Discussion Paper 2019-03

RESERVE BANK
OF NEW ZEALAND
TE PŪTEA MATUA

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

November 2019

Adam Richardson, Thomas van Florenstein Mulder & Tuğrul Vehbi

See *The Journey of Te Pūtea Matua: Our Tāne Mahuta*, available at:
https://www.rbnz.govt.nz/about-us/the-journey-of-te-putea-matua-our-tane-mahuta

Discussion Paper 2019-03*

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

Adam Richardson, Thomas van Florenstein Mulder & Tuğrul Vehbi†

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 machine-learning algorithms are able to significantly improve over standard models used in economics to nowcast macroeconomic variables. We also show machine-learning algorithms have the potential to improve the official forecasts of the Reserve Bank of New Zealand.

Key words: Nowcasting; Machine learning; Forecast evaluation.

JEL classification: C52, C53, C55

*We thank Christopher Ball, Punnoose Jacob, an anonymous referee 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.

†Adam Richardson, Economics Department, Reserve Bank of New Zealand, 2 The Terrace, Wellington, New Zealand. E-mail: Adam.Richardson@rbnz.govt.nz.
Thomas van Florenstein Mulder, Economics Department, Reserve Bank of New Zealand, 2 The Terrace, Wellington, New Zealand. E-mail: Thomas.vanFlorensteinMulder@rbnz.govt.nz.
Tuğrul Vehbi, Bank of England, CAMA. E-mail: Tugrul.Vehbi@bankofengland.gsi.gov.uk. This work was completed while Tuğrul was at the Reserve Bank of New Zealand. He is now affiliated with the Bank of England.

ISSN 1177-7567 ©Reserve Bank of New Zealand.

Chapter 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 (Giannone et al. 2008; Banbura et al. 2013; Jansen et al. 2016 and Bloor 2009).

Machine learning (ML hereafter) 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 datasets 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 a large features set - 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 benchmark, a dynamic factor model, and the official forecasts of the Reserve Bank of New Zealand. We choose a dynamic factor models 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 it allows us to assess if machine learning 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 benchmark. The top-performing models - boosted trees, support vector machine regression and neural networks - can reduce average nowcast errors by approximately 20-23 per cent relative to the AR benchmark. The majority of the ML algorithms also outperform the dynamic factor model. In addition, an average of the the ML models appears to add value to the official forecasts produced by the Reserve Bank of New Zealand.

This paper joins a growing literature that evaluates the relative success of ML models in forecasting over the more traditional time-series techniques. Makridakis et al. (2018) compares the forecast accuracy of various popular ML algorithms with eight types of traditional statistical benchmarks and finds that the out-of-sample forecasting accuracy of ML models is lower than that of more traditional statistical methods. Chakraborty and Joseph (2017), on the other hand, conduct an out-of-sample forecasting exercise using UK data and argue that ML models generally outperform traditional modelling approaches in prediction tasks. Several authors have also found that using machine learning models to predict asset prices and inflation tend to result in significant improvements in forecasting performance (e.g., McAdam and McNelis 2005; Plakandaras et al. 2015 and Kim 2003). However, 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 contribute to the literature by comparing the performance of these models under the exact conditions which the practitioner would use them.

Given we use a large dataset 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 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 rate and exchange rate 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.

Chapter 2: Empirical Application

In this section, we provide a brief description of the various ML and benchmark models we considered for nowcasting GDP. We also discuss how we made key hyperparameter choices.[1]

2.1 Models

2.1.1 Autoregressive Model (AR)

As a benchmark, we use an 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 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 hyper-parameters. 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 largely follow the simulations and examples in 2001 in choosing these hyperparameters. We opt for 500 learners,

[1] We initially investigated using hyperparameter optimization through k-fold cross validation. However, this methodology proved computationally intensive, and resulted in poor out-of-sample forecast performance compared to adopting standard or default values for hyperparameters.

a learning rate of 0.1 and a number of terminal nodes of 11 (which governs the characteristics of the underlying regression trees). These hyperparameter choices are as close to optimal values established in 2001.

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 dataset. 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:

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

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 except that the Lasso uses $L1$ 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:

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

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:

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

In our application, we set λ to 0.1, reflecting the results of cross-validation . We setα equal to 0.5 in the case of elastic net regression, resulting in a model that is a balance of the Lasso and Ridge approach.

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. In the case of a linear function, the function is specified as:

$$f(x) = (w, x) + b, \tag{6}$$

where w is the weight vector, x is the input vector and b is the bias. The SVM algorithm aims to find this linear function while ensuring that it is as 'smooth' as possible. 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^*. Additionally, the cost function includes an additional term,w, that penalizes non-smoothness by minimising the coefficients on the explanatory variables. The goal of support vector regression is then given by:

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

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

The constant term, C, is known as the regularisation parameter and controls the trade-off between minimising errors and penalizing over fitting. For our modelling, we use linear epsilon-insensitive SVM (ϵ-SVM) regression with a polynomial kernel function of order 2. As is commonly used in the literature, we use an epsilon value of a tenth of the interquartile range of the target variable; a proxy for the noise level in the training sample. The hyperparameter C is set in a similar way using the interquartile range of the target variable, which is the default value in Matlab implementation of the algorithm. Following Schölkopf and Smola (2003), Gunn (1998), and Vapnik (1995), we solve this minimisation problem in its dual form.

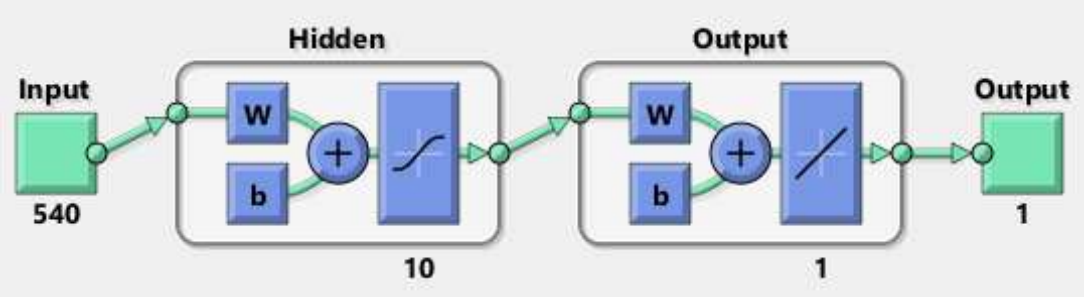
2.1.5 Neural Network

A neural network (specifically a feed-forward neural network in this paper) is a model that is able to capture and represent complex non-linear relationships. Figure 1 shows the structure of our estimated neural network model. The estimation of the model is an iterative process comprising several steps. The initial inputs into the model are the values of each of the explanatory variables, x. These inputs are then weighted using randomly initialised weights and transformed via a sigmoid function (the use of a sigmoid functional form allows for non-linearities in the relationship between the explanatory variables and the target). The weights are then changed and the process is repeated, up to a pre-specified number of nodes. The nodes in this first 'layer' then act as the new data set for the next 'layer', and this process is

repeated through a pre-specified number of layers. The node values in the penultimate layer are then weighted, bias is added, and transformed via a linear regression function to create the final prediction. Using the Levenberg-Marquardt back propagation algorithm, the portion of the forecast error attributed to each node in the model can be identified and used to update the weights to minimise forecast errors. This backpropogation process continues until either a maximum number (1000 in this case) of iterations is reached, or a minimum performance improvement threshold is obtained.

In our initial application, we set the number of total layers at 3 (this would be termed 1 output layer and 2 hidden layers in machine learning literature) and the number of nodes in layers one and two to 10 to capture the overall variation in the feature set. In practice, we find that a neural network with one hidden layer and 10 nodes has similar nowcasting performance and significantly reduces computation time. For each vintage, 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]

Figure 1
Neural Network



Source: MATLAB

2.1.6 Dynamic Factor Model

We estimate the factor model outlined in Matheson (2006). However, we modify the 2006 approach slightly by using the same data set used in the ML models - to allow for a like-for-like performance comparison.

The key parameter choice in the model is the number of factors to include. All of the criteria outlined in Bai and Ng (2001) identify 6 factors are suitable to summarise the data set. However, Matheson (2006) noted in practice that including 1-2 factors tended to produce better GDP forecasts. As a result, we conduct our real time forecast exercises with dynamic factor models with the number of factors ranging from 1-6. We only report the results of the model with the lowest RMSE - in this case a model with 5 factors.

[2] As a sensitivity check, we also tested models with 5 nodes, and models with an early stopping criteria linked to a minimum MSE and found similar out-of-sample results.

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 for model estimation. Each series is individually assessed, and either left in a level form or transformed to a quarterly percent change depending on which form is likely to be more predictive of GDP growth. All data are standardized and seasonally adjusted.

The storage of historical model runs by the RBNZ has allowed us to create 41 real-time vintages of these data. Every 3 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 down 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 quarters GDP estimate. For example, the initial projections for the March 2015 *Monetary Policy Statement* would have been finalized 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 the snapshot of these macroeconomic statistics has been saved. The December quarter GDP estimate was then released 19th 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.[3]The data available with each vintage differ somewhat, as data were added and removed from the RBNZ's data banks through time. After making the modifications described above, from a candidate of 668 series, we are left with between 532 to 634 series at each vintage.

In total, we have 41 real-time vintages of this dataset, covering the period 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 which the practitioner would use them - capturing the revision properties of the predictors and the target variable.

[3]From 2015Q3, the data storage methodology was changed. The 'global database' containing 668 series (the contents of which we described above) was routinely saved in 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 3 weeks after the data snapshot was taken. This process gives us a set of data vintages from 2015Q3 to 2019Q1.

2.3 Forecast evaluation methodology

We evaluate the performance of the models using an out-of-sample forecast exercise. We train each algorithm over an expanding window thereby replicating an actual forecasting situation starting from 2009Q1 and moving forward a quarter at a time through to 2019Q1. For example, for the first vintage of the data, the models are estimated over the period 1995Q1 to 2008Q4 using real-time data for both the predictors and quarterly GDP growth. The resultant fitted models are used to nowcast the 2009Q1 growth rate of real GDP. Overall, we generate 41 real-time nowcasts of quarterly GDP growth. Next, we measure the forecast accuracy of each model by calculating the Root Mean Square Error (RMSE) defined as:

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

where y_t and $\hat{y}_t$ are the actual and forecast values of GDP growth 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 and Mariano 1995)) test to determine whether the forecasts obtained from each ML model are significantly different than those from the AR model.

Chapter 3: Empirical Results

Table 1 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 1
Real-time nowcast performance of models and Reserve Bank (RMSE), 2009Q1-2018Q1

Models	RMSE	RMSE (Rel. to AR)	p-value
LSBoost	0.384	0.773	0.079
SVM	0.393	0.791	0.075
Neural Net	0.397	0.799	0.116
Lasso	0.425	0.855	0.247
Elastic Net	0.426	0.858	0.273
Ridge	0.483	0.973	0.817
RBNZ	0.367	0.739	0.037
Model Average	0.385	0.774	0.068
Factor	0.460	0.925	0.591
AR	0.497		

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.

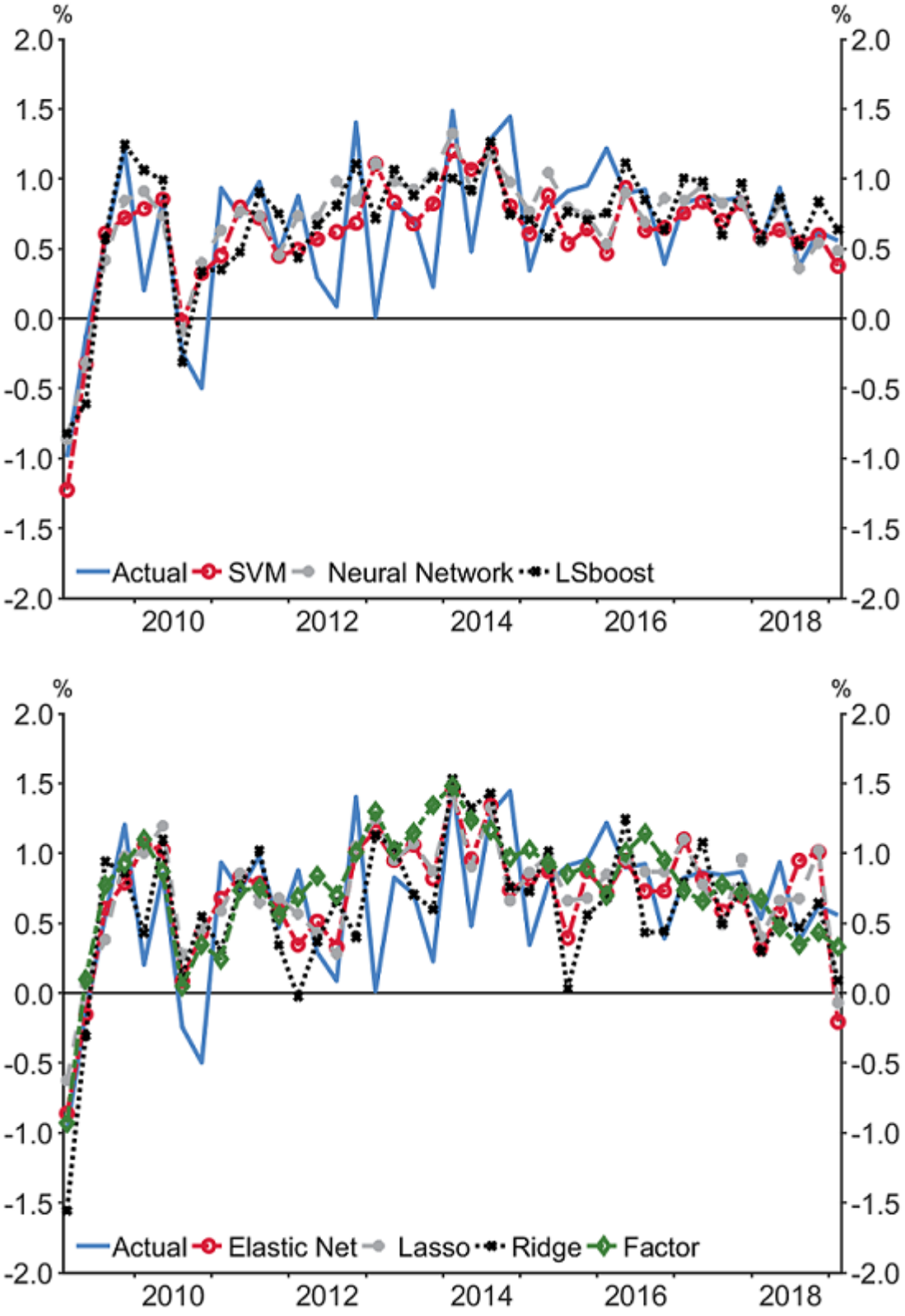
The results indicate that all of the ML models produce forecasts that have RMSEs lower than the AR benchmark. The top three models based on RMSEs are the boosted trees, support vector machine regression and neural network algorithms which are able to reduce the average forecast errors by approximately 20-23 percent relative to the AR benchmark. The 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 are in line with previous findings in the literature (Teräsvirta et al. 2005; Ahmed et al. 2010). The majority of the ML models are also able to produce RMSEs lower than the dynamic factor model. An alternative approach is to combine forecasts from the set of all models under consideration. Using equal weighting strategy and combining the 6 ML models results in a significant increase in forecast performance relative to the AR benchmark.[4]

Figure 2 presents quarterly GDP growth and its nowcasts obtained from each model over the sample period. It can be seen that all ML models have successfully predicted the sharp

[4]We also calculated the inverted and rank RMSE weighted combinations in a real-time way - dropping the first set of out of sample forecast, 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.

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.

Figure 2
Real-time nowcasts of quarterly GDP growth



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*. 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 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 2. 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 the support vector machine regression. Taking a simple average of the forecasts of these models (ML average in table 2) would have also helped improve the forecast accuracy of the Reserve Bank of New Zealand.

Table 2
Estimated value of beta and significance from equation 9

Models	Estimated value	p-value
ML average	0.457	0.089
LSBoost	0.429	0.052
Neural Net	0.359	0.137
Model Average	0.359	0.199
SVM	0.285	0.291
Lasso	0.147	0.576
Ridge	0.125	0.444
Elastic Net	0.119	0.629
Factor	0.012	0.960

Chapter 4: Conclusion

In this paper, we evaluate the real-time performance of popular ML algorithms in obtaining accurate nowcasts of real gross domestic product growth for New Zealand. We estimate 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 compare the forecasts obtained from these models with the forecasting accuracy of a naive autoregressive benchmark as well as a dynamic factor model and the official forecasts produced by the Reserve Bank of New Zealand. We find that most of the ML models are able to produce more accurate forecasts than those of the AR and dynamic factor models. The results also suggest 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, Nesreen K., et al. 2010. “An Empirical Comparison of Machine Learning Models for Time Series Forecasting”. *Econometric Reviews* 29 (5-6): 594–621.

Bai, Jushan, and Serena Ng. 2001. “Determining The Number Of Factors In Approximate Factor Models”. *Econometrica* 70 ().

Banbura, Marta, et al. 2013. “Now-casting and the real-time data flow”. *Handbook of economic forecasting* 2 (Part A): 195–237.

Bloor, Chris. 2009. “The use of statistical forecasting models at the Reserve Bank of New Zealand”. *Reserve Bank of New Zealand Bulletin* 72:21–26.

Chakraborty, Chiranjit, and Andreas Joseph. 2017. *Machine learning at central banks*. Bank of England working papers 674. Bank of England.

D. Romer, Christina, and David H. Romer. 2008. “The FOMC versus the Staff: Where Can Monetary Policymakers Add Value?” *American Economic Review* 98 (): 230–35.

Diebold, Francis X, and Roberto S Mariano. 1995. “Comparing Predictive Accuracy”. *Journal of Business & Economic Statistics* 13 (3): 253–263.

Eickmeier, Sandra, and Tim Ng. 2011. “Forecasting national activity using lots of international predictors: An application to New Zealand”. *International Journal of Forecasting* 27 (2): 496–511.

Friedman, Jerome H. 2001. “Greedy function approximation: a gradient boosting machine”. *Annals of statistics*: 1189–1232.

Giannone, Domenico, Lucrezia Reichlin, and David Small. 2008. “Nowcasting: The real-time informational content of macroeconomic data”. *Journal of Monetary Economics* 55 (4): 665 –676.

Gunn, Steve R, et al. 1998. “Support vector machines for classification and regression”. *ISIS technical report* 14 (1): 5–16.

Jansen, W. Jos, Xiaowen Jin, and Jasper M. de Winter. 2016. “Forecasting and nowcasting real GDP: Comparing statistical models and subjective forecasts”. *International Journal of Forecasting* 32 (2): 411 –436.

Kim, Kyoung. 2003. “Financial time series forecasting using support vector machines”. *Neurocomputing* 55 (1): 307 –319. ISSN: 0925-2312.

Makridakis, Spyros, Evangelos Spiliotis, and Vassilios Assimakopoulos. 2018. “Statistical and Machine Learning forecasting methods: Concerns and ways forward”. *PloS one* 13 (3).

Matheson, Troy D. 2006. “Factor Model Forecasts for New Zealand”. *International Journal of Central Banking* 2 (2).

McAdam, Peter, and Paul McNelis. 2005. “Forecasting inflation with thick models and neural networks”. *Economic Modelling* 22 (5): 848 –867.

Plakandaras, Vasilios, et al. 2015. “Forecasting the U.S. real house price index”. *Economic Modelling* 45:259 –267.

Schölkopf, Bernhard, and Alexander J Smola. 2003. “A short introduction to learning with kernels”. In *Advanced lectures on machine learning*, 41–64. Springer.

Teräsvirta, Timo, Dick van Dijk, and Marcelo C. Medeiros. 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. ISSN: 0169-2070.

Vapnik, Vladimir N. 1995. *The Nature of Statistical Learning Theory*. Berlin, Heidelberg: Springer-Verlag. ISBN: 0-387-94559-8.