} = arg\min_{\theta}(y_{t+h} - X_t\theta)^2 + \lambda \sum_{j=1}^{N} \left(\alpha |\theta_j| + (1-\alpha)\theta_j^2\right) \qquad (8)$$

avec $\alpha = [0,1]$; fixer α à 1 ou à 0 génère Lasso ou Ridge, respectivement. La deuxième alternative est Adaptive Lasso (Zou, 2006) :

$$\hat{\theta}_{AdLasso} = arg\min_{\theta}(y_{t+h} - X_t\theta)^2 + \lambda \sum_{j=1}^{N} \psi_j |\theta_j|, \qquad (9),$$

où $\psi_j = \frac{1}{|\tilde{\theta}_j|^{\gamma}}$ sont les poids obtenus au préalable par un estimateur convergent $\tilde{\theta}_j$ et avec $\gamma > 0$.

Dans cette application, $\tilde{\theta}_j$ sont premièrement estimés par Ridge et $\gamma > 1$. À noter que γ peut être sélectionné par validation croisée.

La partie importante, dans ces quatre modèles, est l'optimisation de λ, puisque c'est la force de la régularisation. Ceci est habituellement effectué par validation croisée.

Régressions régularisées par sous-ensembles complets

La méthode par sous-ensembles complets (*complete subset regressions,* ou CSR) a été proposée par Elliott, Gargano et Timmermann (2013). C'est une solution au surajustement en présence d'un très grand nombre de prédicteurs. L'idée est de créer des prévisions à partir d'un grand nombre de petits modèles (en prenant des sous-ensembles de prédicteurs) potentiellement biaisés mais ayant une plus petite variance, et de construire la prévision finale comme la moyenne de ces prédictions. Formellement, soit $X_{t,m}$ un vecteur de *L* variables obtenues du tirage aléatoire $m = 1, \dots M$. Pour chaque tirage *m,* la prévision est construite à partir d'une régression prédictive :

$$\hat{y}_{t+h|t,m} = \hat{c} + \hat{\rho}(L)y_t + \hat{\beta}X_{t,m} \quad (10)$$

et la prévision finale est obtenue en prenant la moyenne sur les M modèles :

$$\hat{y}_{t+h|t} = \frac{1}{M}\sum_{m=1}^{M} \hat{y}_{t+h|t,m}. \qquad (11).$$

Le nombre de tirages devrait être grand pour approximer toutes les $N!/(L! \cdot (N-L)!)$ possibilités. Une amélioration de ce modèle, qui a été proposée par Kotchoni et ses collaborateurs (2019), sera utilisée dans ce chapitre. Il s'agit de deux versions régularisées de la méthode par CSR.

La première consiste à présélectionner un sous-ensemble de prédicteurs exogènes dans X_t avant de procéder à la méthode par CSR comme dans les équations (10) et (11). La présélection se fait par le seuillage (*hard thresholding*) univarié suivant :

$$y_{t+h} = c + \rho(L)y_t + \beta_i X_{i,t} + \epsilon_{t+h}, i = 1, \dots N \qquad (12),$$

où la cible est projetée sur son passé (d'ordre *p*) et sur un prédicteur exogène à la fois. Le sous-ensemble $X_t^* \in X_t$ consiste en des variables correspondantes aux coefficients β_i ayant la statistique t_{β_i} plus grande qu'une valeur critique : $t_c : X_t^* = \{X_{i,t} \in X_t \big| |t_{\beta_i}| > t_c\}$. Après la présélection par (12), la prévision est obtenue en appliquant (10) et (11) à l'ensemble X_t^*. À noter que ce processus est appliqué pour chaque horizon de prévision. La valeur critique t_c est un hyperparamètre qui joue le même rôle de la force de régularisation que λ dans une régression pénalisée. Ce modèle sera nommé T-CSR.

La deuxième variante régularise la méthode par CSR quant aux modèles aléatoires (10). Au lieu d'estimer les coefficients par MCO, ils sont plutôt estimés par Ridge, comme dans l'équation (6). De cette façon, si $X_{t,m}$ contient plusieurs prédicteurs très corrélés ou des éléments non pertinents, la pénalité Ridge en tiendra compte. Pour rendre la procédure numériquement faisable, l'hyperparamètre de Ridge est sélectionné comme dans Hoerl, Kennard et Baldwin (1975). Ce modèle sera nommé CSR-R.

Plusieurs hyperparamètres doivent être fixés dans ces deux modèles. Le nombre de prédicteurs dans chaque tirage, *L,* sera fixé à 10 et à 20. Le nombre de sous-modèles $M = 2000$ et la valeur critique t_c est fixée à 1,65. Quatre retards de la cible sont utilisés dans l'équation (12).

Forêts aléatoires

Le premier modèle utilisant une approximation non linéaire est celui des forêts aléatoires (*random forests,* de Breiman, 2001), qui est basé sur les régressions par arbre. La figure 12-2 montre un exemple simple pour illustrer ce concept. Supposons deux prédicteurs pour y_{t+h} : taux de chômage $(X_{1,t})$ et taux d'intérêt $(X_{2,t})$. Premièrement, les observations sont triées par $X_{1,t}$. Celles qui se trouvent sous le seuil de 7 % sont affectées au nœud 1. Celles qui sont associées aux plus grandes valeurs du taux de chômage sont ensuite triées par $X_{2,t}$. Les observations qui satisfont $X_{1,t} > 7\ \%\ , X_{2,t} < 5\ \%$ remplissent le nœud 2, tandis que les cas $X_{1,t} > 7\ \%\ , X_{2,t} > 5\ \%$ se retrouvent dans le nœud 3.

Taux de croissance annualisé du PIB du Québec

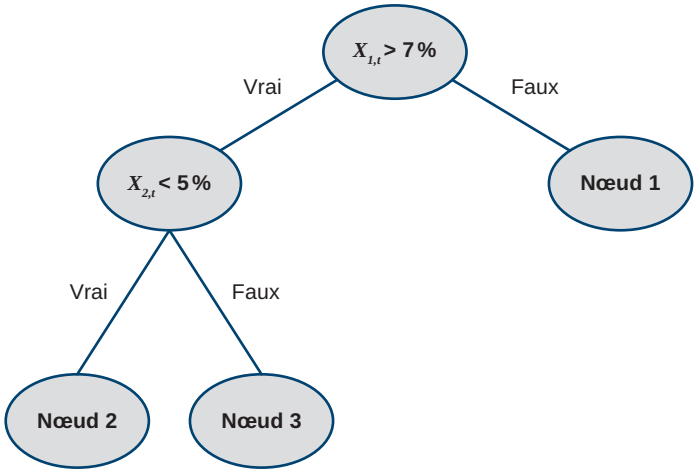


Figure f/2020-c**12-2**

Source : Auteur.

Enfin, les prévisions des observations dans chaque partition sont définies comme la moyenne de la variable dépendante parmi les observations de cette partition. Formellement, la prévision d'un arbre avec B nœuds terminaux (partitions) et de profondeur *L* s'écrit :

$$g(X_t) = \sum_{b=1}^{B} c_b 1_{\{X_t \in P_b(L)\}}$$

où chaque partition est le produit de *L* fonctions indicatrices des prédicteurs. La constante associée au nœud *b*, B, est la moyenne de la cible pour les observations incluses dans ce nœud. L'exemple de la figure 12-2 peut aussi s'écrire comme une régression avec des variables binaires :

$$g(X_t) = c_1 1_{\{X_{1,t}<7\%\}} + c_2 1_{\{X_{1,t}>7\%\}} 1_{\{X_{2,t}<5\%\}} + c_3 1_{\{X_{1,t}>7\%\}} 1_{\{X_{2,t}>5\%\}}.$$

Les prédicteurs et les seuils ont ici été choisis pour illustrer la situation où la prévision d'une cible macroéconomique dépend de l'état du cycle (la probabilité de récession est plus élevée si le taux de chômage est supérieur

à 7 %) et de la position de la banque centrale (taux d'intérêt). Le choix des prédicteurs et des seuils est plutôt déterminé en optimisant une métrique d'évaluation de la prévision.

Il ressort de cet exemple que la régression par arbres peut approximer le vrai modèle $g^*(X_t)$ grâce à sa flexibilité et à un grand nombre d'interactions possibles entre les prédicteurs. Par contre, cette flexibilité mène souvent au surajustement, et l'estimation non paramétrique requiert beaucoup d'observations.

La méthode souvent utilisée pour contrôler le surajustement consiste à combiner les prévisions provenant de beaucoup d'arbres qui sont créés de façon aléatoire (d'où le nom de « forêt aléatoire »). Dans cette application, le modèle procède en créant beaucoup d'arbres à partir des sous-échantillons aléatoires des observations. Par contre, cette astuce ne peut suffire si les prédicteurs sont très corrélés. Alors, une autre couche de randomisation est effectuée. Celle-ci consiste à choisir au hasard, pour chaque arbre, un sous-ensemble de M_{try} prédicteurs pour créer les branches. La prédiction finale est la moyenne des prévisions de tous les arbres aléatoires. Plusieurs hyperparamètres sont à spécifier. Le nombre d'arbres est généralement grand ; par exemple, dans cette application, il est de 1 000. M_{try} est fixé à $N/3$, mais il pourrait aussi être déterminé par CV. Le nombre minimal d'observations dans chaque nœud terminal est fixé à cinq. La profondeur de l'arbre (*L*) et le nombre de nœuds terminaux (*B*) sont ainsi implicitement fonction de ces hyperparamètres.

Régression par vecteurs supports

La particularité du modèle de SVR réside dans le changement de la fonction de perte minimisée lors de l'estimation. Formellement, le problème primal d'un ϵ-SVR est donné par

$$\min_{\gamma} \frac{1}{2}\gamma'\gamma + C\left(\sum_{t=1}^{T}(\xi_t + \xi_t^*)\right)$$

$$\begin{cases} y_{t+h} - \gamma'\varphi(X_t) - c \leq \bar{\varepsilon} + \xi_t \\ \gamma'\varphi(X_t) + c - y_{t+h} \leq \bar{\varepsilon} + \xi_t^* \\ \xi_t, \xi_t^* \geq 0 \end{cases}$$

où $\varphi(\cdot)$ est une fonction de base implicitement définie par le noyau choisi, $(C, \bar{\varepsilon})$ sont des hyperparamètres et (ξ_t, ξ_t^*) sont des variables d'écarts (*slack variables*). Soient (λ_j, λ_j^*), les multiplicateurs de Lagrange associés aux deux premières contraintes présentées plus haut. Les poids optimaux sont $\hat{\gamma} = \sum_{j=1}^{T}(\lambda_j - \lambda_j^*)\varphi(X_j)$ et la prévision s'obtient comme suit :

$$\hat{y}_{t+h|t} = \hat{c} + \sum_{j=1}^{T}(\lambda_j - \lambda_j^*)\varphi(X_j)\varphi(X_t) = \hat{c} + \sum_{j=1}^{T} K(X_j, X_t) \quad (13),$$

où $K(\cdot)$ est une fonction noyau (*kernel function*)[11]. Notez que le noyau gaussien (*radial basis function kernel*) sera utilisé afin d'introduire des non-linéarités :

$$K(X_j, X_t) := exp\left(-\frac{\|X_j - X_t\|^2}{2\sigma^2}\right),$$

où σ^2 est un hyperparamètre.

Discutons maintenant de la fonction de perte particulière employée par les SVR. Le long des valeurs prédites dans l'échantillon d'estimation, il y a une ø ne d'insensibilité dont la largeur est donnée par l'hyperparamètre $\bar{\epsilon} > 0$. Formellement, la fonction de perte s'écrit comme suit :

$$L(e_{t+h}) = \begin{cases} 0 & si\ |e_{t+h}| \leq \bar{\epsilon} \\ |e_{t+h}| - \bar{\epsilon} & sinon \end{cases}.$$

La figure 12-3 compare les fonctions de perte quadratique et $\bar{\epsilon}$-insensible. La zone d'insensibilité est déterminée par l'hyperparamètre $\bar{\epsilon}$, tandis que le coût à l'extérieur de cette zone est fonction des variables d'écart.

C'est cette insensibilité à une zone autour de la valeur prévue que Smola et Schölkopf (2004) appellent la parcimonie des SVR. En effet, seulement une partie des observations contribue concrètement à la valeur prédite[12].

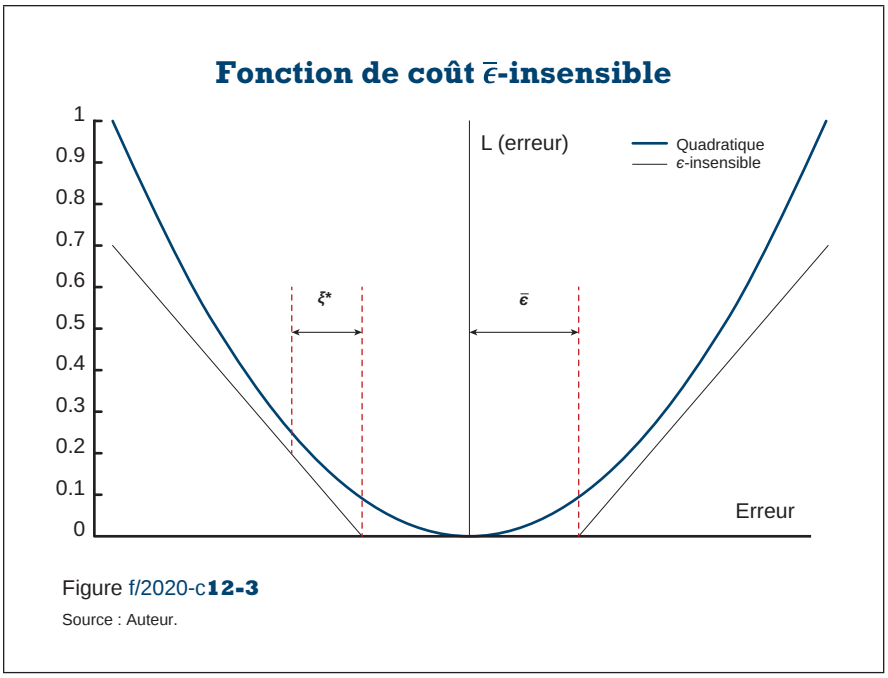


Figure f/2020-c**12-3**

Source : Auteur.

Réseaux de neurones

La dernière classe de modèles considérée est celle des réseaux de neurones (*neural networks,* ou NN). Comme nous l'avons mentionné dans l'introduction, les premières tentatives de prévision macroéconomique avec ces modèles ont été peu fructueuses, probablement en raison de leur structure jadis très simple et du manque de puissance computationnelle et de données (Swanson et White, 1997). De nos jours, ce sont les modèles les plus prometteurs dans la plupart des domaines d'application d'apprentissage automatique. Leur utilisation en prévision macroéconomique est encore très sommaire en raison du manque d'observations, puisque ce sont des structures extrêmement paramétrées (des milliers de paramètres à estimer et beaucoup d'hyperparamètres à choisir) et ayant une fonction objective non convexe et habituellement non linéaire.

Inspirés du fonctionnement du cerveau humain, les réseaux de neurones sont en fait une succession de couches cachées contenant chacune un certain nombre de neurones. Dans le cas qui nous intéresse, comme le montre la figure 12-4, les neurones entre les couches sont tous interreliés.

Exemple d'un réseau de neurones

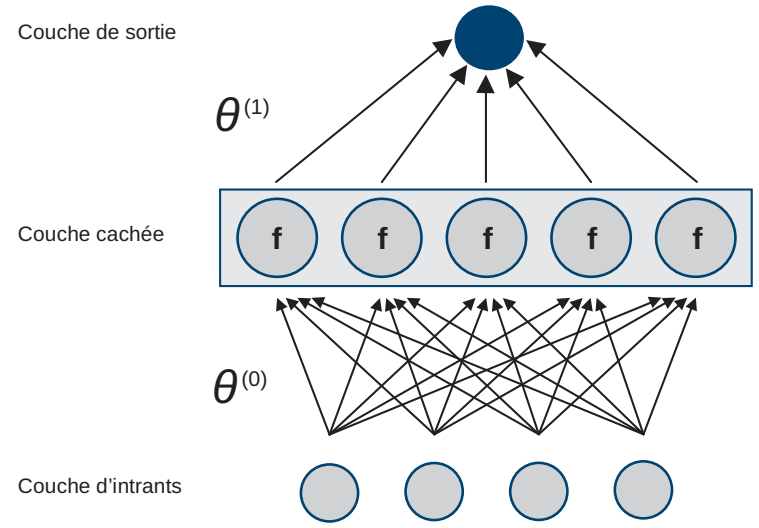


Figure f/2020-c**12-4**

Source : Auteur.

Chaque θ représente le poids accordé aux extrants (*output*) de la couche précédente. Ce type de réseaux se nomme « perceptron multicouche » (*multilayer perceptron,* ou MLP). Il consiste en une couche d'intrants suivie par une ou plusieurs couches cachées qui transforment les combinaisons linéaires des différents intrants à l'aide d'une fonction d'activation f (souvent non linéaire), puis se termine avec une couche de sortie qui agrège l'information obtenue à partir des couches précédentes. Par exemple, dans le cas de la figure 12-4 le NN est représenté par une couche de quatre intrants tous connectés aux cinq neurones de la couche cachée, puis se termine par la mise en commun de l'information dans la couche de sortie.

Les réseaux de neurones apprennent à l'aide d'un algorithme de rétro-propagation. Cet algorithme permet d'effectuer une mise à jour des poids et des biais du modèle selon les erreurs de prévision obtenues à la couche finale. Lors de l'initialisation, les poids des modèles sont établis de manière aléatoire et $\hat{y}_{t+h|t}$ est prédit à l'aide de ces poids de départ. Les poids de la couche de sortie sont d'abord mis à jour selon l'erreur de prévision obtenue, puis à leur tour les poids de l'avant-dernière couche, et ainsi de suite jusqu'au poids de la couche d'intrants. Un cycle de la couche de sortie à la couche d'intrant est appelé cycle d'apprentissage (*epoch*). Le nombre de cycles d'apprentissage constitue un hyperparamètre du réseau, qui sera fixé à un maximum de 1 000. Cependant, un critère d'arrêt basé sur la valeur de l'erreur quadratique moyenne sera utilisé afin d'éviter les problèmes de surapprentissage.

La fonction d'activation est un morceau important de l'architecture des NN. Dans cette application, la fonction ReLU (*Rectified Linear Unit*) est utilisée, soit $\sigma(x) = \max(0, x)$. Celle-ci permet la réduction du nombre de neurones actifs et permet donc une estimation plus rapide (Gu *et al.,* 2019). Pour écrire formellement la prévision à l'aide du réseau de neurones, soit $M^{(l)}$, le nombre de neurones dans chaque couche $l = 1, \dots L$, et $z_m^{(l)}$, l'extrant de chaque neurone m dans la couche l, avec $z^{(l)} = \left(1, z_1^{(l)}, \dots, z_{N^{(l)}}^{(l)}\right)$ La couche d'intrants est initialisée avec les observations. L'extrant de couche cachée $l > 0$ est :

$$z^{(l)} = max\left(0, z^{(l-1)'}\theta^{(l-1)}\right),$$

et la prévision est obtenue par la couche finale :

$$y_{t+h|t} = z^{(L-1)'}\theta^{(L-1)}.$$

Le nombre de couches cachées et de neurones représente des hyperparamètres importants des NN. Le nombre de paramètres à estimer augmente très rapidement avec ceux-ci, ce qui allonge considérablement les temps de calcul. Basée sur les travaux de Gu et ses collaborateurs (2019), une architecture en « pyramide » est souvent privilégiée. Ici, un maximum de 3 couches cachées et de 32 neurones sera considéré. Pour chaque couche cachée, le nombre de neurones est obtenu en divisant par 2 les nombres de la couche précédente. Par exemple, un réseau avec 2 couches cachées contiendra respectivement 32 et 16 neurones.

Dans cette application, le choix de la combinaison optimale des hyperparamètres est effectué par validation croisée, et ce, une seule fois pour chaque horizon au cours de la période allant de 1981Q1 à 2000Q1-*h.* Une pénalisation de type Lasso sur la valeur des poids θ est utilisée afin d'éviter les problèmes liés au surajustement. La valeur optimale de λ est sélectionnée par CV.

Les réseaux de neurones récurrents (*recurrent neural network,* ou RNN) sont également considérés. Ces réseaux peuvent conserver de l'information en mémoire. De cette manière, les RNN peuvent profiter de la structure temporelle des données, ce qui les rend différents des réseaux de neurones « *feed forward* » comme le MLP. En plus de tenir compte des intrants, il prend également ses décisions sur un certain nombre d'états cachés (*hidden states*). La prévision est ainsi calculée séquentiellement pour chaque valeur retardée de l'intrant. Ainsi, pour chaque retard, cet intrant intermédiaire sert d'intrant supplémentaire pour la prévision d'états cachés. Le modèle RNN contient donc autant d'états cachés (*hidden states*) qu'il y a de retards dans les intrants.

En particulier, un réseau de neurones récurrents à mémoire à court terme et à long terme (Long Short-Term Memory LSTM) est utilisé (Hochreiter et Schmidhuber, 1997). Ce qui est intéressant avec l'utilisation de cellules LSTM, comparativement aux RNN traditionnels, c'est qu'elles peuvent décider de garder ou non certaines informations passées grâce à des portes (*gates*) (Chung, Gulcehre, Cho et Bengio, 2014). La cellule LSTM contient essentiellement trois portes : *input gate, output gate* et *forget gate.* Ces portes peuvent apprendre durant l'entraînement quelles sont les informations provenant des états précédents qui sont pertinentes et les conserver, en oubliant les autres. L'*input gate,* de son côté, fait la mise à jour de l'état actuel de la cellule en y ajoutant de la nouvelle information. La *forget gate* décide de ce qui est à conserver des états précédents. Puis, l'*output gate* renvoie le prochain état, qui est la combinaison de la nouvelle information et de l'état précédent. Ce cycle est répété de manière récursive autant de fois qu'il y a de retards dans l'intrant. Comme chez Cook et Hall (2017), une combinaison des deux types de réseaux est également considérée. Ainsi, la première couche cachée est un réseau de type LSTM, puis, les autres couches cachées sont d'architecture MLP. Les hyperparamètres sont aussi les mêmes que pour les autres modèles.

D'un point de vue computationnel, l'algorithme d'apprentissage utilisé est Adam (pour *adaptive moment estimation algorithm*), avec le taux d'apprentissage fixé à 0,001 et la patience à 5.

Plus formellement, il y a trois types de réseaux de neurones qui contiennent chacun la même architecture « pyramidale ». Il y a les réseaux MLP (Dense), qui sont les réseaux multicouches interconnectés décrits plus haut. Puis, il y a les réseaux récurrents (LSTM), qui tiennent compte de l'aspect temporel des données. Ceux-ci sont constitués de couches cachées LSTM placées une à la suite de l'autre. Cependant, ces modèles se terminent par une couche de sortie interconnectée afin de convertir l'extrant dans le bon format. Finalement, le dernier type de réseaux considéré est une combinaison des deux précédents (LSTM-Dense). Ainsi, ces réseaux sont d'abord constitués d'une première couche cachée de type LSTM, puis les suivantes sont des couches cachées et interconnectées comme celles des MLP.

Combinaisons de prévisions

Avec tous ces modèles, il est aussi naturel de les combiner. La littérature en prévision macroéconomique constate depuis longtemps que l'agrégation des prédictions individuelles est une méthode très robuste (Bates et Granger, 1969 ; Hendry et Clements, 2004). Il existe, bien sûr, de nombreuses façons de combiner les prévisions, et plusieurs seront considérées ici.

La simple moyenne (*equal-weighted forecasts,* ou AVRG) est la méthode la plus utilisée ; les prévisions de J modèles, $j = 1, \dots, J$, sont combinées à l'aide de poids égaux $\omega_{jt} = \frac{1}{J}$:

$$\hat{y}_{t+h|t} = \frac{1}{J}\sum_{j=1}^{J} \hat{y}_{t+h|t}^{(j)}.$$

La moyenne est toutefois sensible aux valeurs extrêmes, c'est pourquoi plusieurs solutions de rechange sont proposées. La première (*trimmed average,* ou T-AVRG) enlève les prévisions les plus extrêmes en ordonnançant les valeurs des prédictions en ordre croissant $\left(\hat{y}_{t+h|t}^{(1)} \leq \hat{y}_{t+h|t}^{(2)} \cdots \leq \hat{y}_{t+h|t}^{(J)}\right)$. Ensuite, une proportion ϑ est supprimée des deux côtés :

$$\hat{y}_{t+h|t} = \frac{1}{J(1-2\vartheta)} \sum_{j=\lceil \vartheta J \rceil}^{\lfloor (1-\vartheta) J \rfloor} \hat{y}_{t+h|t}^{(j)},$$

où $\lceil \vartheta J \rceil$ est le nombre entier immédiatement supérieur à ϑJ et $\lfloor (1-\vartheta) J \rfloor$ est le nombre entier immédiatement inférieur à $(1-\vartheta)J$. ϑ est un hyperparamètre et sera fixé à 15 %.

Une solution plus flexible consiste à produire les poids qui dépendent de la performance historique des modèles (Diebold et Pauly, 1987). Ici, l'approche de Stock et Watson (2004) est utilisée (*inversely proportional average,* ou IP-AVRG). Le poids sur la j^e prévision est :

$$\omega_{jt} = \frac{m_{jt}^{-1}}{\sum_{j=1}^{J} m_{jt}^{-1}},$$

où m_{jt} est l'erreur quadratique moyenne escomptée du modèle j :

$$m_{jt} = \sum_{s=T_0}^{t-h} \rho^{t-h-s} \left(y_{s+h} - y_{s+h|s}^{(j)} \right)^2,$$

et où ρ est le facteur d'escompte (hyperparamètre). Ici, deux valeurs seront considérées : $\rho = 1$ et $\rho = 0{,}8$.

Finalement, la simple médiane peut aussi être utilisée :

$$\hat{y}_{t+h|t} = médiane \left(y_{t+h|t}^{(j)} \right)_{j=1}^{J}.$$

Le tableau 12-1 résume tous les modèles utilisés dans cet exercice en spécifiant l'ensemble de prédicteurs utilisés. Les modèles utilisant les facteurs indicés par ARDI contiennent trois composantes principales extraites de chacune des bases de données X_t^{CA} et X_t^{US} séparément. Les hyperparamètres dans les modèles AR,BIC et ARDI,BIC sont sélectionnés par le BIC, tandis que pour tous les autres modèles, ils sont choisis par validation croisée.

Liste des modèles de prévision			
Modèle	**Prédicteurs**	**Modèle**	**Prédicteurs**
AR,BIC	$(1 + L + \cdots L^{p_y})y_t$	CSR-R,20	$(1 + L + \cdots L^4)y_t, X_t$
ARDI,BIC	$(1 + L + \cdots L^{p_y})y_t, (1 + L + \cdots L^{p_f})F_t$	LSTM-Dense-AR	$(1 + L + \cdots L^4)y_t$
ARDI,Lasso	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)F_t$	LSTM-AR	$(1 + L + \cdots L^4)y_t$
ARDI,Ridge	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)F_t$	Dense-AR	$(1 + L + \cdots L^4)y_t$
ARDI,EN	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)F_t$	LSTM-Dense-ARDI	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)F_t$
ARDI, ALasso	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)F_t$	LSTM-ARDI	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)F_t$
Lasso	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)X_t$	Dense-ARDI	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)F_t$
Ridge	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)X_t$	LSTM-Dense-X	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)X_t$
Elastic net	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)X_t$	LSTM-X	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)X_t$
Adaptative Lasso	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)X_t$	Dense-X	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)X_t$
RF-ARDI	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)F_t$	AVRG	
RF-X	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)X_t$	Médiane	
SVR-ARDI	$(1 + L + \cdots L^4)y_t, (1 + L + \cdots L^4)F_t$	T-AVRG,0.1	
T-CSR,10	$(1 + L + \cdots L^4)y_t, X_t$	T-AVRG,0.2	
T-CSR,20	$(1 + L + \cdots L^4)y_t, X_t$	IP-AVRG,1	
CSR-R,10	$(1 + L + \cdots L^4)y_t, X_t$	IP-AVRG,0.8	

Tableau t/2020-c**12-1**

Source : Auteur.

Résultats

Le tableau 12-2 montre la performance prédictive des modèles présentés ci-dessus se rapportant à la racine carrée de l'erreur quadratique moyenne (*root mean squared error,* ou RMSE). Pour simplifier la lecture, la RMSE de chaque modèle compétitif est divisée par celle du modèle de référence, AR,BIC. Les astérisques indiquent la significativité du test d'équivalence prédictive Diebold-Mariano. Lorsque le ratio est plus petit que 1, cela indique que le modèle fait mieux que la référence autorégressive. On remarque que plusieurs modèles font mieux que le modèle AR,BIC aux horizons d'un et deux trimestres en avance. Le meilleur candidat pour $h = 1$ est le modèle CSR-R, avec 10 modèles (voir la sous-section intitulée « Régressions régularisées par sous-ensembles complets »), suivi de près par Ridge (voir la sous-section intitulée « Régressions pénalisées », équation [6]), par CSR-R,20 et par les combinaisons de prévision. CSR-R,10 améliore la RMSE par rapport au modèle de référence de 13 %, tandis que les autres solutions de rechange sont aussi au-dessus de 10 %. La grande majorité des autres modèles fait mieux que le modèle AR,BIC, mais les améliorations sont plus modestes, quoique significatives.

Dans le cas de deux trimestres en avance, le meilleur modèle est LSTM-X, un réseau récurrent utilisant toutes les observables comme entrants. Il améliore la prévision du taux de croissance du PIB de 15 % par rapport à la référence. Il est suivi de près par LSTM-ARDI, dont la seule différence est de considérer les facteurs au lieu des observables. LSTM-AR affiche une bonne performance aussi, ce qui suggère que d'utiliser les couches LSTM est une meilleure idée que de se servir des couches standards MLP (Dense). CSR-R,10 est encore très résilient, ce qui indique que la moyenne sur beaucoup de prévisions est également une façon intéressante de prédire la cible.

À partir d'un an en avance, le pouvoir prédictif relatif au modèle de référence s'estompe. Les meilleurs modèles n'améliorent que de 5 % la prévisibilité et très peu de modèles affichent des performances significativement différentes. Parmi ces modèles, on trouve surtout les réseaux de neurones, en particulier LSTM-ARDI et Dense-ARDI. Ceci fait écho aux résultats obtenus par Goulet Coulombe et ses collaborateurs (2019), qui ont remarqué qu'augmenter le modèle à facteurs par la non-linéarité devenait plus important à long terme.

En résumé, la plupart des modèles font mieux que le modèle de base à court terme (un et deux trimestres en avance), tandis que leur performance pâlit à plus long terme. Les groupes de modèles les plus résilients d'un horizon à l'autre sont les régressions pénalisées par sous-ensembles complets (CSR-R) et les réseaux de neurones, en particulier LSTM-ARDI. Ceci suggère que la combinaison des prévisions et la non-linéarité sont les ingrédients clés dans la prévision du taux de croissance du PIB québécois.

La preuve précédente montre la performance moyenne sur toute la période de test. Or, l'environnement macroéconomique peut varier considérablement dans le temps, et Giacomini et Rossi (2010) ont adapté le test de Diebold-Mariano afin de comparer la performance de deux modèles en présence d'instabilité structurelle. La figure 12-5 montre les résultats pour quatre groupes de modèles : CSR régularisés, réseaux de neurones, forêts aléatoires et SVR, ainsi que pour les régressions pénalisées Elastic net. La statistique de test est construite récursivement comme une moyenne mobile sur 30 % de la période hors échantillon et, donc, les résultats commencent en 2006. Si la statistique est positive, cela signifie que le modèle fait mieux que le modèle AR,BIC. Les lignes horizontales représentent les valeurs critiques pour un niveau de 10 %. On remarque que les modèles CSR-R font significativement mieux autour de 2015, et entre 2010 et 2014, aux horizons d'un et deux trimestres respectivement. Les régressions pénalisées Ridge et Adaptive Lasso affichent des résultats semblables. Pour sa part, le modèle des forêts aléatoires réussit bien par rapport à tous les aspects observables seulement en 2015. Les réseaux de neurones montrent une performance plutôt stable. Un fait remarquable est la baisse commune de la performance de tous les modèles depuis 2016.

Bien que le tableau 12-2 et la figure 12-5 montrent que les modèles de ML améliorent significativement la prédiction du taux de croissance du PIB par rapport au modèle autorégressif standard, ceci ne dit rien sur leur capacité à prévoir cette variable, puisque le modèle AR,BIC peut être simplement incapable de bien prédire la cible. Une façon de vérifier la prévisibilité consiste à comparer les modèles selon le pseudo-R^2, qui est simplement :

$$pseudo - R^2_{h,j} = 1 - \frac{\sum \hat{e}^2_{t,h,j}}{\sum (y_t - \bar{y})^2} \qquad (14),$$

où $\bar{y}$ est la moyenne empirique de la cible jusqu'à $t-h$. Donc, ceci est proportionnel au ratio de l'EQM du modèle j, et l'EQM du modèle prédictif ne contenant que la constante, $y_{t+h} = c + e_{t+h}$. Galbraith (2003) suggère le pseudo-R^2 comme une mesure de prévisibilité d'une variable : si la forme fonctionnelle et l'ensemble d'information, $g(X_t;\theta)$, n'apportent pas de pouvoir prédictif en ce qui a trait à l'EQM, c'est-à-dire le pseudo-$R^2 < 0$, alors la variable est peu prévisible. Le tableau 12-3 présente les résultats. La situation est plutôt négative pour la plupart des modèles puisqu'il est très rare qu'une spécification améliore la prévision par rapport à la moyenne historique. Les modèles résilients présentés précédemment affichent quand même les pseudo-R^2 positifs. Ce constat n'est toutefois ni rare ni décourageant. Les PIB américain et canadien sont également difficiles à prévoir et la moyenne historique peut être un modèle raisonnable dans la mesure où la croissance de l'activité économique est stable à travers le temps. Les modèles alternatifs auront, dans ce cas, une utilité prédictive, surtout lors des grands changements, tels que les récessions et les reprises.

Une autre mesure de performance a été présentée dans la section intitulée « Métriques d'évaluation prédictive » : la capacité à prédire le changement de la cible. Dans le cas du PIB québécois, on vient de constater que la prévisibilité en ce qui concerne l'erreur quadratique moyenne est plutôt faible. Les modèles ont-ils une capacité significative à prévoir le signe de la direction que prendra le PIB à l'avenir ? Les résultats sont présentés dans le tableau 12-4. La proportion de signes correctement prédits est très élevée dans tous les cas. ARDI,Lasso domine pour les courts horiø ns tandis que les versions non linéaires du modèle à facteurs, les forêts aléatoires et les régressions par vecteurs supports, affichent la meilleure performance pour les quatre et les six trimestres en avance. Il faut cependant remarquer ici que le nombre de périodes hors échantillon étant petit, peu de cas sont significatifs et les ratios sont plutôt semblables d'un modèle à un autre.

Performance en termes de l'EQM REQM

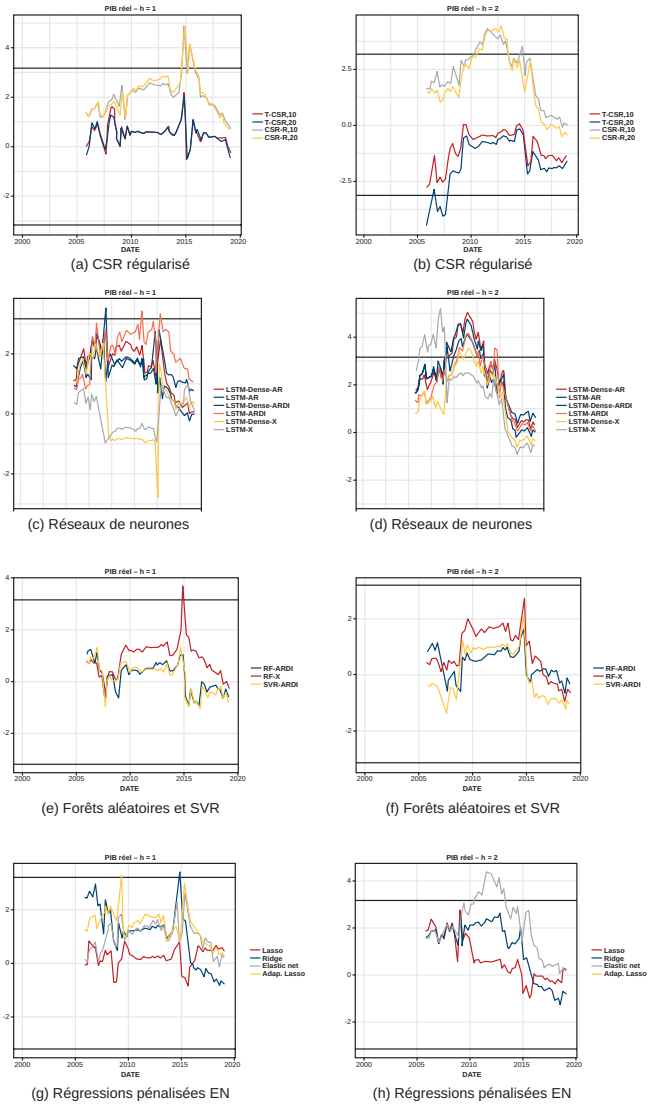
	h=1	h=2	h=4	h=6	h=8
AR,BIC (RMSE)	0.01	0.01	0.00	0.00	0.00
ARDI,BIC	0.92	1.02	1.15*	1.15*	1.06**
ARDI,Lasso	0.95*	0.97	1.06**	1.06	1.01
ARDI,Ridge	0.95**	0.91***	0.97	0.99	0.99
ARDI,EN	0.93**	0.91***	1.00	0.99	0.99
ARDI,ALasso	0.92***	0.91***	1.00	0.99	0.99
Lasso	0.97	0.93	1.09	1.00	0.98
Ridge	0.87**	0.91**	1.03	1.09	1.08
Elastic net	0.95**	0.91***	1.05**	1.01	1.00
Adap. Lasso	0.92***	0.91***	1.00	0.99	0.99
RF-ARDI	0.94	0.92	1.07	1.08*	1.02
RF-X	0.92**	0.94*	1.01	1.00	1.00
SVR-ARDI	0.96	0.96	1.00	1.07	1.03
T-CSR,10	0.95	1.09**	1.02	1.13*	1.04
T-CSR,20	0.97	1.20***	1.07	1.27**	1.08**
CSR-R,10	0.87***	0.89***	1.01	0.98	1.00
CSR-R,20	0.88***	0.91***	1.02	1.00	0.99
LSTM-Dense-AR	0.92***	0.90***	0.99	0.98	0.98
LSTM-AR	0.91***	0.89***	1.00	0.97	0.98
Dense-AR	0.94**	0.90***	1.03*	0.99	0.97*
LSTM-Dense-ARDI	0.90**	0.90***	0.98	0.98	0.99
LSTM-ARDI	0.90***	0.87***	0.95	0.97	0.95*
Dense-ARDI	0.98	1.12	1.26**	0.97*	1.04
LSTM-Dense-X	1.12	0.92**	1.03	0.99	0.97
LSTM-X	1.02	0.85***	1.00	0.98	0.98
Dense-X	0.91***	0.92**	1.00	0.97*	0.97*
AVRG	0.89**	0.92**	0.98	1.00	0.98
Médiane	0.90**	0.91***	0.98	1.00	0.99
T-AVRG,0.1	0.89**	0.92***	0.98	1.00	0.98
T-AVRG,0.2	0.90***	0.91***	0.98	1.00	0.98
IP-AVRG,1	0.89**	0.92***	0.98	1.00	0.98
IP-AVRG,0.8	0.89***	0.92***	0.98	1.00	0.98

Tableau t/2020-c**12-2**

Source : Auteur.

Note : La première ligne contient la racine carrée de l'erreur quadratique moyenne (REQM) du modèle de référence AR,BIC. Les autres éléments sont les ratios de REQM de chacun des modèles alternatifs sur celui du modèle de référence. La valeur minimale pour chaque horizon est soulignée. Les astérisques indiquent la significativité du test Diebold-Mariano par rapport au modèle de base avec les niveaux de 1 %, de 5 % et de 10 %, respectivement représentés par *, ** et ***.

Performance à travers le temps



(a) CSR régularisé (b) CSR régularisé

(c) Réseaux de neurones (d) Réseaux de neurones

(e) Forêts aléatoires et SVR (f) Forêts aléatoires et SVR

(g) Régressions pénalisées EN (h) Régressions pénalisées EN

Figure f/2020-c**12-5**

Source : Auteur.

Note : La figure montre les résultats du *Fluctuation test de* Giacomini et Rossi (2010).

Pseudo-R^2

	h=1	h=2	h=4	h=6	h=8
AR,BIC	-0.21***	-0.28***	-0.04*	-0.07**	-0.04
ARDI,BIC	-0.02	-0.34**	-0.39*	-0.42*	-0.17**
ARDI,Lasso	-0.08	-0.20***	-0.18***	-0.21**	-0.05**
ARDI,Ridge	-0.08*	-0.07**	0.03	-0.05**	-0.03***
ARDI,EN	-0.05	-0.07*	-0.05*	-0.05**	-0.02***
ARDI,ALasso	-0.03*	-0.05*	-0.05*	-0.05**	-0.02***
Lasso	-0.14	-0.11	-0.25	-0.06	0.00
Ridge	0.09	-0.05	-0.11	-0.28*	-0.22*
Elastic net	-0.08*	-0.05*	-0.15***	-0.09***	-0.04
Adap. Lasso	-0.03*	-0.05*	-0.05*	-0.05**	-0.02***
RF-ARDI	-0.08	-0.08	-0.18	-0.26**	-0.09
RF-X	-0.02	-0.13*	-0.05	-0.08**	-0.03
SVR-ARDI	-0.11	-0.19	-0.04	-0.23	-0.11*
T-CSR,10	-0.10	-0.52***	-0.09	-0.38**	-0.12*
T-CSR,20	-0.13	-0.84***	-0.20	-0.73***	-0.21**
CSR-R,10	0.09	-0.02	-0.06**	-0.03**	-0.04*
CSR-R,20	0.07	-0.05	-0.08**	-0.06**	-0.02
LSTM-Dense-AR	-0.03*	-0.03	-0.02	-0.02	0.01
LSTM-AR	-0.01	-0.02	-0.05**	-0.02	0.00
Dense-AR	-0.08*	-0.03	-0.11**	-0.04	0.03
LSTM-Dense-ARDI	0.01	-0.04	-0.01	-0.03*	-0.02
LSTM-ARDI	0.02	0.02	0.07	-0.02	0.06
Dense-ARDI	-0.16*	-0.60*	-0.65**	0.00	-0.12**
LSTM-Dense-X	-0.51	-0.08*	-0.10*	-0.05	0.02
LSTM-X	-0.27*	0.07	-0.04	-0.02	0.00
Dense-X	0.01	-0.07**	-0.03	0.00	0.02
AVRG	0.05	-0.09**	-0.01	-0.08	0.00
Médiane	0.03	-0.05*	0.01	-0.06**	-0.02*
T-AVRG,0.1	0.04	-0.07*	-0.01	-0.07*	-0.01
T-AVRG,0.2	0.03	-0.06*	0.00	-0.07*	-0.01
IP-AVRG,1	0.04	-0.08**	-0.01	-0.08*	0.00
IP-AVRG,0.8	0.04	-0.08**	-0.01	-0.08*	0.00

Tableau t/2020-c**12-3**

Source : Auteur.

Note : Les entrées dans ce tableau représentent le pseudo-R^2 tel qu'il est présenté dans l'équation (14). La valeur maximale pour chaque horizon est soulignée. Les astérisques indiquent la significativité du test Diebold-Mariano avec les niveaux de 1 %, de 5 % et de 10 %, respectivement représentés par *, ** et ***.

Prévision de la direction du PIB

	h=1	h=2	h=4	h=6	h=8
AR,BIC	84.42	83.12	84.42	84.42	84.42
ARDI,BIC	83.12	81.82	84.42**	84.42	84.42
ARDI,Lasso	85.71**	84.42	83.12	84.42	84.42
ARDI,Ridge	84.42	84.42	84.42	84.42	84.42
ARDI,EN	84.42	84.42	84.42	84.42	84.42
ARDI,ALasso	84.42	84.42	84.42	84.42	84.42
Lasso	84.42**	81.82	81.82	83.12	84.42
Ridge	85.71**	83.12	83.12	81.82	81.82
Elastic net	84.42	84.42	84.42	83.12	84.42
Adap. Lasso	84.42	84.42	84.42	84.42	84.42
RF-ARDI	84.42	84.42	83.12	85.71**	84.42
RF-X	84.42	84.42	84.42	84.42	84.42
SVR-ARDI	83.12	83.12	85.71**	83.12	84.42
T-CSR,10	85.71**	80.52	84.42	85.71**	84.42
T-CSR,20	85.71**	79.22	83.12	80.52	84.42**
CSR-R,10	84.42	84.42	84.42	84.42	84.42
CSR-R,20	84.42	84.42	84.42	84.42	84.42
LSTM-Dense-AR	84.42	84.42	84.42	84.42	84.42
LSTM-AR	84.42	84.42	84.42	84.42	84.42
Dense-AR	84.42	84.42	84.42	84.42	84.42
LSTM-Dense-ARDI	84.42	84.42	84.42	84.42	84.42
LSTM-ARDI	84.42	84.42	84.42	84.42	84.42
Dense-ARDI	84.42	83.12	79.22	84.42	84.42
LSTM-Dense-X	84.42	84.42	81.82	83.12	84.42
LSTM-X	84.42	84.42	84.42	84.42	84.42
Dense-X	84.42	84.42	84.42	84.42	84.42
AVRG	85.71**	84.42	84.42	84.42	84.42
Médiane	84.42	84.42	84.42	84.42	84.42
T-AVRG,0.1	85.71**	84.42	84.42	84.42	84.42
T-AVRG,0.2	85.71**	84.42	84.42	84.42	84.42
IP-AVRG,1	85.71**	84.42	84.42	84.42	84.42
IP-AVRG,0.8	84.42	84.42	84.42	84.42	84.42

Tableau t/2020-c**12-4**

Source : Auteur.

Note : Les entrées dans ce tableau représentent les ratios de succès (*success ratios*) de la prévision du signe de la cible, tel qu'ils ont été décrits dans la section intitulée « Métriques d'évaluation prédictive ». La valeur minimale pour chaque horizon est soulignée. Les astérisques indiquent la significativité du test de Pesaran et Timmermann (1992) avec les niveaux de 1 % , de 5 % et de 10 %, respectivement représentés par *, ** et ***.

Références

Athey, S., Tibshirani, J. et Wager, S. (2019). Generalized random forests. *The Annals of Statistics, 47*(2), 1148-1178.

Bates, J. et Granger, C. W. J. (1969). The Combination of Forecasts. *Operational Research Society, 20*(4), 451-468.

Bergmeir, C., Hyndman, R. J. et Koo, B. (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics and Data Analysis, 120,* 70-83.

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

Chen, J., Dunn, A., Hood, K., Driessen, A. et Batch, A. (2019). *Off to the Races: A comparison of machine learning and alternative data for predicting economic indicators.* Rapport technique, Bureau of Economic Analysis.

Chernozhukov, V., Chetverikov, D., Demirer, M., Duflo, E., Hansen, C., Newey, W. et Robins, J. (2018), Double/debiased machine learning for treatment and structural parameters. *The Econometrics ð urnal, 21,* C1-C68.

Chevillon, G. (2007). Direct multi-step estimation and forecasting. *Journal of Economic Surveys, 21*(4), 746-785.

Chung, J., Gulcehre, C., Cho, K., et Bengio, Y. (2014). Empirical evaluation of gated recurrent neural networks on sequence modeling. arXiv preprint arXiv:1412.3555

Colombo, E. et Pelagatti, M. (2020). Statistical learning and exchange rate forecasting. *International ð urnal of Forecasting, 36*(4), 1260-1289.

Cook, T. R. et Hall, A. S. (2017). *Macroeconomic indicator forecasting with deep neural networks.* Rapport technique, Federal Reserve Bank of Kansas City.

Diebold, F. X. et Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business and Economic Statistics, 13,* 253-263.

Diebold, F.X. et Pauly, P. (1987). Structural change and the combination of forecasts. *ð urnal of Forecasting, 6,* 21-40.

Döpke, J., Fritsche, U. et Pierdzioch, C. (2017). Predicting recessions with boosted regression trees. *International ð urnal of Forecasting, 33*(4), 745-759.

Elliott, G., Gargano, A. et Timmermann, A. (2013). Complete subset regressions. *Journal of Econometrics, 177*(2), 357-373.

Fortin-Gagnon, O., Leroux, M., Stevanovic, D. et Surprenant, S. (2018). *A large Canadian database for macroeconomic analysis.* CIRANO Working Papers, 2018s-25.

Galbraith, J. (2003). Content horizons for univariate time series forecasts. *International ð urnal of Forecasting, 19*(1), 43-55.

Giacomini, R. et Rossi, B. (2010). Forecast comparisons in unstable environments. *ð urnal of Applied Econometrics, 25*(4), 595-620.

Giannone, D., Lena , M. et Primiceri, G. E. (2018). *Economic Predictions with Big Data: The illusion of sparsity.* Rapport technique, Federal Reserve Bank of New York.

Goulet Coulombe, P., Leroux, M., Stevanovic, D. et Surprenant, S. (2019). How is machine learning useful for macroeconomic forecasting. CIRANO Working Papers, 2019s-22.

Goulet Coulombe, P. (2020). *The Macroeconomy as a Random Forest.* Mimeo. University of Pennsylvania.

Gu, S., Kelly, B. et Xiu, D. (2019). *Empirical asset pricing via machine learning.* Rapport technique, Chicago Booth Research Paper No. 18-04; *The Review of Financial Studies, 33,* 2223-2273.

Hansen, P., Lunde, A. et Nason, J. (2011). The model confidence set. *Econometrica, 79*(2), 453-497.

Hastie, T., Tibshirani, R. et Friedman, J. (2017). *The Elements of Statistical Learning.* Springer Series in Statistics. Springer-Verlag, New York.

Hendry, D. F. et Clements, M. P. (2004). Pooling of forecasts. *The Econometrics ð urnal, 7*(1), 1-31.

Hochreiter, S. et Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*(8), 1735-1780.

Hoerl, A. E., Kennard, R. W. et Baldwin, K. F. (1975). Ridge regression: Some simulations. *Communications in Statistics, 4*(2), 105-123.

Joseph, A. (2019). *Shapley Regressions: A framework for statistical inference on machine learning models.* Rapport technique, document de travail, Bank of England, 784.

Kim, H. H. et Swanson, N. R. (2018). Mining big data using parsimonious factor, machine learning, variable selection and shrinkage methods. *International Journal of Forecasting, 34*(2), 339-354.

Kotchoni, R., Leroux, M., et Stevanovic, D. (2019). Macroeconomic forecast accuracy in a data-rich environment. *ð urnal of Applied Econometrics, 34,* 1050-1072.

McCracken, M. W. et Ng, S. (2016). FRED-MD: A monthly database for macroeconomic research. *ð urnal of Business and Economic Statistics, 34*(4), 574-589.

Medeiros, M. C., Vasconcelos, G. F. R., Veiga, A. et Zilberman, E. (2019). Forecasting inflation in a data-rich environment: The benefits of machine learning methods. *ð urnal of Business and Economic Statistics.* doi : 10.1080/07350015.2019.1637745

Milunovich, G. (2019). *Forecasting Australian Real House Price Index: A comparison of time series and machine learning methods.* Rapport technique, Maquarie University.

Mullainathan, S. et Spiess, J. (2017). Machine learning: An applied econometric approach. *ð urnal of Economic Perspectives, 31*(2), 87-106.

Nakamura, E. (2005). Inflation forecasting using a neural network. *Economics Letters, 86*(3), 373-378.

Ng, S. (2014). Boosting recessions. *Canadian ð urnal of Economics, 47*(1), 1-34.

Pesaran, H. et Timmermann, A. (1992). A simple nonparametric test of predictive performance. *ð urnal of Business and Economic Statistics, 10*(4), 461-465.

Satchell, S. et Timmermann, A. (1995). An assessment of the economic value of non-linear foreign exchange rate forecasts. *ð urnal of Forecasting, 14*(6), 477-497.

Schwert, G. W. (1989). Tests for unit roots: A Monte Carlo investigation. *ð urnal of Business and Economic Statistics, 7,* 147-159.

Sermpinis, G., Stasinakis, C., Theolatos, K. et Karathanasopoulos, A. (2014). Inflation and unemployment forecasting with genetic support vector regression. *Journal of Forecasting, 33*(6), 471-487.

Smola, A. J. et Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and Computing, 14*(3), 199-222.

Stock, J. H. et Watson, M. W. (1999). A comparison of linear and nonlinear univariate models for forecasting macroeconomic time series (p. 1-44). Dans R. F. Engle et H. White (dir.), *Cointegration, Causality and Forecasting: A festschrift in Honour of Clive W. J. Granger.* Oxford : Oxford University Press.

Stock, J. H. et Watson, M. W. (2002a). Forecasting using principal components from a large number of predictors. *ð urnal of the American Statistical Association, 97*(460), 1167-1179.

Stock, J. H. et Watson, M. W. (2002b). Macroeconomic forecasting using diffusion indexes. *ð urnal of Business and Economic Statistics, 20*(2), 147-162.

Stock, J. H. et Watson, M. W. (2004). Combination forecasts of output growth in a seven-country data set. *ð urnal of Forecasting, 23*(6), 405-430.

Swanson, N. et White, H. (1997). A model selection approach to real-time macroeconomic forecasting using linear models and artificial neural networks. *The Review of Economics and Statistics, 79*(4), 540-550.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *ð urnal of the Royal Statistical Society. Series B, 58*(1), 267-288.

Teräsvirta, T. (2006). Forecasting economic variables with nonlinear models (p. 413-457). Dans G. Elliott, C. Granger et A. Timmermann (dir.), *Handbook of Economic Forecasting,* vol. 1, Elsevier.

Zou, H. (2006). The adaptive lasso and its oracle properties. *ð urnal of the American Statistical Association, 101*(476), 1418-1429.

Zou, H. et Hastie, T. (2004). Regularization and variable selection via the elastic net. *ð urnal of the Royal Statistical Society. Series B, 67*(2), 301-320.

Notes

1. Ce problème est connu dans la littérature sous le nom de « malédiction de la dimensionnalité ».
2. Le surajustement (*overfitting*) survient lorsqu'une analyse statistique confond la variation résiduelle avec celle du modèle sous-jacent et, par conséquent, reproduit complètement (ou presque) la structure des données, mais produit une mauvaise performance prévisionnelle hors échantillon.
3. Le problème de malédiction de dimensionnalité (*curse of dimensionality*) est causé par un trop grand nombre de variables explicatives par rapport au nombre d'observations temporelles, rendant ainsi l'estimation par moindres carrés impossible.
4. https://www.cdhowe.org/turning-points-business-cycles-in-canada-since-1926/19364
5. La ligne rouge sépare l'échantillon en deux. La période à partir de 2000 sera hors échantillon dans l'exercice de prévision.
6. Les données canadiennes se trouvent à cette adresse : http://www.stevanovic.uqam.ca/DS_LCMD.html. La version LCDMA_Q_November_2019 a été utilisée dans ce chapitre.
7. Les données américaines se trouvent à cette adresse : https://research.stlouisfed.org/econ/mccracken/fred-databases/. La version 2019-12 a été utilisée ici.
8. En présence de nombreux modèles, la méthode Model Confidence Set (MCS), de Hansen, Lunde et Nason (2011), aurait pu être appliquée afin de sélectionner un groupe de modèles plus performants. Par contre, le nombre d'observations dans la période de test étant relativement petit, cette méthode n'a pas été utilisée.
9. À titre d'exemple, dans le cas de sélection de l'ordre autorégressif du test ADF par le BIC, Schwert (1989) suggère, pour les échantillons de petite taille, la borne supérieure $p_{max} = \left[12 \cdot \left(\frac{T}{100}\right)^{1/4}\right]$
10. En présence des variables corrélées, Lasso a tendance à écarter les moins importantes, impliquant une sélection de variables non convergente. En fait, la régularisation Lasso joue sur l'arbitrage entre biais et variance, et peut donc induire un biais au profit de la diminution de la variance. C'est l'essence du contrôle du risque de surajustement.
11. Voir Smola et Schölkopf (2004) pour la dérivation des résultats brièvement montrés ici.
12. À noter que le tube peut être assez large pour exclure toutes les observations, auquel cas le modèle de SVR prévoit la moyenne historique.