# Centraal Planbureau (CPB): Cross Validation

Emma Arussi, Marta Chejduk, Kevin Mai, Tess Scholtus

January 2024

**Abstract**

This study focuses on macroeconomic forecasting in the Netherlands using large datasets, specifically exploring the comparison between Cross-Validation methods for hyperparameter tuning on traditional autoregressive (AR) models and machine learning (ML) models. Machine learning, with its flexibility and ability to handle large datasets, is gaining popularity in macroeconomic forecasting. This study utilizes a Dutch macroeconomic dataset (NL-data) and compares traditional AR models with Elastic Net, a penalizing regularization machine learning algorithm. The study evaluates the influence of hyperparameter tuning through cross-validation (CV), specifically comparing Pseudo Out-of-Sample (POOS) and K-fold CV methods. The study replicates a 2019 study, highlighting the influence of different training window techniques on multiple horizon forecasting performances[9] . The methodology involves selecting optimal model hyper parameters through AIC, K-fold, and POOS CV, followed by evaluation of their forecasting performance. The results suggest that the AR models and ML models select varied hyperparameters and exhibit varied performance under different CV methods. Under what conditions either POOS or K-fold are prone to over- or underfit macroeconomic data is further discussed in the discussion section.

# 1 Introduction

### 1.0.1 Macroeconomic forecasting with large data panels

Macroeconomics focuses on the behaviour of the entire economy, studying variables such as gross domestic product (GDP), inflation, unemployment, interest rates, and other aggregate economic indicators. Macroeconomic forecasting studies the potential outcomes of these variables over time, thereby allowing policymakers to make informed decisions, adapt strategies, and address emerging challenges in a dynamic economic landscape. This ongoing assessment is essential to ensure that macroeconomic policies remain relevant and responsive to the evolving needs of the economy [26, 2].

Forecasting is especially important for policy making due to lags in the response of economic activity to variations in macroeconomic policy instruments. Large datasets, such as the well-known FRED-MD monthly panel, are becoming increasingly available for model analysis and forecasting [17]. Large shared economic databases like FRED benefit researchers in econometrics by providing centralized and accessible economic data, streamlining the research process, and ensuring consistency. Following the FRED-MD, similar datasets have been introduced for Canada and the United Kingdom. In the Netherlands, the Central Bureau for Economic Policy Analysis (CPB) provides policy-relevant economic analyses and projections [1]. CPB has developed the Dutch version of these data sets, comprising variables reflecting the Dutch macroeconomy. Given the multifaceted nature of the macroeconomy, the NL-MD encompasses a substantial number of variables. The composition of the NL-MD is derived from three sources: (1) CBS (StatLine), (2) ECB (Statistical Data Warehouse), and (3) Yahoo Finance *(Personal correspondence with CBP, 2024).*

Macroeconomic forecasting depends on time series data, which behaves differently from most data sets used in Machine Learning (ML) or Classical statistical analysis. Time series observations are interdependent, as the value at any given time point is influenced by previous observations. Time series data often displays distinct characteristics, including aspects like stationarity (or its absence), periodic patterns, and seasonality, reflecting the temporal structure and trends inherent in the data [5].

### 1.0.2 Machine Learning vs Classical Models

Statistical methods often applied to univariate or multivariate linear time series models are Autoregressive Integrated Moving Average (ARIMA) for univariate or Vector Autoregression (VAR) for multivariate time series models. ARIMA models are often used as a benchmark for comparison to more complex dynamic models or machine learning models, due to their simple marginal univariate representation characteristic [7]. Multivariate econometric models are useful for describing more complex patterns in macroeconomic data. However, the performance of standard econometric models tends to deteriorate as the dimensionality of the data increases, which is the well-known curse of dimensionality [3].

Machine learning is often applied to large data sets, and it offers many tools that can be used for analytical purposes such as prediction or forecasting [13]. It is characterized by great flexibility and can adapt to complex forms, which can contribute to the accuracy of out-of-sample predictions and forecasts [16]. Machine learning algorithms can automatically identify and select the most relevant features from a high-dimensional dataset. For example, penalizing models like Ridge, Lasso, and the hybrid Elastic Net are regularization techniques commonly used in machine learning to prevent overfitting, therefore counteracting the curse of dimensionality [18].

The increasing availability of macro-economic data through developments such as FRED and the NLdatabase has contributed to a rise in the adoption of machine learning practices in the field. In the paper of Goulet Coulombe, Leroux, Stevanovic, & Surprenant of 2019, 4 typical applications of ML practices have been tested on the FRED database: nonlinearities, regularization, cross-validation and alternative loss function. Next to the effect of the ML applications, their study also showed that target variables, forecasting horizons, and the state of the economy were of significant influence on results. In their study, nonlinearity turned out to be most relevant to forecasting accuracy. The rest of the features appeared less relevant in the following decreasing order of importance: in-sample loss function, hyperparameters' optimization through cross-validation and alternative shrinkage methods. Nevertheless, cross-validation as a method of hyperparameter tuning is a conventional and often used method and results in different outcomes when applied to machine learning or classical models. It can be applied to any model, including statistical as machine learning models, such as the regularization model Enet that is evaluated in this study.

### 1.0.3 Hyperparameter tuning

For almost all models, including econometric/statistical and machine learning, (hyper)parameter fine-tuning and therefore model selection can be done with in-sample criterion and out-of-sample model selection. Since the 1970s the Akaike Information Criterion (AIC) and Bayesian Information Criterion (BIC) have been used as in-sample model selection criteria in time series analysis. Both penalize models for complexity, with AIC being less stringent than BIC, and they help strike a balance between goodness of fit and model simplicity when choosing the order of autoregressive models [22]. Simple out-of-sample model selection involves splitting data into a training and test set, upon which the data is calibrated on the training set and the test set is preserved to evaluate performance. More recently, cross-validation (CV), which is originally a machine learning concept has been applied to ARIMA models too for out-of-sample testing for parameter selection Khim & Liew [14]. CV involves multiple rounds of splitting, providing a more robust analysis. However, cross-validation is deemed more computationally demanding, and when applied to time series data, the theoretical assumption of CV, which assumes independent and identically distributed values, is not always satisfied.

### 1.0.4 POOS vs. K-fold Cross Validation

In the paper of Goulet Coulombe, Leroux, Stevanovic, & Surprenant of 2019, 2 methods for performing cross validation are described. The first, Pseudo-out-of-sample forecasting (POOS CV) with rolling-origin re-calibration takes into account the order of the original time series. This assures that new data is not used to predict older values. The POOS CV can be adjusted by adapting the rolling window size, or opting fixed window. The other CV method described is K-fold. In K-fold cross-validation, the dataset is divided into k equally sized folds, and the model is trained and evaluated k times. Where k-1 subsets are used for training and the remaining fold is used as a test subset. The K-fold CV can be adjusted by adapting the number of folds, and the advantage of k-fold is that the entire data-set can be used for evaluation. Typically 5-20 folds are used.

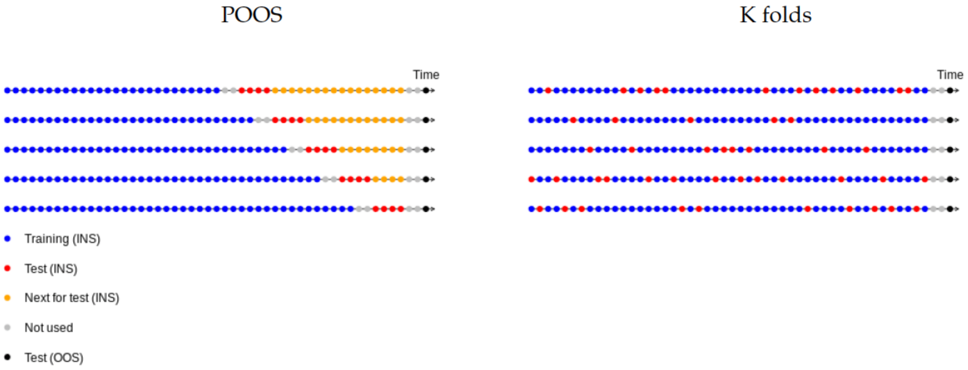

Figure 1: Comparison of Pseudo Out-of-Sample (POOS) and K-fold Cross Validation: Insights from the 2019 Study by Goulet Coulombe, Leroux, Stevanovic, & Surprenant [9].

In this research, the goal was to replicate the 2019 study conducted by Goulet Coulombe, Leroux, Stevanovic, & Surprenant, utilizing the Dutch NLdata set provided by CBP, whilst focusing on CV only. Beyond reproducing the original study, the aim was to highlight the influence of employing different training-window techniques for Pseudo-Out-of-Sample (POOS) CV and varying fold sizes for K-fold CV on forecasting performances. A comparison between an AR(p) benchmark and the elastic net model with regularization penalties, which is an ML strategy/model, was made. Finally, the study compared the effects of both POOS and K-fold CV on longer forecasting horizons for both the benchmark AR(p) model and the Elastic Net model. With these results the paper aims to further discuss the optimal use of CV methods for time series data.

# 2 Methodology

## 2.1 Data preparation

All macroeconomic data of the Netherlands as well as additional R libraries needed for analysis were provided by CPB. To facilitate comparison between the benchmark and the ML models the same cleaning was executed on the time series data. From the NLdata base, a period of roughly 10 years between 2010M3-2020M12, counting 130 observations was selected in which the trade-off between completeness of the time series, versus the amount of variables available was optimized. Irrelevant for the univariate benchmark model but of great importance to the machine learning model was the h-period lagging of all variables except for the variable of interest, depending on the h-step ahead forecast. Out of 160 variables, 96 variables were selected due to their completeness of values in the set time period. Necessary transformations for the purpose of stationarity were applied to all variables through a built-in function of the NLMD library provided by CPB, only for the 1-step ahead forecast. For longer horizons, the 3 and 12- step ahead forecast transformations of the NLMD dataset had to be applied for the machine learning model. The dependent variable had to be separated from the data set and manually transformed to a stationarity variable with the use of the Dicky-Fuller test. The independent variables from the transformed NLMD set were lagged for 3 or 12 periods and afterwards the transformed dependent variable was re-added to those lagged variables. The variable of interest selected for this study was unemployment rate, which is a well-interpretable single scalar metric.

## 2.2 Models

The benchmark model applied in this study is the simple AutoRegressive direct (AR) model of order (p). It is a univariate, linear model. The only hyperparameter in this model is $p_y$, the order of the lag polynomial $\phi(L)$.
It is given as:

$$\hat{y}(h)_t = c + \phi(L)y_t + \varepsilon_{t+h}, \tag{1}$$

for $t = 1, \ldots, T$, where $h$ is the forecasting horizon.
The coefficients of the AR(p) model are determined with a most likelihood function. The likelihood function for the AR(p) model is the probability density function of the observed data given the parameters. Assuming the errors ($\varepsilon_t$) are normally distributed with mean zero and constant variance ($\sigma^2$), the likelihood function $L(\phi_1, \phi_2, \ldots, \phi_p, \sigma^2)$ is given by:

$$L(\phi_1, \phi_2, \ldots, \phi_p, \sigma^2) = \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{1}{2\sigma^2}\sum_{t=1}^{n}(y_t - c - \phi_1 y_{t-1} - \phi_2 y_{t-2} - \ldots - \phi_p y_{t-p})^2\right)$$

Here, $n$ is the number of observations, $y_t$ is the observed value at time $t$, $c$ is a constant, $\phi_1, \phi_2, \ldots, \phi_p$ are the autoregressive coefficients, and $\sigma^2$ is the constant variance of the errors.

The ML model used in this study is Elastic Net. Elastic Net combines the penalties of ridge and lasso, and is a multivariate linear model.
The loss function for elastic net is given by:

$$L = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \mathbf{x}i'\boldsymbol{\beta}\right)^2 + \lambda\left[\alpha \sum j = 1^k|\beta_j| + \frac{1-\alpha}{2}\sum_{j=1}^{k}\beta_j^2\right], \tag{2}$$

where $n$ is the number of observations, $\mathbf{x}_i$ is the vector of predictors for observation $i$, $\boldsymbol{\beta}$ is the vector of coefficients, $k$ is the number of predictors, $\lambda_1$ is the regularization parameter for ridge regression, $\lambda$ is the overall regularization parameter, and $\alpha$ controls the mixing between the ridge ($\alpha = 0$) and lasso ($\alpha = 1$) penalties.

Elastic net utilizes a loss function, it automatically penalizes the difference between predicted values and actual values. The type of loss function used in this study is Mean Squared Error (MSE), which measures the average squared difference between the predicted and actual values:

$$\text{MSE} = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \tag{3}$$

where $n$ is the number of observations, $y_i$ is the actual value for observation $i$, and $\hat{y}_i$ is the predicted value for observation $i$.

## 2.3 Hyper parameter selection through CV

Initially, the data frame of 2010M3-2020M12 was split 80% over 20%, where the first 80% of the data was used as training data for the model. On this training data parameters of the model were selected using either K-fold CV or Rolling/Fixed POOS CV. The remaining 20% test data was reserved to perform the forecasts. By using this format, the forecasting was done on known and existing data, therefore referred to as an additional Pseudo-out-of-sample test for the forecasting performance of the selected models by the hyperparameter tuning for both K-fold and POOS. The forecasting horizons of 1,3 and 12 months were analysed.

As a benchmark for in-sample fit evaluation, AIC was applied to the benchmark AR(P) model.

K-fold CV was applied to both models on the 80% train data with 5, 10 and 20 folds. The K-fold model with the best in-sample fit was then chosen to validate the model on the test data.

POOS-CV was applied as well for AR and for Enet the 80% train data. Multiple initial window sizes and the fixed or rolling window were tested. An rolling window indicates that for each prediction, the train set increases by 1 observation. While for rolling-window for each prediction the window shifts one position, whilst keeping the same size. To capture the potential annual seasonality of time series data we choose to test an initial window for a minimum of 12 months and a maximum of 84 months.

The resulting values obtained from this were on 80% data calibrated AR(P) with optimal parameter P, and on 80% data calibrated Enet models with optimal parameters Alpha and Lambda. Also, the RMSE of the optimal model is obtained, referred to as in-sample fit RMSE.

## 2.4 Forecasting with POOS rolling window approach for all models

The selected models from K-fold and POOS CV that were retrained on the train-data set where tested on the remaining 20% test data. On this test data, the MSE of the ¨out of sample¨ forecasted values compared to the actual data (test data) were obtained.

For this again a rolling window approach was used; each time an h-step ahead prediction was made on a new data point in the test set, this prediction was added to the rolling window and used for predicting the next data point.

## 2.5 Evaluation of performance

For this studies forecasting performances the standard Root-Mean-Square-Error (RMSE) has been used to evaluate model performance. For consistency, RMSE was the evaluator for in-sample fit, inbedded in the CV functions (POOS and Kfold). Also, for AIC, which is a different metric for in-sample validation, additional RMSE for in-sample fit of the final selected model was calculated. Lastly, also the rolling forecasts for all forecasting horizons where compared with RMSE.

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2} \tag{4}$$

The significance of different forecasting RMSEs between ML and AR models have been evaluated with The Diebold-Mariano test.

$$DM = \frac{\bar{d}}{\sqrt{\hat{\text{Var}}(\bar{d})}} \tag{5}$$

where:

- $\bar{d}$ is the mean of the loss differential series $d_t = L(y_t, \hat{y}_{1,t}) - L(y_t, \hat{y}_{2,t})$,
- $L(y_t, \hat{y}_{i,t})$ is the loss associated with forecast $\hat{y}_{i,t}$ from model $i$ at time $t$,
- $\hat{\text{Var}}(\bar{d})$ is an estimate of the variance of $\bar{d}$.

# 3 Results

## 3.1 In-sample fit - AIC vs. POOS vs. K-fold Cross Validation AR(P)

| K-Fold | Best AR Order | Best CV RMSE |
|---|---|---|
| 5 | 1 | 0.0171 |
| 10 | 1 | 0.0172 |
| 20 | 1 | 0.0171 |

Table 1: Summary of k-fold cross-validation results for the AR model.

Table 1 presents results of the in-sample model fit for K-fold CV trained data for AR(P). Changing the number of folds did not show influence on model selection.

| AR Order (p) | Window Size | Fixed Window | CV RMSE |
|---|---|---|---|
| 5 | 12 | TRUE | 0.0179 |
| 1 | 36 | TRUE | 0.0213 |
| Non-stationary | 12 | FALSE | NA |
| 2 | 36 | FALSE | 0.0138 |

Table 2: Summary of POOS cross-validation results for the AR model

Table 2 presents the in-sample model fit for POOS CV trained data for AR(P). Here we see that different initial window sizes do have an effect on RMSE and model selection, where for window size 36 with a rolling window, the model indicates non-stationarity. Rolling window with initial window size 36 achieves the best model fit (0.0138), and opts for AR(2).

| AR (AIC) | AR-kfold-20 | AR-POOS-fixed |
|---|---|---|
| 0.0145 | 1.1793 | 0.9517 |

Table 3: Comparison best POOS-CV and K-fold CV for forecasting

Table 3 presents the relative in-sample RMSE for AR(p). The reference category is RMSE associated with AIC which is 0.0145. AR-kfold-20 and AR-POOS-fixed represent proportional comparison to the baseline AIC RMSE. AR-POOS with a fixed window performs best in-sample, compared to AIC and K-fold CV.

### 3.2 In-sample fit - POOS vs. K-fold Cross Validation in Machine Learning Models

| K-Fold | Alpha | Lambda | Best CV RMSE |
|---|---|---|---|
| 5 | 0 | 0.001 | 0.0140 |
| 10 | 0.1 | 0.008 | 0.0131 |
| 20 | 0 | 0.024 | 0.0134 |

Table 4: Summary of k-fold cross-validation results for the Elastic Net model

Table 4 represents the in-sample fit for the Elastic net model with different numbers of folds. The best performing model is the K-fold 10 with a CV RMSE of 0.0131. K-fold 20 performs second best with a CV RMSE of 0.0134 and k-fold 5 performs the worst with a CV RMSE of 0.0140. See 4.2.1 K-fold CV Enet for further analysis on k-fold.

| POOS-CV | Fixed Window | alpha | Lambda | CV RMSE |
|---|---|---|---|---|
| 12 | FALSE | 0.8 | 0.00379 | 0.0126 |
| 36 | FALSE | 1 | 0.0016 | 0.0105 |
| 60 | FALSE | 0.9 | 0.00379 | 0.0108 |
| 84 | FALSE | 1 | 0.0037 | 0.0110 |

Table 5: Rolling window model selection POOS

| POOS-CV | Fixed Window | alpha | Lambda | CV RMSE |
|---|---|---|---|---|
| 12 | TRUE | 0.2 | 0.0089 | 0.0178 |
| 36 | TRUE | 0.3 | 0.00876 | 0.0192 |
| 60 | TRUE | 0.8 | 0.00379 | 0.0197 |
| 84 | TRUE | 1 | 0.0037 | 0.0191 |

Table 6: Fixed window model selection POOS

Table 5 and 6 represent the model selection with the retrained POOS cross validation method for different initial window sizes and for the rolling window approach (Fixed Window = FALSE) or the fixed window approach (Fixed window = TRUE). The model with the lowest fit for the fixed window approach is the POOS model with a initial window of 12. The selected initial window size of the POOS model with the rolling window was 36. The rolling window approach performed better with in-sample fitting CV RMSE (0.0105) compared to the fixed window approach CV RMSE (0.0178).

| Best model | IW/FOLD | Alpha | Lambda | IS RMSE | H=1 | H=3 | H=12 |
|---|---|---|---|---|---|---|---|
| KFold | 10 | 0.1 | 0.008 | 0.01376 | 0.0388 | 0.0437 | 0.0577 |
| Rolling POOS | 36 | 1 | 0.002 | 0.0105 | 0.0269 | 0.0362 | 0.0579 |
| Fixed POOS | 12 | 0.2 | 0.008 | 0.0118 | 0.0267 | 0.0437 | 0.0576 |

Table 7: Comparison best POOS-CV and K-fold CV for forecasting

Table 7 presents the RMSEs of the best selected machine learning models for Enet and presents the forecasting RMSEs for H= 1, 3 and 12. We observe that the pseudo-out-of-sample with the fixed window size performs slightly better than K-fold CV for horizon 1 and 3 but Kfold performed tiny little better for horizon 12. Pseudo-out-of-sample with fixed window size performs better than Kfold for horizon 1 and 12 but is the same for horizon 3. So in general pseudo-out-of-sample performs slightly better than K-fold when it comes to the elastic net regression model and when comparing the forecasting RMSEs for horizons 1, 3 and 12.

| Model AR | Order (P) | Forecast RMSE (h=1) | Forecast RMSE (h=3) | Forecast RMSE (h=12) |
|---|---|---|---|---|
| AIC | AR(3) | 0,0427 | 0,0470 | 0,0504 |
| Kfold20 | AR(1) | 0,9523 | 0,9164 | 0,9530 |
| POOS ROLL36 | AR(2) | 1,0000 | 1,0000 | 0,9927 |
| POOS FIXED12 | AR(5) | 1,0291 | 1,0051 | 0,9910 |
| **Model Enet** | **Alpha-Lambda** | **Forecast RMSE (h=1)** | **Forecast RMSE (h=3)** | **Forecast RMSE (h=12)** |
| Kfold10 | 0.1-0.008 | 0,8521 | 1,0593 | 1,2077 |
| POOS-ROLL36 | 1-0.00164 | 0,8522 | 1,0489 | 1,2195 |
| POOS-FIXED12 | 0.0088-0.0118 | 1,0393 | 0,9779 | 1,1966 |

Table 8: Forecast Performance AR vs Enet for H=1,3 and 12. RMSE AIC used as benchmark and the numbers below represent the relative, with respect to AR-AIC

Table 8 presents the RMSEs of all forecasts for optimal models selected for different CV techniques, with AR-AIC used as a benchmark. The Enet model performs slightly better than the AR model when it comes to horizon = 1, however when the horizons are longer, the AR performs better compared to the eEnet model. To compare those differences further significance testing were performed.

### 3.2.1 Across model comparison

| Comparison | Horizon | Db Test Statistic | 5% significance |
|---|---|---|---|
| kfold AR vs kfold Enet | 1 | -0.06010496 | not significant |
| kfold AR vs kfold Enet | 3 | 0.09337803 | not significant |
| kfold AR vs kfold Enet | 12 | 0.1768671 | not significant |
| Poos Rolling AR vs Rolling Enet | 1 | -0.08768408 | not significant |
| Poos Rolling AR vs Rolling Enet | 3 | 0.03204858 | not significant |
| Poos Rolling AR vs Rolling Enet | 12 | 0.1532547 | not significant |
| Poos Fixed AR vs Fixed Enet | 1 | 0.006079946 | not significant |
| Poos Fixed AR vs Fixed Enet | 3 | -0.01782638 | not significant |
| Poos Fixed AR vs Fixed Enet | 12 | 0.143628 | not significant |

Table 9: Diebold-Mariano Test Statistics for Different Comparisons and Horizons

Table 9 shows the differences for the CV methods and for the different horizons in the forecasted values were not singificant between AR and Enet models.

### 3.3 within model comparison

| **Comparison** | **Horizon** | **Db Test Statistic** | 5% significance |
|---|---|---|---|
| kfold AR vs poos-rolling AR | 1 | -0.6641859 | not significant |
| kfold AR vs poos-rolling AR | 3 | -1.163307 | not significant |
| kfold AR vs poos-rolling AR | 12 | -0.5528419 | not significant |
| kfold AR vs poos-fixed AR | 1 | -1.068898 | not significant |
| kfold AR vs poos-fixed AR | 3 | -1.234134 | not significant |
| kfold AR vs poos-fixed AR | 12 | -0.5291777 | not significant |
| kfold enet vs poos-rolling Enet | 1 | -0.001392839 | not significant |
| kfold enet vs poos-rolling Enet | 3 | 0.1448535 | not significant |
| kfold enet vs poos-rolling Enet | 12 | -0.164353 | not significant |
| kfold enet vs poos-fixed Enet | 1 | -2.594716 | significant! |
| kfold enet vs poos-fixed Enet | 3 | 1.132867 | not significant |
| kfold enet vs poos-fixed Enet | 12 | 0.1546035 | not significant |

Table 10: Diebold-Mariano Test Statistics for Different Comparisons and Horizons

Table 10 shows the differences for the CV methods within the corresponding models and for the different horizons. There didn't seem to be a statistically difference between the kfold and poos for each model although we did see a statistical signficicance for kfold enet vs poos-fixed enet even for 1 percent significance level with critical value 2.58.

# 4 Discussion

## 4.1 Comparison k-fold vs pseudo-out-of-sample AR(p) models

### 4.1.1 AR as a benchmark model

Prior performance of CV models on AR(P) models fulfils two purposes: First, univariate linear models are easy-to-interpret benchmarks to evaluate more complex ML models, and second, their goodness of fit is more traditionally evaluated in-sample with other statistical techniques such as AIC [5]. Therefore it interesting to see if ML practices such as CV improved in-sample fit. To further evaluate the performance of statistical vs ML performance on macroeconomic forecasting, one can decide to include multivariate models such as VAR [23] or non-linear threshold AR (TAR) model of Tong, or the smooth transition AR (STAR) model [25, 8]. However, within the scope of this study, it was chosen to focus on the performance of CV on Machine learning models with more than one parameter, using simpler AR(p) as a benchmark model.

### 4.1.2 Limitations of the autoregressive modelling

The AR(p) CV is used as the benchmark model to analyze the performance of machine learning models using two discussed cross-validation methods. Nevertheless, it is important to highlight the limitations of our benchmark model. AR(p) models fail to capture high-volatility periods like recessions [10]. Even though, those statistical models do not capture the long forecasting horizon dynamic structure of the unemployment time series, they show some usefulness in short-term predictions [19].

### 4.1.3 Akaike Information Criterion as a benchmark for parameter selection

AIC is often used as an in-sample selection method for statistical models. In this study it was chosen to use AIC as an order -selection method as a comparison/benchmark to the CV selection in AR and Enet. Information criteria such as AIC evaluate goodness of fit based on likelihood function, the number of regressors and the sample size. In Goulet Coulombe et al., 2019, AIC chose significantly shorter lag orders than CV models when performed on the FRED-MD database [9]. However, in this study, AIC chose the AR(3) model, which has 2 more lags than the K-fold selected model, and 1 more lag than the best-performing POOS model.

The AIC was not performed for in sample order selection the Enet model. However, the forecasting RMSE of the AIC-selected AR(3) model is treated as a benchmark for comparison of all models. The expectation was that CV-selected models perform better than AIC-the selected model for forecasting at H=1 , H=3 and H=12. This statement is re-evaluated in the upcoming section.

### 4.1.4 POOS vs K-fold AR(p)

The main insights coming from the comparison between two CV methods for AR(p) models are in regards the in-sample and forecasting performances.

In-sample POOS performs better than K-fold. In the case of K-fold algorithm, the variation of number of folds parameter k did not influence the model selection. For all the folds the AR(1) model was selected with RMSE of 0.017. However, for POOS, choosing different window sizes and opting for either a fixed or rolling window did result in different model selection. The best-performing model was selected with rolling window of size 36, which corresponds with an RMSE of 0.0138. Surprisingly, the in-sample fit of POOS was higher than k-fold, which is counterintuitive considering k-fold could use the entire training set for model selection.

In case of the forecasting performance for horizons 1,3 and 12, the K-fold slightly outperforms POOS and also the benchmark AIC. This was also the case in the study of Goulet Coulombe, Leroux, Stevanovic, & Surprenant, however there, the differences were even smaller. AR(p) models are simple, and considering K-fold and POOS both choose very similar orders (order 1 vs order 2 respectively), the resemblance of forecasting results is not surprising. Perhaps when unleashed on different data, more varying orders by different CV methods would have been selected, and the differences would have been more severe. Machine learning models like the Enet regression model often have more complex forms than AR that are designed to capture patterns and dependencies in the data. As the prediction horizon increases, these models have more time to learn complex relationships and dependencies, leading to improved performance. Therefore, the difference between POOS and CV model selection on longer horizons would have more implications than for a univariate model like AR.

## 4.2 Comparison K-fold vs Pseudo-out-of-sample Enet

### 4.2.1 POOS CV Enet

The pseudo-out-of-sample CV method outperforms the K-fold CV method when forecasting in time series data, nevertheless, its effectiveness depends on the specific choice of model and performance metrics used, which will be discussed below.

There are several methods possible when performing the pseudo-out-of-sample CV. Four widely utilized methods are: 1) fixed-origin, 2) rolling-origin-recalibration, 3) rolling-origin-update, and 4) the rolling-window evaluation method [24]. In the result section, it is illustrated that when using different options for hyperparameter tuning with pseudo-out-of-sample cross-validation, different parameters ($\alpha/\lambda$) and thus different final models are selected. A higher $\alpha$ indicates a model with more weight of the ridge regression and a lower $\alpha$ indicates a model with more weight of the LASSO regression. As presented in Table 7, the best-performing model for the forecast period is the retrained pseudo-out-of-sample cross-validated model with the fixed window [27].

In this study, the main focus was on the rolling-origin evaluation for the pseudo-out-of-sample cross-validation instead of the fixed-origin method. The rolling origin refers to the origin of the first tested value. In the rolling-origin approach, the latest test value is added to the train set for predicting the next (test)value**.** Tashman (2000) has described three major advantages of the rolling-origin evaluation compared to the fixed-origin method [24]. Namely, the model is more adaptive to changing conditions and is less sensitive to events that are unique for a certain period because it does not rely on a single time origin. Secondly, the rolling-origin evaluation generates more data points per step ahead, which facilitates a more efficient and dynamic forecast. And third, the continuous updating of the forecast origin provides a more diverse set of error metrics. This may lead to a more precise understanding of the forecast accuracy, as it considers more different time periods and conditions rather than averaging errors from a single origin point. Further, calibrating the model with the rolling-origin approach is the preferred model due to the continuous optimization of the model. On the other hand, it may be computationally more intensive compared to non-calibrating, especially in large data sets. This aligns with our best-performing Elastic Net model, the retrained pseudo-out-of-sample selected model. A possible explanation for the results is that the model retrains and adapts dynamically to changes in the data over time, this can capture more time series patterns which may evolve more easily compared to the non-calibrated method.

The paper of Inoue et. al (2016) has described that the forecasting performance is sensitive to the choice of window size [12]. Choosing the optimal window size leads to improvements in the forecasting error. In this study, the optimal window size differed with the options fixed or the rolling initial window. For the rolling window, the best-performing initial window size was 36 periods, while for the rolling window approach, the best-performing initial window size was 12 periods. Pesaran et al. (2017) have described the use of cross-validation for determining the optimal initial window size for pseudo-out-of sample forecasting is a successful method for the model selection [21]. Therefore it can be advised while using out of sample method for forecasting to add the initial window size as a variable which can be determined with the cross-validation method.

### 4.2.2 K-fold CV Enet

As displayed in table 4 the best performing K-Fold model is the 10 K-fold model which selected for an $\alpha$ of 0,1. For the 5 and 20 K-fold model an $\alpha$ of 0 was selected which implies a high weight of the ridge regression in the Elastic Net model.

It is generally assumed that a higher k implies each model is trained on a larger training set and tested on a smaller test fold. Theoretically, this could lead to a lower prediction error. However, a lower k implies each model is trained on a smaller training set and tested on a larger test fold. Therefore there is a higher chance of different data in the test/train fold and thus we can expect a higher prediction error. A study that determined the optimal number of folds to use in a K-fold CV for a neural network machine learning model concluded the optimal folds are between 10 and 20 [20]. This aligns with our results the K-fold CV of 10 outperformed the K-fold CV of 5 and 20 when comparing the RMSE of the predictions. Although, the different strategies were a leading factor for the different obtained RMSEs when applying the cross-validation. Therefore we can conclude there is no standard fold sizes when applying the K-fold CV. It is advised to not exceed the number of 20.

### 4.2.3 POOS vs K-fold Enet

According to the Diebold-Mariano test in 3.3 within model comparison the only significant difference was between kfold10 and poos-fixed12 with an horizon of 1 but Kfold10 and POOS-Roll36 have exactly the same root mean squared error so k-fold and poos behave similarly if you compare the optimal models.

A negative Diebold-Mariano test indicates that the K-fold enet model has a lower root mean squared error than the Pseudo-out-of-sample enet model. Even though 3 out 6 comparisons between kfold enet and POOS enet showed that kfold has a smaller root mean squared error only 1 out of the 3 comparison were statistically significant. The other 3 out of 6 comparison between kfold enet and POOS enet had a positive Diebold-Mariano test statistic but it wasn't significant. So in conclusion k-fold and poos don't seem to outperform each other in case of the enet regression model.

When comparing kfold with POOS within Enet, in this study kfold performed really well compared to POOS. However, Depending on the window selection of POOS, POOS was able to compete with k-fold and reach the same accuracy. This is supprising since the alpha and lambda selected where very different between the best performing poos and the best performing kfold. However, considering the size of observations (130), a different research set up could have led to different results. In the end one could suggest from this study, that k-fold and poos performed similarly for forecasting.

### 4.3 Bias-Variance trade-off and over- or underfitting with CV

Both K-fold and POOS-CV have the bias-variance trade-off when assessing model performance. A high bias can lead to underfitting while high variance to overfitting when the training data is fitted closely but the performance is bad on the unseen data [6]. However, for time series data, there is the extra risk that future values are used to determine older timeframes in the case of k-fold CV. Which can increase overfitting even further. On the other hand, k-fold allows for the entire dataframe to be used for forecasting, which could increase accuracy as well [11]. Therefore, in-sample fit can outperform later h-step forecasting predictions.

In the case of K-fold CV, increasing the value of K reduces bias because training is done on larger set. However, variance is increased due to the test set becoming smaller. It works also vice versa when the value of K is smaller what increases bias but the variance is decreased [15]. The results of the in-sample fit of the AR(P) models with are surprising, since the number of k-fold did not have a significant influence on results or on model selection. For Enet, in-sample fit as well ass forecasts RMSEs were best at 10 folds.

In the case of POOS CV with the rolling forecast, the model focused on more recent predictions and less on the past. This allows the adoption of the model to changing conditions. Also, smaller initial window may reduce bias but increase variance. Therefore, POOS CV has the advantage of high adaptability to changing conditions and lower bias but it comes with the increased variance [9].

### 4.3.1 Forecasting

In this study, it was chosen to fix the optimal hyperparameters for both Enet and AR on the CV decided in the 80% for further forecasting performance in the test set. This means that for each new prediction, the hyperparameters were not recalibrated. However, the rolling window approach was used for the prediction using tsCV function developed by C.Hyndman, of which the theoretical background is provided in a 2018 paper by Bergmeir, C.Hyndman and Koo [4]. This means, that for each additional prediction, the coefficient phi of the AR model is recalibrated, which is illustrated in Figure 2. For Enet the same method has been applied with a rolling window forecasting code, where the data was updated for this means that for each additional predictions, the inclusion of independent variables is re-elected. Recalibration of order selection is possible but was left out of this study. Therefore for further research it would be interesting to see if recalibation would lead to different model selection per prediction, and what the effect would be on the overall in-sample fit and forecasting accuracies.

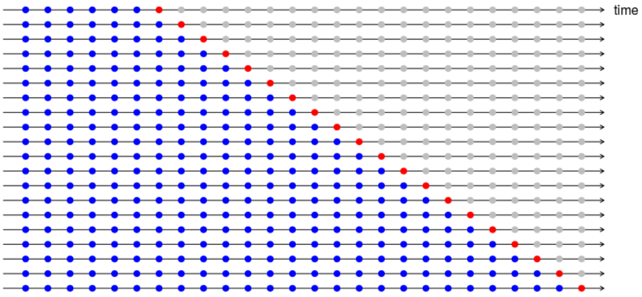

Figure 2: Rolling window prediction approach tsCV, Retrieved from the blog of Rob J. Hyndman.

## 4.4 Conclusions

In this paper we assessed the performance of both K-fold and POOS cross-validation methods for hyperparameter tuning and model evaluation. Although no significant differences between ML and AR models where found in terms of forecasting, different CV methods and CV tuning did result in different model selections. These results are worthy of further testing on different ML models and different datasets. As also the 2019 example paper used in this study did. This study contributed to this outcome by testing different window and fold setting for POOS and k-fold CV respectively, within a single model as well as across models.

### 4.4.1 Empirical implications of CV and Machine Learning in Macroeconomic datasets

The increase of available data calls for increasingly sophisticated modelling methods. CV is extremely useful for model selection and parameter tuning since it can be applied to all models. In machine learning models, CV can help determine hyperparameters of all kinds. However, using CV methods like k-fold on time series is perceived as theoretically incorrect, as they use future observations to describe past observations. However, the main advantage of the discussed k-fold CV techniques is the use of the whole dataset. This way the entire data set could contribute to training the model, which is especially advantageous for sets with fewer observations, such as monthly or quarterly macroeconomic data [11]. In this study, it was also illustrated that they performed well compared to the ¨correct¨ POOS method on longer forecasting horizons, since there was no significant difference. Therefore, empirically, this study would support the use of k-fold CV on macro-economic forecasting, as also the paper of Bergmeir in 2018 did [4]. This study aimed to contribute to the further evolvement of machine learning method and theory tweaking the the field of macroeconomic.

# References

[1] Wat doen wij? — cpb.nl.

[2] Antonello Agostino, Luca Gambetti, and Domenico Giannone. Working paperseriesno1167/april2010 macroeconomic forecasting and structural change, 04 2010.

[3] Richard Ernest Bellman. *Dynamic programming*. Princeton University Press, 1957.

[4] Christoph Bergmeir, Rob J. Hyndman, and Bonsoo Koo. A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics Data Analysis*, 120:70–83, 04 2018.

[5] Miguel Bonilla, Jason Mcdonald, Tamas Toth, Bivin Sadler, and Sadler. Traditional vs machine learning approaches: A comparison of time series modeling methods. *SMU Data Science Review*, 7, 2023.

[6] Justin Domke. Overfitting, model selection, cross validation, bias-variance. *Notes 2*, 2023.

[7] Jean-Marie Dufour and Dalibor Stevanović. Factor-augmented varma models with macroeconomic applications. *Journal of Business Economic Statistics*, 31:491–506, 2013.

[8] Zhaoxing Gao, Shiqing Ling, and Howell Tong. Tests for tar models vs. star models–a separate family of hypotheses approach. *Statistica Sinica*, 28:2857–2883, 2018.

[9] Philippe Goulet Coulombe, Maxime Leroux, Dalibor Stevanovic, and Stéphane Surprenant. How is machine learning useful for macroeconomic forecasting?. *University of Pennsylvania, Université du Québec à Montréal*, 05 2019.

[10] Pablo Guerrón-Quintana and Molin Zhong. Macroeconomic forecasting in times of crises. *Finance and Economics Discussion Series*, 2017, 02 2017.

[11] Christina Han. Cross-validation for autoregressive models. *ThinkIR: The University of Louisville's Institutional Repository*, Paper 3958, 08 2022.

[12] Atsushi Inoue, Lu Jin, and Barbara Rossi. Rolling window selection for out-of-sample forecasting with time-varying parameters. *Journal of Econometrics*, 196:55–67, 01 2017.

[13] Gareth James, Daniela Witten, Trevor Hastie, and Robert Tibshirani. *An Introduction to Statistical Learning*. Springer New York, 2013.

[14] Venus Khim and Sen Liew. On autoregressive order selection criteria on autoregressive order selection criteria, 03 2004.

[15] Ron Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. *Computer Science Department*, 03 2001.

[16] Lisa-Cheree Martin. Machine learning vs traditional forecasting methods: An application to south african gdp, 2019.

[17] M McCracken, Michael W., and Serena Neg. Fred-md: A monthly database for macroeconomic research. *Journal of Business Economic Statistics*, 34:574–589, 09 2016.

[18] L.E. Melkumova and S.Ya. Shatskikh. Comparing ridge and lasso estimators for data analysis. *Procedia Engineering*, 201:746–755, 2017.

[19] Alan L. Montgomery, Victor Zarnowitz, Ruey S. Tsay, and George C. Tiao. Forecasting the u.s. unemployment rate. *Journal of the American Statistical Association*, 93:478–493, 06 1998.

[20] Opeoluwa Oyedele. Determining the optimal number of folds to use in a k-fold cross-validation: A neural network classification experiment. *Research in Mathematics*, 10, 05 2023.

[21] M. Hashem Pesaran and Allan Timmermann. Selection of estimation window in the presence of breaks. *Journal of Econometrics*, 137:134–161, 03 2007.

[22] RITEI SHIBATA. Selection of the order of an autoregressive model by akaike's information criterion. *Biometrika*, 63:117–126, 1976.

[23] Lemya Taha. Forecasting time series using vector autoregressive model. *University of Baghdad*, 09 2021.

[24] Leonard J. Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, 16:437–450, 10 2000.

[25] Howell Tong and Iris Yeung. Threshold autoregressive modelling in continuous time. *Statistica Sinica*, 1:411–430, 1991.

[26] Australia Treasury. Macroeconomic forecasts: Purpose, methodology and performance, 1996.

[27] Hui Zou and Trevor Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)*, 67:301–320, 2005.

# 5 Appendix

## 5.1 Descriptive Statistics

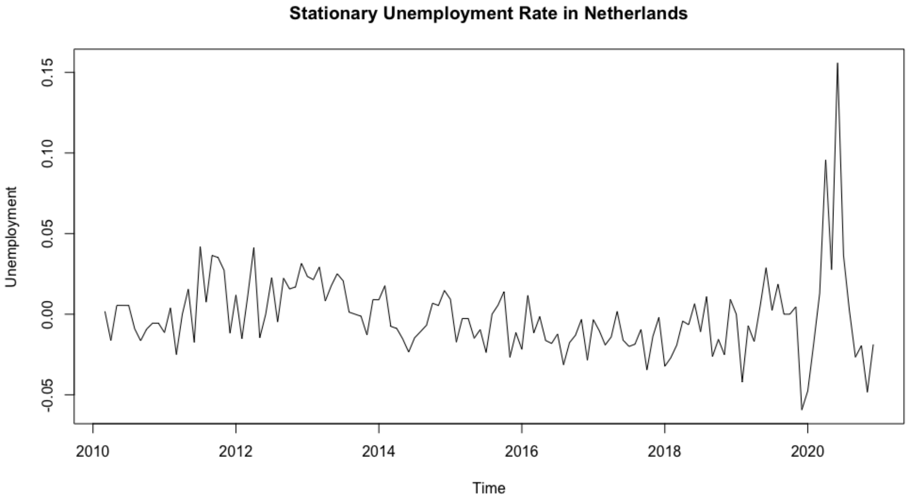

Figure 3: The stationary Unemployment rate in the Netherlands (1st difference logarithm).

Figure 3 presents the stationary unemployment rate in the Netherlands. It has been transformed to achieve stationarity through the first difference of the logarithm. The time series data spans over 11 years, starting from 2010 till 2020. The plot shows the fluctuations in the unemployment rate, smoothed to remove trends and seasonality, thus allowing for the analysis of the underlying stationary process.

### 5.2 AR(p) models

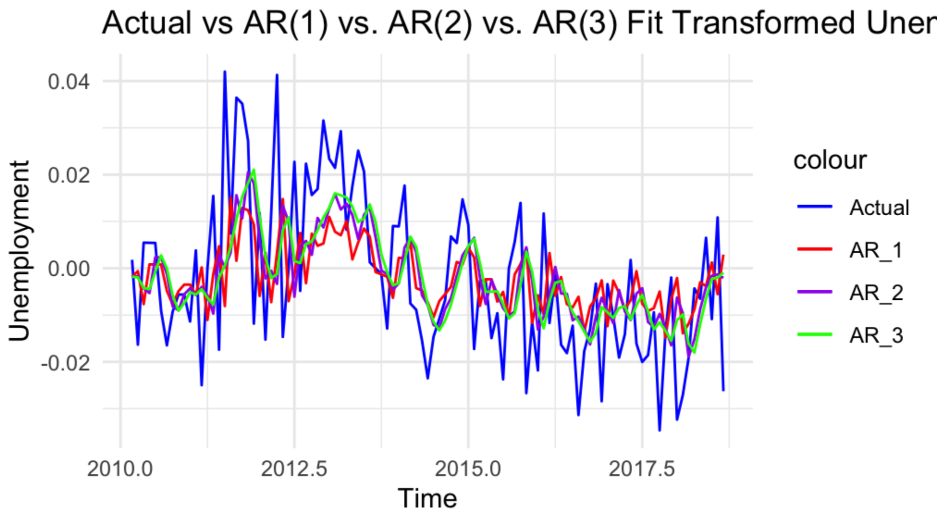

Figure 4: Illustration of in-sample fit The comparison of actual data to the fitted AR(1) and AR(3) models selected by CV and AIC respectively.

Figure 4 presents a plot of the in-sample model fit of selected AR(p) models by different techniques (cv, aic)

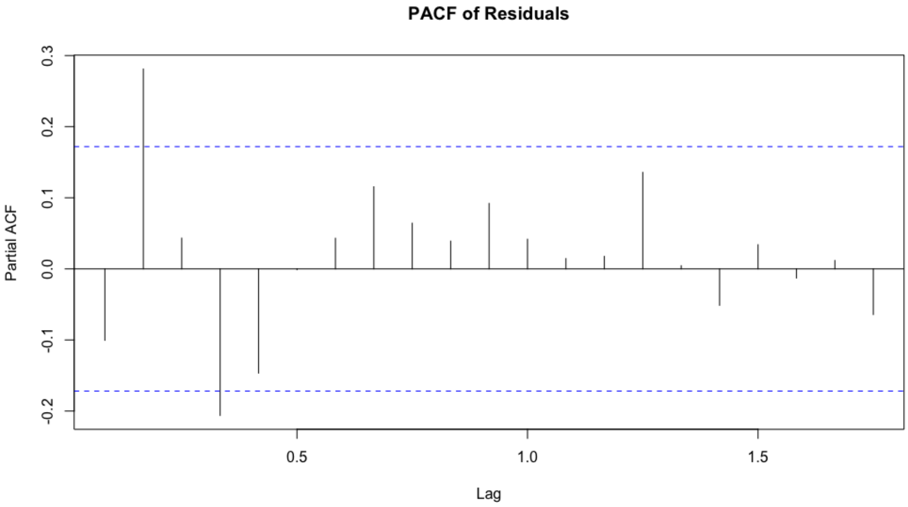

Figure 5: Partial autocorrelation function of residuals for AR(1).

The figure 5 displays the Partial Autocorrelation Function (PACF) of residuals from AR(1) model. The PACF plot is used to identify the autocorrelation in the residuals at different lags. In an ideal AR(p) model, the residuals should not exhibit significant autocorrelation. We analysed all ar(p) models for stationarity using PACF to ensure the build in ¨transform¨ function gave the right tranformation.

## 5.3 ML models

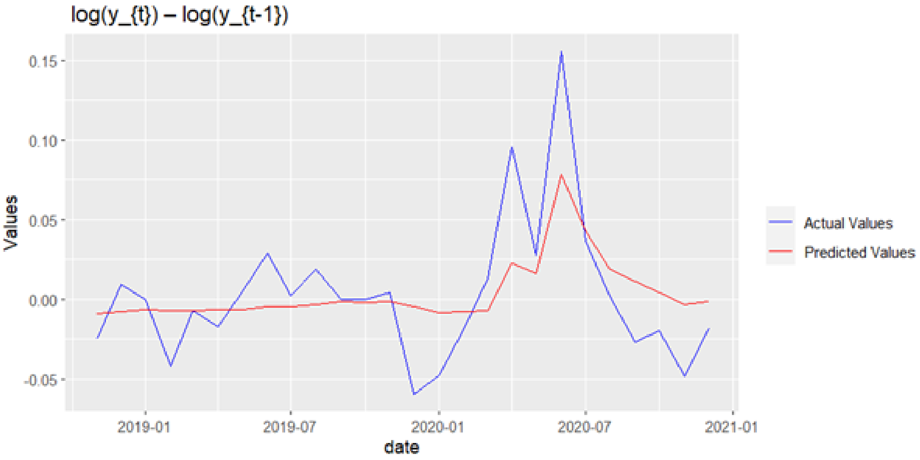

Figure 6: Pseudo out of Sample with horizon = 1

The figure 6 displays the Pseudo out of sample with the right transformation to achieve stationarity which is taking the first logartihmic difference of the dependent variable of unemployment. The graph is forecasted with a horizon of 1.

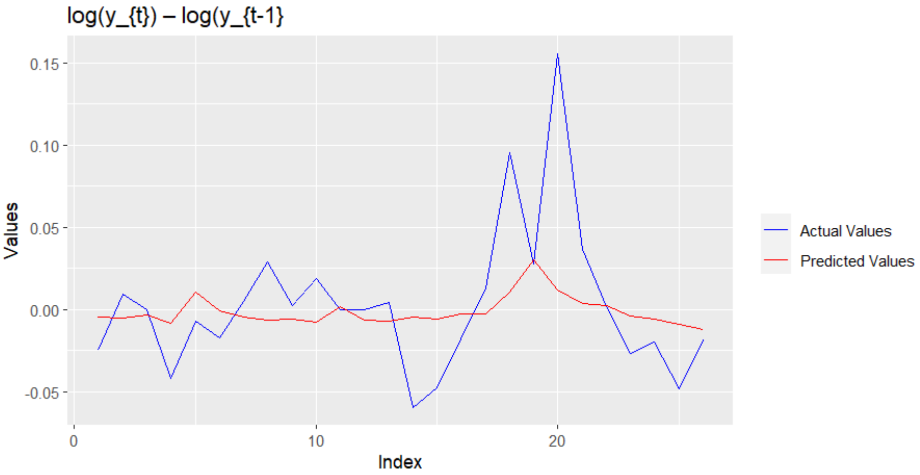

Figure 7: kfold with horizon = 1

The figure 7 displays the kfold with the right transformation to achieve stationarity which is taking the first logartihmic difference of the dependent variable of unemployment. The graph is forecasted with a horizon of 1.

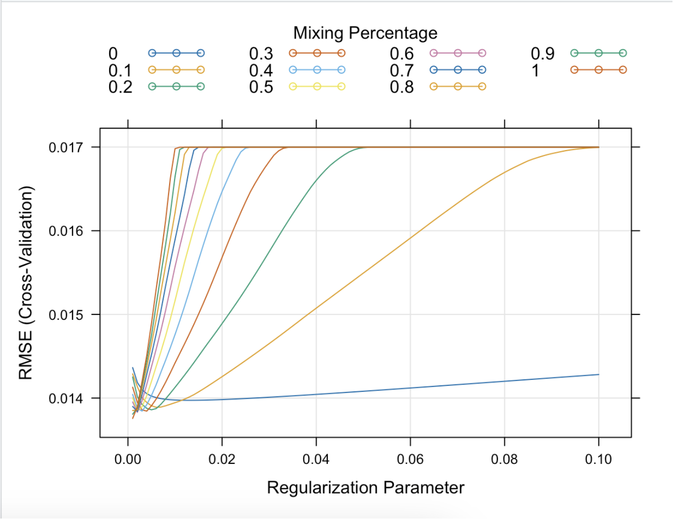

Figure 8: Enter Caption

Figure 8 reflects how the Elastic Net model performed hyperparameter tuning between the lambda and the alpha value. Choosing the optimal line between both values which gives the lowest RMSE of the cross-validation.