Macroeconomic Predictions using Payments Data and Machine Learning[*]

James T.E. Chapman and Ajit Desai

Bank of Canada, Ottawa, ON, Canada

June 25, 2021

Abstract

Predicting the economy's short-term dynamics—a vital input to economic agents' decision-making process—is often done using lagged indicators in the linear models. This is mostly sufficient during normal times, but could be inadequate during the crisis periods such as COVID-19. In this paper, we demonstrate: (a) the payments systems data which captures a variety of economic transactions can provide information to estimate the state of the economy in real time, and (b) the machine learning (ML) can provide a set of econometric tools to effectively handle wide variety in the payments data and to capture sudden and large effects of the crisis. The use of ML models, however, leads to the loss of *interpretability* and the problem of *overfitting*, which diminishes the effectiveness of such models. We mitigate these challenges by: (a) using the Shapley value-based approach to interpret ML model predictions in terms of marginal contribution of each predictor, and (b) devising a novel cross-validation strategy tailored for the macroeconomic prediction models to alleviate the overfitting.

Keywords: Nowcasting, Payments data, Machine learning, Interpretability, Overfitting
JEL Codes: C53, C55, E37, E42, E52

[*] The opinions here are of the authors and do not necessarily reflect those of the Bank of Canada. We would like to thank Narayan Bulusu and seminar participants at the Bank of Canada for their comments. We also like to thank discussant and seminar participants in the 2021 International Association for Applied Econometrics (IAAE) Annual Conference, 7th RCEA Time Series Workshop, and 2021 Big Data and Machine Learning in Finance Conference, for their comments. We also thank Poclaire Kenmogne for his excellent assistance.

1 Introduction

Knowledge of the economy's short-term dynamics is a vital input into every economic agents' decision-making process. However, gauging the current state of the economy—known as nowcasting—is difficult for various reasons. For instance, many different data series are needed to describe the state of the economy adequately, but most of the official sources of these data are released with significant lags (Giannone et al. 2008; Banbura et al. 2010; Angelini et al. 2011). This problem is especially difficult during times of crisis, such as the COVID-19 pandemic, primarily because of the unprecedented economic impacts of the crisis and the unconventional policy responses to mitigate those crises (Spange 2010; Hamilton 2011; Greenwood et al. 2020). During such times, traditional models have difficulty because of the realizations of the target variables are far away from their average values (Vrontos et al. 2020; Coulombe et al. 2021; Chapman and Desai 2021).

To address such challenges, econometricians have either used new data or developed new techniques (Giannone et al. 2008; Choi and Varian 2012; Buono et al. 2017; Bok et al. 2018; Kapetanios and Papailias 2018; Koop and Onorante 2019). In this paper, we combine both the new data and machine learning (ML) approaches to create a nowcast of the Canadian economy. First, we use comprehensive and timely settlement data from Canada's retail and large-value electronic payments systems. We then use ML models[1] to effectively handle wide variety in the payments systems data and to capture sudden, large, and possibly nonlinear effects of the crisis.

The use of ML models, however, leads to many challenges that could reduce the effectiveness of these models for nowcasting. For instance, it is easy to overfit the model on the given set of data, which could reduce the out-of-sample performance of those models. Also, more importantly, it is hard to interpret these models, and the interpretability could be useful in many application including macroeconomic predictions (Varian 2014; Mullainathan and Spiess 2017; Chakraborty and Joseph 2017; Athey and Imbens 2019). To the best of our knowledge, these challenges have not yet convincingly addressed in the context of macroeconomic nowcasting.

In this paper, we attempt to address the interpretability issue by using the SHapley Additive exPlanations (SHAP) methodology (Lundberg and Lee 2017; Lundberg et al. 2020), based on Shapley values from coalition game theory (Shapley 1953; Osborne and Rubinstein 1994). To utilize this approach, we need to consider each nowcasting exercise as a "game" then the Shapley values can be used to fairly distribute the *payout* (i.e., the model prediction) among the *players* (i.e., the predictors) of the game. This is a model-independent approach; therefore, it could be used with any type of nowcasting model. The SHAP provides a way to explain nowcasting model predictions at each nowcasting horizon in terms of the marginal contribution of each predictor towards the final prediction. Furthermore, by averaging each prediction instance's contribution—in terms of Shapley values—we can compute the marginal contribution of each predictor for the entire sample.

[1] We use the following parametric and non-parametric ML models which are popular among time series forecasters (Ahmed et al. 2010; Bok et al. 2018; Athey and Imbens 2019; Coulombe et al. 2021): elastic net, support vector machines, random forest, gradient boosting, feedforward artificial neural network (Hastie et al. 2009).

Next, to alleviate the ML models problem of overfitting and improve the out-of-sample performance, we devise an improved cross-validation (CV) strategy tailored to the macroeconomic nowcasting models. In the cases where the out-of-sample test set has an economic crisis period, but the validation set[2] does not, the traditional CV approaches, such as k-fold or leave-one(or p)-out validation (Hastie et al. 2009), could be challenging because: (a) the usual k-fold splitting breaks the order of data and lead to the use of future data points for the past predictions, (b) the distribution of test and validation sets could be different, and (c) the model tuned on predominantly normal periods might not perform well on out-of-sample crisis period. To overcome this, similar to Kuhn et al. 2013, we use a randomized expanding window approach with k-fold CV, but without changing the order of the data (Figure 3). Since we have the COVID-19 period in the test set, using random sampling helps to include a few samples from the financial crisis period in the validation set. Consequently, making the distributions of validation and test sets similar and hence assist in selecting the model that can perform well on both the normal and crisis periods.

We observe that the retail and large-value payments system data in the ML models—especially a nonlinear gradient boosting regression—can lower nowcast errors significantly. We get a 35-40% reduction in root-mean-square errors (RMSE) in nowcasting GDP, retail-trade sales (RTS), and wholesale trade sale (WTS)[3] over a linear benchmark model[4]. Furthermore, in the presence of payments data, the ML models in comparison against the dynamic factor models—commonly preferred for macroeconomic nowcasting—can reduce nowcasting RMSE by up to 20-25%. The out-of-sample performance gain using payments data and ML models is relatively more during the COVID-19 crisis periods than the pre-COVID normal economic growth period.

The Shapely value-based model interpretation reveals that some of the payments streams are equally important as prominent benchmark predictors in nowcasting GDP, RTS, and WTS; Moreover, during the COVID-19 period, many payments streams contribute strongly toward model prediction compared to the benchmark predictors and provide crucial information about the crisis in real-time. Our analysis suggests that the contribution of payments data—in terms of the Shapley values—is small and linear during the periods of normal growth; however, during the periods of strong negative and positive growths, the payments data contribution is asymmetrical and nonlinear. We also observe an improved model performance when the proposed randomized expanding window approach with k-fold-CV is used for ML model parameter tuning and cross-validation.

[2] The part of the in-sample training set used for ML model parameters tuning and CV (see Figure 3).

[3] We nowcast GDP, because it is a crucial indicator for policymakers and commonly used to test the nowcasting model performance. We nowcast RTS and WTS because we use payments data; therefore, we presume it has value in predicting them. Also, having multiple targets allows us to test our models robustness. Note: all three target indicators are released with about two months of delay in Canada.

[4] As a benchmark model, we use the following series in linear regression model: consumer price index (CPI), and unemployment (UNE), the Canadian financial stress indicator (CFSI), and the Conference Board's consumer confidence index (CBCI). The unemployment incorporates the effects of public sector hiring, and the inflation is useful since we are using nominal predictors (Galbraith and Tkacz 2018). The CFSI is a composite measure of systemic financial market stress for Canada (Duprey 2020). The CBCC is based on a survey of Canadian households, and it is shown to be useful to predict household spending in Canada (Kwan and Cotsomitis 2006).

We are not the first researchers to use payments data for nowcasting. In the past—driven by the need to overcome dependence on lagged variables—econometricians have used payments data for macroeconomic predictions (Galbraith and Tkacz 2007; Carlsen and Storgaard 2010; Barnett et al. 2016; Duarte et al. 2017; Galbraith and Tkacz 2018; Aprigliano et al. 2019). Canadian payments data are a particularly good candidate for nowcasting, because it record transactions processed in various payment instruments. Thus, it captures a broad range of Canadian consumers, firms, and government economic activities. Also, these data are gathered electronically, hence available promptly, and they are free of measurement or sampling errors (Galbraith and Tkacz 2007). Such datasets are shown to be more useful during economic crisis periods such as the 2008 financial crisis and COVID-19 shock (Chetty et al. 2020; Bounie et al. 2020; Carvalho et al. 2020).

Traditionally, researchers have used data from a few selected payment instruments for nowcasting. One issue with this approach is that particular instruments may rise or fall due to economic as well as non-economic reasons.[5] Using one or two payments instruments in isolation might not help to capture the full picture of the economy (Chapman and Desai 2021).

Recently—driven by the need to exploit non-traditional and large data sets—econometricians have started using ML models for macroeconomic nowcasting (Chakraborty and Joseph 2017; Richardson et al. 2020; Maehashi and Shintani 2020; Chapman and Desai 2021). These articles suggest that the ML models often outperform traditional modeling approaches, such as the ordinary least squares and dynamic factor models, in nowcasting.

The ML models we explore in this paper can help capturing sudden and large effects of the economic crisis and impacts of unconventional policies designed to alleviate such crisis (Coulombe et al. 2021; Chapman and Desai 2021). This is important because different crises have reflected differently in the payments streams suggesting a tangled and possibly nonlinear relationship between a few payments streams and macroeconomic targets[6]. The ML models could also be useful to efficiently handle a wide variety in payments data and to effectively manage collinearity in them[7]. Moreover, the ML models are beneficial when the emphasis is on improving prediction accuracy—which is also a focus of this paper (Mullainathan and Spiess 2017; Athey 2017).

We proceed as follows. In section 2 we describe the payments systems data and discuss the adjustments performed on these data for macroeconomic predictions. Section 3 provides a brief overview of various methods employed for nowcasting along with a discussion on challenges associated with using ML models for predictions. Followed by the results and discussion in section 4. Finally, in section 5 we conclude our findings. Several appendices provide further details on the payments data and the nowcasting methodology employed in this paper.

[5] In Canada, the shares of electronic means payments are increasing, and the use of cash is declining primarily due to ease of accessibility driven by technological advancements. For instance, compared to 2018, the share of debit card payments processed through the ACSS was increased by 21%, and cash declined by 27% in 2019.

[6] In April 2020, the Canadian government started provided social benefits to its citizens directly affected by COVID-19. This is reflected by the large increase in the payments flow in the respective stream. Such policy was not implemented during the 2008 financial crisis, where we notice a drop in payments flow in the same stream (see Figure 1).

[7] Some of the payments series used here are strongly correlated to each other (Chapman and Desai 2021).

2 Payments Systems Data

The vast majority of non-cash transactions require settlement to extinguish the debt from the buyer to the seller. In modern economies, this is accomplished via some centralized payments system. The data coming from such systems are potentially useful because (a) they are timely, i.e., available immediately after the end of the period, (b) they are available at high-frequency, i.e., at the transaction or day levels, (c) they are precise, i.e., carry no sampling and measurement error, and (d) they are comprehensive, i.e., capture a broad range of financial activities across the country (Galbraith and Tkacz 2007, 2018; Aprigliano et al. 2019; Chapman and Desai 2021).

In Canada, the automated clearing settlement system (ACSS) and the large-value transfer system (LVTS) are used to settle most transactions.[8] Our data consist of all settled transactions in both ACSS and LVTS payments systems. The ACSS settles the majority of retail and small-value payment items on a net basis. In 2019, the ACSS handled an average of 33 million transactions per business day, with an average daily total value of 29 billion dollars. The ACSS processes twenty-two payments streams. Broadly, these streams can be categorized into two groups: (1) Electronic streams, which include, for example, Automated Fund Transfer, Point-of-Sale payments, and Government Direct Deposit; and (2) paper streams, which incorporate Encoded Paper, Paper Remittances, and Government Paper Items.

In ACSS, due to their usability, the electronic means of payments have become common than paper items. Most of these changes are primarily driven by technological advancements leading to the inception and adoption of new payment instruments; However, an economic crisis like the global financial crisis and the COVID-19 shock also influence the payments flow. Historically, the Encoded Paper stream has the highest value shares in ACSS, followed by the AFT Credit. The POS Payments stream has the largest volume shares, followed by the Encoded paper stream.[9]

The LVTS facilitates the transfer of large-value payments between Canadian financial institutions on a gross basis. In 2019, the LVTS handled an average of 40 thousand transactions per business day, with an average daily total value of \$189 billion dollars. LVTS provides each participant with two options called tranches, T1 and T2, to exchange payments. Each tranche (henceforth also referred to as stream) is differs based on how individual payments are collateralized. Payments in the LVTS comprise foreign exchange payments, payments for the settlement of Canadian-dollar-denominated securities, payments related to the final settlement of the ACSS and Government of Canada transactions, as well as the Bank of Canada's own and its clients' payments.[10]

In LVTS, most of the payments value and volume are processed through T2. Historically, T2 has processed roughly 75% value and 98.7% volume of payments, and T1 has processed roughly 25% value and 1.3% volume.

[8] The ACSS supports 99% percent of the daily transaction volume and 13% of the daily value processed by the Canadian payment systems. The LVTS settles 87% of the total value moving through the Canadian payment systems.

[9] Refer Chapman and Desai 2021 for the breakdown of shares of payments streams in the ACSS.

[10] Refer to Arjani and McVanel 2006 for further details on types of payments settled in the LVTS

Table 1: ACSS and LVTS payments streams used in this study.[a]

ID	Label	Short Description
C	AFT Credit[b]	Direct Deposit (DD): payroll, account transfers, etc.
D	AFT Debit	Pre-authorized debit (PAD): bills, mortgages, utility, etc.
E	Encoded Paper[c]	Paper bills of exchange: cheques, bank drafts, paper PAD, etc.
N	Shared ABM	Debit card payments to withdraw cash at shared ABM network
P	POS Payments[d]	Point of sale (POS) payments using debit card
X	Corporate Payments[e]	Exchange of Corporate-to-Corporate and bill payments
All	Allstream[f]	It is the sum of all payments streams settled in the ACSS
T1	LVTS-T1[g]	Time critical payments and payments to Bank of Canada
T2	LVTS-T2	Security settlement, foreign exchange and other obligations

[a] The first six payments streams are representative of twenty payments instruments processed separately in the ACSS. There are more payments instruments; however, they are not available for the entire period we consider in this paper; therefore, they are excluded from this study. The excluded streams are ICP Regional Image Payment and ICP Regional Image Payments Returns. Note: Excluded streams collectively account for only about 0.001% of the total value settled in the system. For further details on individual ACSS streams, refer to Appendix A.

[b] Stream C is the sum of AFT Credit and Government Direct Deposits streams (GDD). We combine them because starting in April 2012; the GDD was separated from AFT Credit.

[c] Stream E is the sum of multiple streams settled separately in ACSS. It combines Encoded Paper (E), Large-Value Encoded Paper (L), and Image Captured Payments, (O) Canada Savings Bond (B), Receiver General Warrants (G), and Treasury Bills and Bonds (H) streams. It subtracts Image Captured Return (S), Unqualified (U), and Computer Rejects (Z) streams. We combine them because, over time, all these streams were separated from Encoded Paper streams (E).

[d] Value and volume of stream P are obtained by summing Online Payments (J) and POS Payments (P) streams and subtracting Online Returns (K) and POS Refund (Q) streams.

[e] Stream X is the sum of Paper Remittances (F), EDI Payments (X), and EDI Remittances (Y). It is composed of all Corporate-to-Corporate payments and Corporate bill payments and remittances.

[f] Allstream is the sum of all the payments streams processed in the ACSS.

[g] We exclude payments from the Bank of Canada in LVTS-T1.

2.1 Adjustments to the Payments Data

Driven by technological advancements, in the past, some of the payments instruments from ACSS were discontinued or merged into others, and some new payments instruments were created.[11] For example, starting in 2012, a new stream was created to process the Government of Canada's direct deposit payments. This addition caused a sudden drop in the value and volume of payments in the AFT Credit stream, where they were originally processed. To overcome the effects of such sudden changes and to get a better representation of payments flow, we merged a few streams belonging to similar categories and settled related payments.[12] Also, to overcome the effects of consumers' choice of payments, i.e., when they switch payments method,[13] we include the sum of all payments instruments in ACSS, "Allstream" as a separate series. This should help us get the overall picture from the ACSS and mitigate the effects of a few unused streams.

After those adjustments, we are left with seven streams from ACSS[14] and two streams from LVTS that are listed in Table 1 along with a short description. For nowcasting, we use both the monthly gross dollar amount, i.e., *value* and the number of transactions, i.e., *volume* settled in those payments instruments; therefore, we have in total eighteen series.

Like other macroeconomic time series, payments data have a strong seasonal component. We adjust all series (both value and volume) for seasonality using the X-13 ARIMA tool (X13 Reference Manual 2017).[15] Note that the recursive seasonal adjustments are performed in real-time using the data available up to the nowcasting horizon at each time step. The year-over-year (YOY) growth rates of the seasonality-adjusted payments series are used to predict the similarly adjusted YOY growth rates of macroeconomic indicators.[16]

Our dataset does not include some of the payments instruments which are not settled through the ACSS or LVTS, such as credit card and e-transfer payments.[17] However, Galbraith and Tkacz (2018) concluded that the credit card payments data in Canada does not add significant value in nowcasting GDP and retail sales.[18] Furthermore, our dataset does not include *on-us* transactions where both sender and receiver have an account with the same financial institution; therefore, such transactions do not need to be settled in payment system. However, their shares are small and might not drastically influence our analysis.[19]

[11] See Appendix A for specifics on changes in multiple ACSS streams over time.

[12] See Table 1 footnotes for the specifics of each adjustment performed.

[13] For nowcasting, we are interested in capturing if spending (or earning) has slowed (or stopped), rather than a switched payment method.

[14] The seven ACSS streams comprise transactions settled in all of the ACSS payments instruments.

[15] Seasonality adjustments are performed because official macro indicators are released with similar adjustments.

[16] Using growth rates (instead of levels) helps in inducing (approximate) stationarity in both target and predictors.

[17] In 2019, credit card payments accounted for about 6.2% in value and 31.1% in volume of total retail transactions in Canada. Similarly, e-transfer payments accounted for 1.5% in value and 2.5% in volume (Paturi and Chiron 2020).

[18] Note that in Galbraith and Tkacz (2018) the authors used a short sample size in their analysis of credit card data. The results could be different for a larger sample size.

[19] On-us payments amount to roughly 20% more than those settled in ACSS. The values of on-us transactions differ by payments instrument; for instance, in Encoded Paper, it is about 25%, and in POS Payments, it is about 16%.

2.2 Payments Data for Macroeconomic Nowcasting

The crux of the nowcasting problem is that most of the official estimates of macro indicators are released with a substantial delay. For instance, in Canada, GDP is released with a delay of eight weeks, and both retail and wholesale sales are released with six weeks of lag. Furthermore, they undergo multiple revisions, sometimes years after, highlighting the uncertainty of the measurement. Moreover, during a rapid crisis such as COVID-19, macroeconomic predictions are difficult because of the large and unprecedented economic impact. This could undermine the use of lagged data for nowcasting because it does not carry much information about the impact of a crisis. Therefore, it is valuable to use more timely available information—in this case, the payments systems data.

The payments data capture numerous types of transactions from both sides of macroeconomic accounts: For example, consumers' income and expenditures, business-to-business payments, and Canada's government spending. Therefore, this variety, timeliness, and the lack of sampling and measurement errors in the payments dataset make it a rich economic information source (Galbraith and Tkacz 2007, 2018; Chapman and Desai 2021).

For nowcasting exercises, we use Canada's monthly GDP, retail and wholesale trade sales at the latest available vintages (i.e., after revisions) and real-time vintages (i.e., first release) as target variables.[20] We select these indicators because GDP is crucial for policymakers, and since we are using the payments data, we think it has value in predicting RTS, WTS. All these indicators are released with substantial lag in Canada and are available at the monthly frequency for all historical releases; therefore allows us to test the robustness of our models.

The YOY growth rates of the latest monthly GDP are plotted with Encoded Paper and AFT Credit values in Figure 1(top). Similarly, RTS's YOY growth rates are plotted with POS Payments and Shared ABM values in Figure 1(middle). The YOY growth rates of WTS are plotted with Corporate Payments and LVTS-T2 values in Figure 1(bottom). To get a sense of the importance of the payments data during a crisis, we highlight all variables' growth rates during the global financial crisis period (in gray) and the COVID-19 period (in blue).

During the global financial crisis period, the decline and rebound in these payments streams' growth rates go hand-in-hand with macroeconomic indicators. Similarly, during the COVID-19 shock, we can see a sudden drop in most of the payments stream, like macro variables. For instance, GDP and Encoded Paper, RTS and POS Payments, and WTS and Corporate Payments show similar movement during both crisis periods. This is a good indication of the economic value associated with these payments streams during such times.

During the COVID-19 period, however, we observe a tangled relationship between the macro indicators and some of the payments streams. For instance, the value of payments through the AFT Credit stream (which also includes the Government Direct Deposit payments) did not drop

[20] Latest vintages of seasonally adjusted monthly GDP, RTS, and WTS are obtained from Statistics Canada Tables 36-10-0434-01, 20-10-0008-01, 20-10-0074-01, respectively. Similarly, historical releases of GDP, RTS, and WTS are obtained from tables 36-10-0491-01, 20-10-0054-01, 20-10-0019-01.

significantly at the onset of COVID-19 shock; on the contrary, starting in April 2020, the value of payments processed through the AFT Credit stream increased due to flow of Government social payments to those directly affected by the pandemic (Figure 1-top). Similarly, we note that the value of payments through the LVTS-T2 stream surged significantly at the onset of COVID-19, showing an opposite behavior with macro indicators during the same period. Such behavior is not seen during the global financial crisis period where both WTS and T2 growth rates were dropped (Figure 1-bottom).[21] Such twisted behavior could be challenging for the linear models and would justify the use of nonlinear ML models.

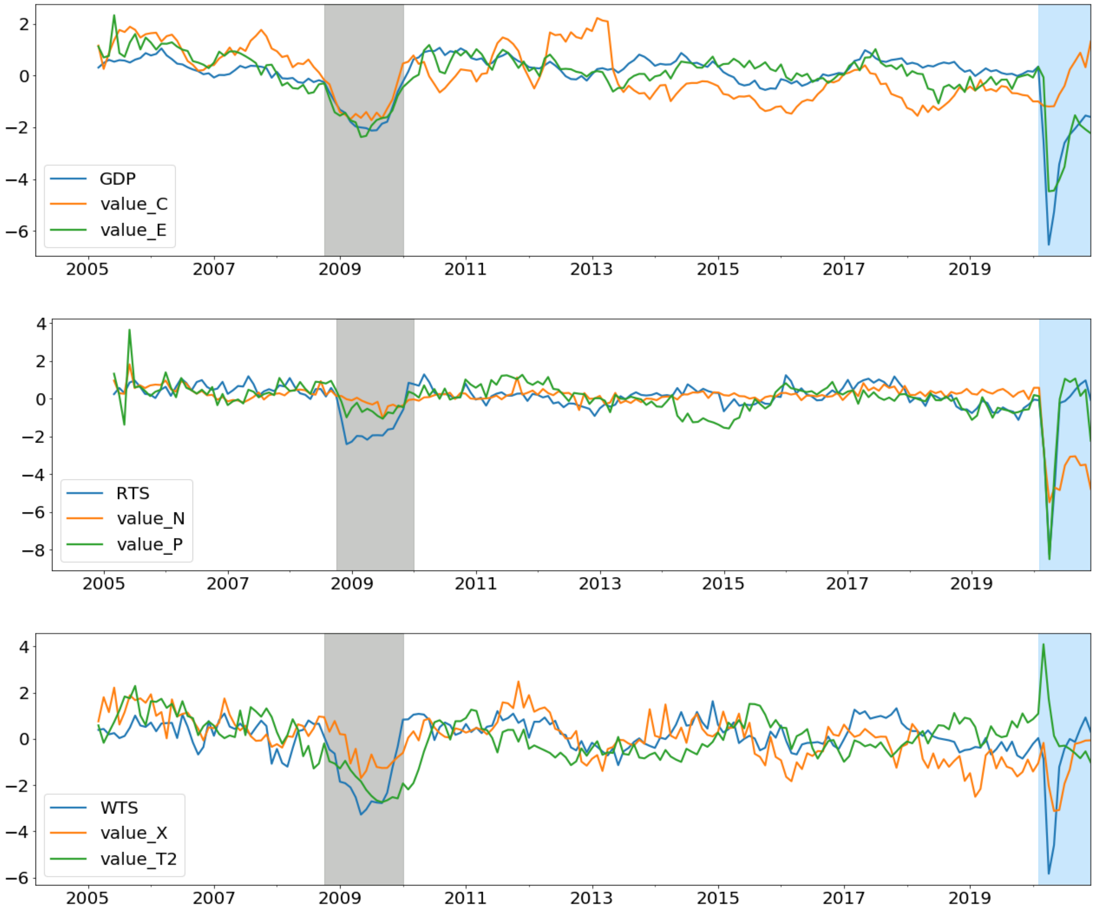

Figure 1: Standardization year-over-year growth rate comparisons of GDP, retail trade sales (RTS), and wholesale trade sales (WTS) with a few selected payments streams for the period between Mar 2005 to Dec 2020. Highlighted in gray is the global financial crisis period; blue shows the COVID-19 period. NOTE: C is AFT Credit, E is Encoded Paper, N is Shared ABM, P is POS Payments, X is Corporate Payments, and T2 is LVTS-T2 Payments. The *value* is the dollar amount.

[21] Similar behavior is observed in LVTS-T1, where a drastic rise in the value of payments is observed during the COVID-19 period due to extraordinary measure taken by the Bank of Canada under its quantitative easing policy (Bank of Canada 2020

3 Methodology

In this section, we briefly discuss the nowcasting models employed in this paper. First, we discuss the ordinary least squares (OLS) and the dynamic factor model (DFM). This is followed by a brief discussion on the ML models used in the paper.

Consider a set $X = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M\}$ of M predictors (also called features or independent variables) and a target $\mathbf{y}$ (dependent variable), each with N data (sample) points. This can be represented as a dataset $(X, \mathbf{y})$ where X is of size $N \times M$ and $\mathbf{y}$ is a vector of size $N \times 1$. Let us denote $\hat{\mathbf{y}}$ as the predicted target, which can be obtained using, for example, an OLS model as

$$\hat{\mathbf{y}}(X, \mathbf{w}) = \quad X\mathbf{w}, \tag{1}$$

where $\mathbf{w}$ is a vector of unknown coefficients (betas or weights) of size $M \times 1$. In OLS the objective is to minimize the residual sum of squares between the observed target variable $\mathbf{y}$ and the predicted target values $\hat{\mathbf{y}}$,

$$\min_{\mathbf{w}} \|\mathbf{y} - \hat{\mathbf{y}}(X, \mathbf{w})\|_2^2, \tag{2}$$

where $\|.\|_*$ is L_* norm. Such linear models have proven to be a valuable and straightforward models for prediction and they are commonly used due to its simplicity and interpretability. However, when some of the predictors are correlated, the OLS estimates become highly sensitive to random errors in the target. Moreover, the OLS can only model relationships linear in the parameters $\mathbf{w}$. Although the linearity assumption make them easy to interpret on a modular level, it generally does not perform well on wide, large and complex data sets (Hastie et al. 2009).

The dynamic factor models are a powerful approach to captures the common dynamics of a large set of predictors into a relatively small number of latent factors. It is a frequently preferred model for macroeconomic nowcasting and forecasting when dealing with a large set of predictors (Giannone et al. 2008; Stock and Watson 2016). Similar to Chernis and Sekkel 2017, we estimate the factors using the model of Bańbura and Modugno (2014), which can effectively handle a large number of predictors and the missing data. The basic representation of the model is:

$$X_t = \Lambda f_t + \varepsilon_t \tag{3}$$

$$f_t = A_1 f_{t-1} + \cdots + A_p f_{t-p} + u_t \tag{4}$$

where X_t is a set of predictors at time t, f_t is the unobserved factor at t, Λ is vector of factor loadings, ε_t is idiosyncratic disturbance at t, A_i are matrices of autoregression coefficients, and u_t is factor disturbance at t. The model parameters can be estimated by maximum likelihood using the Expectation-Maximization (EM) algorithm proposed in Bańbura and Modugno (2014). DFMs are successfully applied for economic monitoring and predictions around the world (Banbura et al. 2010; Stock and Watson 2016; Hindrayanto et al. 2016; Bragoli 2017) including for nowcasting Canada's GDP (Chernis and Sekkel 2017; Chernis et al. 2020).

3.1 Machine Learning Models for Nowcasting

To exploit the non-traditional and large-scale data sources, researchers have recently begun utilizing ML models for economic nowcating (Richardson et al. 2020; Maehashi and Shintani 2020; Chapman and Desai 2021). The ML models are shown to handle wide- and large-scale data efficiently and can manage collinearity. Furthermore, they are demonstrated to capture nonlinear interactions between the predictors and the target (Chakraborty and Joseph 2017; Coulombe et al. 2021).

We use some of the recently popularized parametric and non-parametric machine learning approaches such as elastic net (Zou and Hastie 2005), support vector machines (Smola and Schölkopf 2004), random forest (Breiman 2001; Liaw and Wiener 2002), gradient boosting (Friedman 2001), and feedforward artificial neural network (Bengio et al. 2009). For each considered model, there are many variations proposed in the literature; however, we have focused on the simpler version of each model. In the remaining part of this section, we give a high-level description of these models; for further details, refer to Appendix B.

The elastic net (ENT) is a regularized linear regression model. Here the objective is similar to that of the OLS (shown in Equation 2) with the addition of L_1 and L_2 penalties on how large the sum of the parameters $\mathbf{w}$ can get.[22] In an elastic net regression, the combination of L_1 and L_2 penalties allows for learning a sparse model while encouraging grouping effects, stabilizing regularization paths, and removing limitations on the number of selected variables (Zou and Hastie 2005).

Support vector regression (SVR) is another model useful for problems with multiple predictors. It uses a very different objective function compared to the OLS or ENT. The SVR is based on support vector machines. These are algorithms whose task is to find a hyperplane that separates the entire training dataset into, for example, two groups by using a small subset of training points (called support vectors). In the case where there is no such hyperplane, it is modified to minimize the number of misclassified points in every region (Burges 1998; Smola and Schölkopf 2004).

Another popular approach is random forest (RF) regression. It is a decision tree-based ensemble learning method built using a forest of many regression trees. It is a non-parametric approach that addresses the multicollinearity problem slightly differently from parametric approaches such as OLS or ENT. RF is a bagging (bootstrap aggregation) approach, i.e., each tree is independently built from a subset of the training dataset. Each sample could randomly select a subset of features from the available set of feature—helping in decorrelation. The final prediction is performed by averaging the predictions of all regression trees (Breiman 2001; Liaw and Wiener 2002).

Similar to the RF, gradient boosting (GB) regression is a tree-based non-parametric ensemble learning approach. However, unlike RF, GB is based on boosting in which a sequence of weak learners (decision trees) are built on a repeatedly modified version of the training dataset. The data modification at each boosting interaction consists of applying weights to each of the training samples, and for successive iterations, the sample weights are modified (Friedman 2001).

[22] A regression model that uses only the L_1 penalty is a Lasso regression, and a model that uses only the L_2 penalty is a Ridge regression (Hastie et al. 2009; Zou and Hastie 2005).

The feedforward artificial neural network (ANN) with hidden layers is multiple layers of artificial neurons sandwiched between input and output layers. In this approach, the data always moves forward through the network layers. The weighted sum of the first layers is typically passed through a nonlinear activation function resulting in a nonlinear function of the inputs. Then the outputs are sent to the next layer, and the process continues until the last layer. Once we get the final output from the network, we measure how good that output is compared to the target's actual value using an objective function, for example, mean squared error. Given these results, we go back and adjust the weights and biases of the network. Typically we need a large training dataset to achieve a good performance using ANN (Bengio et al. 2009; Goodfellow et al. 2016).

Note that there are many advanced versions of tree-based methods, such as LightGBM (Ke et al. 2017), and deep neural networks-based method, such as long short-term memory (LSTM) (Hochreiter and Schmidhuber 1997) are proposed in the literature. However, to efficiently utilize them for prediction, we often need a large training sample. Since our dataset is quite small (about 200 sample points), these models did not perform better than other models used in this paper.

3.2 Machine Learning Models Interpretability and Cross-validation

Interpretability could be essential for many classes of problems, including macroeconomic prediction. However, the use of complex ML models often leads to a loss of interpretability (Mullainathan and Spiess 2017; Chakraborty and Joseph 2017; Athey and Imbens 2019).

Some of the ML models employed in this paper, such as elastic net, support vector regression, and the tree-based ensemble learning models, can be interpreted up to a certain extent (Zou and Hastie 2005; Burges 1998; Breiman 2001; Friedman 2001). However, each method has different interpretability approaches, making it hard to compare against each other. To address these challenges, we use the Shapley value-based model agnostic approach—SHAP (SHapley Additive exPlanations)—developed in Lundberg and Lee (2017); Lundberg et al. (2020).

In SHAP, the Shapley value method from coalitional game theory[23] is used to fairly distribute the "payout" (= the prediction) among the "players" (= the predictors) (Lundberg et al. 2020). In nowcasting, the SHAP can be used to fairly distribute the ML model prediction among the set of predictor x_t at each time horizon t for local model interpretations. Furthermore, using Shapley values for each instance t, we can compute the global interpretation of the ML models in the form of feature importance for the entire training (in-sample) or testing (out-of-sample) data sets.

In Lundberg and Lee (2017) the authors propose two approaches based on the type of underlying process to compute the Shapley values: (1) KernelSHAP, a kernel-based estimation approach, which can be used for many ML models, such as elastic net, artificial neural network, and tree-based models; and (2) TreeSHAP, a computationally efficient approach for the Shapley value estimation for only tree-based ML models, such as decision trees, random forests and gradient boosted.

[23] The Shapley value method can be used to fairly distribute payouts among players based on their contribution to the total payout in a coalitional game (Shapley 1953; Osborne and Rubinstein 1994)

Since the SHAP methods are based on the Shapley value, which has game-theoretical foundations, these methods are trustworthy (Molnar 2020). However, the time required to estimate the Shapley values using KernelSHAP could increase exponentially with the number of predictors. This is not a big concern for our application because we have comparatively fever predictors and smaller sample sizes. The KernelSHAP method also suffers from collinearity in the features. This could be concerning for our case, given that a few predictors are correlated. These problems can be mitigated—up to a certain extent—using TreeSHAP, but only for tree-based models. Another challenge with these approaches is that it is possible to create intentionally misleading interpretations in order to hide the bias. Also, in some cases, the outcomes are easy to misinterpret and could lead to ambiguous conclusions. Therefore, SHAP should be used with caution (Slack et al. 2020; Molnar 2020). Further details on the SHAP and Shapley values are given in Appendix E.

Another issue commonly attributed to the use of ML models is the problem of overfitting. The ML models have many parameters which can be optimized to improve the prediction accuracy (commonly called hyperparameters tuning). Therefore, it is easy to tune the model to perform well on a specific set of data, for example, an in-sample training set. However, such models generally fail to perform well when applied to the unseen data (Hastie et al. 2009).

This problem can be alleviated using a k-fold cross-validation techniques (Hastie et al. 2009). In the standard approach, the training sample is randomly split into k-folds, then for each iteration, the $k-1$ folds are used for in-sample training, and the k^{th} fold is used for out-of-sample testing. Such a procedure effectively tunes the model parameters and avoids overfitting; However, the random splitting of the training sample breaks the order of the data and could lead to the use of future data points for the past predictions, which could give an unfair advantage to the model. For these reasons, it is not practical to use it in the *same way* to nowcasting models.

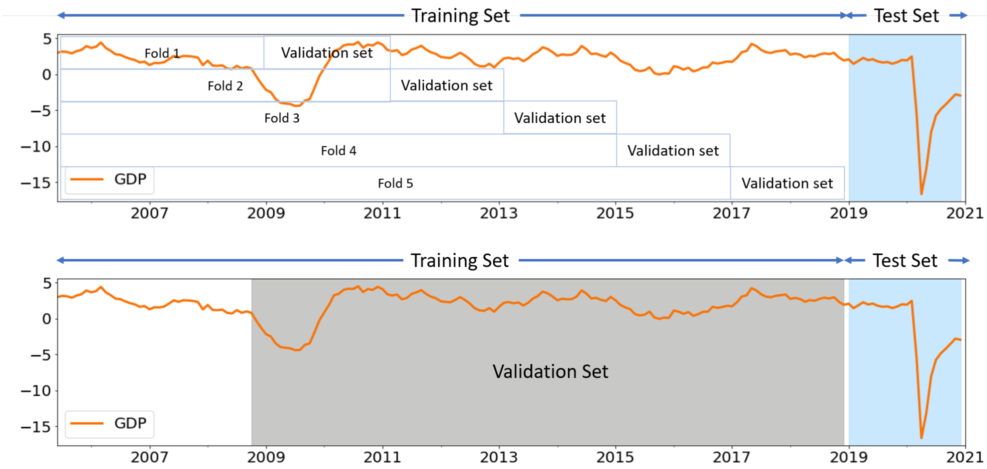

Figure 2: Top: schematic of standard expanding window approach for cross-validation in time-series. The dataset is divided into a training set with validation subsets and a test set (highlighted in blue). Bottom: schematic of the proposed approach, here the validation subsets are sampled from the gray highlighted area. The orange line shows the GDP growth rate.

This challenge can be mitigated using expanding window approach for cross-validation as depicted in Figure 2 (top). Here the end part of the training set, often called a validation set, is kept aside for model tuning and cross-validation.[24] This approach is useful for nowcasting during normal economic periods. However, in the cases where the test sample has an economic crisis period, but the validation sample does not, the traditional expanding window could be challenging because: (a) distribution of test and validation samples are quite different, and (b) the model is tuned only for normal periods; therefore might not perform well on out-of-sample crisis periods.

In this paper, we have devised a slightly altered version of expanding window approach tailored to the macroeconomic nowcasting models (Figure 2-bottom). We randomly sample n-points (one for each iteration) between two specified dates and use them as a validation sample (Figure 3). For each iteration of expanding window validation, only the data points that come before the chosen point are used for training—preserving the order of data and temporal dependency between the observations. In the current exercises, since we have the COVID-19 period in the test set, using a random sampling strategy leads us to include a few sample points from the global financial crisis period in the validation set. The proposed approach helps make the distribution of validation and test sets similar and assist in selecting a model that can perform well on both the normal and crisis periods. Also, it removes the restriction on the number of validation sets we can sample (Figure 2).

Furthermore, instead of using all payments streams in each model or manually selecting a few payments streams for the given macro indicator, we use a *data-driven* approach for predictor selections. We treat the number of payments steams p similar to a model parameter and use the expanding window cross-validation approach to optimally select the best p streams for each target variables based on their performance on the training and validation sets.[25]

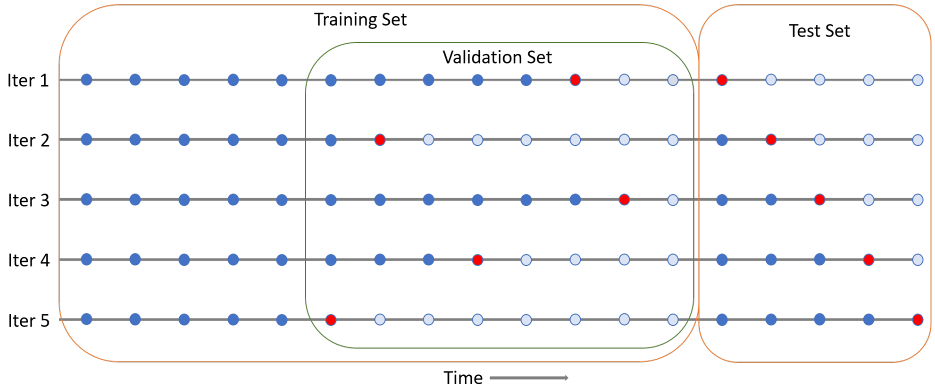

Figure 3: Schematic of expanding window approach for cross-validation (1-fold) and out-of-sample prediction. The available data is divided into the training set with a validation subset and the testing set. In each iteration the (●) represent the training data and (●) represent the test point.

[24] For each iteration of the expanding window, the training sample is increased by one period and then predict the next period from the validation set. Consequently, the model parameters can be chosen based on the model performance on the validation sets. See Appendix C for additional details.

[25] Further details of cross-validation and model selection are discussed in the Appendix D.

3.3 Model Training and Cases Specifications

We train the nowcasting models using the expanding window approach as schematically outlined in the Figure 3. First, we divide the dataset into two subsets: a training set (in-sample) for model training and a testing set (out-of-sample) for predictions. The OLS and DFM models are directly trained on the training set and used for predictions on the test set. We use Root Mean Square Error (RMSE) as the key performance indicator for the out-of-sample model evaluation.

Each ML model, which requires hyperparameters tuning and cross-validation, are trained using the following procedure (we call it randomized expanding window approach with *k*-folds):

1. From the training sample, we select two dates covering the wider range of training data[26] and randomly choose a set of *n* sample points as a validation set (where the *n* is of the same size as the test sample).

2. Thereafter, for each sample date in the validation set, we select all the sample points before that date for training and use the sample date for prediction (Figure 3). This way, we maintain temporal dependency and avoid using future data for the predictions in the past.

3. Next, for each model, we specify the grid for selected hyperparameters. Then, for each value of specified parameters, we iterate over the validation set and compute the validation RMSE.

4. Steps 2 and 3 are repeated *k* times for the same set of hyperparameters but with a different validation set randomly sampled from the training set using step 1 (*k* fold cross-validation).

5. Next, we select the best parameters, i.e., the parameters with the lowest average validation RMSE (averaged over *k* folds) for out-of-sample test set predictions.

6. Finally, the chosen model parameters are used for predictions on the test set by utilizing the standard expanding window approach over the training and test set (Figure 3).

As a benchmark (or base case), we employ a linear regression model using OLS. Here, we use the first available lagged target variable along with the latest available consumer price index (CPI), unemployment (UNE), Canadian financial stress indicator (CFSI)[27] and Conference Board's consumer confidence index (CBCI).[28] The CFSI and CBCI are available immediately after the end of the period, and they carry comprehensive and useful information about the macro indicators. Along with the CPI and UNE (which are available by one-to-two weeks of delay), these predictors make a strong benchmark to assess the information gain using the payments systems data.

[26] We choose the start date just before the global financial crisis period and end data just before the test set, then select *n* random data points between these two dates as a validation set. This helps us to include a few data points from the crisis period in each fold of validation set, and at the same time avoid using a large cross-validation sample.

[27] CFSI is computed using the data from the following seven market segments: the equity market, the Government of Canada bonds market, the foreign exchange market, the money market, the bank loans market, the corporate bonds market, and the housing market.

[28] The CBCI is based on the Conference Board's survey of Canadian households, which provide a measures consumers' levels of optimism on current economic conditions.

In the main case of interest, along with the predictors specified in the base case, we use the payments data listed in Table 1. Here, we first use the DFM to assess the marginal contribution of payments data when used in a sophisticated econometric model. Next, we test the usefulness of the various ML models discussed earlier in the section 3, and finally compare the ML models' performance against the benchmark case and DFM.

For all these cases, using a procedure similar to Giannone et al. (2008); Galbraith and Tkacz (2018), we perform nowcasting at three monthly time horizons, extending from the start of the month of interest (t) until the month before the official release ($t+2$). As we march in time, we include new predictors when they become available. For example, GDP nowcasting at time horizon t, i.e., on the first day of the month of interest, we use the latest available benchmark variables and the monthly aggregated payments data available at $t-1$. The model $\mathscr{F}$ can be specified as[29]

$$\widehat{GDP}_t = \mathscr{F}(GDP_{t-3},\ CPI_{t-2},\ UNE_{t-2},\ CFSI_{t-1},\ CBCC_{t-1},\ Payments_{t-1}). \tag{5}$$

Similarly, at the next nowcasting horizons $t+1$ and $t+2$, using the latest available predictors, the models can be specified as[30]

$$\widehat{GDP}_{t+1} = \mathscr{F}(GDP_{t-2},\ CPI_{t-1},\ UNE_{t-1},\ CFSI_t,\ CBCC_t,\ Payments_t). \tag{6}$$

$$\widehat{GDP}_{t+2} = \mathscr{F}(GDP_{t-1},\ CPI_t,\ UNE_t,\ CFSI_t,\ CBCC_t,\ Payments_t). \tag{7}$$

4 Results and Discussion

The payments data used for nowcasting exercises range from Mar 2004 to Dec 2020. The in-sample training period is Mar 2005 to Dec 2018 ($N = 166$ sample points)[31] and the out-of-sample testing period is Jan 2019 to Dec 2020 ($N = 24$). Our training set includes the 2008 global financial crisis period, and the test set combines a normal economic growth period (Jan 2019 - Feb 2020) and the part of the ongoing COVID-19 crisis period (Mar - Dec 2020). This allows us to examine our models' performance during both normal and crisis periods.

The year-over-year GDP, RTS, and WTS growth rates' nowcasting performance for the various cases outlined in the previous section are discussed next. Table 2 compare the nowcasting performance—in terms of out-of-sample RMSE—of the DFM and ML model (gradient boosting) on the main case against the benchmark models at t, $t+1$ and $t+2$ time horizons.

Our results suggest that the payments systems data in conjunction with ML models can provide notable reductions in nowcasting RMSEs for all three of the macro variables considered in this

[29] NOTE: GDP is released by two months lag, CPI and UNE are released with one-to-two weeks of lags, CFSI, CBCC and payments data are ideally available on next day of the end of the period.

[30] Note that at $t+2$ nowcasting horizon (on the first day of the month in which the target month's macro indicators will be released), we have $t+1$ months payments data; However, we do not include that because we are mainly interested in assessing the usefulness of t month's payment data to predict t month's macro variables.

[31] We lose the first one year of data after computing YOY growth rates.

Table 2: Out of Sample RMSE comparisons for seasonally adjusted YOY growth rate of macro variables at time horizon t-on the first day of the month of interest (top panel) $t+1$-on the first day after the month of interest (middle panel) and $t+2$-on the first day, two months after the month of interest (bottom panel)[a]

Target[b]	**Benchmark**[c]	**Main-DFM**[d]	**Main-ML**[e]	**RMSE Reduction (%)**[f]
GDP	4.58	3.95	3.70	19
RTS	7.88	7.40	7.38	7
WTS	6.34	5.81	5.74	10

Target	**Benchmark**	**Main-DFM**	**Main-ML**	**RMSE Reduction (%)**
GDP	3.97	2.98	2.43*	39
RTS	8.47	6.36	5.44*	35
WTS	7.17	6.18	4.28*	41

Target	**Benchmark**	**Main-DFM**	**Main-ML**	**RMSE Reduction (%)**
GDP	2.84	2.63	2.18	23
RTS	7.60	6.15	5.55	25
WTS	6.24	5.76	4.72	24

[a] In-sample training period: Mar 2005 to Dec 2018 ($p = 166$) and out-of-sample testing period: Jan 2019 to Dec 2020 ($p = 24$).

[b] GDP-Gross Domestic Product, RTS-Retail Trade Sales, WTS-Wholesale Trade Sales. Note that we use the latest available values of these targets. We also perform similar exercises by using target variables at first-release (real-time vintages); These results are presented in the Appendix G.

[c] For benchmark, we use OLS with CPI, UNE, CFSI, CBCC, and the first available lagged target variable (i.e., second lag at nowcasting horizon t).

[d] For the main-DFM case, we use payments data along with the predictors in the benchmark case. Similar to the model employed in Chernis and Sekkel (2017), we use the DFM model with two factors and one lag in the VAR driving the dynamics of those factors. The idiosyncratic components are assumed to follow an AR(1) process.

[e] We use gradient boosting regression (GBR), because it consistently performed better over other models. We select the model parameters using the cross-validation procedure outlined in the Appendix C and D, for example, the selected model for GDP nowcasting At $t+1$: *learning_rate* is 0.1, *max_depth* is 2, *n_estimators* is 1000. See Appendix B for further details of this model.

[f] Percentage reduction in RMSE over the benchmark model using the ML on the main case.

*, **, *** denote statistical significance at the 10, 5, and 1% level, respectively, for the Diebold-Marino test using the benchmark.

paper. Specifically, we get a 35 to 40% reduction in RMSE over the benchmark case in nowcasting GDP, RTS, and WTS at time horizon $t+1$. The main case predictions at this time horizon are statistically significant for the Diebold-Marino test using the benchmark.[32]

Comparatively, the information gain using the payments data is smaller at the nowcasting horizon t, i.e., when we use the first lag of payments data; and $t+2$, i.e., when the first lag of the target variables at $t-1$ is available along with the other benchmark indicators at t. In these cases, we get 7 to 25% reduction in RMSE over the benchmark in nowcasting GDP, RTS, and WTS. These results suggest that the payments data provide the most value in macroeconomic nowcasting when the given month's payments data is used to nowcast the same month's macro variables.

Next, we compare ML models against the DFM (Table 3). Overall, the DFM contributes to increasing prediction accuracy up to 25% at $t+1$.[33] However, in nowcasting GDP, RTS, and WTS at all three time horizons, the gradient boosting regression (GBR), elastic net (ENT), and feedforward artificial neural network (ANN) models—in many cases—perform better than DFM and other ML models considered in this paper. This is probably due to their ability to handle multiple predictors efficiently and capture sudden, large, and nonlinear interaction between the predictors and target variables during the COVID-19 crisis period. Overall, using payments data in the ML models, we get up to a 25% reduction in RMSE over the DFM with the payments data.

Visual comparisons of the best performing ML model against the benchmark model for in-sample and out-of-sample (highlighted in gray) predictions are depicted in Figure 4. Incorporating the payments data in ML models provides downturn and recovery indications much better than the benchmark model in both in-sample and out-of-sample periods. We conjecture that this is due to the new and timely information provided by the payments data and ML models' flexibility, allowing this data to provide better predictions during crisis periods.

Next, we separately test our models' out-of-sample performance during a normal time (Jan 19 to Feb 20) and the COVID-19 period (Mar 20 to Oct 20) of the test sample (see Table 4 in Appendix F).[34] We observe a higher gain using payments data during the time of crisis (up to 35% RMSE reduction) compared to the normal period of the test sample (15 to 25% reduction in RMSE) using payments data. These results suggest that the payments data is useful during normal periods, but its usefulness surges during crisis periods.

Lastly, we compare GDP nowcasting performance of our model with the real-time vintages (first releases) and the latest vintages (see Table 5 in Appendix G). Comparatively, the models using payments data perform better against the latest vintages. This makes sense, given that the latest vintages are more accurate compared to the real-time vintages. Therefore, we conclude that the payments data are effective in providing timely estimates of the key macro-indicators.

[32] We recognize that the Diebold-Mariano test has poor finite-sample properties; however, we use it to be comparable with similar papers where it has been used, for example, in Chernis and Sekkel (2017) and Aprigliano et al. (2019).

[33] In this case, we have used the DFM model with two factors. Including more factors did not improve our results. We note that the DFM model's performance, in some cases, is similar to the OLS model

[34] We use gradient boosting regression for this exercises, because of its consistency.

Table 3: Out of Sample RMSE comparisons of the DFM with ML models for seasonally adjusted YOY growth rate of macro variables at the horizons: t (top panel) $t+1$ (middle panel) and $t+2$ (bottom panel), for the main case.[a]

Target[b]	**DFM**[c]	**ENT**[d]	**SVR**[d]	**RFR**[d]	**GBR**[d]	**ANN**[d]
GDP	3.95	4.20	4.83	4.12	**3.70**[g]	4.26
RTS	7.40	8.18	8.07	8.55	**7.38**	7.69
WTS	5.81	6.59	6.81	6.71	**5.47**	6.08

Target	**DFM**	**ENT**	**SVR**	**RFR**	**GBR**	**ANN**
GDP	2.98	2.89	4.23	3.31	**2.43**	2.45
RTS	6.36	5.68	8.12	6.86	**5.44**	6.53
WTS	6.18	5.97	7.07	5.09	4.28	**3.15**

Target	**DFM**	**ENT**	**SVR**	**RFR**	**GBR**	**ANN**
GDP	2.63	2.30	4.28	3.01	**2.18**	2.25
RTS	6.15	**5.41**	8.41	7.11	5.55	6.01
WTS	5.76	5.14	6.91	5.24	4.72	**4.02**

[a] In-sample training period: Mar 2005 to Dec 2018 ($p = 166$) and out-of-sample testing period: Jan 2019 to Dec 2020 ($p = 24$).

[b] GDP-Gross Domestic Product, RTS-Retail Trade Sales, WTS-Wholesale Trade Sales. Note that we use the latest available values of targets for these exercises.

[c] For the DFM, we use payments data along with the predictors in the benchmark case. We use the DFM model with two factors and one lag in the VAR driving the dynamics of those factors. The idiosyncratic components are assumed to follow an AR(1) process.

[d] We use elastic net (ENT), support vector regression (SVR), random forest regression (RFR), gradient boosting regression (GBR), and artificial neural network (ANN). For these ML models, we select the model parameters and number of payments predictors based on target variables using the cross-validation procedure outlined in the section 3. Further details on these models are provided in Appendix B. Model selection and cross-validation procedures are detailed in Appendix C and D.

[g] The lowest out-of-sample prediction RMSE among the competing model is highlighted (in bold) for each case.

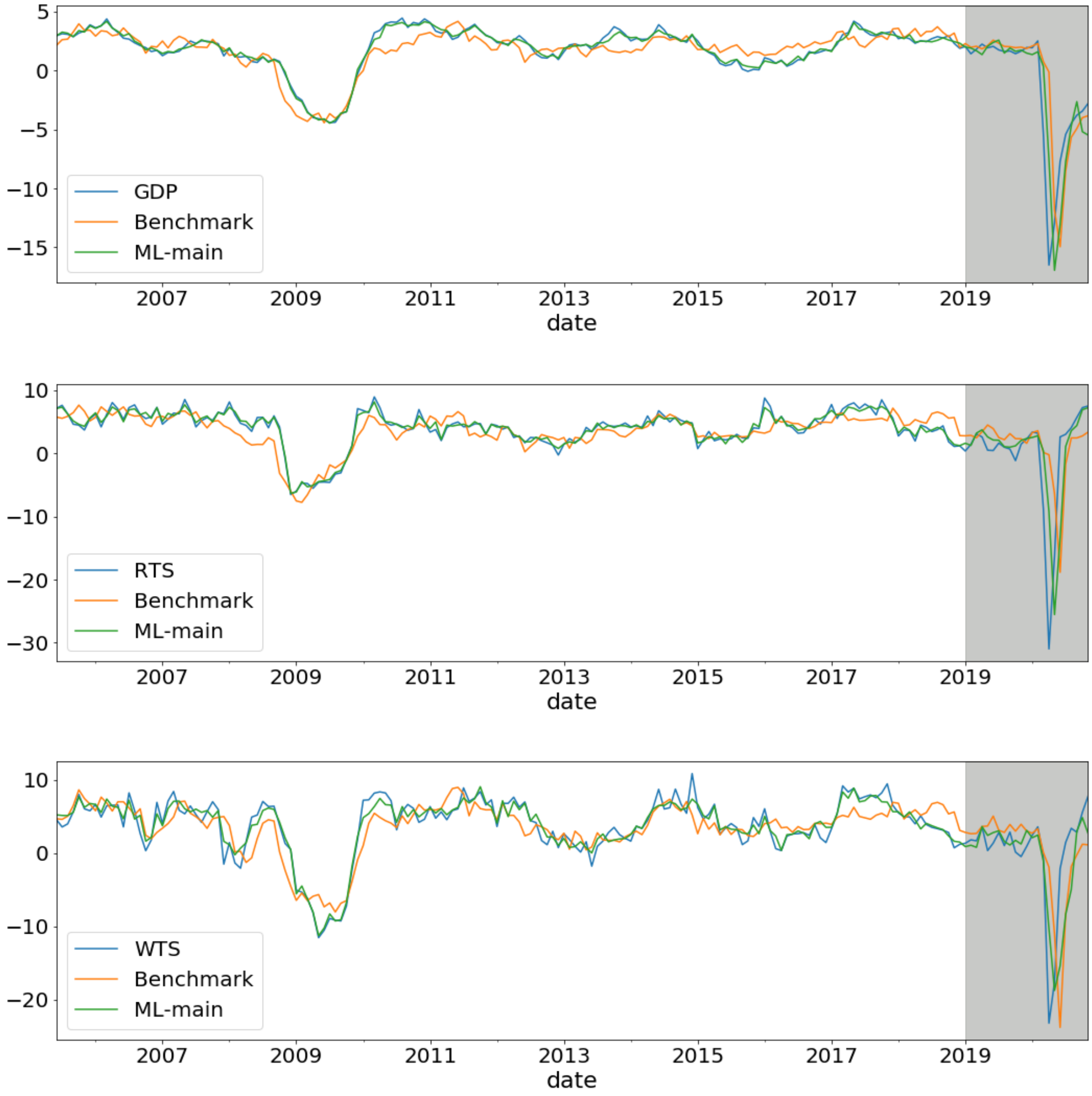

Figure 4: In-sample and out-of-sample predictions comparison of the ML-main case model (with lowest RMSE) with the benchmark model (the OLS with base case) for $t+1$ time horizon. The in-sample training period is Mar 2005 to Dec 2018 and the out-of-sample testing period is Jan 2019 to Dec 2020 (highlighted in gray).

4.1 Nowcasting models interpretation using SHAP

In the next part, we discuss the Shapley value-based interpretation of ML model predictions using the SHAP library Lundberg and Lee (2017); Lundberg et al. (2020). Here, we focus on nowcasting GDP at time horizon $t+1$ using the tuned gradient boosting model.[35] A similar procedure can be applied to the other target variables and ML models employed in this paper.[36]

For demonstration, we use the entire sample (Mar 2005 to Dec 2020) for training. In Figure 5, we plot SHAP global feature importance obtained by averaging the absolute Shapley values for each predictor across the training set (in-sample). This plot shows, on average, how much each feature influences the model prediction. These features are ranked according to their average influence (from high to low). For example, in the case of in-sample training data, GDP lag influences the most; However, the Encoded Paper (E) value stream also has a strong influence (on average changes the GDP growth rate by about 0.5 points). This is followed by the unemployment lag feature (UNE) and the sum of all ACSS streams (Allstreams value).

In Figure 6, we show the global feature importance plot for the COVID-19 period with high negative growth rates (Mar to Dec 2020). During this period, the Encoded Paper and POS Payments streams are more valuable predictors for GDP nowcasting. The GDP lag, a highly important feature for the entire training sample, loses its prediction power during the COVID-19 crisis period. A similar contribution of some of the payments streams is observed during the 2008 global financial crisis periods. These results suggest that the lagged macro indicators influence the GDP growth rates during the normal periods and contribute well to the prediction; however, they do not add much value during crisis periods such as the global financial crisis and COVID-19 shock. During such periods, the payments data becomes more valuable.

Next, using the SHAP "force" plots, we can compute local feature importance, i.e., how useful each feature during a given sample point in the training set. Such insights could be important for nowcasting exercises because, during each step of the expanding window approach (i.e., when we march in time by one month), the force plots could provide additional insights into each month's predictions by highlighting marginal contributions of individual predictors.

For instance, in Figure 7, we plot the Shapley values as forces for prediction on Feb 2020 and Mar 2020, respectively. Here, each Shapley value is an arrow that forces to increase (higher in red) or decrease (lower in blue) the prediction from the baseline (i.e., the average of all predictions). The size of these arrows indicates the magnitude of the Shapley value for that feature. These forces balance at the model prediction of that instance shown as $f(x)$. In Feb 2020, just before the pandemic started affecting Canada's economy, most of the payments predictors are positive (red) and pushing the GDP growth higher; However, during Mar 2020, i.e., in the first month of COVID-19 shock, most of the payments streams have a strong negative signal (blue) and pushing the GDP growth lower (closer to the actual target).

[35] We chose this model because it consistently gives better performance over other models.

[36] We discuss a few key interpretation results for nowcasting RTS and WTS at the end of this section.

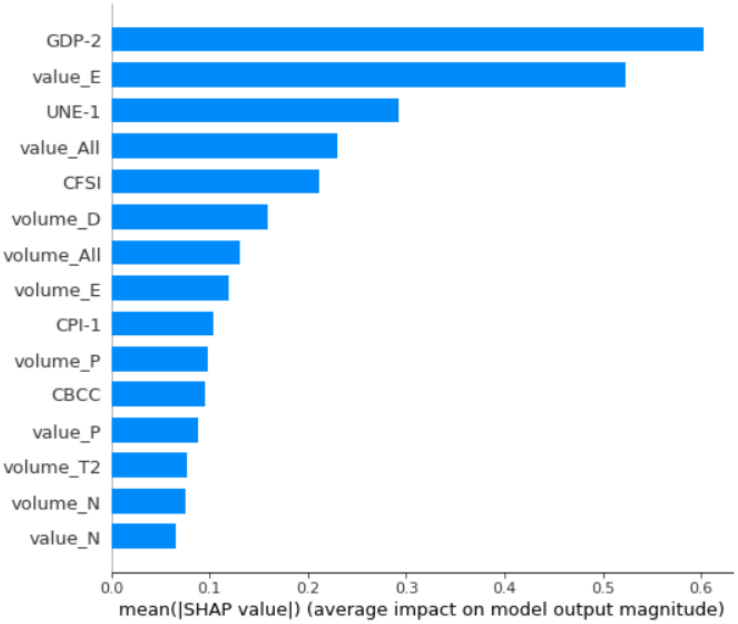

Figure 5: GDP: SHAP global feature importance measured as the mean absolute Shapley values of each instance in the entire training sample (Mar 2005 to Dec 2020). The features are ranked from high (top) to low (bottom) based on average Shapley values.

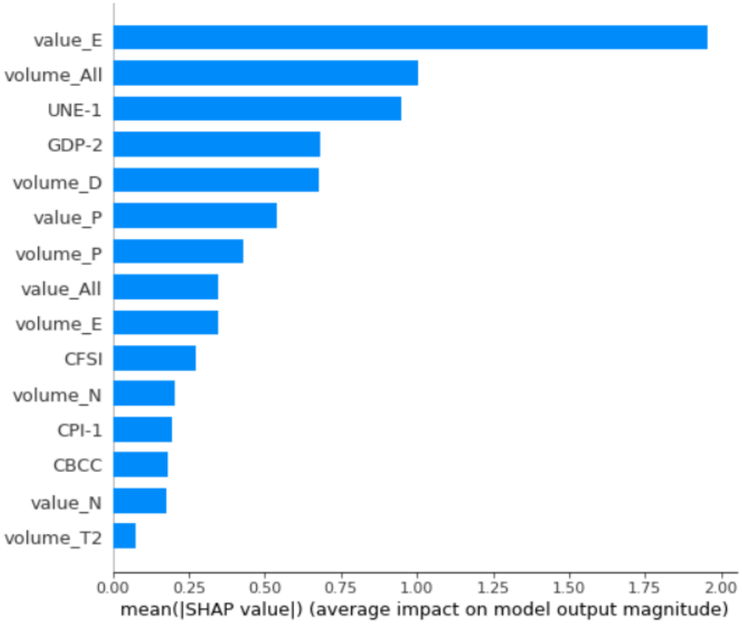

Figure 6: GDP: SHAP global feature importance measured as the mean absolute Shapley values of each instance in the training sample for the COVID-19 period (Mar 2020 to Dec 2020). The features are ranked from high (top) to low (bottom) based on average Shapley values.

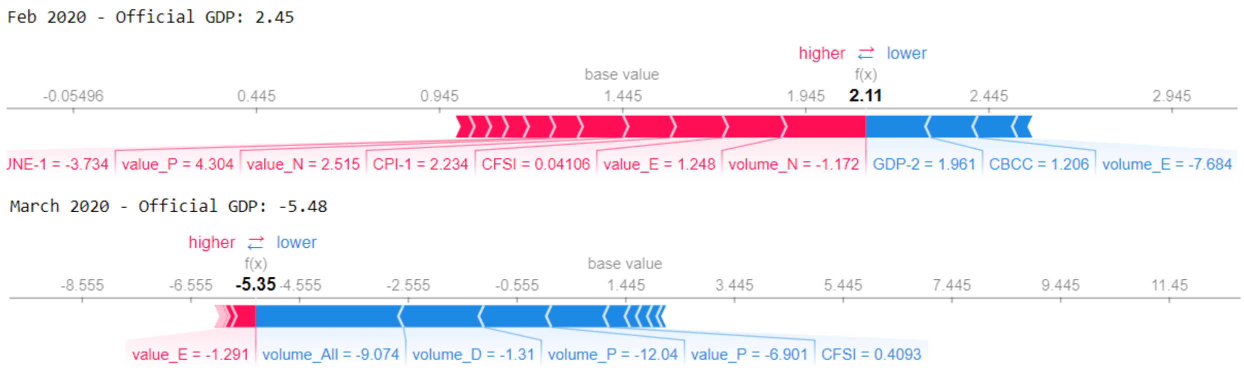

Figure 7: GDP: SHAP force plots for Feb 2020 (top) and March 2020 (bottom) where red arrows are positive Shapley values and blue are negative Shapley values. $f(x)$ is the model prediction, and the base value is the average of all predictions.

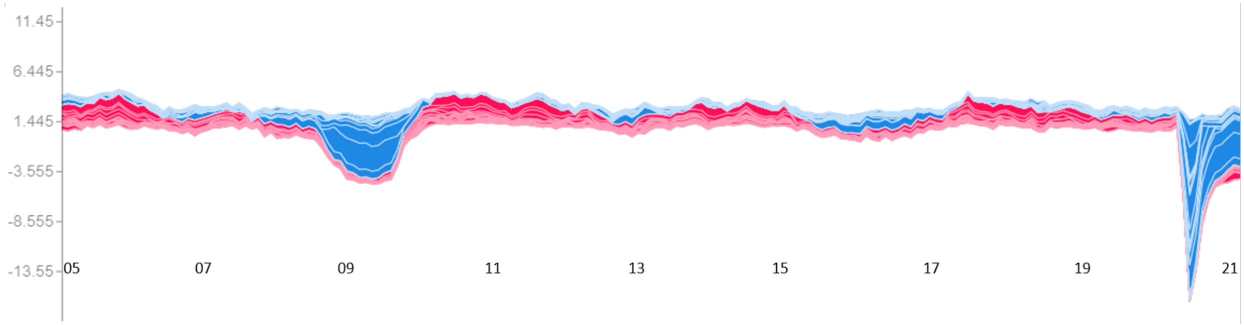

Figure 8: GDP: Clustered force plots for each instance in the training sample, i.e., monthly instance from Mar 2005 to Dec 2020 positioned on the x-axis. Red clusters are positive Shapley values that increase the prediction, and blue clusters are negative Shapley values that decrease the prediction.

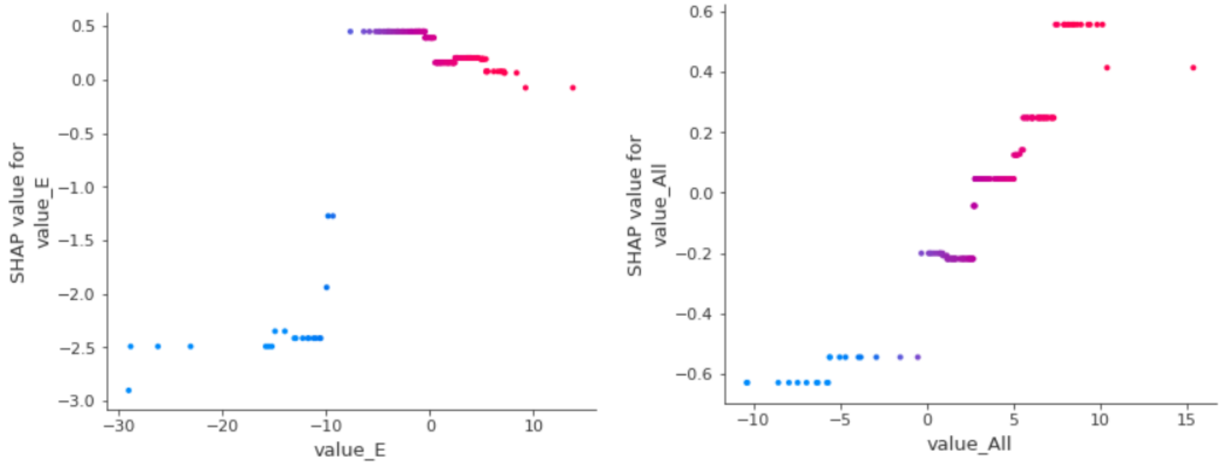

Figure 9: Dependence plots show Shapley value for each instance in the sample and corresponding feature value. On the left, we show a dependence plot for the Encoded Paper (E) value, and on the right, we show the dependence plot for the ACSS Allstream (All) value.

Figure 8 shows the force plots for each instance in the entire training sample, but they are rotated and stacked together vertically. We can observe red clusters of predictors with positive signals during positive economic growth periods and blue clusters with negative signals during the crisis periods such as the 2008 global financial crisis and COVID-19 shock. Such clustered signals could be valuable to track crises in real-time.

In Figure 9, we show the dependence plots for Encoded Paper value (left) and Allstream value (right). These plots capture the relationship between the feature values on the x-axis and the corresponding Shapley values on the y-axis. We can observe that the small and negative values of Encoded Paper growth rates provide higher contributions in Shapley values compare to the positive growth rates. However, both positive and negative growth rates of Allstreams value are contributing similarly (or symmetrically). The Encoded Paper plot (left) suggests that the contribution of payments data—in terms of the Shapley values—is small and linear during the periods of normal growth; however, during the periods of strong negative and positive growths, the contribution of this stream is asymmetrical and nonlinear.

Similar behavior is observed in nowcasting models for RTS and WTS using payments data and gradient boosting. In Figure 10 and 11, we plot SHAP global feature importance for the training set at $t+1$ time horizon for RTS and WTS, respectively. These plots suggest, in the case of RTS, the POS Payments (P) value highly influences the model prediction. This makes sense, given the POS Payments are commonly used for retail sales. In the case of WTS, the Allstreams (All) value stream highly impacts the model prediction along with the corporate-to-corporate bill payment (X) stream, highlighting the importance of those streams in predicting wholesale sales.

Finally, in Figure 12, we show the dependence plots of RTS with POS Payments value (left) and WTS with Allstream value (right). Here we also show how these payments streams get influenced by the Canadian financial stress index (CFSI). These plots suggest that, at high-stress levels, i.e., at high values of CFSI (showed in red) and negative payments growth rates, the signal from these payments streams are strong and their contribution—in terms of Shapley values—is high; However, for low levels of stress (showed in blue) and positive payments growth rates, the payments data contributions are positive but small. This confirms the asymmetrical and potentially nonlinear relationship between these payments streams and the corresponding macro variables.

5 Conclusions

We use comprehensive and timely payments systems data and machine learning models for macroeconomic nowcasting. The payments data provide economic information in real-time and help reduce dependence on lagged variables. Machine learning provides a set of tools to effectively process various payments streams and capture the sudden and large effects of a crisis. To improve the effectiveness of ML models, we use Shapley value-based approach for model interpretability, and device specialized cross-validation strategy to avoid model overfitting.

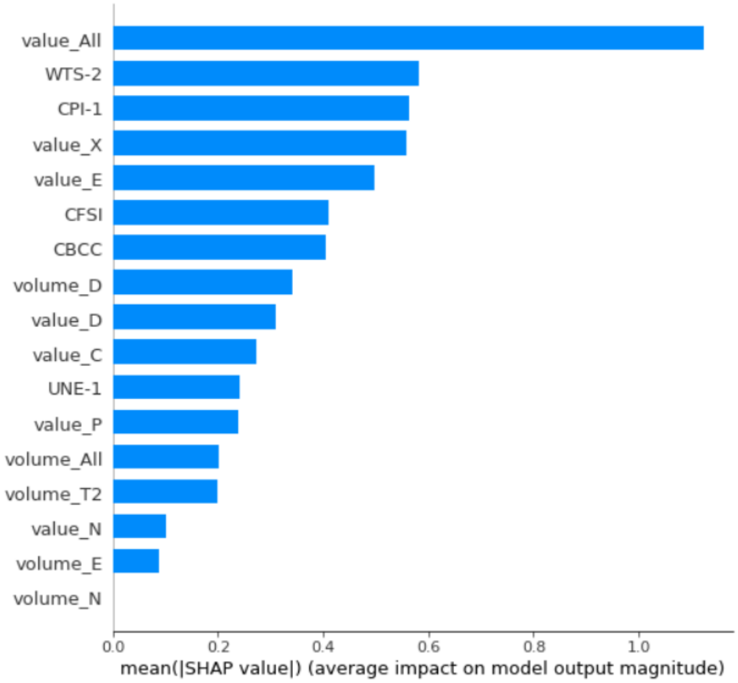

Figure 10: Retail Trade Sales (RTS): SHAP global feature importance measured as the mean absolute Shapley values of each instance in the entire training sample (Mar 2005 to Dec 2020).

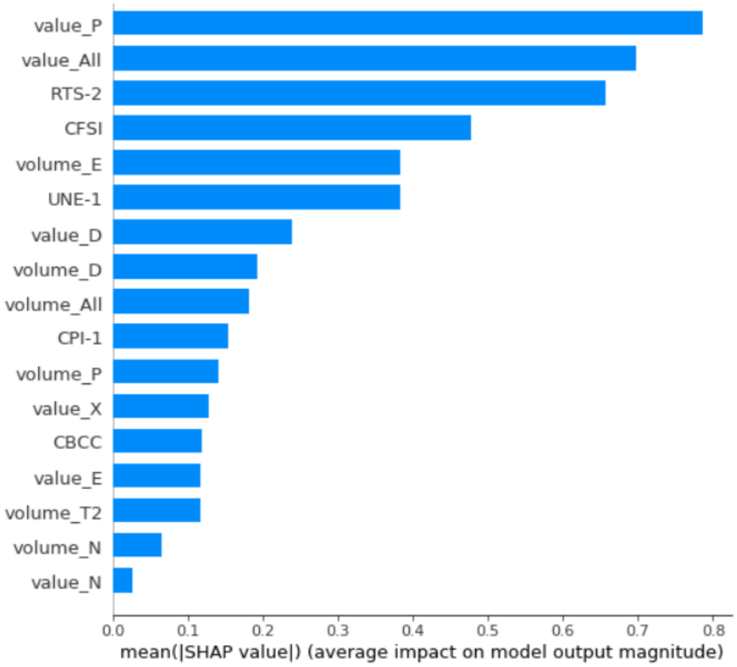

Figure 11: Wholesale Trade Sales (WTS): SHAP global feature importance measured as the mean absolute Shapley values of each instance in the entire training sample (Mar 2005 to Dec 2020).

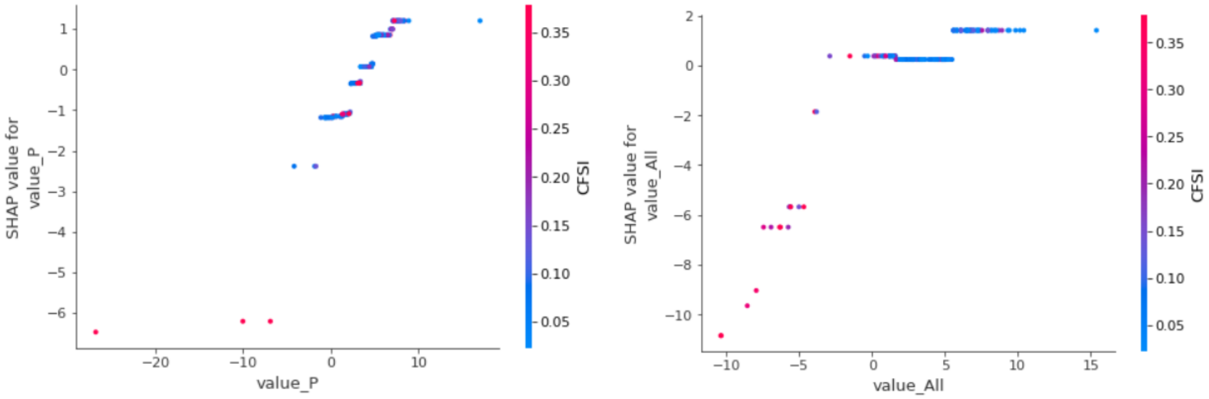

Figure 12: Dependence plots show Shapley value for each instance in the sample and corresponding predictor value. On the left, we show a dependence plot of RTS for the POS Payments value (P), and on the right, we show the dependence plot of WTS for the ACSS Allstream value (All). NOTE: CFSI is a Canadian financial stress index.

Our results suggest that the payments system data and ML models can lower nowcast errors significantly over a linear benchmark models. ML models out-of-sample performance is relatively higher during the COVID-19 crisis period compared to the pre-COVID period. We observe that ML models' performance changes slightly for different nowcasting cases; however, the gradient boosting model gives a consistently good performance. The importance of payments data (especially Encoded paper streams) increases during crisis periods. Nonetheless, some of the payments streams influence the model predictions during the normal periods. Overall, using payments data in nonlinear ML models, we get up to a 50% reduction in RMSE over the linear benchmark.

We also demonstrated the Shapley value-based SHAP approach's usefulness to get insights into the ML model predictions at each nowcasting step and for the entire training sample. This could be a valuable tool in macroeconomic nowcasting, especially during crisis periods. Also, we find that the proposed cross-validation technique can help reduce overfitting and improve prediction accuracy in macroeconomic nowcasting models. To conclude, this paper substantiates the use of payments data and ML models for macroeconomic prediction and provides a set of econometric tools to overcome the associated challenges.

References

Ahmed, N. K., A. F. Atiya, N. E. Gayar, and H. El-Shishiny (2010). An empirical comparison of machine learning models for time series forecasting. *Econometric Reviews 29*(5-6), 594–621.

Angelini, E., G. Camba-Mendez, D. Giannone, L. Reichlin, and G. Rünstler (2011). Short-term forecasts of euro area gdp growth. doi: 10.1111/j.1368-423X.2010.00328.x.

Aprigliano, V., G. Ardizzi, L. Monteforte, et al. (2019). Using the payment system data to forecast the economic activity. *International Journal of Central Banking 15*(4), 55–80.

Arjani, N. and D. McVanel (2006). A primer on canada's large value transfer system.

Athey, S. (2017). The impact of machine learning on economics. In *Economics of Artificial Intelligence*. University of Chicago Press.

Athey, S. and G. W. Imbens (2019). Machine learning methods that economists should know about. *Annual Review of Economics 11*(1), 685–725. doi: 10.1146/annurev-economics-080217-053433.

Banbura, M., D. Giannone, and L. Reichlin (2010). Nowcasting. Technical report, ECB Working Paper No. 1275. https://ssrn.com/abstract=1717887.

Bańbura, M. and M. Modugno (2014). Maximum likelihood estimation of factor models on datasets with arbitrary pattern of missing data. *Journal of Applied Econometrics 29*(1), 133–160. doi: 10.1002/jae.2306.

Bank of Canada (2020, April). Monetary policy report – April 2020. Technical report, Bank of Canada. https://www.bankofcanada.ca/wp-content/uploads/2020/04/mpr-2020-04-15.pdf.

Barnett, W., M. Chauvet, D. Leiva-Leon, L. Su, et al. (2016). Nowcasting nominal GDP with the credit-card augmented divisia monetary. Technical report, The Johns Hopkins Institute for Applied Economics. https://ideas.repec.org/p/pra/mprapa/73246.html.

Bengio, Y. et al. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning 2*(1), 1–127.

Bok, B., D. Caratelli, D. Giannone, A. M. Sbordone, and A. Tambalotti (2018). Macroeconomic nowcasting and forecasting with big data. *Annual Review of Economics 10*, 615–643. doi: 10.1146/annurev-economics-080217-053214.

Bounie, D., Y. Camara, and J. W. Galbraith (2020). Consumers' mobility, expenditure and online-offline substitution response to COVID-19: Evidence from French transaction data. Technical report, CIRANO Working Papers 2020s-28. https://ssrn.com/abstract=3588373.

Bragoli, D. (2017). Now-casting the Japanese economy. *International Journal of Forecasting 33*(2), 390–402. doi: 10.1016/j.ijforecast.2016.11.004.

Breiman, L. (2001). Random forests. *Machine learning 45*(1), 5–32. doi: 10.1023/A:1010933404324.

Buono, D., G. L. Mazzi, G. Kapetanios, M. Marcellino, and F. Papailias (2017). Big data types for macroeconomic nowcasting. *Eurostat Review on National Accounts and Macroeconomic Indicators 1*(2017), 93–145. https://ec.europa.eu/eurostat/cros/system/files/euronaissue1-2017-art4.pdf.

Burges, C. J. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery 2*(2), 121–167. doi: 10.1023/A:1009715923555.

Carlsen, M. and P. E. Storgaard (2010). Dankort payments as a timely indicator of retail sales in Denmark. Technical report, Danmarks Nationalbank Working Papers 66. doi: http://hdl.handle.net/10419/82313.

Carvalho, V. M., S. Hansen, A. Ortiz, J. R. Garcia, T. Rodrigo, S. Rodriguez Mora, and P. Ruiz de Aguirre (2020). Tracking the covid-19 crisis with high-resolution transaction data. https://www.repository.cam.ac.uk/bitstream/handle/1810/310898/cwpe2030.pdf?sequence=5.

Chakraborty, C. and A. Joseph (2017). Machine learning at central banks. Technical report, Bank of England Working Paper No. 674. https://ssrn.com/abstract=3031796.

Chapman, J. and A. Desai (2021). Using payments data to nowcast macroeconomic variables during the onset of COVID-19. Technical report, Bank of Canada Staff Working Paper 2021-2. https://www.bankofcanada.ca/2021/01/staff-working-paper-2021-2.

Chernis, T., C. Cheung, and G. Velasco (2020). A three-frequency dynamic factor model for nowcasting Canadian provincial GDP growth. *International Journal of Forecasting 36*(3), 851–872. doi: 10.1016/j.ijforecast.2019.09.006.

Chernis, T. and R. Sekkel (2017). A dynamic factor model for nowcasting Canadian GDP growth. *Empirical Economics 53*(1), 217–234. https://www.bankofcanada.ca/wp-content/uploads/2017/02/swp2017-2.pdf.

Chetty, R., J. N. Friedman, N. Hendren, M. Stepner, et al. (2020). How did COVID-19 and stabilization policies affect spending and employment? A new real-time economic tracker based on private sector data. Technical report, National Bureau of Economic Research. doi: 10.3386/w27431.

Choi, H. and H. Varian (2012). Predicting the present with Google Trends. *Economic Record 88*, 2–9. doi: 10.1111/j.1475-4932.2012.00809.x.

Coulombe, P. G., M. Marcellino, and D. Stevanovic (2021). Can machine learning catch the covid-19 recession? *Available at SSRN 3796421*. doi: 10.2139/ssrn.3796421.

Duarte, C., P. M. Rodrigues, and A. Rua (2017). A mixed frequency approach to the forecasting of private consumption with ATM/POS data. *International Journal of Forecasting 33*(1), 61–75. doi: 10.1016/j.ijforecast.2016.08.003.

Duprey, T. (2020). Canadian financial stress and macroeconomic conditions. Technical report, Bank of Canada. https://www.bankofcanada.ca/2020/06/staff-discussion-paper-2020-4/.

Friedman, J., T. Hastie, and R. Tibshirani (2001). *The elements of statistical learning*, Volume 1. Springer series in statistics. New York: Springer. doi: 10.1007/978-0-387-84858-7.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. *Annals of Statistics 29*(5), 1189–1232. doi: 10.1214/aos/1013203451.

Galbraith, J. and G. Tkacz (2007). Electronic transactions as high-frequency indicators of economic activity. Technical report, Bank of Canada. https://www.bankofcanada.ca/wp-content/uploads/2010/02/wp07-58.pdf.

Galbraith, J. W. and G. Tkacz (2018). Nowcasting with payments system data. *International Journal of Forecasting 34*(2), 366–376. doi: 10.1016/j.ijforecast.2016.10.002.

Giannone, D., L. Reichlin, and D. Small (2008). Nowcasting: The real-time informational content of macroeconomic data. *Journal of Monetary Economics 55*(4), 665–676. doi: j.jmoneco.2008.05.010.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep learning*. MIT Press.

Greenwood, R., S. G. Hanson, A. Shleifer, and J. A. Sørensen (2020). Predictable financial crises. Technical report, National Bureau of Economic Research.

Hamilton, J. D. (2011). Calling recessions in real time. *International Journal of Forecasting 27*(4), 1006–1026. doi: 10.1016/j.ijforecast.2010.09.001.

Hastie, T., R. Tibshirani, and J. Friedman (2009). *The elements of statistical learning: data mining, inference, and prediction*. Springer Science & Business Media.

Hindrayanto, I., S. J. Koopman, and J. de Winter (2016). Forecasting and nowcasting economic growth in the euro area using factor models. *International Journal of Forecasting 32*(4), 1284–1305. doi: 10.1016/j.ijforecast.2016.05.003.

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

Kapetanios, G. and F. Papailias (2018). Big data & macroeconomic nowcasting: Methodological review. Technical report, Discussion Papers ESCoE DP-2018-12, Economic Statistics Centre of Excellence.

Ke, G., Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu (2017). Lightgbm: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems*, pp. 3146–3154. doi: https://lightgbm.readthedocs.io/en/latest/.

Koop, G. and L. Onorante (2019). Macroeconomic nowcasting using Google probabilities. *Topics in identification, limited dependent variables, partial observability, experimentation, and flexible modeling: Part A (Advances in Econometrics) 40*, 17–40. doi: RePEc:eme:aecozz:s0731-90532019000040a003.

Kuhn, M., K. Johnson, et al. (2013). *Applied predictive modeling*, Volume 26. Springer.

Kwan, A. C. and J. A. Cotsomitis (2006). The usefulness of consumer confidence in forecasting household spending in Canada: A national and regional analysis. *Economic Inquiry 44*(1), 185–197. doi: 10.1093/ei/cbi064.

Liaw, A. and M. Wiener (2002). Classification and regression by random forest. *R news 2*(3), 18–22.

Lundberg, S. M., G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee (2020). From local explanations to global understanding with explainable AI for trees. *Nature Machine Intelligence 2*(1), 2522–5839. doi: 10.1038/s42256-019-0138-9.

Lundberg, S. M. and S.-I. Lee (2017). A unified approach to interpreting model predictions. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (Eds.), *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf.

Maehashi, K. and M. Shintani (2020). Macroeconomic forecasting using factor models and machine learning: an application to Japan. *Journal of the Japanese and International Economies 58*, 101104. doi: 10.1016/j.jjie.2020.101104.

Molnar, C. (2020). *Interpretable machine learning*. Lulu. com.

Mullainathan, S. and J. Spiess (2017). Machine learning: an applied econometric approach. *Journal of Economic Perspectives 31*(2), 87–106. doi: 10.1257/jep.31.2.87.

Osborne, M. J. and A. Rubinstein (1994). *A course in game theory*. MIT press.

Paturi, P. and C. Chiron (2020). Canadian payments: Methods and trends 2020. Technical report, Payments Canada Report. https://www.payments.ca/sites/default/files/paymentscanada_canadianpaymentsmethodsandtrendsreport_2020.pdf.

Pedregosa, F., G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, et al. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research 12*, 2825–2830.

Richardson, A., T. van Florenstein Mulder, and T. Vehbi (2020). Nowcasting GDP using machine-learning algorithms: A real-time assessment. *International Journal of Forecasting*.

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games 2*(28), 307–317.

Slack, D., S. Hilgard, E. Jia, S. Singh, and H. Lakkaraju (2020). Fooling lime and shap: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, pp. 180–186.

Smola, A. J. and B. Schölkopf (2004). A tutorial on support vector regression. *Statistics and Computing 14*(3), 199–222. doi: 10.1023/B:STCO.0000035301.49549.88.

Spange, M. (2010). Can crises be predicted. *Danmarks National*.

Stock, J. and M. Watson (2016). Chapter 8 - dynamic factor models, factor-augmented vector autoregressions, and structural vector autoregressions in macroeconomics. Volume 2 of *Handbook of Macroeconomics*, pp. 415–525. Elsevier. doi: 10.1016/bs.hesmac.2016.04.002.

Varian, H. R. (2014). Big data: New tricks for econometrics. *Journal of Economic Perspectives 28*(2), 3–28.

Vrontos, S. D., J. Galakis, and I. D. Vrontos (2020). Modeling and predicting us recessions using machine learning techniques. *International Journal of Forecasting*. doi: 10.1016/j.ijforecast.2020.08.005.

X13 Reference Manual (2017). *X-13ARIMA-SEATS Reference Manual*, version 1.1. Technical report, Time Series Research Staff, Center for Statistical Research and Methodology, U.S. Census Bureau, Washington, DC. https://www.census.gov/ts/x13as/docX13AS.pdf.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Statistical Methodology 67*(2), 301–320. doi: 10.1111/j.1467-9868.2005.00503.x.

A Overview of ACSS and LVTS Payments Instruments

The historical list of payment streams processed through the ACSS payment system. Note: the first letter indicates the stream-ID, then the stream label followed by a short description.

- A: ABM Adjustments - POS payment items used to correct errors from shared ABM network transactions (Stream N)

- B: Canada Savings Bond - Part of Government items. It includes bonds (Series 32 and up and Premium Bonds) issued by the Government of Canada. It started in April 2012.

- C: AFT Credit - Direct deposit such as payroll, account transfers, government social payments, business to consumer non-payroll payments, etc.

- D: AFT Debit - Pre-authorized debit (PAD) payments such as bills, mortgages, utility payments, membership dues, charitable donations, RRSP investments, etc.

- E: Encoded Paper - Paper bills of exchange which includes cheques, inter-member debits, money orders, bank drafts, settlement vouchers, paper PAD, money orders etc.

- F: Paper-Based Remittances - These are used for paper bill payments, that is MICR-encoded with a CCIN, for credit to a business. This stream is similar to electronic bill payments (Stream Y).

- G: Receiver General Warrants - Part of Government Items. Paper payment items payable by the Receiver General for Canada. It started in April 2012.

- H: Treasury Bills and Old-style Bonds - Part of Government paper items. Certain Government of Canada paper payment items such as Treasury bills, old-style Canada Savings Bonds, coupons, etc. It started in April 2012.

- I: ICP Regional Image Captured Payment - Items entered into the ACSS/USBE on a regional basis. It started in Oct 2015.

- J: Online Payments - Electronic payments initiated using a debit card through an open network, most commonly the internet, to purchase goods and services. It started in June 2005.

- K: Online Payment Refunds - Credit payments used to credit a Cardholder's Account in the case of refunds or returns of an Online Payment (Stream J). It started in June 2005.

- L: Large-value Paper - This is similar to Stream E with value cap; starting in Jan 2014, this stream merged into E

- M: Government Direct Deposit - Recurring social payments such as payroll, pension, child tax benefits, social security, and tax refunds. It started in April 2012.

- N: Shared ABM Network - POS debit payments used to withdraw cash from a card-activated device.

- O: ICP National - Image Captured Payments are electronically imaged paper items that can be used to replace the physical paper item: cheques, bank drafts, etc.

- P: POS Payments - Point-of-service payment items resulting from the point-of-sale purchase of goods or services using a debit card

- Q: POS Return - Credit payments used to credit a cardholder's account in the case of refunds or returns of a POS payment (Stream P)

- S: ICP Returns National - National image captured payment returned items entered into the ACSS/USBE on a national basis. It started in Oct 2015.

- U: Unqualified Paper Payment - Paper items that are all other bills of exchange that do not meet Canada Payments Association requirements for Encoded Paper classification

- X: EDI Payment - Electronic data interchanges are an exchange of corporate-to-corporate payments such as purchase orders, invoices, and shipping notices

- Y: EDI Remittances - Electronic data interchange remittances are used for Electronic Bill Payments such as online bill payments and telephone bill payments

- Z: Computer Rejects - Encoded paper items whose identification and tracking information could not be verified through automated processes

The LVTS settles payments through two tranche T1 and T2. There are different types of payments which include both interbank and third-party funds transfers. It also includes transactions to and from the Bank of Canada (Refer to Arjani and McVanel 2006 for more details.)

- Foreign exchange payments and also payments related to the settlement of the Canadian-dollar leg of FX transactions undertaken in the Continuous Linked Settlement (CLS) system;

- Payments related Canadian-dollar-denominated securities the CDSX

- Payments related to the final settlement of the ACSS

- Large- value Government of Canada transactions (federal receipts and disbursements) and transactions relating to the settlement of the daily Receiver

- The Bank of Canada's own large-value payments and those of its other clients which includes Government of Canada, other central banks and certain international organizations.

B Machine Learning Models

In this section, we briefly discuss the machine learning models employed for nowcasting. For each considered model, there are many variations proposed in the literature; however, we have focused on the basic version of each model. Note that all models are implemented using the Scikit-learn machine learning library (Pedregosa et al. 2011). See Appendix C for more details on the model training, tuning, and cross-validation procedures.

B.1 Elastic Net Regularization

Elastic net is a regularized linear regression model. In ENT, the objective is similar to that of the OLS with the addition of L_1 and L_2 penalties. A regression model that uses only the L_1 penalty is called a Lasso regression, and a model that uses only the L_2 penalty is called a Ridge regression. In ENT, the combination of L_1 and L_2 penalties allows for learning a sparse model like Lasso, where only a few of the weights are non-zero. It also maintains the advantages of the Ridge regression, such as encouraging grouping effects, stabilizing regularization paths, and removing limitations of the number of selected variables (Zou and Hastie 2005; Hastie et al. 2009).

Consider a set $X = \{\mathbf{x^1}, \mathbf{x^2}, \ldots, \mathbf{x^M}\}$ of M attributes (independent variables) and a target $\mathbf{y}$ (dependent variable) and denote $\hat{\mathbf{y}}$ as the predicted target. With these specifications, in ENT, the objective function to minimize is

$$\min_{\mathbf{w}} \|\mathbf{y} - \hat{\mathbf{y}}(X, \mathbf{w})\|_2^2 + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2, \tag{8}$$

where $\mathbf{w}$ is a vector of unknown coefficients, and $\|.\|_*$ is L_* norm. This procedure can be viewed as a penalized least squares method with penalty factor $\lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2$. The ENT is particularly useful with multiple correlated features. Note that we explore and tune the following parameters: λ_1 and λ_2 by controlling constant α that multiplies the penalty terms, mixing parameter $l1\_ratio$ and the maximum number of iterations. For other parameters, we use the default values (see Scikit-learn library documentation for details (Pedregosa et al. 2011)).

B.2 Support Vector Regression

Support vector regression is another model useful for the problems with multiple predictors. It uses a different objective function compared to the OLS or ENT. The SVR is based on support vector machines where the task is to find a hyperplane that separates the entire training dataset into, for example, two groups by using a small subset of training points (called support vectors). In SVR the goal is to find a function, for instance, a linear function $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b$ (where b is a bias and $i = 1, 2, \ldots N$), that has at most ε deviation from the actual $\mathbf{y}$ for all the training data. Therefore the objective function to minimize is

$$\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}|\mathbf{y}_i - f(\mathbf{x}_i)|_{\varepsilon}, \tag{9}$$

subject to

$$\mathbf{y}_i - f(\mathbf{x}_i) \leq \varepsilon \tag{10}$$

$$f(\mathbf{x}_i) - \mathbf{y}_i \leq \varepsilon, \tag{11}$$

where N is the number of training samples and C is a regularization parameter constant (Smola and Schölkopf 2004). A different type of kernel function (linear, polynomial, sigmoid, etc.) can be specified for the decision function; therefore, it is versatile. For further details of SVM theory and formulation, refer to Smola and Schölkopf 2004; Hastie et al. 2009. Note that we explore and tune the following hyperparameters: kernel type, polynomial degree of the polynomial kernel function, and regularization parameter constant C and ε. We use the default values for other parameters (refer to Pedregosa et al. 2011 for details).

B.3 Random Forest

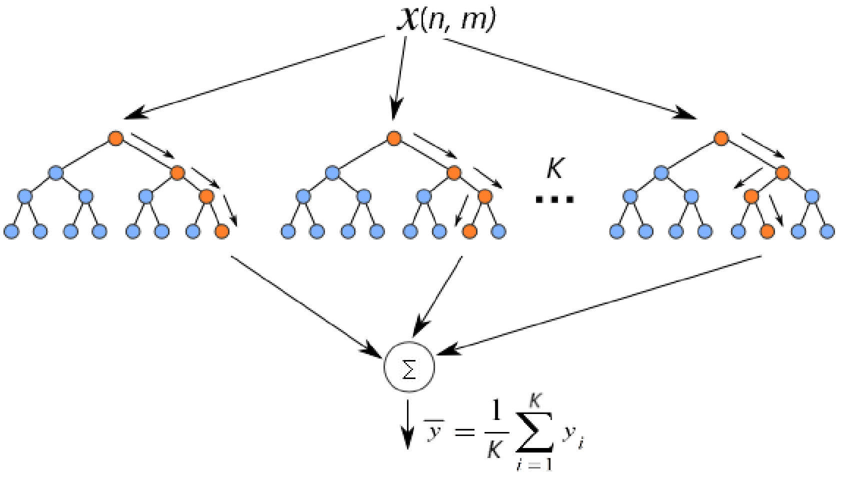

Figure 13: Random forest with K trees using n samples and m features for each tree.

Another popular approach is the random forest regression. It is a decision tree-based ensemble learning method built using a forest of many regression trees. It is a non-parametric method and hence approaches the multicollinearity problem slightly differently from parametric approaches such as OLS or ENT. In RF, each tree is independently built from a bootstrapped subset of the training dataset. Each bootstrap sample could randomly select a subset of features from the available set or the full features set. The final prediction is performed by averaging the predictions of all regression trees. The procedure is visually depicted in Fig. 13. The two levels of randomness (i.e., a random subset of the sample and the features) incorporated to build decision trees can help to reduce variance in the predictions. RF has been shown to handle highly non-linear interactions between multiple predictors and a target variable (Breiman 2001; Liaw and Wiener 2002).

Note, we explore and tune the following hyperparameters: the number of trees in the forest *n_estimators*, the maximum depth of the tree *max_depth*, and the minimum number of samples required to split an internal node *min_samples_split*. We use the default values for other parameters (refer to Pedregosa et al. 2011 for details).

B.4 Gradient Boosting

Similar to the random forest, gradient boosting (GB) regression is a tree-based non-parametric ensemble learning approach. It is a general technique of boosting in which a sequence of weak learners (for example, small decision trees) are built on a repeatedly modified version of the training dataset. The data modification at each boosting interaction consists of applying weights to each of the training samples, and for successive iterations, the sample weights are modified. Basically, the next learner is fit on the residual of the previous learner (Friedman 2001; Friedman et al. 2001).

Gradient Boosting Regression Trees are additive models whose prediction $\hat{\mathbf{y}}$ for a given input X for each instance i can be written as

$$\hat{\mathbf{y}}_i = H_p(X_i) = \sum_{1}^{p} h_p(X_i), \tag{12}$$

where h_p are weak learners, for example, decision trees (Friedman et al. 2001) and p is number of learners. The model $H_P(X)$ is built as

$$H_p(X) = H_{p-1}(X) + \gamma h_p(X), \tag{13}$$

where the γ is learning rate used to regularize the contribution of each new weak learner and the newly added weak learner h_p (tree) is used in order to minimize a sum of losses L_p:

$$h_p = {}^{\text{arg}}{}_{\mathbf{p}}{}^{\text{min}} L_p. \tag{14}$$

Note: we explore and tune the following hyperparameters: The number of trees in the forest *n_estimators*, the maximum depth of the tree *max_depth*, and the learning rate—which helps shrink the contribution of each tree. We use default values for all other parameters (Pedregosa et al. 2011). Both random forest and gradient boosting techniques are interpretable up to a certain extent. These models use decision trees as their base learners. These decision trees perform feature selection from the provided set by selecting appropriate split points. This information can be used to measure the importance of each feature (see Pedregosa et al. 2011 for additional details).

B.5 Feed-Forward Artificial Neural Network

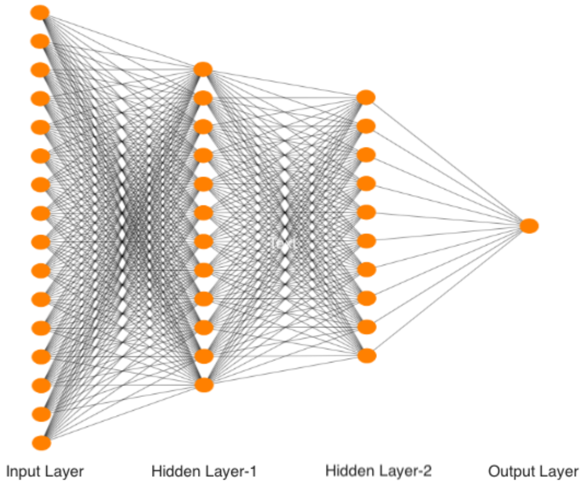

Figure 14: Schematic of densely connected feed forward neural network with two hidden layers.

A feed-forward artificial neural network with hidden layers is multiple layers of artificial neurons sandwiched between input and output layers as depicted in Figure 14. In a feed-forward ANN, the data always moves forward through the network layers. It starts in the input layer; for instance, each of the input feature instance $\mathbf{x}_i$ is multiplied by their corresponding layer's weight $\mathbf{w}$. Then, the weighted sum of these inputs $\mathbf{w}^T\mathbf{x}_i + b$ (where b is a bias) is passed through a non-linear activation function σ resulting in a non-linear function of the inputs $\sigma(\mathbf{w}^T\mathbf{x}_i + b)$. Then the outputs are sent to the next layer. This process continues until the last layer. Once we get the final output from the network, let us denote it as $\hat{\mathbf{y}}$, we measure how good that output is compared to the actual value of the target $\mathbf{y}$. This is done by using an objective function, for example, mean squared error. Given these results, we go back and iterative adjust the weights and biases of the network to optimize the objective function. For further details on the activation function and optimization procedure, refer to Bengio et al. (2009); Goodfellow et al. (2016).

The higher the number of layers, the deeper the network is; therefore, it is generally referred to as the deep neural network (DNN). The multilayer architectures enable a combination of features from lower layers, potentially modeling complex data with fewer units. Therefore, the DNN can be used to model complex non-linear relationships between the input and output. However, DNN requires tuning of a large number of hyperparameters as the number of hidden layers grows; therefore, generally, it needs a large training dataset to achieve a good performance.

Note: we use Scikit-learn's multi-layer perception (*MLPRegressor*), and we explore and tune the following hyperparameters: The number of neurons in the hidden layers *hidden_layer_sizes*, the activation function for the hidden layer *activation*, the learning rate schedule for weight updates, and the default values for other parameters (Pedregosa et al. 2011).

C Model Parameter Selection and Cross-Validation

The hyperparameter tuning and cross-validation of each ML model employed in this paper are performed using the randomized expanding window approach with k-folds as follows:

1. Split the original dataset into a training set and test set (Figure 15). In our case, the training set is Mar 2005 - Dec 2018, and the test set is Jan 2019 to Dec 2020.

2. Select two dates in the training set to randomly sample validation set. To include the global financial crisis period, we choose validation sample between Oct 2008 to Dec 2018 and sample 24 points (same size as test set) between these two dates as the validation set.

3. Specify the hyperparameters to tune and select the range for each parameter. See Appendix B for individual model parameters selected for tuning.

4. Using the selected parameters grid, for each fold of validation sample, do the following:
   (a) For each iteration in the expanding window, select a data point from the randomly sampled validation set and use the sample up to that point for training (Figure 3).
   (b) Fit the model on the selected training sample.
   (c) Using the trained model, predict for the selected sample point in the validation set.
   (d) Repeat steps a, b and c for each point in the validation set.

5. After finishing iterating over chosen validation set, compute the validation RMSE.

6. Repeat steps 4 and 5 for k-times (in our case $k = 5$) each with different validation set randomly sampled from the training set using step 2.

7. Compute the average validation RMSE over the k-folds.

8. Select the parameters for which the average validation RMSE is smallest.

9. Use the tuned model to get the RMSE for the testing set by re-utilizing the standard expanding window approach as illustrated in Figure 3.

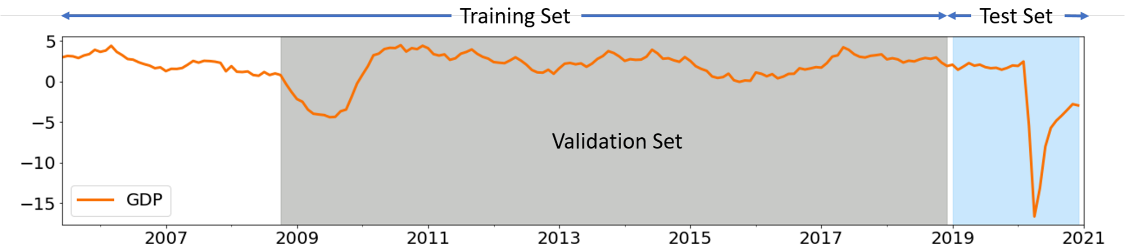

Figure 15: Schematic of data splits for cross-validation. The dataset is divided into a training set with a validation sub set (sampled from highlighted gray area), and a test set (highlighted in blue).

D Feature Selection

To select k best predictors from the set of available attributes, we employ the ***SelectKBest*** method from Scikit-learn (Pedregosa et al. 2011). This method removes all but the k highest-scoring features using univariate linear regression tests. It is a linear model for testing the individual effect of each of many regressors. To select K-best variables, it employs the following steps: First, the correlation between each predictor and the target is computed. Next, the computed correlations are converted to F-scores (using the F-test), then to p-values. Finally, these F-scores with p-values are used to select k highest-scoring features.

In Figure 16, we plot the scores of a few of the selected ***value*** streams (top) and ***volume*** streams (bottom) for GDP over the expanding window for the period ranging from Oct 2008 to Dec 2020. The prediction scores for most of the value and volume streams are high during the global financial crisis (GFC). The scores are steady and low during normal times (2011 to 2019) except for Encoded value (E), Allstream value (All), and LVTS-T2 volume (T2), for which scores remain high. During the COVID19-crisis (Mar to Dec 2020), however, we see opposite behaviour in the prediction scores of few streams. For example, AFT Credit (C) and LVTS-T2 value streams have strong prediction scores during the GFC. However, their scores are weak during the COVID-19 period. Similarly, the ABM stream (both value and volume) has low scores during the GFC, but, the scores are high during the COVID-19 period.

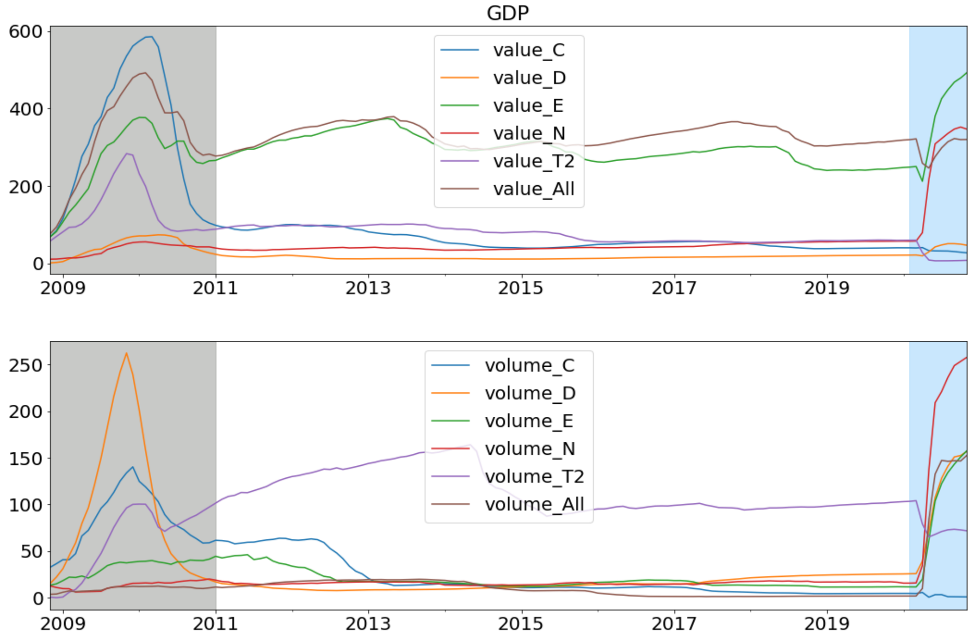

Figure 16: The F-score of a few selected payments streams (values-top, volumes-bottom) for GDP nowcasting. Higher scores mean a high prediction value. These plots are obtained after each training session of the expanding window approach, ranging from Oct 2008 to Dec 2020. Highlighted in gray is the GFC period; blue shows the COVID-19 period.

E The Shapley Values and SHAP for Model Interpretation

The Shapley values is a method from coalitional game theory which provides a way to fairly distribute the *payout* among the *players* by computing the average marginal contribution of each player across all possible coalitions (Shapley 1953; Osborne and Rubinstein 1994).

For a coalitional games (N, v), where, N is a finite set of players indexed by i, and v is utility function or payoff function, the Shapley value can be obtained by this theorem which satisfy the symmetry, dummy and additivity axioms (Osborne and Rubinstein 1994):

$$\phi_i(N,v) = \underbrace{\frac{1}{N!}\sum_{S\subseteq N\setminus\{i\}}}_{\text{average over all } S} \underbrace{|S|!\left(|N|-|S|-1\right)!}_{\text{possible coalitions}} \underbrace{\left[v(S\cup\{i\})-v(S)\right]}_{\text{marginal value}}$$

At a high level, the above equation can be split into three parts. The last part of the equation (the marginal value) gives the marginal contribution of an individual player i when added to a coalition S that does not have i. The middle part provides the way to compute different possible ways in which we could have formed the coalitions. Then, we take an average over possible ways that we could have done the marginal value calculation.

The SHAP (SHapley Additive exPlanations) proposed by Lundberg et al. 2020 uses the Shapley values to explain the model predictions in terms of marginal contribution of each predictor. The SHAP specifies the explanation of model $\mathscr{F}$ as a linear model of coalitions:

$$\mathscr{F}(S) = \phi_0 + \sum_{i=1}^{M} \phi_i S_i \tag{15}$$

where $S \in \{0,1\}^M$ is coalition vector with maximum M coalitions and ϕ_i the Shapley value for i^{th} player. In S the entry 1 means corresponding player is present and 0 means player is absent.

For illustration, consider nowcasting is a "game" then the Shapley values can be used to fairly distribute the *payout* (= the prediction) among the *players* (= the predictors). Note: for the computation of the Shapley values in the SHAP, the zero means the corresponding predictor is absent; in that case, the absent predictors' value is replaced by a random value from its sample (Lundberg et al. 2020; Molnar 2020). The procedure is further illustrated as follows:

1. Consider a nowcasting problem with three predictors (Figure 17) in a prediction model (it could be any model) to predict a target (for instance, monthly GDP growth).

2. The average prediction of the model, i.e., the base value is 0.2, and for the current instance (for example, a month t), our model predicts GDP growth 0.5.

3. By computing the Shapley values for all possible coalitions among these predictors, we can explain the difference between actual prediction (0.5) and the base value (0.2) in terms of each predictor's contribution.

4. In the current example: predictor-1 increases the growth rate by 0.5 percentage points, and predictor-2 is pushing it down by 0.3 points, and predictor-3 contributes +0.1 points. Thus, together these three predictors increase the prediction by +0.3 points from the average predictions of the entire sample.

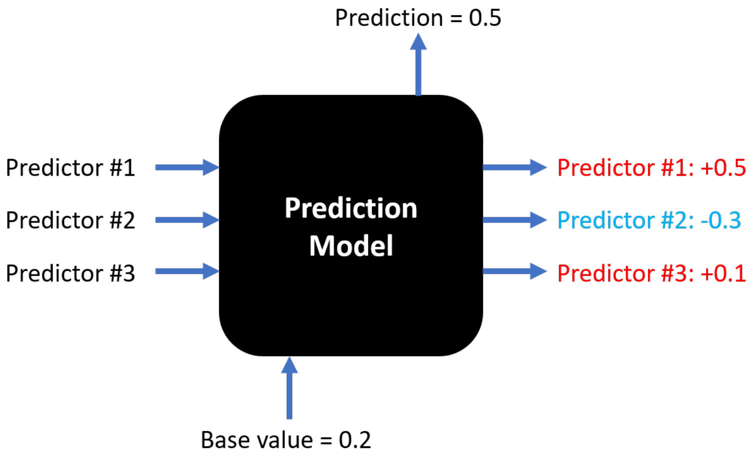

Figure 17: The SHAP explainer provides marginal contribution of each predictor

The SHAP values tell us which predictor contributes the most in the current instance of the prediction, i.e., a local interpretation. Similarly, by using the Shapley values for each instance in the sample, we could get the average contribution of each predictor over that sample. That could give us a global interpretation of the model in terms of feature importance. However, it is important to remember that these are only for the chosen model, and they do not explain the causality.

The SHAP package developed by Lundberg and Lee 2017; Lundberg et al. 2020 provides various tools to visualize the Shapley values computed for various ML models commonly used for predictions. For instance, force plots or clustered force plots (Figure 7 and 8) are useful for local interpretation, i.e., at each instance of prediction. The feature importance plots (Figure 5 and 6) and summary plots are useful for global model interpretations. Also, the dependence plots (Figure 9) could be valuable for understanding the relationships between given predictors and the targets in terms of the Shapley values.

The SHAP, although a powerful tool developed based on theoretical foundations for ML model interpretability, there are few pitfalls, and it should be used with caution (Molnar 2020; Slack et al. 2020). For example, the KernelSHAP is computationally intensive and could be very slow for problems with a large number of predictors. Also, it is sensitive to colinearity in the predictors. The TreeSHAP has overcome some of these challenges to a certain extent; however, it brings other challenges (Molnar 2020). Furthermore, as shown by Slack et al. 2020, it is possible to miss use such ad-hoc tools to hide model biases. However, the authors conclude that the SHAP is less prone to such problems than a few other interpretation tools.

F Nowcasting Performance for Normal and Covid-19 Periods

In this section, we separately test our models' out-of-sample performance during a normal time (Jan 19 to Feb 20) and the COVID-19 period (Mar 20 to Oct 20) of the test sample highlighted in gray and blue, respectively in Figure 18. For demonstration, we use gradient boosting regression for these exercises. We observe a higher gain using payments data during the time of crisis (up to 35% RMSE reduction) compared to the normal period of the test sample (15 to 25% reduction in RMSE) using payments data (Table 4). These results demonstrate the usefulness of payments data during normal periods and crisis periods.

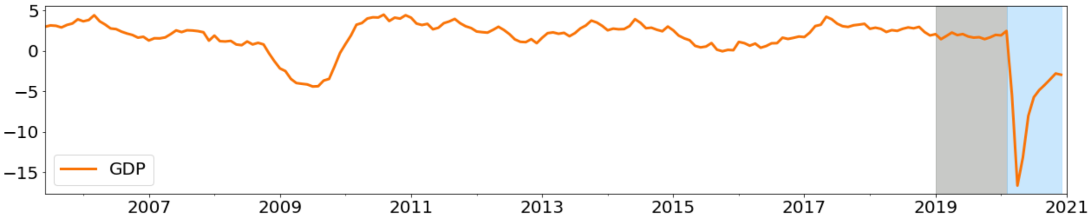

Figure 18: The test sample of GDP nowcasting exercises is divided into two sets: the pre-Covid-19 test set (highlighted in gray) and the Covid-19 test set (highlighted in blue).

Table 4: Out of sample RMSE comparisons for seasonally adjusted YOY growth rates of GDP, RTS and WTS at nowcasting horizon $t+1$ using the gradient boosting model[a]

Targets	Pre-COVID-19 test set[b]	COVID-19 test set[c]
GDP	16	34
RTS	14	35
WTS	27	37

[a] At $t+1$ time horizon, we use current, i.e., t month's payments data, to predict the same month's macro variables on the first day of the subsequent month.

[b] For the pre-Covid-19 test set (or normal period): In-sample training period: Mar 2005 to Dec 2018 and out-of-sample testing period: Jan 2019 to Feb 2020. Those numbers show the percentage gain over benchmark cases for the same period. We use OLS with CPI, UNE, CFSI, CBCC, and the first available lagged target variable for the benchmark.

[c] For Covid-19 test set (or crisis period): In-sample training period: Mar 2005 to Feb 2020 and out-of-sample testing period: Mar 2020 to Dec 2020. Those numbers show the percentage gain over benchmark cases for the same period.

G Nowcasting Performance for First and Latest Vintages

In this section, we compare the GDP nowcasting performance of our model with the real-time vintages (first releases) and the latest vintages (both shown in Figure 19). Comparatively, the models using payments data perform better against the latest vintages (we get smaller RMSEs). However, the gains are small (Table 5). This makes sense, given that the latest vintages are more accurate compared to the real-time vintages. Note: the performance gain is higher (about 10%) at $t+1$ nowcasting horizon compared to other time horizons.

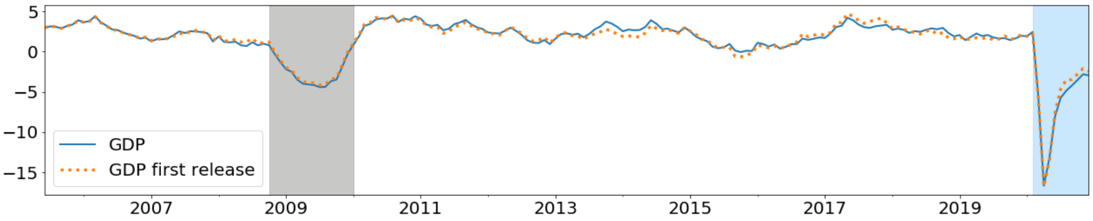

Figure 19: YOY seasonally adjusted GDP growth rates comparison for the first releases with latest releases. Highlighted in gray is the 2008 financial crisis period, and blue is the Covid-19 period.

Table 5: Out of sample RMSE comparisons for seasonally adjusted YOY growth rate of GDP at nowcasting horizon t, $t+1$, and $t+2$ using gradient boosting model[a]

Nowcasting Horizon[b]	Latest Vintages[c]	Real-time vintages[d]
t	3.73	3.88
$t+1$	2.61	2.92
$t+2$	2.66	2.68

[a] In-sample training period: Mar 2005 to Dec 2018 and out-of-sample testing period: Jan 2019 to Dec 2020.

[b] Nowcasting horizons: t is on the first day of the month of interest (top panel), $t+1$ is on the first day after the month of interest (middle panel), and $t+2$ is on the first day, two months after the month of interest (bottom panel)

[c] We use the latest available monthly levels of seasonally adjusted GDP from Statistics Canada Tables 36-10-0434-01

[d] We use the historical real-time vintages (available as of Mar 2020) of seasonally adjusted monthly GDP from Statistics Canada Tables 36-10-0491-01.