CEWP 21-05

Forecasting Canadian GDP Growth with Machine Learning

Shafiullah Qureshi
Carleton University

Ba Chu
Carleton University

Fanny S. Demers
Carleton University

May 17, 2021

CARLETON ECONOMICS WORKING PAPERS

Carleton
UNIVERSITY

Department of Economics

1125 Colonel By Drive
Ottawa, Ontario, Canada
K1S 5B6

Forecasting Canadian GDP Growth with Machine Learning

Shafiullah Qureshi[*] Ba Chu[†] Fanny S. Demers[‡]

May 17, 2021

Abstract

This paper applies state-of-the-art machine learning (ML) algorithms to forecast monthly real GDP growth in Canada by using both Google Trends (GT) data and official macroeconomic data (which are available ahead of the release of GDP data by Statistics Canada). We show that we can forecast real GDP growth accurately ahead of the release of GDP figures by using GT and official data (such as employment) as predictors. We first pre-select features by applying up-to-date techniques, namely, *XGBoost*'s variable importance score, and a recent variable-screening procedure for time series data, namely, *PDC-SIS+*. These pre-selected features are then used to build advanced ML models for forecasting real GDP growth, by employing tree-based ensemble algorithms, such as XGBoost, LightGBM, Random Forest, and GBM. We provide empirical evidence that the variables pre-selected by either PDC-SIS+ or the XGBoost's variable importance score can have a superior forecasting ability. We find that the pre-selected GT data features perform as well as the pre-selected official data features with respect to short-term forecasting ability, while the pre-selected official data features are superior with respect to long-term forecasting ability.

We also find that *(1)* the ML algorithms we employ often perform better with a large sample than with a small sample, even when the small sample has a larger set of predictors; and *(2)* the Random Forest (that often produces nonlinear models to capture nonlinear patterns in the data) tends to under-perform a standard autoregressive model in several cases while there is no clear evidence that the XGBoost and the LightGBM can always outperform each other.

[*] Department of Economics, Carleton University, 1125 Colonel By Dr., Ottawa, Ontario, Canada; and Department of Economics, Sector H-9, Islamabad, Pakistan. Email: suqureshi@numl.edu.pk. Tel: +1.613.520.2600 (ext. 3778).

[†] Department of Economics, Carleton University, 1125 Colonel By Dr., Ottawa, Ontario, Canada. Email: ba.chu@carleton.ca. Tel: +1 613-520-2600 (ext. 1546).

[‡] Department of Economics, Carleton University, 1125 Colonel By Dr., Ottawa, Ontario, Canada. Email: fanny.demers@carleton.ca. Tel: +1 613-520-2600 (ext. 3775).

1 Introduction

Gross domestic product (GDP) is the primary measure for assessing the performance of an economy. It enables policymakers to judge whether the economy is expanding or contracting, and permits them to make appropriate monetary or fiscal policy decisions accordingly. In this respect, the accurate and timely forecast of GDP growth ahead of the release of official GDP figures is vital. Both quarterly and monthly GDP data are published with a two-month delay by Statistics Canada due to multiple revisions. Yet, some other official macroeconomic information, such as employment, housing starts, and retail trades, often produced by national statistical bureaus, central banks, or other government institutions are available much sooner in some cases, with only a few days' delay. In addition, the use of Google Trend (GT) data for economic forecasting has become popular since the seminal work of Choi and Varian (2009) and Choi and Varian (2012). GT data has been exploited to predict retail sales, automotive sales, home sales, and travel volume, among many others. GT allows users to download the time series data for a keyword search of particular interest in index form by selecting a specific geographic region and a specific time. Notably, Tkacz (2013) used GT data to predict the 2008-09 recession in Canada with a Probit model, and observed an enormous surge in the search frequency of the word 'recession' about one month before the actual date of the 2008-09 recession. In the context of GDP forecasting, Götz and Knetsch (2019) used GT and other survey data to forecast German GDP with the Bridge regression.[1] They obtained good forecasts when GT data was used instead of survey data (suggesting that GT data can be a good substitute for survey data when the latter are not available), but the forecast improvement tended to be smaller when both GT and survey data were used together. Ferrara and Simoni (2019) also used a Bridge regression to nowcast Euro-area GDP with GT data. They employed Fan and Lv's (2008) Sure Independence Screening (SIS) to pre-select GT features as the first step. A Ridge regression was then used in the second step to construct forecasts given the pre-selected features. They then evaluated their forecasts using various combinations of data sources, and found GT data to be a useful predictor only before official data becomes available.

In this paper, we apply state-of-the-art machine learning (ML) algorithms to 'nowcast' monthly and quarterly Canadian GDP growth by using both Google Trends (GT) and official macroeconomic data that are available ahead of Statistics Canada's release of GDP data. We hereafter refer to the latter as Official data. (See Table 3 for a list of the variables we use and their release dates). We show that we can forecast real GDP growth accurately ahead of the release of GDP figures by using GT data and Official data variables as predictors. In particular, we investigate whether GT data are still useful for forecasting GDP in the presence of Official variables. As mentioned above, most previous works on forecasting with GT data found this dataset to be useful when no Official data are available, and that its importance diminishes as soon as Official data become available (as in e.g., Ferrara and Simoni (2019) and Götz and Knetsch (2019)). We find that the forecasting accuracy of GT data alone is quite comparable to that of using both GT and Official data for short term forecasting (such as one- or three-steps ahead forecasts), but that Official variables yield somewhat greater forecasting accuracy for longer term forecasting (such as six-steps ahead forecasts). Hence, our results indicate that GT data can be very useful for forecasting GDP growth in the short term.

Work on forecasting with ML has recently been growing. It would be impossible to provide a complete list here. We shall unfortunately be able to discuss only a few, certainly nonrepresentative, contributions that are closely related to the topic of the present paper. Biau and D'Elia (2012) used the Random Forest (RF) algorithm to predict Euro-area GDP, and found that a combination of the RF and a linear model can outperform the benchmark autoregressive model. Other authors, such as Tiffin (2016) and Jung, Patnam, and Ter-Martirosyan (2018), have applied the Elastic Net, the SuperLearner, and the Deep learners algorithms to produce forecasts that can outperform those made by traditional statistical methods. A comprehensive survey of applications of ML methods for macroeconomic forecasting is also provided by Fuleky (2019). Recently, Medeiros, Vasconcelos, Veiga, and Zilberman (2021) extensively applied standard ML methods to forecast the U.S. inflation rate, and find that the Random Forest (RF) algorithm can deliver the best forecasts among all the ML algorithms that they considered. With respect to fore-

[1] The Bridge regression has been used by policymakers to make short-term forecasts since at least the 1980s. Bridge models are linear dynamic models that use short-term indicators (available at a higher frequency) to forecast low-frequency variables such as aggregate GDP or its components.

casts of Canadian GDP, Tkacz (2001) adopted a neural network approach, while Chernis and Sekkel (2017) used a dynamic-factors model. As the success of a supervised ML method depends on its capability to capture nonlinearities in the relationship between the response and predictive variables as well as in the (nonlinear) interactions between two predictive variables, in this study we attempt to forecast Canadian GDP growth rate with (nonlinear) tree-based ensemble algorithms that have not been fully explored for economic forecast, namely, Gradient Boosting Machine (GBM), eXtreme Gradient Boosting (XGBoost), and Microsoft's Light Gradient Boosting Machine (LightGBM).[2] Such ensemble methods have proved their empirical success in many other applications because averaging several nonlinear models with possibly different sets of predictive variables is perhaps the best way to cope with model uncertainty which is very common in practice [see, e.g., Athey, Bayati, Imbens, and Qu (2019)].

To implement the aforementioned ML algorithms, we used the automated machine learning (AutoML) – an application interface which automates the process of simultaneously training a large number of models using the GBM, the XGBoost, the Generalized Linear Model (GLM), the Distributed Random Forest (DRF), Deep Learning algorithms, and Stacked Ensembles.[3] Our proposed forecasting procedure is a three-step procedure detailed in Section 3. In the first step, we pre-select the most relevant GT features and Official variables through the *variable importance measure* of XGBoost, which has recently become a very popular and powerful algorithm in applied ML and Kaggle competitions.[4] It should be noted that variable pre-selection in our procedure plays a crucial role in improving forecast performance. Therefore, we rigorously train and validate several thousand models in order to achieve optimal forecast accuracy. In the second step, the selected variables are then introduced as inputs to train and validate the RF regression and boosted tree models. In the last step, the trained models are used to make out-of-sample forecasts.

With respect to forecast evaluation, a common tradition is to calculate the root mean squared error (RMSE), the mean absolute error (MAE), or R^2 to evaluate and rank forecasts. While these metrics are useful in differentiating among the forecasting accuracy of various models, using only RMSE, MAE or R^2 does not always provide an accurate assessment of the forecasting ability of an individual model against the actual target variable. Thus, as recently emphasized by Brown and Hardy (2021), who refer to this phenomenon as "the mean squared prediction error paradox," a model may yield a poor prediction of the target variable even if its RMSE is low. They propose the correlation among variables as a superior assessment method, and show analytically, empirically and graphically that a forecast with the lowest RMSE may have the worst correlation with the target variable, and thus be a poor predictor. Only under certain regularity conditions, these two metrics coincide. In this paper, we provide a graphical representation of the actual versus predicted trajectories to give a clear picture of how well our predicted GDP growth rate (with both GT and Official variables) tracks actual GDP growth. We have also calculated Hansen, Lunde, and Nason's (2011) model confidence set (MCS) *p-values* of all models considered. This MCS procedure exploits all information available in the data to decide if a model belongs to the set of the best models by implementing a sequence of statistical tests to verify the null hypothesis of equal predictive ability.

The rest of this paper is outlined as follows: Section 2 provides a brief description of the ML algorithms employed in this paper. Although this description may appear non-technical, interested readers can find a rigorous treatment of these algorithms in many good ML textbooks, for example, Hastie, Tibshirani, and Friedman (2009) and Murphy (2012). Section 3 explains our three-step forecasting procedure. Section 4 presents GT and Official data that we used to predict GDP growth rate. Section 5 provides the main empirical findings of this paper. Section 6 concludes this paper. Finally, further details about the software packages used to implement the ML methods and other tables and figures are collected in two appendices at the end of the paper.

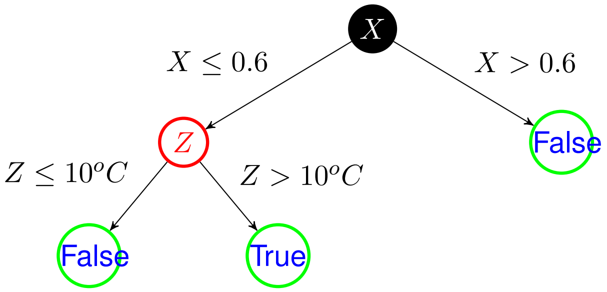
[2] For example, the XGBoost by default utilizes a nonlinear booster, called *gbtree*. However, one could use the linear booster *gblinear* to generate linear models.

[3] AutoML is available in H2O (an open-source platform for machine learning).

[4] https://www.kaggle.com/competitions.

2 Description of ML algorithms

We start by explaining the concept of *decision tree* which is used as the building block for the RF regression and other (tree-based) boosting algorithms.
Decision Trees: This model provides a tree-based method of splitting the data recursively using one feature at a time. At the end of the process, we end up with *leaf nodes* – that is, nodes where there are no further splits and where decisions are made. To take a simple example as an illustration, suppose the dependent variable Y is a binary (true-false) variable representing the statement "I will go out to play football today." The outcome of Y depends on two independent variables (or features): the chance of raining (X) and the temperature (Z). Given a training sample, say, $X = \{0.1, 0.8, 0.2, 0.1\}$, $Z = \{30^oC, 20^oC, 7^oC, 15^oC\}$, and $Y = \{True, False, False, True\}$, the algorithm can then automatically learn the following set of rules:



Here, X represents the root, Z represents a branch node, and True and False represent the leaf node. Starting with X, if X is sufficiently high (say, above 0.6), then the leaf node Y takes the value 'False'. Otherwise, we continue to split Z. If Z is high enough (say, above 10^oC), then the leaf node Y takes the value 'True'; otherwise, it takes the value 'False'. With this set of decision rules in hand, one can easily predict any future value of Y given values of X and Z. A critical issue is to find an appropriate threshold at which to split a node (for example, the numbers '0.6' and '10^oC' used to construct this decision tree). An ideal split would partition the sample into two non-overlapping sets. In our case, we were able to partition the training sample into two distinct sets: $\{\text{True: } (X \leq 0.6) \bigcap (Z > 10^oC)\}$ and $\{\text{False: } (X > 0.6) \bigcup ((X \leq 0.6) \bigcap (Z \leq 10^oC))\}$. In reality, there may be many features with multiple classes, and two classes may be overlapping. We cannot always get a pure split as we create more branches further down a tree, but we can make the leaf nodes as pure as possible by setting the node thresholds so as to minimize the Gini impurity or, equivalently, the cross entropy (the entropy is minimal if all members in a node belong to the same class and the entropy is maximal if all these members are uniformly distributed across classes).[5]

Although the preceding example illustrates the concept of a decision tree based on categorical variables (specifically, a binary variable in this case), the same concept applies if our dependent variable (Y) and features ($\boldsymbol{X}$) are real numbers. Imagine that the set of all possible values [that Y can take] has a finite dense set, say $w_1, \dots, w_M$, and each element w_m, $m = 1, \dots, M$, (which can be viewed as a class label) is associated with a region R_m constructed by recursively partitioning the sample space of $\boldsymbol{X}$ with a given set of thresholds. The tree regression function can then be approximated as: $E[Y|\boldsymbol{X} = \boldsymbol{x}] \approx \sum_{m=1}^{M} w_m \mathbb{I}(\boldsymbol{x} \in R_m)$, where $\mathbb{I}(A)$ is an indicator function that takes the value of one if the event A is true and zero otherwise. A tree grown using real-valued data on Y and $\boldsymbol{X}$ is called a regression tree.

Thus, a decision (*or* regression) tree is a sequence of greedy searches for the best splitting points for each feature, based on a training sample. However, this procedure tends to cause overfitting as the optimal splitting points may only work well for the training sample, but will not generalize well enough to new data. To prevent overfitting, it is necessary to prune the tree by setting limits for the maximal depth of the tree, the minimum number of observations needed in a level to allow further splitting, and the minimum number of observations needed in a level to allow it to become a leaf node. These parameters may be obtained via cross-validation (CV), which employs the model

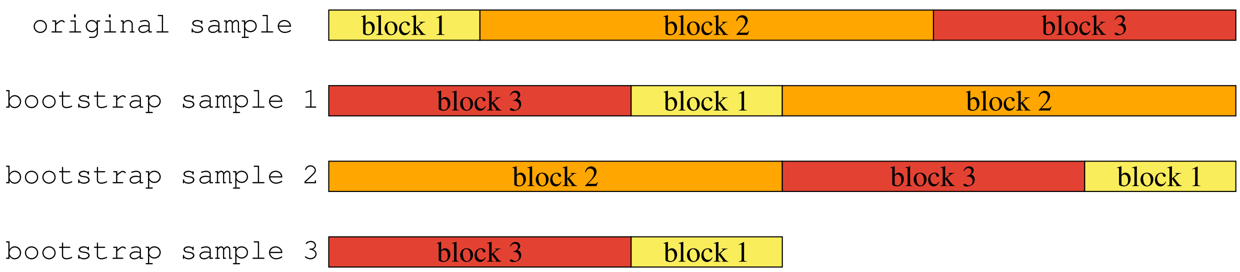
[5] The mathematical expressions of these criteria are given in Murphy (2012, Chapter 16).

estimated from a training sample to predict the dependent variable on a validating sample (that does not overlap with the training sample) such that the accuracy level on the validating sample is maximal. However, there are other more effective algorithms that can prevent overfitting, and at the same time, reduce the high variance that a tree regression may suffer from. We shall next describe two ensemble methods, namely the Random Forest and boosting algorithms.

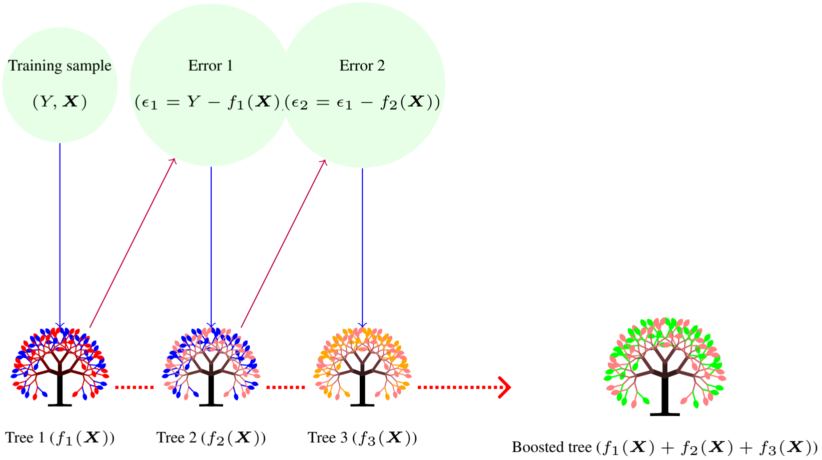
Random Forests: The method of random forests was first introduced by Ho (1995) and later formalized by Breiman (2001). A random forest can be considered as an ensemble of trees. The main idea is to average many different trees that individually suffer from high variance to build a more robust model that can produce better out-of-sample forecasts whilst being less susceptible to overfitting. The RF algorithm can be implemented in the following four steps:

1. Draw a bootstrap sample of size less than or equal to the original sample size.
2. Grow a decision tree from each bootstrap sample. At each node,
    a. Randomly select a subset of features without replacement.
    b. Split the node using the feature that provides the best split using the threshold obtained by minimizing the Gini impurity or the cross entropy.
3. Repeat the above two steps k times to create k independent trees.
4. Aggregate the predictions produced by the k trees.

For time series data [which have a natural ordering], the step 1 of the above RF algorithm is usually implemented via a "block bootstrap" method according to which a sufficiently long block of time-series observations is resampled in order to capture serial dependence in the block. The *regular block bootstrap* does this with a fixed block length while the *stationary bootstrap* uses random block lengths, where the length may be drawn from an auxiliary distribution (see, e.g., Lahiri (2003) for a detailed discussion of dependent bootstrap methods). Decreasing the size of the bootstrap sample (like the case with `bootstrap sample 3` below) improves the diversity of trees as the probability that a particular observation is included in all the bootstrap samples is lower. This diversity may yield a more random forest, which can help to reduce the effect of overfitting. On the contrary, increasing the size of the bootstrap sample can exacerbate the overfitting problem as trees are more likely to become similar to each other, and therefore learn to fit the original training data more closely. However, an issue with block bootstrap is that points around each joint of two blocks may not well mimic the serial dependence structure in the original sample. Therefore, if the time series is long, one could implement sub-sampling (i.e., sampling just one block at a time) instead. Another potential sampling scheme that could be used in the first step of the RF algorithm is Dahl and Sørensen's (2021) resampling using generative adversarial networks. This method can effectively generate many time series that mimic the serial dependence structure of the original series.



Boosting: The idea of 'boosting' was first introduced by Freund and Schapire (1996) and further developed by Breiman (1998) and Friedman (2001). Unlike a random forest, which is made of trees grown independently using different bootstrap samples of the original dataset, a boosted tree consists of individual trees trained in a sequence such that the next tree corrects the error made by the previous one, where the error is defined as the negativity of the gradient of a loss function. The gradient boosted trees algorithm builds one tree at a time and combines the results recursively as in the following diagram:


Training sample
$(Y, \boldsymbol{X})$
Error 1
$(\epsilon_1 = Y - f_1(\boldsymbol{X}))$
Error 2
$(\epsilon_2 = \epsilon_1 - f_2(\boldsymbol{X}))$
Tree 1 $(f_1(\boldsymbol{X}))$
Tree 2 $(f_2(\boldsymbol{X}))$
Tree 3 $(f_3(\boldsymbol{X}))$
Boosted tree $(f_1(\boldsymbol{X}) + f_2(\boldsymbol{X}) + f_3(\boldsymbol{X}))$

As mentioned above, in prediction problems, growing a very deep decision tree at each boosting step may lead to overfitting because doing so may reduce the expected loss on the training data beyond some point, but it can cause the expected loss on the testing data to stop decreasing and often to start increasing. The XGBoost developed by Chen and Guestrin (2016) overcomes the overfitting problem by using regularization at every boosting step to regularize the degree to which expected loss on the training data can be minimized.[6] As a consequence, regression trees built by the XGBoost are usually simple, but they can achieve a high level of accuracy when being applied to predict new data. One novel aspect of this algorithm is the fast node splitting method proposed for handling sparse data. Therefore, the XGBoost runs more than ten times faster than the standard boosting algorithms and it can scale to big datasets in shared-memory or distributed-computing environments.

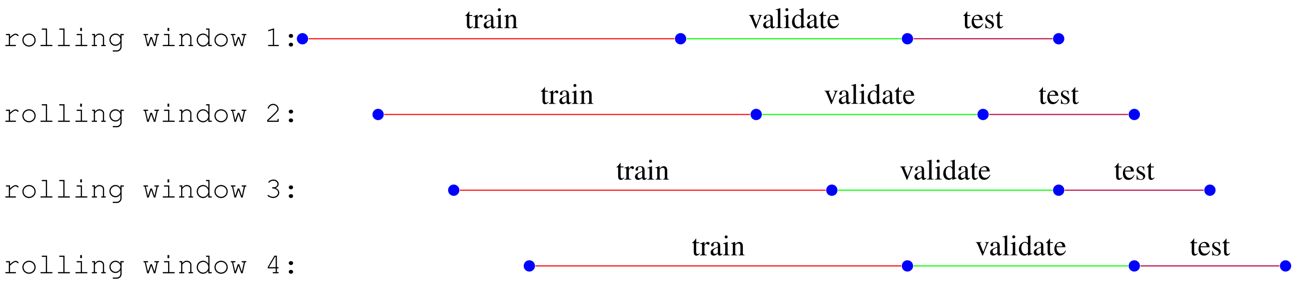
From a theoretical perspective, tree-based boosting algorithms (especially the XGBoost) can outperform the RF algorithm because the former algorithms can be much more immune to the curse of dimensionality [see, e.g., Nielsen (2016)].[7] The XGBoost learns the similarity between data points by adapting the neighbourhoods according to the amount of data in each region through various types of regularization while the RF employs only one type of regularization (i.e., row and column sampling as described above).

3 Forecasting Method

We implement the rolling-window strategy where the sample in each window is split into three sub-samples used to train, validate, and test a model respectively.

[6] To give an example of regularization, suppose that one tries to fit a quadratic polynomial, say $y = a + bx + cx^2$, into points that appear to follow the linear model: $y = a + bx$. The regression algorithm may produce curves that score well but look more complicated than what the original data would be like. We can influence this algorithm so that it can produce straight lines instead of curves by adding a penalty to the loss function, say $loss(x, y) + \lambda |c|$, where λ is a positive constant that controls how much we want to penalize c. If λ is set to zero, then there is no regularization. As λ is set to larger values, the larger c will then be heavily penalized. In general, regularization sacrifices some of the flexibility of a model in order to achieve better generalization to unseen data.

[7] The curse of dimensionality is defined as a phenomenon where the RMSE of a nonparametric estimator increases as the number of explanatory variables (*or* features) becomes larger, *ceteris paribus*. It is very common across most nonparametric estimation problems, such as splines, kernels, or trees.



We take the following steps to produce forecasts in each rolling window:

Step 1: In each training and validating sample, pre-select the most relevant predictors from the Official and GT variables by using the XGBoost or a variable screening procedure [such as Yousuf and Feng's (2021) PDC-SIS+].

Step 2: Use the above pre-selected predictors as inputs to train and validate random forest regression and boosted tree models.

Step 3: Use the models obtained in *Step 2* to produce out-of-sample forecasts for three horizons: $h = 1,\ 3,$ and 6.

We use data from January 2004 to 2019 (192 months) as the earliest date for which GT data is available is January 2004. The training periods range from January 2004 to December 2015 (143 months), the validating periods range from January 2016 to December 2017 (24 months), and the test periods range from January 2018 to December 2019 (24 months). We take the first differences of the logarithmic transformations of the data, then multiply them by 100 for both Official and GT data. We have also implemented the Kwiatkowski–Phillips–Schmidt–Shin (KPSS) test for stationarity on the data, and found that all the transformed variables are indeed stationary.

We evaluate our forecasts by using the following three metrics: the root mean squared error (RMSE), the mean absolute error (MAE) – which is more robust to outliers than the RMSE, and the out-of-sample (OoS) R^2 :

$$RMSE = \sqrt{\frac{1}{T-h-t_1}\sum_{t=t_1+1}^{T-h}\left(Y_{t+h}-\widehat{Y}_{t+h}\right)^2}, \tag{1}$$

$$MAE = \frac{1}{T-h-t_1}\sum_{t=t_1+1}^{T-h}\left|Y_{t+h}-\widehat{Y}_{t+h}\right|, \tag{2}$$

$$R^2 = 1 - \frac{\sum_{t=t_1+1}^{T-h}\left(Y_{t+h}-\widehat{Y}_{t+h}\right)^2}{\sum_{t=t_1+1}^{T-h}\left(Y_{t+h}-\widehat{Y}_{t+h}^{(AR)}\right)^2}, \tag{3}$$

where t_1 is the first observation of the out-of-sample period, T is the total number of observations, $\widehat{Y}_{t+h}$, $t = t_1+1,\ldots,T-h$, are the h-period ahead forecasts of Y_{t+h}, and $\widehat{Y}_{t+h}^{(AR)}$, $t = t_1+1,\ldots,T-h$, are the baseline h-period ahead forecasts of Y_{t+h} calculated from an autoregressive (AR) model. The OoS R^2 takes value in between $-\infty$ and one. Therefore, a negative value of the OoS R^2 means that the baseline forecast is better (in terms of the RMSE) while a positive value merely means that our forecast is better than the baseline forecast, and as the value of R^2 is closer to one, the better our forecast becomes.

When there are many methods or models available to forecast the same economic variable, an important question is to assess which models provide superior predictive ability. As pointed out by Hansen et al. (2011), since different competing models are usually built to answer a specific econometric question, a single model may not dominate all other models either because they are statistically equivalent or because there is not enough information coming from the data to discriminate the models. Therefore, we apply Hansen et al.'s (2011) model confidence set (MCS) procedure to determine if a method is one of the best forecasting methods. A method with a small MCS *p-value* is unlikely to belong to the set of best methods.

4 Data

Official data: We use both Canadian and U.S. variables to forecast the Canadian GDP growth rate. It has long been noted that some U.S. variables have predictive power for Canadian GDP because of the extensive Canada-U.S. trading relationship. Some variables that we use here have also been used by Chernis and Sekkel (2017) and the Bank of Nova Scotia's nowcasting model for the Canadian economy. There are nine Canadian economic variables and five U.S. variables. All the variables (together with their actual release and reference periods) used to predict the monthly real GDP growth rates are provided in Table 3.

GT data: This data set (available from `www.google.com/trends`) provides all search volumes and shows the percentage of web searches performed on Google. GT data set reports data in terms of an index of queries rather than in terms of the raw number of queries. This index is then normalized so that the maximum search volume at a given point in time and location is set to 100%, and every other observation is calculated relative to that maximum value. The resulting numbers on the scale from zero to 100 are thus based on the total number of searches for a specific category relative to all other search queries. Choi and Varian (2009) have used the term "Google Trends" while Götz and Knetsch (2019) and Ferrara and Simoni (2019) have used the term "Google search data". GT data is an unbiased sample of the Google search data. According to the Canadian internet usage survey in 2018, 94% of the population had internet access at home, and the share of all the Canadians aged 15 or above who use the internet was 91%. Interestingly, approximately 84% of the internet users bought about $57.4 billions worth of goods or services online in 2018 compared to $18.9 billions in 2012. We obtained all the data used in this study from Bloomberg, Statistics Canada, Google, and the Fed. GT data are only available for normal users from January 2004 to date.

We proceeded to divide our GT dataset into two main categories. The search keywords in the first category, which we call *general search terms*, are extracted from the R package '*gtrendsR*'. We have downloaded 26 categories and 269 subcategories of GT data for a total of 1776 search terms. Only 464 search terms remain after removing columns with missing values and outliers. The second category, which we call the *relevant search terms*, has the same variable names as those in the monthly and quarterly Canadian GDP components. There are 80 search terms in this category. If a column in the second category has no more than two missing values, we replaced these missing values with the column mean. We also used the program *X-13ARIMA-SEATS* developed by the U.S. Census Bureau and the Bank of Spain to seasonally adjust GT data since these data may not be seasonally adjusted.

5 Empirical Results

5.1 Selection of Predictors

We implemented *Step 1* of the rolling window strategy described in Section 3.[8]

Using the Official data, we have selected the best eight variables out of 14 official macroeconomic variables by using the variable importance measure of XGBoost.[9] This variable selection method is different from other traditional methods, such as PCA, PLS, SIS, and LASSO, in the sense that the former selects variables with an aim to maximizing their out-of-sample predictive ability for a target variable. We can notice how the model performance changes as we add or drop any variable based on the variable importance measure. Specifically, we run 1000 XGBoost models (constructed from decision trees) at a time, and compute the variable importance measures. We obtained the best predictive performance [using the validating sample] when using 1000 models (instead of 800 or 1600 models). Training a larger number of models for the current AutoML project may be accomplished by using different random seeds. These 1000 models have different parameter values, such as "depth of tree ", "sample rate ", or type of "grow

[8] We made use of the XGBoost function of the AutoML package.

[9] One of the advantages of using gradient boosting is that it provides importance scores for each feature after the boosted tree is constructed. Variables with high importance are, by definition, the major drivers of the target variable. We use the H2O to determine variable importance. In H2O, the variable importance score measures the relative influence of each feature according to the extent to which this feature can reduce the overall RMSE.

policy or booster". We chose the model with minimum RMSE (called leader model), and use the *scaled importance measure* to choose variables. A plot of the selected variables versus their importance scores is presented on the left panel of Figure 3. Here, we sequentially dropped the variables with less than 4-5 % of importance score, repeating the same process until we obtained sufficient improvement in the RMSE.

Using GT data, we select GT features by using either the XGBoost's variable importance scores or Yousuf and Feng's (2021) variable screening procedure (explained below). As illustrated in Figure 4, we first apply the XGBoost variable importance measure on the *general search terms* [listed in Table 5] and run 1000 tree-based models.[10] The whole process was repeated until we obtained a sufficient improvement in the RMSE. We dropped variables with importance scores less than 5-10 %. Here, we used slightly stricter variable selection criteria relative to the one used for Official data because GT data have a much larger number of features. We have ended up with 26 features from this category of search terms. Next, the same procedure was applied on the *relevant search terms* [listed in Table 4]. We eventually obtained 23 features for this category. Finally, we combined the top features from both the categories of GT data: there is a total of 22 features to be used for forecasting the GDP growth.

Yousuf and Feng's (2021) variable screening procedure based on Székely and Rizzo's (2014) partial distance correlation, namely *PDC-SIS+*, can be used to select time-series variables. This procedure defines the set of possible predictors as: $S_{k,l} = (Y_{t-1}, \ldots, Y_{t-L}, X_{t-1,k}, \ldots, X_{t-l+1,k})$ for $1 \leq k \leq K$ and $1 \leq l \leq L$, where K is the number of features and L is a maximum number of lags to be set. Using the partial distance correlation to screen out irrelevant variables has an important advantage over the conventional partial correlation as a null partial distance correlation between two variables implies conditional independence between these two variables while this is not necessarily the case with other measures of conditional correlation. Therefore, if an element in the set of predictors has zero partial distance correlation with the target variable, then this element has no predictive ability power. All the ML algorithms used in this paper actually train nonlinear models in which features and the target variable may have complex relationships. PDC-SIS+ can effectively select predictors that can be either linearly or nonlinearly associated with the target variable. To apply PDC-SIS+, we first combine *general search terms* and *relevant search terms* to make a set of total 544 predictors, then take square roots of these predictors. There are 24 variables being selected by this procedure, as listed in Table 7.

5.2 Comparison of Forecasts

Given the set of predictors selected in Section 5.1 above, we proceeded to *Step 2* of the rolling window strategy described in Section 3. The following ML algorithms were implemented: *Random Forest*, *Gradient Boosting Machine* (GBM), *Light Gradient Boosting Machine* (LightGBM) described in Appendix A, and the *XGBoost*, together with a simple benchmark *autoregressive* (AR) model of order one. After training and validating forecast models by employing these ML algorithms, we then used the trained models to compute out-of-sample forecasts (as described in *Step 3* in Section 3).

All the forecasts [at three horizons, $h = 1, 3$, and 6 steps ahead] of the monthly GDP growth rate using both Official and GT data for a shorter period of time (the entire sample period is from February 2004 to December 2019 and the out-of-sample period is from January 2018 to December 2019) are presented in Table 1. The first column lists ML algorithms being implemented, the second column presents types of data being used as the predictors of the GDP growth rate together with the procedures used to pre-select variables [explained in Section 5.1]. The rest of the columns provide the RMSE, the MAE, the OoS R^2, and the MCS *p-value* of each forecast method. As mentioned earlier, the Official data features (for both Canada and U.S.) presented here were pre-selected with the XGBoost. The key takeaways from this table are:

- All of the ML models have a superior forecasting accuracy than the AR model. (The only exception is the RF which performs less well than the AR in the case of the one-step-ahead forecast.) As expected, one-step-ahead

[10] Note that we can only provide a few GT data feature names (out of 1776 search terms in this category) in this table. The full list of features is available upon request.

forecasts are more accurate than two- or three- steps-ahead forecasts.

- The RF does not seem to perform as well as the XGBoost or the GBM across all the three forecast horizons. When it is applied to variables pre-selected (with PDC-SIS+) from both Official and GT variables, the RF's performance can be on par with that of other boosting algorithms using only GT or Official data.

- The LightGBM using Official data can perform better than other models in terms of the RMSE and the MCS *p-value* at the one-step-ahead forecast.

- The XGBoost and the GBM provide more accurate long-horizon forecasts. Specifically, the GBM with GT features selected using the XGBoost achieves the minimum RMSE among other models and various data categories at the three-steps-ahead forecast. Meanwhile, the GBM with GT features selected using PDC-SIS+ provides the minimum MAE for three-steps-ahead forecasts.

- The GT features pre-selected by PDC-SIS+ can be better predictors than those pre-selected by the XGBoost. Both the GT features and the Official variables perform equally well in a shorter horizon for the models trained by the XGBoost and the LightGBM. In a longer horizon, forecasts with the Official variables can outperform those with GT data. However, we do not observe an improvement in the forecasts using both GT and Official data. This is consistent with Ferrara and Simoni (2019) and Götz and Knetsch (2019) who conclude that forecast accuracy does not improve by using both GT and Official data.

A more detailed analysis of Table 1 also permits a comparison of the predictive success of models depending on the different forecast horizons and different data sets. Given the overall lower performance of the RF model in our case, we will only compare the GBM, XGBoost and LightGBM.

One-step ahead forecasts: In general, it seems that the GBM model is the best overall model across (almost) all data and variable selection categories and in terms of all the four metrics. The only exception is the case when Official data is used alone. In this case, the LightGBM model performs the best of all models, and has the lowest RMSE (0.00137), the lowest MAE (0.00086), and the best MCS p-value (1.000).

As for the XGBoost model, we can see that the latter is a relatively close second to the GBM in terms of its forecasting accuracy. The greatest difference in accuracy between the two models is observed when only GT data are used, but their accuracy in terms of RMSE is almost identical when both Official and GT data are used together. However, even in this case, the GBM is superior to XGBoost according to its MAE and its MCS p-value.

It is also interesting to note that for the one-step-ahead forecast, the GBM model yields smaller RMSE's in the case when only GT features are selected (0.00322 and 0.00077 respectively for the two selection criteria) than in the case when only Official data features are selected (0.0123), or in the case when both GT and Official data are selected (0.02627). This indicates that the GBM is best suited to produce very good short-term forecasts when only GT data are available.

Three-step ahead forecasts: While forecasting accuracy clearly diminishes across all models as the forecasting horizon lengthens, here too, we reach the same general conclusion as for the one-step-ahead forecasting horizon: GBM is still the best overall model with the XGBoost being a close second for all data and variable selection categories. The XGBoost's performance is very close to that of the GBM especially when GT variables are selected by PDC-SIS+ (which also emerges, overall, as the more successful variable selection method). Comparing the GBM model's performance when using both the Official variables and GT features selected with PDC-SIS+, with its performance when using only GT variables (selected with PDC-SIS+), we observe that Official data contribute to greater forecasting accuracy in terms of a higher MCS p-value (1.000 compared to 0.759), while the OoS R^2 is high (0.96) in both cases. The RMSE's and MAE's of the two cases are also comparable, with the RMSE being slightly higher (0.04831 compared to 0.04526) but the MAE being slightly lower (0.01272 compared to 0.01433).

Alternatively, when we compare the GBM's performance where only Official data is used with the case when both sets of data are used, we observe an improvement in forecasting accuracy according to all four metrics. (RMSE:

0.04831 versus 0.06164; MAE: 0.01272 versus 0.01875; OoS R^2 0.96 versus 0.94; MCS p-value 1.000 versus 0.759). Clearly, GT data contributes to forecasting accuracy here (in contrast to Ferrara and Simoni's (2019) finding).

Hence, we can conclude that the most successful model for the three-steps ahead forecast horizon is the GBM model using both pre-Official variables and GT variables selected with PDC-SIS+. The LightGBM model does not perform as well as the other two models in the three-steps ahead forecasting horizon even when Official data are used alone.

Six-step ahead forecasts: At this longer forecasting horizon, forecasting accuracy diminishes yet again across models. However, interestingly, we observe that now the XGBoost model tends to dominate the GBM model in terms of RMSE across all data and variable selection categories, but with the GBM model having somewhat superior MCS p-values. We can generally say that the XGBoost model is the most successful model except when Official data are used alone. In this latter case, once more, (as in the one-step-ahead-forecast) the LightGBM model performs the best of all models (according to its RMSE, its MAE, its OoS R^2 and its MCS p-value.)

Table 2 presents forecast results using only Official data for a longer period of time (the entire sample period is from January 1981 to December 2019 and the out-of-sample period is from January 2018 to December 2019). We implemented the same set of ML algorithms mentioned in the first column of Table 1. In general, forecast performance in terms of the RMSE and the MAE significantly improves with a longer sample of data. The values of the OoS R^2 reported in these tables are consistent with the values reported in other papers on GDP nowcasting. For example, de Valk, de Mattos, and Ferreira (2019) nowcasted Brazilian GDP growth with a dynamic factor model and found an OoS R^2 value of roughly 0.90. The tree-based boosting algorithms continue to outperform the RF algorithm in this case as well. Forecasts using the Canadian variables as predictors may be slightly better in a shorter horizon, and can deteriorate over longer horizons. This implies that the effect of the U.S. economy on the Canadian economy becomes visible over an extended period of time. Moreover, forecasts using only the Canadian employment data with the lags and a rolling average of GDP growth rates perform better than those using other Canadian variables and/or U.S. variables.

Figures 1 and 2 plot the out-of-sample forecast performance (for the period from January 2018 to December 2019) with GT plus Official data and Official data only, respectively. These plots show that one-step-ahead forecasts are quite close to the observed GDP growth rates, thus they can track very well the GDP growth path. All the models can produce good forecasts with employment as the only predictor. Indeed, Foroni, Marcellino, and Stevanovic (2020) have mentioned that employment and industrial production are good predictors of GDP. However, we could use only employment as it is available with a ten-day lag while industrial production is only available with a 50-day lag. Therefore, we can forecast the Canadian GDP growth well prior to the release of official GDP figures.

6 Conclusion

This paper compares the performance of the tree-based ML algorithms (i.e., the LightGBM, the XGBoost, the GBM, and the RF) in a GDP growth forecasting exercise using Official and GT data as predictors. In particular, we used the automated machine learning (AutoML) with the XGBoost algorithm to pre-select the most important variables from a broad set of potential candidate predictors. We run 1000 XGBoost models and implement AutoML 5 to 15 times until the RMSE can be sufficiently improved. We have dropped all variables with less than 5-10 % of variable importance.

We also employed another novel variable screening procedure, namely *PDC-SIS+*, to pre-select the best GT features. We have found that the features pre-selected by this variable screening procedure can produce a better forecast accuracy than the features pre-selected with the XGBoost. All the algorithms that we use can forecast the Canadian GDP growth very well, except for the RF. This result indicates that the GBM, the LightGBM, and the XGBoost can be very useful ML algorithms for macroeconomic prediction. We have also found that GT data can be a good substitute for Official data as the predictor of the monthly Canadian real GDP growth rate in shorter forecast horizons. However, the forecast ability of GT data can decrease over a longer horizon.

An important limitation of tree-based algorithms is that, although they are quite good at capturing the seasonal and nonlinear patterns in the data, these algorithms may not track time trends or structural breaks very well. We will study this issue further in a future research. We conjecture that the accuracy of forecasts in different scenarios can be enhanced by combining various boosting methods (for instance, the XGBoost with the LightGBM).

References

Athey, S., M. Bayati, G. Imbens, and Z. Qu (2019). Ensemble methods for causal effects in panel data settings. *AEA Papers and Proceedings 109*, 65–70.

Biau, O. and A. D'Elia (2012). Euro area GDP forecasting using large survey datasets: A random forest approach. mimeo, available from https://ec.europa.eu/eurostat/web/products-eurostat-news/-/EWP-2011-002.

Breiman, L. (1998). Arcing classifier. *Annals of Statistics 26*(3), 801–849.

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

Brown, P. P. and N. Hardy (2021). The mean squared prediction error paradox. mimeo.

Chen, T. and C. Guestrin (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785–794. ACM.

Chernis, T. and R. Sekkel (2017). A dynamic factor model for nowcasting Canadian GDP growth. *Empirical Economics 53*(1), 217–234.

Choi, H. and H. Varian (2009). Predicting initial claims for unemployment benefits. mimeo, available from https://static.googleusercontent.com/media/research.google.com/en//archive/papers/initialclaimsUS.pdf.

Choi, H. and H. Varian (2012). Predicting the present with Google Trends. *Economic Record 88*, 2–9.

Dahl, C. M. and E. N. Sørensen (2021, February). Time series (re)sampling using generative adversarial networks. mimeo.

de Valk, S., D. de Mattos, and P. Ferreira (2019). Nowcasting: An R package for predicting economic variables using dynamic factor models. *The R Journal 11*(1), 1–15.

Fan, J. and J. Lv (2008). Sure independence screening for ultrahigh dimensional feature space. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 70*(5), 849–911.

Ferrara, L. and A. Simoni (2019). When are Google data useful to nowcast GDP? an approach via pre-selection and shrinkage. mimeo, available from https://arxiv.org/abs/2007.00273.

Foroni, C., M. Marcellino, and D. Stevanovic (2020). Forecasting the Covid-19 recession and recovery: Lessons from the financial crisis. *International Journal of Forecasting* (in press).

Freund, Y. and R. E. Schapire (1996). Experiments with a new boosting algorithm. In L. Saitta (Ed.), *Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996)*, San Francisco, California, pp. 148–156. Morgan Kaufmann.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annal of Statistics 29*(5), 1189–1232.

Fuleky, P. (2019). *Macroeconomic Forecasting in the Era of Big Data: Theory and Practice*, Volume 52. Springer.

Götz, T. B. and T. A. Knetsch (2019). Google data in Bridge equation models for German GDP. *International Journal of Forecasting 35*(1), 45–66.

Hansen, P. R., A. Lunde, and J. M. Nason (2011). The model confidence set. *Econometrica 79*(2), 453–497.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The Elements of Statistical Learning: Data Mining, Inference, and Prediction* (second ed.). Springer.

Ho, T. K. (1995). Random decision forests. *Proceedings of the 3rd International Conference on Document Analysis and Recognition 1*(1), 278–282. Montreal, QC, Canada.

Jung, J.-K., M. Patnam, and A. Ter-Martirosyan (2018). An algorithmic crystal ball: Forecasts-based on machine learning. *IMF* working paper.

Lahiri, S. N. (2003). *Resampling Methods for Dependent Data*. Springer.

Medeiros, M. C., G. F. R. Vasconcelos, A. Veiga, and E. Zilberman (2021). Forecasting inflation in a data-rich environment: The benefits of machine learning methods. *Journal of Business & Economic Statistics 39*(1), 1–22.

Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. The MIT Press.

Nielsen, D. (2016,). Tree boosting with XGBoost: Why does XGBoost win "every" machine learning competition? Master's thesis, Norwegian University of Science and Technology.

Székely, G. J. and M. L. Rizzo (2014). Partial distance correlation with methods for dissimilarities. *Annals of Statistics 42*(6), 2382–2412.

Tiffin, A. (2016). Seeing in the dark: A machine-learning approach to nowcasting in Lebanon. *IMF* working paper.

Tkacz, G. (2001). Neural network forecasting of Canadian GDP growth. *International Journal of Forecasting 17*(1), 57–69.

Tkacz, G. (2013). Predicting recessions in real-time: Mining Google trends and electronic payments data for clues. *CD Howe Institute Commentary No. 387*.

Yousuf, K. and Y. Feng (2021). Targeting predictors via partial distance correlation with applications to financial forecasting. *Journal of Business & Economic Statistics* (forthcoming).

Figure 1: Predicted vs. actual real GDP growth (January 2018 - December 2019) with both GT and Official data (Full sample period: January 2004 - December 2019)

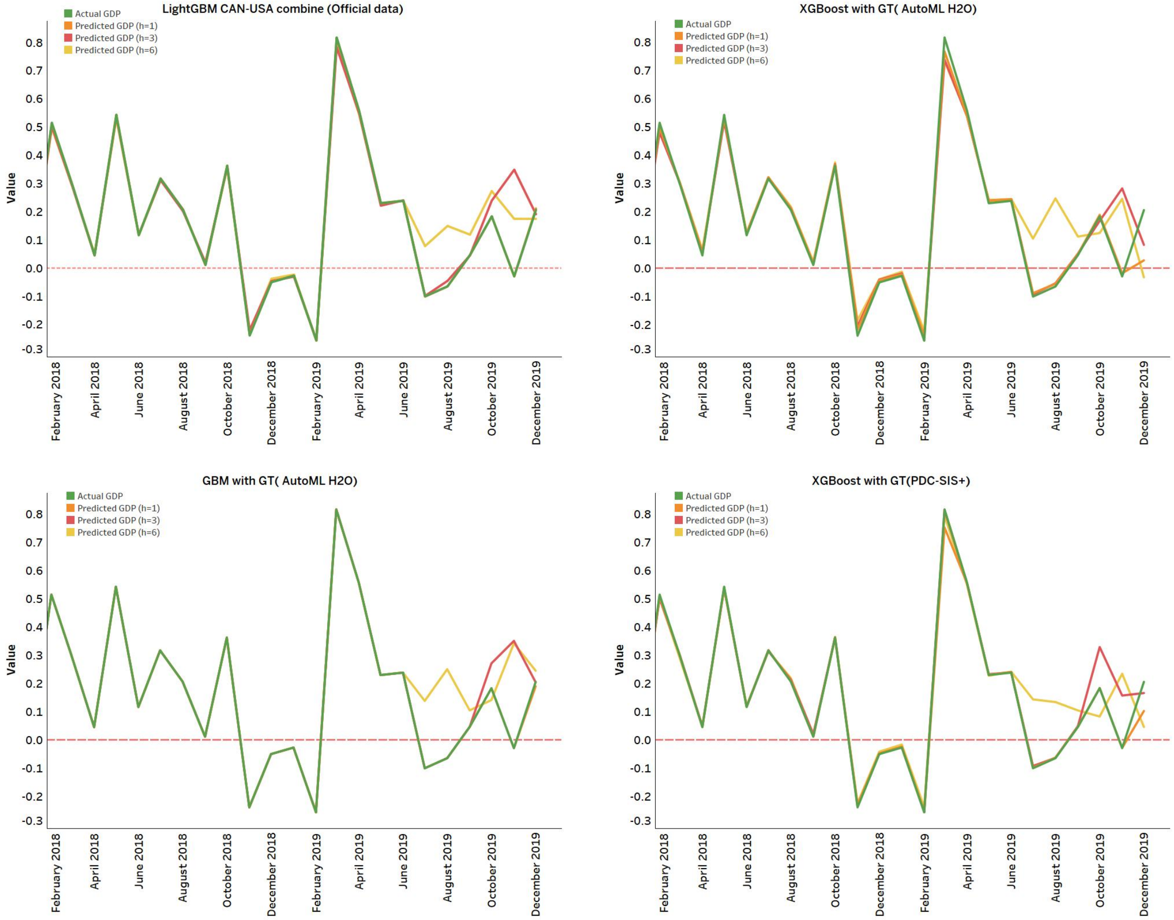


Figure 2: Predicted vs. actual real GDP growth (January 2018 - December 2019) with only Official data (Full sample period: January 1981 - December 2019)

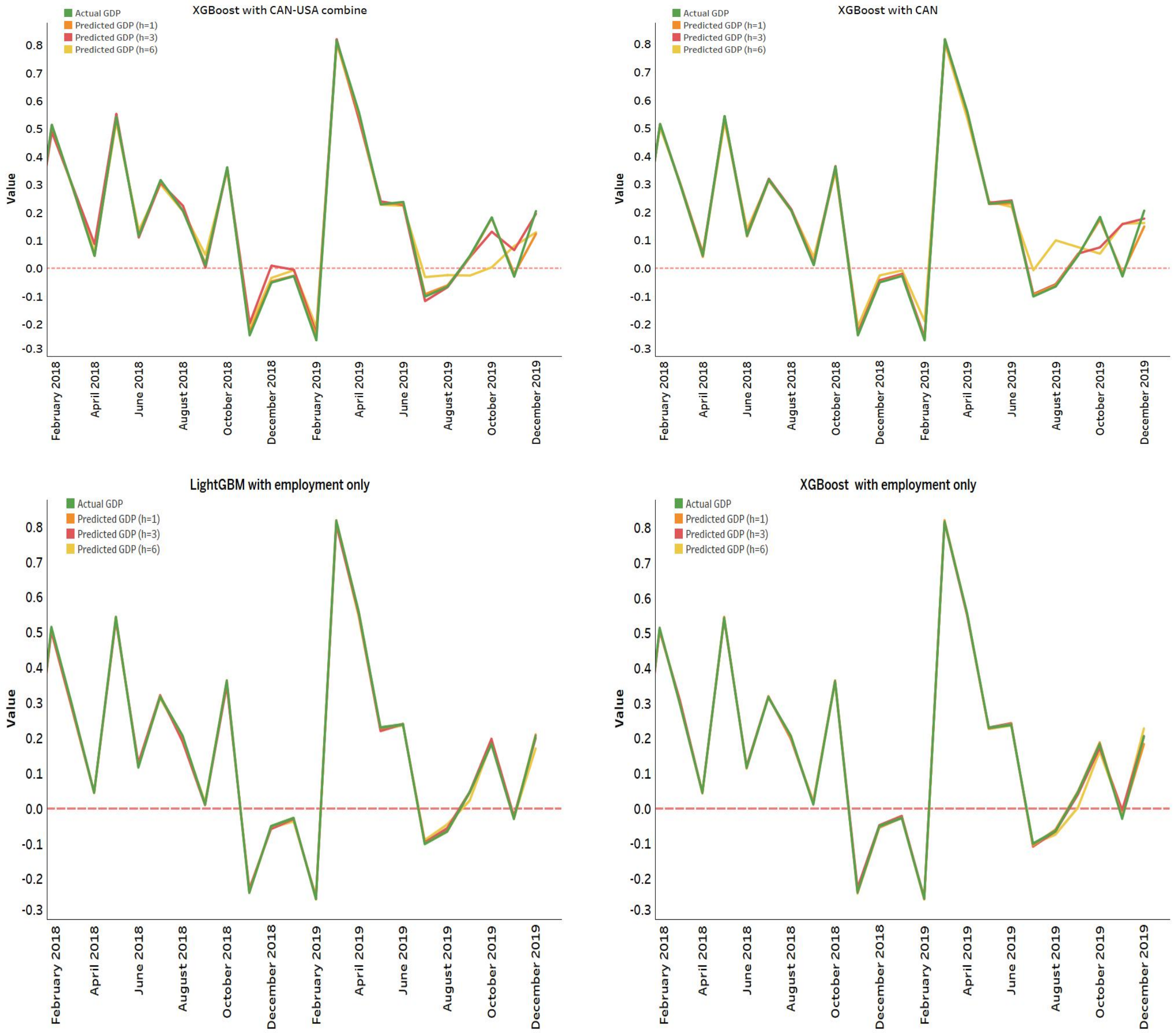


Table 1: Summary of the out-of-sample forecasts using both the GT and Official data features for the period from January 2018 to December 2019 (Full sample period: January 2004 - December 2019)

Algorithm	Data with Variable Selection Method	$h=1$				$h=3$				$h=6$			
		$RMSE$	MAE	OoS R^2	MCS *p-value*	$RMSE$	MAE	OoS R^2	MCS *p-value*	$RMSE$	MAE	OoS R^2	MCS *p-value*
Autoregression (AR)	GDP	0.09831	0.09831	-	0.028	0.11301	0.09397	-	0.008	0.17979	0.14967	-	0.015
Random Forest	GT with XGBoost	0.13843	0.10571	0.72	0.023	0.14085	0.10702	0.71	0.031	0.18387	0.14823	0.49	0.005
GBM	GT with XGBoost	0.00322	0.00065	0.99	0.580	0.01957	0.07968	0.91	0.759	0.11207	0.04441	0.81	0.390
XGBoost	GT with XGBoost	0.03905	0.01772	0.97	0.153	0.07161	0.03155	0.92	0.428	0.11054	0.06311	0.82	0.245
LightGBM	GT with XGBoost	0.09029	0.06887	0.87	0.000	0.15989	0.12814	0.62	0.020	0.16378	0.09864	0.60	0.258
Random Forest	GT with PDC-SIS+	0.17098	0.13193	0.56	0.024	0.11830	0.09431	0.79	0.041	0.18709	0.15048	0.48	0.027
GBM	GT with PDC-SIS+	0.00077	0.00380	0.99	0.378	0.04526	0.01433	0.96	0.759	0.14085	0.06312	0.71	0.390
XGBoost	GT with PDC-SIS+	0.02501	0.00907	0.99	0.138	0.04921	0.01862	0.96	0.759	0.09347	0.04911	0.87	0.384
LightGBM	GT with PDC-SIS+	0.00313	0.00129	0.99	0.378	0.04561	0.02460	0.96	0.759	0.12045	0.09707	0.78	0.190
Random Forest	Official*	0.12515	0.09576	0.76	0.039	0.14551	0.11450	0.68	0.049	0.15430	0.12334	0.64	0.049
GBM	Official	0.01230	0.00251	0.99	0.378	0.06164	0.01875	0.94	0.759	0.10321	0.04651	0.84	0.390
XGBoost	Official	0.01411	0.00721	0.99	0.028	0.06443	0.01992	0.93	0.759	0.09057	0.04839	0.87	0.390
LightGBM	Official	0.00137	0.00086	0.99	1.000	0.07868	0.02538	0.90	0.759	0.07501	0.03820	0.91	1.000
Random Forest	Official plus GT with PDC-SIS+	0.16612	0.12916	0.59	0.029	0.12575	0.09940	0.76	0.759	0.17953	0.14680	0.52	0.039
GBM	Official plus GT with PDC-SIS+	0.02627	0.00536	0.98	0.378	0.04831	0.01272	0.96	1	0.09170	0.03752	0.87	0.390
XGBoost	Official plus GT with PDC-SIS+	0.02632	0.01291	0.98	0.186	0.07479	0.03939	0.91	0.759	0.07916	0.04024	0.90	0.282
LightGBM	Official plus GT with PDC-SIS+	0.13310	0.10642	0.73	0.029	0.14541	0.11757	0.68	0.759	0.16264	0.12763	0.60	0.017

* The variable selection procedure used for Official data is always the XGBoost.

Table 2: Summary of the out-of-sample forecasts using only the Official data features for the period from January 2018 to December 2019 (Full sample period: January 1981 - December 2019)

Algorithm	Data	$h = 1$				$h = 3$				$h = 6$			
		$RMSE$	MAE	OoS R^2	MCS *p-value*	$RMSE$	MAE	OoS R^2	MCS *p-value*	$RMSE$	MAE	OoS R^2	MCS *p-value*
Autoregression (AR)	GDP	0.01617	0.01617	-	0.017	0.12956	0.08113	-	0.014	0.19603	0.15953	-	0.010
Random Forest	Canada & U.S.	0.08605	0.06485	0.89	0.029	0.07972	0.05983	0.90	0.013	0.08321	0.06706	0.89	0.004
GBM	Canada & U.S.	0.02635	0.00537	0.98	0.575	0.05858	0.02045	0.94	0.417	0.07318	0.03198	0.92	0.231
XGBoost	Canada & U.S.	0.01766	0.00804	0.99	0.090	0.03133	0.02237	0.98	0.025	0.05453	0.03717	0.95	0.218
LightGBM	Canada & U.S.	0.01865	0.00458	0.99	0.264	0.03932	0.01742	0.97	0.409	0.05959	0.04407	0.94	0.152
Random Forest	Canada	0.08519	0.06269	0.89	0.029	0.08054	0.05914	0.90	0.014	0.08211	0.06340	0.90	0.005
GBM	Canada	0.01432	0.00292	0.99	0.469	0.04953	0.01525	0.96	0.417	0.04356	0.01813	0.97	0.231
XGBoost	Canada	0.01298	0.00680	0.99	0.036	0.04489	0.01815	0.97	0.417	0.06595	0.04340	0.93	0.166
LightGBM	Canada	0.01214	0.00348	0.99	0.264	0.03312	0.01846	0.98	0.212	0.07438	0.05833	0.91	0.051
Random Forest	Employment only	0.01908	0.01541	0.99	0.001	0.01925	0.01537	0.99	0.005	0.02285	0.01874	0.99	0.013
GBM	Employment only	0.01228	0.00250	0.99	0.469	0.00847	0.00235	0.99	0.816	0.01907	0.00879	0.99	0.225
XGBoost	Employment only	0.00283	0.00210	0.99	0.575	0.00760	0.00529	0.99	1.000	0.01176	0.00647	0.99	1.000
LightGBM	Employment only	0.00248	0.00135	0.99	1.000	0.00829	0.00369	0.99	0.816	0.01183	0.00985	0.99	0.981

A Implementation Softwares

A.1 LightGBM

The LightGBM algorithm introduced by Microsoft is faster than the XGBoost. The LightGBM is different from the XGBoost in that the former algorithm grows trees vertically (and leaf-wise) whereas the latter algorithm grows trees level-wise. The LightGBM selects and splits the leaf that contributes the most to loss reduction; it uses a technique, called 'Gradient-based One-Side Sampling' (GOSS), which filters out data samples to select an optimal split point. The XGBoost utilizes a pre-sorted algorithm and a histogram-based algorithm to find the best split point. The XGBoost typically uses the nonlinear booster *gbtree* or the linear booster *gblinear* while the LightGBM uses the same booster type as the Gradient Boosting Decision, Random Forest, Dropouts meet Multiple Additive Regression Trees (DART), or GOSS.

A.2 Hyper-tuning Parameters

We have used both `R` and `Python` packages of *H2O* ver. 3.26.0.10 for the XGBoost. We also used the `Scikit-learn`, XGBoost, and LightGBM libraries of `Python` to calculate rolling-window forecasts, and the PDC-SIS+ `R` routine to select predictors. All the codes to replicate the results reported in the paper can be downloaded from `https://github.com/Shafi2016/Canadian-GDP-Forecast`.

Hyper-tuning parameters is one of the most complicated and vital tasks in ML in order to improve forecast performance. It is relatively easy to build a XGBoost model, but to fine-tune this model is a daunting task as the XGBoost has a large number of parameters. We used the package *AutoML* of H2O to initialize the parameters. We then utilized the routine *RandomizedSearchCV* with rolling time-series based cross-validation to hyper-tune the parameters. *RandomizedSearchCV* searches randomly through a large set of all possible combinations of the parameters, then calculates the optimal parameters. We initialized this routine with many different random seeds in order to obtain the parameter values with minimum RMSE while avoiding the effects of overfitting.

The XGBoost has many tuning parameters that may help us to cope with the overfitting issue. The first parameter controls the complexity of a decision tree. It splits the node only when the resulting split improves the loss function. Other regularization parameters are penalty levels to balance loss and the amount of penalization defined by an Elastic Net function. The parameter *max_depth* specifies the maximum depth of a tree. By increasing this parameter, we essentially increase the model complexity, and thus this can lead to overfitting. The parameter *learning_rate* shrinks the step size of the stochastic gradient descent algorithm used to minimize the loss function. The parameter *min_child_weight* defines the minimum summation of the weights of all observations required in a child. The parameter *sample_rate* specifies the fraction of observations randomly sampled for each tree. The parameter *booster* sets the learner type, either linear or nonlinear.

B Other Tables and Figures

Table 6: Final GT data features selected by *AutoML XGBoost*

Property.Management	Army	Ford
Small.Business	Financial.Markets	Volvo
Car.Electronics	Property.Development	Transportation
Finance	rental	consumption
Construction	Farm	Mining
Energy.sector	Services	US.employment
US.retail	Export	

Table 3: Official data variables for Canada and the U.S. used to forecast the Canadian GDP growth rate

Variable	Reference period	Release date	Publication lag (days)	Frequency
US : All Employees, Total Nonfarm (PAYEMS)	Dec 2019	Jan 8, 2020	8	M
CAN : Employment rate	-	Jan 10, 2020	10	M
US: Industrial Production: Total Index (INDPRO)	-	Jan 15, 2020	15	M
Motor Vehicle Assemblies: Autos and Light Truck Assemblies (MVAAUTLTTS)	-	Jan 15, 2020	15	M
Housing Starts: Total: New Privately Owned Housing Units Started (HOUST)	-	Jan 21, 2020	21	M
CAN: Retail trade	-	Jan 22, 2021	22	M
US : Light Weight Vehicle Sales: Autos and Light Trucks (ALTSALES)	-	Jan 29, 2020	29	M
CAN : Housing starts	-	Jan 30, 2020	30	M
CAN: Export	-	Feb 5, 2020	37	M
CAN : Import	-	Feb 5, 2020	37	M
CAN : Manufacturing new orders (total)	-	Feb 18, 2020	50	M
CAN: Industrial production	-	Feb 18, 2020	50	M
Manufacturing inventories (total)	-	Feb 18, 2020	50	M
CAN : wholesale trade	-	Feb 24, 2020	56	M
GDP by Industry	-	Feb 28, 2019	60	M

Table 4: Google Trends (GT) customized search keywords (relevant search terms)

Energy sector	Content	Construction	Rental
Business sector	Media	Manufacturing	Leasing
Business industries	Cannabis	Wholesale trade	Professional
Industrial production	Agriculture	Retail trade	Scientific
Manufacturing industries	Forestry	Transportation	Technical services
Durable good	fishing and hunting	Warehousing	Management companies
Information	Mining	cultural industries	Management enterprises
communication technology	Quarrying	Cultural	Administrative support
Technology sector	Oil extraction	Finance	Waste management
Public Sector	Utilities	Real estate	Health care
Social assistance	Arts	Entertainment	Recreation
Accommodation	Food services	Public administration	Perishable goods
Export	Import	Trade balance	US industrial
US Shipment	Building Permits	Consumer confidence index	Inflation
Employment	US employment	GDP growth	Price Index
Housing starts	US Retail	Shipment	US Shipment
US pmi	global pmi	Inventories	Retail trade
US retail	US housing	Consumption	Expenditure
Services	Investment	Residential	Machinery
Equipment	Intellectual.property	Farm	

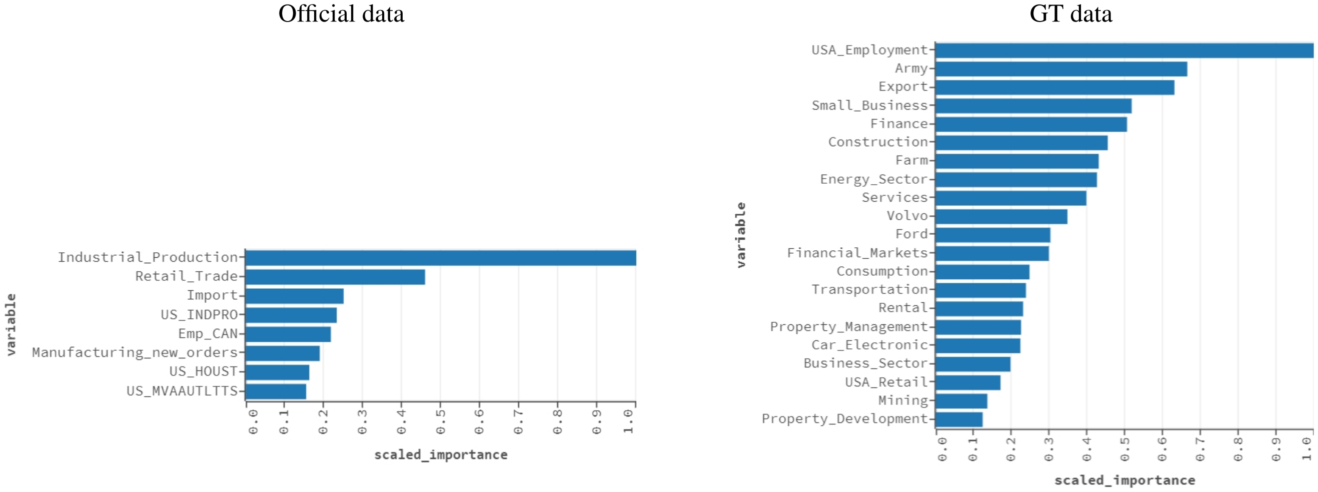
Table 5: GT general search terms

Porsche	Aquaculture	Office.Supplies	Import.Export
Rolls.Royce	Food.Production	Office.Furniture	Maritime.Transport
Saab	Forestry	Printers	Packaging
Saturn	Horticulture	Scanners	Parking
Subaru	Livestock	Outsourcing	Public.Storage
Suzuki	Business.Education	Signage	Rail.Transport
Toyota	Business.Finance	Civil.Engineering	Computer.Hardware
Scion	Investment.Banking	Electricity	Computer.Components
Volkswagen	Risk.Management	Nuclear.Energy	Computer.Memory
Volvo	Venture.Capital	Oil...Gas	Hard.Drives
Auto.Interior	Financial.Markets	Waste.Management	Network.Storage
Car.Electronics	Business.Operations	Recycling	Copiers
Car.Audio	Human.Resources	Data.Management	Desktop.Computers
Car.Video	Management	Hospitality.Industry	Computer.Security
Body.Art	Business.Process	Event.Planning	Network.Security
Cosmetic.Surgery	Project.Management	Food.Service	Audio.Equipment
Fitness	Project.Management.Software	Restaurant.Supply	Headphones
Bodybuilding	Strategic.Planning	Generators	Speakers
Hair.Care	Supply.Chain.Management	Heavy.Machinery	Camera.Lenses
Hair.Loss	Business.Services	Manufacturing	Cameras
Massage.Therapy	Consulting	Small.Business	Nintendo
Weight.Loss	Corporate.Events	Home.Office	Sony.PlayStation
Marketing.Services	Knowledge.Management	Aviation	Infectious.Diseases

Table 7: Final GT data features selected by *PDC-SIS+*

Finance_lag2	Food Production	Consumption	Farm
Construction_lag2	Agriculture	Ford_lag1	Energy sector
Finance	Construction_lag1	Electricity_lag1	Finance_lag1
Food.Service_lag1	Audi	Administrative.support_lag2	Consumption_lag2
Audi_lag2	Electricity	Expenditure_lag2	Automotive.Industry_lag2
Business. sector_lag1	Ford	Food Production_lag2	Business.sector

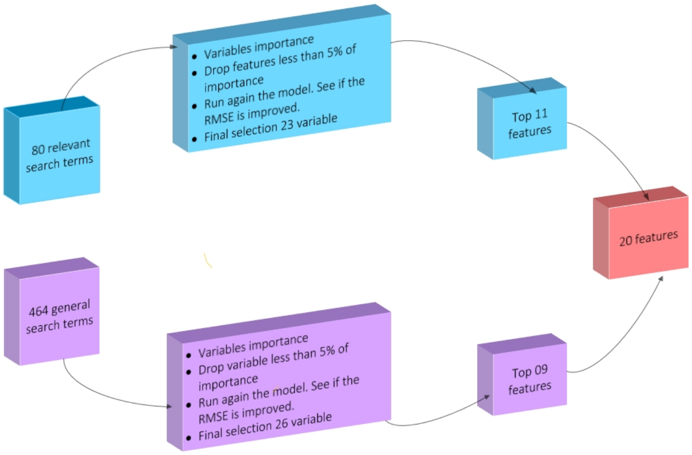
Figure 3: Variable importance plot



Official data

GT data

Figure 4: Variable selection procedure for GT data



- Variables importance
- Drop features less than 5% of importance
- Run again the model. See if the RMSE is improved.
- Final selection 23 variable

- Variables importance
- Drop variable less than 5% of importance
- Run again the model. See if the RMSE is improved.
- Final selection 26 variable