IJSA (2021), Vol. 5, No. 2, 355-368
DOI: https://doi.org/10.29244/ijsa.v5i2p355-368

Indonesian Journal of Statistics and Its Applications
Available at https://ijsa.stats.id
e-ISSN: 2599-0802

Nowcasting Indonesia's GDP Growth Using Machine Learning Algorithms*

N D Muchisha[1], N Tamara[2], Andriansyah[3], and A M Soleh[4‡]

[1,2,4]Department of Statistics, IPB University, Indonesia
[1,2]Badan Pusat Statistik, Indonesia
[3]Fiscal Policy Agency, Ministry of Finance, Indonesia
‡corresponding author: agusms@apps.ipb.ac.id

Abstract

GDP is very important to be monitored in real time because of its usefulness for policy making. We built and compared the ML models to forecast real-time Indonesia's GDP growth. We used 18 variables that consist a number of quarterly macroeconomic and financial market statistics. We have evaluated the performance of six popular ML algorithms, such as Random Forest, LASSO, Ridge, Elastic Net, Neural Networks, and Support Vector Machines, in doing real-time forecast on GDP growth from 2013:Q3 to 2019:Q4 period. We used the RMSE, MAD, and Pearson correlation coefficient as measurements of forecast accuracy. The results showed that the performance of all these models outperformed AR (1) benchmark. The individual model that showed the best performance is random forest. To gain more accurate forecast result, we run forecast combination using equal weighting and lasso regression. The best model was obtained from forecast combination using lasso regression with selected ML models, which are Random Forest, Ridge, Support Vector Machine, and Neural Network.

Keywords: gdp growth, machine learning, nowcasting.

* Received: Jan 2021; Reviewed: Jan 2021; Published: Jun 2021

1. Introduction

Gross domestic product (GDP) is useful for measuring the rate of national economic growth, comparing economic progress between countries, and knowing the economic structure of a country. In Indonesia this figure is measured by Statistics Indonesia (BPS), but the release of data is delayed 5 weeks after the end of each quarter. Therefore, forecasting real-time GDP for the current quarter is important to plan short-term economic policies while waiting for the GDP to be released.

Time series analysis includes the development of models to describe the time series observational data and gather information contained in that data. Time series forecasting uses the best fitting model to predict future observations based on consideration and data patterns of previous observations. Currently, time series models using machine learning (ML) techniques are used as an alternative to time series regression models. The advantage of ML techniques is that it is more effective in capturing the patterns in the sequence structured and unstructured data, and its further analysis for accurate predictions. ML models have attracted attention and have proven themselves to be serious competitors of classical statistical models in the forecasting studies.

Several studies have shown good results in the use of ML models in forecasting time series data in the economic field. Richardson et al. (2018) found that most ML models are able to produce more accurate estimates than autoregressive (AR) model and other statistical benchmarks, such as factor model and Bayesian VAR, on nowcasting New Zealand GDP growth from 2009-2018. Adriansson & Mattsson (2015) showed that Random Forest proved to have a better performance than the linear bridge model and AR (1) benchmark. Chakraborty & Joseph (2017) found that ML models generally outperform traditional modelling approaches in prediction cases.

We evaluate some ML algorithms performance in forecasting real-time GDP growth. We use some vintage historical GDP growth data and 18 variables that available in quarterly period to evaluate the real-time performance during 2013:Q3 to 2019: Q4. We employ some popular ML modelling approaches, such as Random Forest, LASSO, Ridge, Elastic Net, Neural Networks, and Support Vector Machine. To evaluate the ML models, we compare their forecast accuracy performances relative to AR benchmark.

This paper is aimed to build the best ML model in forecasting real-time Indonesia's GDP growth by comparing the RMSE, MAD, and Pearson Correlation Coefficient values. First, we built an individual model of six ML algorithms. Then we do a combination of forecasting from the best individual models to get better forecasting accuracy performance. The best model is chosen from the model that shows the outperform performance compared to other models.

2. Materials and Methods

2.1 Data

Dataset used in this research was for the period from 2009:Q4 to 2019:Q4. It contained reference series quarterly GDP Indonesia's growth. We used 18 predictor variables that consist a number of quarterly macroeconomic and financial market statistics. The

details can be seen in appendix. In making the forecasting models for quarterly GDP growth, we used 19 variables contained in the dataset for lags 1, 2, 3, 4, and 8. Thus, there were 95 features in the model, as shown in Table 1. We did this in order to do features engineering to add more predictive information for modelling GDP growth.

Table 1. Features Dataset.

	y					x^1					…	x^{18}				
t	$t-1$	$t-2$	$t-3$	$t-4$	$t-8$	$t-1$	$t-2$	$t-3$	$t-4$	$t-8$	…	$t-1$	$t-2$	$t-3$	$t-4$	$t-8$
	(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)	…	(91)	(92)	(93)	(94)	(95)
1																
2																
…																
n																

We created 26 new datasets from the initial dataset to forecast GDP growth from 2013:Q3 to 2019:Q4. The dataset was created over an expanded window, from the initial period (2009: Q3) to 1 period before the period for which to forecast. For example, to forecast the 2013:Q3 period, we modelled using a dataset from 2009:Q4 to 2013:Q2. To forecast the 2013:Q4 period, we modelled using a dataset from 2009:Q4 to 2013:Q3. And so on, until forecasting for the period 2019:Q4.

2.2 Models

2.2.1. Autoregressive Model.

We used a univariate Autoregressive/AR (1) model as our simple benchmark for quarterly GDP growth (y_t). The model is below:

$$y_t = \phi_0 + \phi_1 y_{t-1} + e_t$$

where ϕ_0 and ϕ_1 are parameters, and e_t is the residual term.

2.2.2. Ridge Regression.

Ridge regression is one of the shrinkage methods that very similar to ordinary least squares (OLS). This method was designed to overcome the instability of the least square's estimator by penalizing the coefficients on L_2-norm. Its coefficient estimates $\hat{\beta}^{Ridge}$ are the values that minimize:

$$\sum_{t=1}^{n}\left(y_t - \beta_0 - \sum_{j=1}^{p}\beta_j x_{tj}\right)^2 + \lambda\sum_{j=1}^{p}{\beta_j}^2$$

where $\lambda > 0$ as a tuning parameter, which controls the strength of penalty term. p is the number of features used in modelling. In this study, $p = 95$. x_{tj} are components of the feature dataset as shown in Table 1. $\lambda\sum_{j=1}^{p}{\beta_j}^2$, called shrinkage penalty, is small when $\beta_1, \ldots, \beta_k$ are close to zero. The effect of shrinkage penalty is shrinking the estimates of β_j towards zero.

2.2.3. Lasso.

Lasso (Least Absolute Shrinkage and Selection Operator) was first introduced by Tibshirani (1996). Lasso uses L_1 penalty to shrinking the coefficients. The Lasso coefficients, $\hat{\beta}^{Lasso}$, are the values that minimize:

$$\sum_{t=1}^{n}\left(y_t-\beta_0-\sum_{j=1}^{p}\beta_j x_{tj}\right)^2+\lambda\sum_{j=1}^{p}\left|\beta_j\right|$$

We see that the Lasso formula has similar formulation like ridge. The differences only on the penalty used. Lasso shrinkage penalty is $\lambda\sum_{j=1}^{p}\left|\beta_j\right|$. The impact that occurs by changing this penalty is very large, which causes the coefficients shrink towards zero as in the ridge regression and some coefficients produce a zero value appropriately. This allows Lasso can be used for selecting variables.

2.2.4. Elastic Net.

Elastic net (ENET) is a method that combines the L_1 and L_2 penalties of the Lasso and ridge methods. It improves some limitations on Lasso when the number of parameters is greater than the number of observations ($p > n$) and on the problem of grouping variables (Zou & Hastie, 2005). The ENET coefficients, $\hat{\beta}^{EN}$, can obtained by minimizing this formula:

$$\sum_{t=1}^{n}\left(y_t-\beta_0-\sum_{j=1}^{p}\beta_j x_{tj}\right)^2+\lambda\sum_{j=1}^{p}\left[(1-\alpha){\beta_j}^2+\alpha\left|\beta_j\right|\right]$$

Elastic net is the same as Lasso when $\alpha = 1$, and the same as ridge when $\alpha = 0$.

2.2.5. Random Forest.

Random forest (RF) is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest (Breiman, 1999). It is a substantial modification from bagging technique, that there is no interaction between trees while building random forest. RF can be used for classification and regression modelling. If it uses for regression, RF take the average from all predictions from trees. The illustration of RF algorithm can be seen in Figure 1.

The base learner used in RF is regression trees. RF combine many binary regression trees, built using several bootstrap samples on dataset that consist a response and *p* inputs, for each of *N* observations. Let us consider a learning set L consists of (x_i, y_i) for $i = 1, 2, \ldots, N$, with $x_i = \left(x_{i1}, x_{i2}, \ldots, x_{ip}\right)$. To grow regression tree, the algorithm needs to decide on the splitting variables and splitting points, and also what shape the tree should have. The steps carried out by the algorithm are as follow (Hastie et al., 2017):

1. Suppose first we have M regions that partitioned the dataset L into $R_1, R_2, \ldots, R_M$.
2. The model of the response is a constant value c_m in each region:

$$f(x)=\sum_{m=1}^{M}c_m I(x\in R_m)$$

3. Predicted value of c_m obtained by averaging y_i in region R_m:

$$\hat{c}_m = \text{ave}(y_i | x_i \in R_m)$$

The best binary partition in regression tree found by trying different threshold values, and selecting the threshold that has minimum sum of squares. For example, for region R_1 and R_2, we seek the splitting variable *j* and splitting point *s* that solve:

$$\min_{j,s}\left[\min_{c_1}\sum_{x_i \in R_1(j,s)}(y_i - c_1)^2 + \min_{c_2}\sum_{x_i \in R_2(j,s)}(y_i - c_2)^2\right]$$

After found the best split, the dataset is the partitioned into two resulting subsets. Then the process continues until each node reaches a user-specified minimum *nodesize* and becomes a terminal node.

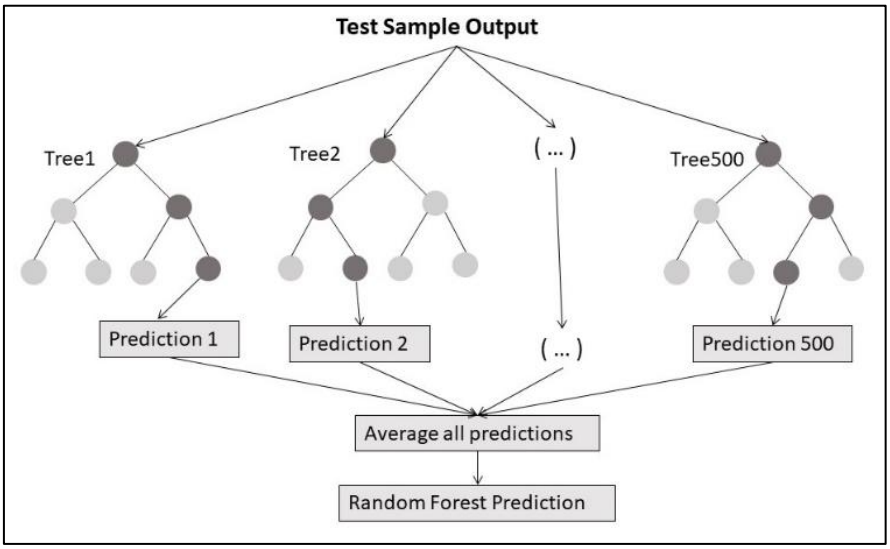

Figure 1. Random Forest Algorithm Illustration.

2.2.6. Neural Network.

Neural Network (NN) theory is a form in which the output is performed by a predetermined non-linear function on multiple inputs (Lee et al., 2017). All neurons in the neural network model are divided into an input layer, a hidden layer, and an output layer depending on the function, and each layer is functionally connected. The input layer connects the external input mode and is transmitted in units of hidden layers according to the input unit. The hidden layer is the inner processing unit layer of the neural network and the output layer is used to generate the output model.

NN architecture is divided into two parts; single layer network and multiple layer network. Models of multiple layer network's category such as backpropagation. Backpropagation network has 3 phases: advance phase, reverse phase, and weight modification phase to decrease error that might occur. The weights are initially set with random values and are updated on each iteration using this algorithm.

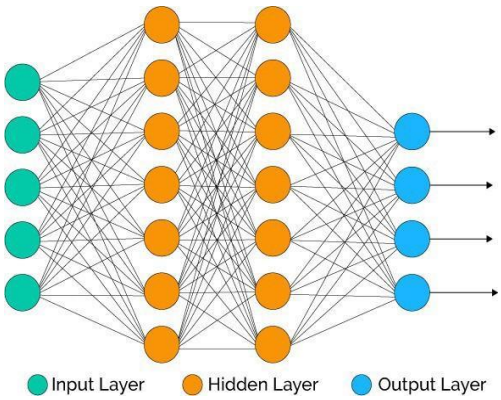

Figure 2. Neural Network that outputs Result Value through Two Hidden Layer.

In the Figure 2, each circular node represents an artificial neuron, and an arrow represents the input from one neuron output to another neuron. Each node of the first layer then becomes the new input variable for layer 2 and gets reweighted. The nodes in the last layer passed through an activation function to give the output value.

2.2.7. Support Vector Machine.

Support Vector Machine (SVM) first proposed by Vapnik (1995) can also be used for regression. The main idea is to minimize error, individualizing the hyperplane which maximizes the margin, keeping in mind that part of the error is tolerated. The hyperplane is a linear function of the form:

$$f(x) = wx + b$$

where w is the weight vector, x is the input vector, and b is the bias. In order to maximize the margin, we need to minimize:

$$0.5\|w\|^2 + C\sum_{i=1}^{l} |(y_i - f(x_i))|_\epsilon$$

where l is the sum of training points, $C > 0$ is the regularization parameter that constrains/ regularizes or shrinks the coefficient estimates towards zero. The first term in the error function is a penalty term that increases as the model becomes more complex. The second term is the ϵ-insensitive loss function that penalises errors that are greater than ϵ, allowing flexibility to the model. For transform the data into a higher dimensional feature space to make it possible to perform the linear separation, we use gaussian radial basis function is according to this formula:

$$k(x_i, x_j) = \exp\left(\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

2.3 Forecast Evaluation

We evaluated the performance of the models by looking at the performance in forecasting GDP growth on the samples from 2013:Q3 to 2019:Q4. We trained each algorithm methods over an expanding window to forecast real-time GDP growth from 2013:Q3 to 2019:Q4. The illustration can be seen in Figure 3. For example, for the first real-time nowcast on 2013:Q3, we used dataset from 2009:Q4 to 2013:Q2. For the second real-time nowcast on 2013:Q4, we used dataset from 2009:Q4 to 2013:Q3. This is done until the last out-of-sample period. Overall, we generated 26 nowcasts of quarterly GDP growth. Then, we calculated the Root Mean Square Error (RMSE) and Mean Absolute Deviance (MAD) measurement for each model with the following formula:

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

$$MAD = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|$$

where y_t and $\hat{y}_t$ are the actual and forecast values of GDP growth, and n is the total number of forecasts. In addition, we also calculate Pearson correlation coefficient values to see the closeness of the pattern between the actual and forecast values, with the following formula:

$$r_{y\hat{y}} = \frac{n\sum_{t=1}^{n}(y_t\hat{y}_t) - (\sum_{t=1}^{n} y_t)(\sum_{t=1}^{n}\hat{y}_t)}{\sqrt{n\sum_{t=1}^{n} {y_t}^2 - (\sum_{t=1}^{n} y_t)^2}\sqrt{n\sum_{t=1}^{n}\hat{y}_t{}^2 - (\sum_{t=1}^{n}\hat{y}_t)^2}}$$

The RMSE, MAD, and Pearson correlation coefficient are then compared with the results of our simple benchmark AR (1).

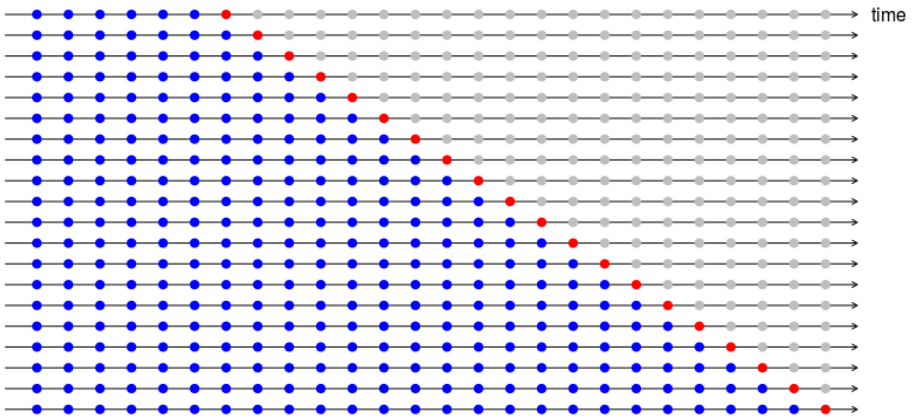

Figure 3. Expanding Window Illustration.

3. Results and Discussion

The following section describes the main results of our analysis. We present the performance of each ML models in conducting real-time forecasting on Indonesia's GDP growth data in the 2013:Q3-2019:Q4 period. We also present the results of forecasting by combining several ML models.

3.1 Forecasting Results

Table 2 shows the results of the nowcast performance of the models for the sample period from 2013:Q3 to 2019:Q4. We use RMSE and MAD measurements to see the closeness of the values between actual and forecast values. Meanwhile, to see the proximity of the pattern, we use Pearson correlation coefficient. All ML models produce lower RMSE and MAD values than AR benchmark. ML models are able to reduce the average forecast errors around 38-63% relative to AR benchmark. While the correlation coefficient value, the ML models are able to produce a pattern that is very close between the actual and forecasting values. This is indicated by the large correlation coefficient, which is above 0.7. Whereas the correlation coefficient of AR actually shows a mismatch of patterns between actual and forecast values, and it produces a minus coefficient of Pearson correlation.

Table 2. Real-Time Nowcast Performances of Models (2013:Q3-2019:Q4).

Models	RMSE	MAD	Pearson Correlation Coefficient
AR	2.725	2.549	-0.279
RF	1.273	0.923	0.886
LASSO	1.703	1.108	0.765
RIDGE	1.622	1.400	0.858
ENET	1.312	0.989	0.865
SVM	1.352	0.973	0.853
NN	1.358	1.034	0.878

The results of the RF model outperform the results of other ML models as seen from the value of RMSE, MAD, and the correlation coefficient which is in the first position. In fact, the RMSE value dropped by more than 50% relative to AR benchmark. This excellent performance could be due to the RF model based on ensemble learning which takes the average prediction results from all trees in the forest. So, it can improve the results of predictions. The lowest performance among other ML models is shown by LASSO. This result is different from previous finding, where LASSO occupies the 3rd best position in performing nowcast GDP growth (Richardson, Mulder, & Vehbi, 2018). Other results from Table 2 show that the ML models that occupy the top four positions if inferred from the RMSE, MAD values, and the correlation coefficients are RF, ENET, SVM, and NN.

From Figure 4 we can see that in general, forecast results from ML models have patterns that is in line with their actual values. RF plot shows pattern like a perfect 45-degree straight line, and the spread of points is not too scattered. LASSO actually also shows a 45-degree straight line pattern, it's just that there are two observations far from the other point patterns. This is what causes the correlation coefficient of LASSO is the lowest.

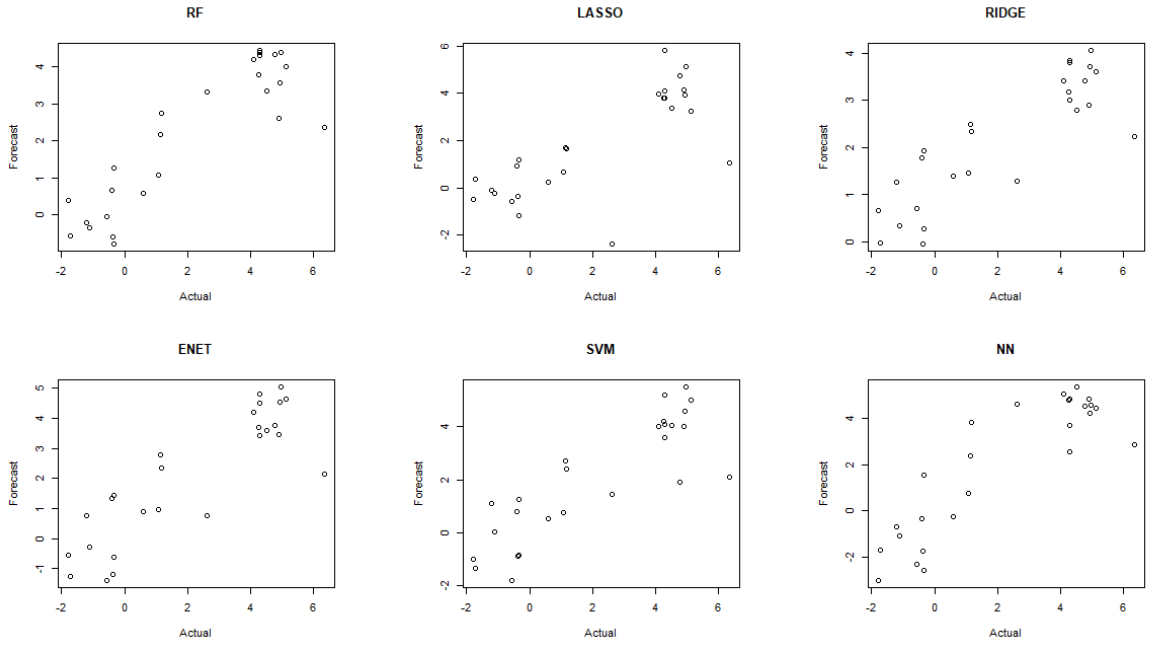

Figure 4. Scatter Plot Actual and Forecast Values for Each ML Methods.

From the time series plot of actual and forecast values in Figure 5, we can see that all ML models can predict the increase and decrease that occurs in the actual data. It's just that the LASSO and RIDGE are not too good in predicting the magnitude of the increase or decrease in GDP growth during the 2013:Q3-2019:Q4 period. RF and ENET are able to produce predictions that are very close to the actual data. It appears that the plot between the forecast result and the actual value almost coincides. From the results presented in this section, it can be seen that the ML model occupying the top three positions based on the RMSE and MAD are RF, ENET, and SVM. While the top three positions based on the correlation coefficient are RF, NN, and ENET.

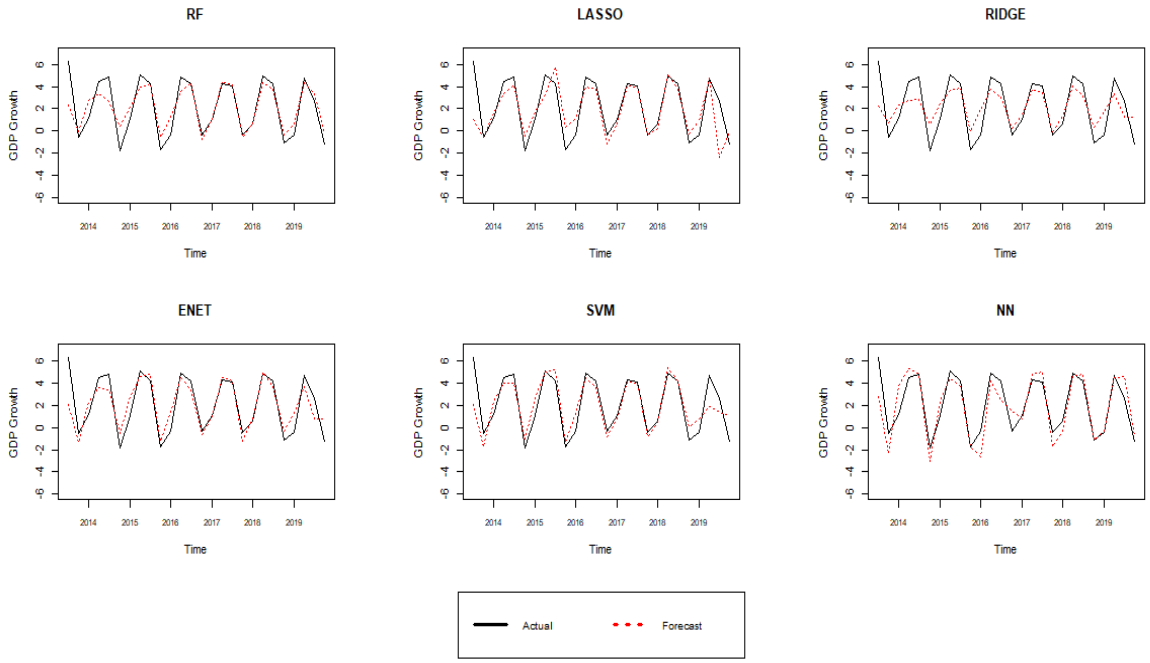

Figure 5. Real-Time Nowcasts of Quarterly GDP Growth for Each ML Methods.

3.2 Forecast Combination

In the previous section, we compared the forecast results from individual models based on RMSE, MAD, and correlation coefficient values. In this section, we strive to improve forecasting results by combining forecast results from several models. Forecast combining has often been done in empirical studies to produce better forecasting accuracy. In fact, Clemen (Clemen, 1989) says that combining forecasts should be part of the mainstream of forecasting practice.

We used two types of forecast combinations, namely equal weighting and Lasso Regression. For forecast combination using equal weighting, we did five combinations using the top four best models generated from the previous section, that is RF, ENET, SVM, and NN models. The five combinations are the RF-SVM-ENET-NN, the RF-SVM-ENET, the RF-ENET, the RF-SVM, and the RF-NN combination.

For forecast combination using lasso regression, we do this by regressing all forecast results from six ML models against actual values. Then, the coefficients obtained from the lasso regression results are used as weights. We use a combination type of lasso regression with consideration to overcome the multicollinearity that occurs between the forecast results of each ML models, and also to select variables from existing ML models.

Table 3. Forecast Combination Results

Combination Models	RMSE	MAD	Pearson Correlation Coefficient
RF-SVM-ENET-NN equal weighting	1.092	0.764	0.911
RF-SVM-ENET equal weighting	1.232	0.903	0.887
RF-ENET equal weighting	1.230	0.900	0.892
RF-SVM Equal weighting	1.218	0.867	0.892
RF-NN equal weighting	1.070	0.775	0.912
Lasso Regression	1.024	0.704	0.919

The results summarized in Table 3 show that the equal weighting combination from RF and NN get the best results than the other equal weighting combinations. This result is also better than the result obtained from individual RF model. RMSE and MAD on the combined RF-NN results decrease by around 16% relative to the RF model. While the correlation value increases about 3% relative to the RF model. The forecast combination result from the lasso regression outperforms the combined forecast results from equal weighting. It can improve the forecasting accuracy of the RMSE and MAD values, and increase the correlation coefficient to 0.919. We can see the comparison of the forecast and actual plot in Figure 6. It appears that the pattern in the forecast result is very close to the actual pattern. Even from the period 2017 to 2019 the pattern almost coincides. So far, the best model produced from this study is a model that combines forecasting result using lasso regression method.

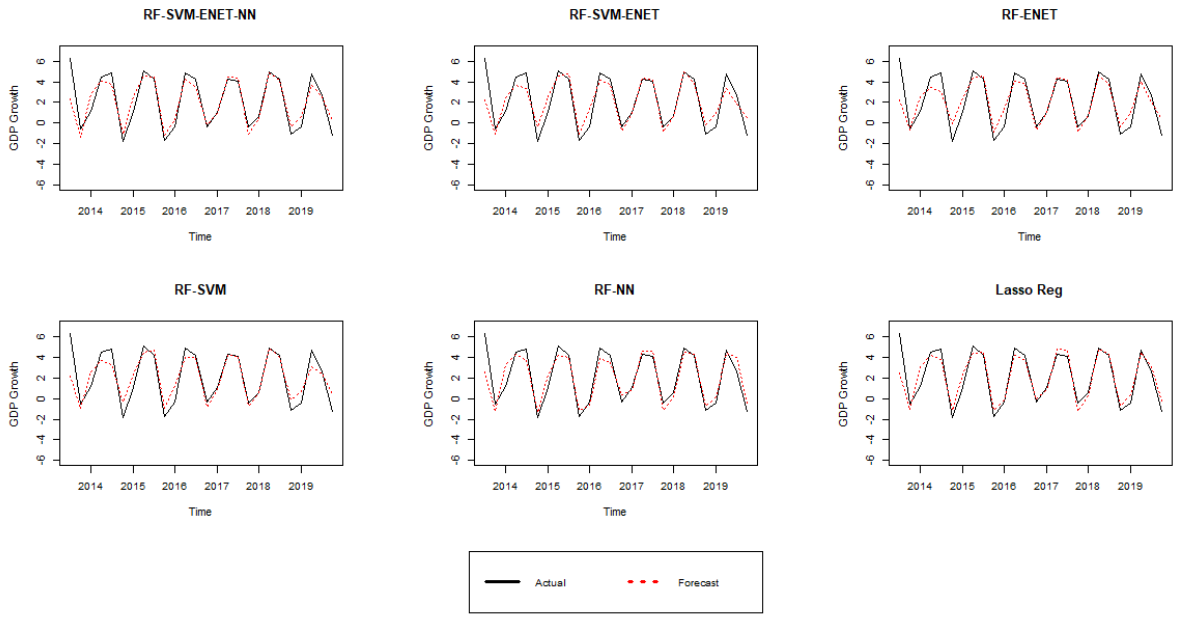

Figure 6. Real-time nowcasts of quarterly GDP Growth for Forecast Combination Methods.

By using Lasso regression as a method for forecast combination, we can make a selection of six individual models used in this study at the same time. Selection in Lasso regression is done by shrinking the regression coefficient to exactly zero on the unselected variable. From the Lasso regression coefficient presented in Table 4, it can be seen that the unselected variable is forecast result from ENET model. From Table 4 we can also see that the RF and NN play important roles in the formation of forecast combination result using Lasso regression. This is indicated by the high values of the regression coefficient, which is 0.400 and 0.390. While the SVM gets a small role in formation of forecast combination, with a regression coefficient of 0.020.

Table 4. Lasso Regression Coefficients.

Variable	Coefficient
Intercept	-0.235
RF	0.400
LASSO	0.082
RIDGE	0.255
ENET	0
SVM	0.020
NN	0.390

4. Conclusion

In this paper, we have evaluated the performance of several ML algorithms in doing real-time forecast on Indonesia's GDP growth data. We trained each algorithm methods to forecast real-time GDP growth from 2013:Q3 to 2019:Q4 using 18 predictor variables in lag 1, 2, 3, 4, and 8. The 18 predictor variables consist of a number of quarterly macroeconomic and financial market statistics in Indonesia. We compared the real-time performance nowcast of each algorithm by looking at the RMSE, MAD, and Pearson correlation coefficient values. We found that all ML models are able to produce more accurate forecasts than AR(1) benchmark. The individual model that has shown the best performance is random forest. Combining forecast results from several individual models can improve forecast accuracy in a better direction. Our results have shown that the real-time forecast of GDP growth using the forecast combination method using the Lasso regression provides better results than the other methods used in this study.

References

Adriansson , N., & Mattsson, I. (2015). *Forecasting GDP Growth, or How Can Random Forests Improve Predictions in Economics? [Bachelor Thesis].* Uppsala, Sweden: Uppsala University.

Breiman, L. (1999). *Random Forests - Random Features.* Technical Report, University of California, Statistics Department , Barkeley.

Chakraborty, C., & Joseph, A. (2017). *Machine Learning at Central Banks.* London: Bbank of England.

Clemen, R. (1989). Combining Forecasts: A Review and Annotated Bibliography. *International Journal of Forecasting 5*:559-583.

Hastie, T., Tibshirani, R., & Friendman, J. (2017). *The Elements of Statistical Learning, Second Edition.* Springer Verlag.

Lee, K.-Y., Kim, K.-H., Kang, J.-J., Choi, S.-J., Im, Y.-S., Young-Dae, L., & Lim, Y.-S. (2017). Comparison and Analysis of Linear Regression & Artificial Neural Network. *Research India Publications*, 9821.

Richardson, A., Mulder, T., & Vehbi, T. (2018). Nowcasting New Zealand GDP using Machine Learning Algorithms. *IFC - Bank Indonesia International Workshop and Seminar on* , (pp. 1-15). Bali, Indonesia.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso. *Journal of the Royal Statistical Society. Series B (Methodological), 58*:267-288.

Vapnik, V. N. (1995). *The Nature of Statistical Learning Theory.* Berlin: Springer-Verlag.

Zou, H., & Hastie, T. (2005). Regularization and Variable Selection Via The Elastic Net. *JR Statist Soc B, 67*:301-320.

Appendix
Data decription

Code	Variable	Freq	Unit
y	Gross Domestic Product Current Price	Q	Percent
x1	Consumption Expenditure	Q	Percent
x2	Private Consumption Expenditure	Q	Percent
x3	Government Expenditure	Q	Percent
x4	Gross Fixed Capital Formation	Q	Percent
x5	Change in Stocks	Q	Percent
x6	Export of Goods and Services	Q	Percent
x7	Import of Goods and Services	Q	Percent
x8	Agriculture	Q	Percent
x9	Industry	Q	Percent
x10	Services	Q	Percent
x11	Current Account	Q	Percent
x12	External Debt	Q	Percent
x13	Foreign Direct Investment	Q	Percent
x14	Consumer tendency index	Q	Percent
x15	BS: PC Utilization: Manufacturing Industry (MI)	Q	Percent
x16	Business Tendency Index	Q	Percent
x17	Domestic Investment	Q	Percent
x18	Foreign Investment	Q	Percent