International Journal of Forecasting 37 (2021) 941–948

Contents lists available at ScienceDirect

International Journal of Forecasting

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

Nowcasting GDP using machine-learning algorithms: A real-time assessment☆

Adam Richardson [a,*], Thomas van Florenstein Mulder [a], Tuğrul Vehbi [a,b,1]

[a] Reserve Bank of New Zealand, New Zealand
[b] CAMA, Australia

ARTICLE INFO

Keywords:
Nowcasting
Machine learning
Forecast evaluation
Forecasting practice
Macroeconomic forecasting

ABSTRACT

Can machine-learning algorithms help central banks understand the current state of the economy? Our results say yes! We contribute to the emerging literature on forecasting macroeconomic variables using machine-learning algorithms by testing the nowcast performance of common algorithms in a full 'real-time' setting—that is, with real-time vintages of New Zealand GDP growth (our target variable) and real-time vintages of around 600 predictors. Our results show that machine-learning algorithms are able to significantly improve over a simple autoregressive benchmark and a dynamic factor model. We also show that machine-learning algorithms have the potential to add value to, and in one case improve on, the official forecasts of the Reserve Bank of New Zealand.

1. Introduction

Policy makers typically make decisions in real time using incomplete information on current economic conditions. Many key statistics are released with lags and are subject to frequent revisions. Nowcasting models have been developed to mitigate some of these uncertainties, and they have been widely used by forecasters at many central banks and other institutions (Banbura, Giannone, Modugno, & Reichlin, 2013; Bloor, 2009; Giannone, Reichlin, & Small, 2008; Jansen, Jin, & de Winter, 2016).

Machine-learning (ML) algorithms are potential alternatives to time-series regression models typically used by central banks for forecasting key macroeconomic variables. ML models are particularly suited for handling data sets where the number of potential regressors is large. In this paper, we investigate the performance of different ML algorithms in obtaining accurate nowcasts of real gross domestic product (GDP) growth for New Zealand. We use multiple vintages of historical GDP data and multiple vintages of predictors—comprising approximately 600 domestic and international variables—to evaluate the real-time nowcast performance of these algorithms over the 2009Q1–2019Q1 period. We then compare the forecasts obtained from these algorithms with the forecasting accuracy of a naive autoregressive (AR) benchmark, a dynamic factor model, and the official forecasts of the Reserve Bank of New Zealand. We choose a dynamic factor model, as it represents a robust current approach in economics to using a large number of predictors to forecast macroeconomic variables. We use the forecasts of the Reserve Bank of New Zealand as a benchmark, as they allows us to assess whether ML algorithms would be useful to the nowcast practitioner.

Our results show that the majority of the ML models produce point nowcasts that are superior to the simple AR

☆ We thank Christopher Ball, Punnoose Jacob, several anonymous referees, the editor, and seminar participants at the Reserve Bank of New Zealand, the BIS/Bank Indonesia, the University of Tasmania and the 2018 Australasian Macroeconomic Society meeting for their valuable comments. The views expressed here are the views of the authors and do not necessarily reflect the views of the Reserve Bank of New Zealand.

* Corresponding author.
E-mail addresses: Adam.Richardson@rbnz.govt.nz (A. Richardson), Thomas.vanFlorensteinMulder@rbnz.govt.nz (T. van Florenstein Mulder), Tugrul.Vehbi@bankofengland.gsi.gov.uk (T. Vehbi).

[1] This work was completed while Tuğrul was at the Reserve Bank of New Zealand. He is now affiliated with the Bank of England.

https://doi.org/10.1016/j.ijforecast.2020.10.005
0169-2070/

benchmark. The top-performing models—boosted trees, support vector machine regression, and neural networks—can reduce average nowcast errors by approximately 20%–30% relative to the AR benchmark. ML algorithms also outperform the dynamic factor model. In addition, an average of ML models appears to add value to the official forecasts produced by the Reserve Bank of New Zealand. The boosted trees model outperforms the Reserve Bank of New Zealand's forecasts.

This paper joins a growing body of literature that evaluates the relative success of ML models in forecasting over the more traditional time-series techniques. Chakraborty and Joseph (2017), Dopke, Fritsche, and Pierdzioch (2017), Sermpinis, Stasinakis, Theofilatos, and Karathanasopoulos (2014), and Cook and Hall (2017) use a variety of ML models to predict macroeconomic outcomes, and generally find that these methodologies improve over standard statistical methods. In contrast, Makridakis, Spiliotis, and Assimakopoulos (2018) compare the forecast accuracy of various popular ML algorithms with eight types of traditional statistical benchmarks and find that the out-of-sample forecasting accuracy of ML models is lower than that of more traditional statistical methods. Fornaro and Luomaranta (2019) nowcast Finish GDP using micro-data on firm turnover and traffic volumes. They find that a combination of statistical and ML models can improve the timeliness of GDP estimates without significantly sacrificing forecast accuracy. Several authors have also found that using ML models to predict asset prices and inflation tends to result in significant improvements in forecasting performance (e.g. Kim, 2003; McAdam & McNelis, 2005; Plakandaras, Gupta, Gogas, & Papadimitriou, 2015).

Our paper adds to this literature in two ways. First of all, we generally assess a wider range of ML models than the existing literature. In addition, to our knowledge, none of these papers focus on the real-time forecasting performance of these models. Our data vintages, on the other hand, enable us to test the forecast performance of these models in a fully real-time setting by capturing the revision properties of both the response variable and the predictors. Therefore, we also contribute to the literature by comparing the performance of these models under the exact conditions that the practitioner would use them.

Faust and Wright (2009) use a large real-time set of predictors and a range of time-series models to forecast US GDP and inflation. Similar to this paper, the data vintages are observed at the same time that US Federal Reserve Greenbook forecasts are produced, allowing for a fair comparison of statistical model forecasts and those produced by the practitioner. Their results show that both the Fed and time-series model forecasts for GDP fail to outperform a simple benchmark. Given that we use a large data set for New Zealand for our analysis, our paper is also related to Eickmeier and Ng (2011) who use the elastic net and ridge regression models (amongst other shrinkage methods) to forecast New Zealand's GDP from a large number of domestic and international predictors. They find that data-rich methods result in gains in forecast accuracy over common statistical methods using small data sets. Also, Matheson (2006) uses a factor model to produce real-time forecasts of New Zealand's GDP, inflation, interest rates, and exchange rates from a large number of predictors. This model has good forecast performance at longer-term horizons when compared to other statistical models. The remainder of this paper is as follows. Section 2 explains the models and the data used. Section 3 presents the results, and Section 4 concludes.

2. Empirical application

In this section, we provide a brief description of the various ML and benchmark models we considered for nowcasting GDP. Please refer to the original citations for greater detail. We focus more on the methodologies we use to choose key hyperparameters, and on the details of the real-time forecast exercise we use to assess nowcast performance.

2.1. Models

2.1.1. Autoregressive model (AR)

As a benchmark, we use a univariate AR model of order 1 for quarterly GDP growth (y_t):

$$y_t = \alpha_0 + \alpha_1 y_{t-1} + u_t, \tag{1}$$

where α_0 and α_1 are parameters, and u_t is the residual term.

2.1.2. Gradient boosting

Gradient boosting is a method used to build a high-quality predictor from a series of individual weak models. In this literature, these weak models are termed 'learners'. Generally, these algorithms start by fitting an initial model to the target variable, guided by a loss function (e.g., least squares, least absolute deviation, etc.). A subsequent model is added, with the loss function applied to the residuals of the models fitted so far. This process continues until a certain stopping criterion is reached. At a high level, we are iterating through multiple steps of the following:

$$F_m(x) = F_{m-1}(x) + \nu \Delta_m(x), \tag{2}$$

where $F_m(x)$ is the new model mapping x to the target, $F_{m-1}(x)$ is the previous model, $\Delta_m(x)$ is the weak learner, and ν is a shrinkage parameter. The parameter ν is termed the learning rate in this literature. It is used as a form of regularisation. This parameter scales the update of the model at each iteration. A value less than one acts as a form of shrinkage, and improves the generalisation of gradient boosting beyond the test set data. In our case, our 'learners' are a series of regression trees. Specifically, we use the *LS_TreeBoost* algorithm described and assessed in Friedman (2001). Given this model set-up, we have three choices of hyperparameters. These are the learning rate, the number of trees to include in the model (i.e., the 'number of learning cycles') and the characteristics of the underlying regression trees. We discuss in the next section how we choose these hyperparameters.

2.1.3. Ridge, lasso, and elastic net

The ridge, Lasso, and elastic net regression methods are commonly used regularisation techniques to help reduce model complexity when we have a large number of features in our data set. Ridge regression shrinks the coefficients of each of the features in the set by imposing a penalty on their size in the form of *L2* regularisation, where the penalty is on the squared magnitude of the coefficients. More formally, the ridge regression takes the following form:

$$\beta = argmin[\sum_{i=1}^{l}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} \beta_j^2], \quad (3)$$

where λ is a hyperparameter that determines the severity of the shrinkage applied, allowing some coefficients with a minor contribution to the response variable to get close to zero.

Lasso regression is very similar to ridge regression except that the Lasso uses *L*1 regularisation to create sparse models by imposing a penalty equal to the absolute value of the magnitude of the coefficients. The coefficient of the parameters can thus be driven to zero as well during the regularization process. Hence, Lasso can be used for feature selection and generating more parsimonious models. The Lasso regression is specified as follows:

$$\beta = argmin[\sum_{i=1}^{l}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p} |\beta_j|], \quad (4)$$

The elastic net regression contains a hybrid of the ridge and Lasso penalties and has the effect of both effectively shrinking coefficients and setting some coefficients to zero. The penalty parameter is therefore a convex sum of the ridge and Lasso penalties, and the relative weights of the two penalties are determined by an additional tuning variable α. The regression equation has the following form:

$$\beta = argmin[\sum_{i=1}^{l}(y_i - \beta_0 - \sum_{j=1}^{p} x_{ij}\beta_j)^2 + \lambda \sum_{j=1}^{p}(1-\alpha)\beta_j^2 + (\alpha)|\beta_j|], \quad (5)$$

Our method to choose λ and α, where relevant, is described in the next section.

2.1.4. Support vector machine regression (SVM)

The SVM algorithm aims to find a function $f(x)$ that has at maximum ε deviation from the targets, y_i, for all the training data, while being as flat as possible. In the case of a linear function, the function is specified as:

$$f(x) = (w, x) + b, \quad (6)$$

where w is the weight vector, x is the input vector, and b is the bias. Flat, in this case, means a small value of w. It is possible, however, that no such function exists to satisfy these conditions. Therefore, the algorithm allows for slack variables, ξ_i, which allow regression errors to exist up to the values of ξ_i and ξ_i^*. The goal of support vector regression is then given by:

$$min\ 0.5\|w\|^2 + C\sum_{i=1}^{l}(\xi_i + \xi_i^*) \quad (7)$$

$$subject\ to \begin{cases} y_i - (w, x_i) - b \le \varepsilon + \xi_i \\ (w, x_i) + b - y_i \le \varepsilon + \xi_i^*. \end{cases} \quad (8)$$

The constant term, C, is known as the regularisation parameter and controls the trade-off between minimising errors and penalising overfitting. The model can be made non-linear by mapping the input data into a linear space using a range of kernel functions. See Schölkopf and Smola (2003), Gunn et al. (1998), and Vapnik (1995) for greater detail. Overall, we need to choose values for the parameter C, the range of the slack variables (also called the size of the ξ_i insensitive band), and the form of the kernel function. Again, the choice of these variables is discussed in more detail below.

2.1.5. Neural network

A neural network is a process to map a series of inputs to a target output. It starts with a series of input data, and reduces these data down to a pre-specified number of nodes. These nodes are a series of weights of the input data, with an added bias. Additional 'layers' can be added to this process, where the nodes from the preceding layer become the input data for additional layers—where the preceding nodes are mapped to additional nodes through weights and an added bias. Eventually, a final set of nodes is mapped to the target output.

To implement the model, we need to choose the number of 'layers', the number of nodes in each layer, and the functional form of the mapping in each layer. In addition, we need to choose an algorithm to estimate the model and performance criteria to maximise, to choose weights and bias.

We opt for a relatively simple model, with two layers. One layer is a 'hidden' layer that maps the input data to ten nodes. This layer has a sigmoidal functional form. The other is an output layer that maps the nodes to the target, which has a linear form. We experimented with additional layers and node values. However, the forecast performance was similar in more complex models.

We use the Levenberg–Marquardt back propagation algorithm to estimate the model (Hagan & Menhaj, 1994). This algorithm iterates through an optimization routine to choose weights and bias within the model. The process continues for 1000 iterations of the algorithm, or until a minimum performance threshold is reached. This 'early stopping criterion' helps to ensure the model does not overfit the training data. In our case, the performance criterion is the mean squared error, with the minimum performance criterion set to 0.2.

The resulting model is dependent on the initial choice of weights. Therefore, to further improve the generalisation to unseen data, we run the model 50 times and average the results to reduce the risk that our nowcast is purely determined by the random initialisation of the weights.

2.1.6. Dynamic factor model

We estimate the approximate dynamic factor model outlined in Stock and Watson (2002) using the input data set used in our ML models. We use the procedure outlined in Bai and Ng (2001) to choose the appropriate number of factors and lags in the model. This results in a model with four factors, five factor lags, and one lag of the target variable.

2.2. Data

The data consist of a number of continuous real-time vintages of a range of macroeconomic and financial market statistics. These include New Zealand business surveys, consumer and producer prices, general domestic activity indicators (e.g., concrete production, milk-solids production, spending on electronic cards, etc.), domestic trade statistics, international macroeconomic variables, and international and domestic financial market variables. The data range from daily to quarterly, with the mean used to aggregate higher frequency data to quarterly data for model estimation. Each series is individually assessed, and either left in a level form or transformed to a quarterly percentage change depending on which form is likely to be more predictive of GDP growth. All data are standardised and seasonally adjusted.

The storage of the historical model run by the Reserve Bank of New Zealand (RBNZ) has allowed us to create 41 real-time vintages of these data. Every three months, the RBNZ publishes a *Monetary Policy Statement*. In working towards this publication, the RBNZ's staff put together an initial set of macroeconomic projections. The data banks containing the data described above were saved along with these projections each quarter. This process, therefore, gives us a quarterly real-time snapshot of a range of macroeconomic and financial market series. Conveniently, these snapshots were generally taken about four weeks before the release of the preceding quarter's GDP estimate. For example, the initial projections for the March 2015 *Monetary Policy Statement* would have been finalised on about the 20th of February. At this point, almost all of the key macroeconomic and financial market indicators for the December 2014 quarter would have been released—and it is at this point that the snapshot of these macroeconomic statistics has been saved. The December quarter GDP estimate was then released on the 19th of March. New Zealand does not produce flash estimates of GDP, so there is a significant lag between the end of the quarter and the publication of the GDP estimate.[2] The data available with each vintage differ somewhat, as data were added and removed from the RBNZ's data banks over time. After making the modifications described above, from a candidate of 668 series, we are left with between 532 and 634 series in each vintage.

In total, we have 41 real-time vintages of this data set, covering the period from 2009Q1 to 2019Q1. The data in each vintage begin in 1995Q1. These data vintages enable us to test how the forecast performance of these models compares under the conditions that the practitioner would use them, capturing the revision properties of the predictors and the target variable.

2.3. Model estimation and hyperparameter optimisation

In order to estimate the ML models and choose optimal hyperparameters, we divide the predictors and target data into training, validation, and test sets. In this set up, the model is estimated using the data in the training set. Optimal hyperparameters are then chosen by finding the parameters that minimise the forecast error in the validation set. We reserve the test set to assess the out-of-sample predictive accuracy of the fitted model, once hyperparameter optimization is complete.

In practice, the training and validation sets are combined, and we use k-fold cross-validation to optimise hyperparameters (Fig. 1). We divide the data into five random partitions. One partition is held back to be the validation set. The model is fitted on the remaining four partitions. We then specify the hyperparameters we would like to optimise, and the range to explore for each hyperparameter. We then use a search algorithm to find the set of hyperparameter values that minimises forecast errors on the partition that we held back. This process is then repeated, so each partition is used as a validation set. We use a Bayesian optimisation search algorithm to choose optimal hyperparameters through cross-validation. See Snoek, Larochelle, and Adams (2012) for further details.

The test set is excluded from this process, and is used to assess the out-of-sample forecast performance of the model, once hyperparameter optimisation is complete. To avoid overfitting the model to the training data set, we restrict the hyperparameter search to a range around values established in previous literature. This helps the models generalise to unseen data. Table 1 presents the hyperparameters we are optimising for each model, the range we are exploring, and the values that are chosen for the final vintage of data (2019Q1). We implement all models in MATLAB. Where a parameter is not specified, we are using the MATLAB default value.

Our real-time forecast performance exercise adds another dimension of complexity to this problem. To avoid look-ahead bias in our test of out-of-sample forecast performance, we have to repeat the model fit and hyperparameter optimisation process for each vintage of real-time data that we have. In essence, we are training the model on data that would have been available to the forecaster in real time and using it to assess the real-time prediction that the forecaster would have made. Fig. 1 outlines this process for the first vintage of data. Each model is fitted on data from 1995Q1 to 2008Q4. The estimated model is

[2] From 2015Q3, the data storage methodology was changed. The 'global database' containing 668 series (the contents of which are described above) was routinely saved when estimating the Bank's existing suite of statistical models. A version of this data set is saved at the end of the month following each *Monetary Policy Statement*. For example, for the May 2018 *Monetary Policy Statement*, a version of the data set was saved on the last working day of May. This data set contains most of the indicators up to the end of 2018Q1. The 2018Q1 GDP figures were then released on the 21 June 2018—around three weeks after the data snapshot was taken. This process gives us a set of data vintages from 2015Q3 to 2019Q1.

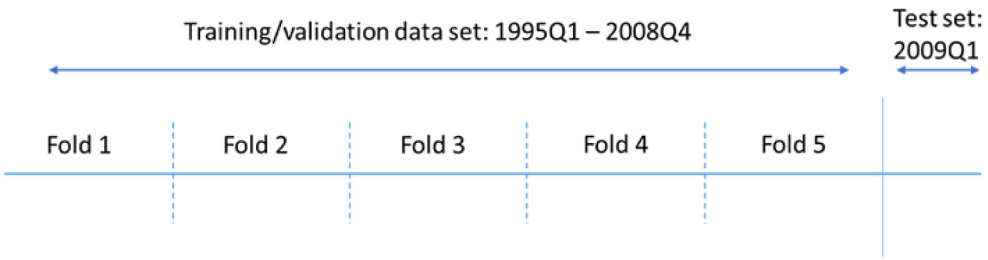

Fig. 1. Stylised representation of model estimation and forecast procedure..

Table 1
Hyperparameter ranges and values.

Model	Hyperparameter	Range	Optimised value
SVM	C	0.8 to 1.2	0.56
	Epsilon	iqr(Y)/13.49 +/− 20 percent	0.05
	Kernel function	Linear/Polynomial/RBF	Polynomial
	Polynomial order	2 to 4	2
LSboost	No. of learning cycles	500 to 1500	1498
	Learn rate	0.001 to 0.15	0.0075
	Max no. of splits	9 to 13	9
Elastic net	Alpha	0.1 to 0.9	0.5
	Lambda	0.1 to 0.9	0.1
Lasso	Lambda	0.001 to 0.9	0.011
Ridge	Lambda	0.01 to 0.99	0.989

then used to make a prediction of 2009Q1 GDP growth using the 2009Q1 indicator data that would have been available at that point in time. This estimate is then saved. The estimation window is then expanded by one quarter, and a new vintage of indicators and target variable are used to fit an additional set of models. This continues until we have 41 fully out-of-sample estimates of GDP growth, spanning from 2009Q1 to 2019Q1.

2.4. forecast evaluation methodology

We measure the forecast accuracy of each model by calculating the root mean squared error (RMSE), defined as:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(y_t - \hat{y}_t)^2}{T}} \quad (9)$$

where y_t and $\hat{y}_t$ are the actual and forecast values of GDP growth, respectively, and T is the total number of forecasts. The forecasts of a univariate (i.e., AR(1)) model provide the main benchmark for our comparisons. We use the Diebold–Mariano (Diebold & Mariano, 1995) test to determine whether the forecasts obtained from each ML model are significantly different than those from the AR model.

3. Empirical results

Table 2 documents the nowcast performance of the models and the official published nowcasts of the Reserve Bank of New Zealand for the sample period 2009Q1–2019Q1. In addition to the models outlined in Section 2.1, we also present the results obtained by combining the forecasts from all the ML models using an equal weighting strategy.

Table 2
Real-time nowcast performance of models and Reserve Bank (RMSE), 2009Q1–2019Q1.

Models	RMSE	RMSE (Rel. to AR)	*p*-value
LSBoost	0.3571	0.719	0.028
SVM	0.3796	0.764	0.049
Neural net	0.4070	0.819	0.127
Lasso	0.4387	0.883	0.310
Elastic net	0.4280	0.861	0.272
Ridge	0.4374	0.880	0.331
RBNZ	0.3673	0.739	0.037
Model average	0.3806	0.766	0.055
Factor	0.6010	1.209	0.178
AR	0.4970		

Notes: The fourth column refers to the p-values obtained from the Diebold–Mariano test of the significance of the forecast accuracy of each method versus that of the AR model.

The results indicate that all of the ML models produce forecasts that have RMSEs lower than the AR benchmark. The top three models based on RMSEs are the boosted trees, support vector machine regression, and neural network algorithms, which are able to reduce the average forecast errors by approximately 20%–30% relative to the AR benchmark. The Diebold and Mariano (1995) test statistic indicates these improvements are significant for the boosted trees and support vector machine regression algorithms. The relative success of the neural network and support vector machine models is in line with previous findings in the literature (Ahmed, Atiya, Gayar, & El-Shishiny, 2010; Teräsvirta, van Dijk,

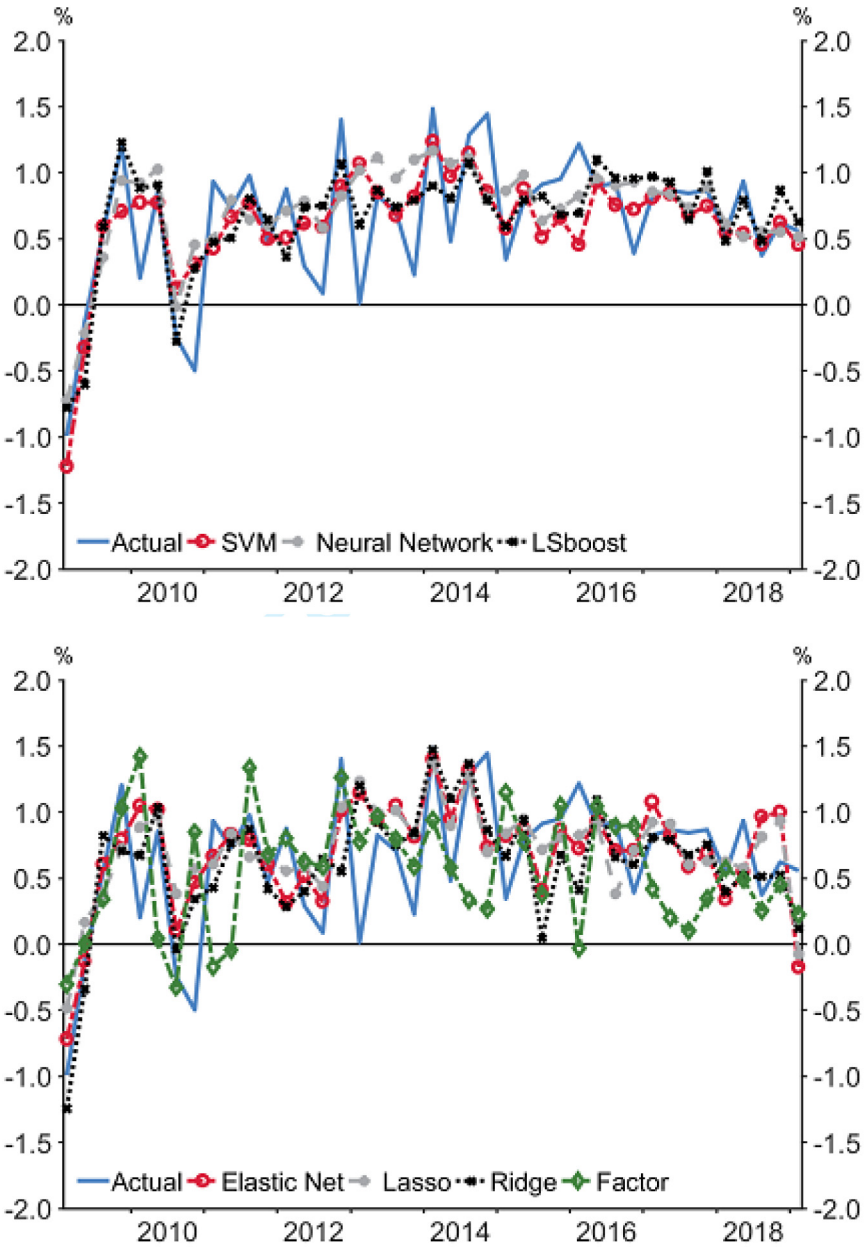

Fig. 2. Real-time nowcasts of quarterly GDP growth.

& Medeiros, 2005). All of the ML models are also able to produce RMSEs lower than the dynamic factor model. In addition, the boosted trees algorithm produces more accurate forecasts than the Reserve Bank of New Zealand. An alternative approach is to combine forecasts from the set of all models under consideration. Using an equal weighting strategy and combining the six ML models results in a significant increase in forecast performance relative to the AR benchmark.[3]

Fig. 2 presents quarterly GDP growth and its nowcasts obtained from each model over the sample period. It can be seen that all ML models successfully predicted the sharp downturn in activity that occurred in the first quarter of 2009 and also predicted the other major upturns and downturns in the GDP data successfully.

Furthermore, we investigate whether the ML model nowcasts add value to the nowcasts generated by the Reserve Bank of New Zealand for the quarterly *Monetary Policy Statement*. The Reserve Bank nowcasts are produced

[3] We also calculated the inverted and rank RMSE weighted combinations in a real-time way—dropping the first set of out-of-sample forecasts, and weighting each subsequent set of forecasts with the RMSE information available up to that vintage. The improvements over a simple average are minor. These results are available upon request.

Table 3
Estimated value of alpha and significance from Eq. (3).

Models	Estimated value	*p*-value
ML average	0.540	0.083
LSBoost	0.572	0.017
Neural net	0.226	0.444
Model average	0.377	0.236
SVM	0.371	0.205
Lasso	0.070	0.796
Ridge	0.149	0.472
Elastic net	0.100	0.690
Factor	−0.050	0.733

using a combination of statistical models (McDonald and Thorsrud (2011), bottom up sector based indicators, and expert judgement by sectoral analysts.

To test this formally, we follow the approach by D. Romer and H. Romer (2008) and estimate the following regression equation:

$$y_t = c_i + \alpha_i FC_i + \beta_i RB + e_t, \tag{10}$$

where y_t is the actual real GDP growth, *FC* is the real-time nowcast for GDP growth produced by model *i*, *RB* is the real-time GDP growth nowcast produced by the Reserve Bank of New Zealand, and e_t is the residual term. The values of α for each model are presented in Table 3. The results show that, if the Reserve Bank of New Zealand had added the boosted trees algorithm to its suite of nowcast tools, it could have improved its forecast accuracy. We also test whether the models that would be least familiar to the macroeconomist would have helped add value to the Reserve Bank forecast process—namely, the boosted trees algorithm, the neural network, and support vector machine regression. Taking a simple average of the forecasts of these models ('ML Average' in Table 3) would have also helped improve the forecast accuracy of the Reserve Bank of New Zealand.

4. Conclusion

In this paper, we evaluated the real-time performance of popular ML algorithms in obtaining accurate nowcasts of real GDP growth for New Zealand. We estimated several ML models over the 2009–2019 period using multiple vintages of historical GDP data and multiple vintages of a large features set comprising approximately 600 domestic and international variables. We then compared the forecasts obtained from these models with the forecasting accuracy of a naive AR benchmark as well as a dynamic factor model and the official forecasts produced by the Reserve Bank of New Zealand. We found that all of the ML models were able to produce more accurate forecasts than those of the AR and dynamic factor models. The results also suggest that the Reserve Bank of New Zealand's forecast accuracy could have been improved through the use of ML models. Our results thus recommend the use of ML algorithms as complementary tools to help policy makers understand the current state of the economy.

References

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

Bai, J., & Ng, S. (2001). Determining the number of factors in approximate factor models. *Econometrica*, *70*.

Banbura, M., Giannone, D., Modugno, M., & Reichlin, L. (2013). Now-casting and the real-time data flow. *Handbook of Economic Forecasting*, *2*(Part A), 195–237.

Bloor, C. (2009). The use of statistical forecasting models at the Reserve Bank of New Zealand. *Reserve Bank of New Zealand Bulletin*, *72*, 21–26.

Chakraborty, C., & Joseph, A. (2017). *Machine learning at central banks*: *Bank of England working papers 67*, Bank of England.

Cook, T. R., & Hall, A. S. (2017). Macroeconomic indicator forecasting with deep neural networks. Federal Reserve Bank of Kansas City, Research Working Paper 17–11.

D. Romer, C., & H. Romer, D. (2008). The FOMC versus the staff: Where can monetary policymakers add value?. *American Economic Review*, *98*, 230–235.

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

Dopke, J., Fritsche, U., & Pierdzioch, C. (2017). Predicting recessions with boosted regression trees. *International Journal of Forecasting*, *33*(4), 745–759.

Eickmeier, S., & Ng, T. (2011). Forecasting national activity using lots of international predictors: An application to New Zealand. *International Journal of Forecasting*, *27*(2), 496–511.

Faust, J., & Wright, J. H. (2009). Comparing Greenbook and reduced form forecasts using a large realtime dataset. *Journal of Business & Economic Statistics*, *27*(4), 468–479.

Fornaro, P., & Luomaranta, H. (2019). Nowcasting Finnish real economic activity: a machine learning approach. *Empirical Economics*, *58*, 55–71.

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

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

Gunn, S. R., et al. (1998). Support vector machines for classification and regression. *ISIS Technical Report*, *14*(1), 5–16.

Hagan, M., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, *5*(6), 989–993.

Jansen, W. J., Jin, X., & de Winter, J. M. (2016). Forecasting and nowcasting real GDP: Comparing statistical models and subjective forecasts. *International Journal of Forecasting*, *32*(2), 411–436.

Kim, K. (2003). Financial time series forecasting using support vector machines. *Neurocomputing*, *55*(1), 307–319.

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLoS One*, *13*(3).

Matheson, T. D. (2006). Factor model forecasts for New Zealand. *International Journal of Central Banking*, *2*(2).

McAdam, P., & McNelis, P. (2005). Forecasting inflation with thick models and neural networks. *Economic Modelling*, *22*(5), 848–867.

McDonald, C., & Thorsrud, L. A. (2011). *Evaluating density forecasts: model combination strategies versus the RBNZ*: *Reserve bank of new zealand discussion paper series*, Reserve Bank of New Zealand.

Plakandaras, V., Gupta, R., Gogas, P., & Papadimitriou, T. (2015). Forecasting the U.S. real house price index. *Economic Modelling*, *45*, 259–267.

Schölkopf, B., & Smola, A. J. (2003). A short introduction to learning with kernels. In *Advanced lectures on machine learning* (pp. 41–64). Springer.

Sermpinis, G., Stasinakis, C., Theofilatos, K., & Karathanasopoulos, A. (2014). Inflation and unemployment forecasting with genetic support vector regression. *Journal of Forecasting*, *33*(6), 471–487.

Snoek, J., Larochelle, H., & Adams, R. P. (2012). Practical Bayesian optimization of machine learning algorithms. In F. Pereira, C. J. C. Burges, L. Bottou, & K. Q. Weinberger (Eds.), *Advances in neural information processing systems, Vol. 25* (pp. 2951–2959). Curran Associates, Inc., http://papers.nips.cc/paper/4522-practical-bayesian-optimization-of-machine-learning-algorithms.pdf.

Stock, J. H., & Watson, M. W. (2002). Macroeconomic forecasting using diffusion indexes. *Journal of Business & Economic Statistics*, *20*(2), 147–162.

Teräsvirta, T., van Dijk, D., & Medeiros, M. C. (2005). Linear models, smooth transition autoregressions, and neural networks for forecasting macroeconomic time series: A re-examination. *International Journal of Forecasting*, *21*(4), 755–774.

Vapnik, V. N. (1995). *The nature of statistical learning theory*. Berlin, Heidelberg: Springer-Verlag.