International Journal of Forecasting 39 (2023) 1122–1144

Contents lists available at ScienceDirect

International Journal of Forecasting

journal homepage: www.elsevier.com/locate/ijforecast

Mixed-frequency machine learning: Nowcasting and backcasting weekly initial claims with daily internet search volume data☆

Daniel Borup [a,b], David E. Rapach [c,*], Erik Christian Montes Schütte [a,b,d]

[a] Aarhus University, Denmark
[b] CREATES, Denmark
[c] Federal Reserve Bank of Atlanta, United States of America
[d] DFI, Denmark

ARTICLE INFO

Keywords:
Mixed-frequency data
LASSO
Elastic net
Neural network
Unemployment insurance
Internet search
Variable importance

ABSTRACT

We propose an out-of-sample prediction approach that combines unrestricted mixed-data sampling with machine learning (mixed-frequency machine learning, MFML). We use the MFML approach to generate a sequence of nowcasts and backcasts of weekly unemployment insurance initial claims based on a rich trove of daily Google Trends search volume data for terms related to *unemployment*. The predictions are based on linear models estimated via the LASSO and elastic net, nonlinear models based on artificial neural networks, and ensembles of linear and nonlinear models. Nowcasts and backcasts of weekly initial claims based on models that incorporate the information in the daily Google Trends search volume data substantially outperform those based on models that ignore the information. Predictive accuracy increases as the nowcasts and backcasts include more recent daily Google Trends data. The relevance of daily Google Trends data for predicting weekly initial claims is strongly linked to the COVID-19 crisis.

1. Introduction

With the advent of "big data," empirical macroeconomic models increasingly include large numbers of features or predictors. Conventional estimation approaches are usually ill-suited to estimating high-dimensional models, especially with respect to out-of-sample performance. Accordingly, machine-learning techniques – which are designed to guard against overfitting and can allow for nonlinearities – are growing in popularity in macroeconomics (e.g., Borup & Schütte, 2022; Diebold & Shin, 2019; Kotchoni, Leroux, & Stevanovic, 2019; Medeiros, Vasconcelos, Veiga, & Zilberman, 2021).

Based on the mixed-data sampling (MIDAS) approach (Ghysels, Santa-Clara, & Valkanov, 2005), the use of mixed-frequency data has also become popular in macroeconomics (e.g., Brave, Butters, & Justiniano, 2019; Clements

☆ We are grateful to Scott Brave, Marco Colagrossi, Philippe Goulet Coulombe, Francesco D'Amuri, Andrea Geraci, Edward Kong, Michael McCracken, Daniel Prinz, Dalibor Stevanovic, and Nicolas Woloszko, as well as conference and seminar participants at the 2020 Banca d'Italia and Federal Reserve Board Joint Conference on Nontraditional Data and Statistical Learning with Applications to Macroeconomics, 41st International Symposium on Forecasting, Competence Centre on Microeconomic Evaluation Seminar Series at the European Commission Joint Research Centre, Dale T. Mortensen COVID-19 Workshop, 2020 International Conference on Computational and Financial Econometrics, Aarhus University, and Instituto Tecnológico Autónomo de México (ITAM), for helpful comments. We are especially grateful to the Associate Editor and two referees for valuable comments that helped to significantly improve the paper. Borup and Schütte thank the Center for Research in Econometric Analysis of Time Series (CREATES), and Schütte thanks the Danish Finance Institute (DFI), for research support.

* Correspondence to: Research Department, Federal Reserve Bank of Atlanta, 1000 Peachtree Street N.E., Atlanta, GA 30309, United States of America.

E-mail addresses: dborup@econ.au.dk (D. Borup), dave.rapach@gmail.com (D.E. Rapach), christianms@econ.au.dk (E.C.M. Schütte).

https://doi.org/10.1016/j.ijforecast.2022.05.005

& Galvão, 2008; Foroni & Marcellino, 2014). The MIDAS approach allows for more information in higher-frequency data for predicting lower-frequency variables. The MIDAS approach imposes a lag-polynomial structure on the higher-frequency data to reduce the number of parameters that need to be estimated. In contrast, the unrestricted MIDAS (U-MIDAS) specification (Foroni, Marcellino, & Schumacher, 2015) does not impose a lag-polynomial structure, thereby allowing each higher-frequency feature to have its own coefficient. An advantage of the U-MIDAS approach is that it facilitates interpretation, as we separately estimate the effect of each higher-frequency feature on the target. However, this comes at the cost of estimating more coefficients, which is a problem for conventional estimation methods, especially as the number of features observed at the higher frequency increases.

In this paper, we propose combining the U-MIDAS approach with machine learning. The U-MIDAS approach allows each higher-frequency observation for each feature to enter separately into the model, thereby facilitating model interpretation. In settings with many features, the model quickly becomes quite large, so that we harness machine-learning techniques to guard against overfitting and thus improve out-of-sample performance. We call our approach mixed-frequency machine learning (MFML).

MFML can be used for both linear and nonlinear prediction models. For linear specifications, we can employ the least absolute shrinkage and selection operator (LASSO, Tibshirani, 1996) and/or elastic net (ENet, Zou & Hastie, 2005) to estimate high-dimensional models. The LASSO and ENet are popular machine-learning devices that improve prediction in high-dimensional settings by including a penalty term in the objective function for estimating the model's parameters. Intuitively, the penalty term works to shrink the parameters toward zero, thereby helping to prevent overfitting. Because their penalty terms include an ℓ_1 component, the LASSO and ENet permit shrinkage to zero, further facilitating model interpretation by performing variable selection.

We can employ artificial neural networks (ANNs) from machine learning to allow for more complex nonlinear predictive relationships. ANNs contain one or more hidden layers, each of which contains multiple neurons that transmit predictive signals through the network via nonlinear activation functions. Under a reasonable set of restrictions, a single-layer ANN with a sufficient number of neurons can approximate any smooth function (e.g., Barron, 1994; Cybenko, 1989; Funahashi, 1989; Hornik, 1991; Hornik, Stinchcombe, & White, 1989). In practice, ANNs with multiple hidden layers are often used (e.g., Goodfellow, Bengio, & Courville, 2016; Rolnick & Tegmark, 2018).

We illustrate the value of the MFML approach by generating a sequence of nowcasts and backcasts of US weekly unemployment insurance initial claims (UI) based on a rich trove of daily internet search volume data from Google Trends (GT). The recent COVID-19 crisis created economic upheaval in the United States, including historically unprecedented levels of UI in March and April 2020. UI has become perhaps the most closely watched economic variable since the advent of the COVID-19 crisis, as it provides important information about the US labor market and is reported at a weekly frequency.[1] UI furnishes relevant information to managers, financial market participants, and policymakers. We compute nowcasts and backcasts of weekly UI from ten days to one day before its initial release on Thursday by the Department of Labor, thereby providing timely predictions for tracking labor market conditions.

We employ a high-dimensional set of search terms related to *unemployment* that reflects individuals' searches for information about filing for unemployment benefits when they become (or anticipate becoming) unemployed, as we cannot know a priori the most relevant GT search terms for predicting UI. Specifically, we use daily GT search volume data for 88 *unemployment*-related terms for the most recent seven days to generate a sequence of nowcasts and backcasts of a given week's (Sunday through Saturday) UI. This is in anticipation of the figure's release by the Department of Labor on Thursday of the following week. The sequence of nowcasts and backcasts incorporates the most recent daily GT data as they become available, which allows us to investigate the "term structure" of the flow of information with respect to predictive accuracy.

Each of our prediction models relates UI to its first or second lag – in recognition of the serial correlation in UI – as well as seven days of GT data.[2] Each model thus contains $7 \times 88 + 1 = 617$ predictors (or features), so that ours is a high-dimensional setting.[3] We estimate linear models via the LASSO and ENet, as well as ANNs with one to three hidden layers (NN1, NN2, and NN3, respectively).[4] We train the ANNs using the recently developed Adam stochastic gradient descent (SGD) algorithm (Kingma & Ba, 2015). Both aspects of the MFML approach play key roles in our application. The mixed-frequency data allow us to analyze how the flow of daily search volume information affects the accuracy of the sequence of nowcasts and backcasts, while the machine-learning techniques are appropriate for exploiting the information in the high-dimensional set of predictors.

We find that the information in our high-dimensional daily GT search volume data is useful for nowcasting and backcasting weekly UI, even up to ten days before its release date.[5] Specifically, the predictions for models that include daily GT search volume data generate substantial improvements in root mean squared error (RMSE) vis-à-vis an autoregressive (AR) benchmark model. For an out-of-sample period spanning the first week of January 2015 through the last week of December 2020, all of

[1] Reflecting its relevance and timeliness, Lewis, Mertens, and Stock (2020) include UI in their recently developed weekly economic indicator for the United States.

[2] The inclusion of the first or second lag of UI is determined by the timing of the UI release, as explained in Section 4.

[3] Since the target variable has a weekly frequency and the GT data have a daily frequency, there are seven predictors per search term.

[4] ANNs with one or two (three or more) hidden layers are typically referred to as shallow (deep) neural networks.

[5] We focus on seasonally unadjusted UI. The results are similar for seasonally adjusted UI.

the nowcasts and backcasts based on the daily GT data deliver a lower RMSE than the AR benchmark. Many of the reductions in RMSE are quite sizable, reaching as high as 65%. The improved predictive accuracy offered by the daily GT search volume data is dramatic during the advent of the COVID-19 crisis. In addition, the daily GT data largely subsume the relevant information for predicting weekly UI relative to a set of daily macro-financial variables (including the stock market return and its volatility, government and corporate bond market indicators, and newspaper-based measures of uncertainty).

The nowcasts and backcasts based on linear models and ANNs perform well overall, although predictions based on ANNs typically perform better than those based on linear models. We also consider three ensembles for a given nowcast or backcast of UI.[6] The first takes the average of the LASSO and ENet predictions, the second is the average of the predictions generated by the three ANNs, and the last is the average of all five predictions (LASSO, ENet, NN1, NN2, and NN3). The ensemble based on all of the individual predictions works quite well, producing a reduction in RMSE vis-à-vis the AR benchmark that is larger or nearly as large as that of the best individual model (which we cannot know a priori).

Highlighting the usefulness of the MFML approach, the term structure of the information flow reveals a pronounced increase in predictive accuracy as more timely daily GT data are incorporated into the sequence of nowcasts and backcasts. For the first nowcast, which is made ten days before the UI release, the GT data improve the RMSE vis-à-vis the AR benchmark by approximately 25%. For the backcast made three days before the UI release, which incorporates GT data for seven more recently available days and corresponds to full overlap between the seven days of GT search volume data and the UI week, the improvement in RMSE is around 60% on average.

We also test whether the predictive accuracy of the models incorporating the daily GT search volume data relative to that of the AR benchmark varies with social conditions related to the COVID-19 crisis. Using four measures of social conditions (UI, weekly COVID-19 deaths, government response stringency index, and workplace closure stringency sub-index), we find that the relative performance of the models incorporating the daily GT data significantly improves as social conditions worsen. This provides further evidence that the predictive information in the GT data is linked to the COVID-19 crisis.

To better understand how the MFML approach improves out-of-sample prediction, we compute variable-importance measures (Greenwell, Boehmke, & McCarthy, 2018) and accumulated local effect (ALE) plots (Apley & Zhu, 2020) for the individual predictors in the fitted models. The variable-importance measures allow us to see which predictors are the most relevant in the fitted models. Before the COVID-19 crisis, the lag of UI is the most important predictor in nearly all of the fitted models. With the advent of the crisis, the situation changes markedly, as GT search volume data for terms related to the application process for UI benefits (e.g., *how to file for unemployment*, *unemployment office*) become substantially more important. In addition, the fitted models underlying the sequence of nowcasts and backcasts attach greater importance to GT search volume data for the most recent day of available data, demonstrating the usefulness of allowing each day to have its own coefficient, as in the MFML approach. According to ALE plots, the fitted ANNs evince notable nonlinearities, with the effect of GT search volume on UI becoming stronger as the search volume for terms increases in connection with the COVID-19 crisis.

We contribute to an emerging literature that uses internet search volume data to predict labor market variables. For example, D'Amuri and Marcucci (2017) show that GT search volume data for terms including *jobs* improves US unemployment rate prediction. Niesert, Oorschot, Veldhuisen, Brons, and Lange (2020) find that GT search volume data for an array of terms are useful for predicting unemployment rates in a collection of developed countries. Borup and Schütte (2022) use GT search volume data for a large number of terms and machine-learning tools to improve US employment growth prediction. In contrast to these studies, which predict variables available at the monthly frequency, our target is weekly UI – which has become perhaps the most closely watched variable since the COVID-19 crisis – and we use a mixed-frequency approach. Some recent studies use GT data to predict UI during the crisis (Aaronson et al., 2022; Caperna, Colagrossi, Geraci, & Mazzarella, 2022; Goldsmith-Pinkham & Sojourner, 2020; Larson & Sinclair, 2022; Yi, Ning, Chang, & Kou, 2021). We use GT search volume data for a much larger number of terms and employ our MFML approach to analyze the flow of daily information on predictions of weekly UI.[7]

Our MFML approach complements Babii, Ghysels, and Striaukas (2021), who apply the sparse-group LASSO (Simon, Friedman, Hastie, & Tibshirani, 2013) to estimate MIDAS models with large numbers of predictors. In contrast, we use the U-MIDAS approach and consider ANNs to accommodate more general nonlinearities, as well as ensemble predictions that combine multiple MFML models.

The rest of the paper is organized as follows. Section 2 outlines the MFML approach, specifies the prediction models, and describes their estimation. Section 3 describes the data, while Section 4 explains the information flow for the sequence of nowcasts and backcasts. Section 5 reports results for the out-of-sample exercise. Section 6 interprets the fitted models via variable-importance measures and ALE plots. Section 7 concludes.

2. Mixed-frequency machine learning

To fix ideas – and in line with our application – suppose that the target is sampled at a weekly frequency,

[6] Ensembles are often referred to as combinations in the econometrics literature; see Timmermann (2006) for a survey of forecast combination.

[7] Choi and Varian (2012) use GT search volume data for a small number of terms to predict UI through mid-2011.

while the predictors are sampled at the daily frequency.[8] The general prediction model is given by

$$y_t = f^{(j)}\left(y_{t-1}, \boldsymbol{x}_t^{(j)}; \boldsymbol{\theta}^{(j)}\right) + \varepsilon_t^{(j)}, \tag{1}$$

where y_t is the week-t target variable; $f^{(j)}$ is the conditional mean or prediction function;

$$\underbrace{\boldsymbol{x}_t^{(j)}}_{7K_d \times 1} = \left[\begin{array}{cccc} \boldsymbol{x}_{t-j/7}^{d\prime} & \boldsymbol{x}_{t-(j+1)/7}^{d\prime} & \cdots & \boldsymbol{x}_{t-(j+6)/7}^{d\prime} \end{array}\right]'; \tag{2}$$

$\boldsymbol{x}_{t-i/7}^d = \left[\begin{array}{cccc} x_{1,t-i/7}^d & \cdots & x_{K_d,t-i/7}^d \end{array}\right]'$ is a $K_d \times 1$ vector of predictors for the $(7-i)$th day of week t for $i = 0, \ldots, 6$;[9] $\boldsymbol{\theta}^{(j)}$ is a vector of model parameters specific to $f^{(j)}$; and $\varepsilon_t^{(j)}$ is a zero-mean disturbance term.

Eq. (1) includes an AR term to account for potential serial correlation in y_t; if needed, additional lags of y_t can be included. It is straightforward to extend Eq. (1) to include a vector of predictors sampled at the same frequency as y_t. The definition of $\boldsymbol{x}_t^{(j)}$ in Eq. (2) provides flexibility in that it does not require the daily observations to completely overlap with week t. Complete overlap corresponds to $j = 0$, while $j > (<) 0$ corresponds to the inclusion of one or more daily observations from week $t-1$ ($t+1$). We use this flexibility to create a sequence of nowcasts and backcasts for weekly UI based on daily GT data.

We begin with a linear specification for the prediction model:

$$y_t = \alpha^{(j)} + \beta_{\text{AR}}^{(j)} y_{t-1} + \boldsymbol{\beta}_x^{(j)\prime} \boldsymbol{x}_t^{(j)} + \varepsilon_t^{(j)}, \tag{3}$$

where $\boldsymbol{\beta}_x^{(j)}$ is a $7K_d \times 1$ vector of slope coefficients for the daily variables. In terms of Eq. (1), the vector of model parameters is given by

$$\boldsymbol{\theta}^{(j)} = \left[\begin{array}{ccc} \alpha^{(j)} & \beta_{\text{AR}}^{(j)} & \boldsymbol{\beta}_x^{(j)\prime} \end{array}\right]'. \tag{4}$$

Because it allows each of the higher-frequency predictors in $\boldsymbol{x}_t^{(j)}$ to have its own coefficient, Eq. (3) can be viewed as a U-MIDAS model (Foroni et al., 2015). A restricted MIDAS specification imposes a lag-polynomial structure on the daily observations. We use a U-MIDAS approach for two reasons. First, the daily observations only naturally align with a calendar week when $j = 0$ or $j = 7$. Second, we employ machine-learning methods that allow us to flexibly estimate the coefficients – rather than somewhat arbitrarily imposing a lag-polynomial structure – while still guarding against overfitting when K_d is large.[10]

2.1. Penalized regression

The LASSO (Tibshirani, 1996) is a machine-learning device based on penalized regression. It alleviates overfitting by augmenting the objective function for estimating $\boldsymbol{\theta}^{(j)}$ in Eq. (3) with an ℓ_1 penalty term:

$$\underset{\boldsymbol{\theta}^{(j)} \in \mathbb{R}^{7K_d+2}}{\arg\min} \frac{1}{2T} \left\{ \sum_{t=1}^{T} \left[y_t - \left(\alpha^{(j)} + \beta_{\text{AR}}^{(j)} y_{t-1} + \boldsymbol{\beta}_x^{(j)\prime} \boldsymbol{x}_t^{(j)} \right) \right]^2 \right\} + \lambda \|\boldsymbol{\beta}^{(j)}\|_1, \tag{5}$$

where

$$\boldsymbol{\beta}^{(j)} = \left[\begin{array}{cc} \beta_{\text{AR}}^{(j)} & \boldsymbol{\beta}_x^{(j)\prime} \end{array}\right]', \tag{6}$$

T is the number of weekly y_t observations used to fit the model, $\|\cdot\|_1$ is the ℓ_1 norm, and $\lambda \geq 0$ is a regularization hyperparameter for controlling the degree of shrinkage. Unlike the ℓ_2 penalty in ridge regression (Hoerl & Kennard, 1970), the ℓ_1 penalty in Eq. (5) permits shrinkage to zero (for sufficiently large λ), so that the LASSO performs variable selection.

Although the LASSO is effective at selecting relevant predictors in certain environments (e.g., Bickel, Ritov, & Tsybakov, 2009; Meinshausen & Yu, 2009; Zhang & Huang, 2008), it tends to arbitrarily select one predictor from a group of highly correlated predictors. The ENet (Zou & Hastie, 2005) is a refinement of the LASSO, which helps to mitigate this tendency by including both ℓ_1 (LASSO) and ℓ_2 (ridge) components in the penalty term for the objective function:

$$\underset{\boldsymbol{\theta}^{(j)} \in \mathbb{R}^{7K_d+2}}{\arg\min} \frac{1}{2T} \left\{ \sum_{t=1}^{T} \left[y_t - \left(\alpha^{(j)} + \beta_{\text{AR}}^{(j)} y_{t-1} + \boldsymbol{\beta}_x^{(j)\prime} \boldsymbol{x}_t^{(j)} \right) \right]^2 \right\} + \lambda P_\delta\left(\boldsymbol{\beta}^{(j)}\right), \tag{7}$$

where

$$P_\delta\left(\boldsymbol{\beta}^{(j)}\right) = 0.5(1-\delta)\|\boldsymbol{\beta}^{(j)}\|_2^2 + \delta\|\boldsymbol{\beta}^{(j)}\|_1, \tag{8}$$

$\|\cdot\|_2$ is the ℓ_2 norm, and $0 \leq \delta \leq 1$ is a blending hyperparameter for the ℓ_1 and ℓ_2 components of the penalty term.[11] When $\delta = 1$, $P_\delta = \|\boldsymbol{\beta}^{(j)}\|_1$ in Eq. (8), so that the ENet reduces to the LASSO. We follow the recommendation of Hastie and Qian (2016) and set $\delta = 0.5$.[12]

2.2. Artificial neural networks

For modeling $f^{(j)}$ in Eq. (1), we also consider feed-forward ANNs, which permit nonlinearities in the conditional mean and have proven useful for prediction in

[8] It is straightforward to modify the notation for other sampling frequencies.

[9] The vectors of predictors for the days of week t are as follows: Sunday, $\boldsymbol{x}_{t-6/7}^d$; Monday, $\boldsymbol{x}_{t-5/7}^d$; Tuesday, $\boldsymbol{x}_{t-4/7}^d$; Wednesday, $\boldsymbol{x}_{t-3/7}^d$; Thursday, $\boldsymbol{x}_{t-2/7}^d$; Friday, $\boldsymbol{x}_{t-1/7}^d$; Saturday, $\boldsymbol{x}_t^d$.

[10] Foroni et al. (2015) find that a "small" difference in sampling frequency between the higher- and lower-frequency variables (as with weekly and daily data) favors the U-MIDAS over the restricted MIDAS approach.

[11] We also consider the adaptive LASSO (Zou, 2006). Given our high-dimensional setting, as in Medeiros and Mendes (2016), we use LASSO estimates in the first step to construct the weights in the second step. Medeiros and Mendes (2016) show that the adaptive LASSO is model selection consistent and satisfies the oracle property (Fan & Li, 2001) in high-dimensional time-series models. However, the out-of-sample performance of the adaptive LASSO is worse than that of the LASSO and ENet in our application in Section 5.

[12] To better guard against overfitting, we tune the regularization hyperparameter λ for the LASSO and ENet in Eqs. (5) and (7), respectively, via the Bayesian information criterion (BIC, Schwarz, 1978), as suggested by Zou, Hastie, and Tibshirani (2007). We obtain similar results when we tune λ via the extended regularization information criterion (Hui, Warton, & Foster, 2015), which is a modification of the BIC.

numerous domains. An ANN architecture is comprised of multiple layers. The first, the input layer, is the set of predictors, which we denote by $x_1, \ldots, x_{P_0}$. One or more hidden layers follow. Each hidden layer l contains P_l neurons, each of which takes signals from the neurons in the previous hidden layer to generate a subsequent signal:

$$h_m^{(l)} = g\left(w_{m,0}^{(l)} + \sum_{k=1}^{P_{l-1}} w_{m,k}^{(l)} h_k^{(l-1)}\right) \quad \text{for } m = 1, \ldots, P_l;\ l = 1, \ldots, L, \tag{9}$$

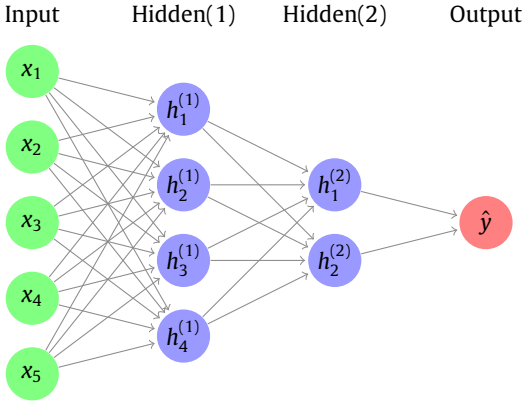
where $h_m^{(l)}$ is the signal corresponding to the mth neuron in the lth hidden layer;[13] $w_{m,0}^{(l)}, w_{m,1}^{(l)}, \ldots, w_{m,P_{l-1}}^{(l)}$ are weights; and $g(\cdot)$ is a (nonlinear) activation function. The final layer is the output layer, which translates the signals from the last hidden layer into a prediction:

$$\hat{y} = w_0^{(L+1)} + \sum_{k=1}^{P_L} w_k^{(L+1)} h_k^{(L)}, \tag{10}$$

where $\hat{y}$ denotes the prediction of the target variable. For the activation function, we use the popular rectified linear unit (ReLU) function:

$$g(x) = \begin{cases} 0 & \text{if } x < 0, \\ x & \text{otherwise.} \end{cases} \tag{11}$$

In response to a sufficiently strong signal, Eq. (11) activates a neuronal connection and relays the signal forward through the network. To illustrate the basic structure of an ANN, the following diagram portrays a feedforward ANN consisting of five inputs and two hidden layers with four and two neurons, respectively:



The myriad of interactions among the inputs and neurons in the network and the nonlinear activation function allow for complex nonlinear predictive relationships.

Theoretically, a single hidden layer with enough nodes is sufficient for approximating any smooth function (e.g., Barron, 1994; Cybenko, 1989; Funahashi, 1989; Hornik, 1991; Hornik et al., 1989). Nevertheless, ANNs with multiple hidden layers are often used in practice (e.g., Goodfellow et al., 2016; Rolnick & Tegmark, 2018). We consider ANNs with one, two, and three hidden layers (NN1, NN2, and NN3, respectively). For our application, we include 89 neurons in the single hidden layer for NN1, corresponding to the dimension of $\boldsymbol{x}_{t-i/7}^d$ ($K_d = 88$) plus the lag of y_t. Based on the "pyramid" strategy (Masters, 1993), NN2 (NN3) contains 89 and ten (89, ten, and three) neurons in its first and second (first, second, and third) hidden layers, respectively.

Fitting (or training) an ANN requires estimating the weights. We train the ANNs by minimizing an objective function based on the MSE for the training sample, which we augment with an ℓ_1 penalty term to help guard against overfitting.[14] We use the recently developed Adam SGD algorithm (Kingma & Ba, 2015) to train the ANNs, which we implement in Python using the keras package. In the algorithm, we set the number of epochs to 700 and batch size to 32 (the keras default).[15]

2.3. Ensembles

We also consider ensemble predictions, which are popular in machine learning. In recognition of model uncertainty, instead of relying on a prediction based on a single model, we take an average of the predictions generated by multiple models. We construct three ensembles. The first (Ensemble-Linear) is an average of the predictions based on the linear models fitted via the LASSO and ENet. The second (Ensemble-ANN) is an average of the predictions based on the fitted NN1, NN2, and NN3 models. The final ensemble (Ensemble-All) is an average of the predictions based on the linear models fitted via the LASSO and ENet and fitted NN1, NN2, and NN3 models.

3. Data

This section describes the data for our application. Our target variable is US weekly UI. We use the MFML approach to generate a sequence of nowcasts and backcasts of weekly UI using daily GT data. Our sample spans the first week of January 2005 (when enough search terms have sufficient volume) to the last week of December 2020.

3.1. Unemployment insurance initial claims

UI is available weekly, corresponding to initial claims for Sunday through Saturday. Each Thursday morning at 8:30 EST, the Department of Labor releases the UI figure for the previous week. We take this publication lag into

[13] For the first hidden layer, $h_k^{(0)} = x_k$ for $k = 1, \ldots, P_0$.

[14] We set the value of the hyperparameter for the ℓ_1 penalty term to 0.001 in the ANN models. The results are very similar for reasonable changes to this hyperparameter or if we use cross-validation to tune it. Since the value is fixed, our results can be considered conservative compared to data-driven (and more computationally intensive) alternatives.

[15] To reduce the influence of starting values for the random-number generator in the SGD algorithm, we compute an ensemble prediction by training a given model ten different times with a different seed each time and taking the average of the ten predictions.

account when computing our predictions.[16] As detailed in Section 4, we are careful in tracking the information flow, so that we only use information available at the time of prediction formation. The choice between targeting seasonally unadjusted or adjusted UI has been the subject of recent debate during the COVID-19 crisis. The issue is whether the conventional multiplicative seasonal-adjustment process overstates the actual seasonality in the data during the crisis when UI reached historically unprecedented levels. Indeed, the Department of Labor recently switched to an additive seasonal adjustment process for UI (Davidson, 2020).[17] Overall, Rinz (2020) argues that seasonally unadjusted UI is more informative during the COVID-19 crisis. In light of these considerations, we focus on predicting seasonally unadjusted UI in this paper; the results are qualitatively similar for seasonally adjusted data (see Table A.1 in the Appendix).

3.2. Google Trends

Daily internet search volume data are obtained from GT, which provides an index of the proportion of queries for a specific search term within a geographical area. The index is released with a maximum delay of 36 hours. The delay results from Google filtering irregular search activity, such as automated searches or queries that may be associated with attempts to spam search.[18]

We construct a high-dimensional set of predictors based on daily GT data for a large number of search terms. Starting with the source term *unemployment*, we use Google Keyword Planner, which provides the most relevant terms to include in a webpage to increase web traffic, to obtain the following top 15 keywords associated with this term: (1) *unemployment*, (2) *unemployment benefits*, (3) *unemployment office*, (4) *unemployment insurance*, (5) *file for unemployment*, (6) *apply for unemployment*, (7) *unemployment claim*, (8) *how to file for unemployment*, (9) *ui online*, (10) *unemployment application*, (11) *unemployment weekly claim*, (12) *unemployment compensation*, (13) *unemployment number*, (14) *unemployment online*, (15) *employment insurance*. These "primitive terms" appear quite plausible, as they are associated with the actions of a person who becomes unemployed. Our out-of-sample period begins in 2015 in Section 5, so that, to avoid look-ahead bias, the set of primitive terms is based on GT data through the end of 2014.

We expand each of the primitive terms via a GT feature that provides a list of 25 related terms (from the "top" category), again based on GT data through the end of 2014. This step adds terms that are specific to individual US states (e.g., *ny unemployment benefits*, *unemployment benefits california*); semantically related to the primitive terms (e.g., *how to apply for unemployment*, *unemployment phone number*, *filing unemployment online*, *state unemployment office*); closely related to unemployment, such as health care coverage and tax policies (e.g., *unemployment health insurance*, *unemployment insurance tax*); and narrowly defined (e.g., *edd online*, which refers to the Employment Development Department, through which unemployment insurance benefits can be applied for in California). After excluding duplicates, this produces a total of 255 keywords.[19]

After removing low-volume queries (defined as series with less than 95% non-zero values), we have $K_d = 88$ unique terms at the daily frequency. In the context of predicting employment growth, Borup and Schütte (2022) find that minor variations in the wording of queries (like adding or removing an *s* for a plural or singular version of a word) can have a notable influence on their predictive power. We cannot know a priori which specific terms or variations are the most relevant for predicting UI, so that we include many related terms and rely on supervised machine-learning methods to place greater weight on those deemed the most relevant. Terms specific to individual US states capture idiosyncrasies for each state, which is useful if, for example, New York is suddenly the main driver of unemployment claims. This is relevant, given the heterogeneity in the spreading of COVID-19 and political responses across different parts of the United States. Table 1 lists the 88 unique GT search terms for our analysis.

GT only allows for downloading daily data in blocks that do not cover the full sample period. Therefore, we concatenate data from each download to construct complete time series. For a given download, GT scales the search volume for each term to have a range of zero to 100, where a value of zero (100) corresponds to the day with the lowest (highest) volume. Thus, we need to adjust the levels of each downloaded block of data to chain together series that are comparable over time. To do so, we download seven-month blocks of data with a month of overlap. We compute the average daily value for each GT term for the current and preceding blocks for the overlapping month. We then use the ratio of the two averages to adjust the levels for all preceding blocks.[20]

Each block of downloaded GT data covering a particular period is based on a randomized sample (about 1%) of total search queries during the period. Thus, the values for the block corresponding to the period change according

[16] Compared to other macroeconomic variables (e.g., gross domestic product and consumption), UI data are subject to relatively minor revisions, as UI is based on government administrative data (rather than surveys). UI is typically revised only once during the following week. We accommodate this by using vintage data when computing the predictions.

[17] The change was only made going forward, so that historical seasonally adjusted data will not be revised.

[18] See "FAQ about Google Trends data."

[19] The order of the words does not matter for computing GT search volume. For example, *unemployment benefits* and *benefits unemployment* are equivalent in that they have the same value for the search volume index.

[20] A simple example illustrates the basic idea. Suppose that the first set of downloaded data for an arbitrary series is 90 and 99 for periods 1 and 2, respectively; the next set of downloaded data is 85 and 76 for periods 2 and 3, respectively. We take the ratio of the values for period 2, $85/99 = 0.86$; we then use the ratio to adjust the period 1 value to $90 \times 0.86 = 77.27$, which gives us a comparable series of 77.27, 85, and 76 for periods 1, 2, and 3, respectively. This is a straightforward procedure to chain together a comparable time series for GT search volume data. Alternative methods are available; for example, Bleher and Dimpfl (2021) use a regression-based algorithm to chain together comparable time series for GT data.

Table 1
Google Trends search terms.

unemployment	*unemployment benefits new york*	*how to file unemployment*
unemployment benefits	*unemployment benefits florida*	*file unemployment claim*
unemployment office	*unemployment florida*	*file for unemployment benefits*
unemployment insurance	*office of unemployment*	*how apply for unemployment*
file for unemployment	*the unemployment office*	*to apply for unemployment*
apply for unemployment	*state unemployment office*	*how to apply for unemployment*
unemployment claim	*unemployment ky*	*apply for unemployment benefits*
how to file for unemployment	*ky unemployment office*	*unemployment florida*
unemployment application	*unemployment ca*	*file unemployment*
unemployment compensation	*ca unemployment office*	*edd online*
unemployment number	*employment office*	*edd*
unemployment online	*employment*	*my ui*
unemployment rate	*unemployment office nj*	*application for unemployment*
unemployment nys	*unemployment nj*	*unemployment iowa*
unemployment ny	*unemployment pa*	*wi unemployment*
unemployment texas	*unemployment il*	*pa unemployment compensation*
unemployment oregon	*unemployment office pa*	*workers compensation*
unemployment california	*unemployment office illinois*	*florida unemployment compensation*
new york unemployment	*unemployment illinois*	*unemployment compensation benefits*
unemployment washington	*state unemployment insurance*	*pennsylvania unemployment compensation*
unemployment wisconsin	*california unemployment insurance*	*ohio unemployment compensation*
unemployment indiana	*unemployment insurance ny*	*pennsylvania unemployment*
unemployment nc	*unemployment health insurance*	*unemployment ohio*
texas unemployment benefits	*unemployment insurance nys*	*federal unemployment*
texas benefits	*unemployment insurance new york*	*unemployment alabama*
ny unemployment benefits	*unemployment insurance tax*	*unemployment phone number*
claim unemployment benefits	*unemployment insurance claim*	*number for unemployment*
unemployment benefits nj	*unemployment insurance nj*	*filing unemployment online*
unemployment insurance benefits	*unemployment insurance office*	
unemployment benefits california	*to file for unemployment*	

Note: The table lists the 88 daily Google Trends search terms used to construct nowcasts and backcasts of weekly unemployment insurance initial claims.

to the time and IP address of the request to download the data. To reduce sampling error, we make ten requests for a particular period and take the average of the values over the ten downloaded blocks.[21]

Fig. 1 depicts UI, along with GT search volumes for two selected terms (*file for unemployment* and *unemployment office*), for the first week of January 2020 to the last week of December 2020.[22] The time stamp on the horizontal axis indicates the UI release. UI exhibits a dramatic increase to 2.9 million for the March 26 release (corresponding to the week ending March 21). This is followed by another sharp increase to 6 million for the April 2 release (corresponding to the week ending March 28). The number again increases to a historical high of 6.2 million for the April 9 release (corresponding to the week ending April 4). UI then generally decreases in Fig. 1, although it remains elevated from a historical perspective.

GT search volumes for the two terms in Fig. 1 appear to track UI well. Specifically, search volumes for the terms start to increase markedly in the weeks around the sharp increase in UI, and they follow the subsequent downward trajectory fairly closely. Fig. 1 suggests that GT search volumes are relevant for predicting UI. This is economically intuitive, as individuals are likely to search for information about filing for unemployment benefits when they become (or anticipate becoming) unemployed. Such searches leave a footprint in the search volume of relevant queries, which we harness to predict UI.

4. Information flow

Table 2 explains the flow of information for generating our sequence of predictions. In terms of notation, we denote the days comprising week t by Sunday_t, Monday_t, ..., Saturday_t. The Department of Labor releases the week-$(t-1)$ UI figure on Thursday_t. Since GT data are released with a maximum 36-hour delay, we employ a conservative two-day lag, so that search volume data for queries

[21] We seasonally adjust the search volume series for each of the terms to remove intra-weekly seasonality using the popular STL filtering procedure (Cleveland, Cleveland, McRae, & Terpenning, 1990). To avoid look-ahead bias, we recursively seasonally adjust the search volume series using data available at the time of prediction formation.

[22] These are two of the most important terms for predicting UI in Section 6.

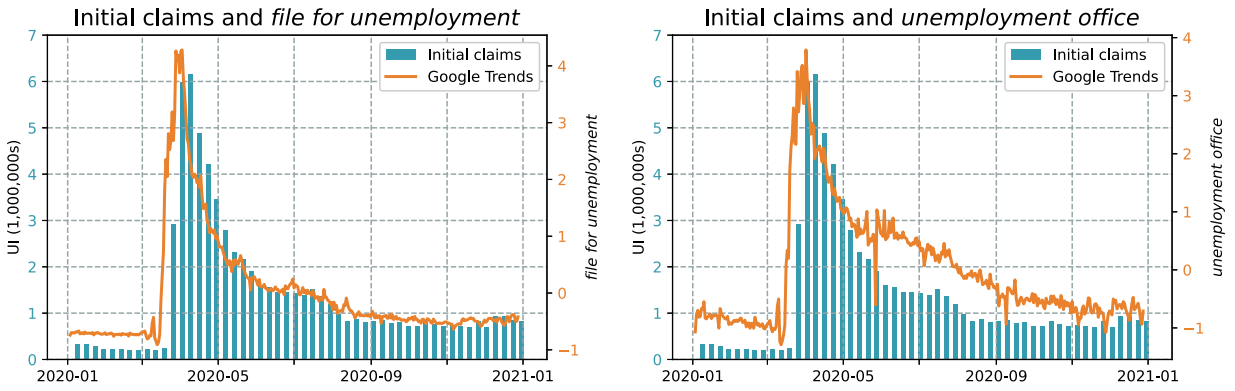


Fig. 1. Weekly initial claims and Google Trends. Notes: The figure depicts weekly unemployment insurance initial claims (UI, left axis) at their release date and (standardized) daily Google Trends search volume for two terms (right axis): *file for unemployment* (left panel) and *unemployment office* (right panel). Google Trends search volume data are for the UI release date. The sample period spans the first week of January 2020 through the last week of December 2020.

Table 2
Information flow.

(1) Prediction formation	(2) Nowcast/ backcast	(3) Google Trends data used for prediction	(4) Data overlap (days)	(5) j	(6) Latest available UI release
Monday_t	Nowcast	Sunday_{t-1} to Saturday_{t-1}	0	7	Week $t-2$
Tuesday_t	Nowcast	Monday_{t-1} to Sunday_t	1	6	Week $t-2$
Wednesday_t	Nowcast	Tuesday_{t-1} to Monday_t	2	5	Week $t-2$
Thursday_t	Nowcast	Wednesday_{t-1} to Tuesday_t	3	4	Week $t-1$
Friday_t	Nowcast	Thursday_{t-1} to Wednesday_t	4	3	Week $t-1$
Saturday_t	Nowcast	Friday_{t-1} to Thursday_t	5	2	Week $t-1$
Sunday_{t+1}	Backcast	Saturday_{t-1} to Friday_t	6	1	Week $t-1$
Monday_{t+1}	Backcast	Sunday_t to Saturday_t	7	0	Week $t-1$
Tuesday_{t+1}	Backcast	Monday_t to Sunday_{t+1}	6	-1	Week $t-1$
Wednesday_{t+1}	Backcast	Tuesday_t to Monday_{t+1}	5	-2	Week $t-1$

Notes: The table reports the information flow for daily Google Trends data and data releases of unemployment insurance initial claims (UI) used for nowcasts and backcasts of week-t UI. The first column provides the prediction-formation day, where the subscript denotes the week when the prediction is made. The second column provides the classification as a nowcast or backcast of week-t UI. The third column provides the range of daily Google Trends data used for prediction. The fourth column provides the number of days of overlap between the daily Google Trends data in the third column and week-t UI. The fifth column provides the value for j in Eq. (1). The sixth column provides the latest release of UI available at the time of prediction formation.

for, say, Saturday_t are available on Monday_{t+1}. We use the seven most recently available daily GT search volume observations for each of the 88 terms when generating each prediction.

We begin with a prediction of week-t UI formed on Monday_t, which corresponds to a nowcast. After accounting for the maximum 36-hour reporting lag, the seven most recently available daily search volume observations for the 88 terms cover Sunday_{t-1} through Saturday_{t-1}. We compute the nowcast by first using historical data available at the time of prediction formation to estimate one of the prediction models in Section 2, which relates UI for a given week to GT search volumes for the 88 terms for the seven days in the previous week and the second lag of UI. The UI lag accounts for the strong autocorrelation in UI. We use the second lag because, as indicated in the last column of Table 2, the most recent UI observation available for computing the nowcast is for the week $t-2$ (due to the reporting lag for UI). We then plug the search volumes for the 88 terms for Sunday_{t-1} through Saturday_{t-1} and most recent UI observation (for week $t-2$) into the fitted model to generate the nowcast of week-t UI.

Next, we compute a prediction of week-t UI formed on Tuesday_t, which again corresponds to a nowcast. Since an additional day of GT data is available, this nowcast is based on search volumes for the 88 terms for Monday_{t-1} to Sunday_t. There is now a one-day overlap between the daily GT and weekly UI data; see the fourth column of Table 2. To compute the nowcast, we first fit the prediction model, which relates UI in a given week to GT search volumes for the 88 terms for Sunday of that week and Monday through Saturday of the previous week (as well as the second lag of UI). We then plug the search volumes for the 88 terms for Monday_{t-1} to Sunday_t (and week-$(t-2)$ UI) into the fitted model. We proceed analogously to compute the nowcast of week-t UI formed on

Wednesday$_t$, which is characterized by a two-day overlap between the daily GT and weekly UI data.

The next three nowcasts in Table 2 are formed on Thursday$_t$, Friday$_t$, and Saturday$_t$. In addition to incorporating GT data through Tuesday$_t$, Wednesday$_t$, and Thursday$_t$, respectively, the latest available UI release allows us to use UI for week $t-1$. We thus use the first (instead of the second) lag of UI in the prediction model. Observe that as we move from Thursday$_t$ to Saturday$_t$ when forming the nowcasts, we go from a three- to a five-day overlap between the available daily GT and weekly UI data. Otherwise, we compute the nowcasts in the same manner as the first three nowcasts in Table 2.

The remaining predictions of week-t UI, formed on Sunday$_{t+1}$ through Wednesday$_{t+1}$, constitute backcasts. The backcast formed on Monday$_{t+1}$ employs the maximum overlap of seven days between the available daily GT and weekly UI data. The backcast formed on Tuesday$_{t+1}$ (Wednesday$_{t+1}$) uses GT data for six (five) days from week t and one (two) day(s) from week $t+1$ to predict week-t UI.

The sequence of predictions in Table 2 allows us to investigate the term structure of the information flow for predicting UI. As we proceed from the nowcast formed on Monday$_t$ to the backcast formed on Monday$_{t+1}$, the degree of overlap between the days used to predict UI increases. For the final two backcasts in Table 2, we include GT search volume data for the 88 terms for the first one or two days of week $t+1$ when predicting week-t UI. We are interested in how the availability of more recent daily GT data affects the accuracy of the nowcasts and backcasts.

5. Out-of-sample results

We generate simulated out-of-sample nowcasts and backcasts of UI for the first week of January 2015 through the last week of December 2020. Mimicking the situation of a forecaster in real time, we proceed as follows using a rolling-window approach. We first use UI and GT data available for the first week of January 2005 through the last week of December 2014 to fit the prediction models. We then plug the most recent UI and relevant GT search volume values into the fitted models to generate the nowcasts and backcasts for UI for the first week of January 2015. Next, we refit the prediction models using UI and GT data available for the second week of January 2005 through the first week of January 2015. We plug the most recent UI and relevant GT search volume values into the fitted models to compute the UI nowcasts and backcasts for the second week of January 2015. We continue in this manner through the end of the out-of-sample period. We reiterate that the nowcasts and backcasts only use information available at the time of prediction formation, as described in Section 4.

Although it substantially increases computational cost, it is important to refit the models each week as new data become available. In particular, as we discuss in Section 6, GT search volumes play a more limited role in models fitted using data that exclude the COVID-19 crisis, while they become substantively more important when the training sample includes data from the crisis. By refitting the models each week, we can capture the growing importance of GT search volumes in a more timely manner.[23]

An AR model based on the first or second lag of UI is used to generate the benchmark nowcasts and backcasts, where we again fit the AR models using a rolling-window approach.[24] An AR model is a standard benchmark in the macroeconomic forecasting literature. Due to the significant autocorrelation in UI, it is a relevant benchmark in our application, and it performs substantially better than a naïve model that ignores the autocorrelation in UI and simply uses the rolling mean to predict UI. Using a seasonal AR model that includes monthly indicator variables leads to slightly worse out-of-sample performance when compared to the AR model that excludes the monthly indicator variables (see Table A.2 in the Appendix). This is primarily driven by a deterioration in forecasting performance during the early weeks of the COVID-19 pandemic, again stressing that seasonality is difficult to model in periods of abrupt changes.[25]

5.1. Predictive accuracy

Table 3 reports out-of-sample results for the individual linear models and ANNs.[26] The third column reports the RMSE for the AR benchmark for each of the nowcasts and backcasts. The MSE is defined as

$$\text{MSE} = \frac{1}{T_{\text{OS}}} \sum_{t=1}^{T_{\text{OS}}} \left(y_t - \hat{y}_t\right)^2, \tag{12}$$

where y_t and $\hat{y}_t$ denote the realized and predicted values of week-t UI, respectively, and T_{OS} is the number of out-of-sample observations available for analyzing the predictions; the RMSE is the square root of Eq. (12).

For the nowcasts formed on Monday$_t$ through Wednesday$_t$, the RMSE for the AR benchmark in the third column is 551,450. Beginning with the nowcast on Thursday$_t$, the AR model is based on the first (instead of the second) lag of UI, since the week-$(t-1)$ UI figure becomes available on Thursday$_t$. This leads to a substantial reduction in RMSE to 345,001.

The fourth through eighth columns of Table 3 report the RMSE ratio for the model in the column heading vis-à-vis the AR benchmark. The ratios for the individual

[23] Using a desktop computer with an Intel Core i9 10940X processor (14 cores running at 4.2 GHz) with 64 GB of DDR4 RAM, it takes just under ten minutes to generate the complete set of nowcasts and backcasts for a given week.

[24] Based on the information flow (see Section 4), the AR benchmark model is given by $y_t = \rho_0 + \rho_2 y_{t-2} + \varepsilon_t$ for the nowcasts formed on Monday$_t$ through Wendesday$_t$; it is given by $y_t = \rho_0 + \rho_1 y_{t-1} + \varepsilon_t$ for the nowcasts and backcasts formed on Thursday$_t$ through Wendesday$_{t+1}$. We estimate the AR models via ordinary least squares.

[25] The seasonal AR model that we use implies additive seasonal adjustment, which is consistent with the changes implemented by the Department of Labor in 2020. We also consider an AR model that allows for additional lags as indicated by the BIC, but this model underperforms the AR benchmark based on the first available lag of UI.

[26] We assess the accuracy of the nowcasts and backcasts using revised UI data.

Table 3
RMSE ratios for individual models.

(1)	(2)	(3)	(4) Linear	(5)	(6) ANN	(7)	(8)
Prediction formation	Nowcast/ backcast	AR RMSE	LASSO	ENet	NN1	NN2	NN3
Monday_t	Nowcast	551,450	0.767	0.767	0.752	**0.748**	0.763
Tuesday_t	Nowcast	551,450	0.686	0.692	0.708	0.712	**0.681**
Wednesday_t	Nowcast	551,450	0.634	0.622	0.728	0.639	**0.619**
Thursday_t	Nowcast	345,001	0.678	0.651	0.769	**0.620**	0.628
Friday_t	Nowcast	345,001	0.587	**0.575**	0.694	0.658	0.579
Saturday_t	Nowcast	345,001	0.531	0.526	0.573	0.580	**0.461**
Sunday_{t+1}	Backcast	345,001	0.504	0.472	0.487	0.494	**0.470**
Monday_{t+1}	Backcast	345,001	0.443	0.433	0.428	**0.408**	0.503
Tuesday_{t+1}	Backcast	345,001	0.443	0.463	0.390	**0.349**	0.455
Wednesday_{t+1}	Backcast	345,001	0.494	0.506	0.423	**0.364**	0.435

Notes: The table reports out-of-sample results for nowcasts and backcasts of week-t unemployment insurance initial claims (UI) formed on the day indicated in the first column, where the subscript denotes the week when the prediction is made. The second column provides the classification as a nowcast or backcast of week-t UI. The third column reports the root mean squared error (RMSE) for an autoregressive (AR) benchmark model. The fourth through eighth columns report the RMSE ratio for the competing model in the column heading vis-à-vis the AR benchmark. The competing models incorporate the information in daily Google Trends search volumes for terms related to *unemployment*. The fourth and fifth columns are for linear models fitted via the LASSO and elastic net (ENet), respectively. The sixth through eighth columns are for fitted artificial neural networks (ANNs) with one (NN1), two (NN2), and three (NN3) hidden layers, respectively. The out-of-sample period begins in the first week of January 2015 and ends in the last week of December 2020. Bold indicates the prediction with the lowest RMSE for the row.

models are all below one. Thus, the nowcasts and backcasts based on the machine-learning methods – which incorporate the information in daily GT search volume data – always outperform the AR benchmark in terms of RMSE. The improvements in predictive accuracy are typically sizable. For the initial nowcast formed on Monday_t, which is formed ten days before the UI release, the reductions in RMSE relative to the AR benchmark are approximately 25%. The RMSE ratios decrease close to monotonically as we move from the nowcast formed on Monday_t to the backcast formed on Monday_{t+1}, which uses the largest data overlap (seven days) between the daily GT and weekly UI data. For the Monday_{t+1} backcast, the reductions in RMSE vis-à-vis the AR benchmark range from around 50% to 60%. Relative to the Monday_{t+1} backcast, the RMSE ratios typically increase for the backcasts formed on Tuesday_{t+1} and Wednesday_{t+1} for the linear models. They continue to decrease on occasion for the ANNs, so that incorporating daily GT search volumes from week $t+1$ to predict week-t UI further improves accuracy in some cases. In sum, the predictions generally become more accurate in Table 3 as we include additional days of GT data for the Monday_t nowcast through the Wednesday_{t+1} backcast, with the reduction in RMSE vis-à-vis the AR benchmark reaching as high as 65% (for the NN2 backcast formed on Tuesday_{t+1}).

The different machine-learning methods in Table 3 perform reasonably similarly, although the most accurate predictions are nearly always based on ANNs, so that accommodating nonlinearities appears important. NN2 (NN3) produces the most accurate prediction in five (four) cases. The only case where a linear model provides the most accurate prediction is the ENet for the Friday_t nowcast.

Table 4 reports results for the ensemble predictions. Among the ensembles, Ensemble-Linear, Ensemble-ANN, and Ensemble-All perform the best for two, three, and five of the ten cases, respectively. Note that Ensemble-All – which is an average of the LASSO, ENet, NN1, NN2, and NN3 predictions – is nearly as accurate or more accurate than the best performing individual nowcasts and backcasts in Table 3. Because we cannot know a priori the best method, the Ensemble-All approach provides a promising practical strategy for nowcasting and backcasting UI.

To further assess the information content in the nowcasts and backcasts that incorporate the daily GT search volume data, we consider an augmented benchmark that includes daily data for ten macro-financial variables as predictors. We consider the following ten variables: S&P 500 return; VIX (Chicago Board Options Exchange volatility index); gold return; TED spread (difference between the three-month LIBOR and three-month Treasury bill yields); term spread (difference between the ten-year Treasury bond and three-month Treasury bill yields); expected five-year inflation rate (the rate at which the five-year Treasury note and five-year TIPS achieve the same yield); default spread (difference between Baa- and Aaa-rated corporate bond yields); Wilshire US real estate investment trust return; Baker, Bloom, and Davis (2016) newspaper-based economic policy uncertainty index; Baker et al. (2020) newspaper-based infectious disease equity market volatility tracker. The S&P 500 return is from Global Financial Data, while the remaining variables are from Federal Reserve Economic Data. All variables are available at the daily frequency with no publication lag.

The daily macro-financial predictors are treated analogously to the daily GT data in the prediction models.

Table 4
RMSE ratios for ensembles.

(1) Prediction formation	(2) Nowcast/ backcast	(3) AR RMSE	(4) Ensemble-Linear	(5) Ensemble-ANN	(6) Ensemble-All
Monday$_t$	Nowcast	551,450	0.766	0.749	**0.748**
Tuesday$_t$	Nowcast	551,450	0.689	0.698	**0.685**
Wednesday$_t$	Nowcast	551,450	0.626	0.644	**0.621**
Thursday$_t$	Nowcast	345,001	**0.660**	0.688	0.666
Friday$_t$	Nowcast	345,001	**0.579**	0.627	0.589
Saturday$_t$	Nowcast	345,001	0.526	0.509	**0.487**
Sunday$_{t+1}$	Backcast	345,001	0.486	0.451	**0.426**
Monday$_{t+1}$	Backcast	345,001	0.437	**0.388**	0.389
Tuesday$_{t+1}$	Backcast	345,001	0.451	**0.373**	0.395
Wednesday$_{t+1}$	Backcast	345,001	0.498	**0.393**	0.430

Notes: The table reports out-of-sample results for nowcasts and backcasts of week-t unemployment insurance initial claims (UI) formed on the day indicated in the first column, where the subscript denotes the week when the prediction is made. The second column provides the classification as a nowcast or backcast of week-t UI. The third column reports the root mean squared error (RMSE) for an autoregressive (AR) benchmark model. The fourth through sixth columns report the RMSE ratio for the ensemble prediction in the column heading vis-à-vis the AR benchmark. The ensembles are based on models that incorporate the information in daily Google Trends search volumes for terms related to *unemployment*. Ensemble-Linear is an average of predictions for linear models fitted via the LASSO and elastic net. Ensemble-ANN is an average of predictions for fitted artificial neural networks (ANNs) with one, two, and three hidden layers. Ensemble-All is an average of predictions for the linear models and ANNs. The out-of-sample period begins in the first week of January 2015 and ends in the last week of December 2020. Bold indicates the prediction with the lowest RMSE for the row.

Table 5
RMSE ratios with the macro-financial benchmark.

(1)	(2)	(3) Linear		(4)	(5) ANN		(6)	(7) Ensemble		
Prediction formation	Nowcast/ backcast	LASSO	ENet	NN1	NN2	NN3	Linear	ANN	All	
Monday$_t$	Nowcast	0.967	0.983	0.988	1.115	1.069	0.974	1.057	1.025	
Tuesday$_t$	Nowcast	0.870	0.892	0.978	0.997	0.895	0.882	0.998	0.964	
Wednesday$_t$	Nowcast	0.816	0.809	0.992	0.890	0.788	0.813	0.853	0.834	
Thursday$_t$	Nowcast	0.799	0.795	0.871	0.875	0.734	0.795	0.827	0.871	
Friday$_t$	Nowcast	0.708	0.700	0.876	0.895	0.716	0.704	0.870	0.871	
Saturday$_t$	Nowcast	0.653	0.685	0.729	0.783	0.620	0.667	0.683	0.822	
Sunday$_{t+1}$	Backcast	0.624	0.623	0.634	0.758	0.559	0.626	0.610	0.681	
Monday$_{t+1}$	Backcast	0.552	0.570	0.609	0.648	0.650	0.561	0.567	0.579	
Tuesday$_{t+1}$	Backcast	0.601	0.641	0.499	0.539	0.569	0.621	0.511	0.573	
Wednesday$_{t+1}$	Backcast	0.673	0.719	0.633	0.549	0.631	0.695	0.597	0.651	

Notes: The table reports out-of-sample results for nowcasts and backcasts of week-t unemployment insurance initial claims (UI) formed on the day indicated in the first column, where the subscript denotes the week when the prediction is made. The second column provides the classification as a nowcast or backcast of week-t UI. The third through tenth columns report the ratio of the root mean squared error (RMSE) for the competing model in the column heading that incorporates the information in daily Google Trends search volumes for terms related to *unemployment* to the RMSE for a benchmark model with the same specification using a set of daily macro-financial predictors. The third and fourth columns are for linear models fitted via the LASSO and elastic net (ENet), respectively. The fifth through seventh columns are for fitted artificial neural networks (ANNs) with one (NN1), two (NN2), and three (NN3) hidden layers, respectively. The eighth through tenth columns are for ensemble predictions. Linear is an average of predictions for linear models fitted via the LASSO and ENet. ANN is an average of predictions for fitted ANNs with one, two, and three hidden layers. All is an average of predictions for the linear models and ANNs. The out-of-sample period begins in the first week of January 2015 and ends in the last week of December 2020.

For the benchmark that includes the macro-financial variables, the length of $\boldsymbol{x}^d_{t-i/7}$ in Eq. (2) becomes $K_d = 10$, so that the total number of predictors (after including the lag of UI) is $7 \times 10 + 1 = 71$. Table 5 reports RMSE ratios for prediction models based on the GT search volumes vis-à-vis models based on the macro-financial variables.

Of the 80 RMSE ratios in Table 5, only four are greater than one, and these four only occur for the initial Monday$_t$ nowcast (NN2, NN3, Ensemble-ANN, Ensemble-All). Thus, for the vast majority of cases, models based on the daily GT search volumes outperform those based on the daily macro-financial variables in terms of RMSE. Similarly to Table 3, the RMSE ratios decrease close to monotonically as we move from the Monday$_t$ nowcast to the Monday$_{t+1}$ backcast. The improvements in RMSE are often substantial in Table 5. For example, they range from 35% to 45% for the Monday$_{t+1}$ backcast, and the improvement reaches as high as 50% (for the Tuesday$_{t+1}$ backcast based on

NN1). Overall, the information in daily GT search volumes appears more relevant for predicting weekly UI than that in daily macro-financial variables.[27]

5.2. COVID-19 crisis

Fig. 2 depicts selected nowcasts and backcasts and realized UI values for the first week of March 2020 through the third week of April 2020, near the advent of the COVID-19 crisis. According to the upper-left panel, the AR benchmark model has substantial difficulty tracking UI at the start of the crisis. Not surprisingly, the Monday$_t$ nowcast (which is based on the second lag of UI) anticipates very little of the sharp increases in the third and fourth weeks of March. Although the prediction for the second week of April is quite accurate, that for the third week of April massively overstates the actual value. The Monday$_{t+1}$ backcast (which is based on the first lag of UI) anticipates little of the sharp increase in the third week of March and still significantly understates the increase in the next week. The backcast then significantly overstates UI in the first two weeks of April, while the prediction is much more accurate for the third week of April.[28]

The models that incorporate information from GT search volumes are typically much more accurate than the AR benchmark in Fig. 2. The upper-right panel shows the Monday$_t$, Tuesday$_t$, and Thursday$_t$ nowcasts and Monday$_{t+1}$ backcast for linear models fitted via the LASSO. Apart from the second week of April, the Monday$_t$ and Tuesday$_t$ nowcasts perform much better than the corresponding AR benchmark (Monday$_t$ nowcast) in the upper-left panel. The inclusion of more timely GT data as they become available improves the performance of the models. The Thursday$_t$ nowcast and Monday$_{t+1}$ backcast in the upper-right panel are typically substantially more accurate than the corresponding AR benchmark (Monday$_{t+1}$ backcast) in the upper-left panel. The results for the predictions based on the fitted NN2 models in the lower-left panel follow a similar pattern, although the Monday$_t$ and Tuesday$_t$ nowcasts evince larger errors in the second and third weeks of April vis-à-vis those for the LASSO in the upper-right panel.

The Monday$_{t+1}$ backcast based on the fitted NN2 models in the lower-left panel displays the best overall performance in Fig. 2. Recall that the daily GT data have the maximum overlap with weekly UI data (seven days) for the Monday$_{t+1}$ backcast. With the seven-day overlap, the fitted NN2 models track weekly UI quite well during the extremely challenging predictive environment created by the advent of the COVID-19 crisis. The Monday$_{t+1}$ backcast based on the Ensemble-All also performs well in the lower-right panel of Fig. 2. Since we cannot know a priori the best individual model, this further highlights the practical benefits of the Ensemble-All approach.

Compared to the other predictions in Fig. 2, the Monday$_{t+1}$ backcast based on the fitted NN2 model in the lower-left panel is better able to predict the spike in UI in the third week of March. This is because GT search volumes play a relatively more important role in the fitted NN2 model for the Monday$_{t+1}$ backcast when it is trained using data before the start of the COVID-19 crisis. The sharp increases in search volumes for many GT terms near the beginning of the crisis thus translate into a comparatively larger UI prediction for the third week of March for the Monday$_{t+1}$ backcast in the lower-left panel. When the training sample includes the UI observation for the third week of March, GT search volumes become much more important in general in the fitted models, which substantially improves the accuracy of the predictions. This highlights the value of retraining the models each week in our rolling-window estimation procedure. Section 6 analyzes the importance of GT search volumes in models fitted using samples that exclude and include data from the crisis.

Next, we investigate whether the relative performance of competing models vis-à-vis the AR benchmark varies systematically with measures of social conditions related to the COVID-19 crisis. Suppose that we want to test whether the difference between MSEs for the AR benchmark and a competing model changes with a measure of social conditions, denoted by q_t. Following Diebold and Mariano (1995) and West (1996), we first define the period-t loss differential for prediction model j:

$$d_t^{(j)} = \left(y_t - \hat{y}_t^{\mathrm{AR}}\right)^2 - \left(y_t - \hat{y}_t^{(j)}\right)^2 \quad \text{for } t = 1, \ldots, T_{\mathrm{OS}}, \quad (13)$$

where $\hat{y}_t^{\mathrm{AR}}$ and $\hat{y}_t^{(j)}$ are the predictions based on the AR benchmark and model j, respectively. We then use the following regression to test whether the difference between MSEs for the benchmark and model j varies with q_t:

$$d_t^{(j)} = \phi_0^{(j)} + \phi_1^{(j)} q_t + e_t^{(j)}, \quad (14)$$

where $e_t^{(j)}$ is a zero-mean disturbance term. Because we estimate the models using a rolling window, computing a Wald statistic for testing the null hypothesis that $\phi_0^{(j)} = \phi_1^{(j)} = 0$ in Eq. (14) is equivalent to the Giacomini and White (2006) test for conditional predictive ability (Giacomini, 2011). If we reject the null hypothesis, then we have evidence of a significant difference in conditional predictive ability between the AR benchmark and model j. To glean further insight into how the predictive ability varies with q_t, we also estimate $\phi_1^{(j)}$ via ordinary least squares (OLS) and test the null hypothesis that $\phi_1^{(j)} = 0$. Under the null, the difference in MSE does not vary with q_t; alternatively, if, say, $\phi_1^{(j)} > 0$, then the MSE difference increases as q_t increases (i.e., the AR benchmark becomes less accurate relative to model j as q_t increases).[29]

We use the following four variables for q_t in Eq. (14):

[27] We also consider models that include both the daily GT search volumes and macro-financial variables as predictors. Overall, the models with both types of variables perform similarly to those that include the GT search volumes alone. In some cases, predictive accuracy deteriorates substantively for the models with both the GT search volumes and macro-financial predictors relative to those with the GT search volumes alone. This provides further evidence that the information in daily GT search volumes subsumes that in daily macro-financial variables.

[28] In the upper-left panel of Fig. 2, the Monday$_t$ nowcast is the same as the Tuesday$_t$ and Wednesday$_t$ nowcasts, while the Monday$_{t+1}$ backcast is the same as the Thursday$_t$ through Saturday$_t$ nowcasts and Sunday$_{t+1}$, Tuesday$_{t+1}$, and Wednesday$_{t+1}$ backcasts.

[29] Following the recommendation of Giacomini (2011), we use a heteroskedasticity-robust covariance matrix estimator to compute the

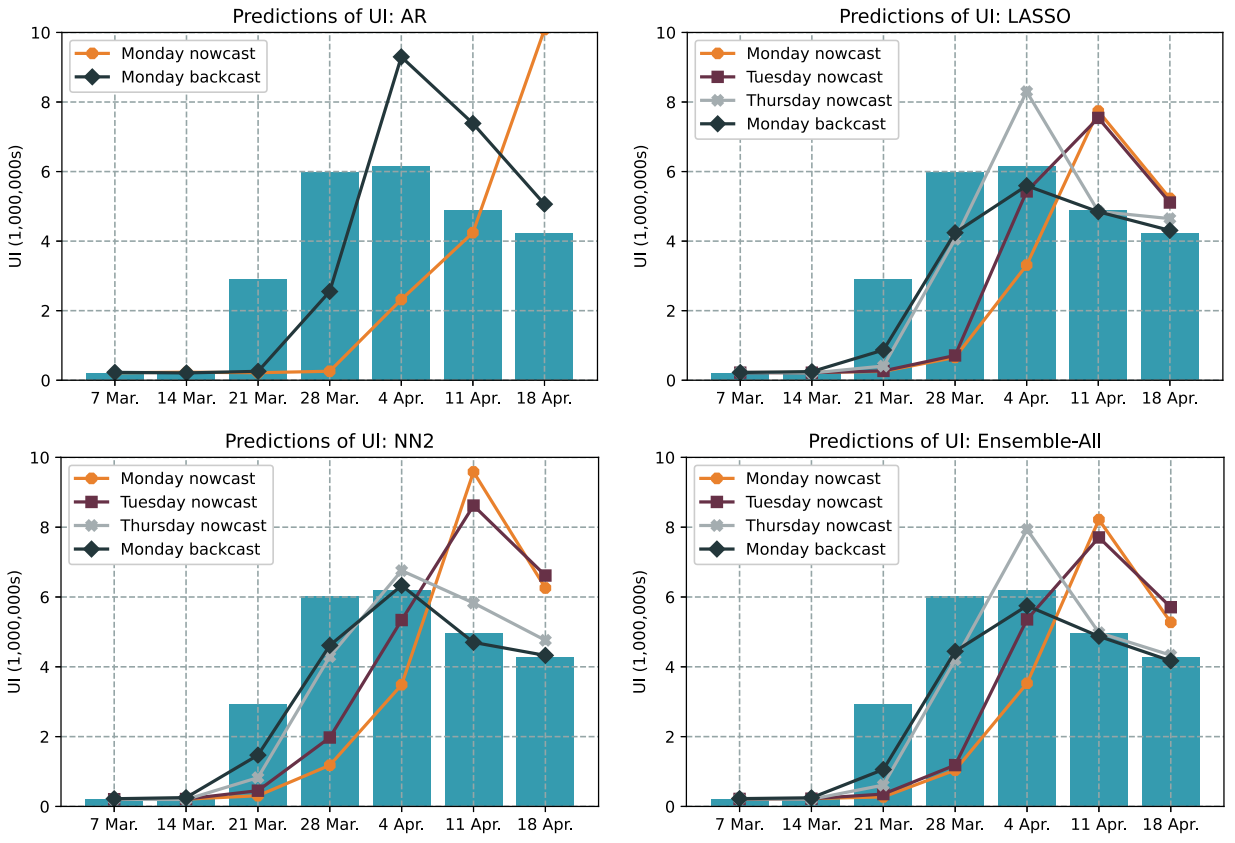


Fig. 2. Predictions near the advent of COVID-19. Notes: The figure depicts predictions of weekly unemployment insurance initial claims (UI) based on fitted autoregressive (AR) benchmark models, linear models fitted via the LASSO, fitted two-layer artificial neural networks (NN2), and an ensemble based on linear models fitted via the LASSO and elastic net and three fitted artificial neural networks. Apart from the AR benchmark, the predictions incorporate the information in daily Google Trends search volumes for terms related to *unemployment*. Each panel contains results for four prediction-formation days: Monday$_t$, Tuesday$_t$, and Thursday$_t$ nowcasts and Monday$_{t+1}$ backcast. The blue bars show the realized values of UI.

- *UI*. Level of UI.
- *Log of weekly COVID-19 deaths*. Log of weekly reported US deaths due to COVID-19. Data for reported US deaths due to COVID-19 are from the European Centre for Disease Prevention and Control.[30]
- *Stringency indicator*. An indicator variable that takes a value of one when the government imposes stringent measures to mitigate COVID-19 and zero when such measures are not in place or easing. The indicator variable is based on the stringency index from the Oxford COVID-19 Government Response Tracker (Hale et al., 2020). The index increases when measures such as school and retail establishment closures and travel restrictions are imposed and decreases when they are relaxed.
- *Workplace indicator*. Indicator variable that takes a value of one when the government imposes stringent workplace closure measures to mitigate COVID-19 and zero when such measures are not in place or easing. The indicator variable is based on the workplace closure sub-index of the Hale et al. (2020) stringency index.

Table 6 reports the Wald statistic for testing $\phi_0^{(j)} = \phi_1^{(j)} = 0$ and OLS estimate of $\phi_1^{(j)}$ ($\hat{\phi}_1^{(j)}$) in Eq. (14) for each model for the Monday$_{t+1}$ backcast.[31] The second and third columns report results for UI. The Wald statistic is significant at the 1% level for all models, so that there is significant evidence of conditional predictive ability according to the Giacomini and White (2006) test. The $\hat{\phi}_1^{(j)}$ estimates are all positive and significant at the 1% level in the third column. The competing models that include GT search volumes thus become significantly more accurate than the AR benchmark in terms of MSE as UI increases. Results for weekly COVID-19 deaths are reported in the fourth and fifth columns of Table 6. No Wald statistics are significant at conventional levels. However, the $\hat{\phi}_1^{(j)}$ estimates are all positive and significant (at the 5% level), so that the competing models deliver significantly more

Wald statistic for testing $\phi_0^{(j)} = \phi_1^{(j)} = 0$ and t-statistic for testing $\phi_1^{(j)} = 0$.

[30] Because COVID-19 deaths are zero before the last week of February 2020, we add a very small positive number to the series to compute the log.

[31] The results are qualitatively similar for the other nowcasts and backcasts. To make the numbers in Table 6 easier to read, we divide the actual and predicted UI values by 100,000 before estimating Eq. (14).

Table 6
Relative performance and COVID-19.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)
	UI		Log weekly COVID-19 deaths		Stringency indicator		Workplace indicator	
Model	Wald	$\hat{\phi}_1^{(j)}$	Wald	$\hat{\phi}_1^{(j)}$	Wald	$\hat{\phi}_1^{(j)}$	Wald	$\hat{\phi}_1^{(j)}$
LASSO	10. 76***	9. 90***	4. 06	2. 13**	4. 29	124. 85**	12. 95***	72. 77**
ENet	11. 02***	9. 95***	4. 12	2. 15**	4. 22	126. 05**	12. 39***	73. 54**
NN1	11. 23***	9. 81***	3. 93	2. 15**	17. 18***	130. 62**	4. 66*	74. 36**
NN2	9. 86***	9. 97***	3. 87	2. 19**	20. 15***	133. 34**	9. 04**	76. 03**
NN3	11. 49***	8. 95***	4. 44	1. 98**	8. 50**	117. 01**	8. 14**	67. 71**
Ensemble-Linear	10. 90***	9. 93***	4. 10	2. 15**	4. 19	125. 57**	12. 94***	73. 26**
Ensemble-ANN	12. 26***	10. 19***	4. 11	2. 23**	14. 41***	134. 37**	5. 06*	77. 09**
Ensemble-All	11. 66***	10. 25***	4. 23	2. 24**	7. 66**	132. 55**	14. 99***	76. 82**

Notes: The table reports estimation results for the regression model, $d_t^{(j)} = \phi_0^{(j)} + \phi_1^{(j)} q_t + e_t^{(j)}$, pertaining to the Monday$_{t+1}$ backcast for different q_t variables, where $d_t^{(j)}$ is the loss differential in Eq. (13); the loss differential corresponds to the week-t loss for a prediction of unemployment insurance initial claims (UI) based on an autoregressive benchmark model relative to that for the competing model in the first column. The competing models incorporate the information in daily Google Trends search volumes for terms related to *unemployment*. LASSO and ENet are linear models fitted via the LASSO and elastic net, respectively. NN1, NN2, and NN3 are fitted artificial neural networks (ANNs) with one, two, and three hidden layers, respectively. Ensemble-Linear (Ensemble-ANN) is an average of the predictions based on the LASSO and ENet (NN1, NN2, and NN3) models. Ensemble-All is the average of the predictions based on the LASSO, ENet, NN1, NN2, and NN3 models. The q_t variable for the regression is given at the top of the table. The second, fourth, sixth, and eighth columns report the Wald statistic for testing $\phi_0^{(j)} = \phi_1^{(j)} = 0$; the third, fifth, seventh, and ninth columns report the ordinary least squares estimate of $\phi_1^{(j)}$; *, **, and *** indicate significance at the 10%, 5%, and 1% level, respectively.

accurate predictions vis-à-vis the AR benchmark as the number of weekly COVID-19 deaths increases.

The sixth and seventh (eighth and ninth) columns of Table 6 report results for the indicator variable based on the stringency index (workplace sub-index). Five of the eight Wald statistics are significant at conventional levels for the stringency indicator, while all eight are significant for the workplace indicator. For both the stringency and workplace indicators, the $\hat{\phi}_1^{(j)}$ estimates are all positive and significant (at the 5% level), meaning that increases in the indicators are associated with significant improvements in the predictive ability of the competing models relative to the AR benchmark. Overall, the results in Table 6 indicate that the information in GT search volume data significantly improves predictive accuracy during periods when COVID-19 creates greater social stress.

6. Interpreting the fitted models

While the LASSO and ENet facilitate the interpretation of fitted linear models by performing variable selection, fitted ANNs are "black boxes" that are difficult to interpret. In this section, we use ALE plots (Apley & Zhu, 2020) and variable-importance measures (Greenwell et al., 2018) to peer into the black box of the fitted ANNs and compare them to the fitted linear models.

6.1. ALE plots and variable importance

For ease of exposition, we gather the predictors in the 617×1 vector,

$$\boldsymbol{x}_t = \begin{bmatrix} y_{t-1} & \boldsymbol{x}_t^{(j)\prime} \end{bmatrix}'. \tag{15}$$

Suppose that we are interested in analyzing the effect of a given predictor x_s on the predicted value of the target for a fitted model. Letting $\boldsymbol{x}_{c(s)} = \boldsymbol{x} \setminus x_s$ and $\hat{f}$ be the fitted prediction function, the ALE of x_s on $\hat{f}$ is defined as

$$\begin{aligned} g_{\text{ALE}}(x_s) &= \int_{x_{s,\min}}^{x_s} \mathbb{E}_{\boldsymbol{x}_{c(s)}} \left[\hat{f}^s \left(X_s, \boldsymbol{x}_{c(s)} \right) \middle| X_s = z_s \right] dz_s \\ &\quad - \text{constant} \\ &= \int_{x_{s,\min}}^{x_s} \left[\int \hat{f}^s \left(z_s, \boldsymbol{x}_{c(s)} \right) p_{c(s)|s} \left(\boldsymbol{x}_{c(s)} \middle| z_s \right) d\boldsymbol{x}_{c(s)} \right] dz_s \\ &\quad - \text{constant}, \end{aligned} \tag{16}$$

where $\hat{f}^s \left(x_s, \boldsymbol{x}_{c(s)} \right) = \partial \hat{f} \left(x_s, \boldsymbol{x}_{c(s)} \right) / \partial x_s$, $x_{s,\min}$ is a value near the lower bound of the training observations for predictor s, $p_{c(s)|s} \left(\boldsymbol{x}_{c(s)} \middle| x_s \right)$ is the joint conditional probability density for $\boldsymbol{x}_{c(s)}$ given x_s, and the constant term is used to center the ALE plot (as subsequently explained). Intuitively, Eq. (16) measures how the prediction changes in a narrow window around z_s and uses the partial derivative to control for the effects of correlated predictors, producing the local effect; the local effects are then accumulated via the outer integral in Eq. (16).[32]

[32] Partial dependence plots (Friedman, 2001) are a popular tool for measuring how changes in a predictor affect the prediction for a fitted model. However, partial dependence plots can produce biased results when predictors are highly correlated, and the ALE addresses the bias. See Molnar (2022) for a textbook exposition of partial dependence plots and ALEs.

To estimate the ALE in Eq. (16), the uncentered component is first estimated as follows:

$$\tilde{g}_{\text{ALE}}(x_s) = \sum_{k=1}^{k_s} \frac{1}{t_s(k)} \times \sum_{t:x_{t,s}\in\mathcal{T}_s(k)} \left[\hat{f}\left(z_{k,s}, \boldsymbol{x}_{t,c(s)}\right) - \hat{f}\left(z_{k-1,s}, \boldsymbol{x}_{t,c(s)}\right)\right], \tag{17}$$

where the ordered $x_{t,s}$ training observations are divided into K (nonoverlapping) intervals, $\mathcal{T}_s(k)$ for $k = 1, \ldots, K$, defined by $z_{0,s}, z_{1,s}, \ldots, z_{K,s}$, with $z_{0,s}$ ($z_{K,s}$) set just below (equal to) the smallest (largest) observation for $x_{t,s}$; $t_s(k)$ is the number of training observations in interval k, defined by $x_{t,s} \in \left(z_{k-1,s}, z_{k,s}\right]$; and k_s denotes the index for the interval containing the value of x_s. The intervals span the support of ordered $x_{t,s}$ training observations. In essence, the support of the $x_{t,s}$ observations is divided into a large number of relatively narrow intervals, and the partial derivative in Eq. (16) is then approximated using the average value of the differences in the fitted prediction function at the end and beginning of a given interval. Using the differences in the fitted prediction function obviates the need to compute the partial derivative and can be applied to prediction functions for which the partial derivative does not exist. The average differences are then summed to get the estimated ALE. Finally, the ALE is centered by using the deviation from its mean:

$$\hat{g}_{\text{ALE}}(x_s) = \tilde{g}_{\text{ALE}}(x_s) - \frac{1}{T}\sum_{t=1}^{T} \tilde{g}_{\text{ALE}}\left(x_{t,s}\right). \tag{18}$$

In our application, we use $K = 50$ when computing Eqs. (17) and (18).

By construction, the ALE plot for a fitted linear model will have a constant slope, while it will be a horizontal line at zero for a predictor that the LASSO or ENet does not select. By inspecting ALE plots for the fitted ANNs, we can gauge the importance of nonlinearities in the the fitted models.[33]

Greenwell et al. (2018) develop a convenient and intuitive variable-importance metric based on the variation of the partial dependence plot (Friedman, 2001) around its average value. Similarly to Christensen, Siggaard, and Veliyev (2021), we compute the Greenwell et al. (2018) variable-importance measure by replacing the partial dependence plot with the ALE plot:

$$\tilde{\mathcal{I}}(x_s) = \left\{\frac{1}{T-1}\sum_{t=1}^{T}\left[\hat{g}_{\text{ALE}}\left(x_{t,s}\right)\right]^2\right\}^{0.5}. \tag{19}$$

Eq. (19) measures the importance of a predictor via the standard deviation of the ALE plot. For a predictor with a horizontal ALE plot, the prediction of the target does not vary with the predictor, so that its variable importance is zero. As the prediction corresponding to the ALE plot fluctuates more about its average value, the variable importance measure increases. To facilitate comparison across predictors, we scale Eq. (19) using the sum of the individual measures:

$$\hat{\mathcal{I}}(x_s) = \frac{\tilde{\mathcal{I}}(x_s)}{\sum_{p=1}^{P}\tilde{\mathcal{I}}\left(x_p\right)}, \tag{20}$$

where P is the total number of predictors, so that $\hat{\mathcal{I}}(x_s)$ ranges from zero to one.

6.2. Importance of Google Trends search volumes

Figs. 3 and 4 depict variable-importance measures based on Eq. (20) for the top 25 predictors for linear models fitted via the LASSO and fitted NN2 models, respectively. The left panels of the figures correspond to models trained with data through the penultimate week of December 2019 (used to generate predictions for the last week of December 2019), before the start of the COVID-19 crisis. The right panels are for models estimated using data through the penultimate week of December 2020 (used to generate predictions for the last week of December 2020), so that the training sample includes the crisis. Note that a given search term can appear up to seven times in the same plot due to our mixed-frequency framework. To conserve space, we focus on fitted models for the Monday$_t$ and Thursday$_t$ nowcasts and Monday$_{t+1}$ backcast (with data overlaps of zero, three, and seven days, respectively).

For the pre-crisis sample, the lag of UI (*lag*) is the most important predictor for the linear models fitted via the LASSO in Fig. 3, with $\hat{\mathcal{I}}(x_s)$ scores for *lag* ranging from approximately 0.14 to 0.30. For the fitted NN2 models and pre-crisis sample in Fig. 4, *lag* is also the most important predictor for the Monday$_t$ and Thursday$_t$ nowcasts, with $\hat{\mathcal{I}}(x_s)$ values of nearly 0.07, while it is the fourth most important predictor for the Monday$_{t+1}$ backcast. Taken together, the GT search volumes are more important for the fitted NN2 models vis-à-vis the linear models fitted via the LASSO for the pre-crisis sample.

The GT search volumes become substantially more important in the right panels of Figs. 3 and 4 when we include data from the COVID-19 crisis, especially for the Monday$_{t+1}$ backcast. Although *lag* remains the most important predictor for the Monday$_t$ and Thursday$_t$ nowcast across both models, it drops out entirely of the top 25 for the Monday$_{t+1}$ backcast for both models. In other words, when we use the maximum data overlap in computing the predictions, the fitted models assign little importance to the autocorrelation in UI, so that the GT search volumes dominate the AR component.

Table 7 provides additional information on the growing importance of the GT search volumes in the sequence of nowcasts and backcasts, especially for the sample that includes the COVID-19 crisis. The table reports the joint importance of the GT search volumes for each prediction-formation day. For the linear models fitted via the LASSO and pre-crisis sample in the fourth column, the GT search volumes grow in importance from around 0.60 for the initial nowcasts to about 0.80 for the later backcasts. When the training sample includes the COVID-19 crisis

[33] Another approach for analyzing the role of nonlinearities is to include a linear unit in the neural networks, as in Trapletti, Leisch, and Hornik (2000). While beyond the scope of the present paper, this extension is an interesting avenue for future research.

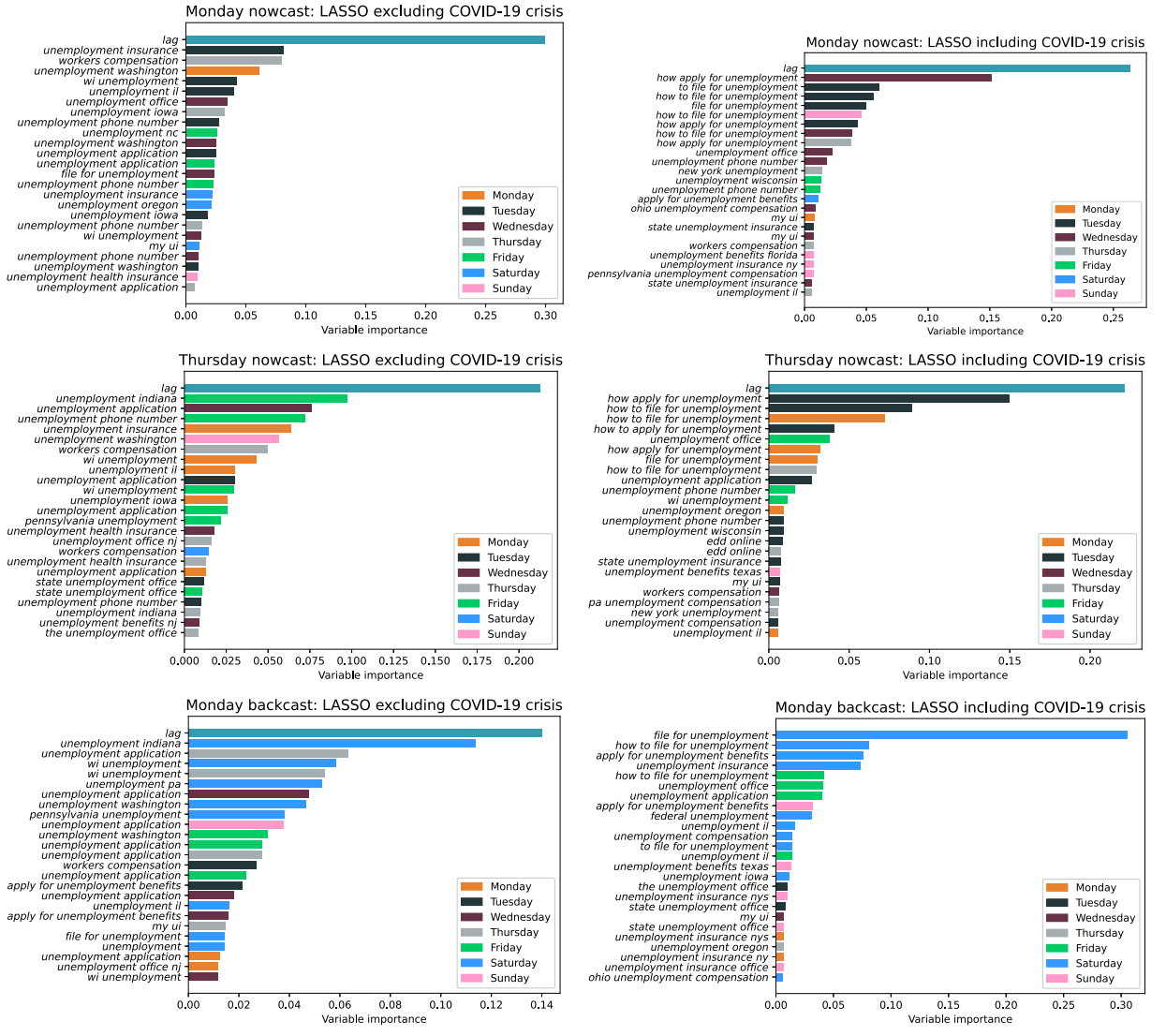


Fig. 3. Variable-importance measures for fitted linear models. Notes: The figure depicts variable-importance measures for the top 25 predictors for linear models fitted via the LASSO. Results are reported for the Monday$_t$ and Thursday$_t$ nowcasts and Monday$_{t+1}$ backcast. The training sample for the left (right) panels ends in the penultimate week of December 2019 (2020), thereby excluding (including) the COVID-19 crisis.

(see the fifth column), the GT search volumes again grow in importance, but the level is markedly higher for each prediction-formation day. For the Monday$_t$ nowcast, the joint importance score for the GT search volumes is 0.705. It reaches 1.000 for the Saturday$_t$ nowcast through the Wednesday$_{t+1}$ backcast so that the AR component becomes unimportant. A similar pattern holds for the fitted NN2 models in the last two columns. The joint importance measures for the GT search volumes are typically larger (often substantially so) than the corresponding values in the third and fourth columns.

The increasing importance of the GT search volumes since the start of the COVID-19 crisis is also evident in Fig. 5. It shows the number of predictors selected by the LASSO and ENet for rolling-window estimation of the linear prediction models underlying the Monday$_t$, Tuesday$_t$, and Thursday$_t$ nowcasts and Monday$_{t+1}$ backcast. The number of selected predictors sharply increases in the fitted linear models with the advent of the crisis, with spikes evident in the number of selected predictors for training samples ending in the third week of March 2020. As expected, the ENet usually selects more predictors than the LASSO in Fig. 5.

Returning to Figs. 3 and 4, for the prediction-formation days where the GT search volumes matter the most – most notably, the Monday$_{t+1}$ backcast – there is a tendency for both the fitted linear and NN2 models to place more weight on recently available search queries. For example, the Monday$_{t+1}$ backcast attaches the greatest importance to GT search volumes for Saturday and Friday (recall the two-day lag in the availability of GT data). To explore this issue further, Fig. 6 shows heatmaps for the joint importance of GT search volumes organized according to the day of the week. For the pre-crisis sample in the left panels, there is no discernible pattern in the importance of GT search volumes across the days of the week. For the sample that includes the COVID-19 crisis in the right panels, a strong pattern is evident: except for the

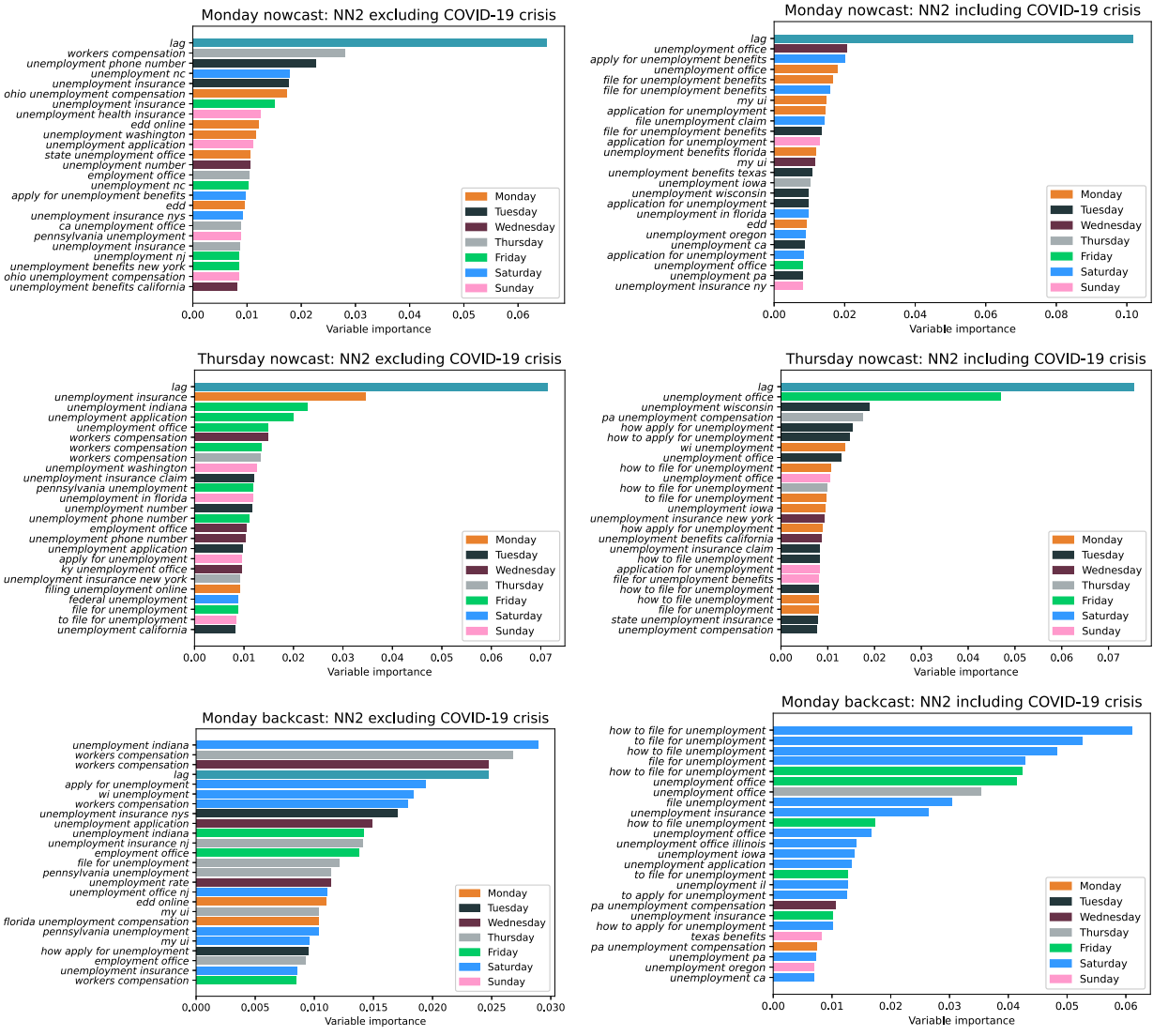


Fig. 4. Variable-importance measures for fitted artificial neural networks. Notes: The figure depicts variable-importance measures for the top 25 predictors for fitted artificial neural networks with two hidden layers (NN2). Results are reported for the Monday$_t$ and Thursday$_t$ nowcasts and Monday$_{t+1}$ backcast. The training sample for the left (right) panels ends in the penultimate week of December 2019 (2020), thereby excluding (including) the COVID-19 crisis.

Wednesday$_{t+1}$ backcast, the nowcasts and backcasts attach relatively high importance to GT search volumes for the most recent day of available data. This is most evident for the Monday$_{t+1}$ and Tuesday$_{t+1}$ backcasts, which use data overlaps of seven and six days, respectively, as the collective importance of GT search volumes is 0.67 and 0.56 (0.44 and 0.46) for the linear model fitted via the LASSO (NN2 model) for the Monday$_{t+1}$ and Tuesday$_{t+1}$ backcasts, respectively. This highlights the usefulness of mixed-frequency GT data for anticipating UI during the COVID-19 crisis.

Looking back to Figs. 3 and 4, an interesting pattern emerges in the types of search terms that appear important across the two samples. For the fitted linear and NN2 models, the pre-COVID-19 sample is characterized by a wide variety of search queries (e.g., some geographical terms and some generic terms like *workers compensation*). For the sample that includes the COVID-19 crisis, there appears to be greater emphasis on search queries related to the application process for unemployment insurance benefits (e.g., *how to file for unemployment*, *unemployment office*, *unemployment application*).

6.3. ALE plots for Google Trends search volumes

In this section, we investigate ALE plots for some important predictors. We again report results for linear models fitted via the LASSO and fitted NN2 models for training samples that exclude and include the COVID-19 crisis. Fig. 7 presents ALE plots for GT search volumes for the terms *file for unemployment*, *how to file for unemployment*, and *workers compensation*. The figure reports results for

Table 7
Importance of Google Trends search volumes.

(1) Prediction formation	(2) Nowcast/ backcast	(3) Data overlap (days)	(4) LASSO excluding COVID-19	(5) LASSO including COVID-19	(6) NN2 excluding COVID-19	(7) NN2 including COVID-19
Monday$_t$	Nowcast	0	0.574	0.705	0.921	0.931
Tuesday$_t$	Nowcast	1	0.642	0.721	0.881	0.942
Wednesday$_t$	Nowcast	2	0.664	0.711	0.812	0.833
Thursday$_t$	Nowcast	3	0.692	0.724	0.811	0.961
Friday$_t$	Nowcast	4	0.582	0.751	0.881	0.831
Saturday$_t$	Nowcast	5	0.552	1.000	0.871	0.982
Sunday$_{t+1}$	Backcast	6	0.611	1.000	0.881	0.981
Monday$_{t+1}$	Backcast	7	0.788	1.000	0.949	0.912
Tuesday$_{t+1}$	Backcast	6	0.799	1.000	0.962	0.991
Wednesday$_{t+1}$	Backcast	5	0.712	1.000	0.951	0.997

Notes: The table reports joint variable-importance measures for all of the daily Google Trends search volumes in fitted linear models estimated via the LASSO and fitted artificial neural networks with two hidden layers (NN2). The third column provides the number of days of overlap between the daily Google Trends data and week-t unemployment insurance initial claims. The fourth and sixth (fifth and seventh) columns report results for training samples excluding (including) the COVID-19 crisis.

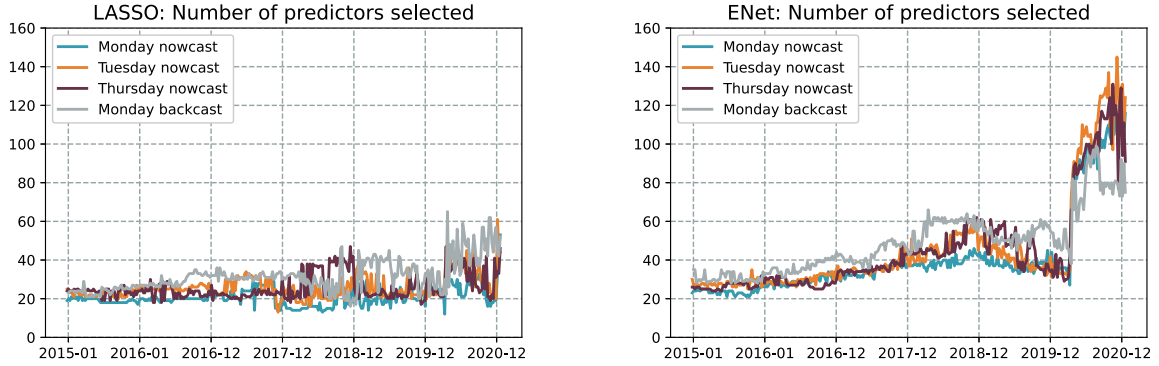


Fig. 5. Number of predictors selected by the LASSO and ENet. Notes: The figure depicts the number of predictors selected by the LASSO and elastic net (ENet) for rolling-window estimation of linear models used to generate out-of-sample predictions, where each model can include up to 617 predictors. Results are reported for the Monday$_t$, Tuesday$_t$, and Thursday$_t$ nowcasts and Monday$_{t+1}$ backcast. The out-of-sample period spans the first week of January 2015 through the last week of December 2020.

the Thursday$_t$ nowcast and Monday$_{t+1}$ backcast. Search volumes for the three terms are for the most recently available day of GT data. The first term, *file for unemployment*, is the first (fourth) most important term for the fitted linear (NN2) model for the Monday$_{t+1}$ backcast for the sample that includes the crisis. The second term, *how to file for unemployment*, is the second (first) most important predictor for the fitted linear (NN2) model. The last term, *workers compensation*, is among the most important predictors for the fitted NN2 model for the pre-crisis sample. Because the observation supports differ across the training samples that exclude and include the crisis, the values on the horizontal axis are normalized to lie between zero and one.

For *file for unemployment* in the first row of Fig. 7, the LASSO does not select the term for the pre-crisis sample for the Thursday$_t$ nowcast in the left panel, while the LASSO selects the term for the sample that includes the crisis, where the predictive relationship is positive for the fitted linear model. For the fitted NN2 models, the slopes of the ALE plots are positive for both samples for the Thursday$_t$ nowcast. Observe that the ALE plots are close to flat for relatively small search volume values, while they become considerably more positively sloped after that. For the linear models fitted via the LASSO, the results for the Monday$_{t+1}$ backcast in the right panel are similar to those in the left panel. The results are also similar across the Thursday$_t$ nowcast and Monday$_{t+1}$ backcast for the fitted NN2 models, with the exception that the ALE plot is essentially flat for the pre-crisis sample for the Monday$_{t+1}$ backcast.

Regarding *how to file for unemployment* in the middle row of Fig. 7, the LASSO again does not select the term for the pre-crisis sample for the Thursday$_t$ nowcast in the left panel, while the LASSO selects the term for the sample that includes the crisis, and the fitted linear model generates a positively sloped ALE plot. For the fitted NN2 model and pre-crisis sample, the ALE plot for *how to file for unemployment* is essentially flat for the Thursday$_t$ nowcast. For the sample that includes the crisis, it has a positive slope. The slopes for the fitted linear and NN2 models are similar for relatively small search volume

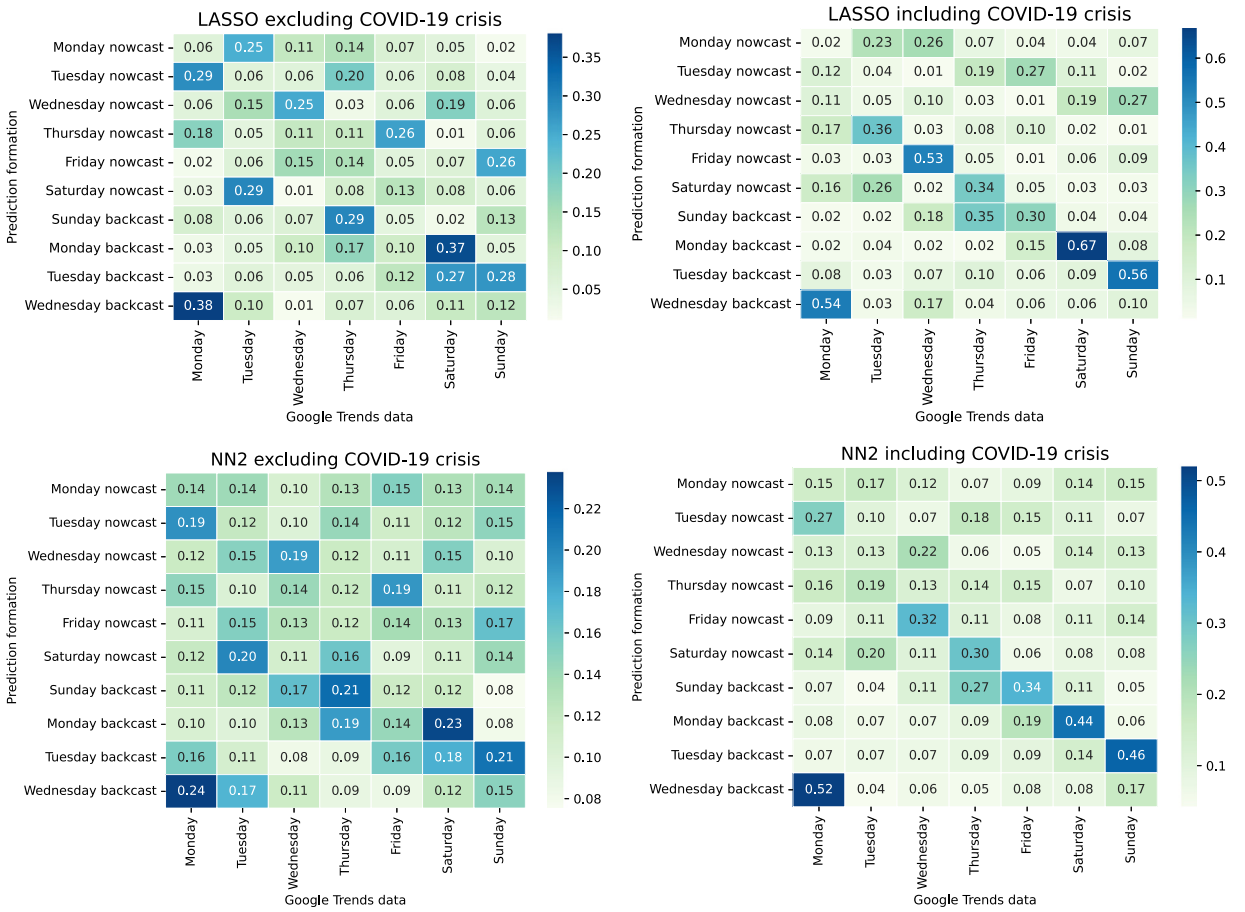


Fig. 6. Day-of-the-week effects. Notes: The figure depicts heatmaps for joint variable-importance measures of Google Trends search volumes grouped by the day of the week. The results are for linear models fitted via the LASSO and fitted artificial neural networks with two hidden layers (NN2). The fitted models underpin the nowcasts and backcasts indicated on the vertical axis. The training sample for the left (right) panels ends in the penultimate week of December 2019 (2020), thereby excluding (including) the COVID-19 crisis.

values. The slope of the ALE plot for the NN2 model then becomes substantially steeper than that for the linear model, providing evidence of important nonlinearities. For the Monday$_{t+1}$ nowcast in the right panel, the LASSO does not select *how to file for unemployment* for either sample. For the NN2 model, the ALE plot is essentially flat for the pre-crisis sample for the Monday$_{t+1}$ nowcast. Although it is also nearly flat for relatively small search volume values for the sample that includes the crisis, it then becomes steeply positively sloped.

Finally, for *workers compensation* in the last row of Fig. 7, the LASSO does not select the term for the sample that includes the crisis for the Thursday$_t$ nowcast in the left panel. The LASSO selects the term for the pre-crisis sample, where the ALE plot is positively, but not very steeply, sloped. The ALE plot is essentially flat for the fitted NN2 model for the pre-crisis sample for the Thursday$_t$ nowcast. It then becomes steeply positively sloped for the sample that includes the crisis. The LASSO does not select *workers compensation* for either sample for the Monday$_{t+1}$ backcast in the right panel. For the fitted NN2 models, the ALE plots are positively sloped for both samples for the Monday$_{t+1}$ backcast. Following the pattern in the first two rows, the ALE plots for the fitted NN2 models are less steeply sloped for relatively small search volume values, so that the predictive relationships become stronger for larger search volume values.

7. Conclusion

We propose an MFML approach for generating out-of-sample predictions that combines the U-MIDAS method with machine-learning techniques. The U-MIDAS component allows for the use of higher-frequency predictor data, while machine-learning techniques help to guard against overfitting in high-dimensional settings and can accommodate nonlinear predictive relationships. By combining the U-MIDAS method with machine-learning devices, MFML is designed to help improve out-of-sample prediction and model interpretation in the era of big data.

In an application of MFML, we show that the information in high-dimensional daily internet search volume data can be used to substantially improve predictions of weekly UI in anticipation of its Thursday release by the Department of Labor. We construct a sequence of nowcasts and backcasts formed ten days to one day ahead of

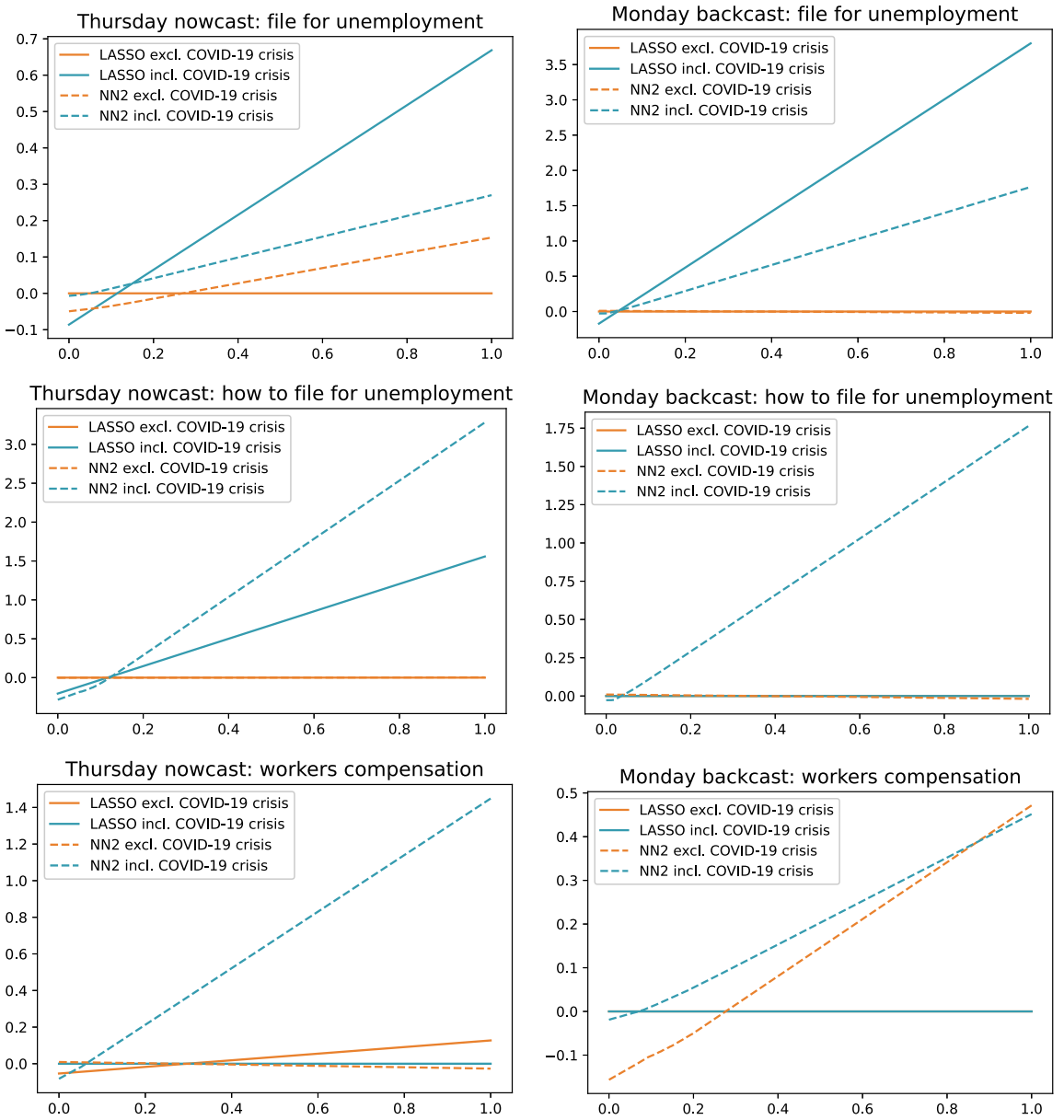


Fig. 7. ALE plots. Notes: The figure depicts accumulated local effect plots for Google Trends search volumes for three terms (*file for unemployment*, *how to file for unemployment*, and *workers compensation*) in linear models fitted via the LASSO and fitted artificial neural networks with two hidden layers (NN2). The values on the horizontal axis are normalized to lie between zero and one. Results are reported for the Thursday$_t$ nowcast (left panels) and Monday$_{t+1}$ backcast (right panels). Search volumes for the three terms are for the most recently available day of Google Trends data. The training sample ends in the penultimate week of December 2019 (2020), thereby excluding (including) the COVID-19 crisis.

the UI release on Thursday of each week. The predictions are based on models with daily GT search volumes for a large number of terms related to *unemployment*, including linear models estimated via the LASSO and ENet and nonlinear ANNs. The mixed-frequency component of the MFML approach allows us to incorporate daily GT data as they become available, thereby providing more timely information for predicting weekly UI. The machine-learning component is appropriate for our high-dimensional setting. The mixed-frequency component also allows us to measure the predictive power of each additional day of information. In a simulated out-of-sample exercise, nowcasts and backcasts based on daily GT data substantially outperform an AR benchmark in terms of RMSE. As the sequence of nowcasts and backcasts incorporates more recent daily GT data, predictive accuracy generally improves, leading to reductions in RMSE of up to 65% vis-à-vis the AR benchmark. The information in daily GT search volumes also largely subsumes that in a set of daily macro-financial predictors.

We detect strong links between the relevance of daily GT data for predicting weekly UI and the COVID-19 crisis. Daily GT data are beneficial for improving the accuracy

Table A.1
RMSE ratios for seasonally adjusted initial claims.

(1)	(2)	(3)	(4) Linear	(5)	(6) ANN	(7)	(8)	(9) Ensemble	(10)	(11)
Prediction formation	Nowcast/ backcast	AR RMSE	LASSO	ENet	NN1	NN2	NN3	Linear	ANN	All
Monday_t	Nowcast	635,843	0.736	0.717	0.795	0.886	0.901	0.726	0.812	0.752
Tuesday_t	Nowcast	635,843	0.642	0.636	0.712	0.792	0.899	0.637	0.819	0.697
Wednesday_t	Nowcast	635,843	0.604	0.591	0.686	0.615	0.724	0.597	0.684	0.640
Thursday_t	Nowcast	495,825	0.598	0.554	0.758	0.599	0.781	0.574	0.633	0.592
Friday_t	Nowcast	495,825	0.522	0.491	0.589	0.580	0.666	0.505	0.638	0.548
Saturday_t	Nowcast	495,825	0.433	0.390	0.546	0.486	0.612	0.406	0.535	0.462
Sunday_{t+1}	Backcast	495,825	0.396	0.369	0.423	0.432	0.599	0.376	0.467	0.444
Monday_{t+1}	Backcast	495,825	0.384	0.344	0.491	0.482	0.538	0.356	0.495	0.428
Tuesday_{t+1}	Backcast	495,825	0.358	0.331	0.382	0.377	0.421	0.339	0.389	0.355
Wednesday_{t+1}	Backcast	495,825	0.356	0.318	0.399	0.394	0.424	0.329	0.420	0.389

Notes: The table reports out-of-sample results for nowcasts and backcasts of week-t seasonally adjusted unemployment insurance initial claims (UI) formed on the day indicated in the first column, where the subscript denotes the week when the prediction is made. The second column provides the classification as a nowcast or backcast of week-t UI. The third column reports the root mean squared error (RMSE) for an autoregressive (AR) benchmark model. The fourth through eleventh columns report the RMSE ratio for the competing model in the column heading vis-à-vis the AR benchmark. The competing models incorporate the information in daily Google Trends search volumes for terms related to *unemployment*. The fourth and fifth columns are for linear models fitted via the LASSO and elastic net (ENet), respectively. The sixth through eighth columns are for fitted artificial neural networks (ANNs) with one (NN1), two (NN2), and three (NN3) hidden layers, respectively. The ensemble in the ninth (tenth) column is an average of the predictions for the models in the fourth and fifth (sixth through eighth) columns. The ensemble in the eleventh column is an average of the predictions for the models in the fourth through eighth columns. The out-of-sample period begins in the first week of January 2015 and ends in the last week of December 2020.

Table A.2
RMSE ratios for seasonal AR benchmark and competing models.

(1)	(2)	(3)	(4) Linear	(5)	(6) ANN	(7)	(8)	(9) Ensemble	(10)	(11)
Prediction formation	Nowcast/ backcast	Seasonal AR RMSE	LASSO	ENet	NN1	NN2	NN3	Linear	ANN	All
Monday_t	Nowcast	576,756	0.733	0.735	0.720	0.761	0.799	0.733	0.773	0.754
Tuesday_t	Nowcast	576,756	0.652	0.657	0.652	0.678	0.691	0.654	0.688	0.650
Wednesday_t	Nowcast	576,756	0.596	0.595	0.649	0.623	0.682	0.595	0.633	0.622
Thursday_t	Nowcast	354,337	0.603	0.607	0.656	0.711	0.727	0.604	0.712	0.612
Friday_t	Nowcast	354,337	0.547	0.587	0.691	0.533	0.631	0.566	0.615	0.574
Saturday_t	Nowcast	354,337	0.466	0.524	0.589	0.519	0.526	0.492	0.545	0.486
Sunday_{t+1}	Backcast	354,337	0.532	0.484	0.448	0.443	0.602	0.507	0.512	0.501
Monday_{t+1}	Backcast	354,337	0.487	0.474	0.410	0.416	0.454	0.480	0.436	0.439
Tuesday_{t+1}	Backcast	354,337	0.471	0.498	0.489	0.435	0.503	0.483	0.445	0.468
Wednesday_{t+1}	Backcast	354,337	0.551	0.555	0.458	0.451	0.508	0.552	0.462	0.482

Notes: The table reports out-of-sample results for nowcasts and backcasts of week-t unemployment insurance initial claims (UI) formed on the day indicated in the first column, where the subscript denotes the week when the prediction is made. The second column provides the classification as a nowcast or backcast of week-t UI. The third column reports the root mean squared error (RMSE) for an autoregressive (AR) benchmark model, which includes monthly indicator variables to account for seasonality. The fourth through eleventh columns report the RMSE ratio for the competing model in the column heading vis-à-vis the AR benchmark. The competing models incorporate the information in daily Google Trends search volumes for terms related to *unemployment*, along with monthly indicator variables to account for seasonality. The fourth and fifth columns are for linear models fitted via the LASSO and elastic net (ENet), respectively. The sixth through eighth columns are for fitted artificial neural networks (ANNs) with one (NN1), two (NN2), and three (NN3) hidden layers, respectively. The ensemble in the ninth (tenth) column is an average of the predictions for the models in the fourth and fifth (sixth through eighth) columns. The ensemble in the eleventh column is an average of the predictions for the models in the fourth through eighth columns. The out-of-sample period begins in the first week of January 2015 and ends in the last week of December 2020.

of the nowcasts and backcasts near the advent of the crisis. We also find that the predictive accuracy of models that include daily GT data improves significantly relative to that of the AR benchmark model as social conditions associated with the crisis worsen. Furthermore, variable-importance measures for fitted models reveal that daily

GT data become more relevant for predicting weekly UI when the training sample includes data from the crisis.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix

See Tables A.1 and A.2.

References

Aaronson, D., Brave, S. A., Butters, R. A., Fogarty, M., Sacks, D. W., & Seo, B. (2022). Forecasting unemployment insurance claims in realtime with Google Trends. *International Journal of Forecasting*, *38*(2), 567–581.

Apley, D. W., & Zhu, J. (2020). Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *82*(4), 1059–1086.

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

Baker, S. R., Bloom, N., & Davis, S. J. (2016). Measuring economic policy uncertainty. *Quarterly Journal of Economics*, *131*(4), 1593–1636.

Baker, S. R., Bloom, N., Davis, S. J., Kost, K., Sammon, M., & Viratyosin, T. (2020). The unprecedented stock market reaction to COVID-19. *Review of Asset Pricing Studies*, *10*(9), 742–758.

Barron, A. R. (1994). Approximation and estimation bounds for artificial neural networks. *Machine Learning*, *14*(1), 115–133.

Bickel, P. J., Ritov, Y., & Tsybakov, A. B. (2009). Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics*, *37*(4), 1705–1732.

Bleher, J., & Dimpfl, T. (2021). Knitting multi-annual high-frequency Google Trends to predict inflation and consumption. *Econometrics and Statistics*, in press.

Borup, D., & Schütte, E. C. M. (2022). In search of a job: Forecasting employment growth using Google Trends. *Journal of Business & Economic Statistics*, *40*(1), 186–200.

Brave, S. A., Butters, R. A., & Justiniano, A. (2019). Forecasting economic activity with mixed frequency BVARs. *International Journal of Forecasting*, *35*(4), 1692–1707.

Caperna, G., Colagrossi, M., Geraci, A., & Mazzarella, G. (2022). A babel of web-searches: Googling unemployment during the pandemic. *Labour Economics*, *74*(1), Article 102097.

Choi, H., & Varian, H. (2012). Predicting the present with Google Trends. *Economic Record*, *88*(S1), 2–9.

Christensen, K., Siggaard, M., & Veliyev, B. (2021). A machine learning approach to volatility forecasting. CREATES Research Paper 2021-03.

Clements, M. P., & Galvão, A. B. (2008). Macroeconomic forecasting with mixed-frequency data: Forecasting output growth in the United States. *Journal of Business & Economic Statistics*, *26*(4), 546–554.

Cleveland, R. B., Cleveland, W. S., McRae, J. E., & Terpenning, I. (1990). STL: A seasonal-trend decomposition procedure based on loess. *Journal of Official Statistics*, *6*(1), 3–73.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems*, *2*(4), 303–314.

D'Amuri, F., & Marcucci, J. (2017). The predictive power of Google searches in forecasting US unemployment. *International Journal of Forecasting*, *33*(4), 801–816.

Davidson, P. (2020). Unemployment claims figures could be much lower because of new seasonal adjustment approach. *USA Today*, September 3.

Diebold, F. X., & Mariano, R. S. (1995). Comparing predictive accuracy. *Journal of Business & Economic Statistics*, *13*(3), 253–263.

Diebold, F. X., & Shin, M. (2019). Machine learning for regularized survey forecast combination: Partially-egalitarian Lasso and its derivatives. *International Journal of Forecasting*, *35*(4), 1679–1691.

Fan, J., & Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American Statistical Association*, *96*(456), 1348–1360.

Foroni, C., & Marcellino, M. (2014). A comparison of mixed frequency approaches for nowcasting Euro area macroeconomic aggregates. *International Journal of Forecasting*, *30*(3), 554–568.

Foroni, C., Marcellino, M., & Schumacher, C. (2015). Unrestricted mixed data sampling (MIDAS): MIDAS regressions with unrestricted lag polynomials. *Journal of the Royal Statistical Society. Series A. Statistics in Society*, *178*(1), 57–82.

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

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural Networks*, *2*(3), 183–192.

Ghysels, E., Santa-Clara, P., & Valkanov, R. (2005). There is a risk-return trade-off after all. *Journal of Financial Economics*, *76*(3), 509–548.

Giacomini, R. (2011). Testing conditional predictive ability. In M. P. Clements, & D. F. Hendry (Eds.), *Oxford handbook of economic forecasting*. Oxford, UK: Oxford University Press, chap. 15.

Giacomini, R., & White, H. (2006). Tests of conditional predictive ability. *Econometrica*, *74*(6), 1545–1578.

Goldsmith-Pinkham, P., & Sojourner, A. (2020). Predicting initial unemployment insurance claims using Google Trends. Manuscript.

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

Greenwell, B. M., Boehmke, B. C., & McCarthy, A. J. (2018). A simple and effective model-based variable importance measure. Working Paper arXiv:1805.04755v1.

Hale, T., Angrist, N., Cameron-Blake, E., Hallas, L., Kira, B., Majumdar, S., et al. (2020). *Oxford COVID-19 government response tracker*. Blavatnik School of Government.

Hastie, T., & Qian, J. (2016). Glmnet vignette. Manuscript.

Hoerl, A. E., & Kennard, R. W. (1970). Ridge regression: Applications to nonorthogonal problems. *Technometrics*, *12*(1), 69–82.

Hornik, K. (1991). Approximation capabilities of multilayer feedforward networks. *Neural Networks*, *4*(2), 251–257.

Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*(5), 359–366.

Hui, F. K. C., Warton, D. I., & Foster, S. D. (2015). Tuning parameter selection for the adaptive Lasso using ERIC. *Journal of the American Statistical Association*, *110*(509), 262–269.

Kingma, D. P., & Ba, J. (2015). Adam: A method for stochastic optimization. In *Third annual international conference on learning representations*.

Kotchoni, R., Leroux, M., & Stevanovic, D. (2019). Macroeconomic forecast accuracy in a data-rich environment. *Journal of Applied Econometrics*, *34*(7), 1050–1072.

Larson, W. D., & Sinclair, T. M. (2022). Nowcasting unemployment insurance claims in the time of COVID-19. *International Journal of Forecasting*, *38*(2), 635–647.

Lewis, D., Mertens, K., & Stock, J. H. (2020). US economic activity during the early weeks of the SARS-Cov-2 outbreak. NBER Working Paper No. 26954.

Masters, T. (1993). *Practical neural network recipes in C++*. Academic Press.

Medeiros, M. C., & Mendes, E. F. (2016). ℓ_1-Regularization of high-dimensional time-series models with non-Gaussian and heteroskedastic errors. *Journal of Econometrics*, *191*(1), 255–271.

Medeiros, M. C., Vasconcelos, G. F. R., Veiga, A., & Zilberman, E. (2021). Forecasting inflation in a data-rich environment: The benefits of machine learning methods. *Journal of Business & Economic Statistics*, *39*(1), 98–119.

Meinshausen, N., & Yu, B. (2009). Lasso-type recovery of sparse representations for high-dimensional data. *The Annals of Statistics*, *37*(1), 246–270.

Molnar, C. (2022). *Interpretable machine learning: A guide for making black box models explainable*. Independently published.

Niesert, R. F., Oorschot, J. A., Veldhuisen, C. P., Brons, K., & Lange, R.-J. (2020). Can Google search data help predict macroeconomic series? *International Journal of Forecasting*, *36*(3), 1163–1172.

Rinz, K. (2020). Understanding unemployment insurance claims and other labor market data during the COVID-19 pandemic. Manuscript.

Rolnick, D., & Tegmark, M. (2018). The power of deeper networks for expressing natural functions. In *Sixth annual international conference on learning representations*.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, *6*(2), 461–464.

Simon, N., Friedman, J., Hastie, T., & Tibshirani, R. (2013). A sparse-group Lasso. *Journal of Computational and Graphical Statistics*, *22*(2), 231–245.

Tibshirani, R. (1996). Regression shrinkage and selection via the LASSO. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *58*(1), 267–288.

Timmermann, A. (2006). Forecast combinations. In G. Elliott, C. W. J. Granger, & A. Timmermann (Eds.), *Handbook of economic forecasting, vol. 1* (pp. 135–196). Amsterdam: Elsevier.

Trapletti, A., Leisch, F., & Hornik, K. (2000). Stationary and integrated autoregressive neural network processes. *Neural Computation*, *12*(10), 2427–2450.

West, K. D. (1996). Asymptotic inference about predictive ability. *Econometrica*, *64*(5), 1067–1084.

Yi, D., Ning, S., Chang, C.-J., & Kou, S. C. (2021). Forecasting unemployment using internet search data via PRISM. *Journal of the American Statistical Association*, *116*(536), 1662–1673.

Zhang, C.-H., & Huang, J. (2008). The sparsity and bias of the Lasso selection in high-dimensional linear regression. *The Annals of Statistics*, *36*(4), 1567–1594.

Zou, H. (2006). The adaptive Lasso and its oracle properties. *Journal of the American Statistical Association*, *101*(476), 1418–1429.

Zou, H., & Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, *67*(2), 301–320.

Zou, H., Hastie, T., & Tibshirani, R. (2007). On the 'degrees of freedom' of the Lasso. *The Annals of Statistics*, *35*(5), 2173–2192.