Inflation forecasting with machine learning methods: A case of Mongolia

A Research Paper

submitted in partial fulfilment of the requirement for

The Degree of Master of Public Policy

Submitted by

Batbold Narmandakh

51-208207

Submitted to

Graduate School of Public Policy

The University of Tokyo

Academic Advisor

Assistant Professor Kucheryavyy Konstantin

June 2022

Tokyo, Japan

TABLE OF CONTENTS

I. INTRODUCTION

Preserving price stability is a main policy objective of national central banks. In practice, due to the existence of operation lag (or effect lag) of monetary actions, the central banks make monetary policy decisions based on the short and medium-term outlook for inflation rather than its observed values. Thus, accurate forecast of inflation plays a key role in taking effective and efficient policy measures by monetary authority.

Over the past decades a number of models, ranging from simple univariate time series models to fully fledged, sophisticated structural macroeconomic models (such as dynamic stochastic general equilibrium models), have been developed and used by central banks for forecasting key macroeconomic variables such as inflation, GDP growth and so on. Moreover, due to the recent advancement in computing technology and availability of big data set, machine learning (ML) methods have drawn attention and been considered as potential alternatives to statistical forecasting models typically used by monetary authorities. The ML methods provide us with opportunity to better handle the main issues (such as nonlinearity, multi-collinearity, predictor relevance and dimensionality) from which traditional statistical forecasting techniques based on ordinary least square method usually suffer. Moreover, ML methods provide chances to find optimal bias-variance trade-off[1] for the forecasting model, leading to more accurate forecasts.

Since 1990, inflation targeting framework has been practiced in many countries. For instance, the central bank of Mongolia (BoM) has adopted an inflation targeting regime in 2007 for better fulfilling its price stability objective[2]. However, due to its forward looking manner, a well-

[1] The concept of bias-variance trade-off lies at the heart of forecasting and the machine learning literatures. If the loss function for forecasting model is quadratic, the expected prediction error to minimize is decomposed into three terms; namely squared bias, variance and irreducible error. Here, variance captures how much the learned model changes if we train it on a different training dataset. Bias represents the difference between the expected value predicted by the model and the correct value. Unfortunately, it is unable to lower both bias and variance at the same time. Generally, the more complex prediction models, the higher variance and the less bias.

[2] The BoM has developed and been using a number of models for constructing short and medium term inflation forecasts, namely Structural Vector Autoregressive (SVAR) model, Bayesian Vector Autoregressive (BVAR) model, Seasonal Autoregressive Integrated Moving Average (SARIMA) model and Factor Augmented Vector Autoregressive (FAVAR).

functioning inflation forecasting framework is strongly required for implementing the regime successfully.

In this analysis, I conduct a horse race analysis using several popular machine learning algorithms (Ridge, Lasso, Elastic net and Extreme gradient boosting algorithm), the factor model and the traditional univariate AR model in forecasting the one to four quarter ahead inflation for Mongolia. The predictions experiments are based on recursive out-of-sample forecasting procedure.

II. LITERATURE SURVEY

This section refers the findings of several studies that investigate the inflation forecasting which used the ML methods and the literatures on the inflation forecasting in Mongolian.

Over the past decade, the application of ML and big data has been growing rapidly in the literatures relevant to forecasting inflation. The Nakamura (2005) is an early attempt to apply neural networks for forecasting U.S inflation. Inoue and Kilian (2008) considered U.S. inflation forecasts from lasso and ridge regression. The recent popular papers, in which ML methods have been used to predict inflation, include Chakraborty and Joseph (2017), Garcia et al. (2017), Medeiros et al. (2019) and Maehashi and Shintani (2020), among others. The results of the studies show that ML methods are able to produce more accurate inflation forecasts than benchmark models. For example, in Medeiros et al. (2019), authors applied some ensemble ML methods to U.S inflation forecasts. Authors found that random forest model dominates all other models. Chakraborty and Joseph (2017) explored areas of application of ML to central banking and policy analyses. They presented three specific case studies and one of them was related to projection exercise for UK's inflation using ML methods - Ridge regression, Nearest Neighbors, Random Forest, Neural Networks, Support Vectors. The results show that ML methods outperform benchmarks in the form of VAR and AR models. In Maehashi and Shintani (2020), authors conducted a horse race analysis using factor models and 9 different ML methods in forecasting the Japanese 7 target macroeconomics variables including inflation. Authors found that ML methods perform particularly well for longer forecast horizons and the joint application of factor models and ML shows better result than factor models or ML alone.

There is also a growing body of literature on inflation forecasting in Mongolia. For instance, Doojav Gan-Ochir (2011) have used SARIMA model and Davaadalai et al. (2011) have used

BVAR model to forecast inflation in Mongolia. In Altan-ulzii and Ganbat (2018), authors have employed a principal component based FAVAR model to forecast short-term inflation in Mongolia using big data. The results have shown that FAVAR model performs better than traditional univariate auto regressive model.

III. EMPIRICAL MODELS

This section provides a brief description of the benchmark and different machine learning models used in the paper for constructing one to four quarter ahead forecasts of inflation in Mongolia. The multi-step ahead forecasting approach considered here is a direct approach in which h period ahead inflation (π_{t+h}) is modeled as a function of predictor variables observed and available at period t.

$$\boldsymbol{\pi_{t+h} = F(X_t) + u_{t+h}}$$

where $F(x_t)$ is a functional mapping of predictors, u_{t+h} is the forecast error, and $X_t = (x_{1t}, \dots, x_{Nt})'$ is a set of predictor variables possibly including lags of dependent variable, exogenous predictors and underlying factors (unobserved latent variables) extracted from a large set of covariates and lags of the factors [3].

3.1 Univariate autoregressive (AR) model

A simple univariate autoregressive AR(p) model is used as a benchmark model. The order p is determined based on the Bayesian information criterion (BIC) and the estimates of the parameters are obtained by OLS method.

$$\boldsymbol{\pi_{t+h} = \alpha_0 + \sum_{i=1}^{p} \alpha_i \ \pi_{t-i+1} + u_{t+h}}$$

where π_{t+h} is h period ahead inflation, $\alpha_0, \dots, \alpha_p$ are parameters and u_{t+h} is the forecast error.

[3] A popular alternative approach is iterated (or recursive) forecasting approach where multi-step ahead forecasts are constructed iteratively based on one-period ahead forecasting model's estimation. Theoretically, the iterated strategy generates more efficient estimates of parameters if the specification of one-period ahead model is correct. However, it is susceptible to bias if the model is misspecified (M. Marcellino et al., 2006). On the other hand, the direct strategy is not prone to model misspecification and a unique approach applicable for all machine learning methods.

3.2 Factor augmented autoregressive (FAAR) model

One of the most widely used statistical methods for macroeconomic forecasting is a factor analysis. In this analysis, information contained in a large set of candidate predictors is summarized by few unobserved latent factors which are then used in forecasting equation as the predictors. The popularity of the factor analysis is attributed to its benefits such as mitigating effectively the curse of dimensionality issues, reducing the risk of omitting important predictor variables (omitted variable bias) in the models and being more robust to the presence of structural breaks in the dataset. For the formal setup, let x_{it} be the value of observed large number of predictors, $i = 1, \dots, N$ and $t = 1, \dots, T$. In a static factor model, each x_{it} can be decomposed as follows:

$$\boldsymbol{x_{it} = \Lambda_i' F_t^k + \epsilon_{it},} \quad i = 1, \dots, N \text{ and } t = 1, \dots, T,$$

where $F_i^k = (f_{1t}, f_{2t}, \dots, f_{kt})'$ *is a* $k \times 1$ *vector of factors,* $\boldsymbol{\Lambda}_i$ *is a* $k \times 1$ *vector of factor loadings (constants) associated with* F_t^k *and* ϵ_{it} *is the idiosyncratic shocks.*

According to the principal component approach proposed by Stock and Watson (2002b), which I follow for the factors extraction in our analysis, the factors and their loading are simultaneously estimated by solving the following minimization problem:

$$\min_{\Lambda, F^k} \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(x_{it} - \Lambda_i' F_t^k \right)^2$$

Determining optimal number of factors (k) is a critical issue. There are a number of approaches for selecting the optimal number of factors in approximate static factor models. One of the most frequently used approaches is the information criterion estimator proposed by Bai and Ng (2002). The authors propose six different types of information criteria yielding consistent estimates of k by minimizing them. Among the criteria the most commonly applied in practice is IC_2 criteria which is defined as follows:

$$\boldsymbol{IC_2(k) = lnV(k) + k\left(\frac{N+T}{NT}\right) ln(min\{N,T\})}$$

where $V(k) = \frac{1}{NT} \sum_{i=1}^{N} \sum_{t=1}^{T} \left(x_{it} - \Lambda_i' F_t^k \right)^2$.

Practically, it is advised to use several information criteria at the same time for determining optimal number of factors because no singe approach outperforms others (Choi and Hanbat Jeong 2017). Another popular approach considered in this paper is eigenvalue ratio estimator proposed by Ahn and Horenstein (2013), which obtains k by maximizing the ratio of two adjacent eigenvalues of $\frac{XX'}{NT}$

$$\boldsymbol{ER(k)} = \frac{\tilde{\boldsymbol{\mu}}_{\boldsymbol{NT,k}}}{\tilde{\boldsymbol{\mu}}_{\boldsymbol{NT,k+1}}},$$

Where $X = (X_1, \dots, X_T), X_t = (x_{1t}, \dots , x_{Nt})'$ *and* $\tilde{\mu}_{NT,k}$ *denotes the* k^{th} *largest eigenvalue of* $\frac{XX'}{NT}$

Then, the forecasting equation including the common factors is determined as following factor augmented autoregressive (FAAR) type:

$$\boldsymbol{\pi}_{\mathbf{t+h}} = \boldsymbol{\alpha_0} + \sum_{i=1}^{p} \boldsymbol{\alpha_i} \ \boldsymbol{\pi}_{\mathbf{t}-\boldsymbol{i}+\mathbf{1}} + \sum_{j=1}^{p} \boldsymbol{\beta_j} \ \mathrm{F}_{\mathbf{t}-\boldsymbol{j}+\mathbf{1}}^{\boldsymbol{k}} + \boldsymbol{u_{t+h}}$$

3.3 Models with regularization (Penalized Regression)

Macroeconomic forecasting, under the situation in which there are a huge number of correlated predictors or number of predictors (N) is much higher than number of observation (T), can be not an easy task. In this setting, the major problem likely to face with is an overfitting (high variance of the model performance) --- in-sample performance of the model is quite accurate, but its out-of-sample forecasts is highly inaccurate. In the linear regression model, putting the constraints on the magnitude of the coefficients (so called regularization) is a one of the possible ways to tackle the issue. The objective function of regularized regression method to minimize is quite similar to that of OLS regression, only difference is the additional penalty term:

$$\sum_{t=1}^{T} \left(\boldsymbol{\pi_{t+h}} - \boldsymbol{\beta_0} - \sum_{i=1}^{N} \boldsymbol{\beta_i x_{it}} \right)^{\mathbf{2}} + \boldsymbol{\lambda P(\beta_1, .., \beta_N)}$$

where $P(\beta_1, .., \beta_N)$ *is penalty term and* λ *is a positive hyper parameter.*

The value of hyper parameter λ defines the magnitude of the penalty term and its value determines bias-variance tradeoff. Generally speaking, the higher value of λ, the greater shrinkage of the regression coefficients (the more regularization) and the more bias and less variance for the forecasting model.

3.3.1 Ridge Regression

Ridge regression was originally introduced by Hoerl and Kennard (1970), in which the penalty term is given by $P(\beta_1, .., \beta_N)$= $\sum_{i=1}^{N} \beta_i^2$. The minimization problem of the ridge regression is then written as follows:

$$\widehat{\boldsymbol{\beta}}^{ridge} = \mathbf{arg}\,\underset{\boldsymbol{\beta}}{\mathbf{min}}\ [\sum_{t=1}^{T} \left(\boldsymbol{\pi}_{t+h} - \boldsymbol{\beta}_0 - \sum_{i=1}^{N} \boldsymbol{\beta}_i x_{it}\right)^2 + \boldsymbol{\lambda} \sum_{i=1}^{N} \boldsymbol{\beta}_i^2\ ]$$

In the Ridge regression, the coefficients of the linear regression model are shrunk close to zero, which helps to prevent overfitting. However, the coefficients does not reach exactly zero for any value of λ.

3.3.2 Least Absolute Shrinkage and Selection Operator (LASSO) Regression

LASSO is alternative regularization method proposed by Tibshirani (1996). In the LASSO regression, the penalty term is defined as $P(\beta_1, .., \beta_N) = \sum_{i=1}^{N} |\beta_i|$ and overall minimization problem is given as follows:

$$\widehat{\boldsymbol{\beta}}^{lasso} = \mathbf{arg}\,\underset{\boldsymbol{\beta}}{\mathbf{min}}\ [\sum_{t=1}^{T} \left(\boldsymbol{\pi}_{t+h} - \boldsymbol{\beta}_0 - \sum_{i=1}^{N} \boldsymbol{\beta}_i x_{it}\right)^2 + \boldsymbol{\lambda} \sum_{i=1}^{N} |\boldsymbol{\beta}_i|\ ]$$

The LASSO regression also shrinks the coefficients to zero like ridge regression. However, it is notable that the LASSO conducts variable selection by forcing some coefficients to exactly zero due to the nature of the penalty term.

3.3.3 Elastic Net Regression

Zou and Hastie (2005) proposed the Elastic net regression method which combines penalty terms of both the Ridge and LASSO. By applying these penalties, Elastic net regression not only effectively shrinks the coefficients toward zero (as in ridge), but also pushes some coefficients to exactly zero (like in LASSO). The minimization problem is defined as follows:

$$\widehat{\boldsymbol{\beta}}^{elastic\,net} = \mathbf{arg}\,\underset{\boldsymbol{\beta}}{\mathbf{min}}\ [\sum_{t=1}^{T} \left(\boldsymbol{\pi}_{t+h} - \boldsymbol{\beta}_0 - \sum_{i=1}^{N} \boldsymbol{\beta}_i x_{it}\right)^2 + \boldsymbol{\lambda} \sum_{i=1}^{N} \{(\mathbf{1} - \boldsymbol{\alpha})\boldsymbol{\beta}_i^2 + \boldsymbol{\alpha}\,|\boldsymbol{\beta}_i|\ \}]$$

where $\alpha \in [0, 1]$ *is the hyper parameter to determine relative weights of the two penalty terms.*

In practice, optimal values of the hyper parameters of regularized regressions (λ for ridge and lasso, λ and α for elastic net) are usually determined by grid search with iterative k-fold cross-validation (CV) technique. Grid search is a popular technique that searches the candidate best hyper parameters exhaustively from the grid of manually specified space of the hyper parameters. Major drawback of the method is that it can be computationally very expensive when there are many hyper parameters and many possible combination of hyper parameters. K-fold CV is a statistical technique to evaluate the performance of predictive models by randomly dividing the original sample into a set of folds; training set for training the predictive model, and a test set for evaluating it. However, K-fold CV is not appropriate method for hyper parameter tuning when time series dataset is used. In this setting, we face the “data leakage” issue due to the random partition procedure of K-fold CV. Thus, considering the inappropriateness of the traditional k-fold CV for time-series forecasting, an alternative technique walk-forward (or rolling origin evaluation) validation method for the hyper parameter optimization is used in this analysis (see Section 4.2). Moreover, I manually specified the search space of the hyper parameters for the regularized regressions as follows:

$\lambda = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$ and
$\alpha = [0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1]$.

The regularized linear regression models (ridge, lasso and elastic net) cannot capture the nonlinear relationships between a target variable and the predictors. Thus, I use Extreme gradient boosting algorithm which is an uptrend machine learning algorithm in time series forecasting nowadays.

3.4 Extreme Gradient Boosting (XGBoost) Algorithm

Extreme gradient boosting (XGBoost), developed by Chen and Guestrin (2016), is a powerful state-of-art machine learning technique based on boosting tree models. XGBoost is an advanced version of gradient boosting decision tree algorithm[4], providing high accuracy, efficiency and

[4] Boosting is an ensemble meta-algorithm which transforms a set of "weak" learners into strong learners. Gradient boosting is a method in which new weak learners (usually shallow decision trees) are sequentially built by using the errors or residuals of previous weak learner and gradient descent algorithm.

scalability. The algorithm constructs a forest of shallow trees (with high bias and low variance) sequentially such that each of subsequent trees reduce the prediction errors[5]. I consider the following derivation of XGBoost algorithm from Chen and Guestrin (2016). A tree ensemble model uses K additive functions to predict the output:

$$\boldsymbol{\hat{y}_i = \sum_{k=1}^{K} f_k(x_i)}, \quad \boldsymbol{f_k \in F}$$

$$F = \{f(x) = \omega_q(x)\}, \ (q: \mathbb{R}^m \to T, \omega \in \mathbb{R}^T)$$

where $\hat{y}_i$ is predicted value, F is the space of regression trees. Each tree f_k is determined by two parameters: tree structure q and leaf weights ω (output values). T is number of leaves in a tree, K is the number of trees. m and n represent the features and the sample size, respectively.

The regularized objective function is defined as follows:

$$L(\boldsymbol{\phi}) = \boldsymbol{\sum_{i=1}^{n} l(y_i, \hat{y}_i)} + \boldsymbol{\sum_{k=1}^{K}} \Omega(\boldsymbol{f_k}) \qquad \textit{(1)}$$

$$\textit{Where } \Omega(\mathrm{f_k}) = \gamma T + \frac{1}{2}\lambda||\omega||^2$$

where $l(y_i, \hat{y}_i)$ is a loss function, $\Omega(\mathrm{f_k})$ is regularization term, λ is hyper parameter controlling the degree of regularization of each f_k and γ is a parameter controlling the extent of complexity penalty for tree structure on T (splitting threshold).

The loss function $l(y_i, \hat{y}_i)$ is a continuous twice-differentiable convex function representing difference between actual and true values and measures the fitness of the model to the training data. $\Omega(\mathrm{f_k})$ regularization term helps to prevent overfitting by controlling the model's complexity. As mentioned in Chen and Guestrin (2016), due to impossibility to optimize loss function $l(y_i, \hat{y}_i)$ for tree-ensemble model, it is needed to train the model in an additive manner. It means that we add the tree f_i which improves the model in equation (1).

$$\boldsymbol{L}^{(\boldsymbol{t})} = \boldsymbol{\sum_{i=1}^{n} l\left(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)\right)} + \Omega(\boldsymbol{f_t}) \qquad \textbf{\textit{(2)}}$$

It means that the algorithm trains each new trees based on gradient of previous tree's loss function (instead of errors of the previous tree).

[5] Despite the new tree is quite shallow, whenever a new tree is added to the ensemble, the bias of the model decreases and the model complexity increases.

where $\hat{y}_i^{(t-1)}$ *is the prediction from the previous iteration.*

After the second-order Taylor expansion of the objective function equation (2) and with some calculations, we can finally get optimized weight ω_j^* for node j for a fixed structure q as:

$$\omega_j^* = -\frac{\sum_{i\in I_j} g_i}{\sum_{i\in I_j} h_i + \lambda}$$

where g_i *and* h_i *are the first and second order gradient statistics on the loss function.*

Moreover, the net gain of the objective function after each split is calculated as:

$$Gain = \frac{1}{2}\left[\frac{\left(\sum_{i\in I_L} g_i\right)^2}{\sum_{i\in I_L} h_i + \lambda} + \frac{\left(\sum_{i\in I_R} g_i\right)^2}{\sum_{i\in I_R} h_i + \lambda} - \frac{\left(\sum_{i\in I} g_i\right)^2}{\sum_{i\in I} h_i + \lambda}\right] - \gamma$$

where I_L *and* I_R *are the set of instances on left and right nodes after the split and* $I = I_L \cup I_R$

As for the tuning the hyper parameters of the XGBoost, grid search with walk-forward (or rolling origin evaluation) cross-validation method is used. The table below shows the manually specified search spaces of the hyper parameters for XGBoost.

Table 1. Manually specified search spaces of the hyper parameters of XGBoost

Parameters	**Description**	**Search space/default values**
nrounds	maximum number of boosting iterations	[100, 300, 500]
max_depth	maximum depth of a tree	[2, 4, 6]
eta	learning rate	[0.1, 0.2, 0.3]
gamma	minimum loss reduction required to make a further partition on a leaf node of the tree.	0
colsample_bytree	subsample ratio of columns when constructing each tree	1
min_child_weight	minimum sum of instance weight (hessian) needed in a child.	1
subsample	subsample ratio of the training instance.	1

IV. DATA DESCRIPTION AND METHODS

4.1 Data description

The dataset used in this analysis consists of 120 quarterly Mongolian macroeconomic variables (including inflation) and covers the period from third quarter of 2007 to fourth quarter of 2021

(N =120, T =58). The target variable inflation is measured by annual change of log of consumer price index in Mongolia.

$$\boldsymbol{\pi_t = \Delta \log(CPI_t) = \log(CPI_t) - \log(CPI_{t-4})}$$

Other 119 time-series are the main indicators in all four macroeconomic sectors of Mongolia (real sector, money and financial sector, external sector and public sector), reflecting the state of the economy as a whole. Non-stationary series are transformed into stationary series by taking differences. The Table 3 in the Appendix shows the full list of 120 series and their relevant stationarity transformations. In addition, all macroeconomic variables are standardized prior to the estimations.

4.2 Forecasting procedure and hyper parameters tuning

The general forecasting procedure used in this analysis is that each of the models are sequentially trained over expanding window process and one-step-ahead out-of-sample forecasts of π_{t+h} are constructed. It means that estimation sample for every one-step-ahead out-of-sample forecasts of π_{t+h} covers all previous observations of the respective forecast (Figure 1). I set the size of the initial training window as 80 percent of the total observations T[6].

Cross-validation is widely used method for tuning hyper parameters of the models. However, standard cross-validation techniques such as k-folds and leave-on-out cross-validation are not appropriate when time series dataset are used. Because, data leakage issue—in which hold out validation set leaks into the training dataset, leading to incorrect estimate of model's performance---arises due to the random partitions[7] of the dataset. As mentioned in previous section, in this analysis I apply a cross-validation technique called walk-forward (or rolling origin evaluation) validation method originally discussed by Tashman (2000). In this method,

[6] It means that the total number of one-step-ahead out-of-sample forecasts is equal to 20 percent of length of total observations T.

[7] Random partition is the basic principle of the standard cross-validation methods. Moreover, the standard cross-validation methods require to have independent and identically distributed ($i.i.d$) data. Having $i.i.d$ data is one of the most important and general assumptions for statistical procedure and machine learning. However, time series data are likely to be highly auto correlated which means the $i.i.d$ assumption does not hold well for them.

there are a series of single observation hold-out validation sets and relative training sets of each of them contains the observations that occurred before them (Figure 2).

Figure 1. General forecasting procedure (outer loop)

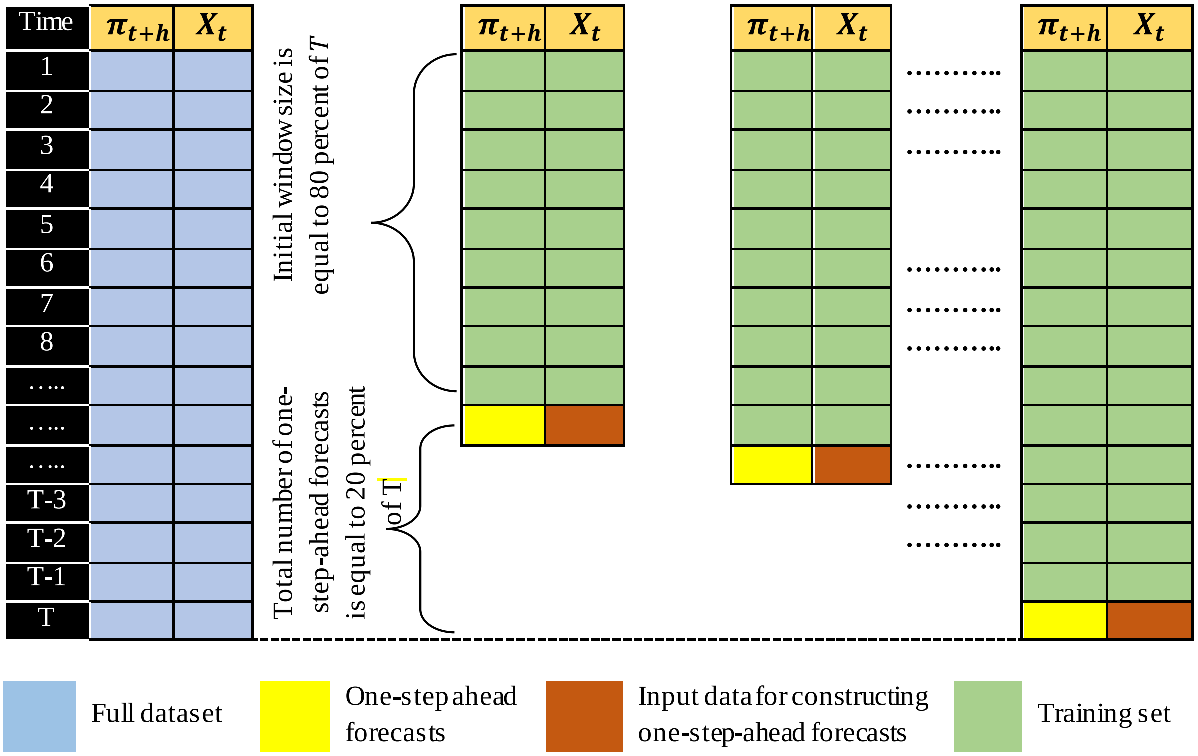
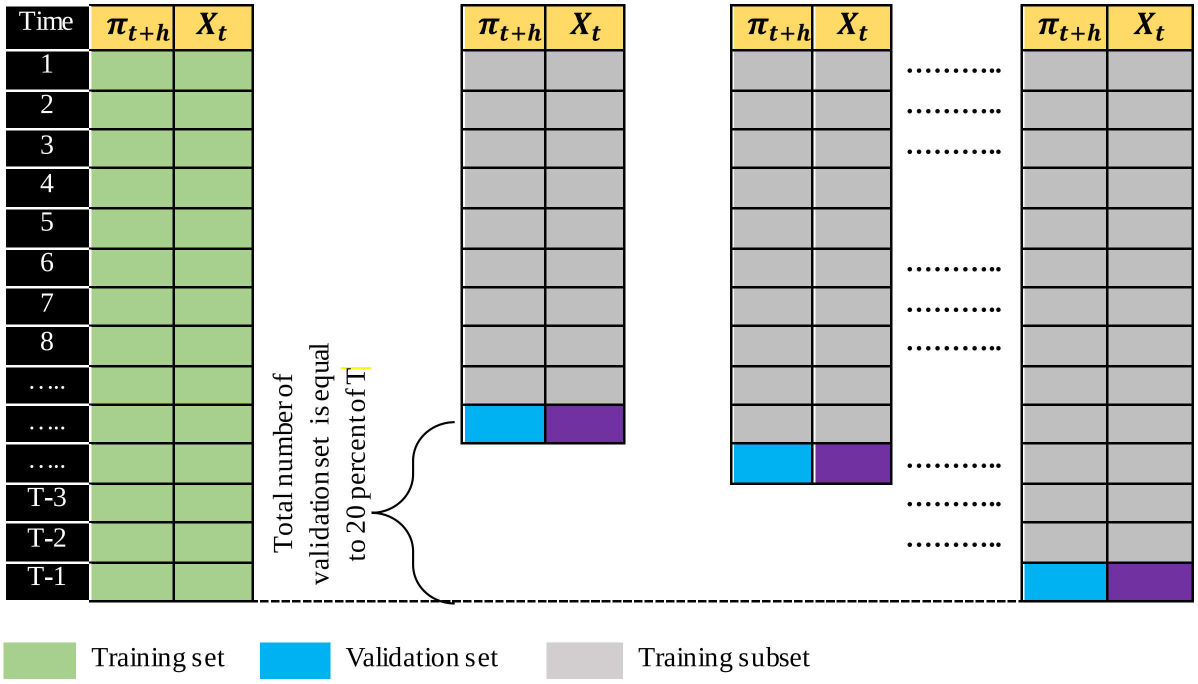

Time
π_{t+h}
X_t
1
2
3
4
5
6
7
8
…..
…..
…..
T-3
T-2
T-1
T
Initial window size is equal to 80 percent of T
Total number of one-step-ahead forecasts is equal to 20 percent of T
…………
Full dataset
One-step ahead forecasts
Input data for constructing one-step-ahead forecasts
Training set

Figure 2. Walk forward cross-validation (inner loop) for hyper parameter tuning



4.3 Forecasting evaluation

As a measurement of the forecasting accuracies of the models, the root mean squared forecast errors (RMSFEs) are calculated as follows:

$$RMSFE = \sqrt{\frac{1}{T-h-T_0+1}\sum_{t=T_0}^{T-h}(\pi_{t+h}-\hat{\pi}_{t+h})^2}$$

where $\hat{\pi}_{t+h}$ is h quarter ahead forecast of inflation constructed by forecasting models, T_o is the sample size used in estimating the model.

4.4 Forecast combination

In this analysis, I consider three types of forecast combination method, namely, the simple average, the trimmed average, and the median forecast combination[8].

- Simple average forecast combination

Suppose $f_t = (f_{1t}, \ldots, f_{Nt})'$ are M number of imperfect collinear predictions for the variable of interest π_{t+h}. The simple average assigns equal weights ($w = \frac{1}{M}$) to all predictions and calculates combined forecast as follows:

$$\hat{\pi}_{t+h} = f_t' w$$

- Trimmed average forecast combination

Again, suppose $f_t = (f_{1t}, \ldots, f_{Nt})'$ are M number of imperfect collinear predictions for the variable of interest π_{t+h} and the ordered forecasts for each point in time,:

$$f_{t_{ordered}} = (f_{\{(1)t\}}, \ldots, f_{\{(N)t\}})'$$

[8] There are several other popular forecast combination methods like Bates/Granger, OLS and Newbold/Granger. However, the methods compute the combination weights using both of actual values and matrix of models' forecasts. It means that in order to compare the performances of different composite forecasts, we need to split our dataset (which includes actual values and matrix of models' forecasts) into training and testing sets. Unfortunately, in our case, the length of the dataset is quite short (only 11 or 12 quarter depending on forecast horizon), meaning that splitting dataset into training and testing set and comparing the performance of the forecast combination methods might be misleading. Thus, in our analysis, I only consider the simple average, the trimmed average, and the median forecast combination methods.

Then, we can compute trimmed average forecast using a trim factor λ as follows[9]:

$$\hat{\pi}_{t+h} = \frac{1}{M * (1 - 2\lambda)} \sum_{i = \lambda M + 1}^{(1-\lambda)M} f_{\{(i)t\}}$$

- Median forecast combination

In median forecast combination, weight 1 is given to median forecast and weight 0 is given to other forecasts for each point in time.

$$\hat{\pi}_{t+h} = median(f_t)$$

V. EMPIRICAL RESULT

This section describes the main results of the recursive out-of-sample prediction experiments applied in this analysis. By the comparison of various models' forecasting performance, several interesting results have been found (Table 2).

Table 2. The ratio of the RMSFEs of the models to the MSFE of the benchmark AR model

Name of the models	Forecast horizon			
	One-quarter ahead	Two-quarter ahead	Three-quarter ahead	Four-quarter ahead
	I	II	III	IV
Part A: The factor model and machine learning algorithms:				
FAAR /ah/	**76.4%**	**79.3%**	**74.9%**	117. 3%
FAAR /bn/	215.6%	207.1%	132.5%	134.3%
Ridge	108.7%	**79.6%**	**76.7%**	**97.5%**
Lasso	116.7%	**81.5%**	**97.2%**	103.2%
Elastic net	122.7%	**85.2%**	**82.3%**	107.7%
XGBoost	126.1%	**67.1%**	**76.1%**	**86.3%**
Part B: Forecast combinations:				
Simple average	**95.9%**	**72.0%**	**78.7%**	**94.5%**
Median	100.1%	**71.2%**	**83.8%**	**96.2%**
Trimmed average	**99.5%**	**72.0%**	**80.2%**	**95.5%**

Note: The bolded numbers denote the relative methods' forecasting performances are better than that of benchmark AR model.

Firstly, even though there are a few exceptions, most of the entries in first and fourth columns of Table 2-Part A are more than 100 percent. It means that our benchmark AR model is quite

[9] In the analysis, I set the value of trim factor $\lambda = 0.2$

competitive with other machine learning and factor models at the one-quarter and four-quarter prediction horizon. For instance, the FAAR model based on eigenvalue estimator of Ahn and Horenstein (2013) is the only model which outperforms the AR model at one-quarter prediction horizon. However, at two and three-quarter prediction horizon, the base AR model is dominated by all other models excluding the FAAR model based on information criterion estimator of Bai and Ng (2002)[10].

Secondly, among the regularized linear regression models, Ridge regression shows the best performance at every prediction horizon. Specifically, at the four-quarter prediction horizon, it is one of the two models (another one is XGBoost) which dominate the benchmark AR model. Ridge regression's dominance over other two regularized regression methods, namely, Lasso and Elastic net, might be caused by the high degree of correlation between the predictor variables.

Thirdly, XGBoost algorithm, which captures the nonlinear interaction between the variables, provides quite satisfactory results in terms of forecasting accuracy. For instance, at the two and four-quarter prediction horizon, XGBoost dominates all other models. More specifically, at the two-quarter prediction horizon the algorithm reduces average forecast error significantly by 32.9 percent relative to the benchmark model, which is the highest gain of prediction accuracy among all models at all prediction horizon.

Fourthly, FAAR models show different performance depending on approach to determine optimal number of factors. FAAR model based on information criterion estimator of Bai and Ng (2002) shows the worst performance at every period of prediction horizon. However, FAAR model based on eigenvalue ratio estimator of Ahn and Horenstein (2013) shows the good performance specially at one and three quarter prediction horizon.

[10] The selected optimal numbers of factors by information criterion estimator of Bai and Ng (2002) are relatively high (seven and eight factors) over the expanding window training period, whereas the determined number of factors by eigenvalue ratio of Ahn and Horenstein (2013) are relatively low only one and two factors. Moreover, FAAR model based on information criterion estimator of Bai and Ng (2002) shows the worst performance in terms of forecasting accuracy at every period of prediction horizon.

Finally, as seen from the Table 2-Part B, almost all composite forecasts outperform the benchmark AR model forecasts at every prediction horizon. Among them simple average forecast combination perform slightly better than other two composite forecasts, namely trimmed average and median forecast combination.

VI. CONCLUSION

In this paper, I conduct a horse race analysis using several popular machine learning algorithms, the factor model and the traditional univariate AR model in forecasting the one to four quarter ahead inflation for Mongolia. The results of this study show that all machine learning methods are likely to dominate the benchmark AR model in terms of the forecasting accuracy in medium term (at two and three quarter prediction horizon). However, not all methods work equally well – XGBoost, FAAR-ah and Ridge show the best performance. Moreover, the composite forecasts provide quite satisfactory results in terms of forecasting accuracy. Therefore, it can be concluded that the machine learning methods and the forecast combination techniques can be the potential alternative forecasting tools for the BoM to make short and medium term inflation forecasts in Mongolia.

VII. REFERENCES

[1] Baybuza, I., (2018). "*Inflation Forecasting Using Machine Learning Methods". Russian Journal of Money and Finance, 77(4), pp. 42–59.

[2] Chakraborty, C. and Joseph, A., (2017). *"Machine Learning at Central Banks".* Bank of England Working Papers, N 674.

[3] Choi, I., & Jeong, H. (2019). *"Model selection for factor analysis: Some new criteria and performance comparisons".* Econometric Reviews, 38, 577 - 596.

[4] Chen, T, Guestrin, C, (2016). *"XGBoost: a scalable tree boosting system. In: Proceedings of the 22nd ACM SIGKDD".* International Conference on Knowledge Discovery and Data Mining (KDD '16), New York, NY, USA. ACM, pp. 785–794.

[5] Chuluun.A, and Atarsaikhan.G, (2018). *"Theoretical background and application of FAVAR model",* The Bank of Mongolia Working Paper series №5, 2018.

[6] Hastie, T., Tibshirani, R., & Friedman, J., (2009*). "The Elements of Statistical Learning: Data Mining, Inference, and Prediction".* Springer Science & Business Media.

[7] Hoerl, A.E., Kennard, R.W., (1970*). "Ridge regression: biased estimation for nonorthogonal Problems".* Technometrics 12 (1), 55–67.

[8] Hyndman, R.J., Athanasopoulos, G, (2018). *"Forecasting: Principles and Practice",* 2nd edition OTexts, Melbourne, Australia.

[9] J. Bai and S. Ng (2002), *"Determining the number of factors in approximate factor models",* Econometrica, vol. 70, pp. 191–221.

[10] Jeffrey C. Chen and Abe Dunn and Kyle Hood and Alexander Driessen and Andrea Batch, (2019). *"Off to the Races: A Comparison of Machine Learning and Alternative Data for Predicting Economic Indicators",* University of Chicago Press, Big Data for 21st Century Economic Statistics

[11] Jin-Kyu Jung, Manasa Patnam, and Anna Ter-Martirosyan, (2018). *"An Algorithmic Crystal Ball: Forecasts-based on Machine Learning",* IMF Working Paper No. 18/230 (Washington: International Monetary Fund).

[12] Kohei Maehashi, Mototsugu Shintani, (2020). *"Macroeconomic forecasting using factor models and machine learning: an application to Japan",* Journal of the Japanese and International Economies, Volume 58, 2020,

[13] Marijn A. Bolhuis and Brett Rayner, 2020. *"Deus ex Machina? A Framework for Macro Forecasting with Machine Learning"*, IMF Working Paper No. 20/45 (Washington: International Monetary Fund).

[14] Massimiliano Marcellino, James H. Stock and Mark W. Watson, 2006, *"A comparison of direct and iterated multistep AR methods for forecasting macroeconomic time series",* Journal of Econometrics 135 (2006) 499–526.

[15] Medeiros, Marcelo C., Vasconcelos, Gabriel F. R., Veiga, Álvaro, Zilberman, Eduardo, 2021. *"Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods",* Journal of Business & Economic Statistics.

[16] S. C. Ahn and A. R. Horenstein (2013) *"Eigenvalue ratio test for the number of factors".* *Econometrica*, vol. 81, pp. 1203–1227.

[17] Stock,J.H., Watson, M.W., 2002a. *"Forecasting using principal components from a large number of predictors",* J. Am. Stat. Assoc. 97 (460), 1167-1179

[18] Stock,J.H., Watson, M.W., 2002b. *"Macroeconomic forecasting using diffusion indexes"*, J. Bus. Econ. Stat. 20 (2), 147-162.

[19] Tashman, L.,J., 2000, *"Out-of-sample tests of forecasting accuracy: an analysis and review",* Int. J. forecast. 16 (4), 437-450

[20] Tibshirani, R., 1996. *"Regression shrinkage and selection via lasso"*. Journal of the Royal Statistical Society: Series B 58 (1), 267–288.

[21] Tiffin, Andrew J., 2016, *"Seeing in the Dark; A Machine-Learning Approach to Now casting in Lebanon",* IMF Working Paper No. 16/56 (Washington: International Monetary Fund).

[22] 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.

VIII. APPENDIX

Table 3. List of variables and transformation

№	Name of variable	Transformation[11]
Monetary aggregates		
1	M2	5
2	M1	5
3	Loan outstanding of banks	5
4	Policy rate	1
5	CBB rate	1
6	Interbank market rate	1
7	Lending rate	1
8	Exchange rate of USD, average	5
9	Exchange rate of USD, end of period	5
10	NEER	5
11	REER	5
Labor market		
12	Employment	5
13	Labor force	5
14	Unemployment Rate	5
15	Nominal Wage	5
16	Real wage	5
Balance of payment		
17	FDI	2
18	FDI inflow	2
19	Portfolio flow	2
20	Total export	5
21	Export /goods/	5
22	Export / services/	5
23	Total import	5
24	Import /goods/	5
25	Import /services/	5
Output (Production method)		
26	GDP (Production method)	5
27	Real GDP: Agriculture	5
28	Real GDP: Mining	5
29	Real GDP: Manufacturing	5
30	Real GDP: Electricity	5
31	Real GDP: Construction	5
32	Real GDP: Trade	5
33	Real GDP: Transportation	5
34	Real GDP: Communication	5
35	Real GDP: Services	5
36	Real GDP: Net tax on products	5
37	Agriculture: Livestock	5
Output (Expenditure method)		
38	GDP (Expenditure method)	5
39	Final consumption	5
40	Household consumption	5
41	Government consumption	5
42	Gross capital formation	5
43	Gross fixed capital formation	5
44	Changes in inventories	2
45	Net exports	2
46	Export	5
47	Import	5
Deflator (expenditure method)		
48	Deflator consumption	5
49	Deflator government spending	5

[11] Transformation (1 – no transformation; 2 – year on year change; 4 – logarithm; 5 – year on year change of logarithm)

50	Deflator gross capital formation	5
51	Deflator export	5
52	Deflator import	5
Budget		
53	Revenue	5
54	Tax revenue	5
55	Non-tax revenue	2
56	Expenditure	2
57	Budget balance	2
58	Current Expenditure	5
59	Interest payment	2
60	Net loan	2
61	Investment Expenditure	2
External sector		
62	Copper price	5
63	Gold price	5
64	Iron ore price	5
65	Brent oil price	5
66	US GDP growth	2
67	Russia GDP growth	2
68	China GDP growth	2
69	Coal price /Thermal/	5
70	Crude oil price /Ural/	5
Real Estate		
71	Top-20 index	5
72	Market Capitalization	5
73	Value of Transaction	5
74	Rent for apartment, 1 room	5
Price		
75	Consumer Price Index	5
76	Food Consumer Price Index	5
77	Non-Food Consumer Price Index	5
78	Core Consumer Price Index	5

79	Meat Consumer Price Index	5
80	Non-meat Food Consumer Price Index	5
81	Fuel Consumer Price Index	5
82	Administrated Consumer Price Index	5
83	Others CPI	5
84	Imported Goods CPI from Others	5
85	Domestic Goods + Services CPI from Others	5
86	Goods CPI from Others	5
87	Services CPI from Others	5
88	Beef CPI	5
89	Flour CPI	5
90	Milk CPI	5
91	Mutton CPI	5
92	Vegetables CPI	5
93	Domestic goods CPI from Others	5
94	Other Foods CPI	5
95	Other Meats CPI	5
External Trade		
96	Consumer goods imports, CIF	5
97	Non-durable goods import, CIF	5
98	Durable imports, CIF	5
99	Cars imports, CIF	5
100	Industrial imports, CIF	5
101	Capital goods imports, CIF	5
102	Construction goods imports, CIF	5
103	Machinery goods imports, CIF	5

104	Other capital goods imports, CIF	5
105	Fuels imports, CIF	5
106	Diesels imports, CIF	5
107	Other fuels imports, CIF	5
108	Other imports, CIF	5
109	Mining goods export	5
110	Copper exports	5
111	Coal exports	5
112	Iron ore exports	5
113	Cash goods exports	5
114	Export volume of copper	5
115	Export volume of coal	5
116	Export volume of iron ore	5
Uncategorized		
117	New loan issued by banks	5
118	Business loan outstanding of banks	5
119	New business loan issued by banks	5
120	Household income	5