forecasting

MDPI

Article

Macroeconomic Predictions Using Payments Data and Machine Learning

James T. E. Chapman † **and Ajit Desai** *,†

Bank of Canada, Ottawa, ON K1A 0G9, Canada; jchapman@bank-banque-canada.ca
* Correspondence: adesai@bank-banque-canada.ca
† Note: the opinions expressed herein are those of the authors and do not necessarily reflects those of the bank of Canada.

Abstract: This paper assesses the usefulness of comprehensive payments data for macroeconomic predictions in Canada. Specifically, we evaluate *which* type of payments data are useful, *when* they are useful, *why* they are useful, and whether machine learning (ML) models enhance their predictive value. We find payments data with a factor model can help improve accuracy up to 25% in predicting GDP, retail, and wholesale sales; and nonlinear ML models can further improve the accuracy up to 20%. Furthermore, we find the *retail* payments data are more useful than the data from the wholesale system; and they add more value during *crisis* and at the *nowcasting* horizon due to the timeliness. The contribution of the payments data and ML models is small and linear during low and normal economic growth periods. However, their contribution is large, asymmetrical, and nonlinear during crises such as COVID-19. Moreover, we propose a cross-validation approach to mitigate *overfitting* and use tools to overcome *interpretability* in the ML models to improve their effectiveness for policy use.

Keywords: nowcasting; payments data; machine learning; interpretability; overfitting

JEL Classification: C53; C55; E37; E42; E52

check for updates

Citation: Chapman, J.T.E.; Desai, A. Macroeconomic Predictions Using Payments Data and Machine Learning. *Forecasting* **2023**, *5*, 652–683. https://doi.org/10.3390/forecast5040036

Academic Editor: Piotr Fiszeder

Received: 6 October 2023
Revised: 10 November 2023
Accepted: 17 November 2023
Published: 27 November 2023

1. Introduction

Payment systems are often metaphorically referred to as the plumbing system of the economy, operating discreetly in the background and typically garnering attention only when they are not functioning as intended. However, with the increasing digitalization, the rise of instant payment systems, and the potential for digital currencies, we propose a new perspective: modern payment systems can be referred to as the circulatory system of the economy. Therefore, the data generated by the continuous flow of transactions in these systems can serve as valuable resources for central banks, facilitating real-time assessments of economic health, particularly in times of crisis.

Timely prediction of the economy's short-term dynamics—known as nowcasting—is a vital input into every economic agent's decision-making process, including central banks. However, it is difficult for several reasons. For instance, many different data series are needed to adequately describe the state of the economy, but many of these, particularly official national account statistics, are released with significant lags [1,2]. 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 [3,4].

In the past, econometricians have used new data and developed new techniques for nowcasting [5–13]. In a similar spirit, 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; and asses which types of data are useful, when they are useful, and why they are useful, using a factor model. We then test whether ML can improve the prediction accuracy using the following five models: elastic net, support vector machines, random forest, gradient boosting, and artificial neural network [14]. We use these parametric and non-parametric ML models because they are popular among time series forecasters and preferred in macroeconomic prediction problems [7,15,16].

The payments data have been used for macroeconomic predictions in the past [9,17–20]. Canadian payments data are a particularly good candidate because they record transactions processed in various payment instruments. Thus, they capture a broad range of Canadian consumers, firms, and government economic activities. Furthermore, these data are gathered electronically and hence are immediately available, and they are free of measurement or sampling errors [21]. However, traditionally, researchers have used data from a few selected payment instruments for nowcasting [9,20]. One issue with this approach is that the use and importance of particular payment instruments may rise or fall for economic and non-economic reasons. For example, 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 [22]. Using data from one or two payment streams in isolation might not capture the full economic picture; therefore, in [23], 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 wholesale payments system and cover the wider COVID-19 period, allowing us to assess which types of data are more useful and when they are useful. Furthermore, 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. Additionally, this paper employs sophisticated econometric and ML models for analysis.

To exploit non-traditional and large-scale data, researchers have recently begun using nonlinear ML models for nowcasting [23–26]. These articles suggest that ML models complement traditional econometric tools and are useful in extracting economic value from non-traditional data sources. Furthermore, they show that, in nowcasting, ML models often outperform traditional modeling approaches, such as ordinary least squares and dynamic factor models. The flexible 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 and nonlinearity in such data [27,28]. This is beneficial because some of the payment streams used here are strongly correlated with each other. 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 [29–31]. 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. Moreover, ML models are beneficial when the emphasis is on improving prediction accuracy [27,32,33].

Our results suggest that timely retail and wholesale payments system data in a factor model can help improve prediction accuracy up to 25% in nowcasting GDP, retail trade sales (RTS), and wholesale trade sales (WTS). We use a linear benchmark model with timely market and survey indices and lagged macroeconomic indicators. Specifically, consumer price index (CPI), unemployment (UNE), Canadian financial stress indicator (CFSI), and the Conference Board's consumer confidence index (CBCI). UNE incorporates the effects of public sector hiring, and CPI is useful since we are using nominal predictors [9]. CFSI is a composite measure of systemic financial market stress for Canada [34]. CBCC is based on a survey of Canadian households and has been shown to be useful in predicting household spending in Canada [35].

Moreover, in the presence of payments data and ML models—especially nonlinear gradient boosting regression (GBR)—compared to the dynamic factor models, can reduce prediction errors by as much as 20% at the nowcasting horizon. Out-of-sample performance gains using payments data and ML models are relatively greater (15 to 20%) during the COVID-19 crisis than in the pre-COVID-19 normal economic growth period; these gains are significantly higher only at the nowcasting horizon. Also, 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.

However, the use of ML models leads to a few 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 [12,31,36,37]. Likewise, we mitigate the nowcasting ML models 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 [38–40]. In addition, we address the difficulty in interpreting these models, which is important to understand their predictions—especially if they are used to support policy decisions [16,24,32,41].

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 does not. The traditional k-fold or leave-p-out cross-validation [14,38] could be challenging for the following reasons: (a) the standard k-fold splitting breaks the order (serial correlation) of the series; (b) the distribution of test and validation sets could differ; (c) the model tuned predominantly on normal periods might not perform well on the out-of-sample crisis period. To overcome this, similar to [42], we use a randomized expanding window approach with k-fold cross-validation but without changing the order of the data (see Figure 4). 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 A4 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 [43,44], based on Shapley values from the coalition game theory [45,46]. 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. The SHAP provides a way to interpret any ML model predictions in terms of the marginal contribution of each predictor toward the final prediction. Therefore, it proves valuable for determining feature importance.

Similar Shapley-value-based approaches were employed in the recent articles by [31,37] for macroeconomic predictions. They note that although such methods 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 [12], the authors propose ML-based mixed data sampling and develop the asymptotics in the context of linear regularized regressions. However, the literature is still evolving in this direction, and much progress needs to be made to use such asymptotics analysis for nonlinear ML approaches such as gradient boosting, which in many cases, including ours, outperform regularized linear models [25,28,31].

We 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; however, the data from the retail system are more useful than the data from the wholesale system. Moreover, during the COVID-19 crisis period, the contribution of the retail payments data are much higher than wholesale data and other 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.

In summary, this paper demonstrates that combining timely payments data, nonlinear ML methods, tailored cross-validation approaches, and interpretability tools can provide policymakers with sophisticated models to accurately estimate key macroeconomic indicators in near real time. Consequently, this paper adds to the growing literature on the use of non-traditional data and nonlinear ML models for macroeconomic predictions; it provides additional econometric tools to mitigate challenges in such models to improve their effectiveness for policy use.

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

2. Payments Systems Data

The vast majority of non-cash transactions require settlements 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 [9,21,23,47].

In Canada, the ACSS and LVTS are used to settle most transactions. Note that, as of September 2021, the LVTS system was replaced by a new system called Lynx [48]. 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 CAD 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); (2) paper streams, which incorporate encoded paper, paper remittances, and government paper items. Note that the ACSS supports 99% of the daily transaction volume and 13% of daily value processed by Canadian payment systems.

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 crisis 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 (see [23] for the breakdown of shares of payment streams in the ACSS).

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 CAD 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 (see [49] for further details on payment types settled in the LVTS). In the LVTS, payment value and volume are mostly processed through T2. Note that, 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.

2.1. 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. 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 obtain a better representation of payment flow, we merged several streams belonging to similar categories and settled related payments. Furthermore, to overcome the effects of consumers' payment choices, i.e., when they switch payment method, 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. Note that, for nowcasting, we are interested in capturing whether spending (or earning) has slowed (or stopped), rather than a switched payment method.

After these adjustments, we are left with seven streams from the ACSS, which comprise transactions settled in all ACSS payment instruments 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.

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

Stream (ID)	**Short Description**
AFT credit (C) [b]	Government direct deposit (GDD): payrolls and account transfers
AFT debit (D)	Pre-authorized debit (PAD): automated bill and mortgage payments
Encoded paper (E) [c]	Paper bills of exchange: cheques, bank drafts, and paper PAD
Shared ABM (N)	Debit card payments to withdraw cash at shared ABM network
POS payments (P) [d]	Point-of-sale (POS) payments using debit card
Corporate payments (X) [e]	Exchange of corporate-to-corporate and bill payments
Allstream (All) [f]	The sum of all payment streams settled in the ACSS
LVTS-T1 (T1) [g]	Time-critical payments and payments to the Bank of Canada
LVTS-T2 (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.

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 [50]. Such adjustments are performed because official macro indicators are released with similar adjustments. Specifically, we perform recursive seasonal adjustments in pseudo-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. Using growth rates (instead of levels) helps to induce (approximate) stationarity in both the target and predictors).

Our dataset does not include some payment instruments not settled through the ACSS or LVTS, such as credit card and also e-transfer payments. In 2019 credit card payments accounted for 6.2% of the value and 31.1% of the volume of total retail payments in Canada, likewise e-transfer payments accounted for 1.5% of the value and 2.5% of the volume [22]. However, the authors of [9] concluded that credit card payment data for Canada do not add significant value in nowcasting GDP and retail sales. It's important to note that in their analysis they utilized a limited sample size when examining credit card data. As such, the results might vary if a larger sample size were employed.

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. Note that, 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% [22].

2.2. 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 and timeliness and the 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. 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. (Note: the 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.)

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 obtain 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 crisis, 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 movements during both crisis periods. This is a good indication of the economic value associated with these payment streams during such times.

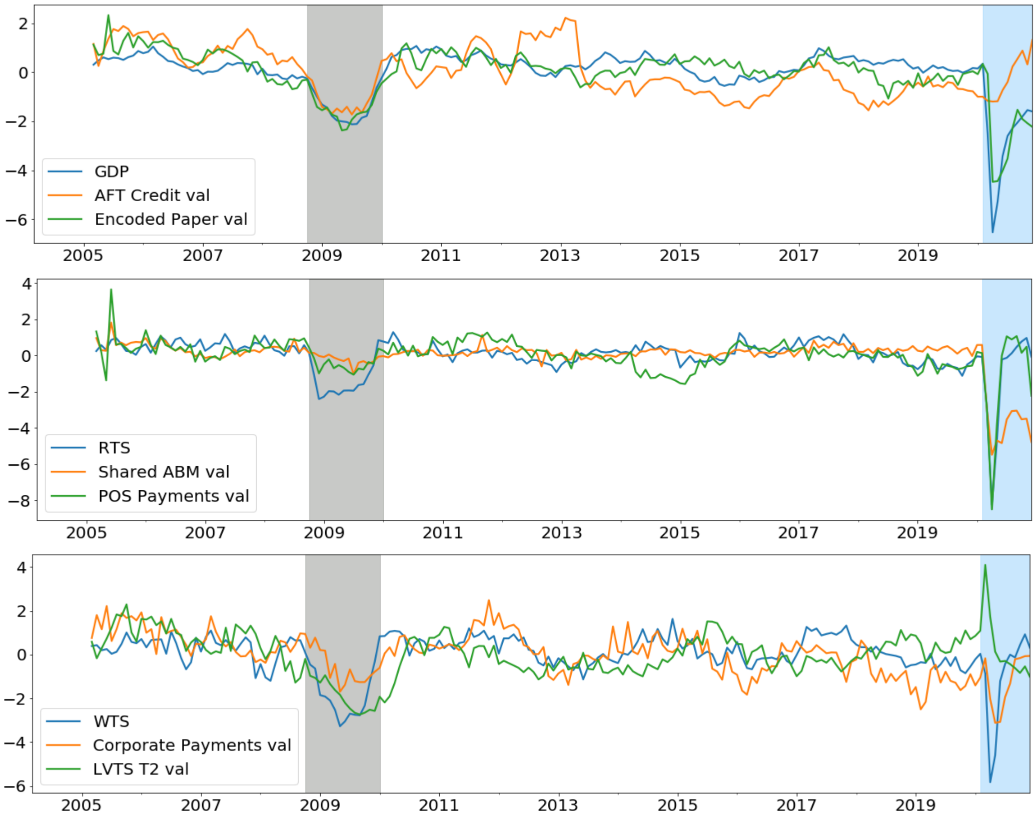

Figure 1. Standardized YOY growth rate comparisons of GDP, RTS, and WTS, with selected payment streams. Gray highlighting–GFC period; blue highlighting–COVID-19 period. Note: AFT credit includes Government direct deposit; encoded paper is the sum of multiple streams settled separately in the ACSS; POS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

During the COVID-19 period, however, we observe a complicated relationship between the macro indicators and some payment streams. 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. 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 crisis. On the contrary, starting in April 2020, the value of payments processed through the AFT 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). 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 [51]. Such complex behavior could be challenging to capture using traditional linear models.

3. 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 $X = \{\mathbf{x}^1, \mathbf{x}^2, \ldots, \mathbf{x}^M\}$ of M predictors (also called features or independent variables) and a target $\mathbf{y}$ (dependent variable), each with N data (sample) points. For example, predictors could be monthly aggregated values settled in each of the payments streams, 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 [14]. Moreover, multicollinearity in the predictors, while not reducing 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 [1,52]. Similar to [53], we estimate the factors using the model of [54], which effectively handles a large number of predictors. The basic representation of the model is

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

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

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 [52,55–57] including nowcasting Canada's GDP [53]. Therefore, we employ this model to nowcast various Canadian macro indicators using payments data.

3.1. Machine Learning Models for Nowcasting

To exploit non-traditional data, researchers have recently begun utilizing ML models for economic nowcasting [23,25,26]. 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 [24,27,30,37,58].

We use some of the recently popularized parametric and non-parametric ML approaches, such as elastic net [59], support vector machines [60], random forest [61], gradient boosting [62], and feedforward artificial neural networks [63]. 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 [14,37,63,64].

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. 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 [59]. (Note: 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 [14,59].)

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 [60,65].

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 [61,66].

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 [62].

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 [63,67].

Note that there are many advanced versions of both tree-based methods, such as LightGBM [68], and ANN-based methods, such as long short-term memory (LSTM) [69], proposed in the literature. However, in our application, the performance of these models did not surpass that of GBR. This could be attributed to our limited data sample. However, with a larger number of data points, the performance of these advanced approaches may improve.

3.2. 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 [14].

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 kth fold is used for out-of-sample testing [14]. 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 [38].

This challenge can be mitigated using an expanding window approach for cross-validation (Figure 2). Here, the end portion of the training set, often called a validation set, is kept aside for model tuning and cross-validation. Note that 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 [38]. Consequently, the model parameters can be chosen based on model performance on the validation sets (see Appendix C for additional details). 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.

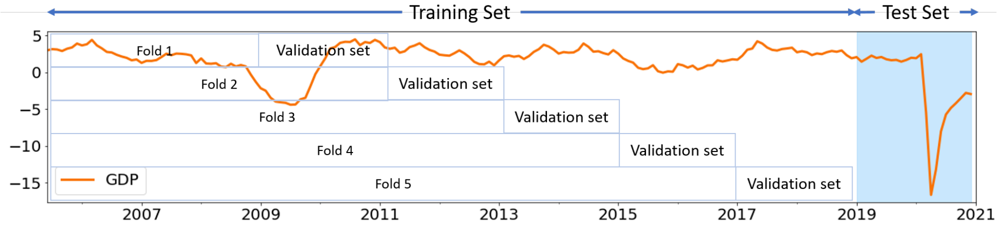

Figure 2. Schematic of standard expanding window approach for cross-validation in time series. The dataset is divided into a training set with validation subsets and a test set (highlighted in blue).

We implement a slightly altered version of the expanding window approach tailored to macroeconomic nowcasting models (see Figure 3). 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 4).

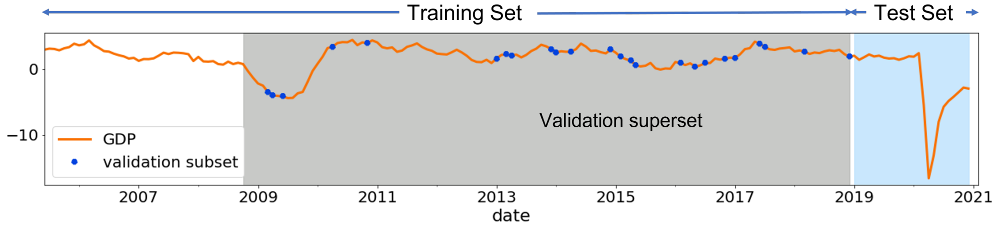

Figure 3. Schematic of the proposed randomized expanding window approach showing a typical validation subsets (represented by ●) randomly sampled from the validation superset (highlighted in gray). In both plots, the orange line shows the GDP growth rate.

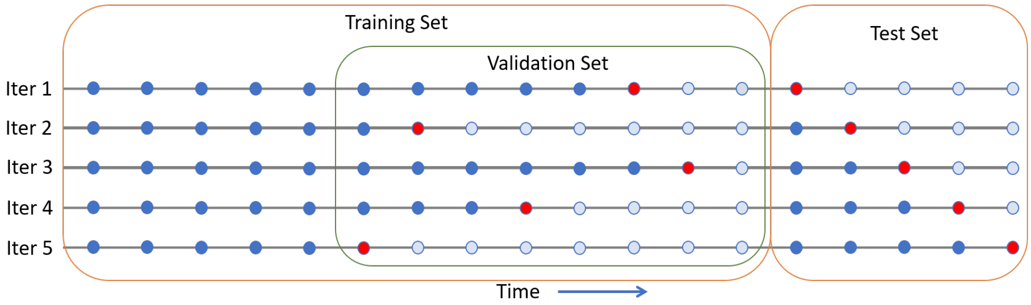

Figure 4. Schematic of expanding window approach for a typical fold in k-folds cross-validation and out-of-sample prediction. The available data are divided into training, validation, and test sets. For the given iterations of the expanding window (Iter), ● represents in-sample training points, and ● 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.

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 A4 in Appendix C) and assists in selecting a model that can perform well for both normal and crisis periods. Furthermore, our approach removes the restriction on the number of validation sets we can sample that could be binding in traditional cross-validation approaches. (Note: 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.)

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. Further details on cross-validation and model selection are discussed in Appendix D. Note that the ENT model is designed to handle a large number of predictors. However, in our case, the out-of-sample performance did not improve when we used ENT with all predictors. Similarly, RFR can also accommodate a large set of predictors by constructing trees with different subsets of features, but using RFR with all predictors did not outperform the selected approach in terms of prediction performance.

3.3. 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 [16,32,37].

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 [59,61]. Tree-based ensemble approaches such as GBR and RFR can also be interpreted—to some extent—using impurity or permutation-based feature importance methods [61,70]. However, they are mostly used for global (entire sample) interpretations. Furthermore, 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 [43]. The SHAP can be used for both the local (each prediction instance) and global interpretations. Moreover, the SHAP can be used for dependence plots which could be valuable to obtain further insights.

In SHAP, the Shapley value method from coalitional game theory [45,46] is used to fairly distribute the "payout" (=the prediction) among the "players" (=the predictors) [44]. 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.

Ref. [43] 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; (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. Furthermore, in some cases, the outcomes can easily misinterpret and lead to ambiguous conclusions. Therefore, SHAP should be used with caution [70–72]. 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 towards prediction. 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* [43,70]. Further details on Shapley values and SHAP with a test example are given in Appendix E.

3.4. 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 along with the latest CPI, unemployment, CFSI, and CBCI. The 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 and CBCI is based on the survey of Canadian households, which provides a measure of consumer optimism on current economic conditions, and both are available immediately 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.

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 econometric model. Next, we test the usefulness of the various ML models discussed above in Section 3. 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 [1,9], 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 $\mathcal{F}$ can be specified as (Note: 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)

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

Similarly, at the next nowcasting horizons $t+1$ and $t+2$, using the latest available predictors, the models can be specified as follows. (Note: 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. Furthermore, note that the left-hand sides in Equations (4)–(6) represent the same month's target but they are estimated at different time horizons.)

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

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

We train the nowcasting models using the expanding window approach as schematically outlined in Figure 4. 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). (Note: 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.)
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 4).
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 4).

4. Results and Discussion

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

Year-over-year GDP, RTS, and WTS growth rate nowcasting performance for the various cases outlined in Section 3.4 are discussed below. Table 2 compares nowcasting performance—in terms of out-of-sample RMSE—of the DFM and ML model, specifically gradient boosting model which consistently performed better across different target variables and time-horizons compared to the other models (see Table A1 in Appendix B) 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 a factor model can provide notable reductions in nowcasting RMSEs for all three macro variables considered. (Note: in this case, we use the DFM model with two factors. Including more factors does not improve the results.) Specifically, we observe a 20–25% reduction in RMSE over the benchmark case in nowcasting GDP, RTS, and WTS at time horizon $t+1$, i.e., when we use the same month's payments data as the target variable. Comparatively, the information gained using payments data is smaller at prediction 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. These results suggest that payments data provide the most prediction 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).

Next, we use payments data in the ML models (see Table 2). Overall, in predicting GDP, RTS, and WTS at all three-time horizons, the GBR model performs better than DFM (up to 20%) and other ML models (see Table A1 in Appendix B). Specifically, we observe a 35–40% reduction in RMSE over the benchmark model at time horizon $t+1$. The main case predictions at this time horizon are statistically significant for the Diebold–Mariano test using the benchmark. These results suggest that flexible ML models, such as GBR, can handle multiple predictors efficiently and are useful in increasing prediction accuracy. Moreover the proposed randomized expanding window CV approach selects models that perform slightly better (3 to 12%) than the standard expanding window CV approach in nowcasting GDP, RTS, and WTS.

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
GDP	3.97	2.98	2.43 *	39
RTS	8.47	6.36	5.44 *	36
WTS	7.17	6.18	4.28 *	41
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–Dec 2018, and out-of-sample testing period, Jan 2019–Dec 2020.
[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 G.
[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 [53], 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 A1 in Appendix B). We select model parameters using the cross-validation procedure outlined in Appendices 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% level, for the Diebold–Mariano test using the benchmark.

We separately test our model's out-of-sample performance during a normal economic period (Jan 2019–Feb 2020) and the COVID-19 period (Mar 2020–Oct 2020) of the test sample (see Table A2 in Appendix F). We observe a higher gain using payments data during the time of crisis (up to 35% RMSE reduction) compared to the normal period (15–25% reduction in RMSE). These results show that the payments data are useful during normal periods, and their significance surges during crises, substantially improving our model's performance during those periods.

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.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 A3 in Appendix G). Comparatively, the models using payments data perform better against the latest vintages. This makes sense, given that the latest vintages are more accurate (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 the COVID-19 pandemic.

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–Dec 2020) for training. In Figure 5, (left), 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 high to low). For example, in the case of in-sample training data, both the encoded paper value and the 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). Note that for the tree-based models like GBR, we can also use impurity or permutation-based global feature importance [61]. 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 A7).

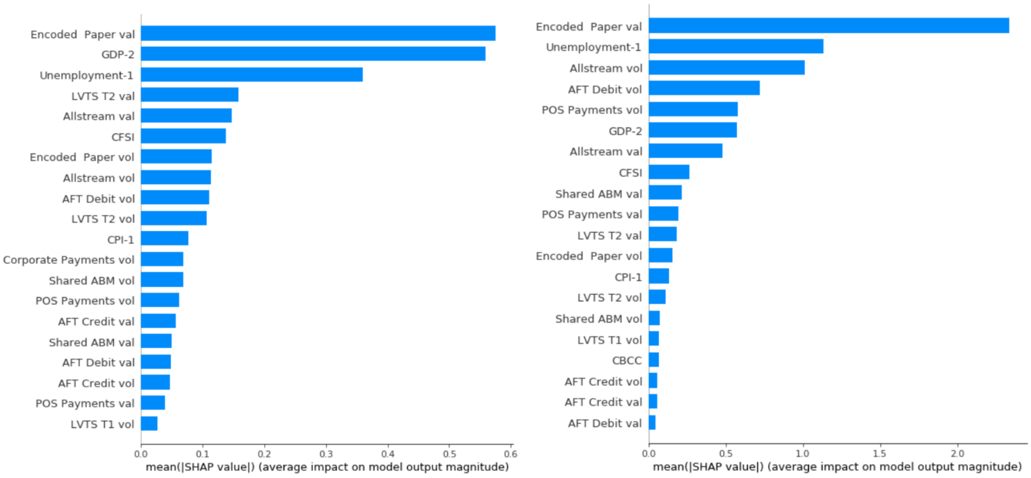

Figure 5. GDP: SHAP global feature importance measured as mean absolute Shapley values for each instance in the sample. The left plot is for the entire training sample (Mar 2005–Dec 2020) and the right only for the COVID-19 period (Mar 2020–Dec 2020). The features are ranked from high (top) to low (bottom) based on average Shapley values.

In Figure 5, (right), we show the global feature importance plot for the COVID-19 period with high negative growth rates (Mar–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 crisis), 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.

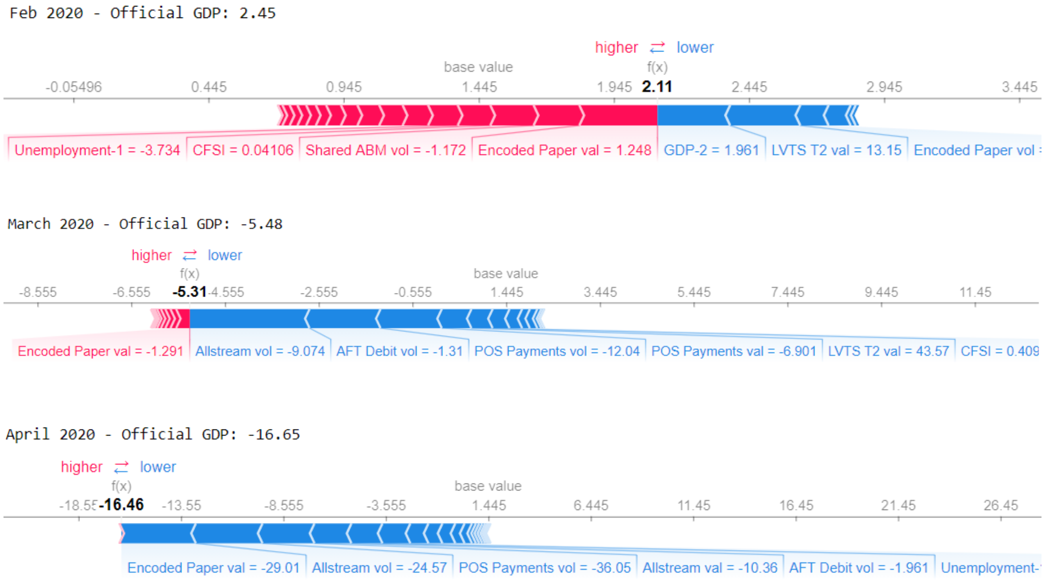

Figure 6. GDP: SHAP force plots showing the feature contribution at each nowcasting instance during the onset of the pandemic, i.e., for Feb 2020 (**top**), Mar 2020 (**middle**), and Apr 2020 (**bottom**). The red arrows are positive Shapley values (contributing positively to GDP growth), and the blue 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 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 those months in our period, for instance, in the top plot in Figure 6) during positive economic growth periods and blue clusters with negative signals during crisis periods, such as the GFC and COVID-19 crisis (most predictors contributing negatively to GDP prediction during those months in our period, for instance, in the bottom plot in Figure 6). 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 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.

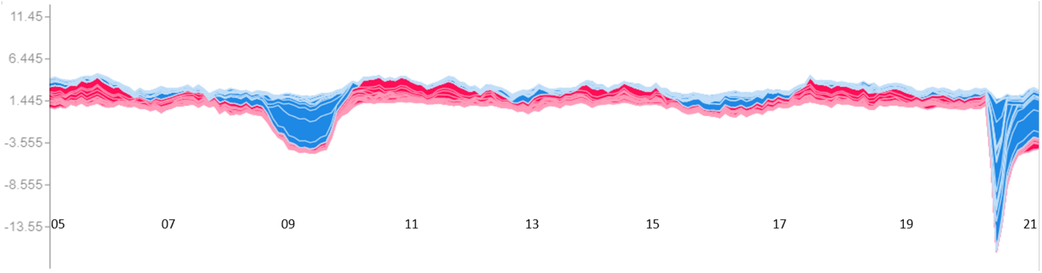

Figure 7. GDP: Clustered force plots for each instance in the training sample, i.e., monthly instance from Mar 2005 to Dec 2020 positioned on the x-axis. Red clusters are positive Shapley values, i.e., the highest number of predictors contributing positively to GDP prediction, therefore pushing the prediction 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.

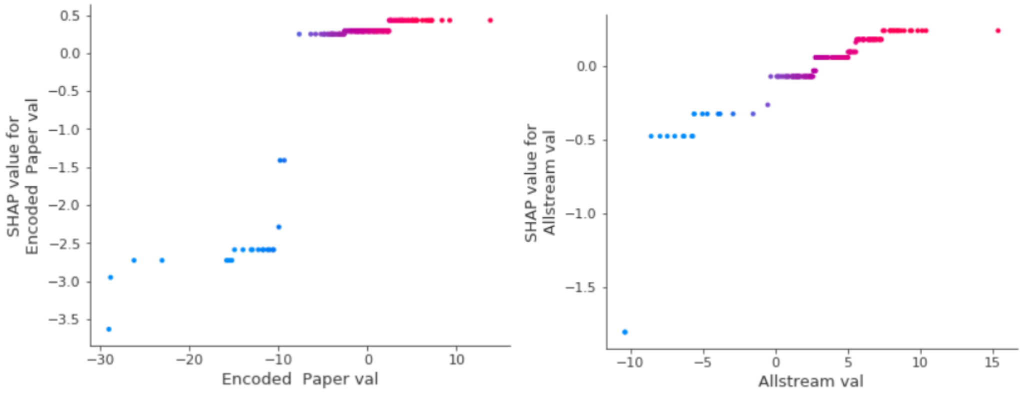

Figure 8. Dependence plots show the Shapley value for each instance in the training sample of a chosen feature on y-axis and the corresponding feature value on x-axis. On the left, we present the dependence plot for the encoded paper (E) value stream, while on the right, we provide the dependence plot for the ACSS Allstream (All) value stream.

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.

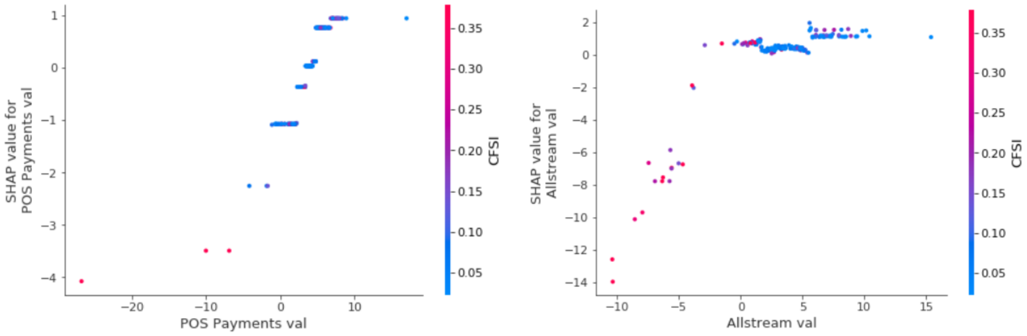

Figure 9. Dependence plots show the Shapley value for each instance in the sample and corresponding predictor value. On the left, we show a dependence plot of RTS for the POS payments value, and on the right, we show the dependence plot of WTS for the ACSS Allstream value.

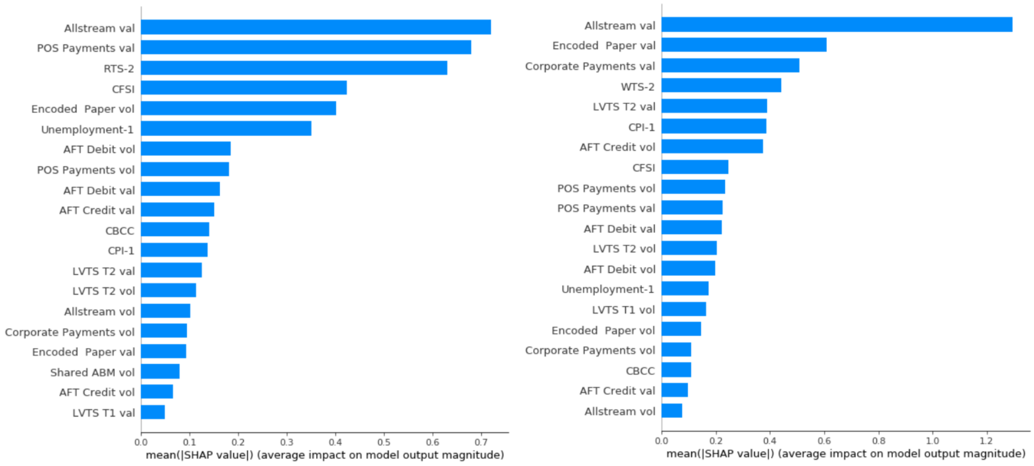

Figure 10. (**Top**) Retail trade sales (RTS) and (**bottom**) wholesale trade sales (WTS). The SHAP global feature importance measured as the mean absolute Shapley value of each instance in the entire training sample (Mar 2005–Dec 2020) at time horizon $t + 1$ using the gradient boosting model.

5. Conclusions

We use comprehensive and timely payment systems data with DFM and ML models for macroeconomic nowcasting. We find the payments data provide information about the economy in real-time and help reduce dependence on variables released with a significant delay. ML provides a set of advanced 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 devise a 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 in nowcasting GDP, RTS, and WTS. The most significant improvements are observed when we use the same month's payments data as the target variables, suggesting the timeliness of the data is driving the prediction gains. Our model's out-of-sample performance is relatively higher during the COVID-19 period compared to the pre-COVID-19 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.

Additionally, the proposed cross-validation technique can help reduce overfitting and improve accuracy in macroeconomic prediction models. 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 crises. 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.

Author Contributions: Both authors are contributed to the conceptualization, methodology, formal analysis, investigation, and writing—original and revised draft. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: This is propriotary data and not possilble to make it public.

Acknowledgments: 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 their excellent assistance on this project. We also thank Alessandro Bitetto, Dalibor Stevanovic, Ingomar Krohn, Jonathan Chiu, Marc Glowka, 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. ASA/Joint Statistical Meeting Nowcasting Session (2023), NBER-Time Series Workshop (2023), 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.

Conflicts of Interest: The Copyright will be hold by the Bank of Canada, not the authors. The authors declare no conflict of interest.

Appendix 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 [49] 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.

Appendix 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 [73]. See Appendix C for more details on model training, tuning, and cross-validation procedures.

Appendix 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 [14,59].

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

$$\min_{\mathbf{w}} \|\mathbf{y} - \hat{\mathbf{y}}(X, \mathbf{w})\|_2^2 + \lambda_1\|\mathbf{w}\|_1 + \lambda_2\|\mathbf{w}\|_2^2, \tag{A1}$$

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 [73].

Appendix 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 $i = 1, 2, \ldots N$) that has at most ε deviation from the actual $\mathbf{y}$ for all the training data. Therefore, the objective function to minimize is

$$\frac{1}{2}\|\mathbf{w}\|_2^2 + C\sum_{i=1}^{N}|\mathbf{y}_i - f(\mathbf{x}_i)|_\varepsilon, \tag{A2}$$

subject to

$$\mathbf{y}_i - f(\mathbf{x}_i) \leq \varepsilon \tag{A3}$$

$$f(\mathbf{x}_i) - \mathbf{y}_i \leq \varepsilon, \tag{A4}$$

where N is the number of training samples and C is a regularization parameter constant [60]. 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 [14,60].

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 $\epsilon = 0.3$. We use the default values for other parameters.

Appendix B.3. Random Forest

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 A1. 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 [61,66].

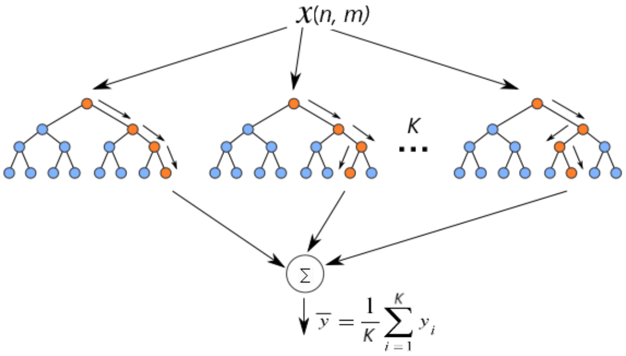

Figure A1. Random forest with K trees using n samples and m features for each tree.

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.

Appendix 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 [62,64].

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

$$\hat{\mathbf{y}}_i = H_p(X_i) = \sum_{1}^{p} h_p(X_i), \tag{A5}$$

where h_p are weak learners, for example, decision trees [64] and p is the number of learners. The model $H_P(X)$ is built as

$$H_p(X) = H_{p-1}(X) + \gamma h_p(X), \tag{A6}$$

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:

$$h_p = {}^{\text{arg}}{}_{\mathbf{p}}{}^{\text{min}} L_p. \tag{A7}$$

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 [73] 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.

Appendix B.5. Feed-Forward Artificial Neural Network

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 A2. In a feed-forward ANN, the data always move forward through the network layers. They start 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 obtain 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 [63,67].

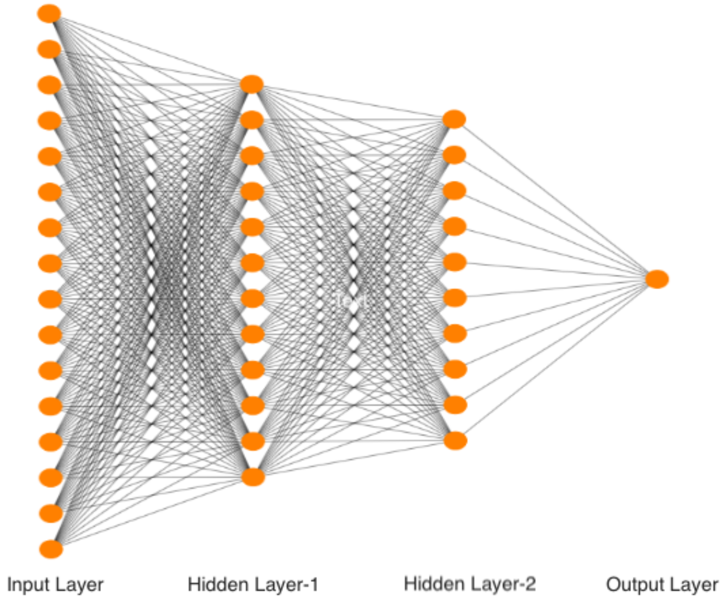

Figure A2. Schematic of densely connected feed-forward neural network with two hidden layers.

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 multilayer 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.

Appendix 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 [53], 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 A1. 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
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–Dec 2018, ($p = 166$) and out-of-sample testing period, Jan 2019–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, the GBR model performed better compared to the 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 3. Further details on these models are provided in Appendix B. Model selection and cross-validation procedures are detailed in Appendices C and D.

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

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

Appendix 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 A3). In our case, the training set is Mar 2005–Dec 2018, and the test set is Jan 2019–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 A3). To include the global financial crisis, we chose 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 perform the following steps:
   (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 4, 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 and 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 obtain the RMSE for the testing set by reusing the standard expanding window approach, as illustrated in Figure 4.

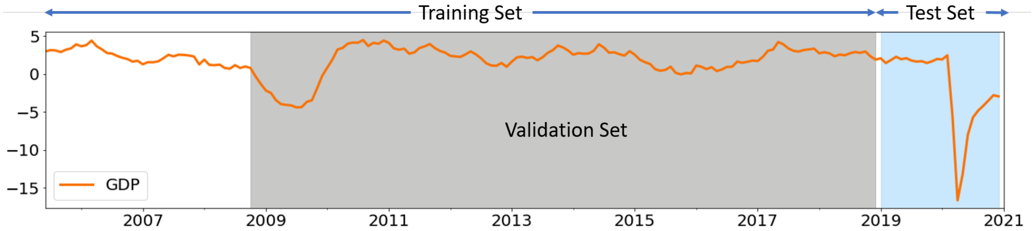

Figure A3. Schematic of data splits for cross-validation. First, the dataset is divided into a training set with a validation subset sampled from the highlighted gray area and a test set (highlighted in blue). The orange line shows the GDP growth rate.

In Figure A4, we present standardized distribution of the target variables (GDP growth rate) for the out-of-sample testing period (Jan 2019–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).

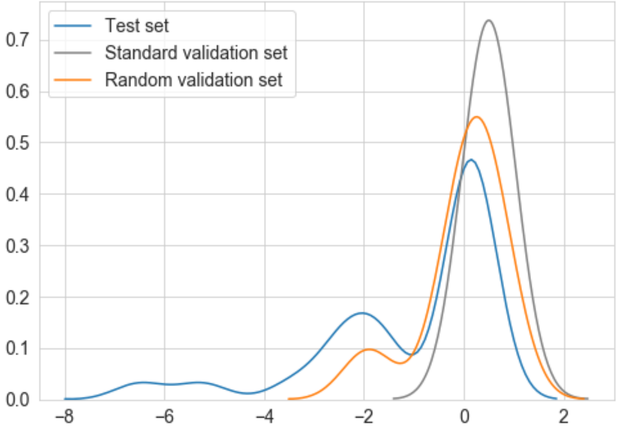

Figure A4. GDP: Distribution of the standardized test set and a typical validation set for standard *k*-fold expanding window approach (standard validation set) and random expanding window approach proposed here (random validation set). The distribution of the proposed random validation set remains similar across all *k*-folds; however, the distribution of the standard validation set could change based on the sample period.

Appendix D. Feature Selection

To select the *k*-best predictors from the set of available attributes, we employ the *SelectKBest* method from scikit-learn [73]. 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 A5, 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–Dec 2020), however, we see opposite behavior 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.

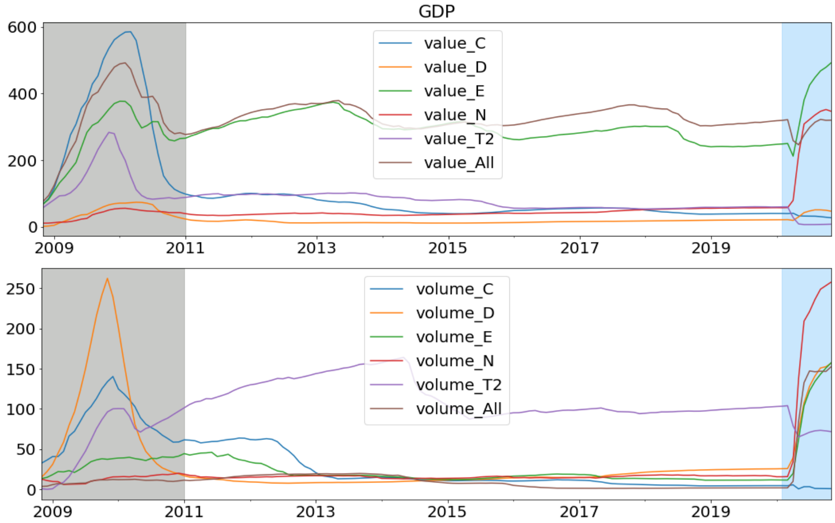

Figure A5. The F-score of a few selected payments streams (values—**top**; volumes—**bottom**) for GDP nowcasting. Higher scores mean a high prediction value. These plots are obtained after each training session of the expanding window approach, ranging from Oct 2008 to Dec 2020. The 2008 GFC period is highlighted in gray; blue shows the COVID-19 period.

Appendix 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 [45,46].

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 [46]:

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

At a high level, the above equation can be split into three parts. The last part of 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 [44] 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 $\mathcal{F}$ as a linear model of coalitions:

$$\mathcal{F}(S) = \phi_0 + \sum_{i=1}^{M} \phi_i S_i, \tag{A8}$$

where $S \in \{0,1\}^M$ is the coalition vector with maximum M coalitions and ϕ_i the Shapley value for ith 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 [44,70]. The procedure is further illustrated as follows:

1. Consider a nowcasting problem with three predictors (Figure A6) 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.

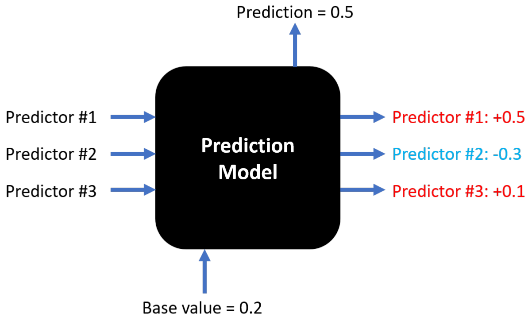

Figure A6. The SHAP explainer provides the marginal contribution of each predictor.

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 obtain 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 [43,44] 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 5) are useful for global model interpretations; the force plots or clustered force plots (Figures 6 and 7) are useful for local interpretation, that is, at each instance of prediction. Furthermore, the dependence plots (Figure 8) could be valuable for understanding the relationships between given predictors and the targets.

The SHAP—while being 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 [70,72]. 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 fewer 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 [44] overcomes some of those challenges to a certain extent [70]. Furthermore, as shown by [72], 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.

Global Feature Importance Comparison

We can also use impurity- or permutation-based global feature importance approaches for tree-based models like GBR and RFR. Among the two, the permutation-based approach is shown to be more useful for nonlinear models [70,74]. In Figure A7, we compare the feature importance of the gradient boosting model trained on the entire training sample (Mar 2005–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.

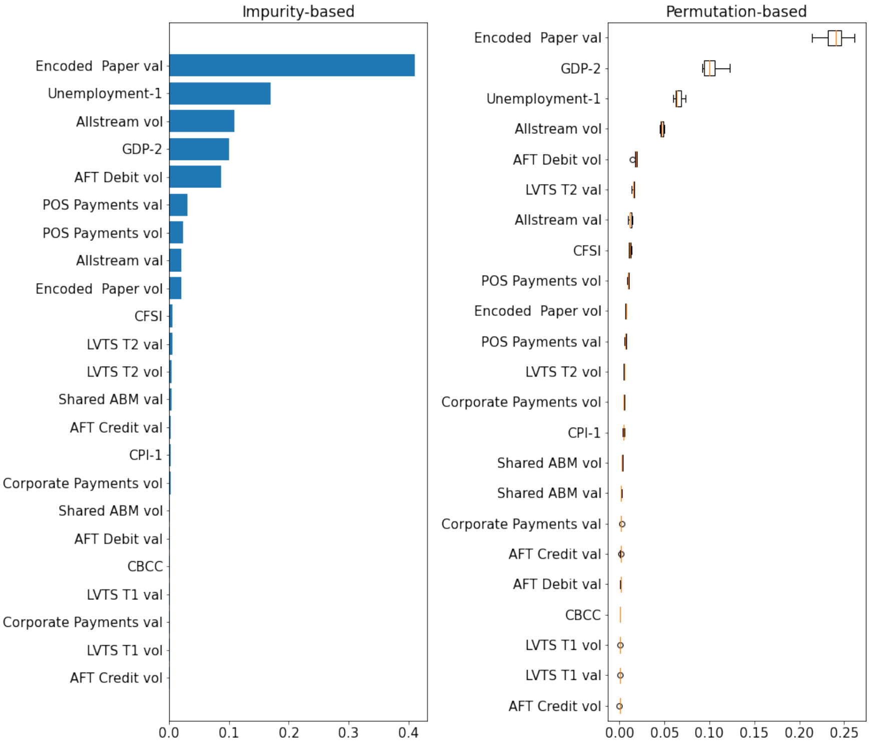

Figure A7. GDP: Global feature importance for the entire training sample (Mar 2005–Dec 2020) at time horizon $t+1$ using the gradient boosting model: (**left**) impurity-based feature importance; (**right**) permutation-based feature importance.

Appendix F. 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–Feb 20) and the COVID-19 period (Mar 20–Oct 20) of the test sample highlighted in gray and blue, respectively, in Figure A8. 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 A2). These results demonstrate the usefulness of payments data during normal periods and crisis periods.

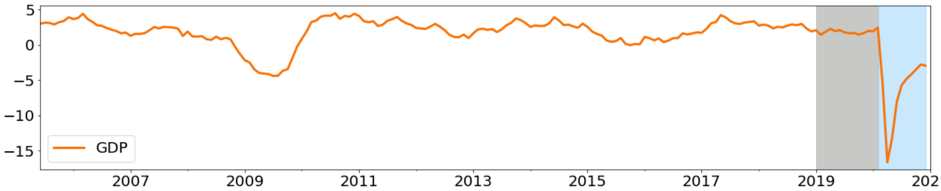

Figure A8. The test sample of GDP nowcasting exercises is divided into two sets: the pre-COVID-19 test set (highlighted in gray) and the COVID-19 test set (highlighted in blue).

Table A2. 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–Dec 2018; out-of-sample testing period, Jan 2019–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–Feb 2020; out-of-sample testing period, Mar 2020–Dec 2020. These numbers show the percentage gain over benchmark cases for the same period.

Appendix G. Nowcasting Performance for First and Latest Vintages

In this section, we compare the GDP nowcasting performance of our model with the real-time vintages (first releases) and the latest vintages (both shown in Figure A9). Comparatively, the models using payments data perform better against the latest vintages (we obtain smaller RMSEs). However, the gains are small (Table A3). 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.

Table A3. 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–Dec 2018, and out-of-sample testing period, Jan 2019–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.

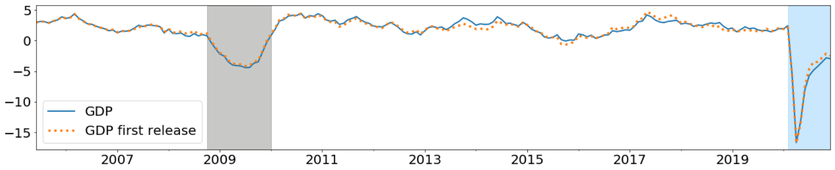

Figure A9. YOY seasonally adjusted GDP growth rates comparison of the first releases with latest releases. The GFC is highlighted in gray and the COVID-19 period is highlighted in blue.

References

1. Giannone, D.; Reichlin, L.; Small, D. Nowcasting: The real-time informational content of macroeconomic data. *J. Monet. Econ.* **2008**, *55*, 665–676. [CrossRef]
2. Angelini, E.; Camba-Mendez, G.; Giannone, D.; Reichlin, L.; Rünstler, G. Short-term forecasts of Euro area GDP growth. *Econom. J.* **2011**, *14*, C25–C44. [CrossRef]
3. Spange, M. Can Crises be Predicted. Danmarks National Monetary Review. 2010. Available online: https://www.nationalbanken.dk/en/publications/Documents/2010/07/can%20crises_2q_2010.pdf (accessed on 6 October 2023).
4. Hamilton, J.D. Calling recessions in real time. *Int. J. Forecast.* **2011**, *27*, 1006–1026. [CrossRef]
5. Choi, H.; Varian, H. Predicting the present with Google Trends. *Econ. Rec.* **2012**, *88*, 2–9. [CrossRef]
6. Buono, D.; Mazzi, G.L.; Kapetanios, G.; Marcellino, M.; Papailias, F. Big data types for macroeconomic nowcasting. *Eurostat Rev. Natl. Acc. Macroecon. Indic.* **2017**, *1*, 93–145.
7. Bok, B.; Caratelli, D.; Giannone, D.; Sbordone, A.M.; Tambalotti, A. Macroeconomic nowcasting and forecasting with big data. *Annu. Rev. Econ.* **2018**, *10*, 615–643. [CrossRef]
8. Kapetanios, G.; Papailias, F. *Big Data & Macroeconomic Nowcasting: Methodological Review*; Technical Report; Discussion Papers ESCoE DP-2018-12; Economic Statistics Centre of Excellence: London, UK, 2018.
9. Galbraith, J.W.; Tkacz, G. Nowcasting with payments system data. *Int. J. Forecast.* **2018**, *34*, 366–376. [CrossRef]
10. Koop, G.; Onorante, L. Macroeconomic Nowcasting Using Google Probabilities. In *Topics in Identification, Limited Dependent Variables, Partial Observability, Experimentation, and Flexible Modeling: Part A*; Emerald Publishing Limited: Bradford, UK, 2019; Volume 40, pp. 17–40.
11. Foroni, C.; Marcellino, M.; Stevanovic, D. Forecasting the COVID-19 recession and recovery: Lessons from the financial crisis. *Int. J. Forecast.* **2022**, *38*, 596–612. [CrossRef]
12. Babii, A.; Ghysels, E.; Striaukas, J. Machine learning time series regressions with an application to nowcasting. *J. Bus. Econ. Stat.* **2021**, *40*, 1094–1106. [CrossRef]
13. Cimadomo, J.; Giannone, D.; Lenza, M.; Monti, F.; Sokol, A. Nowcasting with large Bayesian vector autoregressions. *J. Econom.* **2022**, *231*, 500–519. [CrossRef]
14. Hastie, T.; Tibshirani, R.; Friedman, J. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer Science & Business Media: Dordrecht, The Netherlands, 2009.
15. Ahmed, N.K.; Atiya, A.F.; Gayar, N.E.; El-Shishiny, H. An empirical comparison of machine learning models for time series forecasting. *Econom. Rev.* **2010**, *29*, 594–621. [CrossRef]
16. Athey, S.; Imbens, G.W. Machine Learning Methods That Economists Should Know About. *Annu. Rev. Econ.* **2019**, *11*, 685–725. [CrossRef]
17. Carlsen, M.; Storgaard, P.E. *Dankort Payments as a Timely Indicator of Retail Sales in Denmark*; Technical Report; Danmarks Nationalbank Working Papers 66; Danmarks Nationalbank: Copenhagen, Denmark, 2010. Available online: http://hdl.handle.net/10419/82313 (accessed on 6 October 2023).
18. Barnett, W.; Chauvet, M.; Leiva-Leon, D.; Su, L. *Nowcasting Nominal GDP with the Credit-Card Augmented Divisia Monetary Aggregates*; Technical Report; The Johns Hopkins Institute for Applied Economics: Baltimore, MD, USA, 2016. Available online: https://ideas.repec.org/p/pra/mprapa/73246.html (accessed on 6 October 2023).
19. Duarte, C.; Rodrigues, P.M.; Rua, A. A mixed frequency approach to the forecasting of private consumption with ATM/POS data. *Int. J. Forecast.* **2017**, *33*, 61–75. [CrossRef]
20. Aprigliano, V.; Ardizzi, G.; Monteforte, L.; d'Italia, B. Using the payment system data to forecast the economic activity. *Int. J. Cent. Bank.* **2019**, *15*, 55–80.
21. Galbraith, J.; Tkacz, G. *Electronic Transactions as High-Frequency Indicators of Economic Activity*; Technical Report; Bank of Canada: Ottawa, ON, Canada, 2007. [CrossRef]
22. Paturi, P.; Chiron, C. *Canadian Payments: Methods and Trends 2020*; Technical Report; Payments Canada Report; Payments Canada: Ottawa, ON, Canada, 2020. Available online: https://www.payments.ca/sites/default/files/paymentscanada_canadianpaymentsmethodsandtrendsreport_2020.pdf (accessed on 6 October 2023).
23. Chapman, J.T.; Desai, A. Using Payments Data to Nowcast Macroeconomic Variables During the Onset of COVID-19. *J. Financ. Mark. Infrastructures* **2020**, *9*, 1–29. [CrossRef]
24. Chakraborty, C.; Joseph, A. *Machine Learning at Central Banks*; Technical Report; Bank of England Working Paper No. 674; Elsevier: Amsterdam, The Netherlands, 2017. Available online: https://ssrn.com/abstract=3031796 (accessed on 6 October 2023).
25. Richardson, A.; van Florenstein Mulder, T.; Vehbi, T. Nowcasting GDP using machine-learning algorithms: A real-time assessment. *Int. J. Forecast.* **2021**, *37*, 941–948. [CrossRef]
26. Maehashi, K.; Shintani, M. Macroeconomic forecasting using factor models and machine learning: an application to Japan. *J. Jpn. Int. Econ.* **2020**, *58*, 101104. [CrossRef]
27. Yoon, J. Forecasting of real GDP growth using machine learning models: Gradient boosting and random forest approach. *Comput. Econ.* **2021**, *57*, 247–265. [CrossRef]
28. Gogas, P.; Papadimitriou, T.; Sofianos, E. Forecasting unemployment in the Euro area with machine learning. *J. Forecast.* **2022**, *41*, 551–566. [CrossRef]

29. Vrontos, S.D.; Galakis, J.; Vrontos, I.D. Modeling and predicting US recessions using machine learning techniques. *Int. J. Forecast.* **2021**, *37*, 647–671. [CrossRef]
30. Coulombe, P.G.; Marcellino, M.; Stevanovic, D. Can machine learning catch the COVID-19 recession? *Natl. Inst. Econ. Rev.* **2021**, *256*, 71–109. [CrossRef]
31. Liu, J.; Li, C.; Ouyang, P.; Liu, J.; Wu, C. Interpreting the prediction results of the tree-based gradient boosting models for financial distress prediction with an explainable machine learning approach. *J. Forecast.* **2022**, *42*, 1112–1137. [CrossRef]
32. Mullainathan, S.; Spiess, J. Machine learning: An applied econometric approach. *J. Econ. Perspect.* **2017**, *31*, 87–106. [CrossRef]
33. Athey, S. The impact of machine learning on economics. In *Economics of Artificial Intelligence*; University of Chicago Press: Chicago, IL, USA, 2017. Available online: http://www.nber.org/chapters/c14009 (accessed on 6 October 2023).
34. Duprey, T. *Canadian Financial Stress and Macroeconomic Conditions*; Technical Report; Bank of Canada: Ottawa, ON, Canada, 2020. [CrossRef]
35. Kwan, A.C.; Cotsomitis, J.A. The usefulness of consumer confidence in forecasting household spending in Canada: A national and regional analysis. *Econ. Inq.* **2006**, *44*, 185–197. [CrossRef]
36. Athey, S.; Tibshirani, J.; Wager, S. Generalized random forests. *Ann. Stat.* **2019**, *47*, 1148–1178. [CrossRef]
37. Buckmann, M.; Joseph, A.; Robertson, H. Opening the Black Box: Machine Learning Interpretability and Inference Tools with an Application to Economic Forecasting. In *Data Science for Economics and Finance*; Springer: Cham, Switzerland 2021; pp. 43–63. [CrossRef]
38. Bergmeir, C.; Benítez, J.M. On the use of cross-validation for time series predictor evaluation. *Inf. Sci.* **2012**, *191*, 192–213. [CrossRef]
39. Bergmeir, C.; Hyndman, R.J.; Koo, B. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Comput. Stat. Data Anal.* **2018**, *120*, 70–83. [CrossRef]
40. Chu, C.K.; Marron, J.S. Comparison of two bandwidth selectors with dependent errors. *Ann. Stat.* **1991**, *19*, 1906–1918. [CrossRef]
41. Varian, H.R. Big data: New tricks for econometrics. *J. Econ. Perspect.* **2014**, *28*, 3–28. [CrossRef]
42. Kuhn, M.; Johnson, K. *Applied Predictive Modeling*; Springer: Berlin/Heidelberg, Germany, 2013; Volume 26. Available online: https://vuquangnguyen2016.files.wordpress.com/2018/03/applied-predictive-modeling-max-kuhn-kjell-johnson_1518.pdf (accessed on 6 October 2023).
43. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*; Curran Associates, Inc.: Nice, France, 2017; pp. 4765–4774. Available online: https://arxiv.org/abs/1705.07874 (accessed on 6 October 2023).
44. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From local explanations to global understanding with explainable AI for trees. *Nat. Mach. Intell.* **2020**, *2*, 2522–5839. [CrossRef]
45. Shapley, L.S. A value for n-person games. *Contrib. Theory Games* **1953**, *2*, 307–317.
46. Osborne, M.J.; Rubinstein, A. *A Course in Game Theory*; MIT Press: Cambridge, MA, USA, 1994. Available online: https://arielrubinstein.tau.ac.il/books/GT.pdf (accessed on 6 October 2023).
47. Dahlhaus, T.; Welte, A. *Payment Habits During COVID-19: Evidence from High-Frequency Transaction Data*; Technical Report; Bank of Canada: Ottawa, ON, Canada, 2021. [CrossRef]
48. Desai, A.; Lu, Z.; Rodrigo, H.; Sharples, J.; Tian, P.; Zhang, N. From LVTS to Lynx: Quantitative assessment of payment system transition in Canada. *J. Paym. Strategy Syst.* **2023**, *17*, 291–314.
49. Arjani, N.; McVanel, D. A Primer on Canada's Large Value Transfer System. 2006. Available online: https://www.bankofcanada.ca/wp-content/uploads/2010/05/lvts_neville.pdf (accessed on 6 October 2023).
50. X13 Reference Manual. *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, USA, 2017. Available online: https://www.census.gov/ts/x13as/docX13AS.pdf (accessed on 6 October 2023).
51. Bank of Canada. *Monetary Policy Report—April 2020*; Technical Report; Bank of Canada: Ottawa, ON, Canada, 2020. Available online: https://www.bankofcanada.ca/wp-content/uploads/2020/04/mpr-2020-04-15.pdf (accessed on 6 October 2023).
52. Stock, J.; Watson, M. *Chapter 8—Dynamic Factor Models, Factor-Augmented Vector Autoregressions, and Structural Vector Autoregressions in Macroeconomics*; Elsevier: Amsterdam, The Netherlands, 2016; Volume 2, pp. 415–525. [CrossRef]
53. Chernis, T.; Sekkel, R. A dynamic factor model for nowcasting Canadian GDP growth. *Empir. Econ.* **2017**, *53*, 217–234. [CrossRef]
54. Bańbura, M.; Modugno, M. Maximum likelihood estimation of factor models on datasets with arbitrary pattern of missing data. *J. Appl. Econom.* **2014**, *29*, 133–160. [CrossRef]
55. Bańbura, M.; Giannone, D.; Reichlin, L. *Nowcasting*; Technical Report; ECB Working Paper No. 1275; Elsevier: Amsterdam, The Netherlands, 2010. Available online: https://ssrn.com/abstract=1717887 (accessed on 6 October 2023).
56. Hindrayanto, I.; Koopman, S.J.; de Winter, J. Forecasting and nowcasting economic growth in the Euro area using factor models. *Int. J. Forecast.* **2016**, *32*, 1284–1305. [CrossRef]
57. Bragoli, D. Now-casting the Japanese economy. *Int. J. Forecast.* **2017**, *33*, 390–402. [CrossRef]
58. Coulombe, P.G.; Leroux, M.; Stevanovic, D.; Surprenant, S. How is machine learning useful for macroeconomic forecasting? *arXiv* **2020**, arXiv:2008.12477.
59. Zou, H.; Hastie, T. Regularization and variable selection via the elastic net. *J. R. Stat. Soc. Stat. Methodol.* **2005**, *67*, 301–320. [CrossRef]

60. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
61. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
62. Friedman, J.H. Greedy function approximation: A gradient boosting machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
63. Bengio, Y. Learning deep architectures for AI. *Found. Trends® Mach. Learn.* **2009**, *2*, 1–127. [CrossRef]
64. Friedman, J.; Hastie, T.; Tibshirani, R. *The Elements of Statistical Learning*; Springer Series in Statistics; Springer: New York, NY, USA, 2001; Volume 1. [CrossRef]
65. Burges, C.J. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.* **1998**, *2*, 121–167. [CrossRef]
66. Liaw, A.; Wiener, M. Classification and regression by Random Forest. *R News* **2002**, *2*, 18–22.
67. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: https://www.deeplearningbook.org/ (accessed on 6 October 2023).
68. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. Lightgbm: A highly efficient gradient boosting decision tree. In *Proceedings of the Advances in Neural Information Processing Systems*; MIT Press: Cambridge, MA, USA, 2017; pp. 3146–3154. Available online: https://lightgbm.readthedocs.io/en/latest/ (accessed on 6 October 2023).
69. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
70. Molnar, C. *Interpretable Machine Learning*; Lulu.com:Morrisville USA 2020. Available online: https://christophm.github.io/interpretable-ml-book/ (accessed on 6 October 2023).
71. Alvarez-Melis, D.; Jaakkola, T.S. On the robustness of interpretability methods. *arXiv* **2018**, arXiv:1806.08049.
72. Slack, D.; Hilgard, S.; Jia, E.; Singh, S.; Lakkaraju, H. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society, New York, NY, USA, 7–8 February 2020; pp. 180–186. Available online: https://dl.acm.org/doi/pdf/10.1145/3375627.3375830 (accessed on 6 October 2023).
73. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
74. Breiman, L. Bagging predictors. *Mach. Learn.* **1996**, *24*, 123–140. [CrossRef]

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