Macroeconomic Predictions using Payments Data and Machine Learning[*]

James T.E. Chapman and Ajit Desai[†]

Bank of Canada

December 26, 2022

Abstract

This paper aims to demonstrate that non-traditional and timely data, such as retail and wholesale payments, with the aid of nonlinear machine learning approaches such as gradient boosting, can provide policymakers with sophisticated models to accurately estimate key macroeconomic indicators in near real time. Moreover, we employ a set of econometric tools to mitigate *overfitting* and *interpretability* challenges in machine learning models to improve their effectiveness for policy use. Our models with comprehensive payments data, nonlinear methods, and tailored cross-validation approaches help improve out-of-sample macroeconomic nowcasting accuracy up to 40%—with higher gains during the COVID-19 period. We observe that the contribution of payments data for macroeconomic predictions is small and linear during low and normal growth periods. However, due to its timeliness, the payments data contribution is large, asymmetrical, and nonlinear during strong negative or positive growth periods.

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

[*] The opinions expressed herein are those of the authors and do not necessarily reflect those of the Bank of Canada. We thank Poclaire Kenmogne for his excellent assistance on this project. We also thank Alessandro Bitetto, Dalibor Stevanovic, Ingomar Krohn, Jonathan Chiu, Narayan Bulusu, Nicolas Woloszko, Pierre-Yves Yanni, Segun Bewaji, and Zhentong Lu for their comments. In addition, we thank discussants and participants of the following conferences for their comments and suggestions: ASSA/AEA 2023 Annual Meeting Econometrics Session. IAAE Annual Conference (2021), 7th RCEA Time Series Workshop, Big Data and Machine Learning in Finance Conference (2021), 24th Central Bank Macroeconomic Modelling Workshop, Advanced Analytics: New Methods and Applications for Macroeconomic Policy Conference (2021), Time Series in Finance Workshop at ICAIF (2021), Conference on Non-traditional Data, Machine Learning and Natural Language Processing in Macroeconomics (2021), and the seminar at the Bank of Canada.[†]Corresponding author: adesai@bankofcanada.ca

1 Introduction

Consumers are increasingly adopting electronic payments; this has dramatically accelerated due to the COVID-19 pandemic (Paturi and Chiron 2020). In the process, vast amounts of data have been generated. Much of this data is available in nearly real time. Concurrently, recent advances in the field of machine learning (ML) provide a set of advanced econometric tools to analyze non-traditional data and nonlinear relationships, bringing new opportunities to efficiently process large-scale payments data. Thus, our objective in this paper is to demonstrate the usefulness of payments data and ML models to predict the economy's short-term dynamics—known as nowcasting.

Timely prediction of the economy's short-term dynamics is a vital input into every economic agent's decision-making process. However, it is difficult for several reasons. For instance, many different data series are needed to describe the state of the economy adequately, but many of these data series, particularly official national account statistics, are released with significant lags (Giannone et al. 2008; Angelini et al. 2011). This problem is especially difficult during times of crisis, such as the 2008 global financial crisis (GFC) and the COVID-19 pandemic, primarily because of the large and nonlinear economic impact of crises and the unconventional policy responses needed for their mitigation (Spange 2010; Hamilton 2011). During such times, traditional models are inadequate because realizations of target variables are far from their average values (Vrontos et al. 2020; Coulombe et al. 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; Foroni et al. 2020; Babii et al. 2021; Cimadomo et al. 2022). We combine both new data and 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 the following five ML models: elastic net, support vector machines, random forest, gradient boosting, and artificial neural network (Hastie et al. 2009).[1]

Advanced ML models, such as gradient boosting employed in this paper, could prove useful in efficiently handling a wide variety of payments data and effectively managing collinearity in such data (Yoon 2021; Gogas et al. 2022). This is beneficial because some of the payment streams used here are strongly correlated with each other (Chapman and Desai 2020). Such ML models can also help capture sudden, large, and possibly nonlinear effects of economic crises and the impact of unconventional policies designed to alleviate them (Vrontos et al. 2020; Coulombe et al. 2021; Liu et al. 2022). This is important because different crises have reflected differently in payment streams, suggesting a tangled and possibly nonlinear relationship between some payments streams and macroeconomic targets.[2] Moreover, ML models are beneficial when the emphasis is on improving prediction accuracy—a focus of the present paper (Mullainathan and Spiess 2017; Athey 2017; Yoon 2021).

[1] We use these parametric and non-parametric ML models because they are popular among time series forecasters and preferred in macroeconomic prediction problems (Ahmed et al. 2010; Bok et al. 2018; Athey and Imbens 2019).

[2] In April 2020, the Canadian government began providing social benefits to citizens directly affected by COVID-19. This is reflected by a large increase in payment flows in the government direct deposit stream. Such a policy was not implemented during the GFC, yielding a drop in payment flows in this stream (see Figure 1).

However, the use of ML models leads to many challenges, such as overfitting and interpretability, that could reduce the effectiveness of these models for policy use. The literature is evolving to address such challenges (Athey et al. 2019; Buckmann et al. 2021; Liu et al. 2022; Babii et al. 2021). Likewise, we mitigate the nowcasting ML model problem of overfitting, that is, due to the flexibility of these models, it is easy to overfit them on in-sample data, which could reduce their out-of-sample performance (Bergmeir and Benítez 2012; Bergmeir et al. 2018; Chu and Marron 1991). In addition, we address the difficulty in interpreting these models. Interpretability is important to understand their predictions—especially if they are used to support policy decisions (Varian 2014; Mullainathan and Spiess 2017; Chakraborty and Joseph 2017; Athey and Imbens 2019).

To alleviate the classic ML issue of overfitting, we devise an improved cross-validation strategy tailored to macroeconomic nowcasting models. In cases where the out-of-sample test set has an economic crisis but the validation set[3] does not, the traditional k-fold or leave-p-out cross-validation (Hastie et al. 2009; Bergmeir and Benítez 2012) could be challenging because: (a) the standard k-fold splitting breaks the order (serial correlation) of the series, (b) the distribution of test and validation sets could differ, and (c) the model tuned predominantly on normal periods might not perform well on the out-of-sample crisis period. To overcome this, similar to Kuhn et al. 2013, we use a randomized expanding window approach with k-fold cross-validation but without changing the order of the data (see Figure 3). Since we have the COVID-19 crisis period in the test set, using random sampling helps to include a few samples from the GFC period in the validation set. Consequently, the distribution of validation and test sets are somewhat similar (see Figure 14 in Appendix C), which could assist in selecting a model that performs well in both normal and crisis periods.

Next, we 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 the coalition game theory (Shapley 1953; Osborne and Rubinstein 1994). To utilize this approach, we need to consider each nowcasting exercise as a *game*. Shapley values can then be used to fairly distribute the *payout* (i.e., the model prediction) among the *players* (i.e., the predictors) of the game. SHAP provides a way to interpret ML model predictions at each nowcasting horizon in terms of the marginal contribution of each predictor toward the final prediction. Further, 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.

Similar Shapley-value-based approaches are employed in the recent articles by Buckmann et al. 2021 and Liu et al. 2022 for macroeconomic predictions. They note that although the Shapley value methods for interpretation are based on game theory, they do not provide any optimal statistical criterion, and asymptotics for such approaches are not available yet. To overcome such challenges, for instance, in the recent paper by Babii et al. (2021), the authors propose ML MIDAS and develop the asymptotics in the context of linear regularized regressions. Nonetheless, such analysis cannot be used for nonlinear ML approaches such as gradient boosting and neural networks, which in many cases outperform regularized linear models (Richardson et al. 2020; Liu et al. 2022; Gogas et al. 2022).

[3] The part of the in-sample training set used for ML model parameter tuning and cross-validation (see Figure 3).

Our results suggest that timely retail and large-value payments system data in the ML models—especially nonlinear gradient boosting regression (GBR)—can lower nowcast errors significantly. We obtain a 35–40% reduction in root-mean-square error (RMSE) in nowcasting GDP, retail trade sales (RTS), and wholesale trade sales (WTS)[4] over a linear benchmark model.[5] Further, in the presence of payments data and ML models, compared to the dynamic factor models, can reduce nowcasting RMSE by as much as 20–25%. Out-of-sample performance gain using payments data and ML models is relatively greater (15 to 20%) during the COVID-19 crisis period than the pre-COVID normal economic growth period, and the models using payments data perform better against the latest vintages compared to the real-time vintages. These results suggest that the timeliness of payments data and the ability to capture nonlinear interactions of ML models are primarily helpful.

We also observe improved model performance when the proposed randomized expanding window approach with *k*-fold cross-validation is used for ML model tuning. The average RMSE across *k*-folds is 10–15% smaller using the proposed approach compared to the traditional expanding window *k*-fold cross-validation approach. Further, the Shapely value-based nowcasting model interpretations reveal that, in general, many payment streams are important along with other traditional predictors in nowcasting GDP, RTS, and WTS. Moreover, during the COVID-19 crisis period, the contribution of those payments streams is much higher than the benchmark predictors. Our analysis also suggests that the contribution of payments data in terms of Shapley values is small and linear during periods of low and normal growth. However, during periods of strong negative or positive growth, the payments data contribution is asymmetrical and nonlinear.[6]

In summary, this paper demonstrates that combining timely data, nonlinear methods, tailored cross-validation approaches, and model agnostic interpretability tools can provide policymakers with sophisticated models to accurately estimate key macroeconomic indicators in near real-time, which is important to monitor the economy—especially during the crisis periods such as COVID-19.

We proceed as follows. Section 2 provides a review of related literature. Next, section 3 describes the payments data and discusses the adjustments performed on these data for macroeconomic predictions. Subsequently in section 4, we provides a brief overview of various methods employed for nowcasting and discusses challenges associated with using ML models for predictions. This is followed by a discussion of our results in Section 5. Finally, in Section 6, we set forth our conclusions. Several appendices provide further details on the payments data and the nowcasting methodology employed.

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

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

[6] Contributions in terms of Shapley values from strong negative growth rates in payment streams are much stronger than similar values of positive growth rates (Figure 8 and 9).

2 Literature Review

In the past—driven by the need to overcome dependence on lagged variables—econometricians have used payments data for macroeconomic predictions (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 they record transactions processed in various payment instruments. Thus, they capture a broad range of Canadian consumers, firms, and government economic activities. Also, these data are gathered electronically and hence are immediately available, and they are free of measurement or sampling errors (Galbraith and Tkacz 2007). Such datasets are shown to be useful during economic crisis periods such as the GFC and the COVID-19 shock (Chetty et al. 2020; Bounie et al. 2020; Carvalho et al. 2020; Dahlhaus and Welte 2021).

Traditionally, researchers have used data from a few selected payment instruments for nowcasting (Galbraith and Tkacz 2018; Aprigliano et al. 2019). One issue with this approach is that the use and importance of particular payment instruments may rise or fall for both economic and non-economic reasons.[7] Using data from one or two payment streams in isolation might not capture the full economic picture; therefore, in Chapman and Desai (2020), the authors use most of the stream settled in Canada's retail payment system for macroeconomic nowcasting at the onset of COVID-19. In this paper, however, we also include settlement data from Canada's high-value payments system and cover the wider COVID-19 period.[8] Further, this paper addresses the ML models' interpretability issues and implements an improved cross-validation technique to overcome overfitting challenges. Additionally, this paper compares the performance of the nowcasting models in both normal and crisis periods for the target variables available at both real-time vintages and the latest vintages.

Recently, driven by the need to exploit non-traditional, and large-scale datasets, econometricians have begun using nonlinear ML models for macroeconomic nowcasting (Chakraborty and Joseph 2017; Richardson et al. 2020; Maehashi and Shintani 2020; Chapman and Desai 2020). The cited articles suggest that ML models complement traditional econometric tools and are useful in extracting economic value from non-traditional data sources. Also, they show that in nowcasting, ML models often outperform traditional modeling approaches, such as ordinary least-squares and dynamic-factor models. Similarly, in the recent papers by Buckmann et al. (2021) and Liu et al. (2022), efforts are being made to address interpretability challenges for nonlinear ML models and in Babii et al. (2021) the authors develop the asymptotics in the context of linear regularized ML models to address interpretability. In a similar spirit, our paper adds to the growing literature on the use of nonlinear ML models and non-traditional data and provides an additional set of tools to mitigate overfitting and interpretability challenges in such models to improve their effectiveness for policy use.

[7] In Canada, the proportion of electronic means of payment is 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 Automated Clearing Settlement System (ACSS) increased by 21%, and cash payments declined by 27% in 2019 (Paturi and Chiron 2020).

[8] We use Large-Value Transfer System (LVTS) data, and it is among the top five contributors in nowcasting GDP (see Figure 4). Our out-of-sample testing period covers until December 2020.

3 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 centralized payments systems. The data coming from such systems are potentially useful because they are (a) timely, i.e., available immediately after the end of each period, (b) available at high-frequency, i.e., at the transaction or day levels, (c) precise, i.e., carry no sampling or measurement error, and (d) comprehensive, i.e., capture a broad range of financial activities across the country (Galbraith and Tkacz 2007, 2018; Chapman and Desai 2020; Dahlhaus and Welte 2021).

In Canada, the ACSS and LVTS are used to settle most transactions.[9] Our data consist of all settled transactions in both the 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 CA$29 billion. The ACSS processes 22 payment streams. Broadly, these streams can be categorized into two groups: (1) electronic streams, which include, e.g., automated funds transfer (AFT), point-of-sale (POS) payments, and government direct deposit (GDD), and (2) paper streams, which incorporate encoded paper, paper remittances, and government paper items.

In the ACSS, electronic means of payment have become more common than paper items due to their usability. This change is driven primarily by technological advancements leading to the inception and adoption of new payment instruments. However, economic crises such as the GFC and the COVID-19 shock also influence payment flows. Historically, the encoded paper stream has the highest-value shares in the ACSS, followed by AFT credit. The POS payments stream has the largest volume of shares, followed by the encoded paper stream.[10]

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 approximately 40,000 transactions per business day, with an average daily value of CA$189 billion. The LVTS provides each participant with two options called tranches, T1 and T2, to exchange payments. Each tranche (henceforth also referred to as a stream) differs based on how individual payments are collateralized. Payments in the LVTS comprise foreign exchange payments, payments for settlement of Canadian-dollar-denominated securities, payments related to the final settlement of ACSS and Government of Canada transactions, as well as the Bank of Canada's own and its clients' payments.[11] In the LVTS, payment value and volume are mostly processed through T2.[12]

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

[10] See Chapman and Desai 2020 for the breakdown of shares of payment streams in the ACSS.

[11] See Arjani and McVanel 2006 for further details on payment types settled in the LVTS.

[12] Historically, T2 has processed roughly 75% of the value and 98.7% of the volume of payments, and T1 has processed roughly 25% of the value and 1.3% of the volume.

3.1 Adjustments to Payments Data

In the past, driven by technological advancements, some payment instruments from the ACSS were discontinued or merged into others, and several new payment instruments were created.[13] 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 payment flow, we merged several streams belonging to similar categories and settled related payments.[14] Also, to overcome the effects of consumers' payment choices, i.e., when they switch payment method,[15] we include the sum of all payment instruments in the ACSS "Allstream" as a separate series. This should help develop an overall picture from the ACSS and mitigate the effects of unused streams.

After these adjustments, we are left with seven streams from the ACSS[16] and two streams from the LVTS, which are listed in Table 1 along with a short description. For nowcasting, we use both the monthly gross dollar amount, i.e., *value*, and number of transactions, i.e., *volume*, settled in the payment instruments; this yields a total of 18 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).[17] Note that recursive seasonal adjustments are performed in real time using the data available up to the nowcasting horizon at each time step. 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.[18]

Our dataset does not include some payment instruments not settled through the ACSS or LVTS, such as credit card and e-transfer payments.[19] However, Galbraith and Tkacz (2018) conclude that credit card payment data for Canada do not add significant value in nowcasting GDP and retail sales.[20] Further, our dataset does not include *on-us* transactions where both sender and receiver have an account with the same financial institution; such transactions do not need to be settled in a payment system. However, their shares are small and may not materially influence our analysis.[21]

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

[14] See Table 1 footnotes for specifics on each adjustment performed.

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

[16] The seven ACSS streams comprise transactions settled in all ACSS payment instruments.

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

[18] Using growth rates (instead of levels) helps to induce (approximate) stationarity in both the target and predictors.

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

[20] Note that in Galbraith and Tkacz (2018), the authors use a short sample size in their analysis of credit card data. The results could differ for a larger sample size.

[21] On-us payments amount to roughly 20% more than those settled in the ACSS. The value of on-us transactions differs by stream, for instance, in encoded paper, it is about 25%, and in POS payments, it is 16% (Paturi and Chiron 2020).

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

ID	Stream	Short Description
C	AFT credit[b]	Government direct deposit (GDD): payrolls and account transfers
D	AFT debit	Pre-authorized debit (PAD): automated bill and mortgage payments
E	Encoded paper[c]	Paper bills of exchange: cheques, bank drafts, and paper PAD
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]	The sum of all payment streams settled in the ACSS
T1	LVTS-T1[g]	Time critical payments and payments to the Bank of Canada
T2	LVTS-T2[h]	Security settlement, foreign exchange, and other obligations

[a] The first six payment streams are representative of 20 payment instruments processed separately in the ACSS. There are a few additional payment instruments. However, they are not available for the entire period considered in this paper. Therefore, they are excluded from this study. The excluded streams are ICP regional image payments and ICP regional image payments return. Note: Excluded streams collectively account for only 0.001% of the total value settled in the system. For further details on individual ACSS streams, see Appendix A.

[b] Stream C is the sum of AFT credit and Government direct deposit streams. We combine them because, starting in April 2012, Government direct deposit was separated from the AFT credit stream and processed independently.

[c] Stream E is the sum of multiple streams settled separately in the ACSS. It combines encoded paper (E), large-value encoded paper (L), image captured payments (O), Canada Savings Bonds (B), Receiver General warrants (G), and Treasury bills and bonds (H). It subtracts image-captured returns (S), unqualified (U), and computer rejects (Z) streams. We combine all of them because, over time, many of these streams were separated from the encoded paper stream and process similar types of payments.

[d] The 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 refunds (Q) streams.

[e] Stream X is the sum of paper remittances (F), EDI payments (X), and EDI remittances (Y). This stream is composed of all corporate-to-corporate payments and corporate bill payments and remittances.

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

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

[h] The LVTS processes payment values equivalent to the annual GDP every five days, and the majority of the value and volume settled in the LVTS is processed in stream T2.

3.2 Payments Data for Macroeconomic Nowcasting

The crux of the nowcasting problem is that most official estimates of macro indicators are released with a substantial delay. For instance, in Canada, GDP, RTS, and WTS are released with a delay of six to eight weeks. In addition, they undergo multiple revisions, sometimes years later, highlighting the uncertainty of their measurement. Moreover, during a rapid crisis such as COVID-19, macroeconomic predictions are difficult because of the large and unprecedented economic impact. This can undermine the use of lagged data for nowcasting. Therefore, it is valuable to use more timely available information, in this case, payments systems data.

Payments data capture numerous types of transactions from both sides of macroeconomic accounts. For example, consumer income and expenditure, business-to-business payments, and Government of Canada spending. This variety, timeliness, and lack of sampling and measurement error in the payments dataset make it a rich economic information source.

For nowcasting exercises, we use Canada's monthly GDP, RTS, and WTS at the latest available vintages (i.e., after revisions) and real-time vintages (i.e., first release) as target variables.[22] We select these indicators because GDP is crucial for policymakers, and since we are using payments data, we think payments data have value in predicting RTS and WTS. All these indicators are released in Canada with a substantial lag and are available monthly for all historical releases. This variation allows us to test the robustness of our models.

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 payments data during a crisis, we highlight the growth rates of all variables during the 2008 GFC period (in gray) and the COVID-19 period (in blue).

During the GFC period, the decline and rebound in these payment streams' growth rates go hand-in-hand with macroeconomic indicators. Similarly, during the COVID-19 shock, we observe a sudden drop in most payment streams and in the macro variables. For instance, GDP and encoded paper, RTS and POS payments, along with WTS and corporate payments show similar movement during both crisis periods. This is a good indication of the economic value associated with these payment streams during such times.

During the COVID-19 period, however, we observe a complicated relationship between the macro indicators and some payment streams.[23] For instance, the value of payments through the AFT credit stream (which includes GDD payments) did not drop significantly at the onset of the COVID-19 shock. On the contrary, starting in April 2020, the value of payments processed through the AFT

[22] Latest vintages of seasonally adjusted monthly GDP, RTS, and WTS are obtained from Statistics Canada Tables 36-10-0434-01, 20-10-0008-01, and 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, and 20-10-0019-01, respectively.

[23] This may be the result of the difference in the nature of the two crises. In 2008 it was a severe worldwide economic crisis; in 2019, however, there was a rebalancing of portfolios. Therefore, the nature of the 2008 GFC was consistent with a drop in payment flows, while the 2019 COVID-19 crisis led to increased flows in several payment streams.

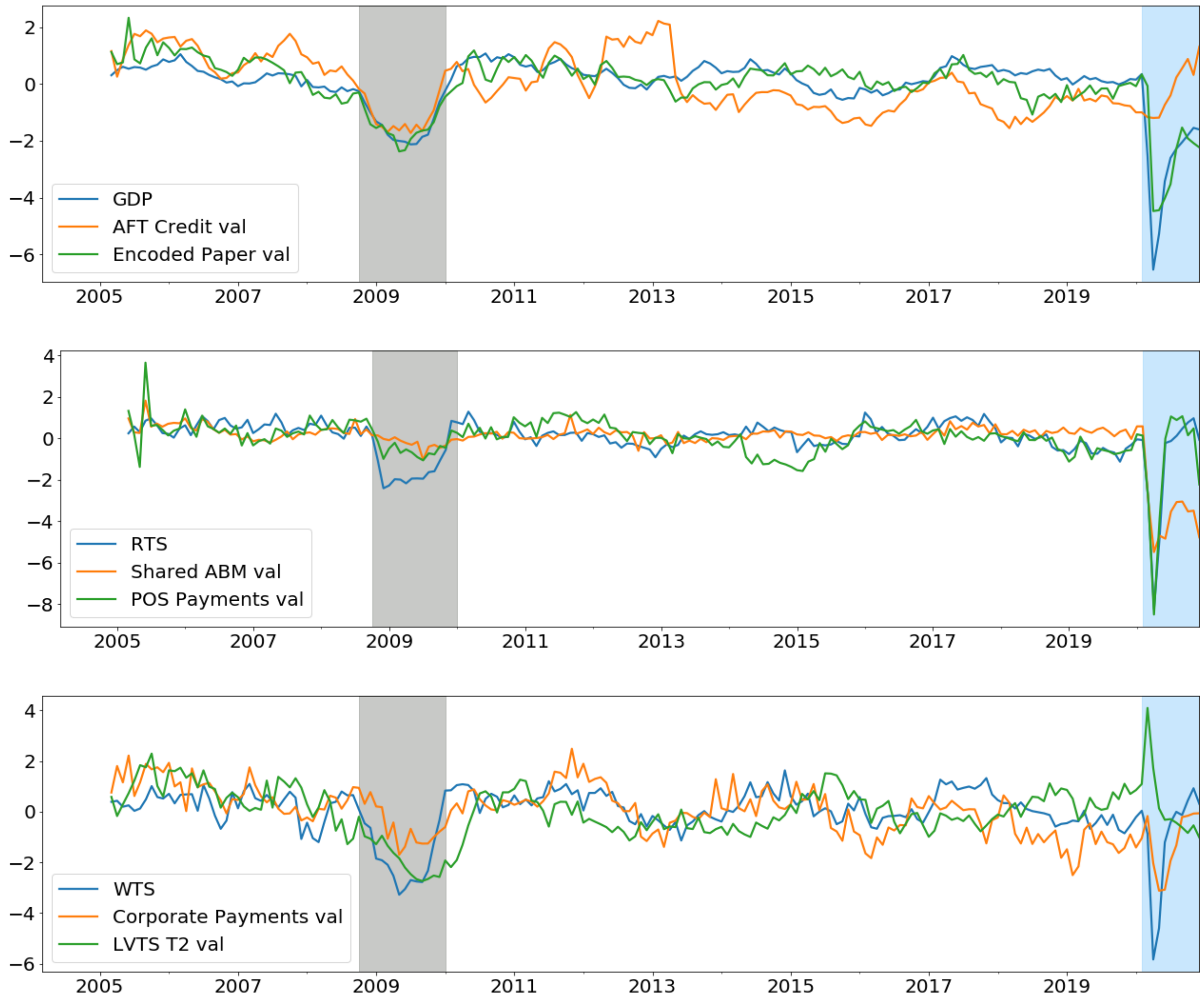

Figure 1: Standardized YOY growth rate comparisons of GDP, RTS, and WTS, with selected payment streams. Gray highlighting–GFC period; blue highlighting–COVID-19 period. Note: AFT credit includes Government direct deposit, encoded paper is the sum of multiple streams settled separately in the ACSS, POS payments include online payments, and corporate payments is the sum of paper remittances, EDI payments, and EDI remittances.

credit stream increased due to the 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 to that of macro indicators during the same period. Such behavior is not seen during the GFC period, where both WTS and LVTS-T2 growth rates drop (Figure 1, bottom).[24] Such complex behavior could be challenging to capture using traditional linear models.

4 Methodology

This section briefly describes the nowcasting models employed. First, we discuss ordinary least squares (OLS) and the dynamic factor model (DFM). This is followed by a brief discussion of the ML models.

Consider a set $X = \{\mathbf{x}^1, \mathbf{x}^2, \dots, \mathbf{x}^M\}$ of M predictors (also called features or independent variables) and a target $\mathbf{y}$ (dependent variable), each with N data (sample) points. For example, predictors could be monthly aggregated values settled in each payments stream, and the target could be monthly GDP (both recorded at the end of each month). 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 estimated, for example, using payments data and 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 $\mathbf{y}$ and the predicted $\hat{\mathbf{y}}$ target variables. Such linear models have proven to be valuable and straightforward for prediction, and they are commonly used due to their simplicity and interpretability. However OLS can model only linear relationships in the parameters $\mathbf{w}$. Although the linearity assumption makes them easy to interpret on a modular level, it generally does not perform well on wide, large, and complex datasets (Hastie et al. 2009). Moreover, multicollinearity in the predictors, although does not reduce the predictive power, could lead to reduced precision of the estimated coefficients.

The DFMs are a powerful approach to capturing the common dynamics of a set of predictors in a relatively small number of latent factors. This can act as a dimension-reduction technique by estimating a small set of dynamic factors from a large set of observed variables. DFM is a frequently selected model for macroeconomic nowcasting and forecasting (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 effectively handles a large number of predictors. The basic representation of

[24] Similar behavior is observed in LVTS-T1, where payment value rose dramatically at the onset of COVID-19 due to extraordinary measures taken by the Bank of Canada under its quantitative easing policy (Bank of Canada 2020).

the model is

$$X_t = \Lambda f_t + \varepsilon_t \tag{2}$$

$$f_t = A_1 f_{t-1} + \cdots + A_p f_{t-p} + u_t, \tag{3}$$

where X_t is a set of predictors at time t, f_t is the unobserved factor at t, Λ is the vector of factor loadings, ε_t is the idiosyncratic disturbance at t, A_i are matrices of autoregression coefficients, and u_t is the factor disturbance at t. DFMs are successfully applied for economic monitoring and predictions around the world (Bańbura et al. 2010; Stock and Watson 2016; Hindrayanto et al. 2016; Bragoli 2017) including nowcasting Canada's GDP (Chernis and Sekkel 2017). Therefore, we employ this model to nowcast various Canadian macro indicators using payments data.

4.1 Machine Learning Models for Nowcasting

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

We use some of the recently popularized parametric and non-parametric ML approaches, such as elastic net (Zou and Hastie 2005), support vector machines (Smola and Schölkopf 2004), random forest (Breiman 2001), gradient boosting (Friedman 2001), and feedforward artificial neural networks (Bengio 2009). For each model, there are many variations proposed in the literature. However, we focus on the simpler version of each model to test their feasibility for macroeconomic nowcasting in Canada. In the remainder of this section, we provide a high-level description of these models. For further details on them, refer to Appendix B and various textbooks (Friedman et al. 2001; Buckmann et al. 2021; Hastie et al. 2009; Bengio 2009).

The elastic net (ENT) is a regularized linear regression model. Here, the objective is similar to that of the OLS with the addition of L_1 and L_2 penalties depending on how large the sum of the parameters $\mathbf{w}$ can become.[25] In an ENT 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 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 main objective is to determine a decision boundary at a distance from the support vectors such that the data points closest to the hyperplane are within that boundary line. This gives us the flexibility to define how much error is acceptable in our model (Burges 1998; Smola and Schölkopf 2004).

[25] 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).

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

Similar to RFR, gradient boosting regression (GBR) is a tree-based, non-parametric ensemble learning approach. However, unlike RFR, GBR is based on a boosting in which a sequence of weak learners (decision trees) are built upon a repeatedly modified version of the training dataset. In this approach, the base learners are sequentially improved by repeatedly applying the same base learner with the target's residuals as the outcome of interest (Friedman 2001).

The feedforward artificial neural network (ANN) with hidden layers consists of multiple layers of artificial neurons sandwiched between input and output layers. In this approach, 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 obtain 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 good performance using ANN (Bengio 2009; Goodfellow et al. 2016).

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

4.2 Machine Learning Model Cross-Validation

Overfitting is commonly attributed to the use of ML models—especially nonlinear models. ML models have many parameters that can be optimized to improve prediction accuracy (commonly known as hyperparameter tuning). Therefore, it is straightforward to tune the model to perform well on a specific dataset, for example, an in-sample training set. However, such models generally fail to perform well when applied to unseen data (Hastie et al. 2009).

This problem can be alleviated using k-fold cross-validation techniques. 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 (Hastie et al. 2009). Such procedure effectively avoids overfitting. However, random splitting of the training sample breaks the order of series (autocorrelation) and could lead to the use of future data points for past predictions, giving an unfair advantage to the model. For these reasons, it may not practical to use it in the *same way* in nowcasting models (Bergmeir and Benítez 2012).

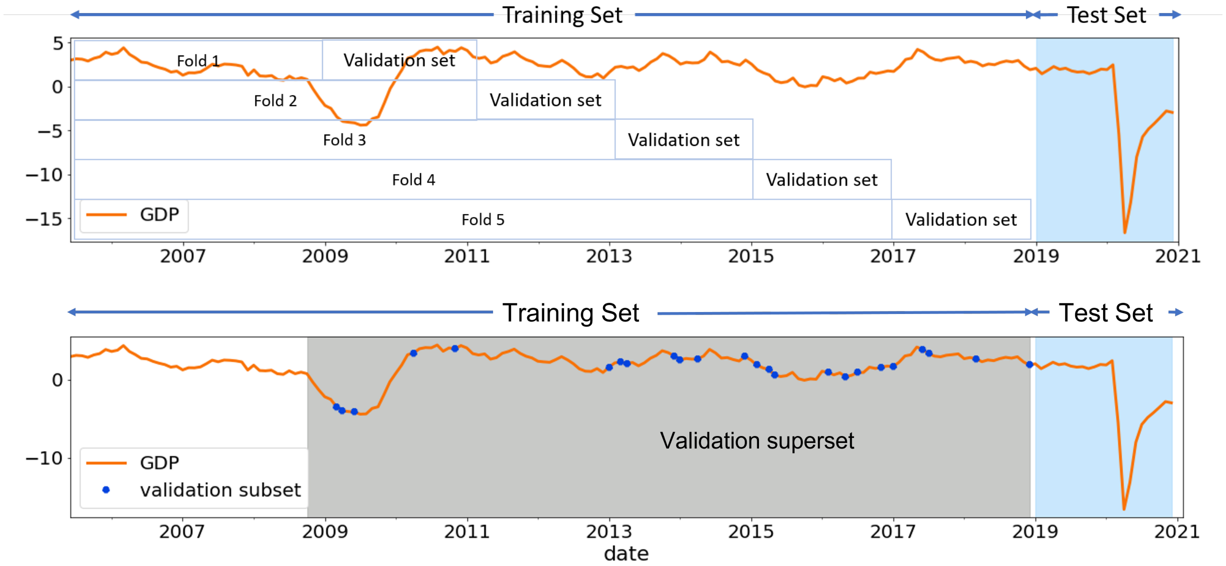

Figure 2: (Top) Schematic of standard expanding window approach for cross-validation in time series. The dataset is divided into a training set with validation subsets and a test set (highlighted in blue). (Bottom) Schematic of the proposed randomized expanding window approach showing a typical validation subsets (represented by ●) randomly sampled from the validation superset (highlighted in gray). In both plots, the orange line shows the GDP growth rate.

This challenge can be mitigated using an expanding window approach for cross-validation (Figure 2, top). Here, the end portion of the training set, often called a validation set, is kept aside for model tuning and cross-validation.[26] This approach is useful for nowcasting during normal economic periods. However, in cases where the test sample includes an economic crisis but the validation sample does not, traditional expanding window cross-validation could be challenging because (a) the distribution of test and validation samples are quite different and (b) the model is predominantly tuned for normal periods and therefore may not perform well for out-of-sample crisis periods.

We implement a slightly altered version of the expanding window approach tailored to macroeconomic nowcasting models (Figure 2, bottom). We randomly sample n-points (one for each iteration) from the validation superset (highlighted in gray), beginning just before the GFC and continuing to the end of the training set, and use them as a validation sample. For each iteration of expanding window validation, only data points that come before the chosen point are used for training—preserving the order of data and temporal dependency between observations (see Figure 3).

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 GFC in each validation subset. Therefore, the proposed approach helps make the distribution of validation and test sets somewhat similar (see Figure 14 in Appendix C) and assists in selecting a model that can perform well for both

[26] For each iteration of the expanding window, the training sample is increased by one period and the model prediction is performed on the next period from the validation sets (Bergmeir and Benítez 2012). Consequently, the model parameters can be chosen based on model performance on the validation sets. See Appendix C for additional details.

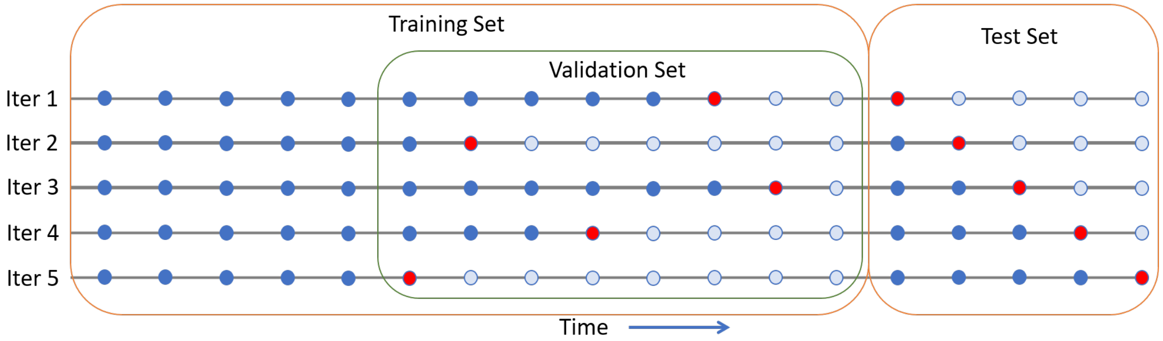

Figure 3: Schematic of expanding window approach for a typical fold in *k*-folds cross-validation and out-of-sample prediction. The available data are divided into training, validation, and test sets. For the given iterations of the expanding window (Iter), 🔵 represents in-sample training points and 🔴 represents out-of-sample test points (for the fold). For each iteration in this fold of cross-validation, we use randomly sampled 🔴 points from the validation superset as the validation subset. Note: the out-of-sample size (the number of 🔴 points) in each validation subset is kept similar to the test set. For instance, both the validation subset and test set have five out-of-sample points each in this schematic.

normal and crisis periods. Also, our approach removes the restriction on the number of validation sets we can sample that could be binding in traditional cross-validation approaches.[27]

Further, instead of using all payment streams in each model or manually selecting a few payment streams for the given macro indicator, we use a *data-driven* approach for feature selection. We treat the number of payment steams p as similar to a model parameter and use the expanding window cross-validation approach to optimally select the best p streams for each target variable based on their performance on the training and validation sets.[28]

4.3 Machine Learning Model Interpretability

Another problem commonly attributed to the use of ML model is the loss of interpretability due to their complex nature. However, interpretability is essential for many applications—including macroeconomic prediction (Mullainathan and Spiess 2017; Athey and Imbens 2019; Buckmann et al. 2021).

Some ML models employed here, such as ENT, SVR, and the simple tree-based learning models, are inherently interpretable, but only to a certain extent (Zou and Hastie 2005; Breiman 2001). Tree-based ensemble approaches such as GBR and RFR can also be interpreted—to some extent—using impurity or permutation-based feature importance methods (Breiman 2001; Molnar 2020). However, they are mostly used for global (entire sample) interpretations. Also, each of those methods has different interpretability approaches, making them difficult to compare. To address these challenges, we use the Shapley value-based model agnostic approach SHAP developed in Lundberg and Lee (2017). The SHAP can be used for both the local (each prediction instance) and global interpretations. Moreover,

[27] In our approach, due to random sampling, some observations may be selected more than once in the validation subsets and some may never be selected. This could lead to overfitting if too many validation sets are sampled.

[28] Further details on cross-validation and model selection are discussed in Appendix D.

the SHAP can be used for dependence plots which could be valuable to get further insights.

In SHAP, the Shapley value method from coalitional game theory (Shapley 1953; Osborne and Rubinstein 1994) is used to fairly distribute the "payout" (= the prediction) among the "players" (= the predictors) (Lundberg et al. 2020). In nowcasting, SHAP can be used to fairly distribute the ML model prediction among the set of predictors X_t at each time horizon t for local model interpretation. Further, 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 or testing datasets.

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

SHAP approaches are reliable because they are developed based on the Shapley value, which has game-theoretical foundations. However, the time required to estimate Shapley values could increase exponentially with the number of predictors. This is not a significant concern for our application because we have comparatively fewer predictors and smaller sample sizes. The KernelSHAP method also suffers from collinearity in the predictors, which could represent a concern for our work, given that several predictors are correlated. These problems can be mitigated to an extent using TreeSHAP, but only for tree-based models. Another challenge with SHAP approaches is that it is possible to create intentionally misleading interpretations to hide the bias. Also, in some cases, the outcomes can easily misinterpret and lead to ambiguous conclusions. Therefore, SHAP should be used with caution (Alvarez-Melis and Jaakkola 2018; Slack et al. 2020; Molnar 2020).[29] Further details on Shapley values and SHAP with a test example are given in Appendix E.

4.4 Case Specifications and Model Training

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 CPI, unemployment, CFSI,[30] and CBCI.[31] CFSI and CBCI are available immediately after period-end and carry comprehensive and useful information about the macro indicators. Along with CPI and UNE (available with a one-to-two week delay), these predictors form a strong benchmark to assess information gain using payments systems data.

[29] Note that the Shapley value-based interpretation approach does not provide causal inference or any optimal statistical criterion. They only explain the marginal contribution of each feature to the difference between the actual prediction and the mean prediction given the set of predictors. The Shapley value-based approaches could also be used to select the best subset of predictors. However, in the context of model interpretation, the focus is only on computing the marginal contribution of *all predictors* involved in the prediction exercise (Lundberg and Lee 2017; Molnar 2020).

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

[31] CBCI is based on the Conference Board's survey of Canadian households, which provides a measure of consumer 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 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 above in Section 4. Finally, we compare the performance of the ML models with the benchmark case and DFM. We use RMSE as the key performance indicator for out-of-sample model evaluation.

For all 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 advance in time, we include new predictors when they become available. For example, GDP nowcasting at time horizon t, that is, on the first day of the month of interest, we use the latest available benchmark variables and the monthly aggregated payments data available at $t-1$. Model $\mathscr{F}$ can be specified as[32]

$$\widehat{GDP}_t = \mathscr{F}(GDP_{t-3},\ CPI_{t-2},\ UNE_{t-2},\ CFSI_{t-1},\ CBCC_{t-1},\ Payments_{t-1}). \tag{4}$$

Similarly, at the next nowcasting horizons $t+1$ and $t+2$, using the latest available predictors, the models can be specified as[33]

$$\widehat{GDP}_{t+1} = \mathscr{F}(GDP_{t-2},\ CPI_{t-1},\ UNE_{t-1},\ CFSI_t,\ CBCC_t,\ Payments_t), \tag{5}$$

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

We train the nowcasting models using the expanding window approach as schematically outlined in 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. The ML models, which require extensive hyperparameter tuning and cross-validation, are trained using the following procedure:

1. From the training sample, we select two dates covering the wider range of training data as a validation superset (Figure 2) and randomly choose a set of n sample points as a validation set (where $n = 24$ points, it is the same size as the test sample).[34]

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

[32] GDP is released with a two-month lag, CPI and UNE are released with a one-to-two week lag, and CFSI, CBCC, and payments data are available the day after the end of the period.

[33] At the $t+2$ nowcasting horizon (on the first day of the month in which the target month's macro indicators are released), we have $t+1$ months of payments data. However, we do not include this because we are interested mainly in assessing the usefulness of t month's payment data to predict t month's macro variables. Also, note that the left-hand sides in Equations 4, 5, and 6 represent the same month's target but they are estimated at different time horizons.

[34] We choose a start date just before the GFC period and an end date just before the test set, then select n random data points between these two dates as a validation set. This helps to include a few data points from the crisis period in each fold of the validation subset, at the same time avoiding use of a large cross-validation sample.

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

4. Steps 2 and 3 are repeated k times for the same set of hyperparameters but with a different validation subset randomly sampled from the validation superset ($k = 5$ fold cross-validation).

5. Next, we select the best set of model parameters, i.e., the parameters with the lowest average validation RMSE (averaged over k folds) as the final model.

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

5 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),[35] and the out-of-sample testing period is Jan 2019 to Dec 2020 ($N = 24$). Our training set includes the 2008 GFC period, and the test set combines a normal economic growth period (Jan 2019 to Feb 2020) and part of the ongoing COVID-19 crisis period (Mar 2020 to Dec 2020). This allows us to examine model performance during both normal and crisis periods.

YOY GDP, RTS, and WTS growth rate nowcasting performance for the various cases outlined in Section 4.4 are discussed below. Table 2 compares nowcasting performance—in terms of out-of-sample RMSE—of the DFM and ML model (gradient boosting regression[36]) on the main case against the benchmark models at the time horizons t, $t+1$, and $t+2$.

Our results suggest that the payments systems data in conjunction with ML models can provide notable reductions in nowcasting RMSEs for all three macro variables considered. Specifically, we observe a 35–40% reduction in RMSE over the benchmark case in nowcasting GDP, RTS, and WTS at time horizon $t+1$, i.e., when we use same month's payments data as the target variable. The main case predictions at this time horizon are statistically significant for the Diebold-Mariano test using the benchmark.[37]

Comparatively, the information gain using payments data is smaller at nowcasting horizon t, that is, when we use the first lag of payments data, and $t+2$, that is, when the values of all benchmark indicators are available at t along with the first lag of the target variables. In these cases, we obtain a 7–25% reduction in RMSE over the benchmark in nowcasting GDP, RTS, and WTS. These results suggest that payments data provide the greatest nowcasting value when the given month's payments data are used immediately to predict the same month's macro variables (at $t+1$ time horizon, i.e., on the first day of the next month).

[35] We lose the first full year of data after computing YOY growth rates.

[36] Gradient boosting regression model consistently performed better across different target variables and time-horizons compared to the other models (see Table 3 in Appendix B).

[37] We recognize that the Diebold-Mariano test has poor finite-sample properties. However, we use it to be comparable with similar papers where it is used, e.g., Chernis and Sekkel (2017) and Aprigliano et al. (2019).

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^{*}	36
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: 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 Appendix H.

[c] As a benchmark, we use OLS with CPI, UNE, CFSI, CBCC, and the first available lagged target variable (i.e., the 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. Idiosyncratic components are assumed to follow an AR(1) process. Note: including additional factors does not improve model performance.

[e] We use GBR because it consistently performs better than other ML models (see Table 3 in Appendix B). We select model parameters using the cross-validation procedure outlined in Appendix C and D. For example, the selected model for GDP nowcasting at $t+1$: *learning_rate* is 0.1, *max_depth* is 1, and *n_estimators* is 1000 (see Appendix B for further details).

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

$^{*}, ^{**}, ^{***}$ denote statistical significance at the 10, 5, and 1% levels, respectively, for the Diebold-Mariano test using the benchmark.

Next, we compare ML models against DFM (see Table 2, and 3 in Appendix B). Overall, DFM contributes to increasing prediction accuracy up to 25% at $t+1$.[38] However, in nowcasting GDP, RTS, and WTS at all three time horizons, the GBR, ENT, and feedforward ANN models—in many cases—perform better than DFM and other ML models considered. 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 observe up to a 12-30% reduction in RMSE over DFM with the payments data.

Incorporating payments data in ML models provides downturn and recovery indications (during crisis periods) 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 model flexibility, allowing this data to provide better predictions during crisis periods. Visual comparisons of the best performing ML model against the benchmark model for in-sample and out-of-sample predictions are depicted in Figure 18 in Appendix F.

Next, we separately test our model's out-of-sample performance during a normal economic period (Jan 2019 to Feb 2020) and the COVID-19 period (Mar 2020 to Oct 2020) of the test sample (see Table 4 in Appendix G). 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 (15–25% reduction in RMSE) using payments data. These results show that the payments data are useful during normal periods and its significance surges during periods of crisis, which substantially improves our model performance during those periods. Therefore, these results suggest that the timeliness of payments data and the ability to capture nonlinear interactions of ML models is primarily helpful.

Finally, we compare the GDP nowcasting performance of our model with the real-time vintages (first releases) and the latest vintages (see Table 5 in Appendix H). Comparatively, the models using payments data perform better against the latest vintages. This makes sense, given that the latest vintages are more accurate (due to multiple revisions) than the real-time ones. Therefore, these results further emphasize the importance of timeliness of payments data for effectively provisioning accurate estimates of key macro indicators in near real-time, which is important to monitor the economy—especially during the crisis periods such as COVID-19.

5.1 Model Interpretation and Payments Data Contribution

We now discuss the Shapley value-based interpretation of ML model predictions using the SHAP. We primarily focus on nowcasting GDP at time horizon $t+1$ using the tuned GBR model. However, we discuss a few key results for nowcasting RTS and WTS using a GBR, at the end of this section.

For demonstration, we use the entire sample (Mar 2005 to Dec 2020) for training. In Figure 4, we plot the 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

[38] In this case, we use the DFM model with two factors. Including more factors does not improve results. Note: the DFM model's performance, in some cases, is similar to the OLS model.

high to low). For example, in the case of in-sample training data, both encoded paper value and GDP lag have a strong influence. However, the encoded paper stream is the strongest predictor (on average, it changes the GDP growth rate by 0.6 points). This is followed by the unemployment lag feature, LVTS-T2 value, and the sum of all the ACSS streams (Allstreams value).[39]

In Figure 5, we show the global feature importance plot for the COVID-19 period with high negative growth rates (Mar to Dec 2020). During this crisis period, the influence of the encoded paper stream increases substantially along with the Allstream, AFT debit, and POS payments. GDP lag, the second most important feature for the entire training sample, loses its prediction power during the COVID-19 crisis period. A similar contribution from several payment streams is observed during the GFC period. These results suggest that, although lagged macro indicators influence GDP growth rates during normal periods, they do not add much value during crisis periods due to the delayed signal. During such periods, payments data become much more valuable and contribute well to macroeconomic prediction.

Next, using the SHAP "force" plots, we compute local feature importance, that is, the usefulness of each feature for a chosen 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 advance 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 6, we plot the Shapley values as forces for Feb (top), Mar (middle), and Apr (bottom) 2020. Here, each Shapley value is an arrow that forces an increase (higher in red) or decrease (lower in blue) in the prediction from the baseline (the average of all predictions). The length 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 $f(x)$. For Feb 2020, just before the pandemic began affecting Canada's economy, most payment predictors are positive (red) and pushing GDP growth higher. However, for Mar 2020 (the first month of the COVID-19 shock), most payment streams show a negative signal (blue). Similarly, for Apr 2020, all predictors show strong negative contributions, pushing the model prediction to the lowest value.

Figure 7 shows the force plots for each instance in the entire training sample rotated and stacked together vertically. We observe red clusters of predictors with positive values (most predictors contributing positively to GDP prediction) during positive economic growth periods and blue clusters with negative signals during crisis periods, such as the GFC and COVID-19 shock. Such clustered signals could prove valuable in tracking crises in real time.

In Figure 8, we show 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. Observe that the small and negative values of encoded paper growth

[39] For the tree-based models like GBR, we can also use impurity or permutation-based global feature importance (Breiman 2001). In our case, the permutation approach gives similar results as SHAP for the top three major predictors and matches eight out of the top ten highest contributors but slightly in a different order. Moreover, all three approaches rank the same three predictors in the top five list, and the Encode paper stream remains the most prominent feature in all approaches (see Figure 17).

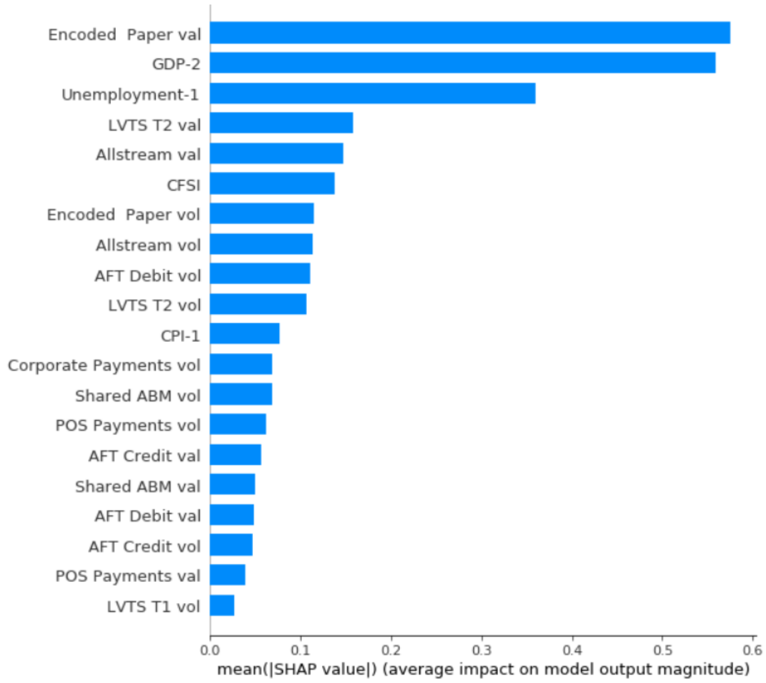

Figure 4: GDP: SHAP global feature importance measured as mean absolute Shapley values for each instance in the entire training sample (Mar 2005 to Dec 2020). The top 20 features are ranked from high (top) to low (bottom) based on average Shapley values.

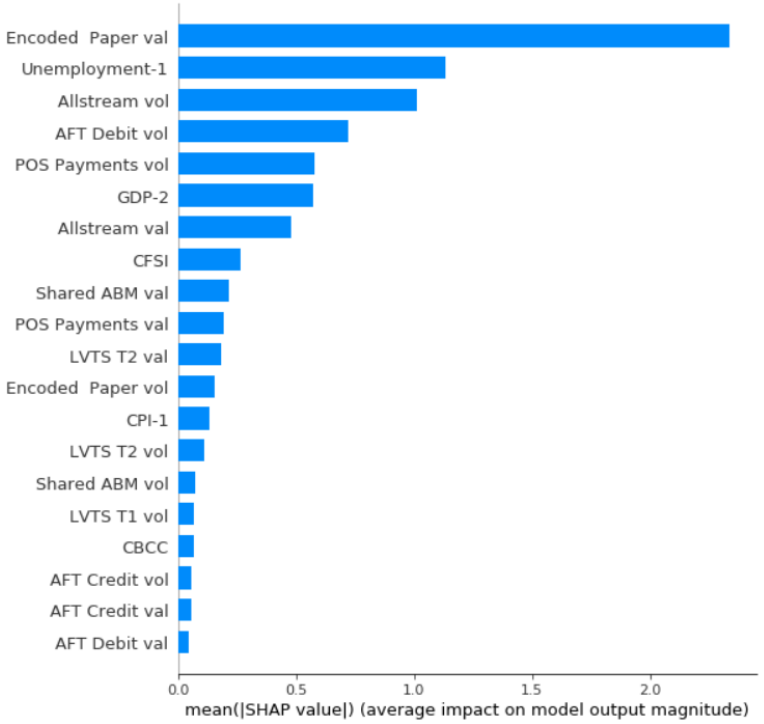

Figure 5: GDP: SHAP global feature importance measured as mean absolute Shapley values of each instance in the training sample for the COVID-19 period (Mar 2020 to Dec 2020). The features are ranked from high (top) to low (bottom) based on average Shapley values.

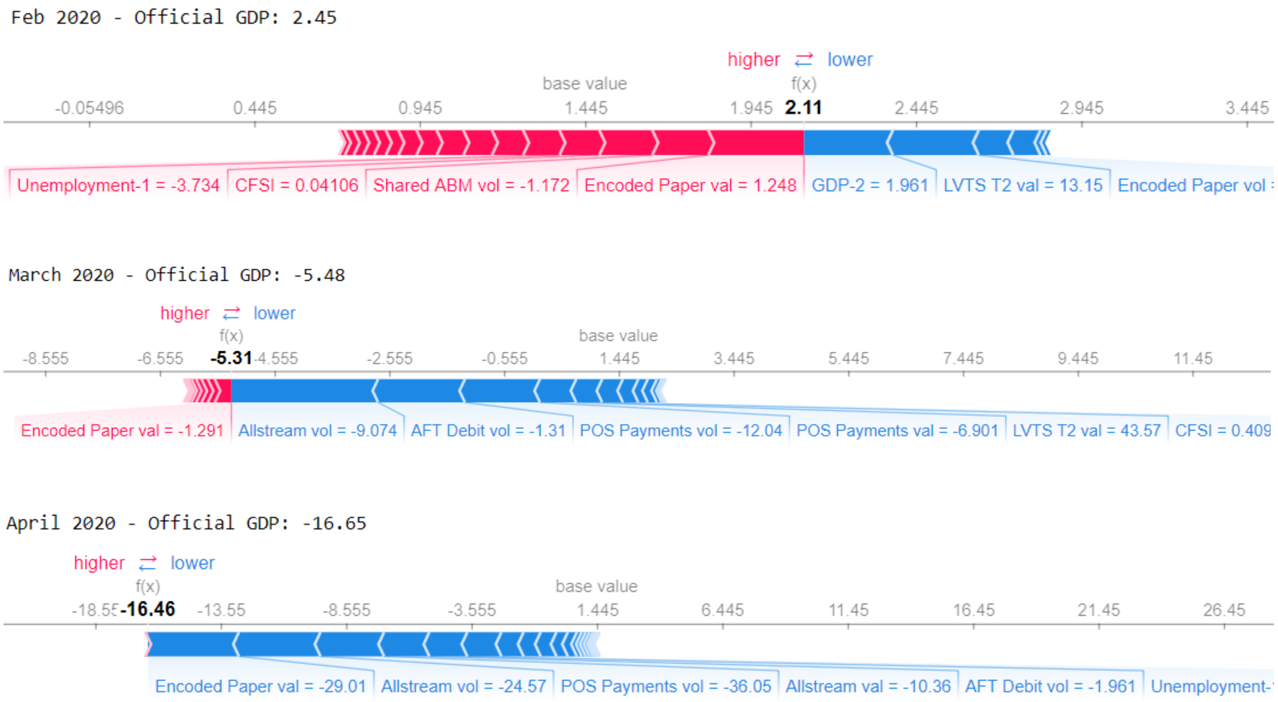

Figure 6: GDP: SHAP force plots showing the feature contribution at each nowcasting instance during the onset of the pandemic, i.e., for Feb 2020 (top), Mar 2020 (middle), and Apr 2020 (bottom). The red arrows are positive Shapley values (contributing positively to GDP growth), and the blue arrows are negative Shapley values (contributing negatively to GDP growth). $f(x)$ is the model prediction at that instance, and the base value is the average of all predictions. Note: Values in red and blue are respective predictor values during that month; e.g., the Encode Paper value in Feb 2020 is 1.248.

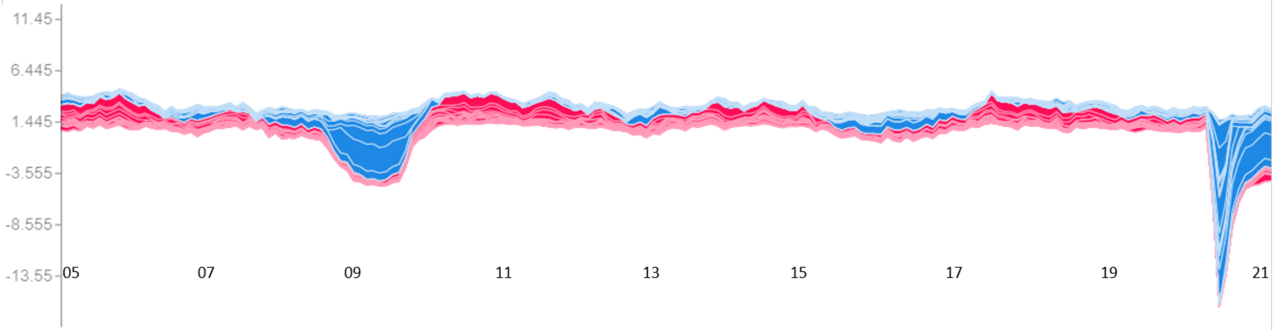

Figure 7: GDP: Clustered force plots for each instance in the training sample, i.e., monthly instance from Mar 2005 to Dec 2020 positioned on the x-axis. Red clusters are positive Shapley values, i.e., the highest number of predictors contributing positively to GDP prediction, therefore pushing the prediction up, and blue clusters are negative Shapley values, i.e., the most predictors contributing negatively to GDP prediction, therefore bringing the prediction down (during the GFC and COVID-19 period). The line at the intersection of blue and red clusters is the actual model prediction.

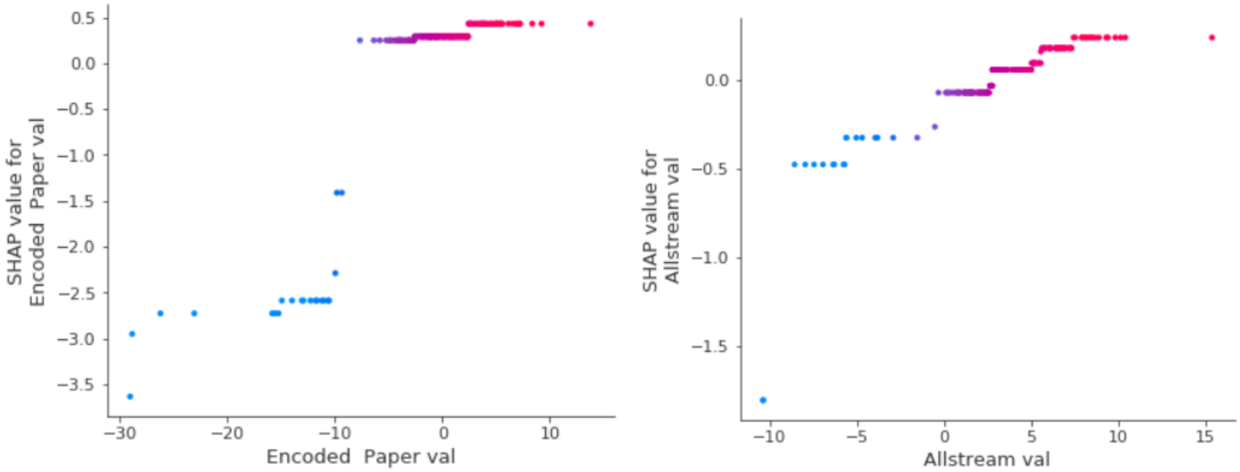

Figure 8: Dependence plots show the Shapley value for each instance in the sample and the corresponding feature value. On the left, we show the dependence plot for the encoded paper (E) value, and on the right, we show the dependence plot for the ACSS Allstream (All) value.

rates provide higher contributions in Shapley values compared to the positive growth rates. However, both positive and negative growth rates of Allstreams value contribute similarly (or symmetrically). The encoded paper plot (left) suggests that the contribution of this stream, in terms of Shapley values, is small and linear during periods of normal growth. However, during periods of strong negative or positive growth ($>= |10|$), the contribution of this stream is asymmetrical and nonlinear.

Similar behavior is observed in nowcasting models for RTS and WTS using payments data and GBR model. In Figure 9, we show the dependence plots of RTS with POS payments value (left) and WTS with Allstream value (right). We also show how these payment streams are influenced by CFSI. These plots suggest that at high-stress levels, that is, at high values of CFSI (shown in red) and negative values of payment growth rates, the signal from these payment streams is strong and their contribution is high. However, for low levels of stress (blue) and high positive payment growth rates, the contributions from these streams are positive but small. This confirms the asymmetrical and potentially nonlinear relationship between these streams and the corresponding macro variables.

Finally, in Figure 10, we plot the SHAP global feature importance for the entire training set at time horizon $t+1$ for RTS (left) and WTS (right), respectively. These plots suggest, in the case of RTS, that the Allstream and POS payments values highly influence the model prediction. This makes sense, given that POS payments are commonly used for retail trade. In the case of WTS, the Allstream value has its strongest impact on the model prediction along with the encoded paper and corporate payments value streams, highlighting the importance of these streams in predicting WTS. These plots suggest that, in general, aggregated payments streams (Allstream and encoded paper) are crucial predictors in nowcasting GDP, RTS, and WTS.

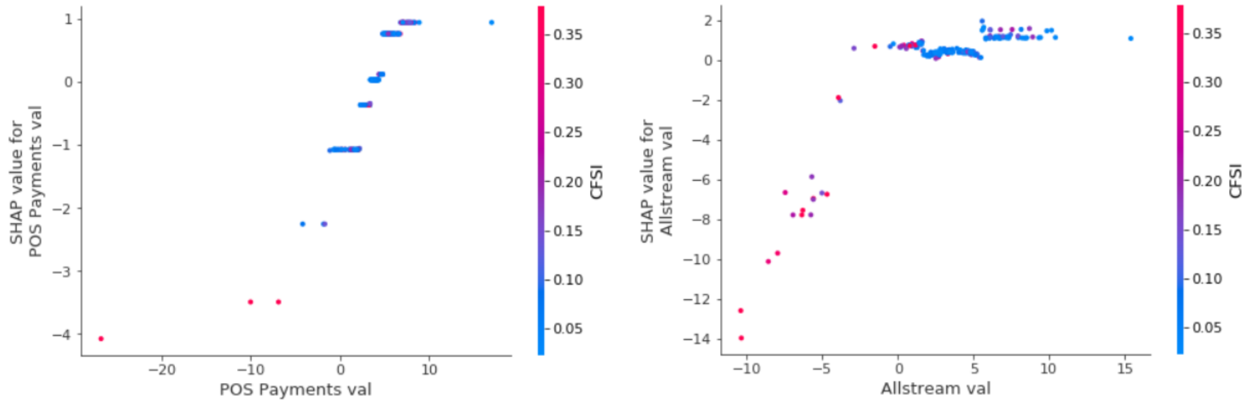

Figure 9: Dependence plots show the Shapley value for each instance in the sample and corresponding predictor value. On the left we show a dependence plot of RTS for the POS payments value, and on the right we show the dependence plot of WTS for the ACSS Allstream value.

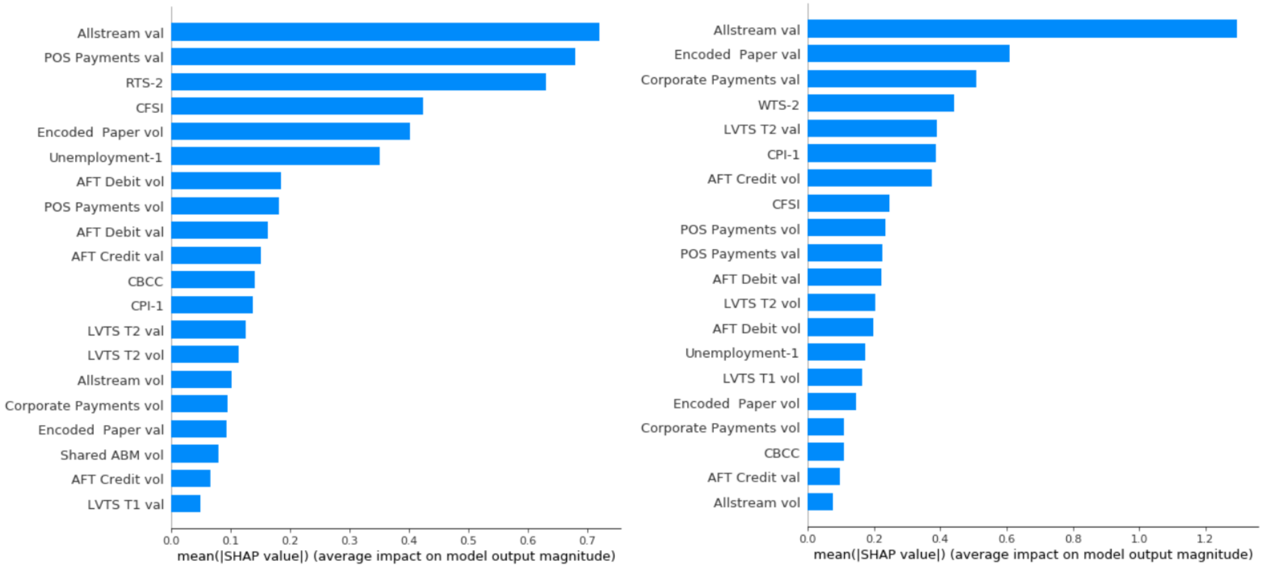

Figure 10: (Top) Retail trade sales (RTS) and (bottom) wholesale trade sales (WTS). The SHAP global feature importance measured as the mean absolute Shapley value of each instance in the entire training sample (Mar 2005 to Dec 2020) at time horizon $t+1$ using the gradient boosting model.

6 Conclusion

We use a set of comprehensive and timely payment systems data with ML models for macroeconomic nowcasting. The payments data provide information about the economy in real time and help reduce dependence on variables that are released with a significant delay. ML provides a set of advance econometric tools to effectively process various payment streams and capture the sudden, large, and nonlinear effects of a crisis. To improve the effectiveness of ML models for prediction, we use a Shapley value-based approach for model interpretability and a device specialized cross-validation strategy to avoid nowcasting model overfitting.

Our results suggest that payments system data and ML models can lower nowcast errors significantly over benchmark models. We observe up to a 40% reduction in RMSE over the linear benchmark in nowcasting GDP, RTS, and WTS. The most significant improvements are observed when we use the same month's payments data as the target variable. Our nowcasting model out-of-sample performance is relatively higher during the COVID-19 period compared to the pre-COVID period. We also notice that the ML model performance changes slightly for different nowcasting cases. However, the gradient boosting regression model shows consistently good performance. The importance of payments data (especially the Encoded Paper stream) increases during crisis periods. Nonetheless, some payment streams strongly influence the model even during normal periods.

We also demonstrate the usefulness of the Shapley value-based SHAP approach in gaining insights into ML model predictions at each nowcasting step and for the entire training sample. Further, we show how the dependence plots could help understand the relationship between the predictors' values and their influence on the target. Such insights could be valuable in macroeconomic monitoring and prediction, especially during crisis periods. Additionally, we find that the proposed cross-validation technique can help reduce overfitting and improve prediction accuracy in macroeconomic nowcasting models. In conclusion, this paper substantiates the use of payments data and ML models for macroeconomic prediction and provides a set of econometric tools to overcome 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. https://doi.org/10.1080/07474938.2010.481556.

Alvarez-Melis, D. and T. S. Jaakkola (2018). On the robustness of interpretability methods. *arXiv preprint arXiv:1806.08049*. https://arxiv.org/pdf/1806.08049.pdf.

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. https://www.ijcb.org/journal/ijcb19q4a2.pdf.

Arjani, N. and D. McVanel (2006). A primer on Canada's large value transfer system. https://www.bankofcanada.ca/wp-content/uploads/2010/05/lvts_neville.pdf.

Athey, S. (2017). The impact of machine learning on economics. In *Economics of Artificial Intelligence*. University of Chicago Press. http://www.nber.org/chapters/c14009.

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.

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

Babii, A., E. Ghysels, and J. Striaukas (2021). Machine learning time series regressions with an application to nowcasting. *Journal of Business & Economic Statistics*, 1–23.

Bańbura, 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. (2009). Learning deep architectures for AI. *Foundations and trends® in Machine Learning 2*(1), 1–127. https://www.iro.umontreal.ca/~lisa/pointeurs/TR1312.pdf.

Bergmeir, C. and J. M. Benítez (2012). On the use of cross-validation for time series predictor evaluation. *Information Sciences 191*, 192–213. doi: 10.1016/j.ins.2011.12.028.

Bergmeir, C., R. J. Hyndman, and B. Koo (2018). A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics & Data Analysis 120*. https://doi.org/10.1016/j.csda.2017.11.003.

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. (1996). Bagging predictors. *Machine learning 24*(2), 123–140.

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

Buckmann, M., A. Joseph, and H. Robertson (2021). Opening the black box: Machine learning interpretability and inference tools with an application to economic forecasting. In *Data Science for Economics and Finance*, pp. 43–63. Springer, Cham. https://doi.org/10.1007/978-3-030-66891-4_3.

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. T. and A. Desai (2020). Using payments data to nowcast macroeconomic variables during the onset of COVID-19. *Journal of Financial Market Infrastructures 9*(1). doi: 10.21314/JFMI.2021.004.

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.

Chu, C.-K. and J. S. Marron (1991). Comparison of two bandwidth selectors with dependent errors. *The Annals of Statistics 19*(4), 1906–1918.

Cimadomo, J., D. Giannone, M. Lenza, F. Monti, and A. Sokol (2022). Nowcasting with large bayesian vector autoregressions. *Journal of Econometrics 231*(2), 500–519.

Coulombe, P. G., M. Leroux, D. Stevanovic, and S. Surprenant (2020). How is machine learning useful for macroeconomic forecasting? *arXiv preprint arXiv:2008.12477*.

Coulombe, P. G., M. Marcellino, and D. Stevanovic (2021). Can machine learning catch the COVID-19 recession? *National Institute Economic Review 256*, 71–109. doi: https://doi.org/10.1017/nie.2021.10.

Dahlhaus, T. and A. Welte (2021). Payment habits during COVID-19: Evidence from high-frequency transaction data. Technical report, Bank of Canada. https://www.bankofcanada.ca/2021/09/staff-working-paper-2021-43/.

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/.

Foroni, C., M. Marcellino, and D. Stevanovic (2020). Forecasting the COVID-19 recession and recovery: Lessons from the financial crisis. *International Journal of Forecasting*. https://www.econstor.eu/bitstream/10419/229082/1/ecb-wp2468.pdf.

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.

Gogas, P., T. Papadimitriou, and E. Sofianos (2022). Forecasting unemployment in the euro area with machine learning. *Journal of Forecasting 41*(3), 551–566.

Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep learning*. MIT Press. https://www.deeplearningbook.org/.

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. https://hastie.su.domains/ElemStatLearn/printings/ESLII_print10.pdf.

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. https://doi.org/10.1162/neco.1997.9.8.1735.

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. http://escoe-website.s3.amazonaws.com/wp-content/uploads/2020/07/13161005/ESCoE-DP-2018-12.pdf.

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. https://www.academia.edu/download/61919163/applied-predictive-modeling-max-kuhn-kjell-johnson_151820200128-106306-1lfa51q.pdf.

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. https://cogns.northwestern.edu/cbmg/LiawAndWiener2002.pdf.

Liu, J., C. Li, P. Ouyang, J. Liu, and C. Wu (2022). Interpreting the prediction results of the tree-based gradient boosting models for financial distress prediction with an explainable machine learning approach. *Journal of Forecasting*.

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 *Advances in Neural Information Processing Systems 30*, pp. 4765–4774. Curran Associates, Inc. https://arxiv.org/abs/1705.07874.

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. https://christophm.github.io/interpretable-ml-book/.

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. https://arielrubinstein.tau.ac.il/books/GT.pdf.

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*. doi: 10.1016/j.ijforecast.2020.10.005.

Shapley, L. S. (1953). A value for n-person games. *Contributions to the Theory of Games 2*(28), 307–317. https://www.rand.org/content/dam/rand/pubs/research_memoranda/2008/RM670.pdf.

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. https://dl.acm.org/doi/pdf/10.1145/3375627.3375830.

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 Monetary Review*. https://www.nationalbanken.dk/en/publications/Documents/2010/07/can%20crises_2q_2010.pdf.

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. https://pubs.aeaweb.org/doi/pdfplus/10.1257/jep.28.2.3.

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.

Yoon, J. (2021). Forecasting of real gdp growth using machine learning models: Gradient boosting and random forest approach. *Computational Economics 57*(1), 247–265. https://doi.org/10.1007/s10614-020-10054-w.

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. This is followed by the stream label and a short description.

- A: ABM adjustments - processes POS payment items used to correct errors from shared ABM network stream N.
- B: Canada Savings Bonds - part of government items. Comprises bonds (series 32 and up and premium bonds) issued by the Government of Canada. Start date: April 2012.
- C: AFT credit - processes direct deposit (DD) items 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 that include cheques, inter-member debits, money orders, bank drafts, settlement vouchers, paper PAD, money orders, etc.
- F: Paper-based remittances - used for paper bill payments, that is, MICR-encoded with a CCIN for credit to a business. It is similar to electronic bill payments (stream Y).
- G: Receiver General warrants - part of government payments items. Processes paper items payable by the Receiver General for Canada. Start date: April 2012.
- H: Treasury bills and old-style bonds - part of government paper items. It processes certain Government of Canada paper payment items such as treasury bills, old-style Canada Savings Bonds, coupons, etc. Start date: April 2012.
- I: Regional image captured payment (ICP) - processes items entered into the ACSS/USBE on a regional basis. Start date: Oct 2015.
- J: Online payments - processes electronic payments initiated using a debit card through an open network to purchase goods and services. Start date: June 2005.
- K: Online payment refunds - processes credit payments used to credit a cardholder's account in the case of refunds or returns of an online payment (stream J). Start date: June 2005.
- L: Large-value paper - similar to stream E with value cap. Starting in Jan 2014, this stream merged into encoded paper stream E.
- M: Government direct deposit - processes recurring social payments such as payroll, pension, child tax benefits, social security, and tax refunds. Start date: April 2012.
- N: Shared ABM network - POS debit payments used to withdraw cash from a card-activated device.

- O: ICP national - processes electronically imaged paper items that can be used to replace physical paper items such as cheques, bank drafts, etc.
- P: POS payments - processes payment items resulting from the POS purchase of goods or services using a debit card.
- Q: POS return - processes 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 - processes national image-captured payment returned items entered into the ACSS/USBE on a national basis. Start date: Oct 2015.
- U: Unqualified paper payments - processes paper-based bills of exchange that do not meet Canada payments association requirements for encoded paper classification.
- X: Electronic data interchanges (EDI) payment - processes exchange of corporate-to-corporate payments such as purchase orders, invoices, and shipping notices.
- Y: EDI remittances - processes remittances for electronic bill payments such as online bill and telephone bill payments.
- Z: Computer rejects - processes encoded paper items whose identification and tracking information cannot be verified through automated processes.

The LVTS settles payments through two tranches, T1 and T2. Each tranche settles two types of payments: interbank and third-party funds transfers. The LVTS also includes transactions to and from the Bank of Canada (See Arjani and McVanel 2006 for more details).

- Foreign exchange payments and payments related to the settlement of the Canadian-dollar leg of FX transactions undertaken in the continuous linked settlement (CLS) system.
- Payments related to Canadian-dollar-denominated securities in the CDSX operated by clearing and depository services (CDS).
- Payments related to the final settlement of the ACSS.
- Large-value Government of Canada transactions (federal receipts and disbursements) and transactions related to the settlement of the daily receiver.
- The Bank of Canada's large-value payments and those of its clients, which include Government of Canada, other central banks, and certain international organizations.

B Machine Learning Models

In this section, we briefly discuss the ML models employed for nowcasting. For each model considered, 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 model training, tuning, and cross-validation procedures.

B.1 Elastic Net Regularization

Elastic net (ENT) 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 on the number of selected variables (Zou and Hastie 2005; Hastie et al. 2009).

Consider a set $X = \{\mathbf{x^1}, \mathbf{x^2}, \dots, \mathbf{x^M}\}$ 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

$$\min_{\mathbf{w}} \|\mathbf{y} - \hat{\mathbf{y}}(X, \mathbf{w})\|_2^2 + \lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2, \tag{7}$$

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 the penalty factor $\lambda_1 \|\mathbf{w}\|_1 + \lambda_2 \|\mathbf{w}\|_2^2$. The ENT is particularly useful with a large set of predictors and correlated features. Note that we use the scikit-learn library for the implementation of ENT. We explore and tune the 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 example, the selected model for GDP nowcasting At $t+1$: $alpha$ is 0.001, $l1\_ratio = 0.5$. For other parameters, we use the default values (Pedregosa et al. 2011).

B.2 Support Vector Regression

Support vector regression (SVR) is another model useful when there are problems with multiple predictors. It uses a different objective function than the 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, the linear function $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i + b$ (where b is a bias and $i = 1, 2, \dots N$) that has at most ε deviation from the actual $\mathbf{y}$ for all the training data. Therefore, the objective function to minimize is

$$\frac{1}{2} \|\mathbf{w}\|_2^2 + C \sum_{i=1}^{N} |\mathbf{y}_i - f(\mathbf{x}_i)|_\varepsilon, \tag{8}$$

subject to

$$\mathbf{y}_i - f(\mathbf{x}_i) \leq \varepsilon \tag{9}$$

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

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, see Smola and Schölkopf 2004; Hastie et al. 2009.

Note: we use scikit-learn library for implementing SVR. We explore and tune the following hyperparameters: kernel type, degree of the polynomial kernel function, and regularization parameter constants C and ε. For example, the selected model for GDP nowcasting At $t+1$: kernel is rbf, degree $= 2$, $C = 3$, and $\varepsilon = 0.3$. We use the default values for other parameters.

B.3 Random Forest

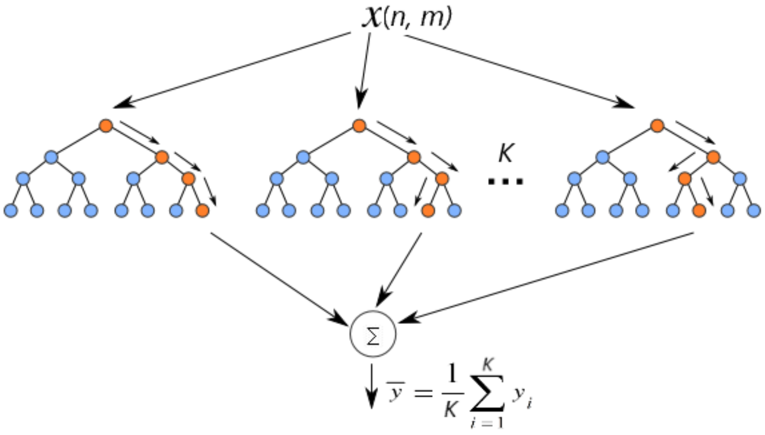

Figure 11: Random forest with K trees using n samples and m features for each tree.

Another popular approach is random forest regression (RFR). It is a decision tree (DT)-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 and ENT. In RFR, each DT is independently built from a bootstrapped subset of the training set. Each bootstrap sample can 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 Figure 11. The two levels of randomness (i.e., a random subset of the sample and the features) incorporated to build the DT can help to reduce variance in the predictions. RFR has been shown to handle highly nonlinear interactions between multiple predictors and a target variable (Breiman 2001; Liaw and Wiener 2002).

Note: we use scikit-learn library for the implementation of the RFR regression. 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*. For example, the selected model for GDP nowcasting At $t+1$: *n_estimators* is 400, *max_depth* is 4 and *min_samples_split* is 2. We use the default values for other parameters.

B.4 Gradient Boosting

Similar to RFR, gradient boosting regression (GBR) is a DT-based non-parametric ensemble learning approach. It is a general technique of boosting in which a sequence of weak learners (e.g., small DTs) are built on a repeatedly modified version of the training set. 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).

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

$$\hat{\mathbf{y}}_i = H_p(X_i) = \sum_{1}^{p} h_p(X_i), \tag{11}$$

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

$$H_p(X) = H_{p-1}(X) + \gamma h_p(X), \tag{12}$$

where γ is the 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:

$$h_p = {}^{\arg}{}_{\mathbf{p}}{}^{\min} L_p. \tag{13}$$

Both RFR and GBR techniques are interpretable to a certain extent because these models use DTs as their base learners. The DTs perform feature selection from the set provided 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). Note: we use scikit-learn library for implementation of GBR. 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. For example, the selected model for GDP nowcasting At $t+1$: *n_estimators* is 1000, *max_depth* is 1, and *learning_rate* is 0.1. We use the default values for other parameters.

B.5 Feed-Forward Artificial Neural Network

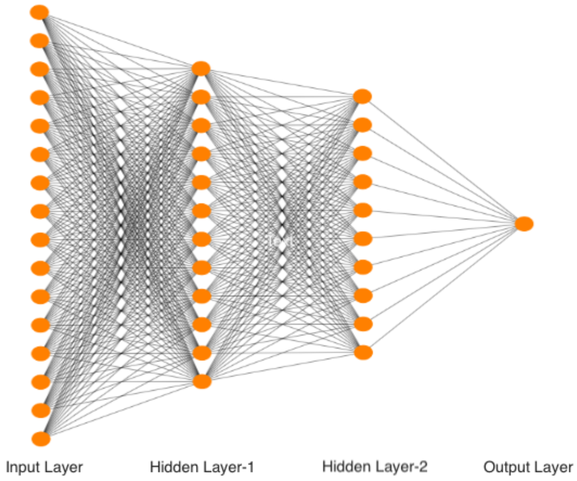

Figure 12: Schematic of densely connected feed-forward neural network with two hidden layers.

A feed-forward artificial neural network (ANN) with hidden layers is multiple layers of artificial neurons sandwiched between input and output layers, as depicted in Figure 12. In a feed-forward ANN, the data always moves forward through the network layers. It starts in the input layer, for instance, each input feature instance $\mathbf{x}_i$ is multiplied by its 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 nonlinear activation function σ, resulting in a nonlinear 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, denoted 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 iteratively adjust the weights and biases of the network to optimize the objective function. For further details on the activation function and optimization procedure, see Bengio (2009); Goodfellow et al. (2016).

The greater the number of layers, the deeper the network. 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 nonlinear relationships between the input and output. However, DNN requires tuning 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*, and the learning rate schedule for weight updates. For example, the selected model for GDP nowcasting At $t+1$: *hidden_layer_sizes* is 3, *activation* is *relu*, and *learning_rate* is 0.05. We use the default values for other parameters.

B.6 ML Models Performance Caparison with DFM

Here we compare ML models' performance against DFM with the payments data (main case). Similar to the model employed in Chernis and Sekkel (2017), we use the DFM model with two factors (including additional factors does not improve model performance) and one lag in the VAR driving the dynamics of those factors. Idiosyncratic components are assumed to follow an AR(1) process. In nowcasting GDP, RTS, and WTS, the GBR, ENT, and feedforward ANN models—in many cases—perform better than DFM and other ML models considered. Overall, using payments data in the ML models, we observe up to a 12-30% reduction in RMSE over DFM with the payments data.

Table 3: Out-of-sample RMSE comparisons of DFM with ML models for seasonally adjusted YOY growth rate of macro variables at the horizons $t+1$ (top 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]	**% Reduction**[e]
GDP	1.00	**0.96**	1.41	1.11	**0.81**[f]	**0.82**	19
RTS	1.00	**0.89**	1.27	1.07	**0.85**	1.02	15
WTS	1.00	**0.96**	1.14	**0.82**	**0.69**	**0.51**	31

Target	**DFM**	**ENT**	**SVR**	**RFR**	**GBR**	**ANN**	**% Reduction**
GDP	1.00	**0.87**	1.62	1.14	**0.82**	**0.85**	18
RTS	1.00	**0.87**	1.36	1.15	**0.90**	**0.97**	11
WTS	1.00	**0.89**	1.19	**0.91**	**0.81**	**0.70**	19

[a] In-sample training period, Mar 2005 to Dec 2018, ($p = 166$) and out-of-sample testing period, Jan 2019 to Dec 2020, ($p = 24$). All RMSEs are normalized with respect to DFM. The performance gain using ML models for time horizon t are much smaller, however, GBR model performed better compred to other ML models.

[b] RTS-retail trade sales, WTS-wholesale trade sales. Note: we use the latest available values of targets for these exercises.

[c] For 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. 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 ANN. For these ML models, we select the model parameters and number of payment predictors based on target variables using the cross-validation procedure outlined in section 4. Further details on these models are provided in Appendix B. Model selection and cross-validation procedures are detailed in Appendix C and D.

[e] Percentage reduction in RMSE over DFM for GBR model.

[e] The models with out-of-sample prediction RMSE less than DFM (< 1) are highlighted (bold) and the best model is also underlined.

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 13). In our case, the training set is Mar 2005 to Dec 2018, and the test set is Jan 2019 to Dec 2020 (highlighted in blue).

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

3. Select two dates in the training set that define the validation superset (highlighted in gray in Figure 13). To include the global financial crisis, we choose those dates between Oct 2008 and Dec 2018.

4. Next, for each fold in the cross-validation, we randomly sample 24 points (it is the same as the test set) from the validation superset as the validation subset (see Figure 2 for an example).

5. Using the selected parameters grid and validation subset, we do the following:
   (a) For each iteration in the expanding window over the validation subset, select a data point from that subset as the out-of-sample test point and use all the data points up to that point for training (see Figure 3 where red dots are test points and blue dots are training points).
   (b) Fit the model on the selected training sample.
   (c) Using the trained model, predict for the selected sample point in the validation subset.
   (d) Repeat steps a, b, and c for each point in the validation subset.
   (e) After finishing iterating the chosen validation subset, compute the validation RMSE.

6. Repeat steps 4 and 5 k-times (typically k is between 5 to 10), each time using a new validation subset.

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 reusing the standard expanding window approach, as illustrated in Figure 3.

In Figure 14, we present standardized distribution of the target variables (GDP growth rate) for the out-of-sample testing period (Jan 2019 to Dec 2020). In the same figure, we plot the distribution for a validation sample of the standard expanding window approach and the proposed randomized expanding window approach for cross-validation. The distribution of each of the randomized validation sets typically contains a few sample points from 2008 GFC; therefore, it is skewed towards the left—similar to the test set; however, this is not true for each standard validation set, except for the first validation set where we have the entire 2008 GFC period (see Figure 2).

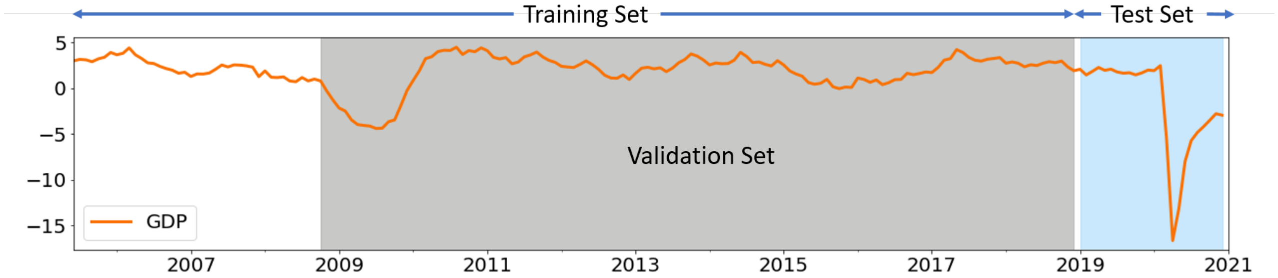

Figure 13: Schematic of data splits for cross-validation. First, the dataset is divided into a training set with a validation subset sampled from the highlighted gray area and a test set (highlighted in blue). The orange line shows the GDP growth rate.

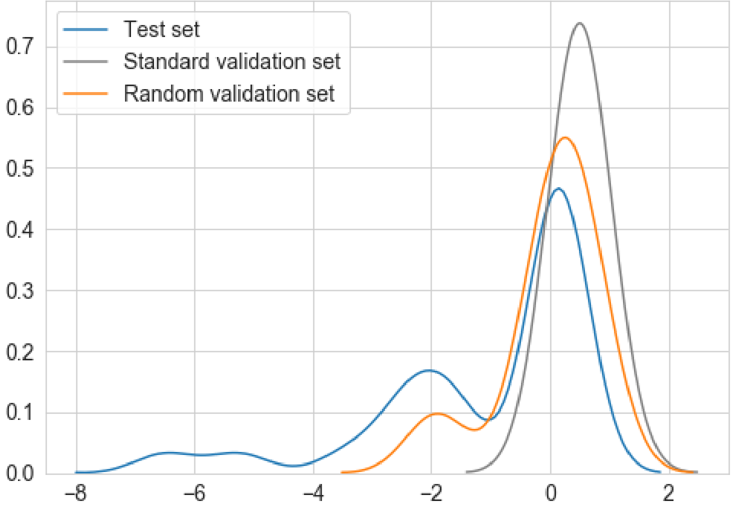

Figure 14: GDP: Distribution of the standardized test set and a typical validation set for standard *k*-fold expanding window approach (standard validation set) and random expanding window approach proposed here (random validation set). The distribution of the proposed random validation set remains similar across all *k*-folds; however, the distribution of the standard validation set could change based on the sample period.

D Feature Selection

To select the *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 the *k*-best variables, we employ 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 the *k* highest-scoring features.

In Figure 15, 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 GFC. The scores are steady and low during normal times (2011–2019) except for the encoded paper value (E), Allstream value (All), and LVTS-T2 volume (T2), for which scores remain high. During the COVID-19 crisis (Mar to Dec 2020), however, we see opposite behaviour in the prediction scores of a 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.

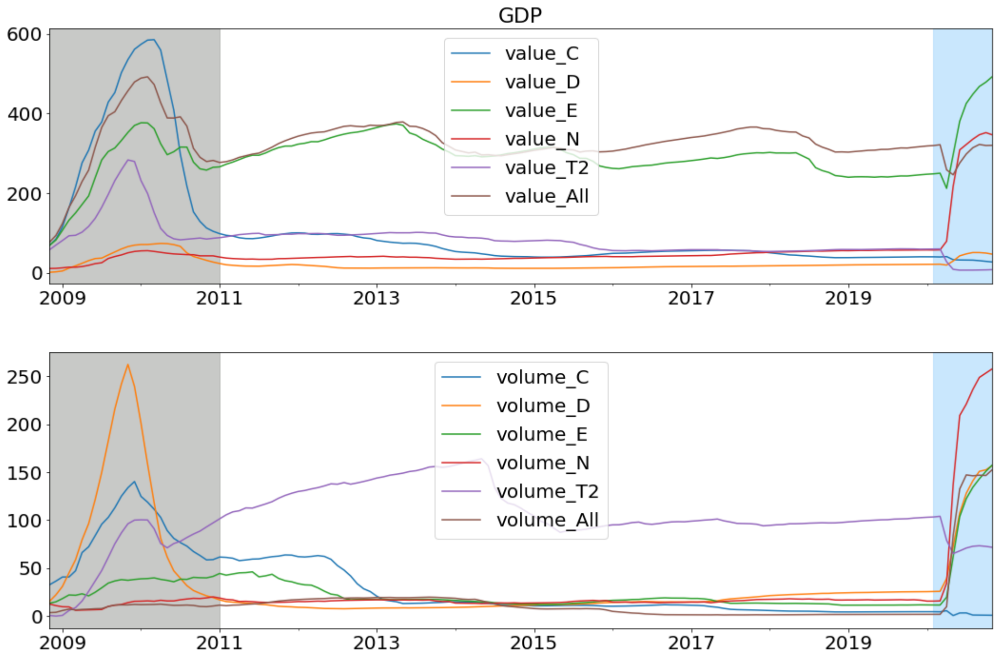

Figure 15: The *F*-score of a few selected payments streams (values-top, volumes-bottom) for GDP nowcasting. Higher scores mean a high prediction value. These plots are obtained after each training session of the expanding window approach, ranging from Oct 2008 to Dec 2020. The 2008 GFC period is highlighted in gray; 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 that 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 game, (N, v), where N is a finite set of players indexed by i and v is the utility function or payoff function, the Shapley value can be obtained by this theorem, which satisfies the symmetry, dummy, and additivity axioms (Osborne and Rubinstein 1994):

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

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 the coalition S that does not have i. The middle part shows how to compute different possible ways in which we could have formed the coalitions. Then, we take an average of possible ways that we could have done the marginal value calculation.

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

$$\mathscr{F}(S) = \phi_0 + \sum_{i=1}^{M} \phi_i S_i, \tag{14}$$

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

To illustrate, 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 16) 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, that is, the base value is 0.2, and for the current instance (for example, month t), our model predicts GDP growth 0.5.

3. By computing the Shapley values for all possible coalitions among three predictors, we can explain the difference between actual prediction (0.5) at current month t 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, predictor 2 pushes 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 of 0.2, leading to the final prediction of 0.5 growth.

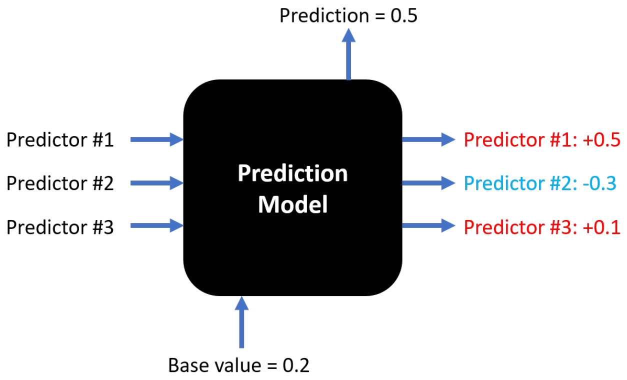

Figure 16: The SHAP explainer provides the marginal contribution of each predictor.

The SHAP values tell us which predictor contributes the most in the current instance of the prediction, that is, a local interpretation. Similarly, by using the Shapley values for each instance in the sample, we can get the average contribution of each predictor for that sample. That would give us a global interpretation of the model in terms of its 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, the feature importance plots and summary plots (Figure 4 and 5) are useful for global model interpretations. The force plots or clustered force plots (Figure 6 and 7) are useful for local interpretation, that is, at each instance of prediction. Also, the dependence plots (Figure 8) could be valuable for understanding the relationships between given predictors and the targets.

The SHAP, although a powerful model-agnostic ad-hoc tool developed based on theoretical foundations for model interpretability, has some shortcomings, 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 many predictors. However, for macroeconomic predictions models, we have comparatively fever predictors (20–50) and fewer instances (a few hundred data points). Therefore, it is not much of an issue in such applications. Another issue with KernelSHAP is that it is sensitive to colinearity in the predictors. The TreeSHAP approach developed in Lundberg et al. 2020 overcomes some of those challenges to a certain extent (Molnar 2020). Furthermore, as shown by Slack et al. 2020, it is possible to misuse such ad hoc tools to hide model biases. However, it is not much of a concern for the macroeconomic prediction models we deal with in this paper. Additionally, the authors conclude that the SHAP is less prone to such problems than several other interpretation tools.

E.1 Global Feature Importance Comparison

We can also use impurity, or permutation-based global feature importance approaches for tree-based models like GBR and RFR. Amongst the two the permutation-based approach is shown to be more useful for nonlinear models (Breiman 1996; Molnar 2020). In Figure 17, we compare the feature importance of the gradient boosting model trained on the entire training sample (Mar 2005 to Dec 2020) at time horizon $t+1$. The permutation-based approach is similar to SHAP for the top three major contributors and matches eight out of the top ten highest contributors but slightly in a different order. Moreover, all three approaches rank the same three predictors in the top five list, and the Encode paper stream remains the most prominent predictor in all three approaches.

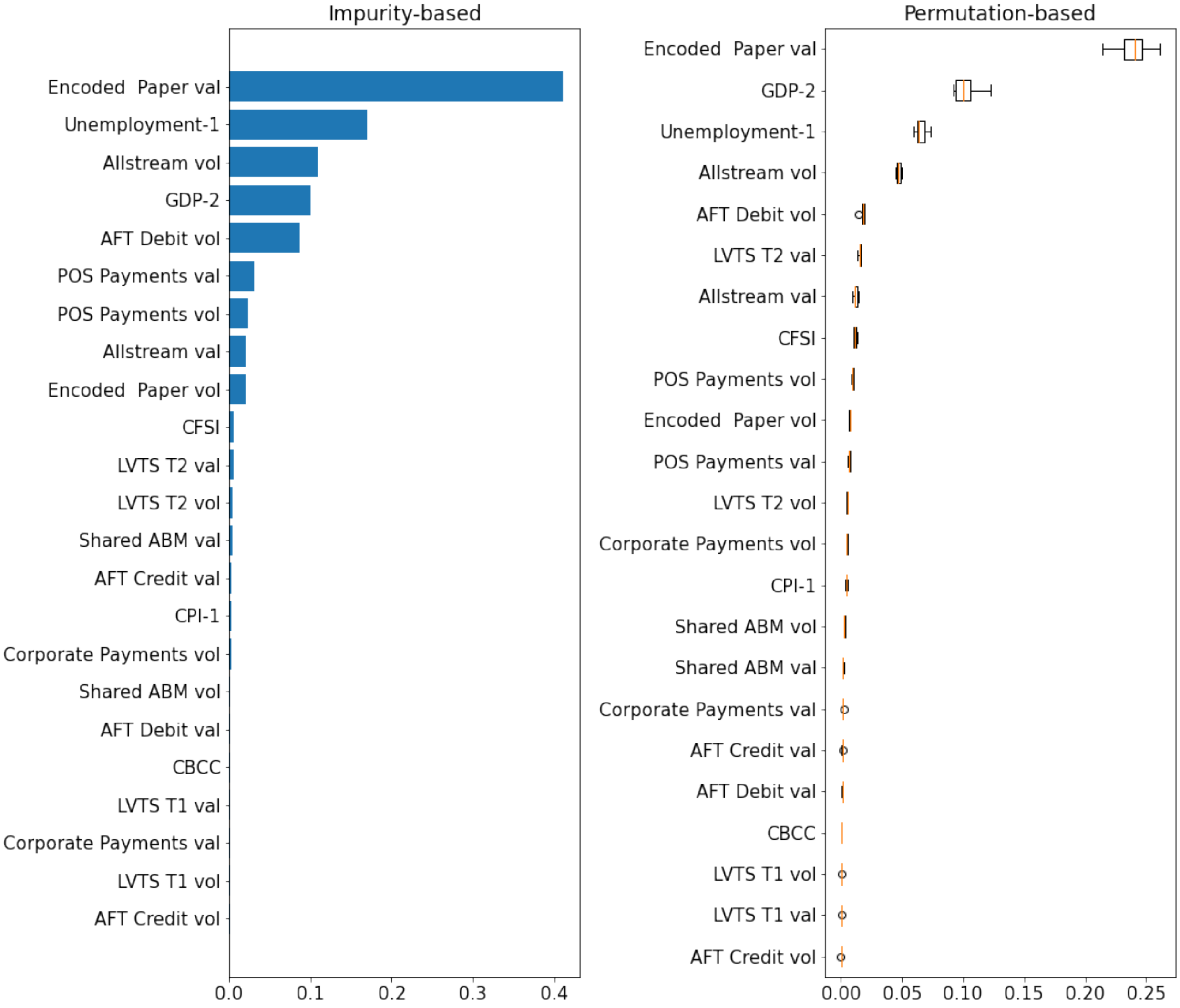

Figure 17: GDP: Global feature importance for the entire training sample (Mar 2005 to Dec 2020) at time horizon $t+1$ using the gradient boosting model. (Left) impurity-based feature importance and (right) permutation-based feature importance.

F Nowcasting Performance for Benchmark and ML Models

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 18. Incorporating payments data in ML models provides downturn and recovery indications (during crisis periods) much better than the benchmark model in both in-sample and out-of-sample periods.

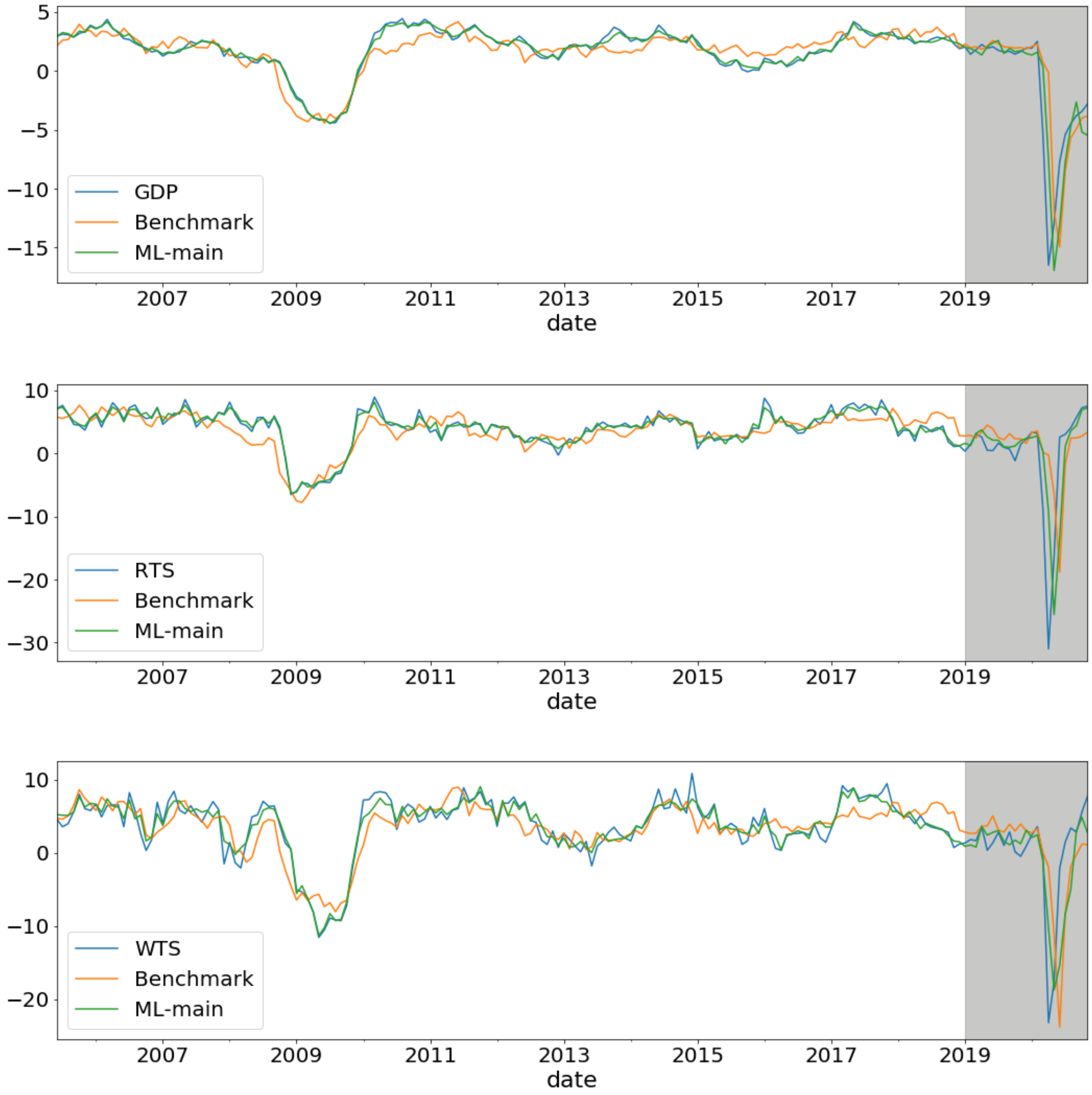

Figure 18: In-sample and out-of-sample prediction comparison for the ML main-case model (with lowest RMSE) and the benchmark model (OLS with base case) for time horizon $t+1$. The in-sample training period is Mar 2005 to Dec 2018, and the out-of-sample testing period is Jan 2019 to Dec 2020 (highlighted in gray).

G Nowcasting Performance for Normal and COVID-19 Periods

In this section, we separately test our nowcasting model's 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 19. To demonstrate, 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 (15–25% reduction in RMSE) using payments data (Table 4). These results demonstrate the usefulness of payments data during normal periods and crisis periods.

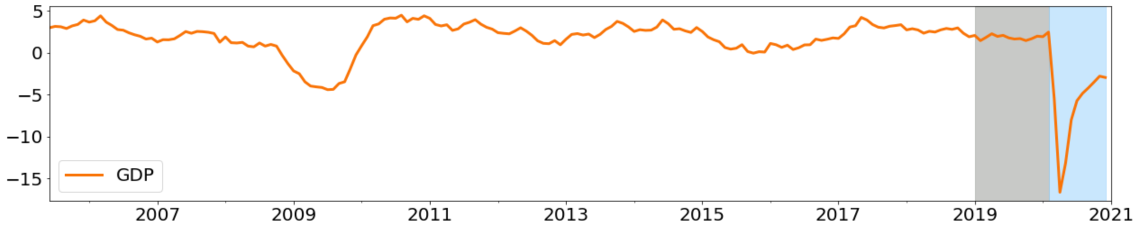

Figure 19: The test sample of GDP nowcasting exercises is divided into two sets: the pre-COVID-19 test set (highlighted in gray) and the COVID-19 test set (highlighted in blue).

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 time horizon $t+1$, 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 the 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. These numbers show the percentage gain over benchmark cases for the same period.

H 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 20). 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 the nowcasting horizon $t+1$ compared to other time horizons.

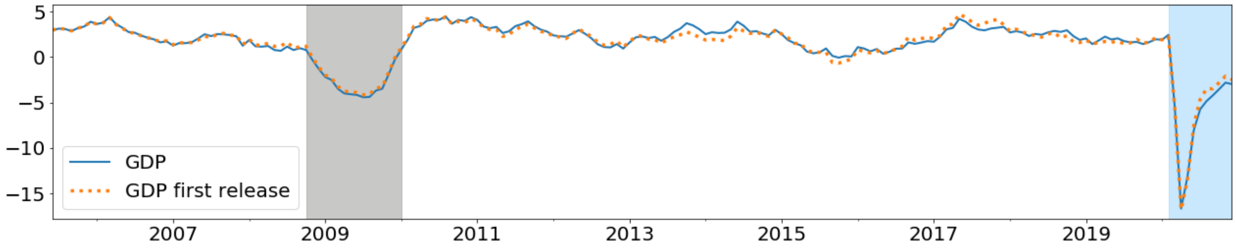

Figure 20: YOY seasonally adjusted GDP growth rates comparison of the first releases with latest releases. The GFC is highlighted in gray and the COVID-19 period is highlighted in blue.

Table 5: Out-of-sample RMSE comparisons for the seasonally adjusted YOY growth rate of GDP at nowcasting horizons t, $t+1$, and $t+2$ using the 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 Table 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 Table 36-10-0491-01.