

Article

Dynamic Feature Selection for Canadian GDP Forecasting: Machine Learning with Google Trends and Official Data

Shafiullah Qureshi ^{1,*}, Ba M. Chu ² , Fanny S. Demers ², Najib Khan ³ and Ateeq ur Rehman Irshad ⁴ ¹ Data Analytics, Ministry of Jobs, Economy, Trade, and Immigration, Edmonton, AB T5J 4G8, Canada² Department of Economics, Carleton University, Ottawa, ON K1S 5B6, Canada; ba.chu@carleton.ca (B.M.C.); fanny.demers@carleton.ca (F.S.D.)³ Department of Finance, College of Business Administration (CBA), Prince Sultan University, P.O. Box 66833, Riyadh 11586, Saudi Arabia; nkhan@psu.edu.sa⁴ Department of Mathematics and Sciences, College of Humanities and Sciences (CHS), Prince Sultan University, P.O. Box 66833, Riyadh 11586, Saudi Arabia; airshad@psu.edu.sa

* Correspondence: qureshi.shafiullah@gmail.com

Abstract

We forecast monthly Canadian real GDP growth using machine learning models trained on Official macroeconomic indicators and Google Trends (GT) data. Predictors are selected dynamically in each rolling window using PDC-SIS, with cross-validation-based tuning to support real-time forecasting and avoid data leakage. The evaluation is conducted on the latest-available (final-vintage) series and should be interpreted as a pseudo out-of-sample forecasting exercise rather than real-time vintage nowcasting. We evaluate GBM, XGBoost, LightGBM, CatBoost, and Random Forest against an ARIMA baseline. Official data deliver the strongest performance at short and medium horizons, while combining Official and GT data yields the clearest improvement at the longest horizon. With GT data alone, LightGBM is the only ML model maintaining positive out-of-sample R^2 across all horizons. Diebold–Mariano tests corroborate these patterns: LightGBM dominates other ML models under GT-only predictors, whereas with Official and combined data, the horizon-specific best models significantly outperform ARIMA, with smaller differences among leading tree-based methods.

Keywords: GDP forecasting; machine learning; google trends; random forest; gradient boosting; variable selection; partial distance correlation; macroeconomic forecasting

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 Trends (GT) data for economic forecasting has become popular since the seminal work of Choi and Varian [1] and Choi and Varian [2]. GT data has been exploited to predict retail sales, automotive sales, home sales, and travel volume,



Academic Editor: Fabrizio Riguzzi

Received: 29 November 2025

Revised: 18 February 2026

Accepted: 24 February 2026

Published: 9 March 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

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 [3] used GT data to predict the 2008–2009 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 2008–2009 recession. In the context of GDP forecasting, Götz and Knetsch [4] used GT and other survey data to forecast German GDP with the Bridge regression. 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. 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 [5] also used a Bridge regression to nowcast Euro-area GDP with GT data. They employed Fan and Lv [6] 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 become available.

In this paper, we apply state-of-the-art machine learning (ML) algorithms to forecast monthly Canadian GDP growth using both Google Trends (GT) and Official macroeconomic data from a large database of Fortin-Gagnon et al. [7]. We hereafter refer to the latter as Official data for the details of features selected by the RF model with PDC-SIS. Our approach introduces a novel dynamic feature selection methodology, which incorporates cross-validation in each rolling window. By dynamically pre-selecting relevant predictors in each rolling window, we ensure robust, up-to-date model adjustments, avoiding data leakage and enhancing real-time applicability.

For variable selection, short-term forecasts ($\tau = 1$) rely primarily on high-frequency market and business indicators, with GT data capturing the automotive sector (Audi), small business trends, and electricity consumption, while Official data focus on residential mortgages, treasury bills, the stock exchange high, and energy exports. Medium-term forecasts ($\tau = 3$) shift towards energy and employment indicators, with GT data emphasizing small business activity, electricity consumption, and rental market trends, and Official data highlighting WTI oil prices, energy exports, transportation imports, and employment services. For longer horizons ($\tau = 6$), industrial and labor market indicators become prominent, with GT data capturing finance sector activity, automotive trends (Volvo), and mining sector searches, while Official data focus on metal prices, consumer imports, manufacturing inventories, and total hours worked. At the very long-term ($\tau = 9$), structural economic indicators dominate, with GT data capturing agriculture sector activity, unemployment trends, and automotive searches (Volvo), while Official data emphasize energy exports, government bond spreads, consumer imports, and oil imports. Each dataset thus contributes distinct and complementary features across horizons, with electricity consumption from GT data and energy trade indicators from Official data maintaining consistent importance throughout all time frames.

We show that real GDP growth can be forecast 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 remains useful for forecasting GDP when Official data are available. 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 Götz and Knetsch [4] and Ferrara and Simoni [5]). Our results

reveal that Official economic data consistently deliver the strongest forecasting performance at short and medium horizons, while combining Official and GT data achieves its clearest improvement only at the longest horizon. Official data produce the lowest RMSE at one, three, and six step horizons, demonstrating that traditional macroeconomic indicators provide robust predictive information. Combined data yield the lowest RMSE at the nine-step horizon, suggesting modest complementary benefits from GT variables at longer forecasts. GT data alone, however, produce substantially higher forecast errors than both Official and Combined approaches across all horizons, with most machine learning models failing to outperform the ARIMA baseline. These findings establish that Official economic indicators remain the primary source of predictive power for GDP forecasting, while GT data provide limited standalone value and modest incremental gains only when combined with Official data at specific horizons. Common Google Trends (GT) features selected by the Random Forest model with PDC-SIS are summarized in Table 1. Model performance proves data-source dependent rather than universal. For Official data, CatBoost achieves the lowest RMSE at short horizons, Random Forest at the medium horizon, and GBM at the long horizon. With GT data alone, LightGBM is the only ML model maintaining positive out of sample R^2 values across all horizons, while other ML models frequently underperform ARIMA. For combined data, CatBoost leads at short horizons, XGBoost at medium, and GBM at long horizons. Diebold–Mariano tests confirm these patterns: with GT data, LightGBM significantly outperforms all other ML competitors, while with Official and combined data, the best performing models at each horizon significantly outperform ARIMA at conventional significance levels, while differences among the leading tree-based models are often small. These results establish a clear data-source-specific model hierarchy: LightGBM for GT data, and CatBoost/Random Forest/GBM for Official and Combined data, depending on horizon. Rather than applying a single model universally, our findings underscore the importance of matching algorithms to data characteristics in macroeconomic forecasting applications.

Work on forecasting with ML has recently been growing. Beyond macroeconomic forecasting, ML has also been widely applied to MOOC course recommendation, data-centric smart cities, and fraud detection in credit cards and blockchain-based financial systems [8–11]. It would be impossible to provide a complete list here. We shall unfortunately only be able to discuss a few, certainly not comprehensive, contributions that are closely related to the topic of the present paper. Biau and D’Elia [12] 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 [13] and Jung et al. [14], have applied the Elastic Net, SuperLearner, and Deep Learning 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 [15]. Recently, Medeiros et al. [16] extensively applied standard ML methods to forecast the U.S. inflation rate, and found that the Random Forest (RF) algorithm can deliver the best forecasts among all the ML algorithms that they considered. Coulombe et al. [17] have shown that employing nonlinear models, regularization, cross-validation, and alternative loss functions can significantly improve the predictive performance of the ML methods in a data-rich environment. With respect to forecasts of Canadian GDP, Tkacz [18] adopted a neural network approach, while Chernis and Sekkel [19] 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, eXtreme Gradient Boosting (XGBoost), CatBoost and Microsoft's Light Gradient Boosting Machine (LightGBM). For example, XGBoost by default utilizes a nonlinear booster, *gbtree*; however, one can use the linear booster *gblinear* to generate linear models. Such ensemble methods have demonstrated empirical success in many applications, as averaging several nonlinear models with potentially different sets of predictive variables is often an effective way to address model uncertainty, which is very common in practice (see, e.g., Athey et al. [20]).

Table 1. Common Google Trends (GT) data features selected by the Random Forest model with PDC-SIS.

Time Horizon	Features
Core Economic	
$\tau = 1$	Small business trends Real estate activity Unemployment
$\tau = 3$	Small business activity Loan activity
$\tau = 6$	Loan trends Mining sector activity Trade activity
$\tau = 9$	Unemployment Agriculture sector Budget activity
Manufacturing/Industry	
$\tau = 1, 3$	Military/Defense sector (Army)
$\tau = 6$	Mining sector
$\tau = 9$	Agriculture sector
Consumer Services	
$\tau = 1$	Automotive sector (Audi) Online activity
$\tau = 3$	Food service industry Wholesale trade Rental market
$\tau = 6$	Travel industry Automotive sector (Volvo)
$\tau = 9$	Automotive sector (Volvo, Ford)
Financial	
$\tau = 3$	Loan activity
$\tau = 6$	Finance sector Insurance sector
$\tau = 9$	Insurance sector
Utilities	
$\tau = 1, 3, 6, 9$	Electricity consumption

Note: This table lists the features that were consistently selected across 109 rolling windows by PDC-SIS variable selection with GT data. Features are grouped by economic category and organized by forecast horizon ($\tau = 1, 3, 6$, and 9 months ahead).

While comparing the methodology of various papers, we focus on four dimensions: (i) how predictors are chosen (external screener versus embedded penalties/factors/Bayesian inclusion); (ii) where and how hyperparameters are tuned (nested or joint procedures versus sequential choices); (iii) the window structure (rolling versus expanding/recursive); and (iv) whether the design is leakage-safe in real time. Closest to our setting is Ferrara

and Simoni [5], who implement a sequential, within-date procedure: they first conduct a targeted preselection of Google Search Data conditional on official indicators (t-stat/SIS-style screening calibrated by asymptotic thresholds), and then estimate a ridge regression on the screened set with the penalty chosen by generalized cross-validation. Estimation is performed in pseudo-real-time using real-time vintages and an expanding (recursive) sample across weekly within-quarter information sets; there is no use of PDC-SIS and no tree-boosting learners. Methodologically related but distinct are the following strands. Bulligan et al. [21] create “targeted predictors” by screening across many estimation windows and then fix the feature pool before the out-of-sample period (i.e., no re-selection in each forecast window). Babii et al. [22] and Beyhum and Striaukas [23] employ penalized MIDAS specifications with blocked cross-validation, where selection is embedded in the penalty rather than performed by an external screener. Soybilgen and Yazgan [24] extract dynamic factors from large monthly panels and feed those factors to bagging, Random Forests, and Gradient Boosting with tuning performed once rather than in every window. Kant et al. [25] compare machine-learning methods for Dutch GDP in expanding windows with per-forecast cross-validation and embedded L_1 selection. Franjic and Schweikert [26] combine Elastic-Net preselection with dynamic factor models in rolling windows.

Compared with Bayesian dynamic-inclusion models in the econometric tradition—which endogenize selection via state-space priors and variational Bayes—our contribution is a practical ML pipeline that dynamically screens with PDC-SIS and conducts tuning and estimation within each rolling window for real-time deployment. Our approach also differs from recent high-dimensional forecasting methods in two key respects. First, whereas Babii et al. [22] embed variable selection within penalized linear regression (sparse-group LASSO), we perform explicit distance-correlation screening (PDC-SIS) to capture nonlinear predictor–target dependence before applying tree-based ML. Second, we integrate PDC-SIS screening, hyperparameter tuning, and model estimation inside a single rolling-window pipeline, using only data available in each window. This one-step, leakage-safe framework enables the use of nonlinear ensemble methods (GBM, XGBoost, CatBoost, and Random Forest)—which do not possess built-in selection mechanisms—while preserving strict temporal separation. Relative to Franjic and Schweikert [26], who combine Elastic Net with factor models, our framework (i) uses PDC-SIS to capture nonlinear dependence beyond linear penalties, (ii) employs tree-based ML rather than factor structures, and (iii) is model-agnostic and thus applicable to any learner. Building on our previous work applying distance-correlation screening with machine-learning models for GDP forecasting Chu and Qureshi [27], this study introduces the first dynamic implementation in which the screening is performed adaptively within each rolling window. To our knowledge, this is the first application of distance-correlation screening for adaptive feature selection in macroeconomic forecasting with nonlinear ensemble methods, executed within a one-step rolling-window pipeline and evaluated head-to-head across Official, Google Trends, and combined data at multi-month horizons.

Our proposed forecasting procedure is a one-step procedure detailed in Section 3. This streamlined approach integrates variable selection and forecasting into a single, continuous process. We employ Yousuf and Feng [28]’s distance-correlation-based variable screening procedure, namely the PDC-SIS, to dynamically select the most relevant predictors within each rolling window. We use the standard forecast evaluation metrics: the root mean squared error (RMSE), the mean absolute error (MAE), the mean absolute deviation (MAD), and OoS R^2 .

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 et al. [29] and Murphy [30]. Section 3 details the forecasting procedure, implemented in two operational steps. 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. Description of ML Algorithms

This section briefly summarizes the learning algorithms used in our forecasting exercise, focusing on the elements that are relevant for replication and interpretation in a rolling real-time design. For a general background on tree-based learners and boosting, we refer the reader to standard texts (e.g., Murphy [30], Chapter 16) and the original methodological papers cited below.

2.1. Decision Trees (Building Block)

Tree-based methods approximate the conditional mean $E[Y | X]$ by recursively partitioning the predictor space into regions and assigning a constant prediction to each region (leaf). In regression settings, split points are chosen greedily to reduce a loss such as the sum of squared errors within leaves. Because unconstrained trees can overfit, practical implementations control tree complexity through hyperparameters such as maximum depth, minimum leaf size, and the minimum number of observations required to split a node. These complexity controls are selected within each rolling window as part of our tuning protocol (see Section 3).

2.2. Random Forests (Bagging-Based Ensemble)

Random Forests, proposed by Breiman [31] reduce the variance of a single tree by averaging predictions across many decorrelated trees trained on bootstrap samples and random subsets of predictors at each split. In our application, the main tuning dimensions are the number of trees and the parameters governing tree complexity (depth/leaf size) and feature subsampling. Averaging across trees yields robust performance in data-rich settings and mitigates overfitting relative to a single deep tree.

2.3. Gradient Boosting and Boosted-Tree Variants

Boosting constructs an additive model by fitting weak learners sequentially, where each new tree is chosen to reduce a loss function given the current ensemble Freund and Schapire [32], Breiman [33], Friedman [34]. In Gradient Boosting, the next tree is fit to the negative gradient of the loss (often interpreted as residual-like information under squared-error loss). Key regularization mechanisms include shrinkage (learning rate), limiting tree depth, and row/column subsampling. These controls govern the bias–variance trade-off and are tuned within each rolling window.

We evaluate several boosted-tree implementations. XGBoost [35] augments Gradient Boosting with explicit regularization terms and efficient split finding, often improving generalization and computational speed. LightGBM and CatBoost follow the same boosted-tree principle but differ in engineering choices (e.g., histogram-based splitting, growth policies, and handling of categorical features), which can materially affect performance in large, sparse, or mixed-type predictor sets. Across boosted-tree methods, the main tuned hyperparameters include the number of boosting iterations, learning rate, maximum depth (or leaf constraints), subsampling rates, and regularization strength.

2.4. Why Tree-Based Ensembles Are Suitable Here

Tree-based ensembles (RF and boosted trees) are attractive for macro-forecasting with many candidate predictors because they can accommodate nonlinearities and interactions without requiring explicit functional-form specification. Moreover, with appropriate regularization and a strict rolling-window training/validation protocol, these methods can provide stable out-of-sample performance in high-dimensional environments. Any additional implementation details (e.g., exact tuning grids and validation design) are provided in the methodology section to avoid redundancy.

3. Forecasting Method

Following Chu and Qureshi [27], we employ a recursive rolling-window strategy to evaluate the out-of-sample (OoS) performance of our forecasting models over time. Multiple rolling sub-samples are constructed by shifting the window forward by one observation, thereby incorporating the most recent data available while discarding the oldest observation. For each sub-sample, we compute the OoS performance metrics (RMSE, MAD, MAE, and OoS R^2) for different forecast horizons (τ), preserving a real-time forecasting context by avoiding the use of future information in the modeling process.

Variable selection is dynamically performed within each rolling window, ensuring that the model continuously adapts to the most relevant predictors as new data becomes available, effectively preventing data leakage. In each rolling sub-sample, the data is split into three segments for training, validation, and evaluation. Each rolling sub-sample maintains fixed sizes for these segments: the training set consists of the initial portion of observations, followed by a validation set of 16 observations, and finally, an evaluation segment used to produce τ -step-ahead forecasts. The training set size varies depending on the forecast horizon, ensuring that the overall sub-sample size (T_1) remains consistent. For instance, with $T_1 = 100$, a one-step-ahead forecast ($\tau = 1$) uses 81 observations for training, whereas a nine-step-ahead forecast ($\tau = 9$) uses 65 observations for training. This structure supports forecasts for one- to nine-period-ahead horizons and ensures consistency across all rolling windows. The tuning spaces used for all ML algorithms are reported in Appendix A, Table A4. The rolling-window forecasting scheme is illustrated in Figure 1. We refer to this as a one-step procedure because variable selection, hyperparameter tuning, and estimation are executed together within each rolling window. In each rolling window, we split the data into training, validation, and evaluation segments. Variable selection is performed dynamically using PDC-SIS on the training segment only, ensuring that only information available up to that window is used and preventing any future information from influencing selection. Conditional on the selected predictors, we tune hyperparameters on the validation segment by selecting the configuration that minimizes out-of-sample (OoS) RMSE. The model is then refit on the combined training and validation segments using the selected predictors and optimal hyperparameters, and it generates τ -period-ahead forecasts for the evaluation segment as the window rolls forward. Each realized observation, aligned with the forecast horizon, is subsequently used to compute forecast errors.

We use data from January 2004 to September 2021, totaling 214 months, as January 2004 is the earliest date for which Google Trends (GT) data is available. Although recent data retrieval is limited by API rate limits and associated costs, we previously downloaded historical GT data to ensure comprehensive analysis and consistency across time. For both Official and GT data, we apply a first-difference transformation to the logarithmic values of the variables and scale these values by multiplying by 100. This transformation standardizes the data for analysis. Additionally, we performed the Kwiatkowski–Phillips–Schmidt–Shin

(KPSS) test to confirm stationarity of the transformed variables, verifying that all variables are indeed stationary.

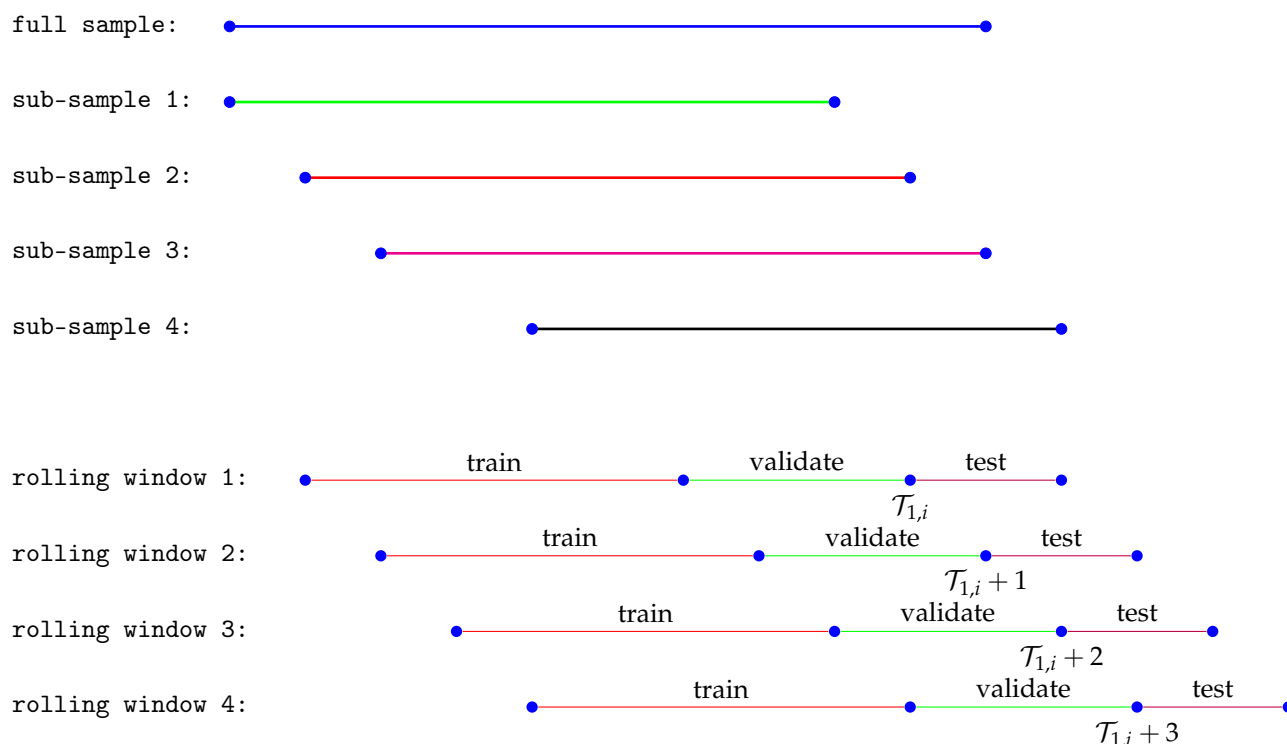


Figure 1. Rolling-window forecasting scheme. Each rolling window is split into training, validation, and test segments, and the window then moves forward through time.

We evaluate forecasts using four metrics: the root mean squared error (RMSE), mean absolute error (MAE), median absolute deviation (MAD), and out-of-sample R^2 (OoS R^2). Let $e_{t+\tau}^{(m)} = Y_{t+\tau} - \hat{Y}_{t+\tau}^{(m)}$ denote the τ -step-ahead forecast error of model m . For each horizon τ , we compute rolling-window accuracy measures over a window of length W :

$$\text{RMSE}_{t,\tau}^{(m)} = \sqrt{\frac{1}{W} \sum_{j=0}^{W-1} \left(e_{t+\tau-j}^{(m)} \right)^2}, \quad (1)$$

$$\text{MAE}_{t,\tau}^{(m)} = \frac{1}{W} \sum_{j=0}^{W-1} \left| e_{t+\tau-j}^{(m)} \right|, \quad (2)$$

$$\text{MAD}_{t,\tau}^{(m)} = \text{median} \left\{ \left| e_{t+\tau-j}^{(m)} \right| : j = 0, \dots, W-1 \right\}. \quad (3)$$

For the OoS R^2 , we use the rolling RMSE-ratio definition reported in our empirical results. Let $\hat{Y}_{t+\tau}^{(\text{bench})}$ denote the benchmark forecast, computed as a backward-looking rolling mean of past realizations using a window of length B . Define the benchmark rolling RMSE $\text{RMSE}_{t,\tau}^{(\text{bench})}$ analogously. The rolling OoS R^2 is:

$$R_{t,\tau}^2 = 1 - \left(\frac{\text{RMSE}_{t,\tau}^{(m)}}{\text{RMSE}_{t,\tau}^{(\text{bench})}} \right)^2. \quad (4)$$

In contrast, ARIMA is used as the baseline model comparator for the forecast evaluations (and the Diebold–Mariano tests); the rolling-mean benchmark is used only for constructing OoS R^2 .

We summarize rolling performance by reporting the median and interquartile range (IQR) of $\text{RMSE}_{t,\tau}^{(m)}$, $\text{MAE}_{t,\tau}^{(m)}$, $\text{MAD}_{t,\tau}^{(m)}$, and $R_{t,\tau}^2$ across the out-of-sample evaluation period.

The out-of-sample R^2 (OoS R^2) takes values in $(-\infty, 1]$. Negative values indicate that the benchmark forecast has lower rolling RMSE than the model, while positive values indicate that the model improves upon the benchmark in rolling RMSE. Values closer to one imply greater predictive accuracy relative to the benchmark.

For each metric (RMSE, MAE, and MAD), our objective is to minimize deviations between predictions and observations, thus obtaining the smallest possible values for these metrics given our choice of predictors, $\hat{y}_j$, $j = 1, \dots, n$. These metrics collectively provide a quantitative measure of the success of our predictions.

4. Data

Official data: The Official dataset includes Canadian economic variables dynamically selected for each forecast horizon and step from a large monthly database by Fortin-Gagnon et al. [7]. Using PDC-SIS for variable selection, relevant predictors are chosen based on their predictive power at each forecast step. Common variables selected across multiple horizons by the Random Forest model include the Toronto Stock Exchange high, six-month Treasury bills, Government bonds (1–3, 3–5, and 5–10 years), WTI crude oil price, IPPI (metal and machinery), Export energy products, Export oil, Import consumption goods, Residential mortgages, New housing price index, Employment services, Total employment, Total hours worked, and Manufacturing inventories. The variables used to predict monthly real GDP growth rates are summarized in Table 2.

GT data: This dataset is available at www.google.com/trends provides all search volumes and shows the percentage of web searches performed on Google. GT dataset 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. To address the well-documented sampling variation in Google Trends data, we follow the methodology proposed by Eichenauer et al. [36]. Specifically, we download multiple random samples of GT data at different periods and average the series to reduce sampling noise. This approach produces more consistent and reliable time series, mitigating the randomness inherent in Google's sampling procedure.

Google Trends indices are based on Google's sampling and normalization and may vary across downloads even when the same query settings are used. For transparency, our replication package archives the processed Google Trends panel used in estimation (after cleaning and seasonal adjustment), together with scripts documenting query settings (e.g., geography, category, time window, and term lists) and download dates recorded at the day level (YYYY-MM-DD). Consequently, the reported results are computationally reproducible from the archived data and code, while exact reconstruction by re-downloading raw Google Trends series from the public interface may not be perfectly identical.

Choi and Varian [1] have used the term “Google Trends” while Götz and Knetsch [4] and Ferrara and Simoni [5] 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 used the internet was 91%. Interestingly, approximately 84% of the internet users bought about \$57.4 billion worth of goods or services online in 2018 compared to \$18.9 billion in 2012. GT data are only available for normal users from January 2004 to date.

Table 2. Common Official data selected by the Random Forest model with PDC-SIS.

Time Horizon	Description
Market Indicators	
$\tau = 1$	Toronto Stock Exchange high Treasury bills, 6 months Government bonds 1–3 years Industrial production price index
$\tau = 3$	WTI crude oil price Treasury bills, Bank rate lag 1 Government bonds 3–5 years, Bank rate lag 2
$\tau = 6$	IPPI metal and construction lag 1–3 Government bonds 5–10 years, Bank rate
$\tau = 9$	Government bonds 1–3 years, Bank rate lag 3 Treasury bills, Bank rate lag 3 Government bonds 3–5 years, Bank rate lag 3 IPPI metal
Energy/Trade	
$\tau = 1$	Export energy products lag 2 Export oil lag 2
$\tau = 3$	Export oil Export energy products Import transportation lag 1 Export transportation lag 1
$\tau = 6$	Import consumption goods lag 3
$\tau = 9$	Export energy products lag 1 Import consumption goods Import oil lag 2 Total imports lag 3
Credit/Housing	
$\tau = 1$	Residential mortgages, Chartered banks Residential mortgages lag 1 New housing price index
Manufacturing/Industrial	
$\tau = 3$	IPPI machinery lag 2
$\tau = 6$	Manufacturing inventories total lag 2 Durable inventory ratio lag 3 Building permits industrial lag 2
Employment/Labor	
$\tau = 1$	Employment services lag 2
$\tau = 3$	Employment services Total employment
$\tau = 6$	Total hours worked lag 2 Employment finance lag 3
Consumer Prices	
$\tau = 9$	CPI health lag 1

Note: This table lists the top 10 features most frequently selected across 109 rolling windows by PDC-SIS variable selection with Official data. Features are grouped by economic category and organized by forecast horizon ($\tau = 1, 3, 6, 9$ months ahead).

We divided our GT dataset into two primary categories. The search keywords in the first category, which we term *general search terms*, were obtained through the R package *gtrendsR* (see Table 3). This initial set included 26 categories and 269 subcategories, yielding a total of 1776 search terms. Only 464 search terms remain after removing columns with missing values and outliers. The second category, referred to as *relevant search terms*, includes variables that directly align with the monthly and quarterly components of Canadian GDP (see Table 4). This category comprises 80 search terms. For variables in the second category with minimal missing values, we replaced up to two missing entries with the column mean. Following a thorough screening process that involved removing columns

with missing values, filtering outliers, and careful manual selection for relevance, 37 search terms were retained.

Table 3. Google Trends (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 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	

The manual relevance screen retained terms only if they had a clear link to Canadian aggregate activity (GDP components and/or industries represented in the monthly GDP accounts) and a stable interpretation over time. Terms were excluded if they were ambiguous in meaning, primarily non-economic in context, event-driven/one-off, or exhibited excessive missingness or unstable spikes inconsistent with broad economic activity. Importantly, this screening was conducted without reference to GDP outcomes or model performance.

Additionally, we applied the X-13ARIMA-SEATS program, developed by the U.S. Census Bureau and the Bank of Spain, to seasonally adjust GT data, ensuring consistency for use in economic forecasting.

5. Empirical Results

5.1. Selection of Predictors

In our approach, dynamic feature selection is applied in each rolling window at each forecast horizon, resulting in a diverse set of predictors tailored to the model's needs at each step. Given the complexity and volume of selected features, it is nearly impossible to present every feature used across all windows and horizons. Therefore, we focus on the features that were consistently selected across windows by PDC-SIS within each dataset, which allows us to highlight the most impactful variables in a manageable format. Therefore, we focus on the features that were consistently selected across windows by PDC-SIS within each dataset, which allows us to highlight the most impactful variables in a manageable format. For comparability, Tables 1, 2 and 5 report these consistently selected predictors using Random Forest as a fixed reference implementation.

Table 5. Common GT plus Official features selected by Random Forest with PDC-SIS.

Time Horizon	Data Type	Primary Features	Secondary Features
Short-term ($\tau = 1$)			
	Google Trends	Automotive sector, Audi Small business trends Electricity consumption	Defense sector, Army Food service activity
	Official data	Residential mortgages Treasury bills, 6 months Toronto Stock Exchange high Government bonds, 1–3 years Export energy products	Industrial production price index Employment services New housing price index Oil exports
Medium-term ($\tau = 3$)			
	Google Trends	Small business trends Electricity consumption Defense sector, Army	Rental market Loan activity
	Official data	WTI oil prices Oil exports Transportation imports Energy exports Employment services	Industrial machinery prices Total employment Toronto Stock Exchange high Government bonds, 3–5 years Treasury bills, Bank rate
Long-term ($\tau = 6$)			
	Google Trends	Finance sector Automotive sector, Volvo Mining sector Electricity consumption	Loan activity Services sector Agriculture sector
	Official data	Metal prices Consumer imports Manufacturing inventories Total hours worked	Employment finance Durables inventory ratio Government bonds, 5–10 years Bank rate
Very Long-term ($\tau = 9$)			
	Google Trends	Agriculture sector Unemployment Automotive sector, Volvo	Mining sector
	Official data	Energy exports Government bonds, 1–3 years Consumer imports Oil imports Treasury bills, Bank rate	Metal prices Total imports Government bonds, 5–10 years Unemployment Government bonds, 3–5 years

Note: This table lists the features that were consistently selected across 109 rolling windows by PDC-SIS variable selection with combined GT and Official data. Features are organized by forecast horizon with primary features appearing more frequently than secondary features.

5.2. Official Data

In the case of Official data, model leadership varies by horizon: CatBoost leads at short horizons ($\tau = 1, 3$), Random Forest at the medium horizon ($\tau = 6$), and GBM at the long horizon ($\tau = 9$). The predictors selected by PDC-SIS reveal distinct patterns across forecast horizons (see Table 2). In the short term ($\tau = 1$), forecasts rely heavily on credit and market variables, such as residential mortgages, treasury bills, stock exchange high, and energy exports, which capture immediate economic fluctuations. For medium-term horizons ($\tau = 3$), energy and employment variables, like WTI oil prices, energy exports, transportation imports, and employment services, gain prominence, reflecting the influence of global energy markets and labor conditions on economic activity. In longer-term forecasts ($\tau = 6$ and $\tau = 9$), industrial production and trade indicators dominate, with metal prices, manufacturing inventories, consumer imports, and government bond spreads highlighting the role of industrial activity and monetary policy in sustaining economic growth over time.

5.3. GT Data

In the case of GT data, LightGBM is the top-performing model. The features selected by PDC-SIS for the GT dataset display unique trends across forecast horizons (see Table 1). Short-term predictions ($\tau = 1$) rely heavily on automotive sector searches (Audi), small business trends, and electricity consumption, reflecting immediate economic activity in consumer and business sectors. Defense sector indicators (Army) show strong persistence across short and medium horizons, suggesting their sustained relevance in economic forecasting. Financial indicators, including loan and finance sector activity, gain prominence in longer horizons ($\tau = 6$), highlighting the role of credit conditions in medium-term growth. Meanwhile, at very long horizons ($\tau = 9$), automotive searches shift to Volvo and Ford, with insurance, unemployment, and agriculture sector activity becoming increasingly important in capturing structural economic trends.

5.4. Combined GT and Official Data

The combined analysis of GT and Official data highlights distinct feature patterns across forecast horizons (see Table 5). According to the forecasting results, CatBoost leads at short horizons ($\tau = 1, 3$), XGBoost at the medium horizon ($\tau = 6$), and GBM at the long horizon ($\tau = 9$). In the short term ($\tau = 1$), forecasts rely heavily on consumer behavior and financial indicators, with Google Trends capturing automotive sector activity (Audi), small business trends, and electricity consumption, while Official data emphasize residential mortgages, treasury bills, stock exchange high, and energy exports. As the horizon extends to the medium term ($\tau = 3$), a shift occurs toward energy and labor market indicators, with Google Trends focusing on small business trends, electricity consumption, and rental market activity, while Official data highlight WTI oil prices, energy exports, transportation imports, and employment services.

For longer horizons ($\tau = 6$), industrial and finance indicators gain importance, such as finance sector searches, automotive trends (Volvo), and mining sector activity in Google data, complemented by metal prices, consumer imports, manufacturing inventories, and total hours worked in Official data. In the very long term ($\tau = 9$), structural economic indicators become dominant, with Google Trends showing agriculture sector activity alongside unemployment and automotive searches (Volvo), while Official data emphasize energy exports, government bond spreads, consumer imports, and oil imports. This evolution across time horizons demonstrates how high-frequency Google Trends data combined with fundamental Official indicators allows short-term forecasts to align with immediate market conditions and consumer behavior, while long-term forecasts increasingly capture structural economic factors and monetary policy indicators. Notably, small business trends

and real estate activity in Google data maintain importance across all time horizons, suggesting their persistent relevance as economic barometers, while Official data show a clear progression from immediate market indicators toward broader monetary and inflation measures as the forecast horizon extends.

Yousuf and Feng [28]’s variable screening procedure based on Székely and Rizzo [37]’s partial distance correlation, namely PDC-SIS, can be used to select time-series variables. This procedure defines the set of possible predictors as

$$S_{k,l} = (Y_{t-1}, \dots, Y_{t-L}, X_{t-1,k}, \dots, X_{t-l+1,k})$$

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.

5.5. Comparison of Forecasts

Given the set of predictors selected in Section 5.1 above, we proceed with the sliding-window forecasting strategy described in Section 3. The following ML algorithms were implemented: *Random Forest*, *Gradient Boosting Machine* (GBM), *CatBoost*, *Light Gradient Boosting Machine* (LightGBM), and *XGBoost* described in Appendix A. We include an ARIMA model as the baseline comparator for forecast evaluation (including Diebold–Mariano tests). In addition, for constructing OoS R^2 , we use a simple rolling-mean benchmark, defined as the historical average of past GDP growth computed over a fixed-length backward-looking window. Our analysis reveals that no single data source or model dominates across all forecast horizons, with performance varying systematically by both horizon and data source.

We multiply the median of all metrics by 100 for better visibility, excluding OoS R^2 . The GT data is available from 2004 to 2021, so for a better comparison, we used Official datasets from 2004 to 2021. All the forecasts [$\tau = 1, 3, 6$, and 9 steps ahead] of the monthly GDP growth rate using both official and GT data for a period (the entire sample period for the Official data and GT is from January 2004 to September 2021, and the out-of-sample period is from August 2012 to September 2021 for Official and GT) are presented in Tables 6–8.

The first column of the table presents the different forecast horizons, while the second column lists the performance metrics, including RMSE, MAE, and OoS R^2 . The remaining columns show the median values of rolling RMSE, and MAE for each machine learning algorithm, with variable selection conducted using the PDC-SIS method. Beneath each median value, dispersion is reported using the interquartile range (IQR), which represents the difference between the first and third quartiles. A model with either a lower median RMSE or MAE or MAD, or a lower IQR, is considered to have better performance. The rolling RMSE and MAE for Official data (2004–2021), and Official plus GT data and Google Trends data, are shown in Figures 2–4. A boxplot shows the five-number summary of a set of data: the minimum value, first quartile (25 percent), median, third quartile (75 percent), and maximum values. Using a vertical number line and a rectangular box to build a box plot, the minimum and maximum values are the endpoints of the axis. The first quartile displays one end of the box, and the third quartile the other end of the box.

The box shows the interquartile range, where a horizontal line goes through the box at the median or second quartile. The outliers are displayed outside the whisker (the whiskers are the two lines outside the box). Here are the key takeaways:

Table 6. Official data variable selection with PDC-SIS: Forecast evaluation using the median of rolling out-of-sample (OoS) RMSE, MAE, and OoS R^2 with a window size of 75 months (August 2012 to September 2021; full sample: 2004–2021).

Horizon	Metric	GBM	XGBoost	CatBoost	RF	LightGBM	ARIMA
$\tau = 1$	RMSE	0.2747 (0.1112)	0.2941 (0.0954)	0.2729 (0.0973)	0.2755 (0.0989)	0.3006 (0.1069)	0.3064 (0.1422)
	MAD	0.2157 (0.0889)	0.2280 (0.1151)	0.1944 (0.0935)	0.1949 (0.1195)	0.2499 (0.0636)	0.1917 (0.0868)
	MAE	0.2356 (0.0935)	0.2373 (0.0885)	0.2252 (0.0886)	0.2226 (0.0865)	0.2589 (0.0753)	0.2506 (0.1194)
	OoS R^2	0.3221 (0.2585)	0.2919 (0.2833)	0.3138 (0.1364)	0.3282 (0.2412)	0.1586 (0.1899)	0.0383 (0.2155)
$\tau = 3$	RMSE	0.2717 (0.0601)	0.2699 (0.0628)	0.2643 (0.0709)	0.2712 (0.0799)	0.2841 (0.0914)	0.3041 (0.1456)
	MAD	0.1833 (0.0846)	0.2012 (0.0591)	0.1948 (0.0811)	0.2151 (0.0677)	0.2222 (0.0876)	0.2133 (0.0971)
	MAE	0.2177 (0.0658)	0.2191 (0.0545)	0.2156 (0.0612)	0.2235 (0.0625)	0.2442 (0.0684)	0.2536 (0.1057)
	OoS R^2	0.2006 (0.2766)	0.2689 (0.3151)	0.2621 (0.2570)	0.2782 (0.2002)	0.0697 (0.6330)	0.0623 (0.1578)
$\tau = 6$	RMSE	0.2680 (0.0874)	0.2616 (0.0987)	0.2688 (0.0727)	0.2603 (0.0927)	0.2967 (0.1053)	0.3029 (0.1351)
	MAD	0.2121 (0.0847)	0.1980 (0.0592)	0.2131 (0.0612)	0.2109 (0.0424)	0.1986 (0.1073)	0.2246 (0.0949)
	MAE	0.2275 (0.0660)	0.2181 (0.0726)	0.2213 (0.0661)	0.2214 (0.0639)	0.2415 (0.1008)	0.2571 (0.0984)
	OoS R^2	0.2600 (0.2928)	0.2988 (0.3246)	0.2875 (0.2854)	0.3177 (0.2124)	0.1966 (0.2739)	0.0711 (0.1492)
$\tau = 9$	RMSE	0.2722 (0.1083)	0.2795 (0.1052)	0.2965 (0.0879)	0.2834 (0.0791)	0.2970 (0.1090)	0.3045 (0.1237)
	MAD	0.2199 (0.0887)	0.2179 (0.0716)	0.2263 (0.1035)	0.2141 (0.0677)	0.1868 (0.1301)	0.2247 (0.1206)
	MAE	0.2406 (0.0775)	0.2394 (0.0780)	0.2390 (0.0914)	0.2328 (0.0696)	0.2341 (0.0960)	0.2574 (0.0985)
	OoS R^2	0.2695 (0.2706)	0.2400 (0.2655)	0.2142 (0.3201)	0.2504 (0.2335)	0.1021 (0.5596)	0.0671 (0.1093)

Note: The table reports, for each model (columns), the median of the rolling RMSE, MAD, MAE (all scaled by $\times 100$), and OoS R^2 at forecast horizons of one, three, six, and nine steps ahead. Values in parentheses report dispersion measured by the interquartile range (IQR) of the corresponding rolling series.

1. Higher Accuracy with Combined Data:

Tables 6–8 show that combining Official + GT data delivers competitive accuracy at short horizons with the clearest gain emerging at the long horizon, while Official data alone remain best at short and medium horizons. At $\tau = 1$, the best Official model is CatBoost with median RMSE 0.2729 (Table 6), marginally lower than the best combined-data RMSE 0.2733 (CatBoost, Table 8) and substantially lower than the best GT-only RMSE 0.3008 (LightGBM, Table 7). At $\tau = 3$, Official again dominates (CatBoost: 0.2643, Table 6) relative to combined (0.2688 by CatBoost, Table 8) and GT-only (0.3021 by LightGBM, Table 7). At $\tau = 6$, Official delivers the lowest RMSE (Random Forest: 0.2603, Table 6), with combined close behind (0.2616 by XGBoost, Table 8), while GT-only remains substantially higher (0.3026 by LightGBM, Table 7). The clearest benefit from combining data emerges at $\tau = 9$, where combined achieves the lowest RMSE overall (0.2710 by GBM, Table 8) versus Official (0.2722 by GBM, Table 6) and

GT-only (0.3020 by LightGBM, Table 7). In terms of out-of-sample R^2 (defined relative to the rolling-mean benchmark in Equation (4)), Official data remain consistently strong across all horizons (Table 6). GT-only shows substantially weaker performance, with most models registering negative OoS R^2 ; only LightGBM and ARIMA maintain positive values across horizons (Table 7). The combined dataset achieves its highest OoS R^2 at $\tau = 1$ (CatBoost: 0.3366, Table 8), exceeding Official data's best at this horizon (RF: 0.3282, Table 6).

Table 7. GT data variable selection with PDC-SIS: Forecast evaluation using the median of rolling out-of-sample (OoS) RMSE, MAE, and OoS R^2 with a window size of 75 months (August 2012 to September 2021; full sample: 2004–2021).

Horizon	Metric	GBM	XGBoost	CatBoost	RF	LightGBM	ARIMA
$\tau = 1$	RMSE	0.3375 (0.1804)	0.3494 (0.1313)	0.3480 (0.1496)	0.3201 (0.1180)	0.3008 (0.1472)	0.3064 (0.1422)
	MAD	0.2481 (0.1255)	0.2558 (0.0969)	0.2399 (0.1215)	0.2323 (0.0848)	0.2210 (0.0902)	0.1917 (0.0868)
	MAE	0.2688 (0.1156)	0.2861 (0.1008)	0.2889 (0.0912)	0.2702 (0.0799)	0.2507 (0.1047)	0.2506 (0.1194)
	OoS R^2	−0.1221 (0.5153)	−0.2973 (0.5120)	−0.1143 (0.3303)	−0.0690 (0.2368)	0.0755 (0.1220)	0.0383 (0.2155)
$\tau = 3$	RMSE	0.3256 (0.1873)	0.3205 (0.1724)	0.3532 (0.1383)	0.3162 (0.1523)	0.3021 (0.1502)	0.3041 (0.1456)
	MAD	0.2009 (0.1296)	0.2332 (0.1234)	0.2170 (0.1514)	0.2112 (0.1175)	0.2252 (0.0978)	0.2133 (0.0971)
	MAE	0.2669 (0.1100)	0.2679 (0.1142)	0.2872 (0.1163)	0.2640 (0.1069)	0.2513 (0.1042)	0.2536 (0.1057)
	OoS R^2	−0.2860 (0.4518)	−0.3256 (0.2770)	−0.2078 (0.2801)	−0.1450 (0.2291)	0.0821 (0.1214)	0.0623 (0.1578)
$\tau = 6$	RMSE	0.3355 (0.1428)	0.3192 (0.1668)	0.3377 (0.1169)	0.3146 (0.1557)	0.3026 (0.1338)	0.3029 (0.1351)
	MAD	0.2310 (0.0910)	0.2454 (0.1204)	0.2392 (0.0821)	0.2525 (0.0819)	0.2218 (0.1162)	0.2246 (0.0949)
	MAE	0.2699 (0.1054)	0.2691 (0.1211)	0.2828 (0.0877)	0.2716 (0.0957)	0.2564 (0.1013)	0.2571 (0.0984)
	OoS R^2	−0.1561 (0.4628)	−0.1507 (0.4887)	−0.0784 (0.1836)	−0.0845 (0.3385)	0.0897 (0.1314)	0.0711 (0.1492)
$\tau = 9$	RMSE	0.3397 (0.1800)	0.3193 (0.1645)	0.3249 (0.1395)	0.3330 (0.1686)	0.3020 (0.1238)	0.3045 (0.1237)
	MAD	0.2598 (0.1293)	0.2759 (0.0973)	0.2483 (0.0701)	0.2630 (0.1239)	0.2272 (0.1116)	0.2247 (0.1206)
	MAE	0.2801 (0.1389)	0.2831 (0.1267)	0.2810 (0.1026)	0.2758 (0.1282)	0.2561 (0.0959)	0.2574 (0.0985)
	OoS R^2	−0.1974 (0.5084)	−0.1397 (0.4472)	−0.0237 (0.2492)	−0.0873 (0.3660)	0.0891 (0.1405)	0.0671 (0.1093)

Note: The table reports, for each model (columns), the median of the rolling RMSE, MAD, MAE (all scaled by $\times 100$), and OoS R^2 at forecast horizons of one, three, six, and nine steps ahead. Values in parentheses report dispersion measured by the interquartile range (IQR) of the corresponding rolling series.

2. Official Data Performance:

Official data (Table 6) deliver the strongest and most consistent accuracy profile overall, with positive OoS R^2 at all horizons for all machine learning models. By RMSE, the best models are CatBoost at $\tau = 1$ (0.2729) and $\tau = 3$ (0.2643), Random Forest at $\tau = 6$ (0.2603), and GBM at $\tau = 9$ (0.2722). By out-of-sample R^2 (relative to the rolling-mean benchmark in Equation (4)), Random Forest achieves the highest values across most horizons: 0.3282 at $\tau = 1$ (ARIMA's out-of-sample R^2 is 0.0383), 0.2782 at $\tau = 3$ (ARIMA's out-of-sample R^2 is 0.0623), and 0.3177 at $\tau = 6$ (ARIMA's out-of-sample R^2 is 0.0711), while GBM leads at $\tau = 9$ with 0.2695 (ARIMA's out-of-sample R^2 is

0.0671). Diebold–Mariano tests (Table 9) confirm CatBoost’s superiority at $\tau = 1$ and 3, Random Forest’s at $\tau = 6$, and GBM’s at $\tau = 9$, with the leading model significantly outperforming ARIMA at each horizon.

3. GT Data Effectiveness:

GT data alone (Table 7) yield materially higher forecast errors than Official or combined data and exhibit substantially weaker overall fit, with most models failing to outperform the ARIMA baseline model. LightGBM emerges as the most consistent GT-only performer, achieving the lowest RMSE at all horizons: $\tau = 1$ (0.3008), $\tau = 3$ (0.3021), $\tau = 6$ (0.3026), and $\tau = 9$ (0.3020), compared to ARIMA’s 0.3064, 0.3041, 0.3029, and 0.3045, respectively. Most non-(LightGBM/ARIMA) models register negative OoS R^2 , while LightGBM maintains modest positive values across horizons: 0.0755 at $\tau = 1$ (vs. ARIMA’s 0.0383), 0.0821 at $\tau = 3$ (vs. 0.0623), 0.0897 at $\tau = 6$ (vs. 0.0711), and 0.0891 at $\tau = 9$ (vs. ARIMA’s 0.0671). Diebold–Mariano tests (Table 10) confirm LightGBM’s statistical dominance across all horizons, significantly outperforming all ML competitors.

Table 8. Official + GT data variable selection with PDC-SIS: Forecast evaluation using the median of rolling out-of-sample (OoS) RMSE, MAE, and OoS R^2 with a window size of 75 months (August 2012 to September 2021; full sample: 2004–2021).

Horizon	Metric	GBM	XGBoost	CatBoost	RF	LightGBM	ARIMA
$\tau = 1$	RMSE	0.2747 (0.1097)	0.2941 (0.0902)	0.2733 (0.0916)	0.2770 (0.0984)	0.2980 (0.1250)	0.3064 (0.1422)
	MAD	0.2157 (0.0714)	0.2260 (0.1083)	0.1963 (0.0939)	0.1962 (0.0792)	0.2495 (0.0750)	0.1917 (0.0868)
	MAE	0.2341 (0.0924)	0.2389 (0.0806)	0.2251 (0.0837)	0.2250 (0.0883)	0.2584 (0.0941)	0.2506 (0.1194)
	OoS R^2	0.3040 (0.2768)	0.2778 (0.2237)	0.3366 (0.1380)	0.3286 (0.1554)	0.1703 (0.1420)	0.0383 (0.2155)
$\tau = 3$	RMSE	0.2770 (0.0876)	0.2709 (0.0742)	0.2688 (0.0979)	0.2712 (0.0981)	0.3108 (0.0860)	0.3041 (0.1456)
	MAD	0.2067 (0.1083)	0.2000 (0.0538)	0.2050 (0.0945)	0.2151 (0.0746)	0.2283 (0.0935)	0.2133 (0.0971)
	MAE	0.2248 (0.0711)	0.2204 (0.0473)	0.2267 (0.0673)	0.2249 (0.0600)	0.2604 (0.0642)	0.2536 (0.1057)
	OoS R^2	0.1074 (0.2058)	0.1618 (0.2811)	0.1545 (0.2198)	0.2317 (0.2013)	0.0657 (0.5722)	0.0623 (0.1578)
$\tau = 6$	RMSE	0.2845 (0.0871)	0.2616 (0.0812)	0.2725 (0.0746)	0.2637 (0.0951)	0.3018 (0.0972)	0.3029 (0.1351)
	MAD	0.2094 (0.0909)	0.1992 (0.0677)	0.2129 (0.0590)	0.2109 (0.0464)	0.2109 (0.1279)	0.2246 (0.0949)
	MAE	0.2322 (0.0688)	0.2196 (0.0634)	0.2220 (0.0536)	0.2215 (0.0632)	0.2421 (0.0858)	0.2571 (0.0984)
	OoS R^2	0.2610 (0.2610)	0.2588 (0.3608)	0.2373 (0.2681)	0.2733 (0.2831)	0.1658 (0.2998)	0.0711 (0.1492)
$\tau = 9$	RMSE	0.2710 (0.1066)	0.2782 (0.1022)	0.2965 (0.0852)	0.2837 (0.0765)	0.2881 (0.1113)	0.3045 (0.1237)
	MAD	0.2182 (0.0746)	0.2179 (0.0654)	0.2274 (0.0805)	0.2146 (0.0462)	0.1871 (0.1054)	0.2247 (0.1206)
	MAE	0.2356 (0.0679)	0.2394 (0.0680)	0.2398 (0.0794)	0.2408 (0.0691)	0.2339 (0.0931)	0.2574 (0.0985)
	OoS R^2	0.1822 (0.2783)	0.2171 (0.2611)	0.1495 (0.3201)	0.2026 (0.2573)	0.2074 (0.5465)	0.0671 (0.1093)

Note: The table reports, for each model (columns), the median of the rolling RMSE, MAD, MAE (all scaled by $\times 100$), and OoS R^2 at forecast horizons of one, three, six, and nine steps ahead. Values in parentheses report dispersion measured by the interquartile range (IQR) of the corresponding rolling series.

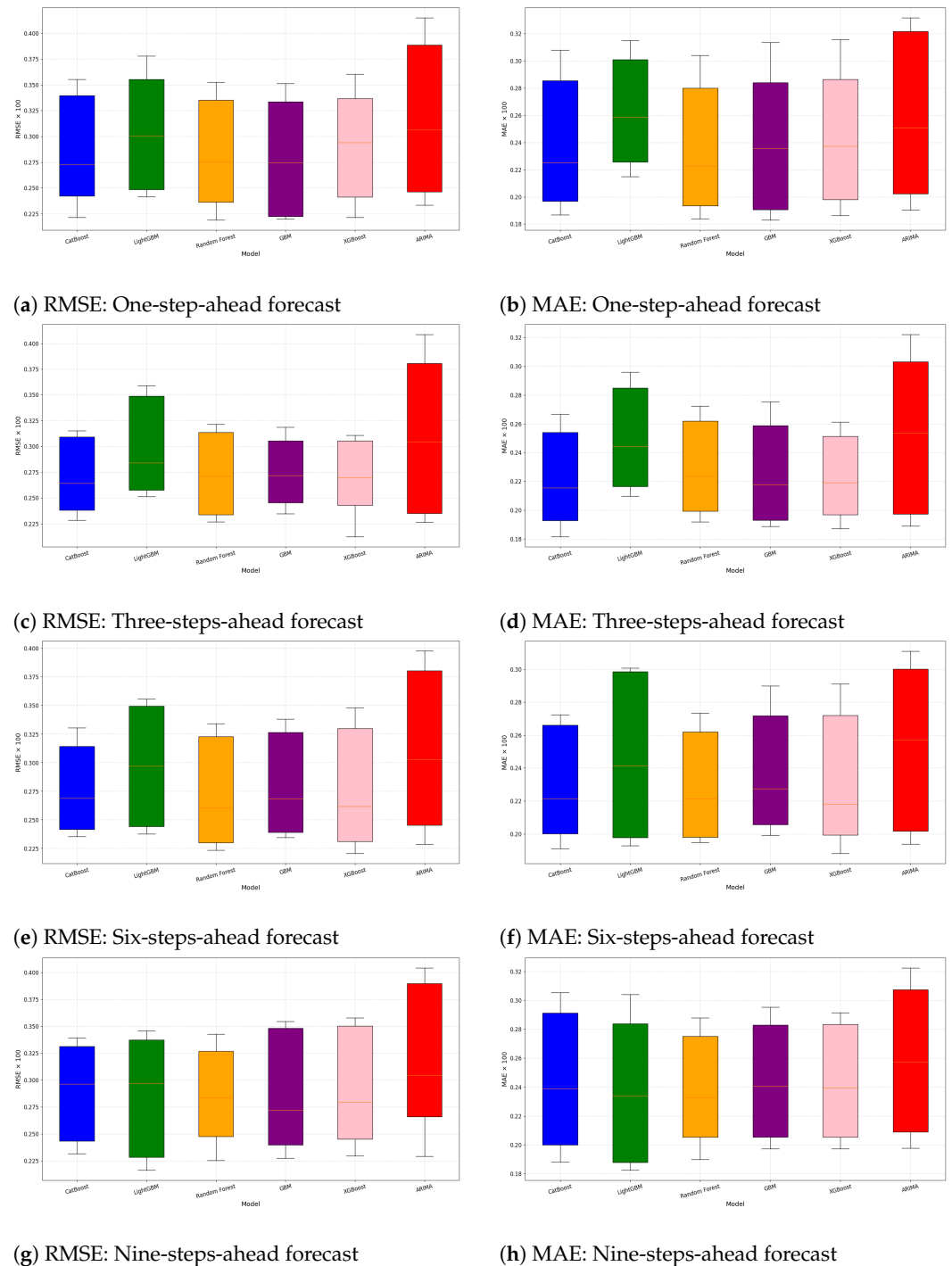


Figure 2. Rolling out-of-sample RMSE and MAE for different forecast horizons using PDC-SIS variable selection with official data. Note: The left panel shows the rolling RMSE and the right panel shows the rolling MAE for different forecast horizons ($\tau = 1, 3, 6, 9$) using PDC-SIS variable selection with Official data.

4. Combining GT with Official Data:

Table 8 demonstrates that combining GT with Official data is competitive at short horizons and provides the clearest RMSE improvement at the long horizon, though the gains are often statistically indistinguishable from Official data alone. By RMSE, the best combined-data models are CatBoost at $\tau = 1$ (0.2733) and $\tau = 3$ (0.2688), XGBoost at $\tau = 6$ (0.2616), and GBM at $\tau = 9$ (0.2710). In terms of OoS R^2 (relative to the rolling-mean benchmark in Equation (4)), combined data achieve the strongest

performance at $\tau = 1$ where CatBoost attains 0.3366 (ARIMA's out-of-sample R^2 is 0.0383), exceeding Official data's best of 0.3282 (RF, Table 6). At $\tau = 3$, CatBoost achieves 0.1545 (vs. ARIMA's 0.0623), at $\tau = 6$, XGBoost reaches 0.2588 (vs. 0.0711), and at $\tau = 9$, GBM attains 0.1822 (vs. 0.0671). While the combined dataset achieves the lowest absolute RMSE at $\tau = 9$ (0.2710), Official data maintain higher OoS R^2 values at $\tau = 3, 6$, and 9 (Table 6 vs. Table 8). Diebold–Mariano tests (Table 11) confirm the best models' superiority: CatBoost at $\tau = 1$ and 3, XGBoost at $\tau = 6$, and GBM at $\tau = 9$, with leading models significantly outperforming ARIMA.

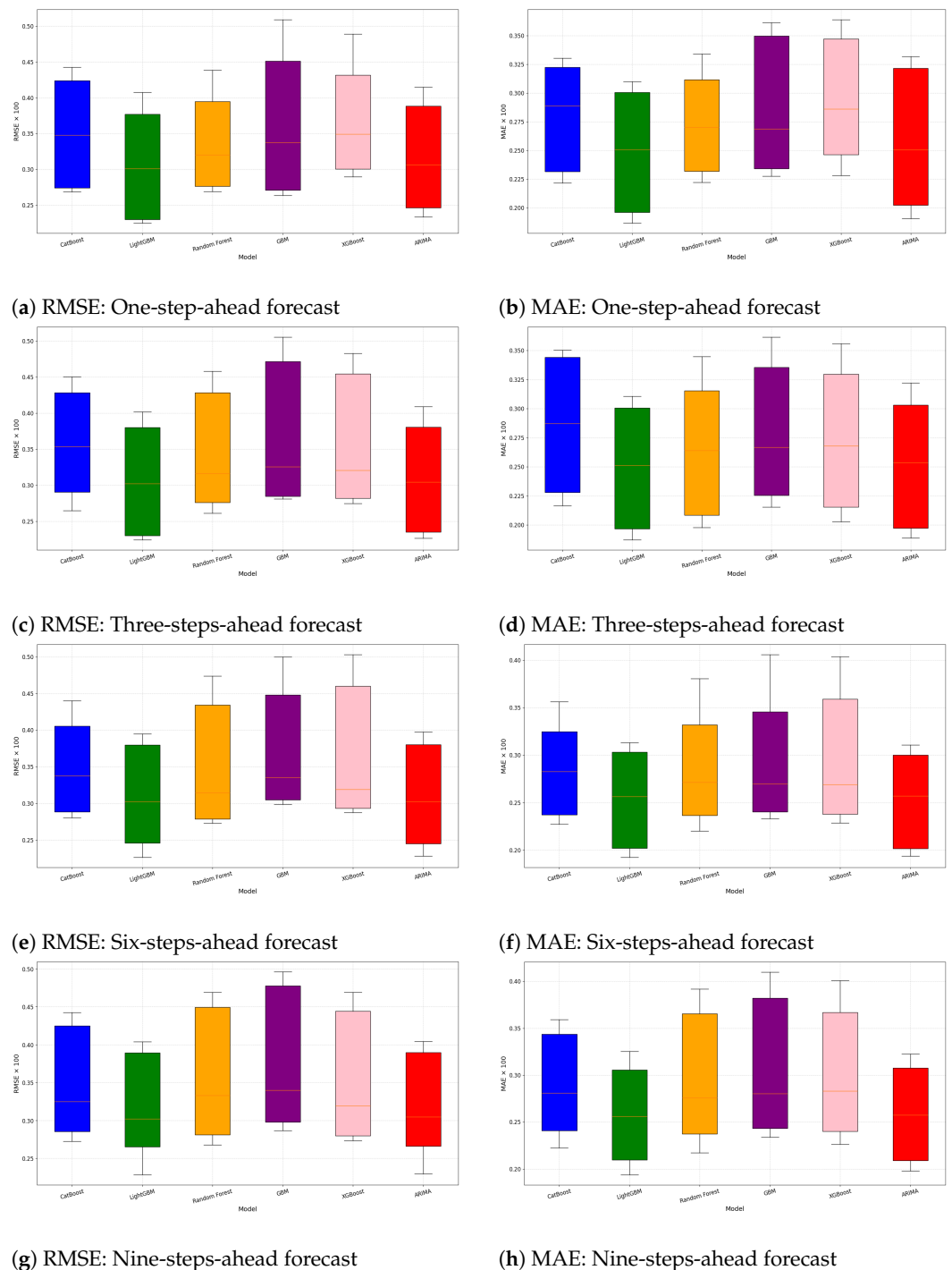


Figure 3. Rolling out-of-sample RMSE and MAE for different forecast horizons using PDC-SIS variable selection with GT data. Note: The left panel shows the rolling RMSE and the right panel shows the rolling MAE for different forecast horizons ($\tau = 1, 3, 6, 9$) using PDC-SIS variable selection with GT data.

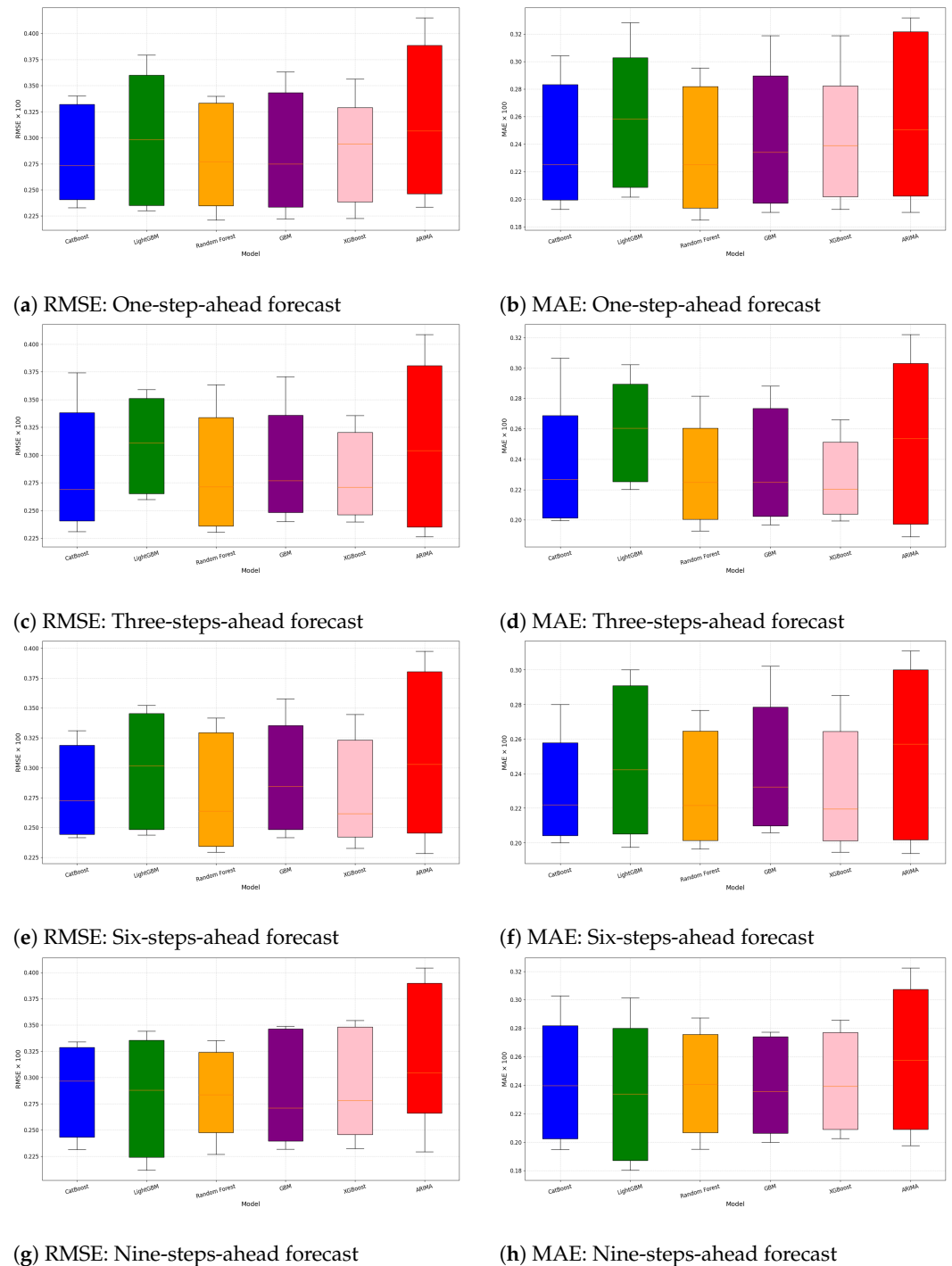


Figure 4. Rolling out-of-sample RMSE and MAE for different forecast horizons using PDC-SIS variable selection with GT plus Official data. Note: The left panel shows the rolling RMSE and the right panel shows the rolling MAE for different forecast horizons ($\tau = 1, 3, 6, 9$) using PDC-SIS variable selection with GT plus Official data.

Table 9. Diebold–Mariano Test Results (Official Data): Pairwise forecast accuracy comparisons. Model 1 shows significantly better performance than the listed Model 2 competitors.

Horizon	Model 1	Model 2 (Significantly Outperformed)
$\tau = 1$	CatBoost	LightGBM ***, ARIMA ***
$\tau = 3$	CatBoost	GBM ***, XGBoost ***, RF *, LightGBM ***, ARIMA ***
$\tau = 6$	Random Forest	LightGBM ***, ARIMA ***
$\tau = 9$	GBM	LightGBM *, ARIMA ***

Notes: Significance levels: * $p < 0.10$ and *** $p < 0.01$. Models not listed show no significant differences. Full pairwise results are reported in Table A1 in Appendix A.

Table 10. Diebold–Mariano Test Results (GT Data): Pairwise forecast accuracy comparisons. Model 1 shows significantly better performance than the listed Model 2 competitors.

Horizon	Model 1	Model 2 (Significantly Outperformed)
$\tau = 1$	LightGBM	GBM ***, XGBoost ***, CatBoost ***, RF **
$\tau = 3$	LightGBM	GBM ***, XGBoost ***, CatBoost ***, RF **
$\tau = 6$	LightGBM	GBM ***, XGBoost ***, CatBoost ***, RF **
$\tau = 9$	LightGBM	GBM ***, XGBoost ***, CatBoost ***, RF **

Notes: Significance levels: ** $p < 0.05$ and *** $p < 0.01$. LightGBM demonstrates statistical superiority across all horizons with Google Trends data. LightGBM shows no significant difference vs. ARIMA. Full pairwise results are reported in Table A2 in Appendix A.

Table 11. Diebold–Mariano Test Results (Official + GT Combined Data): Pairwise forecast accuracy comparisons. Model 1 shows significantly better performance than the listed Model 2 competitors.

Horizon	Model 1	Model 2 (Significantly Outperformed)
$\tau = 1$	CatBoost	LightGBM ***, ARIMA ***
$\tau = 3$	CatBoost	GBM **, LightGBM ***, ARIMA ***
$\tau = 6$	XGBoost	GBM *, LightGBM ***, ARIMA ***
$\tau = 9$	GBM	ARIMA ***

Notes: Significance levels: * $p < 0.10$, ** $p < 0.05$, and *** $p < 0.01$. Models not listed show no significant differences. Full pairwise results are reported in Table A3 in Appendix A.

5. Model Performance Comparison Across Forecast Horizons:

Model dominance proves primarily data-source-specific rather than universally stable across datasets. For Official data (Table 6), tree-based ensemble methods dominate: CatBoost leads by RMSE at short horizons ($\tau = 1$ and 3), Random Forest at the medium horizon ($\tau = 6$), and GBM at the long horizon ($\tau = 9$). For GT data (Table 7), LightGBM is the sole consistent performer, achieving the lowest RMSE at all horizons and (alongside ARIMA) generating the only positive OoS R^2 results. For combined data (Table 8), RMSE leaders mirror the Official pattern at short horizons (CatBoost at $\tau = 1$ and 3), shifting to XGBoost at $\tau = 6$, and to GBM at $\tau = 9$. Diebold–Mariano tests (Tables 9–11) confirm the statistical significance of the RMSE rankings, and Table 12 consolidates the horizon-specific winners across data sources. LightGBM dominates with GT data across all horizons, while with Official and combined data, CatBoost leads at short horizons and different models excel at longer horizons, with the leading model significantly outperforming ARIMA at each horizon.

Table 12. Diebold–Mariano Test Summary: Best performing models by data source and forecast horizon.

Horizon	Official Data	GT Data	Official + GT
$\tau = 1$	CatBoost ***	LightGBM ***	CatBoost ***
$\tau = 3$	CatBoost ***	LightGBM ***	CatBoost ***
$\tau = 6$	Random Forest ***	LightGBM ***	XGBoost ***
$\tau = 9$	GBM *	LightGBM ***	GBM ***
Overall Pattern	Mixed	LightGBM	Mixed

Notes: * $p < 0.10$ and *** $p < 0.01$. Stars indicate significant outperformance in Diebold–Mariano tests. Bold indicates dominance (significantly beats the majority of competitors). LightGBM shows complete dominance with GT data across all horizons. With Official and Combined data, model superiority varies by horizon, with no single model dominating all horizons.

6. Stability Analysis Using IQR Metrics:

The interquartile ranges (values in parentheses) reveal that GT-only forecasts exhibit substantially higher variability across rolling windows than Official or combined data at all horizons. At $\tau = 1$ RMSE, Official CatBoost displays IQR 0.0973 (Table 6), and combined CatBoost is marginally tighter with IQR 0.0916 (Table 8), while GT

LightGBM shows higher dispersion with IQR 0.1472 (Table 7). This pattern persists across horizons: at $\tau = 6$ RMSE, the combined best model (XGBoost: IQR 0.0812, Table 8) proves more stable than GT LightGBM (IQR 0.1338, Table 7). Similar patterns emerge in OoS R^2 metrics, where GT models display wider IQR spreads than Official or combined counterparts. The elevated variability in GT forecasts underscores the challenges of forecasting with search-based indicators: despite LightGBM's consistent RMSE advantage, the higher rolling-window variance in GT data reflects greater instability in the underlying signal.

A more detailed analysis of Tables 6–8 also permits a comparison of the predictive success of models depending on the different forecast horizons and different datasets.

One-step-ahead forecasts: Models trained on Official data (Table 6) show strong positive OoS R^2 values across all models, with RF achieving 0.3282, followed by GBM at 0.3221 and CatBoost at 0.3138. CatBoost achieves the lowest RMSE of 0.2729, closely followed by GBM at 0.2747. In contrast, GT data alone (Table 7) produces weak performance, with LightGBM achieving the lowest RMSE of 0.3008 and a modest OoS R^2 of 0.0755, while most other models show negative OoS R^2 values. For the combined dataset (Table 8), performance improves notably: CatBoost achieves the highest OoS R^2 of 0.3366 and lowest RMSE of 0.2733, followed by GBM (OoS R^2 0.3040, RMSE 0.2747). These results confirm that combining Official and GT data yields competitive short-term forecasting performance, with the combined dataset achieving the highest OoS R^2 observed. Diebold–Mariano tests (see Tables A1–A3 in Appendix A for complete pairwise results) reveal distinct patterns across data sources. For Official data, CatBoost significantly outperforms LightGBM and ARIMA at the 1% level, while showing no significant differences versus GBM, XGBoost, or RF, indicating these top models are statistically tied despite CatBoost's lower average RMSE. For GT data, LightGBM significantly outperforms all ML competitors at the 1% level (except RF at the 5% level), while maintaining statistical parity with ARIMA. For Combined data, CatBoost significantly beats LightGBM and ARIMA at the 1% level, mirroring the Official data pattern.

Three-steps-ahead forecasts: Official data maintains strong predictive accuracy (Table 6), with CatBoost achieving the lowest RMSE of 0.2643 and OoS R^2 of 0.2621, followed by XGBoost (RMSE 0.2699, OoS R^2 0.2689) and RF (RMSE 0.2712, OoS R^2 0.2782). For GT data alone (Table 7), LightGBM performs best with a RMSE of 0.3021 and OoS R^2 of 0.0821, while most other models show negative or near-zero out-of-sample R^2 values. The combined dataset (Table 8) shows CatBoost leading with a RMSE of 0.2688 and OoS R^2 of 0.1545, followed by XGBoost (RMSE 0.2709, OoS R^2 0.1618). Diebold–Mariano tests (Tables A1–A3) show CatBoost's strongest dominance at this horizon. For Official data, CatBoost significantly outperforms all competitors at conventional significance levels: most at the 1% level, with RF at the 10% level. For GT data, LightGBM significantly outperforms all four ML competitors at the 1% level, demonstrating complete statistical superiority. For Combined data, CatBoost significantly beats GBM at the 5% level, and LightGBM and ARIMA at the 1% level, though it shows no separation from XGBoost or RF.

Six-steps-ahead forecasts: Official data continue to deliver strong performance (Table 6), with RF achieving the lowest RMSE of 0.2603 and highest OoS R^2 of 0.3177, followed by XGBoost (RMSE 0.2616, OoS R^2 0.2988) and CatBoost (RMSE 0.2688, OoS R^2 0.2875). For GT data alone in Table 7, LightGBM maintains its lead with a RMSE of 0.3026 and OoS R^2 of 0.0897, while other models show substantially weaker performance with negative OoS R^2 values. The combined dataset (Table 8) shows XGBoost achieving the lowest RMSE of 0.2616 with an OoS R^2 of 0.2588, followed by RF (RMSE 0.2637, OoS R^2 0.2733). Diebold–Mariano tests (Tables A1–A3) reveal data-source-dependent patterns. For Official data, RF significantly outperforms LightGBM and ARIMA at the 1% level,

while showing no significant differences versus GBM, XGBoost, or CatBoost. For GT data, LightGBM achieves complete dominance, significantly outperforming all ML competitors at the 1% level. For Combined data, XGBoost marginally outperforms GBM at the 10% level and significantly beats LightGBM and ARIMA at the 1% level, with no significant differences versus CatBoost or RF.

Nine-steps-ahead forecasts: At the longest horizon, Official data shows GBM achieving the lowest RMSE of 0.2722 with an OoS R^2 of 0.2695 (Table 6), followed by XGBoost (RMSE 0.2795, OoS R^2 0.2400). For GT data alone (Table 7), LightGBM continues its consistent performance with a RMSE of 0.3020 and OoS R^2 of 0.0891, while other models deteriorate substantially. The combined dataset (Table 8) achieves the lowest overall RMSE across all specifications at this horizon: GBM records 0.2710 with an OoS R^2 of 0.1822, closely followed by XGBoost (RMSE 0.2782, OoS R^2 0.2171). Diebold–Mariano tests (Tables A1–A3) show convergence to statistical parity among top ML models at the longest horizon. For Official data, GBM marginally outperforms LightGBM at the 10% level and significantly beats ARIMA at the 1% level, with all other ML models statistically tied. For GT data, LightGBM maintains its dominance, significantly outperforming all four ML competitors at the 1% level. For Combined data, GBM significantly outperforms only ARIMA at the 1% level, with all ML models showing no significant differences among themselves, indicating comparable long-horizon performance.

In summary, our comprehensive analysis reveals robust machine-learning superiority over the ARIMA baseline model in RMSE-based evaluations and Diebold–Mariano tests when using Official economic data; additionally, all models achieve positive out-of-sample R^2 relative to the rolling-mean benchmark (Equation (4)) across horizons. Official data consistently deliver the strongest performance at short and medium horizons, while the combined Official and GT dataset achieves the highest short-horizon OoS R^2 (0.3366 at $\tau = 1$) and lowest long-horizon RMSE (0.2710 at $\tau = 9$). GT data alone show substantially weaker performance, with only LightGBM maintaining modest positive out-of-sample R^2 values across all horizons (relative to the rolling-mean benchmark in Equation (4)), while most other machine-learning models exhibit higher RMSE than ARIMA. The Diebold–Mariano tests Appendix A, Tables A1–A3) reveal data-source-dependent statistical patterns: LightGBM demonstrates complete statistical dominance with GT data across all horizons (significant at the 1% level in most comparisons), while with Official and Combined data, the best-performing models show significant separation from weaker competitors but statistical parity among themselves at conventional significance levels.

The visual analysis in Figure 2 presents the rolling out-of-sample RMSE and MAE distributions for Official data using PDC-SIS variable selection. At the one-step horizon, CatBoost achieves the lowest median RMSE (approximately 0.27), with a relatively compact interquartile range indicating stable performance, with a relatively compact interquartile range indicating stable performance. ARIMA shows the highest median RMSE; among the machine learning models, LightGBM tends to have higher errors and wider dispersion. For MAE, a similar pattern emerges with RF demonstrating the most favorable distribution. At the three-step horizon, CatBoost achieves the lowest median RMSE (approximately 0.26), while LightGBM and ARIMA exhibit higher medians and wider distributions. At the six-step horizon, RF achieves the lowest median RMSE (approximately 0.26), while at the nine-step horizon, GBM achieves the lowest median RMSE (approximately 0.27), with other models showing higher values. Notably, the boxplot heights (IQR) increase at $\tau = 3$ compared to other horizons, indicating greater forecast uncertainty at this intermediate horizon. These visual patterns align with the Diebold–Mariano test results confirming CatBoost’s statistical superiority at $\tau = 3$ and GBM’s at $\tau = 9$ (Official data).

The visual analysis in Figure 3 illustrates the rolling RMSE and MAE distributions when using GT data alone with PDC-SIS variable selection. At the one-step horizon, LightGBM achieves the lowest median RMSE (approximately 0.30), followed by ARIMA (approximately 0.31) and Random Forest, while CatBoost, GBM, and XGBoost show higher medians. As the forecast horizon extends to three steps, LightGBM achieves the lowest median RMSE around 0.31, while CatBoost exhibits the highest median with considerable variability. At the six-step horizon, LightGBM maintains the lowest median RMSE (with ARIMA close behind), while the remaining ML models exhibit higher medians; dispersion remains comparatively elevated for several GT-only specifications. At the nine-step horizon, LightGBM achieves the lowest median RMSE around 0.30, while GBM shows the highest median and widest distribution. The MAE panels reveal similar patterns, with CatBoost and LightGBM consistently showing lower medians and tighter distributions compared to GBM and XGBoost. These findings corroborate the Diebold–Mariano test results showing LightGBM’s statistical superiority over the other ML models across horizons (with statistical parity versus ARIMA), and highlight that GT data alone produce more variable forecasts compared to Official data.

The analysis of the combined GT and Official data, as illustrated in Figure 4, reveals more stable forecast distributions compared to either data source alone. At the one-step horizon, CatBoost achieves the lowest median RMSE (approximately 0.27), with GBM close behind and a compact distribution indicating consistent performance. ARIMA shows the highest median RMSE; among ML models, LightGBM tends to have the highest errors and wider dispersion. The MAE panel confirms RF’s advantage, with CatBoost also showing favorable performance. At the three-step horizon, CatBoost achieves the lowest median RMSE (approximately 0.27), with XGBoost and RF close behind, while LightGBM and ARIMA exhibit higher medians. Notably, the boxplot distributions at $\tau = 3$ show greater variability across all models, consistent with the increased forecast uncertainty at medium horizons. At the six-step horizon, XGBoost achieves the lowest median RMSE (approximately 0.26), with RF close behind and relatively stable performance across rolling windows. At the nine-step horizon, GBM achieves the lowest median RMSE (approximately 0.27), followed by XGBoost and RF; ARIMA remains the weakest and dispersion increases for some models at longer horizons. Comparing across all three figures, the combined data (Figure 4) produce lower median RMSE values and more compact distributions than GT data alone (Figure 3), particularly at short horizons, while remaining broadly competitive with Official-only results.

In summary, the median rolling RMSE results in Tables 6–8 show that Official data deliver the lowest forecast errors at short and medium horizons ($\tau = 1, 3, 6$), while the combined Official and GT specification achieves its clearest RMSE gain at the long horizon ($\tau = 9$). Specifically, for Official data, the models with the lowest RMSE are CatBoost at $\tau = 1$ (0.2729) and $\tau = 3$ (0.2643), Random Forest at $\tau = 6$ (0.2603), and GBM at $\tau = 9$ (0.2722). For GT data alone, LightGBM achieves the lowest RMSE at all horizons (0.3008, 0.3021, 0.3026, and 0.3020). For the combined Official and GT dataset, CatBoost leads at $\tau = 1$ (0.2733) and $\tau = 3$ (0.2688), XGBoost at $\tau = 6$ (0.2616), and GBM at $\tau = 9$ (0.2710). Thus, the combined dataset is competitive at short horizons and yields the lowest RMSE overall at $\tau = 9$ (0.2710), while GT data alone produce materially higher RMSE values across horizons.

Regarding model performance, dominance depends on the data source used. With GT predictors alone, LightGBM is consistently the strongest performer by RMSE and the only ML model that maintains positive OoS R^2 across all horizons (alongside ARIMA). When Official information is available, whether alone or combined with GT data, CatBoost is most competitive at short horizons ($\tau = 1, 3$), while longer horizon leadership shifts to

Random Forest for Official data at $\tau = 6$, XGBoost for combined data at $\tau = 6$, and GBM for both Official and combined data at $\tau = 9$. The Diebold–Mariano results Appendix A, Tables A1–A3) align with this pattern. For GT data alone, LightGBM significantly outperforms the other ML models across horizons and is statistically indistinguishable from ARIMA. For Official data alone and combined data, the RMSE leaders at each horizon are significantly better than weaker competitors (notably LightGBM and ARIMA), while several top tree-based models are not separable at conventional levels at $\tau = 1$ and $\tau = 6$, with differences narrowing further at $\tau = 9$.

6. Conclusions

This paper compares several machine learning algorithms, including Random Forest (RF), Gradient Boosting Machine (GBM), XGBoost, LightGBM, and CatBoost, against an ARIMA baseline in forecasting Canadian GDP growth using Official economic data and Google Trends (GT) data as predictors. We employ a recursive forecasting strategy with dynamic variable selection using PDC-SIS, ensuring timely feature selection while preventing data leakage.

Our analysis reveals that Official economic data delivers the strongest forecasting performance at short and medium horizons, while combining Official and GT data achieves its clearest improvement at the longest horizon. All ML models deliver lower median RMSE than ARIMA when using Official data, and all ML models achieve positive out-of-sample R^2 values across all horizons. For Official data, CatBoost leads at short horizons, Random Forest at the medium horizon, and GBM at the long horizon. With GT data alone, LightGBM is the only ML model maintaining positive out-of-sample R^2 values across all horizons, while other ML models frequently underperform ARIMA. For combined data, CatBoost leads at short horizons, XGBoost at medium, and GBM at long horizons, and Diebold–Mariano tests confirm that the horizon-specific best models significantly outperform ARIMA.

Model performance proves to be data-source dependent. Diebold–Mariano tests show that with GT data alone, LightGBM demonstrates complete statistical dominance over other ML competitors while maintaining parity with ARIMA. With Official and combined data, the best performing models at each horizon significantly outperform ARIMA at conventional significance levels, while separation from other ML competitors varies by horizon with significance levels ranging from 10% to 1%. Across Official and combined specifications, ARIMA is consistently outperformed; CatBoost leads at short horizons ($\tau = 1, 3$), Random Forest/XGBoost lead at $\tau = 6$ depending on the data, and GBM leads at $\tau = 9$, with several top ensembles delivering similarly strong accuracy but clear improvements over ARIMA. The performance hierarchy varies substantially by data source. LightGBM excels with GT data alone, showing no significant difference versus ARIMA with these predictors, but becomes the worst performing ML model when Official data is available. This reflects fundamental signal quality differences: Official indicators provide stronger relationships with GDP growth, allowing various tree-based algorithms to outperform ARIMA, while noisier GT signals prove challenging for most ML models except LightGBM.

These results highlight GT data's limited standalone predictive value, with most ML models failing to beat ARIMA and only LightGBM maintaining positive out-of-sample R^2 values. For context on sample size and evaluation length, the analysis uses monthly data from January 2004 to September 2021 (214 months), with out-of-sample evaluation spanning August 2012 to September 2021. The recursive rolling design yields 109 rolling-window forecast evaluations, providing a substantial out-of-sample record for comparing algorithms under a consistent real-time forecasting protocol. A final limitation concerns

data vintages. The forecasting experiments use the latest-available (final-vintage) series for both GDP and the Official indicators, so the results should be interpreted as final-vintage predictive experiments rather than operational real-time nowcasting under preliminary releases and subsequent revisions. Extending the same rolling-window and dynamic PDC-SIS pipeline to real-time vintage snapshots (data as available at each forecast origin) remains an important direction for future work to quantify the effects of revisions on forecast accuracy.

This contrasts sharply with Official data, where all ML models robustly outperform ARIMA. When combined with Official indicators, GT variables provide modest incremental benefits at the longest horizon. The finding that model selection should depend on data source, with LightGBM excelling for GT data but CatBoost, Random Forest, and GBM superior with Official data, underscores the importance of matching algorithms to data characteristics in macroeconomic forecasting. Future work could explore advanced feature engineering techniques like time-based features, nonlinear splines, and Fourier transformations, as well as time-series-oriented Random Forest variants [see, e.g., Goulet Coulombe et al. [38], Goehry et al. [39]]. While tree based algorithms effectively capture seasonal and nonlinear patterns, they may face limitations with long term trends or structural shifts. Future research could explore hybrid models combining different boosting algorithms to leverage complementary strengths across economic cycles and forecasting horizons.

Author Contributions: Conceptualization, S.Q.; methodology, S.Q., B.M.C. and A.u.R.I.; software, S.Q.; validation, S.Q., F.S.D. and B.M.C.; formal analysis, S.Q.; investigation, S.Q.; resources, N.K. and A.u.R.I.; data curation, S.Q.; writing—original draft preparation, S.Q.; writing—review and editing, S.Q., B.M.C., F.S.D., N.K. and A.u.R.I.; visualization, S.Q.; supervision, B.M.C. and F.S.D.; project administration, S.Q.; funding acquisition, N.K. and A.u.R.I.; literature review, B.M.C., F.S.D. and N.K. All authors have read and agreed to the published version of the manuscript.

Funding: The authors would like to acknowledge the support of Prince Sultan University for paying the Article Processing Charge (APC) of this publication.

Data Availability Statement: Replication code is available online at <https://github.com/Shafi2016/Dynamic-Feature-Selection-for-Canadian-GDP-Forecasting> (accessed on 28 November 2025).

Acknowledgments: The authors would like to express their sincere appreciation to Prince Sultan University for the ongoing support it provides to promote academic research. The authors would also like to thank the referees and the editors for their valuable service to the academic community.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A. Appendix-3

Appendix A.1. Appendix-3: Diebold–Mariano Test Results

Table A1. Full Diebold–Mariano Pairwise Results (Official Data): All pairwise forecast accuracy comparisons by horizon.

h	Model 1	Model 2	DM Stat.	p-Value	Sig.
1	CatBoost	GBM	1.32	0.191	
	CatBoost	XGBoost	−0.26	0.795	
	CatBoost	Random Forest	1.28	0.204	
	CatBoost	LightGBM	−2.87	0.005	***
	CatBoost	ARIMA	−4.26	0.000	***

Table A1. *Cont.*

h	Model 1	Model 2	DM Stat.	p-Value	Sig.
3	CatBoost	GBM	−2.72	0.008	***
	CatBoost	XGBoost	−3.63	0.000	***
	CatBoost	Random Forest	−1.92	0.058	*
	CatBoost	LightGBM	−4.51	0.000	***
	CatBoost	ARIMA	−5.28	0.000	***
6	Random Forest	GBM	−0.53	0.600	
	Random Forest	XGBoost	−1.44	0.154	
	Random Forest	CatBoost	−1.39	0.169	
	Random Forest	LightGBM	−3.33	0.001	***
	Random Forest	ARIMA	−4.99	0.000	***
9	GBM	XGBoost	−0.37	0.715	
	GBM	CatBoost	0.84	0.401	
	GBM	Random Forest	−0.77	0.440	
	GBM	LightGBM	−1.87	0.064	*
	GBM	ARIMA	−4.19	0.000	***

Notes: Significance levels: * $p < 0.10$, and *** $p < 0.01$. A negative DM statistic indicates Model 2 has lower forecast error. For each horizon, only comparisons involving the best-performing model (by RMSE) are shown to reduce redundancy.

Table A2. Full Diebold–Mariano Pairwise Results (GT Data): All pairwise forecast accuracy comparisons by horizon.

h	Model 1	Model 2	DM Stat.	p-Value	Sig.
1	LightGBM	GBM	−5.23	0.000	***
	LightGBM	XGBoost	−4.64	0.000	***
	LightGBM	CatBoost	−3.99	0.000	***
	LightGBM	Random Forest	−2.28	0.025	**
	LightGBM	ARIMA	0.17	0.867	
3	LightGBM	GBM	−4.26	0.000	***
	LightGBM	XGBoost	−2.98	0.004	***
	LightGBM	CatBoost	−5.84	0.000	***
	LightGBM	Random Forest	−3.20	0.002	***
	LightGBM	ARIMA	1.27	0.207	
6	LightGBM	GBM	−5.54	0.000	***
	LightGBM	XGBoost	−4.46	0.000	***
	LightGBM	CatBoost	−4.96	0.000	***
	LightGBM	Random Forest	−5.02	0.000	***
	LightGBM	ARIMA	1.21	0.228	
9	LightGBM	GBM	−4.83	0.000	***
	LightGBM	XGBoost	−5.20	0.000	***
	LightGBM	CatBoost	−6.38	0.000	***
	LightGBM	Random Forest	−5.63	0.000	***
	LightGBM	ARIMA	−0.98	0.330	

Notes: Significance levels: ** $p < 0.05$, and *** $p < 0.01$. A negative DM statistic indicates Model 2 has lower forecast error. LightGBM demonstrates statistical dominance across all horizons when using Google Trends data.

Table A3. Full Diebold–Mariano Pairwise Results (Official + GT Combined Data): All pairwise forecast accuracy comparisons by horizon.

h	Model 1	Model 2	DM Stat.	p-Value	Sig.
1	CatBoost	GBM	0.78	0.434	
	CatBoost	XGBoost	−1.42	0.157	
	CatBoost	Random Forest	1.50	0.136	
	CatBoost	LightGBM	−3.71	0.000	***
	CatBoost	ARIMA	−4.08	0.000	***

Table A3. Cont.

h	Model 1	Model 2	DM Stat.	p-Value	Sig.
3	CatBoost	GBM	−2.52	0.013	**
	CatBoost	XGBoost	−1.58	0.117	
	CatBoost	Random Forest	−0.71	0.478	
	CatBoost	LightGBM	−4.55	0.000	***
	CatBoost	ARIMA	−4.33	0.000	***
6	XGBoost	GBM	−1.69	0.095	*
	XGBoost	CatBoost	1.41	0.162	
	XGBoost	Random Forest	1.40	0.163	
	XGBoost	LightGBM	−3.72	0.000	***
	XGBoost	ARIMA	−4.24	0.000	***
9	GBM	XGBoost	−1.12	0.266	
	GBM	CatBoost	0.55	0.584	
	GBM	Random Forest	−1.21	0.227	
	GBM	LightGBM	−0.77	0.445	
	GBM	ARIMA	−3.73	0.000	

Notes: Significance levels: * $p < 0.10$, ** $p < 0.05$, and *** $p < 0.01$. A negative DM statistic indicates Model 2 has lower forecast error. For each horizon, only comparisons involving the best-performing model (by RMSE) are shown to reduce redundancy.

Appendix A.2. PDC-SIS

PDC, introduced by Székely and Rizzo [37], is a means of capturing non-linear dependence between two random vectors while controlling for the effects of another random vector that can have an impact on both of these. Partial distance correlation of two vectors y and x given a set $\mathcal{C}$ of conditioning vectors S (explained below) is defined as:

$$pdcor(y, x; S) = \frac{dcor^2(y, x) - dcor^2(y, S)dcor^2(x, S)}{\sqrt{1 - dcor^4(y, S)}\sqrt{1 - dcor^4(x, S)}}$$

Consider a response time series y_t and a predictor time series with two features $(x_{1,t-1}, x_{2,t-1}) \equiv \mathbf{x}_{t-1}$. Let $\mathbf{z}_{t-1} = (x_{1,t-1}, x_{2,t-1}, \dots, x_{1,t-n}, x_{2,t-n})$ be a vector of possible covariates with dimension $2n$ where n is the maximum number of lags.

The conditioning vector $S_{i,\ell}$ for the predictor series with feature i , ($i = 1, 2$) and with lag ℓ is given by $S_{i,\ell} = (y_{t-1}, \dots, y_{t-n}, x_{i,t-1}, \dots, x_{i,t-\ell+1})$. (Note that when the lag length is $\ell = 1$, then the conditioning vector $S_{i,1} = (y_{t-1}, \dots, y_{t-n})$, since $x_{i,t-\ell+1} = x_{i,t}$, which is not a predictor.) The set of conditioning vectors $\mathcal{C}$ is now defined by $\mathcal{C} = \{S_{i,\ell}\}$, $i = 1, 2$, and $\ell = 1, \dots, n$. Thus, for example, the element $S_{1,\ell}$ of the set corresponds to the $(2(\ell - 1) + 1)^{st}$ element of $\mathcal{C}$, that is, $\mathcal{C}_{2(\ell-1)+1}$. Hence, the set of chosen predictors $\widehat{\mathcal{M}}_{\mathcal{O}}$ will include all those that satisfy the following condition:

$$\widehat{\mathcal{M}}_{\mathcal{O}} = \{j \in \{1, \dots, 2n\} \mid pdcor(y, \mathbf{z}_{t-1,j}; \mathcal{C}_j) \geq \gamma\}$$

where $\mathcal{C}_j$ is the j th element of $\mathcal{C}$ and γ is a threshold parameter that must be chosen.

Appendix A.3. Extreme Gradient Boosting (XGboost)

As mentioned above, XGBoost is a short name for the eXtreme Gradient Boosting algorithm developed by Chen and Guestrin [35] that has recently been dominating in applied machine learning and data science competitions at Kaggle. It is designed for speed and performance and is a more efficient version of gradient-boosting decision trees. It sequentially makes trees one after the other and reduces error in the subsequent iteration. Boosting is an ensemble meta-algorithm that combines the outputs of many “weak” learners to turn them into strong learners. The idea of boosting is to train weak learners sequentially

such that each subsequent tree aims to reduce the errors of the previous trees. Models are added sequentially until no further improvements are made. The predictions are combined through a weighted average of regressions to produce the final prediction. Gradient Boosting is an approach where new models are created that predict the residuals or errors of prior models and then are added together to make the final prediction. It is called Gradient Boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. Gradient descent is a method that consists of obtaining a vector of coefficients that minimize a loss function. The coefficients represent local minima. The general idea of gradient descent is to tweak parameters iteratively to minimize a loss function. Extreme Gradient Boosting builds a forest of trees in an additive manner. The algorithm iteratively produces trees that minimize the prediction error and ultimately permits the obtaining of an optimal set of predictive trees. If a new model is not satisfactory, new regression trees are added sequentially to reduce the error in predictions. The trees grow in this sequential manner, such that each tree is produced using information from previously grown trees. Each tree is fit on a modified version of the original data.

The XGBoost library implements the Gradient Boosting decision tree algorithm. Boosting employs a nonparametric additive model in which each additive component function is constructed with a decision tree (see, e.g., Chen and Guestrin [35]). The final score is a linear combination of the score of each tree as shown in Equation (A2), where f_k represents a tree.

$$\hat{y}_i = \phi(x_i) = \sum_{k=1}^K f_k(x_i), f_k \in \mathcal{F} \quad (\text{A1})$$

$$\mathcal{F} = \{f(x) = w_q(x)\}, q: \mathbb{R}^m \rightarrow T, w \in \mathbb{R}^T$$

A tree is described by two parameters, q and w : the parameter q determines the structure of the tree, i.e., it maps the sample instances to leaves, and the parameter w determines the weight attached to the leaves. $\mathcal{F}$ denotes the space of regression trees, K is the number of trees, T is number of leaves in a tree, n is the sample size, and m represents the features. Let $l(y_i, \hat{y}_i)$ be a continuous twice-differentiable convex function. The learning objective consists of a loss function $\mathcal{L}(\phi)$ and a regularization term as in (A3).

$$\mathcal{L}(\phi) = \sum_{i=1}^n l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (\text{A2})$$

$$\Omega(f_k) = \gamma T + \frac{1}{2} \lambda \|w\|^2 \quad (\text{A3})$$

The loss function reduces bias and measures how well the model fits the training data. The regularization term, $\Omega(f_k)$, controls the complexity of the model and prevents overfitting by penalizing the complex tree. The parameter λ controls the degree of the regularization of each tree and $\|w\|^2$ is the L^2 -norm of leaf scores. As Chen and Guestrin [35] indicate, the loss function $\mathcal{L}(\phi)$ in (A3) is a tree-ensemble model that has functions as parameters. Hence, it cannot be optimized with usual methods in Euclidean space. Therefore, we need to train the model additively in a sequential manner (*Boosting*) to minimize the loss function in (A3). To do so, we add the f_t (the tree) that most improves the model in Equation (A3) to the forecast $\hat{y}_i^{(t-1)}$, such that the new, t -th iteration forecast, $\hat{y}_i^{(t)} = \hat{y}_i^{(t-1)} + f_t$. That is, the prediction $\hat{y}_i^t$ is obtained iteratively as follows:

$$\begin{aligned} \hat{y}_i^{(0)} &= 0 \\ \hat{y}_i^{(1)} &= f_1(x_i) = \hat{y}_i^{(0)} + f_1(x_i) \\ \hat{y}_i^{(2)} &= f_1(x_i) + f_2(x_i) = \hat{y}_i^{(1)} + f_2(x_i) \\ \hat{y}_i^{(t)} &= \sum_{k=1}^t f_k(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i) \end{aligned}$$

Denoting the loss function at t -th iteration $\mathcal{L}^{(t)}$,

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t)}) + \Omega(f_t) \quad (\text{A4})$$

$$\mathcal{L}^{(t)} = \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \Omega(f_t) \quad (\text{A5})$$

where $\hat{y}_i^{(t-1)}$ is the prediction from the previous iteration. Then, by applying a second-order Taylor expansion to $l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i))$ around $\hat{y}_i^{(t-1)}$, the Taylor expansion of $\theta(z)$ around a yields:

$$\theta(z) = \theta(a) + \frac{\theta'(a)}{1!}(z-a) + \frac{\theta''(a)}{2!}(z-a)^2 + \frac{\theta'''(a)}{3!}(z-a)^3 + \dots \quad (\text{A6})$$

Using a second-order expansion by ignoring higher order terms, letting $\theta(z) \equiv l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i))$, $z = \hat{y}_i^{(t-1)} + f_t(x_i)$, and $a = \hat{y}_i^{(t-1)}$, we obtain the result in Equation (A8).

$$\mathcal{L}^{(t)} \approx \sum_{i=1}^n l(y_i, \hat{y}_i^{(t-1)}) + \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}} f_t(x_i) + \frac{1}{2} \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)^2}} f_t(x_i)^2 + \Omega(f_t) \quad (\text{A7})$$

where $l(y_i, \hat{y}_i^{(t-1)})$ is constant. Letting $g_i \equiv \frac{\partial l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)}}$ and $h_i \equiv \frac{\partial^2 l(y_i, \hat{y}_i^{(t-1)})}{\partial \hat{y}_i^{(t-1)^2}}$, we have

$$\mathcal{L}^{(t)} = \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2] + \Omega(f_t) + \text{constant} \quad (\text{A8})$$

or, after omitting the constant, the objective function becomes:

$$\mathcal{L}^{(t)} = \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2] + \Omega(f_t) \quad (\text{A9})$$

We substitute (A7) into (A12), letting $\|w\|^2 = \sum_{j=1}^T w_j^2$ (since w is a T -dimensional vector) to obtain

$$\tilde{\mathcal{L}}^{(t)} = \sum_{i=1}^n [g_i f_t(x_i) + \frac{1}{2} h_i f_t(x_i)^2] + \gamma T + \frac{1}{2} \lambda \sum_{j=1}^T w_j^2 \quad (\text{A10})$$

Our goal is to minimize $\tilde{\mathcal{L}}^{(t)}$ with respect to f_t . Recall also that each tree has an independent tree structure q and leaf weights w . Let I_j be the set of instances that a data point i belongs to leaf node j . I_j is defined as $I_j = \{i \mid q(x_i) = j\}$, where $q(x_i)$ depicts the tree structure from the root to node j in the decision tree. Then, we can rewrite

$$\tilde{\mathcal{L}}^{(t)} = \sum_{j=1}^T \left[\left(\sum_{i \in I_j} g_i \right) w_j + \frac{1}{2} \left(\sum_{i \in I_j} h_i + \lambda \right) w_j^2 \right] + \gamma T \quad (\text{A11})$$

We want to find the optimized weight w_j^* for node j for a given structure q , by minimizing (A12) with respect to w_j . We find w_j^* by setting

$$\frac{\partial \tilde{\mathcal{L}}^{(t)}}{\partial w_j} = \left(\sum_{i \in I_j} g_i \right) + \left(\sum_{i \in I_j} h_i + \lambda \right) w_j = 0 \quad (\text{A12})$$

$$w_j^* = -\frac{\sum_{i \in I_j} g_i}{\sum_{i \in I_j} h_i + \lambda} \quad (\text{A13})$$

By substituting (A14) into (A12), we obtain:

$$\mathcal{L}^{(t)}(q) = -\frac{1}{2} \sum_{j=1}^T \frac{\left(\sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \lambda} + \gamma T, \quad (\text{A14})$$

the first term indicating that the loss has been reduced by the new forecast obtained after adding $f_t(x_i)$, and the second term indicating the cost γT of increasing the complexity of the tree. Thus, Equation (A14) provides a measure of how good the tree structure $q(x_i)$ is by calculating the *change* in the loss function $\mathcal{L}^{(t)}$ given by $\mathcal{L}^{(t)}(q)$. It denotes the score of the tree. If it were possible to know all possible tree structures, we could choose the tree with the highest score. However, since in practice, we cannot know all possible structures q , we can proceed sequentially, and calculate the *gain* (i.e., the *negative* of the loss) of a split as we add levels to the tree. Suppose that we split the leaf at node j into two leaves. The net gain of adding an additional leaf at node j is given by:

$$\text{gain} = \frac{1}{2} \left[\frac{\left(\sum_{i \in I_{jL}} g_i \right)^2}{\sum_{i \in I_{jL}} h_i + \lambda} + \frac{\left(\sum_{i \in I_{jR}} g_i \right)^2}{\sum_{i \in I_{jR}} h_i + \lambda} - \frac{\left(\sum_{i \in I_j} g_i \right)^2}{\sum_{i \in I_j} h_i + \lambda} \right] - \gamma \quad (\text{A15})$$

where I_{jL} and I_{jR} are the set of instances that a data point i belongs to the left and right child, respectively, after splitting the leaf at node j , and I_j is, as before, the set of instances that a data point i belongs to the leaf at node j . The first two terms in equation (A15) represent the scores of the left and right child while the last term is the score *before* the split. Here, γ represents the cost (or penalty) of increasing complexity by introducing one additional leaf.

Appendix A.4. LightGBM

The LightGBM algorithm introduced by Microsoft is faster than XGBoost. The LightGBM is different from XGBoost in that the former algorithm grows trees vertically (and leaf-wise) whereas the latter algorithm grows trees level-wise. 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. XGBoost utilizes a pre-sorted algorithm and a histogram-based algorithm to find the best split point. XGBoost typically uses the nonlinear booster *gbtree* or the linear booster *gblinear* while LightGBM uses the same booster type as Gradient Boosting Decision, Random Forest, Dropouts meet Multiple Additive Regression Trees (DART), or GOSS.

Appendix A.5. CatBoost

CatBoost, developed by Prokhorenkova et al. [40], supports numerical and categorical features. In CatBoost, the term Cat refers to categorical and Boost to boosting. Though CatBoost is not limited to categorical features, it also supports numerical and text features. There are three main technical innovations of CatBoost if we compare it with XGBoost and LightGBM, as discussed in the previous section. First, Ordered Target Statistics (TS) encoding is applied for all categorical features. Second, Ordered Boosting is a tree trained on one subset of the dataset and used to estimate residuals for another unseen subset. It

divides a given dataset into random permutations. Suppose if we have a datapoint from $x_1, x_2 \dots, x_{10}$ and the model is trained on x_1 to x_5 ; we estimate residuals of x_6, x_7, x_8, x_9 and x_{10} using the model that has been trained on x_1, x_2, x_3, x_4 and x_5 data points. This process refers to ordered boosting. Lastly, it makes use of oblivious decision trees. It grows a balanced tree where the same splitting criteria is applied over an entire tree level. The features with the lowest loss are selected and used for all nodes at the tree level. XGBoost can only work on numerical values like random forest. One must apply different encoding techniques such as label encoding, mean encoding, or one-hot encoding before supplying categorical data to XGBoost and random forest. However, the built-in mechanism that combines one-hot encoding and advanced mean encoding enables CatBoost to deal with categorical variables automatically.

Appendix A.6. Gradient Boosting Machine (GBM)

The Gradient Boosting Machine (GBM), introduced by Friedman [34], is a powerful ensemble learning algorithm that builds predictive models by combining weak learners (typically decision trees) sequentially. The GBM follows an additive training process where each successive tree aims to minimize the residual errors of previous trees through gradient descent optimization. Unlike Random Forests that build trees in parallel, GBM builds trees sequentially where each new tree corrects the errors of the combined ensemble of existing trees.

The algorithm's key technical components include:

1. **Loss Function Optimization:** GBM can optimize various differentiable loss functions (e.g., squared error for regression, log loss for classification). At each iteration, it computes negative gradients of the loss function that serve as the target for the next tree.
2. **Weak Learners:** GBM typically uses decision trees of limited depth (usually 2–6 levels) as base learners. These “weak” learners ensure the model maintains bias-variance balance and avoids overfitting.
3. **Shrinkage:** A learning rate parameter scales the contribution of each tree, controlling the trade-off between model complexity and training time. Smaller learning rates generally yield better generalization but require more trees.
4. **Subsampling:** Similar to Random Forests, GBM can use row and column sampling techniques to introduce randomness and reduce overfitting. This includes selecting random subsets of training data (Stochastic Gradient Boosting) or features for each tree.

The iterative nature of GBM makes it particularly effective at capturing complex nonlinear relationships in data. However, this sequential processing also means it cannot leverage parallel computing as effectively as algorithms like Random Forests. While newer implementations like XGBoost and LightGBM have introduced various optimizations, the original GBM remains foundational to understanding Gradient Boosting principles and serves as a baseline for comparing more advanced boosting algorithms.

Appendix A.7. Hyper-Tuning Parameters

We used Python for model implementation, utilizing the native libraries for GBM, CatBoost, XGBoost, LightGBM, and Random Forest (RF), along with the PDC-SIS Python routine to select predictors. Sliding-window forecasts were calculated for each model, with dynamic feature selection ensuring that relevant predictors were chosen in each rolling window.

Hyper-tuning parameters is one of the most complicated and vital tasks in ML to improve forecast performance. RandomizedSearchCV was used to explore the parameter space efficiently, ensuring a balance between model complexity and forecast accuracy.

Table A4. Model tuning spaces (applied identically across Official, GT, and Official + GT predictor sets). Hyperparameters explored via RandomizedSearchCV within each rolling window.

Model	Hyperparameter	Candidate Values
Random Forest (RF)	n_estimators	20, 50, 100, 200, 300, 400, 500, 1000
	max_depth	3, 5, 10, 20, None
	min_samples_split	2, 5, 10
	min_samples_leaf	1, 2, 4
	bootstrap	True, False
Gradient Boosting (GBM; sklearn)	n_estimators	100, 200, 500
	learning_rate	0.01, 0.03, 0.05, 0.1, 0.2
	max_depth	3, 4, 5, 6, 7
	min_samples_split	2, 5, 10
	min_samples_leaf	1, 2, 4
	subsample	0.6, 0.8, 1.0
	max_features	sqrt, log2, None
XGBoost	n_estimators	100, 200, 300, 400, 500, 600, 1000
	max_depth	3, 5, 7, 8, 10, 15
	learning_rate	0.01, 0.03, 0.05, 0.1, 0.2
	subsample	0.5, 0.6, 0.7, 0.8, 1.0
	colsample_bytree	0.5, 0.6, 0.7, 0.8, 1.0
LightGBM	n_estimators	100, 200, 300, 500, 600, 1000
	max_depth	1, 3, 5, 7, 8, 10
	learning_rate	0.01, 0.03, 0.05, 0.1, 0.2
	num_leaves	20, 31, 40
	feature_fraction	0.5, 0.6, 0.7, 0.8, 1.0
CatBoost	iterations	100, 200, 300, 400, 500, 600, 1000
	depth	2, 4, 6, 8, 10
	learning_rate	0.01, 0.03, 0.05, 0.1, 0.2
	l2_leaf_reg	1, 3, 5

Notes: The same tuning space is used across the three datasets; only the predictor set differs.

Each model has unique tuning parameters:

- **Random Forest (RF):** We optimized key parameters such as `n_estimators` (number of trees), `max_depth` (maximum depth of each tree), `min_samples_split` (minimum samples required to split a node), `min_samples_leaf` (minimum samples required in a leaf), and `bootstrap` (whether to sample with replacement). These parameters control the model's ability to capture complexity and prevent overfitting.
- **XGBoost:** For XGBoost, we focused on tuning parameters such as `n_estimators`, `max_depth`, `learning_rate`, `subsample`, and `colsample_bytree`. The `max_depth` and `learning_rate` parameters balance model complexity and convergence rate, while `subsample` and `colsample_bytree` help in regularizing the model to prevent overfitting.
- **GBM:** For Gradient Boosting Machine (GBM), we tuned `n_estimators`, `learning_rate`, `max_depth`, `min_samples_split`, `min_samples_leaf`, `subsample`, and `max_features`. These parameters collectively control the Gradient Boosting process and help manage the model's complexity, with parameters like `subsample` and `max_features` acting as regularization techniques.
- **LightGBM:** In LightGBM, we optimized `n_estimators`, `learning_rate`, `max_depth`, `num_leaves`, and `feature_fraction`. `num_leaves` and `max_depth` control the growth of each tree, while `feature_fraction` limits the number of features used in each iteration, helping to reduce overfitting.
- **CatBoost:** For CatBoost, we tuned `iterations`, `depth`, `learning_rate`, `l2_leaf_reg`, and `bagging_temperature`. CatBoost's `l2_leaf_reg` parameter helps regularize leaf

values, while `bagging_temperature` controls the random sampling process in each iteration to enhance model stability.

Our hyper-tuning process was structured to balance flexibility and stability, adapting to the dynamic feature selection process and ensuring robust performance across forecast horizons for each model.

References

- Choi, H.; Varian, H. Predicting Initial Claims for Unemployment Benefits. *Google Inc.* **2009**, *1*, 1–5.
- Choi, H.; Varian, H. Predicting the present with Google Trends. *Econ. Rec.* **2012**, *88*, 2–9. [[CrossRef](#)]
- Tkacz, G. Predicting Recessions in Real-Time: Mining Google Trends and Electronic Payments Data for Clues. *C.D. Howe Inst. Comment.* **2013**, 387. Available online: https://www.cdhowe.org/wp-content/uploads/2024/12/Commentary_387_0-1.pdf (accessed on 28 November 2025).
- Götz, T.B.; Knetsch, T.A. Google data in Bridge equation models for German GDP. *Int. J. Forecast.* **2019**, *35*, 45–66. [[CrossRef](#)]
- Ferrara, L.; Simoni, A. When Are Google Data Useful to Nowcast GDP? An Approach via Preselection and Shrinkage. *J. Bus. Econ. Stat.* **2023**, *41*, 1188–1202. [[CrossRef](#)]
- Fan, J.; Lv, J. Sure independence screening for ultrahigh dimensional feature space. *J. R. Stat. Soc. Ser. B (Stat. Methodol.)* **2008**, *70*, 849–911. [[CrossRef](#)] [[PubMed](#)]
- Fortin-Gagnon, O.; Leroux, M.; Stevanovic, D.; Surprenant, S. A large canadian database for macroeconomic analysis. *Can. J. Econ. Can. D'économique* **2022**, *55*, 1799–1833. [[CrossRef](#)]
- Nilashi, M.; Minaei-Bidgoli, B.; Alghamdi, A.; Alrizq, M.; Alghamdi, O.; Nayer, F.K.; Aljehane, N.O.; Khosravi, A.; Mohd, S. Knowledge discovery for course choice decision in massive open online courses using machine learning approaches. *Expert Syst. Appl.* **2022**, *199*, 117092. [[CrossRef](#)]
- Ullah, A.; Anwar, S.M.; Li, J.; Nadeem, L.; Mahmood, T.; Rehman, A.; Saba, T. Smart cities: The role of internet of things and machine learning in realizing a data-centric smart environment. *Complex Intell. Syst.* **2024**, *10*, 1607–1637. [[CrossRef](#)]
- Alfaiz, N.S.; Fati, S.M. Enhanced credit card fraud detection model using machine learning. *Electronics* **2022**, *11*, 662. [[CrossRef](#)]
- Ashfaq, T.; Khalid, R.; Yahaya, A.S.; Aslam, S.; Azar, A.T.; Alsafari, S.; Hameed, I.A. A machine learning and blockchain based efficient fraud detection mechanism. *Sensors* **2022**, *22*, 7162. [[CrossRef](#)]
- Biau, O.; D'Elia, A. Euro Area GDP Forecasting Using Large Survey Datasets: A Random Forest Approach. 2009. Available online: <https://www.mdpi.com/authors/> (accessed on 28 November 2025).
- Tiffin, A. *Seeing in the Dark: A Machine-Learning Approach to Nowcasting in Lebanon*; International Monetary Fund: Washington, DC, USA, 2016.
- Jung, J.K.; Patnam, M.; Ter-Martirosyan, A. *An Algorithmic Crystal Ball: Forecasts-Based on Machine Learning*; International Monetary Fund: Washington, DC, USA, 2018.
- Fuleky, P. *Macroeconomic Forecasting in the Era of Big Data: Theory and Practice*; Springer: New York, NY, USA, 2020; Volume 52.
- Medeiros, M.C.; Vasconcelos, G.F.R.; Veiga, A.; Zilberman, E. Forecasting Inflation in a data-rich environment: The benefits of machine learning methods. *J. Bus. Econ. Stat.* **2021**, *39*, 98–119. [[CrossRef](#)]
- Coulombe, P.G.; Leroux, M.; Stevanovic, D.; Surprenant, S. Macroeconomic data transformations matter. *Int. J. Forecast.* **2021**, *37*, 1338–1354. [[CrossRef](#)]
- Tkacz, G. Neural network forecasting of Canadian GDP growth. *Int. J. Forecast.* **2001**, *17*, 57–69. [[CrossRef](#)]
- Chernis, T.; Sekkel, R. A dynamic factor model for nowcasting Canadian GDP growth. *Empir. Econ.* **2017**, *53*, 217–234. [[CrossRef](#)]
- Athey, S.; Bayati, M.; Imbens, G.; Qu, Z. Ensemble methods for causal effects in panel data settings. *AEA Pap. Proc.* **2019**, *109*, 65–70. [[CrossRef](#)]
- Bulligan, G.; Marcellino, M.; Venditti, F. Forecasting economic activity with targeted predictors. *Int. J. Forecast.* **2015**, *31*, 188–206. [[CrossRef](#)]
- Babii, A.; Ghysels, E.; Striaukas, J. Machine learning time series regressions with an application to nowcasting. *J. Bus. Econ. Stat.* **2022**, *40*, 1094–1106. [[CrossRef](#)]
- Beyhum, J.; Striaukas, J. *Factor-Augmented Sparse MIDAS Regressions with an Application to Nowcasting*; Faculty of Economics and Business Discussion Paper Series; DPS 24.14; KU Leuven: Leuven, Belgium, 2024; 65p.
- Soybilgen, B.; Yazgan, E. Nowcasting US GDP using tree-based ensemble models and dynamic factors. *Comput. Econ.* **2021**, *57*, 387–417. [[CrossRef](#)]
- Kant, D.; Pick, A.; Winter, J.d. Nowcasting GDP using machine learning methods. *Asta Adv. Stat. Anal.* **2025**, *109*, 1–24. [[CrossRef](#)]
- Franjic, D.; Schweikert, K. Predictor Preselection for Mixed-Frequency Dynamic Factor Models: A Simulation Study with an Empirical Application to GDP Nowcasting. *J. Forecast.* **2025**, *44*, 255–269. [[CrossRef](#)]

27. Chu, B.; Qureshi, S. Comparing out-of-sample performance of machine learning methods to forecast US GDP growth. *Comput. Econ.* **2023**, *62*, 1567–1609. [[CrossRef](#)]
28. Yousuf, K.; Feng, Y. Targeting predictors via partial distance correlation with applications to financial forecasting. *J. Bus. Econ. Stat.* **2022**, *40*, 1007–1019. [[CrossRef](#)]
29. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed.; Springer: New York, NY, USA, 2009.
30. Murphy, K.P. *Machine Learning: A Probabilistic Perspective*; The MIT Press: Cambridge, MA, USA, 2012.
31. Breiman, L. Random forests. *Mach. Learn.* **2001**, *45*, 5–32. [[CrossRef](#)]
32. Freund, Y.; Schapire, R.E. Experiments with a new boosting algorithm. In *Proceedings of the Thirteenth International Conference on Machine Learning (ICML 1996)*; Saitta, L., Ed.; Morgan Kaufmann Publishers: San Francisco, CA, USA, 1996; pp. 148–156.
33. Breiman, L. Arcing classifier. *Ann. Stat.* **1998**, *26*, 801–849. [[CrossRef](#)]
34. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [[CrossRef](#)]
35. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*; ACM: Singapore, 2016; pp. 785–794.
36. Eichenauer, V.Z.; Indergand, R.; Martínez, I.Z.; Sax, C. Obtaining consistent time series from Google Trends. *Econ. Inq.* **2022**, *60*, 694–705. [[CrossRef](#)]
37. Székely, G.J.; Rizzo, M.L. Partial distance correlation with methods for dissimilarities. *Ann. Stat.* **2014**, *42*, 2382–2412. [[CrossRef](#)]
38. Goulet Coulombe, P.; Leroux, M.; Stevanovic, D.; Surprenant, S. How is machine learning useful for macroeconomic forecasting? *J. Appl. Econom.* **2022**, *37*, 920–964. [[CrossRef](#)]
39. Goehry, B.; Yan, H.; Goude, Y.; Massart, P.; Poggi, J.M. Random forests for time series. *Revstat-Stat. J.* **2023**, *21*, 283–302.
40. Prokhorenkova, L.; Gusev, G.; Vorobev, A.; Dorogush, A.V.; Gulin, A. CatBoost: Unbiased boosting with categorical features. *arXiv* **2017**, arXiv:1706.09516.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.