

OTDK-dolgozat

2025

POZSONYI KÖZGAZDASÁGTUDOMÁNYI EGYETEM
GAZDASÁGINFORMATIKAI KAR

**MAKROGAZDASÁGI VÁLTOZÓK ELŐREJELZÉSE GÉPI TANULÁSI
MÓDSZEREK SEGÍTSÉGÉVEL: SZLOVÁKIA ESETE**

**FORECASTING MACROECONOMIC VARIABLES USING MACHINE
LEARNING: THE CASE OF SLOVAKIA**

Csápai Ádám

Konzulens: prof. Ing. Martin Lukáčik, PhD.

Kézirat lezárásának a dátuma: 2023.09.29.

Contents

Introduction	3
1. Current state of the field at home and abroad	5
2. Aims.....	11
3. Methodology and methods.....	13
3.1. RStudio.....	13
3.2. Data preparation and preprocessing methods	14
3.3. Benchmark model.....	15
3.4. Regularized Least Squares	15
3.5. Ensemble Machine Learning.....	17
3.6. Neural Networks	19
3.7. Support Vector Machine.....	21
3.8. Performance evaluation.....	22
4. Data	27
5. Results	31
5.1. Industrial production.....	31
5.1.1. Forecasting industrial production using hard macroeconomic data ..	31
5.1.2. Forecasting industrial production using common factors.....	40
5.1.3. Forecasting industrial production based on soft indicators	42
5.2. Inflation.....	45
5.2.1. Forecasting inflation based on hard macroeconomic data.....	45
5.2.2. Forecasting inflation using common factors	53
5.2.3. Forecasting inflation based on soft indicators	55
6. Discussion	60
7. Conclusion	64
References	66

INTRODUCTION

In this day and age, preparing accurate macroeconomic forecasts is of paramount importance. Thankfully, with the advent of big data and advancements in statistical learning theory and computational capacity, economists have a whole new host of methods to choose from. Some of these methods are based on machine learning and they have gained popularity in macroeconomics in the past ten, but mainly in the past three to four years. Given the field is new, global evidence on the performance of these methods is scarce but really compelling. Local evidence for Slovakia, however, is nonexistent.

We choose to do macroeconomic forecasts using machine learning not only because of the compelling global evidence, but also because one of the top researchers in the National Bank of Slovakia inspired us to devote more attention to the subject. He is interested in this forecasting himself. Once we started to see the importance of our results, they became the main driver to finish this study.

In the first chapter we concisely summarize the existing literature on the subject. To our knowledge, there is no comprehensive book on the usage of machine learning in macroeconomics, so we mainly reference actual articles on the subject. We provide justification for choosing our benchmark, explain machine learning the macroeconomic context and cite results of other authors.

In the second chapter we formulate our main and partial aims, as well as our hypotheses. The principal aim of this paper is to thoroughly analyze the forecasting performance of the selected machine learning models compared to a benchmark model using a novel large dataset consisting of all the important macroeconomic variables of the Slovak economy.

In the third chapter we describe our methods. We provide justification of why we select the models we select, then describe stationarization, the selected models and the performance evaluation metrics.

The fourth chapter briefly describes the database of the National Bank of Slovakia. We also explain the choice of the forecasting period and why we choose to include the Great Recession instead of COVID.

In the fifth chapter we present our results and in the sixth chapter we briefly summarize them. We show that we cannot reject our hypotheses from chapter 2 and that our results have important implications. We conclude that the institutions preparing official macroeconomic forecasts in Slovakia should begin the testing of the empirical applications of machine learning models to capture new information in official macroeconomic forecast, while researchers

should devote more attention to directional accuracy measures, soft indicators and regularization instead of PCA.

1. CURRENT STATE OF THE FIELD AT HOME AND ABROAD

Forecasting provides a crucial tool for evaluating the state and direction of the economy and thus serves as the backbone of economic decisions. Government budgets depend on forecasted macroeconomic variables, such as GDP, inflation and unemployment. Policymakers, such as the Central Bank use forecasting models to accurately time policy interventions. Current forecasting models, however, might be inadequate to capture the underlying relationships between economic variables. Medeiros et al. (2019) provide an example. They show that governments and international institutions, mainly the ECB, systematically overestimate projected inflation. Since monetary policy is based on inflation expectations, the consistent overestimates might lead to welfare losses and inaccurate expectation anchoring. It is clear, therefore, that forecasting models are important tools for macroeconomists and finding appropriate models is key for maximizing the welfare of economic participants.

According to Mahajan and Srinivasan (2019), macroeconomists usually build structural models by which they aim to capture the underlying stochastic data generating process (DGP) of macroeconomic variables. These models have two main disadvantages. Firstly, as Jung et al. (2018) states, when the modelled relationships are incorrectly specified the data quality is irrelevant. The model can only be as good as its specification. Secondly, the model is rendered invalid by changes in the data generating process. As an example, Stock and Watson (2007) forecasts US inflation and shows how the coefficients of their estimated integrated moving average model change twice during the estimation period. They conclude that if the DGP has changed twice already, it can change again. It is clear therefore, that the structural models have important disadvantages macroeconomists must consider when trying to forecast the economy.

Breiman (2001) proposes a solution to the shortcomings mentioned in the previous paragraph and advocates for the use of data driven, non-structural models. He supports models that do not make any assumptions about the DGP, making them insensitive to its changes. In addition, Breiman (2001) also states that models should make no assumptions about the relationships between the predictors and target variables so they can be insensitive to model misspecifications. Machine learning models meet these criteria. Before we continue with machine learning, however, we provide justification for the selection of our benchmark and compare machine learning methods to the well-established tools used in the literature.

Stock and Watson (2007) use a time-varying unobserved components model and Stock and Watson (2010) use Random Walk and univariate time series models and show that it is exceedingly difficult to systematically improve upon these models when it comes to

macroeconomic forecasting. Faust and Wright (2013) survey the existing literature and further support the conclusion of these papers. Because of this, we select Random Walk and Arima models as our benchmarks. These models usually serve as the standard benchmarks in the economic forecasting literature.

The literature mainly considers models based on common factors as the main tool for data driven macroeconomic forecasting and compares their performance to the previous benchmark. This is because even though the benchmark models of the previous paragraph are hard to beat, Stock and Watson (2002) points out that models based on common factors can outperform them when forecasting US time series data. Because of this, factor models become the literature standard for data driven forecasting. Due to the preference for common factors, Shintani (2005) and Maehashi and Shintani (2020) employs them as a basis for machine learning models. They conclude that even tough methods based on common factors can beat the benchmark, supporting the conclusion of Stock and Watson (2002), they generally underperform machine learning models based on hard data. To sum up, Random Walk and ARIMA models, as well as factor models are well-established tools for macroeconomic forecasting.

Although the previously mentioned data driven models are important forecasting tools, they also have their own limitations. To begin with, Pescaroti and Zaman (2011) state that for these models to be appropriate the linear relationship between the variables in the dataset must always hold. They say that largely because of this limitation, the models were incapable of identifying business cycles, extreme volatility and unanticipated regime switches during the 1980s and 1990s in the US. This finding is further supported by Sanyal and Roy (2014). To continue with, Bolhuis and Rayner (2020) compare the theoretical properties of Ordinary Least Squares (OLS), another popular method, to that of factor models and machine learning. Using OLS as an example, they show that machine learning is more appropriate than factor models not only empirically, as stated by Shintani (2005) and Maehashi and Shintani (2020), but also theoretically. According to them, the forecasts made by OLS can be inappropriate due to predictor relevance, nonlinearity, collinearity and dimensionality. Factor models, which have been the main tools for data driven forecasting in the past decades, can address the last two issues. These models are, however, incapable of handling the first two problems, leading to generally less accurate forecasts. Machine learning methods are an appropriate tool to capture nonlinear dynamics in a high-dimensional dataset. They learn complex past relationships but do not over-extrapolate them into the future. To sum up, simple data driven models are hard to beat, but their disadvantages led authors in the past decade to consider machine learning models as tools for macroeconomic forecasting.

Another advocate for machine learning models is Vargas (2020). He acknowledges that macroeconomic variables are usually forecasted by either classical econometric models or data driven models. He continues, however, that in the past decade, mainly due to advancements in computing power and statistical learning theory, as well as an increase in the availability of big data, machine learning models have started to get employed by economic forecasters. These models have previously been employed mainly for classification problems, like predicting loan delinquencies or consumer purchasing decisions and they outperformed classical models. These variables are discrete, but the past decade has shown that these models can also forecast continuous time series data like GDP or inflation and, according to of Vargas (2020), the unique properties of the data - mainly nonlinear relationships - captured by these models can help us better understand asymmetric business cycles, stock market volatility and regime switches. To sum up the previous paragraphs, due to the theoretical disadvantages of previously used models and advances in computing power and statistic macroeconomists have begun the exploration of machine learning methods.

Now that we see that machine learning tools can have potential use cases in macroeconomic forecasting, we can start devoting special attention to them. Before we dive deeper into machine learning models, however, we first explain what they are in a macroeconomic context. Masini et al. (2021) surveys the recent advancements in the field and states that machine learning has multiple definitions. For macroeconomic forecasting, they write, machine learning melds automated computer algorithms with robust statistical techniques to uncover underlying patterns in extensive datasets. Given this description, macroeconomics clearly not focuses on machine learning itself, but on applying the most recent advancements in statistical learning theory to macroeconomic problems, mainly forecasting. Masini et al. (2021) name three categories of machine learning methods, namely supervised, unsupervised and reinforcement learning. They, as we do, use supervised learning methods for forecasting. These methods are fed data organized as input-output pairs, such as in the case of linear regression models. Based on this data they learn a function that maps an input, in our case the predictors, to an output, which is our forecasted value of the target variable. Individual machine learning methods are described in further detail in chapter 3. After understanding the general logic behind machine learning in macroeconomics, we describe the empirical evidence.

To return to the promising empirical applications, Athey (2018) states that in the past decade economists have started to intensively explore the possibilities of using machine learning methods for forecasting tasks if large datasets are available. Although Athey (2018) mainly focuses on forecasting using microdata, Mullainathan and Spiess (2017) explore the

possibility of employing these methods in macroeconomic forecasting tasks. They conclude that macroeconometricians should incorporate machine learning methods into their forecasting toolkit. Chakraborty and Joseph (2017) use machine learning to forecast medium-term inflation and find that these models beat the benchmark by at least 29%. Jung et al. (2018) use Elastic Net, SuperLearner and Recurrent Neural Networks to forecast the macroeconomic data of seven advanced and emerging economies and their methods outperform the benchmark WEO forecast. In summary, the preliminary results are promising, but machine learning has its critiques, as seen in the next paragraph.

The most often cited study claiming that machine learning models are inadequate forecasting tools is that of Makridakis et al. (2018). The authors state that machine learning models are unreliable when it comes to forecasting. They, however, base their conclusion on the use of the simplest models, like simple regression trees, shallow neural networks and support vector machines, while they ignore more advanced methods. As a response to their critique, Medeiros et al. (2019) show that machine learning can enhance the forecasting accuracy by up to 30% in the case of US inflation data. They use a large dataset, multiple methods and analyze both periods of relatively high and low economic uncertainty. The main critique against machine learning is, according to Medeiros et al. (2019), unfounded when we use more advanced methods.

Even though these papers acknowledge that machine learning methods can indeed outperform their benchmarks both at the micro and macro levels, but they largely ignore the identification of the driving forces behind this outperformance. Their second main drawback, as mentioned by Coulombe et al. (2022), is that they are limited in scope. These studies use relatively smaller datasets, forecast a single target variable for only a couple of horizons and usually use one to three models, with the sole exception of Medeiros et al. (2019), which is the main response to the critique of Makridakis et al. (2018). In addition, the only aim of these papers, again with the exception of Medeiros et al. (2019), is to do a forecasting horserace using hard macroeconomic data and focus mostly on one metric (either MSE or RMSE). They ignore the performance of machine learning methods based on common factors, only explored in Shintani (2005) and Maehashi and Shintani (2020). In addition, they only focus on minimizing the forecast error while ignoring accuracy altogether. What is more, they mostly ignore the explanation of why these methods do so well and what might cause the outperformance of one method relative to the other.

These shortcomings are identified by Coulombe et al. (2022), who design a forecasting exercise to provide deeper understanding of the underlying factors driving the outperformance.

They use US data for the past 40 years, forecast multiple target variables for five different horizons using multiple models and identify the key drivers of success behind machine learning methods. Firstly, their results are positive and statistically significant. They show that nonlinearities play a statistically significant role in the data generating process of the USA. In addition, regularization can substantially increase the forecasting performance, even better than factor models. Their third contribution is showing that k -fold cross-validation is better for model selection than selection based on the information criteria when it comes to machine learning models.

Medeiros et al. (2019) also show why regularization and nonlinearities are important and how factor models fare when compared to machine learning. Firstly, with regards to regularization, in the last few years, due to the widespread adoption of big data, macroeconomists have begun to question the importance of all variables in the large datasets. Thus, besides factor models, they have started to employ shrinkage and variable selection procedures as well. Medeiros and Mendes (2016) and Giannone et. al (2018) advocates for the usage of Lasso models in macroeconomic forecasting, as they outperform their respective benchmarks. These studies, however, only focus on US inflation data and for short term forecasting. The most comprehensive study is that of Medeiros et al. (2019), who shows that shrinkage methods, although less accurate than tree-based methods in the US, can indeed substantially improve forecasting performance. The authors also find that the high level of aggregation of factor models is inadequate when compared to regularization.

Secondly, the importance of nonlinearities is also highlighted by Medeiros et al. (2019), who also provide a theoretical explanation. Consider inflation for example. The correlation between inflation and employment is nonlinear, being contingent on the level of slackness in the economy. In addition, uncertainty can also cause nonlinearity. For instance, when dismissing workers is costly, the approach to hiring is inherently nonlinear in relation to uncertainty. What is more, the existence of a zero lower bound (ZLB) on nominal interest rates, and the ramifications it has for unorthodox monetary policies, is an additional source of nonlinearity between inflation, employment, and interest rates. Moreover, the role of houses as loan collateral can create interactions with monetary policy and financial intermediation. As observed during the Great Recession, the emergence of a housing bubble can lead to severe credit crashes, and these interactions are inherently nonlinear. Given that our dataset contains both the Great Recession and a relatively long period when interest rates were near or at the ZLB, we assume that nonlinearities play an important role in the analyzed period.

To sum up the theoretical part, we first show why providing accurate macroeconomic forecasts is important. Next, we describe the difference between structural and data driven models. Moreover, we show why we select our benchmarks. What is more, we present the limits of the main forecasting models used in the literature so far and explain how machine learning can overcome them. To continue with, we explain machine learning in the macroeconomic context and show that the empirical applications are promising but limited in scope. Most studies focus on only a few models with few horizons, they are mostly done on large economies and ignore directional accuracy altogether. To end with, we provide a description of why regularization and nonlinear relationships in the data are important to consider.

2. AIMS

In this section we describe the aims of our research. We perform a comparative analysis of machine learning models to forecast Slovakian macroeconomic time series, namely industrial production and inflation. The principal aim of this paper is to thoroughly analyze the forecasting performance of the selected machine learning models compared to a benchmark model using a novel large dataset consisting of all the important macroeconomic variables of the Slovak economy.

To achieve the principal aim, we set the following partial aims:

- There is scarcely any Hungarian or Slovak literature on macroeconomic forecasting using machine learning. In English, there are lots and lots of papers regarding artificial intelligence, and there has been a surge of new papers in the last decade applying these models to macroeconomic forecasting. Our first partial aim is to study these papers, based on which we formulate the aims and select the appropriate methods to accomplish them.
- In chapter 3, we aim to describe the methods in detail. We reference the papers of the original authors but use a simplified and unified notation for every method provided by Maehashi and Shintani (2020).
- We aim to describe the macroeconomic database of the National Bank of Slovakia and the data gathering and preprocessing steps.
- Our fourth partial aim is to thoroughly analyze the performance of machine learning models and see if the below stated hypotheses uphold.
- Our final partial aim is to summarize the results and provide policy and methodological recommendations, as well as suggestions for further research.

To fulfill our aims, we must accept the following primary and secondary hypotheses for both forecasted variables. The primary hypotheses are:

H1	Nonlinearities play a statistically significant role in the data generating process of the Slovak macroeconomic time series.
H2	Regularization can statistically significantly improve the quality of the macroeconomic forecasts of our target variables.

If yes, then when using observable “hard” data, there should be at least one machine learning model both from the Regularized Least Squares and Ensemble Machine Learning categories specified in chapter 3 that statistically significantly outperforms the benchmark at every horizon both in terms of the magnitude of errors and directional accuracy.

Our principal aim is a thorough analysis of model performance, which has no exact definition, but we believe that in addition to the main hypotheses, our following secondary hypotheses, based on theory and the available database, can be considered comprehensive:

H3	There is at least one individual model from each of the two categories that is more accurate and has lower forecast errors than the category average.
H4	When using indicator data, forecasting models based on machine learning are more likely to forecast the correct direction of the change in the variable than the benchmark model.
H5	Machine learning models based on common factors have lower forecast errors and are more likely to forecast the correct direction of the change in the forecasted variable than the benchmark model.
H6	Machine learning models generally outperform the benchmark model for the selected target variables in Slovakia.

Our last secondary hypothesis cannot be exactly defined, so we use expert judgement after consulting with papers, researchers at multiple universities and at the National Bank of Slovakia. The papers cited in chapter 1 are peer-reviewed publications, yet none of them has any model that outperforms their benchmark at every horizon. Despite this, they conclude which model is best and derive scientifically accepted conclusions based on these models. Secondly, these papers report not only results that are significant at the 5% level, but also results that are significant only at the 10% level. One major implication stems from these facts. We aim to see whether machine learning models generally outperform the benchmark. First, we assume that we cannot reject our hypotheses formulated above. If that is the case, then we set the second criteria of general outperformance. Based on the cited papers, we say that when considering all the models at all the horizons, if 70% of forecasts outperform their benchmark at the 5% level and more than 90% at the 10% level, we say that machine learning models generally outperform their benchmarks. These percentages roughly correspond to the papers cited in chapter 1, so we continue with them and if the models meet these percentages, we state that they generally outperform the benchmark.

To sum up, in this section we present the principal aim of the paper, as well as the partial aims and main and secondary hypotheses. In the next section we describe the tools used to achieve these aims.

3. METHODOLOGY AND METHODS

In this subchapter we present the methodology and methods we use to obtain the results. Firstly, we concisely describe the R statistical programming software we use for computations. Secondly, we describe the methods used for data preparation and preprocessing. Thirdly, we present the benchmark model, for which we provide theoretical justification based on empirical results in chapter 1. Fourthly, we present the first group of methods, namely Regularized Least Squares. Fifthly, we present the second group of methods, namely Ensemble Machine Learning. We choose multiple models of these two categories to provide comprehensive overview of machine learning methods and highlight their key advantages identified by Coulombe et al. (2022), namely regularization and the capability to capture nonlinear relationships. There are multiple models available in each category, but we choose three. This ensures that our results are not due to random luck in selecting one model but it also lends robustness to our findings when multiple models arrive at the same conclusion. Sixthly, we present the two types of neural networks we consider for forecasting. Seventhly, we present the Support Vector Machine Regression. To end with, we describe how we evaluate the forecasting performance of the models.

3.1. RSTUDIO

RStudio¹ is an integrated development environment (IDE) specifically designed for the R programming language, which is widely used in scientific research and data analysis. It comes with various packages and provides a user-friendly interface for writing, executing, and debugging R code, making it a valuable tool for researchers conducting statistical analysis, data visualization, and scientific modeling. RStudio offers features like syntax highlighting, code completion, and integrated data visualization tools, helping us streamline our data-driven workflows, thus it is an efficient, intuitive and easy-to-use language and has extensive packages for machine learning models. Instead of setting hyperparameter values manually, we use these packages to automatically calculate the hyperparameters at each iteration. Because of this, we ensure that we get the optimal values for each individual forecast. In summary, RStudio is an efficient and straightforward tool for macroeconomic forecasting using machine learning, that is why we use it.

¹ <https://posit.co/download/rstudio-desktop/>

3.2. DATA PREPARATION AND PREPROCESSING METHODS

In this subchapter we describe the ADF test for stationarity and the data standardization procedure. We begin with the former.

If a series is nonstationary, conventional hypothesis tests, confidence intervals and forecasts can be strongly misleading. One source of nonstationarity can be a trend in the data generating process. Dickey and Fuller (1979) proposed a formal test for stationarity, named the Dickey-Fuller test. The null hypothesis states that the time series is nonstationary (has a unit root), while the alternative hypothesis states that the series is stationary. We are interested in the augmented version of this test (ADF).

Consider the linear regression model

$$\Delta Y_t = \beta_0 + \delta Y_{t-1} + \gamma_1 \Delta Y_{t-1} + \gamma_2 \Delta Y_{t-2} + \dots + \gamma_p \Delta Y_{t-p} + u_t. \quad (1)$$

The ADF test for a unit root tests the hypothesis $H_0: \delta = 0$ (trend) against the alternative $H_1: \delta < 0$ (stationarity) using the ordinary least squares estimate of the t-statistic. The lag length of the test is given by minimizing the Akaike Information Criterion, written as

$$AIC = 2k - 2\ln(\hat{L}) \quad (2)$$

where k is the number of estimated model parameters and $\hat{L}$ is the maximum value of the likelihood function for the model. If the series is non-stationary, there are multiple ways to stationarize it, such as transforming the variables to logarithms to stabilize the variance, taking the differences of the variables, combining these two or using the Hodrick-Prescott filter. In our work, we use every one of these except the HP filter.

To continue with, we standardize the predictors by transforming them to a variable with a zero mean and unit standard deviation. This is done as

$$\text{Standardized value} = \frac{\text{Actual value of the predictor} - \text{Mean of the time series}}{\text{Standard deviation of the time series}}. \quad (3)$$

In summary, in the previous paragraphs we present the methods by which we preprocess the data in chapter 4.

3.3. BENCHMARK MODEL

This subchapter describes our benchmark. We use either a Random Walk or an ARIMA (Autoregressive Integrated Moving Average) model as a benchmark. RStudio allows for selecting the best model of the two for each forecast based on the Akaike Information Criterion, described in subchapter 3.1.

The Random Walk model is the simplest naïve model and presumes that the forecast of the target variable is its last observed value, written as

$$y_{t+h} = y_t + \epsilon_{t+h} \quad (4)$$

where y_{t+h} is the forecast of the target variable, y_t is the observed value of the target variable at time t and ϵ_{t+h} is the error term. Besides Stock and Watson (2010), Atkenson and Ohanian (2001) also show that despite being a naïve model, the Random Walk is hard to beat in forecasting exercises. Because of this, it usually serves as one of the benchmarks for machine learning forecasting models, such as, for example, in Pratap and Sengupta (2019), Mahajan and Srinivasan (2019) and Medeiros et al. (2019).

The second model is the ARIMA model, also used by Pratap and Sengupta (2019) and, for example, Chakraborty and Joseph (2017). This model nests both autoregressive and moving average models and is thus one of the most versatile methods for univariate time series forecasting. The ARIMA model is written as

$$\phi(B)(1 - B^d)y_t = c + \theta(B)\epsilon_t \quad (5)$$

where ϵ_t is a white noise error term, B represents the lag operator, and ϕ and θ are the AR and MA polynomials of order p and q , respectively, which describe the order of the autoregressive and moving average terms. Moreover, d is the order of integration, representing how many times we must difference the time series to ensure stationarity.

3.4. REGULARIZED LEAST SQUARES

In this subchapter we begin the description of machine learning methods. Firstly, we present the Regularized Least Squares (RLS) methods, otherwise known as penalized regression

methods, used for reducing the number of predictors or shrinking the coefficients by employing a penalty term. These methods minimize

$$\sum_{t=1}^T \{(y_{t+h} - \sum_{i=1}^N \beta_i x_{it})^2 + \lambda J(\beta)\}, \quad (6)$$

where the regularization hyperparameter λ plays a crucial role. Regularized Least Squares methods define the penalty term $J(\beta)$, where $\beta = (\beta_1, \beta_2, \dots, \beta_N)'$, in different ways. In our study we use the Ridge, Lasso and Elastic Net regressions.

Ridge. Hoerl and Kennard (1970) introduce the Ridge regression, which uses an L_2 norm penalty function, written as

$$J(\beta) = \sum_{i=1}^N \beta_i^2. \quad (7)$$

In Ridge, the regression coefficients can approach zero without being exactly equal to it. This implies that Ridge is a shrinkage method and aims to mitigate the overfitting problem by shrinking the estimated coefficients.

Lasso. The second method is the Lasso regression, otherwise known as the “least absolute shrinkage and selection operator” introduced by Tibshirani (1996). Lasso specifies the penalty function by using the L_1 norm as the penalty term. We write the penalty function as

$$J(\beta) = \sum_{i=1}^N |\beta_i|. \quad (8)$$

This formulation introduces a kink at 0 in a constrained minimization problem concerning β . Consequently, numerous coefficient estimates can become exactly zero. Hence, Lasso is regarded as a variable selection procedure.

Elastic Net. Zou and Hastie (2005) show that Lasso outperforms Ridge when a significant number of model coefficients tend to be zero, whereas Ridge is more appropriate when predictors are highly correlated. To get the best of both worlds, we also use Elastic Net. It provides the advantages of both methods simultaneously by introducing a penalty function denoted as

$$J(\beta) = \omega \sum_{i=1}^N |\beta_i| + (1 - \omega) \sum_{i=1}^N \beta_i^2, \quad (9)$$

where ω represents a hyperparameter that determines the relative weights of the L_1 norm penalty and the L_2 norm penalty. The value of ω is determined through cross-validation. Notably, when ω is set to 1, Elastic Net is equivalent to Lasso, while setting ω to 0 renders it equivalent to Ridge. Consequently, Elastic Net encompasses both shrinkage and selection characteristics.

To sum up, this subchapter presents three Regularized Least Squares methods, namely Ridge, Lasso and Elastic Net. The difference between Ridge and Lasso lies in the penalty function, while Elastic Net can combine both models.

3.5. ENSEMBLE MACHINE LEARNING

In this subchapter, we present three Ensemble Machine Learning (EML) methods, namely Bagging, Boosting and Random Forest. These EML methods use decision (regression) trees capable of capturing nonlinear relationships and variable interactions in the data. These trees detect groups of similar observations by generating nodes step by step within the tree.

The process works as follows. Assume that y_{t+h} is the forecast of our target variable. Initially, each observation of y_{t+h} is sorted into nodes based on a selected predictor variable $X_t = (x_{1t}, x_{2t}, \dots, x_{Nt})$. Selected nodes are assigned the value of the sample mean y_{t+h} conditional on a selected predictor. If there are nodes left with no assigned values, they are divided by using the remaining predictors. The process ends when we assign values to all the nodes.

Based on this we can write a regression tree with M terminal nodes as

$$y_{t+h} = \sum_{m=1}^M \theta_m \mathbf{1}_{[x_i \in h_m]} + \varepsilon_{i+h_n}, \quad (10)$$

where $\mathbf{1}_{[\cdot]}$ represents the indicator function, R_m is a portion of the space of X_b and θ_m gives us the sample mean of y_{t+h} conditional on $X_t \in R_m$. The aim of the estimation is to select the tree structure which minimizes $\sum_{t=1}^T \varepsilon_{i+h}^2$. To find this tree we select sorting variables from X_t and pick splitting values at each node. We use the literature standard algorithm of Breiman et al. (1984) to select the optimal values for sorting and splitting, respectively.

Regression trees do well if nonlinearity and variable interactions are present. On the other hand, their out-of-sample forecast performance is generally suboptimal since they are sensitive to changes in the dataset. To solve this shortcoming, regression trees are usually not used only

by themselves in the forecasting literature, only as a basis of some EML method. In the next paragraphs, we present these methods.

Bagging. Bagging stands for bootstrap aggregating. Breiman (1996) presents this method and shows that it can substantially improve forecast accuracy when compared to a single decision tree. Bühlmann and Yu (2002) show that Bagging reduces forecast errors for i.i.d. data, while Inoue and Kilian (2008) confirm the outperformance for time series data.

In bagging, we generate bootstrap samples of $X_t = (x_{1t}, x_{2t}, \dots, x_{Nt})'$ and y_{t+h} B times and then a regression tree computes the forecast $\hat{y}_{t+h}^{(b)}$ for each bootstrap sample $X_t^{(b)}$ and $y_{t+h}^{(b)}$. In the last step we average the forecasts of each bootstrap sample $B^{-1} \sum_{b=1}^B \hat{y}_{t+h}^{(b)}$, which diminishes the overfitting and large volatility problem of individual forecasts.

Random Forest. Random Forest is a derivative of Bagging and is described in detail by Breiman (2001). Bagging forecasts are stable only if regression trees of different bootstrap samples are not highly correlated. If they are, averaging might not be sufficient to reduce forecast variance, since individual regression trees in bootstrap samples are correlated. Hastie et al. (2009) proposes a dropout procedure for decorrelating regression trees of individual samples. More precisely, the set of predictors $X_t = (x_{1t}, x_{2t}, \dots, x_{Nt})'$ is reduced by randomly drawing subsets $X_t^* = (x_{1t}^*, x_{2t}^*, \dots, x_{kt}^*)'$, where $k < N$. For each X_t^* Bagging is employed as a forecast method as $B^{-1} \sum_{b=1}^B \hat{y}_{t+h}^{(b)}$ where $\hat{y}_{t+h}^{(b)}$ is computed using a bootstrap sample $X_t^{*(b)}$ and $y_{t+h}^{(b)}$. This procedure is repeated for multiple subsets and the forecast average is calculated for each. The correlation is reduced because subsampling results in differently structured regression trees, which should lead to a stable forecast.

Boosting. Schapire (1990) introduces Boosting as an alternative solution to the overfitting problem. Assume that $\sum_{m=1}^M \theta_m \mathbf{1}_{[X_t \in R_m]}$ gives us a simple regression tree with initial value of $f_0(X_t) = \eta \sum_{m=1}^M \theta_m \mathbf{1}_{[X_t \in R_m]}$. $\eta \in (0,1)$ represents the learning rate set at $\eta = 0.1$. In Boosting the depth of regression trees should be shallow, which renders every base learner $f_s(X_t)$, for $s = 0, 1, \dots, S$ weak learners. In each stage the algorithm employs information of forecast errors from previous trees and searches for a new algorithm using an L_2 loss function based on this information. This produces model updates at the s -th stage using

$$f_s(X_t) = f_{s-1}(X_t) + \eta \sum_{m=1}^{M_s} \theta_{sm} \mathbf{1}_{[X_t \in R_{sm}]}, \quad (11)$$

where $\sum_{m=1}^{M_s} \theta_{sm} \mathbf{1}_{[X_t \in R_{sm}]}$ is estimated for the residual from $(s - 1)$ -th stage, $y_{t+h} - f_{s-1}(X_t)$. The model is updated until s reaches a set limit on boosting stages.

In summary, we presented the EML methods in this chapter. These methods are based on decision trees and aim to provide different solutions to the overfitting problem.

3.6. NEURAL NETWORKS

In this subchapter we present the Feedforward and LSTM neural networks we use. We begin with the former. LeCun et al. (2015) describes the Feedforward Neural Network (FFNN) in depth, along with a complex overview of neural network theory. The FFNN is a forecasting model with a single hidden layer written as

$$y_{t+h} = f(X_t) + \varepsilon_{t+h}, \quad (12)$$

where

$$f(X_t) = \sum_{j=1}^q \theta_j \sigma(w'_j X_t + b_j) + b. \quad (13)$$

Here, σ represents an activation function, q denotes the number of hidden units (neurons), and ε_{t+h} is the forecast error. We estimate the parameters (θ_j, w'_j, b_j, b) for $j = 1, \dots, q$ by minimizing the least square criterion.

For the FFNN the literature standard activation functions are the sigmoid function written as

$$\sigma(z) = \frac{1}{1+e^{-z}} \quad (14)$$

and the ReLU (rectified linear unit) function written as

$$\sigma(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{otherwise} \end{cases}, \quad (15)$$

where z represents the input of a hidden layer. These activation functions help introduce nonlinearity and allow the FFNN model to capture complex nonlinear relationships in the data. We test the performance of both and find they provide similar forecast results. In this study we specify the FFNN according to Masters (1993) and Gu et al. (2019). Their hidden layers range

from 1 to 5 with the number of a hidden unit q_ℓ in each layer ℓ following the geometric pyramid rule of Masters (1993). In conclusion, the FFNN can be an appropriate tool to analyze complex nonlinear relationships, but correctly specifying the model to meet the data is a huge computational cost which is the main reason the performance of this networks lacks behind the performance of other machine learning methods in our study.

To continue with, we present the LSTM network, which is a type of Recurrent Neural Network (RNN).

The forecasting model of a recurrent neural network with one hidden layer can be written as

$$y_{t+h} = \sum_{j=1}^q \theta_j h_{jt} + b + \varepsilon_{t+h} \quad (16)$$

where

$$h_{jt} = \sigma(w'_j X_t + \sum_{k=1}^q \theta_{jk} h_{kt-1} + b_j). \quad (17)$$

where $h_{1t}, h_{2t}, \dots, h_{qt}$ represent the q hidden units in the hidden layer at period t and σ represents the activation function used to produce the output of the hidden layer. The hidden layer of this network takes input from both the preceding layer and the same hidden layer of the previous period through an internal loop in (17) and can thus capture the serial dependence present in time series data. As Hochreiter and Schmidhuber (1997) state, this simple recurrent neural network is subject to the exploding/vanishing gradient problem and therefore we need to use a particular version of the RNN to capture long term dependencies.

Hochreiter and Schmidhuber (1997) propose the Long Short-Term Memory (LSTM) network, a variant of the RNN which is the most widely used in the macroeconomic forecasting literature. LSTM adds a special hidden unit, the memory cell, to the RNN structure. The memory cell captures long term information as it depends only on the memory cell from the previous period. In addition, the LSTM adds two special gating units to the memory cells. The input gate decides when to add new information while the forget gate decides when to drop the old information. In conclusion, this type of network is appropriate for time series data because of its desirable properties.

Due to our limited computational capacity, we do not specify the hyperparameters and activation functions through grid search but manually set the number of hidden units at 10

following Maehashi and Shintani (2020). For the activation functions we follow the literature standard and use the hyperbolic tangent activation function, except for the gates for which we use the sigmoid function.

In summary, we use the LSTM neural network for forecasting with 10 hidden units and two different activation functions. Given better computational capacity, our results could have been more accurate, as stated in chapter 5.

3.7. SUPPORT VECTOR MACHINE

Pratap and Sengupta (2019) use the Support Vector Machine to forecast Indian inflation and they provide a concise description of the method. Sometimes data are linearly non-separable, leading to complications in modeling through a line or hyperplane. What is more, the exact location of this boundary of separation might also be unknown. We can solve the first problem by projecting the data into different dimensions ensuring linear separability, provided the projection is selected suitably. The second issue is addressed by using a method to identify the optimal separation line.

Consider the following optimization problem to find $f(x)$ with the minimal norm value

$$\begin{aligned} f(x) &= \alpha + x'\beta \\ \min. J(\beta) &= \frac{1}{2}\beta\beta' \\ \text{subject to } \forall n: |y_n - (x'_n\beta + \alpha)| &\leq \varepsilon \end{aligned} \tag{18}$$

where x_n represents a set of N feature observations with y_n observed label. No function can satisfy all the constraints for every point. Due to this, we introduce slack variables τ_n and τ_n^* for every point in x_n . These variables allow the regression to satisfy the constraints up until their values. Because of this, we can reformulate the problem as

$$\begin{aligned} \min. J(\beta) &= \frac{1}{2}\beta\beta' + C \sum_{n=1}^N (\tau_n + \tau_n^*) \\ \text{subject to } \forall n: |y_n - (x'_n\beta + \alpha)| &\leq \varepsilon + \tau_n \\ \forall n: |(x'_n\beta + \alpha) - y_n| &\leq \varepsilon + \tau_n^* \\ \forall n: \tau_n &\geq 0; \tau_n^* \geq 0 \end{aligned} \tag{19}$$

where C is a positive value determining the deviations from the margin and smoothness of $f(x)$, while penalizing observations lying outside the margin. This algorithm is then extended to regression in the general case of non-linear separation boundary.

The SVM uses kernels to quantify the similarity of observations and to enlarge the feature space to accommodate a non-linear boundary between the observations. We can write

$$F(x) = \beta_0 + \sum_{i \in S} \alpha_i K(x, x_i), \quad (20)$$

where $F(\cdot)$ is the SVM and $K(\cdot)$ is the kernel. The model learns the parameter values from the data. In our work, we use the radial kernel of the form

$$K(x, x_i) = \exp \left(-\gamma \sum_{j=1}^p (x_{ij} - x_{i'j})^2 \right). \quad (21)$$

To sum up, in this subchapter we present the Support Vector Machine model which we use with a radial kernel. Now that we have presented all the models, we can show how we evaluate their forecasting performance.

3.8. PERFORMANCE EVALUATION

This subchapter presents the forecasting performance evaluation method. Firstly, we describe how we construct forecasts. We also explain the use of cross-validation, how we extract common factors and how we combine individual forecasts. Secondly, we concisely describe the evaluation metrics, namely the MSE, MAE and DA. To end with, we present the DM test of statistical significance.

To begin with, we forecast industrial production, measured by the index of industrial production, and inflation, measured by the harmonized index of consumer prices for 5 periods (1, 3, 6, 9 and 12 months ahead). The index of industrial production is a usually used proxy of GDP, as stated and used by Coulombe et al. (2022) and Maehashi and Shintani (2020). Coulombe et al. (2022) prefers this proxy for machine learning methods, but acknowledges that when using classical factor models, using bridge equations for quarterly GDP might be better. Our dataset begins with a period of high levels of economic uncertainty and volatility, namely the Great Recession. These observations right at the beginning of the dataset could distort the whole dataset. Because of this, we employ a rolling window scheme for forecasting to allow for the time-variance of the parameter estimates. Pratap and Sengupta (2019), Srinivasan and Mahajan (2019), Maehashi and Shintani (2020) and many others do the same for the same reason we state. To be more precise, we always change the initial point of estimation to keep the window size constant, always using 6 years of observations. Maehashi and Shintani (2020)

use a 10-year window despite having a substantially longer sample, so we consider our window choice adequate. In addition, changes in the window size for our estimation lead us to the same conclusions formulated in chapters 5 and 6.

To provide an actual example, take the case when the period is $t = R$. For this, we build the forecast $\hat{y}_{R+h}$ for the target variable y_{R+h} utilizing solely the information available up to $t = R$. We subsequently assess the forecast error $y_{R+h} - \hat{y}_{R+h}$. In the succeeding period $t = R + 1$, the model is reestimated with the data available up to $t = R + 1$ and a new value $\hat{y}_{R+h+1}$ is forecasted, while we drop the initial observation from the original sample. Hence, the complete sequence of hyperparameters, lag lengths and common factors is recomputed when new forecasts are formulated, even if there is no alteration in the model specification.

To add, given every machine learning model needs some form of parameter selection, we follow Coulombe et al. (2022) and aim to avoid the overfitting problem often present when working with large dataset and complex models by using k -fold cross-validation. Bergmeir (2018) establishes the statistical appropriateness of the method. In our case, $k = 5$, meaning that the in-sample set is randomly divided into five separate subsets, each representing, on average, 20% of the in-sample observations. For every one of the five subsets and for each hyperparameter set, four subsets are employed for estimation, while the remaining associated observations of the in-sample set serve as a test subset to calculate forecast errors, with the mean squared error serving as the evaluation metric. In summary, we employ a rolling window scheme to forecast our target variables and use cross-validation to do hyperparameter selection.

In addition, we extract the common factors using Principal Component Analysis (PCA) proposed by Stock and Watson (2002). The aim of using common factors is to capture the unobserved latent force driving the data generating process of a large number of variables and subsequently use a smaller set of factors as predictors for making forecasts. Using PCA we generate a potential predictor x_{it} as

$$x_{it} = \lambda_i' F_t^k + e_{it}, \quad (22)$$

where $F_t^k = (f_{1t}, f_{2t}, \dots, f_{kt})'$ is a $k \times 1$ vector of unobserved common factors, λ_i represents the factor loadings and e_{it} is the idiosyncratic error for $i = 1, \dots, N$ and $t = 1, \dots, T$.

Forecasting using common factors consists of two steps. First, we minimize $\sum_{i=1}^N \sum_{t=1}^T (x_{it} - \lambda_i' F_t^k)^2$ to compute F_t^k as the principal components of the predictors $\{x_{it}\}_{i=1}^N$. These principal components have unit variance and are orthogonal to each other. To select the

optimal number of factors for each forecast we use the Bai and Ng (2002) information criterion, given as

$$IC(k) = \ln V(k) + k \left(\frac{N+T}{NT} \right) \ln C_{NT}^2, \quad (23)$$

where $V(k) = \min_{\{\lambda_i, F_t^k\}} (NT)^{-1} \sum_{i=1}^N \sum_{t=1}^T (x_{it} - \lambda_i' F_t^k)^2$ and $C_{NT} = \min\{\sqrt{N}, \sqrt{T}\}$.

Second, once we have the factors, we construct the h -period ahead forecast using machine learning models. In summary, we use common factors derived by PCA as input variables to the machine learning models.

To finish the explanation of our forecasting setup, we describe the forecast combination methods. Bates and Granger (1969) state that combining individual forecasts can improve forecast accuracy and achieve better results than individual models. Clements and Hendry (2011) state that the combination of individual forecasting models can improve forecast accuracy, but they acknowledge that individual models can still be more accurate than forecast combinations. Bürgi (2015) argues that it is hard to achieve more accurate forecasts than by combining the individual models using equal weights. Because of this, he advises to combine the models by taking the arithmetic mean of the individual forecasts. In addition, Armstrong (2001) argues that a trimmed mean may also improve forecast accuracy. To calculate the trimmed mean, we exclude neural networks, as their performance is unsatisfactory due to limited computational capacity. We then calculate the arithmetic mean of the remaining forecasts. What is more, Araujo and Gaglianone (2023) use machine learning for forecast combination. They use adaLasso and Random Forest as forecast combination devices, meaning these models do not use the predictors but individual forecasts as inputs. They compare their results to the arithmetic mean and median and show that in their context combining individual forecasts using machine learning leads to improvement over the simple mean, which, as stated above, is hard. Because of this, we also employ every machine learning method as a forecast combination device and see whether they can produce more accurate forecast combinations than the timed mean excluding neural networks. In summary, in this study we combine individual forecasts using different weighting procedures.

To continue with, after explaining the forecasting procedure in detail, we describe the literature standard metrics to compare the forecast accuracy of the models. To compare the models, we relativize these metrics, meaning we divide the metric of the actual with by the

metric of the benchmark model. This way we can normalize the benchmark model to 1, which makes the comparison far easier. We use the mean absolute error (MAE), written as

$$\text{MAE} = \frac{\sum_{i=1}^n |y_i - \hat{x}_i|}{n} = \frac{\sum_{i=1}^n |e_i|}{n}. \quad (24)$$

which represents the absolute value of the difference between forecasted values and observed values. The next measurement is the mean squared error, written as

$$\text{MSE} = \frac{\sum_{i=1}^N (x_i - \hat{x}_i)^2}{N} \quad (25)$$

which gives us the quadratic average of the differences between the forecasted and observed values. As Medeiros et al. (2019) states, it is important to use both measures. The MSE is the literature standard for machine learning given most models minimize either this or the RMSE during training, while the MAE serves as a robustness check. If the results of these two metrics are aligned, we can be sure that our results are not due to only a few large forecast errors. To sum up, both MAE and MSE are important to gain overall insight into the forecasting performance of the models.

In addition to the classical forecasting performance measures, we use directional accuracy (DA) as well, which has been largely ignored by the forecasting literature on machine learning so far. We consider this a mistake, as even if errors are small, if the model cannot catch the correct direction of change in the variable, it essentially forecasts the wrong thing even if it does it with only a small error. Constantini et al. (2016) defines DA as

$$\text{DA}_{th} = I(\text{sgn}(y_t - y_{t-h}) = \text{sgn}(\hat{y}_t - y_{t-h})). \quad (26)$$

In summary, DA indicates whether the direction of change in the variable is correctly forecasted or not, irrespective of the magnitude of the forecast error.

To end with, we compare the MSE of each model to the benchmark. If the MSE of a particular model is lower, the model outperforms the benchmark. To check whether this outperformance is statistically significant, we use the Diebold-Mariano (DM) statistic written as

$$\text{DM} = \frac{\bar{d}}{\hat{\sigma}_d}, \quad (27)$$

where $\bar{d}$ is the average difference between the forecast errors of the compared models and $\hat{\sigma}_d$ is the Newey-West standard error of the difference.

To sum up, we first introduce the rolling window scheme used in our forecasting experiment and provide an example. We also describe how we derive common factors as inputs for the models and show how we use forecast combination. Next, we describe our accuracy measures, explain what we consider outperformance and show how we test if this outperformance is statistically significant or not using the DM statistic. In the next chapter we show how we apply the first part of these methods, namely data stationarization and preprocessing, to the database of the National Bank of Slovakia.

4. DATA

In this chapter we describe the data used in the forecasting experiments. First, we briefly describe the database, which is one of the reasons we focus on Slovakia. The National Bank of Slovakia (“NBS”) put together and maintains a novel, free-of-charge and excellent public macroeconomic database.² A public macroeconomic database of this dimensions is, to the authors’ knowledge, currently unparalleled in the V4 countries. The data is available in a clean, consistent and concise manner for multiple periods, formats and categories.

In addition to observed macroeconomic variables, the database contains so-called “soft” or indicators. They are called soft indicators because they are mainly survey based data measuring the expectations of economic participants and not real economic activity. These indicators should be used as input data when forecasting with machine learning models as advised by Coulombe et al. (2022). Compared to observed macroeconomic variables these indicators become available without a substantial time lag, allowing decisionmakers to begin policy discussions without having to wait until hard data becomes available. Because of their characteristics, we cannot expect forecasting models based on soft indicators to have lower forecast errors than models based on hard data. However, their foremost function is to give a hint about the direction to which the economy is heading at a particular time, meaning that we can use the model’s directional accuracy measures to test whether they fulfill their aim or not.

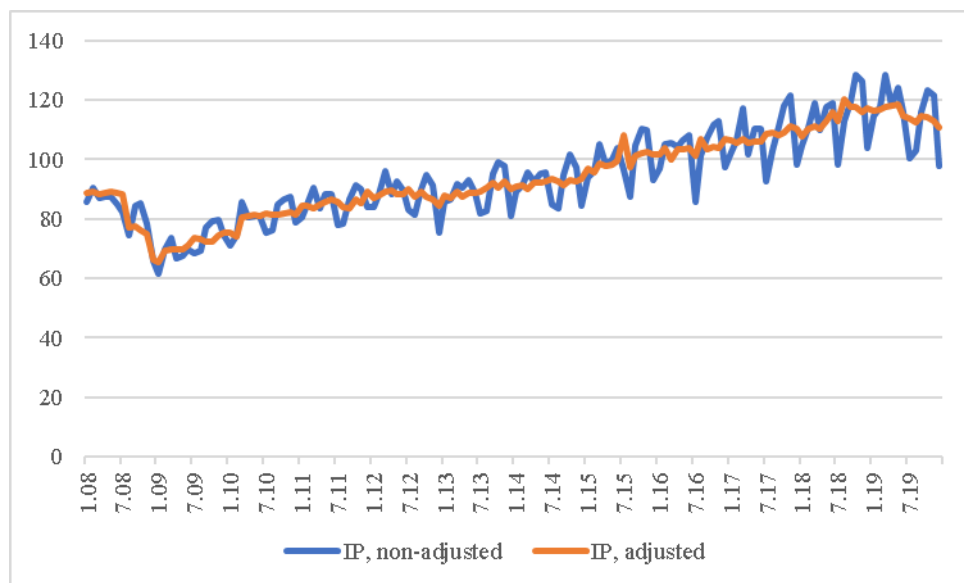
The only downside of the database is that it only contains actual data after revisions, meaning that older vintages before revisions are not available. This makes it impractical to compare the forecast performance of machine learning models to historical point forecasts made by, for example, central banks, because they had different data available at the time of making their forecast than we did. On the other hand, to the authors’ knowledge there is no country or organization that maintains a publicly available revisions database which includes enough data to reliably train machine learning models. Because of this, data driven models like the ones we use, namely ARIMA and Random Walk, are the literature standard for benchmarking, as seen in chapter 1.

Given the extensive database, we split our dataset into two and use machine learning to forecast our selected variables, namely industrial production (measured by the index of industrial production) and inflation (measured by the harmonized index of consumer prices) using not only real macroeconomic variables but also leading indicators. We use these target variables because they are the ones most often used in the machine learning forecasting

² <https://nbs.sk/statisticke-udaje/vybrane-makroekonomicke-ukazovatele/makroekonomicka-databaza/>

literature. For proof, please refer to the papers cited in chapter 1. In the case of soft indicators, we are mainly interested in the directional accuracy of the forecasts. To sum up, the database is appropriate for it contains enough data to reliably train the machine learning models and we use data driven models (Random Walk or ARIMA) as a benchmark for the reason stated above.

Next, we describe the period of observation. We use every times series from the database on a monthly frequency in a seasonally adjusted form when applicable for the period of 2008M01 – 2019M12. This way, we include a period of economic turmoil during the 2008 – 2009 crisis, but also a relatively long period of economic stability which followed the crisis. Unfortunately, we cannot split the data in two and compare the model performance during different economic conditions, because in that case the dataset for the crisis period would have been too short to meaningfully train a machine learning or any other model, thus it would have not been possible to derive any meaningful conclusions from the results. In conclusion, we proceed with using the whole dataset without splitting it into multiple periods and employ a rolling-window scheme, described in subchapter 3.8, to capture any potential time-variability in the regression parameters.



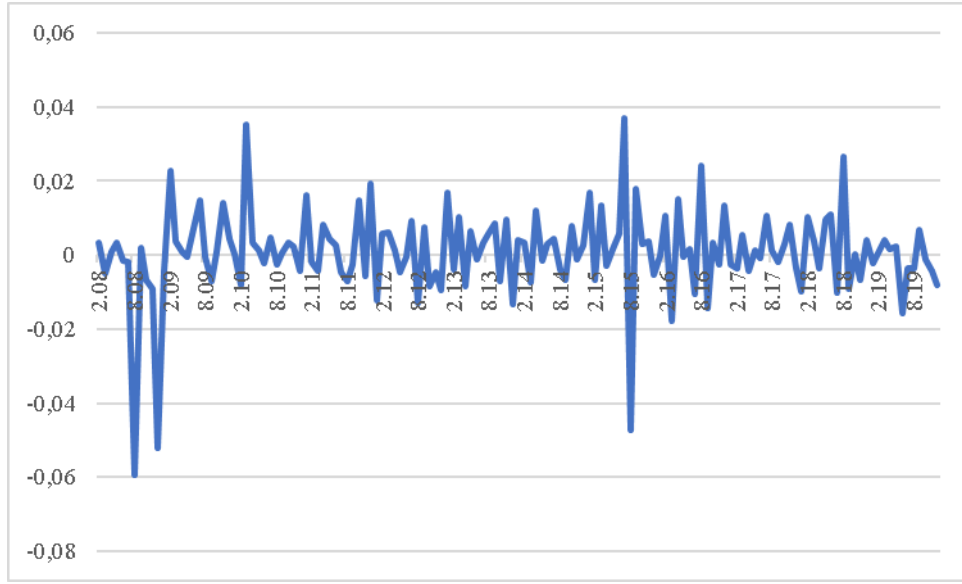
1. Chart: Seasonally adjusted and non-adjusted series
Source: NBS macroeconomic database

We include the period of the Great Recession instead of the COVID-19 pandemic for multiple reasons. First of all, if we included both, 1/3 of our sample would have represented a period of extremely high volatility and economic uncertainty, highly distorting our results. Because of this, we pick only one of the two crisis periods and we choose the Great Recession for multiple reasons. Firstly, it happened in the past and thus the data is not subject to revisions anymore. Secondly, its mid- and long-term effects are understood. Thirdly, it introduces high

levels of uncertainty right at the beginning of our sample, making it harder to train forecasting models. If the models can outperform even under these circumstances, it proves their quality. Fourthly, since the Great Recession happened right at the beginning of the sample, the rolling window scheme can leave out this data after some iterations, thus allowing the models to be trained on periods with relatively lower uncertainty, making our results even more robust. Fifthly, out of the possible nonlinear relationships stated in chapter 1, the housing collateral effect is present mainly during the period of the Great Recession, not COVID. Finally, the Great Recession is usually considered the largest economic crisis since the Great Depression and is thus considered a deeper crisis than COVID. Again, we want our models to earn their outperformance, we do not want to make it easy for them. In summary, we decided to include the period of Great Recession but not of COVID for the reasons stated in this paragraph.

To continue with, we describe the data preprocessing steps, namely stationarization and standardization. We also mention the extraction of common factors. Firstly, stationarity, as described in subchapter 3.2, is an important feature of a time series which we must ensure before the data enters the forecasting model. To ensure stationarity, macroeconomists usually remove the seasonal and trend component of the time series. Fortunately for us, the macroeconomic database contains both seasonally adjusted and unadjusted series, thus we can skip the seasonal adjustment part and just download the series which are seasonally adjusted already. The 1. Chart serves as an example of the effects of seasonal adjustment, illustrating the difference between the seasonally adjusted and non-adjusted series of industrial production. It is clear that the adjusted series is much smoother, making it more appropriate for forecasting. The 1. Chart also shows that the data trends upward in time. Because of this, before the series can enter the model, we have to remove the trend component as well. To remove the trend component, we take the first differences of the variables in percentages (e.g. unemployment rate, interest rates) and the soft indicators, for the former type it is the literature standard and the latter are on a scale between -100 and 100 and we know that we cannot take the log of non-positive numbers. These series become stationary after first differencing. Next, we take the logarithm of the rest of the variables, which serves to stabilize the variance. In our next step, we calculate the first differences or, in the case of 8 time series, the second differences of the time series. The 2. Chart presents the series of industrial production without the trend component. It is clear that the series revolves around a constant mean and has constant variance. There are, however, a couple of seemingly influential observations, which can lead to higher forecast errors, making the use of MAE advisable. At the end of this process each time series is

stationary, as measured by the ADF test, described in subchapter 3.2. Secondly, to finish our preprocessing, we mean-variance standardize the regressors as described in chapter 3.2.



2. Chart: Stationary series

Source: authors' calculations based on the NBS macroeconomic database

Thirdly, we extract common factors from the data. As stated among our aims in chapter 2, we want to test whether dimensionally reduced data can provide similar forecast errors and accuracy as the original time series from which the factors were extracted. We use PCA, described in the subchapter 3.8, and apply the rolling window scheme to extract common factors at each step of the forecast. After the factor extraction we use the factors as regressors for the individual machine learning models. In conclusion, we use a well-known but little tested approach to see if machine learning models can similarly forecast our selected variables from both the real time series and the dimensionally reduced information set.

To sum up, in this chapter we present the NBS macroeconomic database and describe the data we used for forecasting. Next, we describe the preprocessing steps and factor extraction procedure. After this preprocessing the data enters the model and we obtain our results, which we describe in the next chapter.

5. RESULTS

In this chapter we present the results of our forecasting experiments. We split the chapter into two subchapters and analyze industrial production in the first subchapter and inflation in the second. At first, we analyze forecasts based on the “hard” data. We provide the relative MSE and MAE values of each category and chart their evolution over different forecast horizons. Next, we analyze the statistical significance of the model forecasts and present the percentage improvement over the benchmark, including the forecast average for the particular category. We finish with analyzing directional accuracy and its evolution over time. To continue with, we also analyze the magnitude of forecast errors and directional accuracy of the forecasts based on common factors. Lastly, we forecast the target variable based on soft indicators and see whether these forecasts have higher directional accuracy than the benchmark. To sum up, in this chapter we see whether our research hypotheses can be rejected or not. Before we start, however, we must highlight a key point.

For this, we must refer to the papers cited in chapter 1. In these papers we see that it is not possible to find a one-size-fits-all model that is best in every scenario in every country. There is only one consensus, that machine learning models in general outperform the benchmark models, but it is impossible to generalize and select a particular model that is generally more appropriate across time periods and countries than the others. For example, Medeiros et al. (2019) finds that nonlinear methods, such as Random Forest, are considered best when applied to US data, but for India they are practically useless, as pointed out by Pratap and Sengupta (2019). Because of this, we must let the data speak for Slovakia as well, without considering any particular result for different countries. We start with industrial production.

5.1. INDUSTRIAL PRODUCTION

5.1.1. Forecasting industrial production using hard macroeconomic data

As stated above, we start our analysis by describing the relative MSE and MAE results of the forecasts of industrial production based on hard macroeconomic data. We begin by presenting the results of the Regularized Least Squares models in the 1. Table. The results of the benchmark model are normalized to 1.

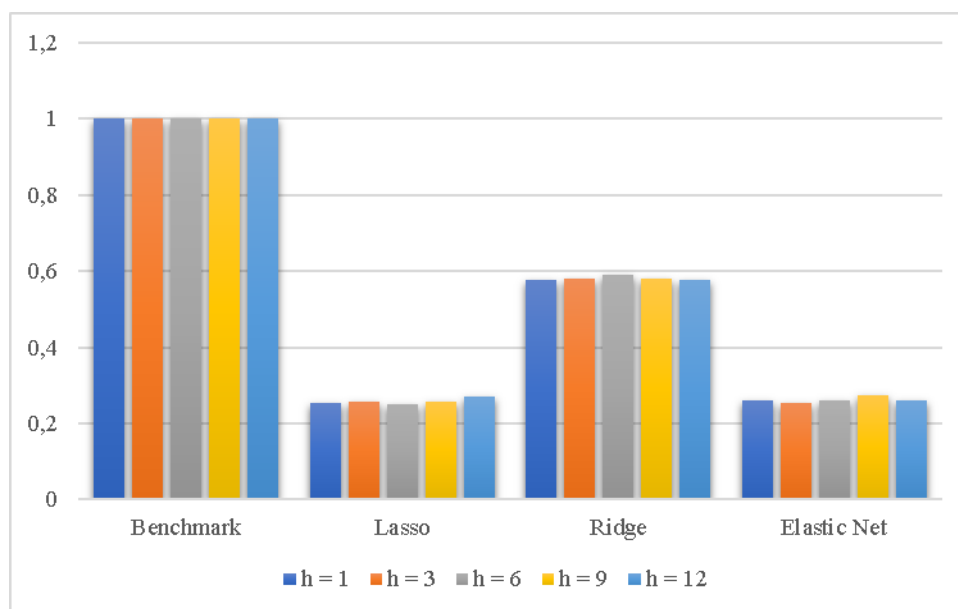
Firstly, when considering MSE, every RLS model outperforms the benchmark. Both Lasso and Elastic Net models have a relative MSE at around 0.25, meaning the Lasso component of the Elastic Net model has a higher weight compared to the Ridge component.

Based on subchapter 3.4 we see that in this case variable selection plays a more important role than coefficient shrinkage. The Ridge model also outperforms the benchmark by having a relative MSE at around 0.57 – 0.59 for every forecast horizon. This means that all RLS models outperform the benchmark model. Secondly, the relative MAE results support our previous conclusion as they are also smaller than their benchmark. Thanks to them we can be sure that the outperformance of the RLS models is not due to a few large forecast errors. In conclusion, the RLS models not only outperform the benchmark, but also do it consistently for every horizon.

1. Table: Relative MSE and MAE of RLS models

Relative MSE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Lasso</i>	0,25376	0,25854	0,25077	0,25699	0,26991
<i>Ridge</i>	0,5764	0,58032	0,59155	0,58178	0,57804
<i>Elastic Net</i>	0,25925	0,25221	0,262	0,27318	0,25893
Relative MAE					
<i>Benchmark</i>	1	1	1	1	1
<i>Lasso</i>	0,45058	0,44097	0,45073	0,44895	0,45683
<i>Ridge</i>	0,80952	0,80917	0,82727	0,82146	0,8108
<i>Elastic Net</i>	0,45879	0,441	0,46591	0,46079	0,44607

Source: authors' calculations



3. Chart: Relative MSE of RLS models over time

Source: authors' calculations

In addition, the magnitude of the outperformance stays consistent for every model and for every horizon, as shown in the 3. Chart.

Based on the evidence presented in the previous paragraphs, it is clear that the RLS models outperform the benchmark. To test whether this outperformance is statistically significant, we use the Diebold-Mariano test, described in subchapter 3.8. The test compares the errors of the benchmark model to the errors of the RLS methods. The 2. Table presents the DM – test p-values. If a cell has two stars next to the p-value, the forecast is statistically significant at the 5% level, while in the case of one star the forecast is significant at the 10% level. If there are no stars, the model cannot significantly improve the forecast compared to the benchmark. Most RLS model forecasts are significantly better at the 5% level and only two forecasts, namely the 9 and 12-month ahead forecasts of the Ridge model are significant at the 10% level. To sum up, the 2. Table clearly shows that the outperformance of the RLS models is statistically significant.

2. Table: DM – test p-values of RLS models

Diebold - Mariano test	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Lasso</i>	0,00639**	0,01358**	0,01289**	0,04393**	0,00794**
<i>Ridge</i>	0,00575**	0,03273**	0,04754**	0,06779*	0,05317*
<i>Elastic Net</i>	0,00664**	0,01321**	0,01174**	0,04661**	0,00935**

Source: authors' calculations

Since the outperformance is statistically significant, we can state the magnitude of outperformance in percentage terms. We also calculate the average of the individual RLS forecasts and the improvement over the benchmark in percentage terms. This is presented in the 3. Table. The 3. Table shows that the outperformance of the Lasso model is at around 74% for all h except for $h = 12$, when it is 73.01%. Ridge generally outperforms the benchmark by 40% - 42%, while Elastic Net is unsurprisingly very similar to Lasso, between 73.80% at $h = 6$ and 74,78% at $h = 3$. What is interesting, however, is that both the Lasso and Elastic Net models manage to achieve higher outperformance than the composite forecast of the RLS models. The composite forecast has on average at around 60% - 62% lower errors than the benchmark. In conclusion, the improvement in percentage terms is substantial regardless of which model we take into consideration.

3. Table: % improvement of RLS forecasts over the benchmark

% improvement	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Lasso</i>	74,62%	74,15%	74,92%	74,30%	73,01%
<i>Ridge</i>	42,36%	41,97%	40,84%	41,82%	42,20%
<i>Elastic Net</i>	74,07%	74,78%	73,80%	72,68%	74,11%
<i>RLS average</i>	61,06%	60,23%	60,84%	60,09%	62,08%

Source: authors' calculations

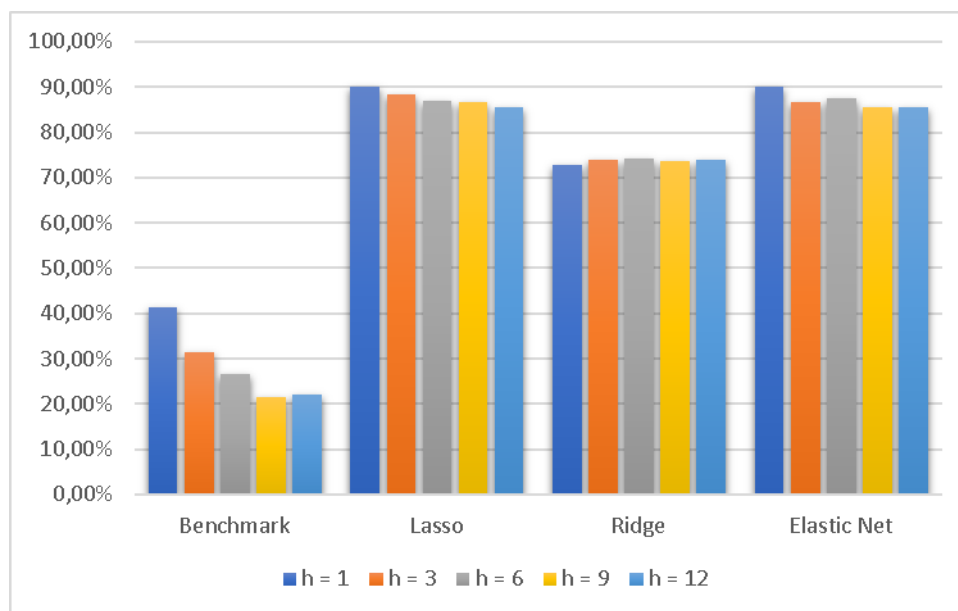
To sum up our findings regarding the errors of the RLS models, every model statistically significantly outperforms the benchmark, and there are only two instances where the outperformance is only significant at the 10% level. What is more, this outperformance is consistent over the different horizons and for different models. In addition, two individual models are capable of consistently outperforming even the RLS category average of individual forecasts.

Moving on to directional accuracy, every RLS model is more likely to capture the correct direction of change in the target variable than the benchmark model. The results are in the 4. Table. The DA of Lasso and Elastic Net are both between 85% and 90%, while the DA of Ridge is 72.86% - 74.10%, compared to the benchmark model, which has a DA between 21.51% and 41.43%, depending on the forecast horizon. In conclusion, the RLS models substantially improve DA with Lasso and Elastic Net introducing a more than twofold increase.

4. Table: Directional accuracy of RLS models

Directional Accuracy	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	41,43%	31,37%	26,67%	21,51%	22,18%
<i>Lasso</i>	90,00%	88,24%	86,92%	86,74%	85,59%
<i>Ridge</i>	72,86%	74,02%	74,10%	73,66%	74,01%
<i>Elastic Net</i>	90,00%	86,76%	87,44%	85,48%	85,45%

Source: authors' calculations



4. Chart: Directional accuracy of RLS models over time

Source: authors' calculations

What is more, this improvement is consistent over time, as shown in the 4. Chart. Another interesting thing stands out. As we increase the forecast horizon, the DA of the benchmark model decreases substantially, at a far faster pace than the slight decreases present in the case of Lasso and Elastic Net. This is because our benchmark model only incorporates short term information by design. The DA of the Ridge model is consistent, albeit at a lower level than that of Lasso or Elastic Net, but still far higher than that of the benchmark.

To sum up, every RLS forecast model has a substantially higher DA than the benchmark model. This DA is consistent for Ridge and only slightly decreases for Lasso and Elastic Net, while it falls dramatically for the benchmark as we increase h .

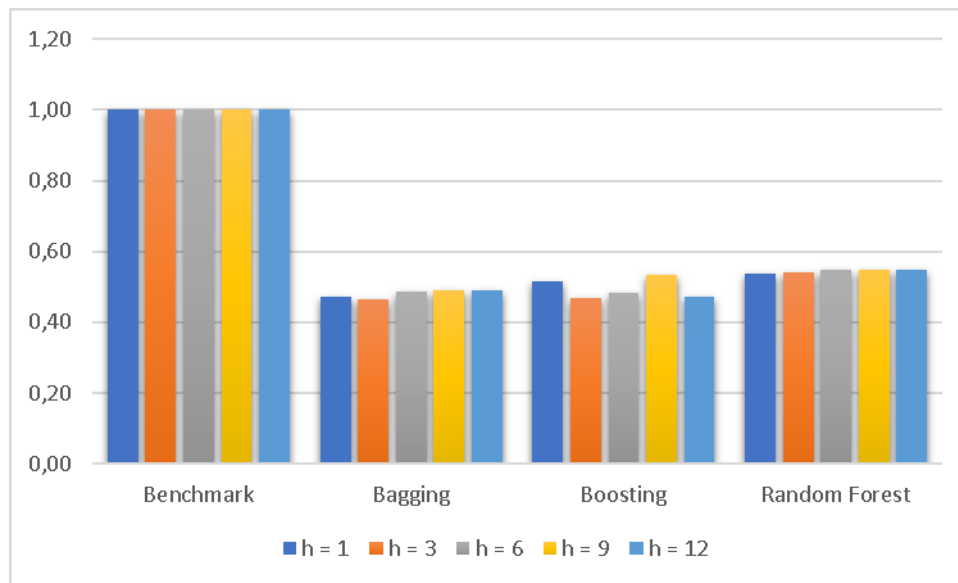
We continue with by presenting the results of the Ensemble Machine Learning (EML) models in the 5. Table. The results of the benchmark model are normalized to 1. Firstly, when considering MSE, every EML model outperforms the benchmark. Bagging has a MSE between 0.47 and 0.49, boosting between 0.467 and 0.533 and the Random Forest between 0.53 and 0.55. This means that Bagging performs best for every horizon except for $h = 12$, where Boosting has a slight advantage. Secondly, the relative MAE results support our previous conclusion as they are also smaller than their benchmark. Thanks to them we can be sure that the outperformance of the EML models is not due to a few large forecast errors. In conclusion, the EML models not only outperform the benchmark, but also do it consistently for every horizon.

5. Table: Relative MSE and MAE of EML models

Relative MSE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Bagging</i>	0,47081	0,46523	0,48503	0,49009	0,48902
<i>Boosting</i>	0,51369	0,46769	0,48275	0,53275	0,47255
<i>Random Forest</i>	0,53703	0,54068	0,54897	0,54848	0,54870
Relative MAE					
<i>Benchmark</i>	1	1	1	1	1
<i>Bagging</i>	0,60385	0,59082	0,61838	0,61948	0,60656
<i>Boosting</i>	0,62291	0,60607	0,63971	0,67181	0,60336
<i>Random Forest</i>	0,67807	0,67354	0,68552	0,66990	0,66560

Source: authors' calculations

In addition, the magnitude of the outperformance stays consistent for every model and for every horizon, as shown in the 5. Chart.



5. Chart: Relative MSE of EML models over time

Source: authors' calculations

Based on the evidence presented in the previous paragraphs, it is clear that the EML models outperform the benchmark. To test whether this outperformance is statistically significant, we use the Diebold-Mariano test, described in chapter 3. The test compares the errors of the benchmark model to the errors of EML models. The 6. Table presents the DM – test p-values. The Bagged model forecasts are significantly better at the 5% level at every horizon. In the case of Boosting, the forecast at $h = 1$ is significant at the 10% level, while the rest of the forecasts are significant at the 5% level. The same is true for the Random Forest model. To sum up, the 6. Table shows that the outperformance of the Bagged model is statistically significant at every horizon at the 5% level, while the outperformance of the rest of the models are statistically significant at every horizon and, with the exception of $h = 1$ at the 5% level.

6. Table: DM – test p-values of EML models

Diebold - Mariano test	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Bagging</i>	0,03742**	0,00904**	0,01138**	0,04915**	0,02405**
<i>Boosting</i>	0,05308*	0,02725**	0,01276**	0,02942**	0,02178**
<i>Random Forest</i>	0,06095*	0,04542**	0,02695**	0,02935**	0,03348**

Source: authors' calculations

Since the outperformance is statistically significant, we can state the magnitude of outperformance in percentage terms. We also calculate the average of the individual EML forecasts and the improvement over the benchmark in percentage terms. This is presented in the 7. Table. The 7. Table shows that the outperformance of the Bagged model is between 50% and 53.5%, which is still impressive, although less so than in the case of the best RLS models.

The outperformance of Boosting is between 46.72% and 53.23%, and that of the Random Forest is between 45% and 47%. Generally, Bagging is the best performing model except for $h = 12$, where Boosting has a slight advantage. The Random Forest model shows the highest consistency of the EML models. In contrast to the RLS specification, where two models produce smaller errors than the composite forecast, in this case only the Bagged model is able to outperform the composite forecast for all horizons, the latter outperforming the benchmark by 47% - 49.22%. Boosting, however, outperforms the composite forecast at $h = 3, 6$ and 12 . In conclusion, the improvement in percentage terms is substantial regardless of which model we take into consideration, but it is smaller when compared to the RLS methods.

7. Table: % improvement of EML forecasts over the benchmark

% improvement	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Bagging</i>	52,92%	53,48%	51,50%	50,99%	51,10%
<i>Boosting</i>	48,63%	53,23%	51,72%	46,72%	52,74%
<i>Random Forest</i>	46,30%	45,93%	45,10%	45,15%	45,13%
<i>EML average</i>	49,28%	50,88%	49,44%	47,62%	49,66%

Source: authors' calculations

To sum up our findings regarding the errors of the EML models, every model statistically significantly outperforms the benchmark. There are two instances where the outperformance is only significant at the 10% level. What is more, this outperformance is consistent over the different horizons and for different models. In addition, one individual model is capable of consistently outperforming even the model average, while Boosting outperforms it more often than not.

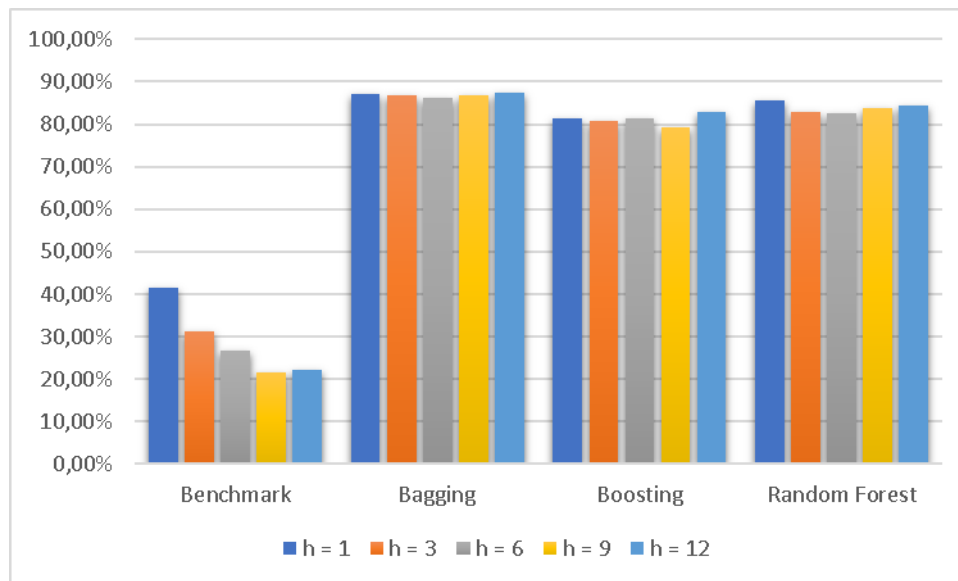
8. Table: Directional accuracy of EML models

Directional Accuracy	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	41,43%	31,37%	26,67%	21,51%	22,18%
<i>Bagging</i>	87,14%	86,76%	86,15%	86,74%	87,43%
<i>Boosting</i>	81,43%	80,88%	81,28%	79,39%	82,77%
<i>Random Forest</i>	85,51%	82,84%	82,56%	83,69%	84,46%

Source: authors' calculations

Moving on to directional accuracy, every EML model is more likely to capture the correct direction of change in the target variable than the benchmark model. The results are in the 8. Table. The DA of Bagging is between 86% and 88%, the DA of Boosting is 79.33% - 82.77%, while the DA of Random Forest is 82.56% - 85.51% compared to the benchmark model, which has a DA between 21.51% and 41.43%, depending on the forecast horizon. In conclusion, the EML models substantially improve DA.

What is more, this improvement is consistent over time, as shown in the 6. Chart.



6. Chart: Directional accuracy of EML models over time

Source: authors' calculations

To sum up, every EML forecast model has a substantially higher DA than the benchmark model. This DA is consistent for every EML model, while it falls dramatically for the benchmark as we increase h .

In the 9. Table we present the results of the remaining models, namely the Support Vector Machine, the Feedforward Neural Network and the Long-Short Term Memory Neural Network. Because these models do not deliver as compelling results as the previous models, we summarize their performance in one table. To begin with, the relative MSE of these models is either closer to that of the benchmark in case of the SVM model, or it is higher in the case of the NN models. In the case of both NN models we see substantial underperformance, which, however, can be explained. When estimating the other models, we were able to optimize hyperparameter search using R, but in the case of the networks we could not, for we did not have the necessary computational capacity. It is clear from the other results as well. The relative MAE further proves the underperformance of the NN models, and we can see that neither of them can produce statistically significantly better forecasts when compared to the benchmark model. Their DA values are, however, slightly higher than those of the benchmark, but given the magnitude of the errors and the fact that other models produce even higher DA values we recommend not relying on these NN models for the time being. Once we will have access to appropriate computational capacity, however, we will dive deeper into the exploration of their performance. Until then, we do not give them further attention in this study. To continue with, the SVM model, as previously stated, manages to outperform the benchmark. This outperformance is, however, smaller than in the case of the other models. In addition, it is

statistically significant at the 5% level at $h = 6$ and $h = 12$, and at the 10% level at $h = 3$, but it is insignificant at $h = 1$ and $h = 9$. The DA values of the model are between 65.71% and 71.89%, which are lower than the rest of the models, despite showing an increase in accuracy with the increase of the forecast horizon. In conclusion, we cannot recommend using NN models, but not because they are inherently unfit for the Slovakian economy, but because our computational limitations did not allow for exploring all their possibilities. Regarding the SVM model, it can be used, but we recommend checking the results and using other models, such as RLS or EML models instead, when forecasting Slovakian industrial production.

9. Table: IP forecasts with SVM, FFNN and LSTM

Relative MSE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Support Vector Machine</i>	0,77375	0,85289	0,75612	0,74687	0,75215
<i>Feedforward NN</i>	7,42215	7,02437	6,87417	6,60251	6,42844
<i>LSTM NN</i>	128,124	18,7122	7,54591	4,03824	3,68138
Relative MAE					
<i>Support Vector Machine</i>	0,92173	0,9459	0,90444	0,89306	0,90576
<i>Feedforward NN</i>	3,71541	3,56651	3,56545	3,48494	3,40986
<i>LSTM NN</i>	13,2955	4,81009	3,17909	2,20504	2,24366
Diebold – Mariano test					
<i>Support Vector Machine</i>	0,20681	0,05121	0,03532	0,13286	0,00035
<i>Feedforward NN</i>	1	1	1	1	1
<i>LSTM NN</i>	1	1	0,99996	0,99192	0,99998
Directional accuracy					
<i>Benchmark</i>	41,43%	31,37%	26,67%	21,51%	22,18%
<i>Support Vector Machine</i>	65,71%	67,65%	69,23%	71,33%	71,89%
<i>Feedforward NN</i>	54,29%	55,39%	56,15%	57,17%	57,77%
<i>LSTM NN</i>	45,71%	48,04%	48,97%	53,41%	54,38%

Source: authors' calculations

To sum up, in this subchapter we analyzed the performance of all the machine learning models mentioned in chapter 3 for forecasting industrial production based on hard data. Most of the machine learning models outperform the benchmark model by a huge margin, except for the neural network models, where we had computational limitations. Firstly, the best performing models are Lasso and Elastic Net, which are basically the same in this context. These models statistically significantly outperform the benchmark model at every forecasting horizon at the 5% level. What is more, these individual models are more accurate than the category average. In addition, the Lasso and Elastic Net models have the highest directional accuracy values out of all models and the results are consistent over time. The performance of

the Ridge model, however, is somewhat weaker. It is still far better than the benchmark, but it performs worst out of all models, except for the SVM. Secondly, the EML models have mildly higher errors and lower directional accuracy values than Elastic Net and Lasso, but they are capable of statistically significantly outperforming the benchmark model. This is especially true for the Bagged model, which performs the best out of the three and statistically significantly outperforms the benchmark model at the 5% level at every horizon. The other models can outperform at the 5% horizon as well, except for $h = 1$. Thirdly, the SVM model delivers mixed results and we recommend using the models from the previous two categories, especially Lasso and Elastic Net for regularization and Bagging for capturing nonlinear relationships in the data. To continue with, we explore the forecasting performance of the machine learning models based on common factors in the next subchapter.

5.1.2. Forecasting industrial production using common factors

Shintani (2005) and Maehashi and Shintani (2020) apply machine learning models to forecasting based on common factors extracted from the available dataset. In this subchapter we see if this approach outperforms the benchmark model and if yes, how does it compare to the forecasting performance based on hard macroeconomic data without dimensional reduction.

To begin with, the 10. Table presents the relative MSE values of the forecasts based on common factors. The benchmark MSE is normalized to 1.

10. Table: Relative MSE values of IP forecasts based on common factors

Relative MSE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Lasso</i>	0,701524	0,700192	0,681737	0,852603	0,698305
<i>Ridge</i>	0,689632	0,705956	0,641994	0,640785	0,687538
<i>Elastic Net</i>	0,715571	0,832926	0,705255	0,667444	0,670397
<i>Bagging</i>	0,873563	0,831503	0,840161	0,812082	0,813511
<i>Boosting</i>	0,729903	0,681461	0,676986	0,663042	0,688355
<i>Random Forest</i>	0,837169	0,873106	0,832147	0,831806	0,825432
<i>Support Vector Machine</i>	0,739667	0,714448	0,720253	0,712005	0,710229

Source: authors' calculations

The 10. Table shows that forecasts based on common factors still outperform the benchmark model. There is no relative MSE that is higher than one. Firstly, considering RLS methods, Lasso and Elastic Net deliver results similar to each other. In addition, the results of Ridge model are also comparable to the previous two, despite it being the worst performing model of the three in the case of forecasting using hard macroeconomic data. Regarding ELS methods,

Boosting outperforms both Bagging and Random Forest, which deliver fairly similar results. The results of the SVM model are almost exactly the same as in the previous subchapter.

To continue with, we show the percentage difference between the relative MSE values of the forecasts based on hard macroeconomic data and common factors in the 11. Table. Positive values mean that the model based on hard data is more accurate than the model based on common factors. Negative values mean the exact opposite. Firstly, the 11. Table shows that in most cases, the models based on hard data are more accurate. In addition, their forecasts are more consistent over multiple horizons. As an example, the 1. Table makes it clear that the Lasso forecasts are fairly consistent. In the 11. Table, however, we see that the magnitude of underperformance of forecasts based on common factors for the Lasso model fluctuates between 158% and 232%, meaning that these forecasts are less consistent over multiple horizons than forecasts based on hard macroeconomic data. The same can be said, for example, of the Ridge model. Secondly, the SVM model has lower errors when forecasting based on factors. The differences are, however, small between 4% and 5.5%, except for $h = 3$ at 16.23%. In conclusion, forecasts based on common factors outperform our benchmark, but underperform forecasting based on hard macroeconomic data. The sole exception is the SVM model, but it is generally the worst performing of all models, thus we advise to focus on forecasting based on hard macroeconomic data. These results are similar to those of Shintani (2005) and Maehashi and Shintani (2020).

11. Table: % difference between forecasts based on factors and hard data

% difference	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Lasso</i>	176,45%	170,83%	171,85%	231,76%	158,72%
<i>Ridge</i>	19,64%	21,65%	8,53%	10,14%	18,94%
<i>Elastic Net</i>	176,01%	230,24%	169,18%	144,32%	158,91%
<i>Bagging</i>	85,55%	78,73%	73,22%	65,70%	66,35%
<i>Boosting</i>	42,09%	45,71%	40,23%	24,46%	45,67%
<i>Random Forest</i>	55,89%	61,48%	51,58%	51,66%	50,44%
<i>Support Vector Machine</i>	-4,40%	-16,23%	-4,74%	-4,67%	-5,57%

Source: authors' calculations

To end with, we show the directional accuracy of forecasts based on common factors in the 12. Table. The 12. Table further supports our findings. Directional accuracy values based on common factors are far higher than the benchmark, but they are lower compared to the results based on hard macroeconomic data. Their values are, however, at least stable over multiple horizons.

In summary, we presented the forecasting results of machine learning models based on common factors in this subchapter. Firstly, forecasts based on common factors outperform the

benchmark. The outperformance is true for both the relative MSE values and DA values. This outperformance is, however, smaller than in the case of forecasts based on hard macroeconomic data. The SVM model is the only exception. Even though they outperform, we cannot consider common factors an appropriate substitute to hard macroeconomic data, given the forecast errors of the models based on them are higher compared to the results in subchapter 5.1.1. The underperformance of common factor based forecasts compared to forecasts based on hard data imply that in Slovakia, we should begin to explore the empirical possibility of slowly moving away from factor based dimensional reduction techniques to forecasting based on regularization using hard data, as both aim to capture all the important information while reducing the number of predictors, and regularization based on hard data seems more accurate in our sample.

12. Table: Directional accuracy of IP forecasts based on common factors

Directional accuracy	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	41,43%	31,37%	26,67%	21,51%	22,18%
<i>Lasso</i>	65,71%	68,63%	67,95%	68,64%	67,66%
<i>Ridge</i>	65,71%	66,67%	68,21%	68,82%	69,07%
<i>Elastic Net</i>	64,29%	67,65%	68,46%	68,10%	68,64%
<i>Bagging</i>	68,57%	68,14%	69,74%	71,51%	72,74%
<i>Boosting</i>	62,86%	69,12%	66,67%	67,03%	67,66%
<i>Random Forest</i>	67,14%	64,22%	67,95%	69,71%	69,77%
<i>Support Vector Machine</i>	67,14%	65,69%	67,18%	68,46%	70,06%

Source: authors' calculations

5.1.3. Forecasting industrial production based on soft indicators

As stated in chapter 4, we cannot expect the soft indicators to produce forecasts with lower errors or higher accuracy than those based on hard data. But, as stated in the same chapter, their primary purpose is not to capture the exact magnitude of change in macroeconomic variables, although their extreme values might signal larger changes ahead. Their principal aim is to capture the sentiments of economic participants about the general direction of the economy and its components, thus providing an approximate overview of where the economy is headed. Their main advantage is their availability, meaning there is no substantial lag as in the case of macroeconomic variables where decision makers have to wait weeks or sometimes even months to obtain data for key decisions. In summary, because of their design and availability, they can be a useful tool for forecasting the direction of the economy.

Because of this, we are interested in the possibility of combining input data from more than 250 indicators and use machine learning models for forecasting. We aim to prove that

combining the indicators this way can in fact lead to a higher probability of forecasting the direction of the change of our target variable, which is industrial production in this case. We do the forecasting exercise and report the relative MSE values in the 13. Table.

13. Table: Relative MSE of IP forecasts based on indicators

Relative MSE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Lasso</i>	1,32312	1,34674	1,39162	1,51222	1,48444
<i>Ridge</i>	1,12521	1,1437	1,22579	1,35124	1,37424
<i>Elastic Net</i>	1,29585	1,16898	1,24093	1,58505	1,50255
<i>Bagging</i>	1,31483	1,38907	1,46642	1,61058	1,63486
<i>Boosting</i>	1,28823	1,24934	1,38982	1,48275	1,52916
<i>Random Forest</i>	1,27363	1,29351	1,36195	1,52735	1,51519
<i>Support Vector Machine</i>	1,21398	1,22918	1,31444	1,42896	1,43386

Source: authors' calculations

As expected, the 13. Table shows that indicators are not the most appropriate source of information when we are interested in the magnitude of changes in the macroeconomic variables. The benchmark model generally does a better job of forecasting industrial production than the machine learning models based on indicators. None of these models have relative MSE values under 1. To sum up, based on our results, hard data should be used if we want to minimize the relative MSE of the forecasts or, if a large dataset is not available, the benchmark data driven model is more reliable than indicators. Our expectations are, therefore, met.

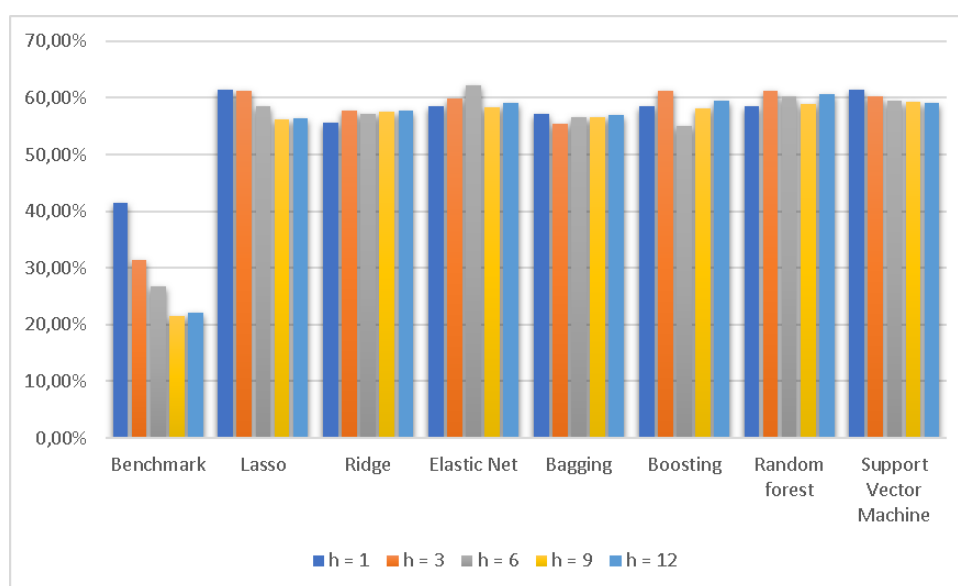
14. Table: Directional accuracy of IP forecasts based on indicators

Directional accuracy	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	41,43%	31,37%	26,67%	21,51%	22,18%
<i>Lasso</i>	61,43%	61,27%	58,46%	56,27%	56,50%
<i>Ridge</i>	55,71%	57,84%	57,18%	57,53%	57,77%
<i>Elastic Net</i>	58,57%	59,80%	62,31%	58,42%	59,18%
<i>Bagging</i>	57,14%	55,39%	56,67%	56,63%	56,92%
<i>Boosting</i>	58,57%	61,27%	55,13%	58,24%	59,46%
<i>Random Forest</i>	58,57%	61,27%	60,26%	58,96%	60,59%
<i>Support Vector Machine</i>	61,43%	60,29%	59,49%	59,32%	59,04%

Source: authors' calculations

To continue with, we analyze directional accuracy, our main measure of interest in the case of indicator data. The DA values are in the 14. Table. A couple of interesting things stand out. Firstly, every machine learning model outperforms the benchmark model by a large margin at every horizon, validating our assumptions. Secondly, the outperformance is smaller than in the case of using hard macroeconomic data. This is, however, expected. Forecasts based on

indicator data have lower DA values than forecasts based on hard macroeconomic data, but this is the price of speed. As stated above, indicators are available earlier and they measure sentiments so we cannot directly compare the DA values in the 14. Table to those in the previous subchapters. We must compare them to the benchmark, which they beat comfortably. Thirdly, this outperformance is also consistent, as presented on the 7. Chart. Fourthly, there are no big differences between models, meaning that we can select the model which requires the least computational capacity (in our case Lasso) and use it for computing directional accuracy. In conclusion, indicators serve their purpose. By using an extensive dataset of soft indicators and machine learning models, we can predict the direction of change in industrial production with a higher likelihood than if we only used the benchmark model.



7. Chart: DA of IP forecast based on indicators over time
Source: authors' calculations

In summary, forecasts based on indicator data produce higher errors but simultaneously have a higher likelihood of correctly predicting the direction of change in a variable than the benchmark model, as expected.

To sum up subchapter 5.1, we use machine learning models for forecasting industrial production. Firstly, both RSE and EML models statistically significantly outperform the benchmark model and has higher directional accuracy values. Most of the outperformance is significant at the 5% level, except for four cases out of 35, which are significant only at the 10% level. This shows how regularization and non-linear relationships play a crucial role in the data generating process of the Slovak economy. In addition, both categories have individual models that outperform the category average. On the other hand, neural networks cannot produce reliable results due to computational limitations and the performance of the Support

Vector Machine model is mixed. Secondly, forecasts based on common factors outperform their benchmarks both in terms of MSE and DA values, but they underperform forecasts based on hard data. Based on this it seems that regularization might be a more appropriate dimensional reduction tool than principal component factor extraction. Thirdly, indicator data produces higher DA values than the benchmark, as expected. Based on these facts, we accept the hypotheses formulated in chapter 3 for the forecasts of industrial production. In the next chapter we analyze our results for the inflation forecasts.

5.2. INFLATION

5.2.1. Forecasting inflation based on hard macroeconomic data

To continue our analysis, we present the relative MSE and MAE results of the forecasts of inflation in the 15. Table. The results of the benchmark model are normalized to 1.

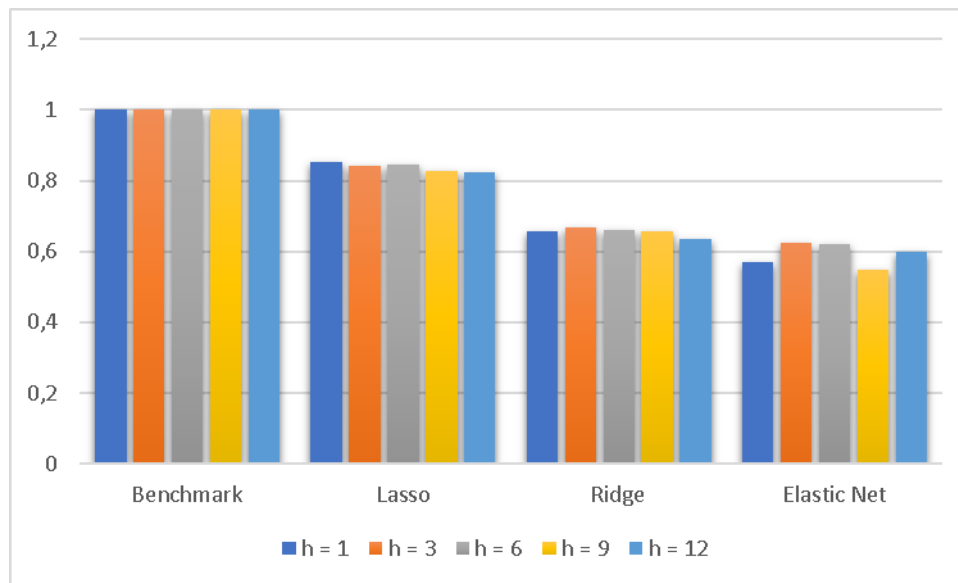
15. Table: Relative MSE and MAE of RLS models - inf

Relative MSE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Lasso</i>	0,85183	0,84309	0,84579	0,82543	0,82269
<i>Ridge</i>	0,65684	0,66673	0,65922	0,65645	0,63384
<i>Elastic Net</i>	0,57095	0,62576	0,62040	0,54723	0,59771
Relative MAE					
<i>Benchmark</i>	1	1	1	1	1
<i>Lasso</i>	0,87313	0,86684	0,86351	0,86148	0,86148
<i>Ridge</i>	0,7981	0,81195	0,80224	0,79797	0,79797
<i>Elastic Net</i>	0,7461	0,75678	0,76212	0,71168	0,71168

Source: authors' calculations

Firstly, when considering MSE, every RLS model outperforms the benchmark. The Lasso model has a relative MSE between 0.86 and 0.82. The Ridge model also outperforms the benchmark by having a relative MSE at around 0.63 – 0.65 for every forecast horizon. The best performing of the three is the Elastic Net model with relative MSE between 0.57 and 0.63. Since the Ridge model has a lower relative MSE than Lasso, the Elastic Net model incorporates more information from the former, showing that in this case, shrinkage is preferred to variable selection. In conclusion, all RLS models outperform the benchmark model. Secondly, the relative MAE results support our previous conclusion as they are also smaller than their benchmark. Thanks to them we can be sure that the outperformance of the RLS models is not due to a few large forecast errors. In conclusion, the RLS models not only outperform the benchmark, but also do it consistently for every horizon.

In addition, the magnitude of the outperformance stays consistent for every model and for every horizon, as shown in the 8. Chart.



8. Chart: Relative MSE of RLS models over time - inf

Source: authors' calculations

Based on the evidence presented in the previous paragraphs, it is clear that the RLS models outperform the benchmark. To test whether this outperformance is statistically significant, we use the Diebold-Mariano test, described in chapter 3. The test compares the errors of the benchmark model to the errors of RLS. The 16. Table presents the DM – test p-values. The 16. Table clearly shows that every RLS model forecast is significantly better at the 5% level.

16. Table: DM – test p-values of RLS models - inf

Diebold - Mariano test	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Lasso</i>	0,00622**	0,00121**	0,00415**	0,00525**	0,00168**
<i>Ridge</i>	0,01335**	0,01527**	0,01263**	0,01498**	0,03328**
<i>Elastic Net</i>	0,00197**	0,00430**	0,00565**	0,00775**	0,02192**

Source: authors' calculations

Since the outperformance is statistically significant, we can state the magnitude of outperformance in percentage terms. We also calculate the average of the individual RLS forecasts and the improvement over the benchmark in percentage terms. This is presented in the 17. Table. The 17. Table shows that the outperformance of the Lasso model is between 14% and 18%. It increases as h increases. Ridge generally outperforms the benchmark by 33% - 36.63%, while Elastic Net produces the highest outperformance, namely between 37.42% at $h = 3$ and 45,28% at $h = 9$. What is also interesting is that the Elastic Net model outperforms the composite forecast at every horizon while Ridge manages to do it from $h = 6$ to $h = 12$. The

composite forecast has on average at around 29.99% - 35.26% lower errors than the benchmark. In conclusion, the improvement in percentage terms is substantial regardless of which model we take into consideration.

17. Table: % improvement of RLS forecasts over the benchmark - inf

% improvement	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Lasso</i>	14,82%	15,69%	15,42%	17,46%	17,73%
<i>Ridge</i>	34,32%	33,33%	34,08%	34,35%	36,62%
<i>Elastic Net</i>	42,91%	37,42%	37,96%	45,28%	40,23%
<i>RLS average</i>	35,26%	33,40%	31,63%	29,99%	33,64%

Source: authors' calculations

To sum up our findings regarding the errors of the RLS models, every model statistically significantly outperforms the benchmark at the 5% level. What is more, this outperformance is consistent over the different horizons and for different models. In addition, two individual models are capable of outperforming even the model average, albeit the Ridge model only does this on the three longest horizons.

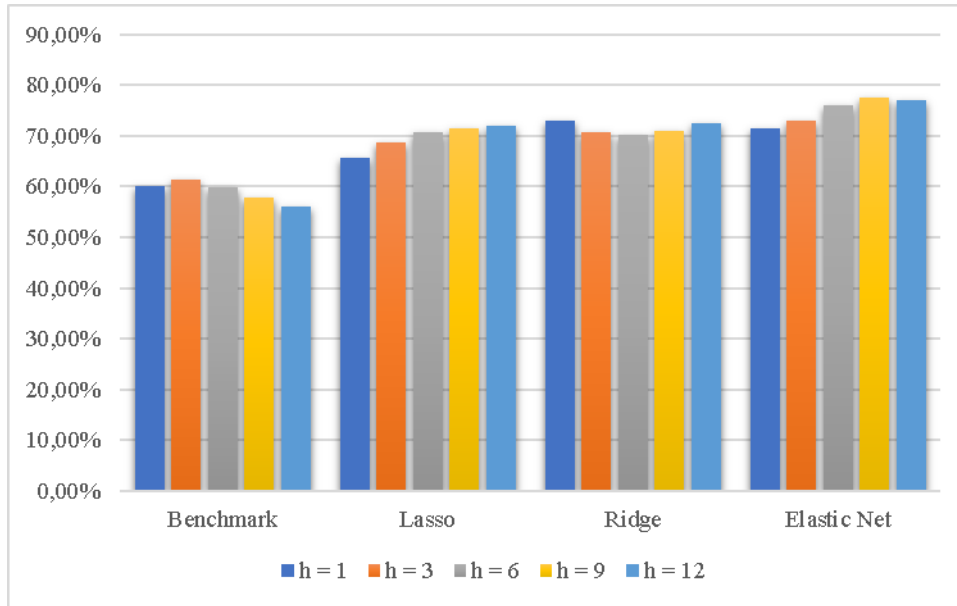
Moving on to directional accuracy, every RLS model is more likely to capture the correct direction of change in the target variable than the benchmark model. The results are in the 18. Table. The DA of Lasso is between 65% and 72%, while the DA of Ridge is 70.26% - 72.86%, compared to the benchmark model, which has a DA between 56.07% and 61.27%, depending on the forecast horizon. In conclusion, the RLS models improve DA.

18. Table: Directional accuracy of RLS models - inf

Directional Accuracy	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	60,00%	61,27%	59,74%	57,89%	56,07%
<i>Lasso</i>	65,71%	68,63%	70,77%	71,33%	71,89%
<i>Ridge</i>	72,86%	70,59%	70,26%	70,97%	72,46%
<i>Elastic Net</i>	71,43%	73,04%	75,90%	77,60%	76,98%

Source: authors' calculations

What is more, this improvement is consistent over time, as shown in the 9. Chart. The DA values of Lasso and Elastic Net slightly grow as we increase h . The DA values of the Ridge model do not show a growing pattern. Another interesting thing stands out. As we increase the forecast horizon, the DA of the benchmark model decreases only slightly. To sum up, every RLS forecast model has a higher DA value than the benchmark model. This DA is consistent for Ridge and slightly increases for Lasso and Elastic Net, while it slightly falls for the benchmark as we increase h .



9. Chart: Directional accuracy of RLS models over time - inf
Source: authors' calculations

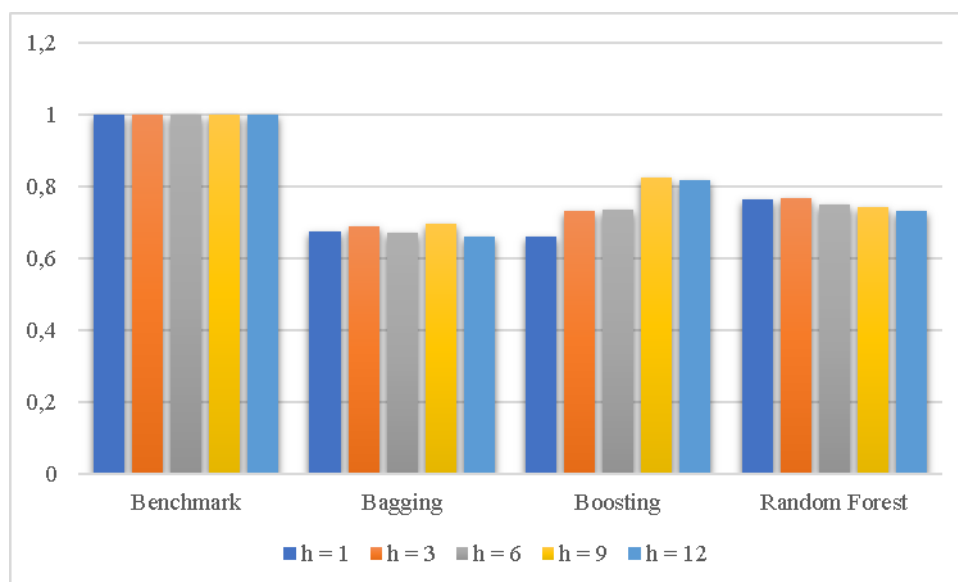
19. Table: Relative MSE and MAE of EML models - inf

Relative MSE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Bagging</i>	0,67586	0,69034	0,67162	0,67980	0,65210
<i>Boosting</i>	0,66167	0,73106	0,73522	0,82561	0,81767
<i>Random Forest</i>	0,76327	0,76775	0,74941	0,74185	0,73045
Relative MAE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Bagging</i>	0,80201	0,80335	0,79988	0,81326	0,81326
<i>Boosting</i>	0,7671	0,83876	0,82218	0,87872	0,82839
<i>Random Forest</i>	0,8565	0,83859	0,82778	0,82819	0,80148

Source: authors' calculations

We continue with by presenting the results of the Ensemble Machine Learning (EML) models in the 19. Table. The results of the benchmark model are normalized to 1. Firstly, when considering MSE, every EML model outperforms the benchmark. Bagging has an MSE between 0.67 and 0.70, Boosting between 0.66 and 0.83 and the Random Forest between 0.73 and 0.77. Again, the best performing model is Bagging, except for $h = 1$, where Boosting has a slightly lower forecast error. Secondly, the relative MAE results support our previous conclusion as they are also smaller than their benchmark. Thanks to them we can be sure that the outperformance of the EML models is not due to a few large forecast errors. In conclusion, the EML models not only outperform the benchmark, but also do it consistently for every horizon.

In addition, the magnitude of the outperformance stays consistent for Bagging as shown in the 10. Chart. The 10. Chart also shows an increase in forecast errors in Boosting while a slight decrease for Random Forest as we increase h . Because of this, we advise to use Bagging, as it is the best overall model and the outperformance is consistent over time. Boosting, however, might be better if we are only interested in the one-period ahead forecast.



10. Chart: Relative MSE of EML models over time - inf

Source: authors' calculations

20. Table: DM – test p-values of EML models - inf

Diebold - Mariano test	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Bagging</i>	0,00318**	0,00246**	0,00215**	0,00228**	0,00701**
<i>Boosting</i>	0,00155**	0,01021**	0,01181**	0,04125**	0,01602**
<i>Random Forest</i>	0,01956**	0,02505**	0,01219**	0,02601**	0,04646**

Source: authors' calculations

Based on the evidence presented in the previous paragraphs, it is clear that the EML models outperform the benchmark. To test whether this outperformance is statistically significant, we use the Diebold-Mariano test, described in subchapter 3.8. The test compares the errors of the benchmark model to the errors of EML. The 20. Table presents the DM – test p-values. Every single EML model statistically significantly outperforms the benchmark at the 5% level at every horizon. Since Bagging has the lowest forecast errors at every horizon except $h = 1$, we consider it the best model.

Since the outperformance is statistically significant, we can state the magnitude of outperformance in percentage terms. We also calculate the average of the individual EML forecasts and the improvement over the benchmark in percentage terms. This is presented in the 21. Table. The 21. Table shows that the outperformance of the Bagged model is between

30.2% and 34.01%, which is still impressive, given that it is higher than that of Lasso and is approximately at the same level as Ridge. Only the Elastic Net model outperforms Bagging in terms of % improvement. The outperformance of Boosting is between 17.44% and 33.83%, and that of the Random Forest is between 23.00% and 27.00%. The latter two models show different tendencies. The outperformance of Boosting is strongest at $h = 1$, but it rapidly falls as we increase h , while the outperformance of the Random Forest slightly increases. What is more, the Bagged model manages to outperform the composite forecast. The composite forecast outperforms the benchmark by 29.40% - 34.33%. In conclusion, the models outperform the benchmark substantially, although less than in the case of industrial production. The outperformance is statistically significant at the 5% level at every horizon and for every model. Compared to the RLS specification, the Bagged model manages to outperform Lasso and is at the same level as Ridge, falling behind only the Elastic Net model.

21. Table: % improvement of EML forecasts over the benchmark - inf

% improvement	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Bagging</i>	32,41%	30,97%	32,84%	32,02%	34,79%
<i>Boosting</i>	33,83%	26,89%	26,48%	17,44%	18,23%
<i>Random Forest</i>	23,67%	23,22%	25,06%	25,81%	26,96%
<i>EML average</i>	31,12%	29,95%	29,40%	31,84%	34,33%

Source: authors' calculations

To sum up our findings regarding the errors of the EML models, every model statistically significantly outperforms the benchmark. In addition, one individual model, namely Bagging, is capable of consistently outperforming even the model average.

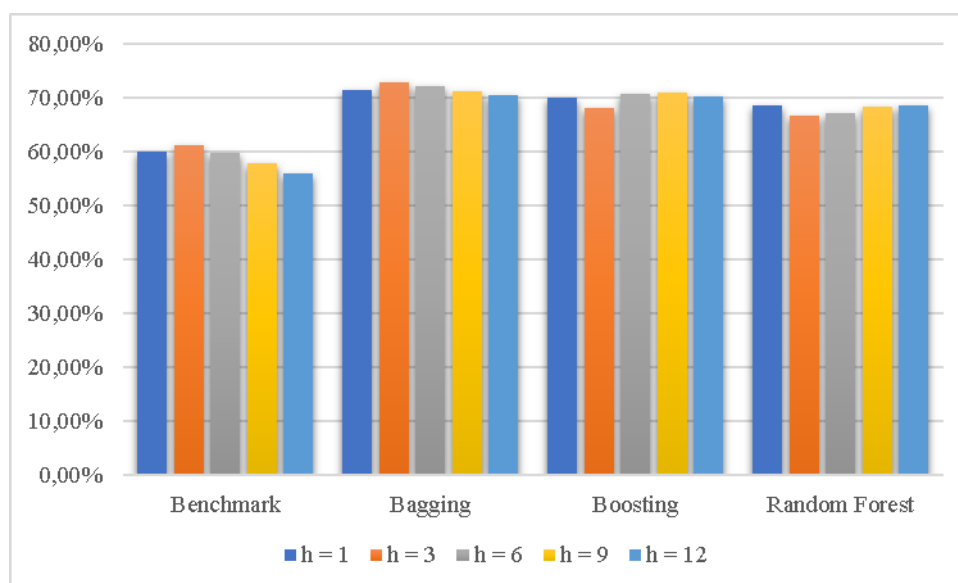
Moving on to directional accuracy, every EML model is more likely to capture the correct direction of change in the target variable than the benchmark model. The results are in the 22. Table. The DA of Bagging is between 70.48% and 73.04%, the DA of Boosting is 65% - 70.62%, while the DA of Random Forest is 68% - 70% compared to the benchmark model, which has a DA between 56.07% and 61.27%, depending on the forecast horizon. In conclusion, the EML models improve DA and our main focus point is again the Bagged model which has the highest DA of all.

22. Table: Directional accuracy of EML models – inf

Directional Accuracy	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	60,00%	61,27%	59,74%	57,89%	56,07%
<i>Bagging</i>	71,43%	73,04%	72,31%	71,15%	70,48%
<i>Boosting</i>	65,71%	69,12%	69,49%	67,38%	70,42%
<i>Random Forest</i>	68,57%	68,63%	70,00%	69,71%	68,64%

Source: authors' calculations

What is more, this improvement is consistent over time, as shown in the 11. Chart. To sum up, every EML forecast model has far higher DA values than the benchmark model.



11. Chart: Directional accuracy of EML models over time - inf

Source: authors' calculations

In the 23. Table we present the results of the remaining models, namely the Support Vector Machine, the Feedforward Neural Network and the Long-Short Term Memory Neural Network. Firstly, we see that the lack of computational resources made the Neural Network models useless in our case. The problem is described in detail in subchapter 5.1.1, so we do not dwell on it here. To continue with, we focus our attention to the Support Vector Machine model, for the performance of this model is far more compelling when forecasting inflation than when forecasting industrial production. On the one hand, it has a higher relative MSE than the others described previously in this subchapter. On the other hand, the relative MSE and MAE values of the model are almost identical, meaning that the model manages to capture important patterns in the data and can accommodate potentially large forecast errors without being severely affected by them. In addition, these results are highly significant at the 5% level for every horizon. The DA values of the model are between 68.77% and 74.15%, meaning the model outperforms the benchmark model in terms of DA, and its performance is generally on par with Ridge and Bagging. The DA values are slightly lower than those of the Elastic Net model and are slightly better than those of Lasso, Boosting and Random Forest. In conclusion, we cannot recommend using NN models, but not because they are inherently unfit for the Slovakian economy, but because our computational limitations did not allow for exploring all their possibilities. Regarding the SVM model, it can be used, for it provides highly significant

outperformance compared to the benchmark and the DA values produced by this model are comparable to the DA values produced by both the RLS and EML methods.

23. Table: Inf forecasts with SVM, FFNN and LSTM

Relative MSE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Support Vector Machine</i>	0,83966	0,85146	0,85869	0,85031	0,82470
<i>Feedforward NN</i>	458,15	451,18	438,00	420,98	418,35
<i>LSTM NN</i>	5716,14	1617,63	1219,65	477,68	637,59
Relative MAE					
<i>Support Vector Machine</i>	0,86623	0,87089	0,87021	0,86015	0,86015
<i>Feedforward NN</i>	27,9724	27,7361	27,2919	26,7198	26,7198
<i>LSTM NN</i>	76,7502	42,9465	35,6332	21,4885	21,4885
Diebold – Mariano test					
<i>Support Vector Machine</i>	0,00709	0,01365	0,01686	0,01555	0,00413
<i>Feedforward NN</i>	1	1	1	1	1
<i>LSTM NN</i>	1	1	0,99997	0,99998	0,99905
Directional accuracy					
<i>Benchmark</i>	60,00%	61,27%	59,74%	57,89%	56,07%
<i>Support Vector Machine</i>	68,57%	70,59%	73,08%	73,48%	74,15%
<i>Feedforward NN</i>	44,29%	45,10%	45,90%	46,42%	46,75%
<i>LSTM NN</i>	48,57%	37,75%	54,62%	57,89%	53,25%

Source: authors' calculations

To sum up, in this subchapter we analyze the performance of all the machine learning models mentioned in chapter 3 for forecasting inflation based on hard data. Most of the machine learning models statistically significantly outperform the benchmark model by a huge margin, except for the neural network models, where we had computational limitations. Firstly, we forecast inflation using the RLS models. All three RLS models statistically significantly outperform the benchmark model at every forecasting horizon at the 5% level. What is more, the Elastic Net model is more accurate than the category average and it has the lowest forecast errors out of all models. In addition, the Elastic Net model has the highest DA values, but the values of Ridge, the Bagging and Support Vector Machine are comparable. The DA values of Bagging and Elastic Net are identical at shorter horizons and Elastic Net only gains advantage from $h = 6$. Secondly, the EML models have slightly higher errors and comparable directional accuracy values, but they are also capable of statistically significantly outperforming the benchmark model at the 5% level at every horizon. Thirdly, the SVM model delivers slightly higher errors but highly statistically significant results at the 5% level at every horizon. In conclusion, we recommend using the Elastic Net model for regularization and the Bagged

model to capture nonlinear relationships in the data. Both of these models have the lowest errors in their respective categories and are statistically significant at the 5% level at every horizon. In addition, we the Support Vector Machine models fits the data as well, but has slightly higher errors than the previous two. To continue with, we explore the forecasting performance of the machine learning models based on common factors in the next subchapter.

5.2.2. Forecasting inflation using common factors

Shintani (2005) and Maehashi and Shintani (2020) apply machine learning models to forecasting based on common factors extracted from the available dataset. In this subchapter we see if this approach outperforms the benchmark model and if yes, how does it compare to the forecasting performance based on hard macroeconomic data without dimensional reduction.

To begin with, the 24. Table presents the relative MSE values of the forecasts based on common factors. The benchmark MSE is normalized to 1.

24. Table: Relative MSE values of inf forecasts based on common factors

Relative MSE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Lasso</i>	0,91935	0,92376	0,93345	0,93323	0,91158
<i>Ridge</i>	0,52143	0,51825	0,52280	0,51764	0,50053
<i>Elastic Net</i>	0,60710	0,61172	0,60056	0,59217	0,56605
<i>Bagging</i>	0,82478	0,83360	0,85232	0,84480	0,80954
<i>Boosting</i>	0,93111	0,93141	0,87634	0,86724	0,94901
<i>Random Forest</i>	0,82074	0,84776	0,81986	0,76227	0,71753
<i>Support Vector Machine</i>	0,70042	0,71128	0,71516	0,70507	0,69529

Source: authors' calculations

The 24. Table shows that forecasts based on common factors still outperform the benchmark model. There is no relative MSE that is higher than one. Firstly, considering RLS methods, the performance of the Lasso model deteriorates. Elastic Net still delivers good results, while the results of the Ridge model improved substantially and it is now the best performing RLS specification. Regarding ELS methods, Bagging and Random Forest have an almost identical performance except for $h = 12$, which where the latter outperforms the former by almost 0.08. Boosting performs worse than these two. The results of the SVM model improve substantially. To sum up, forecasts based on common factors still outperform the benchmark.

To continue with, we show the percentage difference between the relative MSE values of the forecasts based on hard macroeconomic data and common factors in the 25. Table. Positive values mean that the model based on hard data is more accurate than the model based on

common factors. Negative values mean the exact opposite. Firstly, the 25. Table shows that in most cases, the models based on hard data are more accurate. There are, however, a couple of exceptions. Firstly, the Ridge model based on common factors outperforms the Ridge based on hard macroeconomic data. In addition, Elastic Net has some horizons at which it outperforms the forecasts based on hard macroeconomic data. The percentage differences are small though, between 2.2% and 5.3%. To continue with, the SVM model also has lower errors compared to the model forecast based on hard data. The performance of the SVM model is also consistent. In conclusion, forecasts based on common factors outperform our benchmark, but underperform forecasting based on hard macroeconomic data in most cases, except for the Ridge and SVM models, while the performance of Elastic Net models is almost identical. Even though SVM is better when used on common factors, it has the higher errors of all models when used on hard data. In addition, none of the EML models, which can capture nonlinearity, manages to do well using factor models. Because of these, we still advise to use hard macroeconomic data and not the dimensionally reduced information set.

25. Table: % difference between forecasts based on factors and hard data - inf

% difference	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Lasso</i>	7,93%	9,57%	10,37%	13,06%	10,81%
<i>Ridge</i>	-20,62%	-22,27%	-20,69%	-21,15%	-21,03%
<i>Elastic Net</i>	6,33%	-2,24%	-3,20%	8,21%	-5,30%
<i>Bagging</i>	22,03%	20,75%	26,90%	21,03%	22,67%
<i>Boosting</i>	40,72%	27,41%	19,19%	5,04%	16,06%
<i>Random Forest</i>	7,53%	10,42%	9,40%	2,75%	-1,77%
<i>Support Vector Machine</i>	-16,58%	-16,46%	-16,72%	-17,08%	-15,69%

Source: authors' calculations

To end with, we show the directional accuracy of forecasts based on common factors in the 26. Table. The 26. Table further supports our finding that hard macroeconomic data should be used. Directional accuracy values based on common factors are generally higher than the benchmark, but they are lower compared to the results based on hard macroeconomic data. In addition, for example at $h = 1$, the benchmark outperforms Boosting in terms of DA, which is not the case when using real data. To sum up, factor models have generally higher DA values than the benchmark, but hard data should be used for forecasting as it delivers more compelling results.

In summary, in this subchapter we present the forecasting results of machine learning models based on common factors. Firstly, forecasts based on common factors outperform the benchmark. The outperformance is true for both the relative MSE values and DA values. This outperformance is, however, smaller than in the case of forecasts based on hard macroeconomic

data. The SVM and Ridge models are the only exception, with the Elastic Net model delivering similar results. Because of our findings, we cannot consider common factors an appropriate substitute to hard macroeconomic data, given the forecast errors of the models based on them are higher compared to the results in subchapter 5.2.1. In the next chapter we assess the likelihood of accurately forecasting the direction of change in inflation based on soft indicators.

26. Table: Directional accuracy of inf forecasts based on common factors

Directional accuracy	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	60,00%	61,27%	59,74%	57,89%	56,07%
<i>Lasso</i>	64,29%	65,69%	66,15%	66,13%	66,10%
<i>Ridge</i>	67,14%	67,16%	67,44%	67,56%	67,09%
<i>Elastic Net</i>	68,57%	67,16%	68,72%	68,64%	68,50%
<i>Bagging</i>	68,57%	71,57%	70,77%	69,53%	69,49%
<i>Boosting</i>	58,57%	60,29%	59,74%	57,53%	62,29%
<i>Random Forest</i>	64,29%	64,22%	62,56%	62,19%	65,11%
<i>Support Vector Machine</i>	60,00%	61,27%	59,74%	57,89%	56,07%

Source: authors' calculations

5.2.3. Forecasting inflation based on soft indicators

27. Table: Relative MSE of inf forecasts based on indicators

Relative MSE	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	1	1	1	1	1
<i>Lasso</i>	0,99920	1,00930	1,02142	1,01731	0,97916
<i>Ridge</i>	1,00141	1,00944	1,02160	1,01633	0,97927
<i>Elastic Net</i>	0,94591	0,95314	0,96228	0,95526	0,91147
<i>Bagging</i>	1,16871	1,20947	1,18994	1,18321	1,14751
<i>Boosting</i>	1,16607	1,11865	1,06712	1,09477	1,07865
<i>Random Forest</i>	1,00079	0,99324	1,00011	1,00889	0,99492
<i>Support Vector Machine</i>	0,94591	0,95053	0,96228	0,95526	0,91147

Source: authors' calculations

The introduction in subchapter 5.1.3 holds true for soft indicators. The results of machine learning and benchmark model forecasts based on these indicators are in the 27. Table. As expected, the 27. Table shows that indicators are not the most appropriate source of information when we are interested in the magnitude of changes in the macroeconomic variables. This is because they essentially produce the same forecasts as the benchmark model, which requires far less data and computational resources. Some machine learning models have relative MSE values under 1, what is more, the SVM and Elastic Net models have no relative MSE over 1, but all the values are so close to 1 that they are practically the same. To sum up, based on our

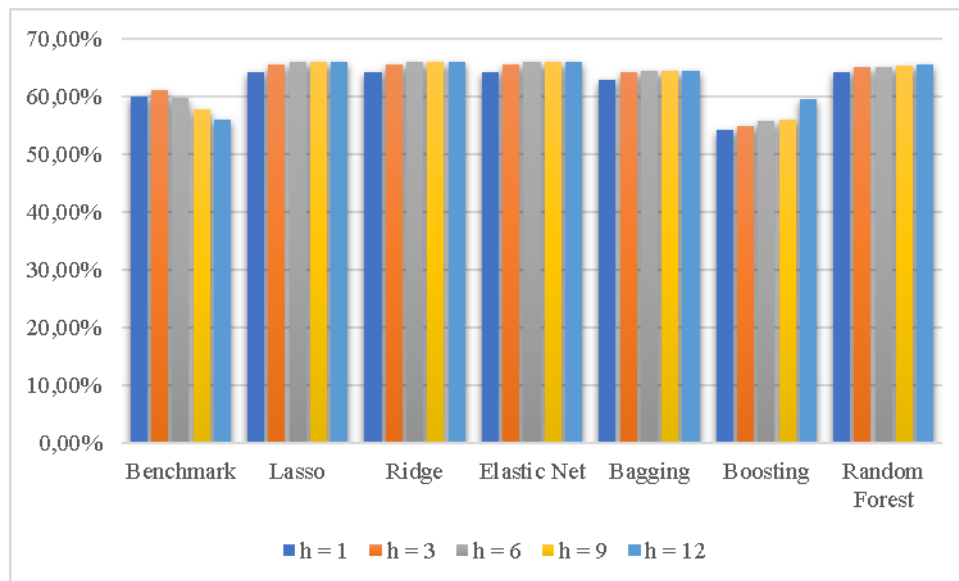
results, hard data should be used if want to minimize the relative MSE of the forecasts or, if a large dataset is not available, the benchmark data driven model is as reliable as models estimated using indicator data. Our expectations are, therefore, met.

28. Table: Directional accuracy of inf forecasts based on indicators

Directional accuracy	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Benchmark</i>	60,00%	61,27%	59,74%	57,89%	56,07%
<i>Lasso</i>	64,29%	65,69%	66,15%	66,13%	66,10%
<i>Ridge</i>	64,29%	65,69%	66,15%	66,13%	66,10%
<i>Elastic Net</i>	64,29%	65,69%	66,15%	66,13%	66,10%
<i>Bagging</i>	62,86%	64,22%	64,62%	64,52%	64,41%
<i>Boosting</i>	54,29%	54,90%	55,90%	56,09%	59,60%
<i>Random forest</i>	64,29%	65,20%	65,13%	65,41%	65,68%
<i>Support Vector Machine</i>	64,29%	65,69%	66,15%	66,13%	66,10%

Source: authors' calculations

To continue with, we analyze directional accuracy, our main measure of interest in the case of indicator data. The DA values are in the 28. Table. A couple of interesting things stand out. Firstly, every machine learning model outperforms the benchmark model at every horizon. The RLS models produce the same DA values, making the choice between them irrelevant. The EML models also generally outperform the benchmark model, except for Boosting, which has lower DA values. Although one model performs poorly, the others perform as expected. Secondly, the outperformance is smaller than in the case of using hard macroeconomic data. This is, however, expected. Forecasts based on indicator data have lower DA values than forecasts based on hard macroeconomic data, but this is the price of speed. As stated above, indicators are available earlier and they measure sentiments so we cannot directly compare the DA values in the 28. Table to those in the previous subchapters. We must compare them to the benchmark, which they beat comfortably with one exception. Thirdly, this outperformance is also consistent, as presented on the 12. Chart. Fourthly, there are no big differences between models, meaning that we can select the model which requires the least computational capacity (in our case Lasso) and use it for computing directional accuracy. In conclusion, indicators serve their purpose. By using an extensive dataset of soft indicators and machine learning models, we can predict the direction of change in inflation with a higher likelihood than if we only used the benchmark model.



12. Chart: DA of inf forecast based on indicators over time

Source: authors' calculations

In summary, forecasts based on indicator data produce similar errors but simultaneously have a higher likelihood of correctly predicting the direction of change in a variable than the benchmark model, as expected.

To sum up subchapter 5.2, we use machine learning models for forecasting inflation. Firstly, both RSE and EML models statistically significantly outperform the benchmark model and has higher directional accuracy values. The models in both categories statistically significantly outperform the benchmark at every horizon at the 5% level, showing that regularization results in substantial forecast improvement and nonlinear relationships play a crucial role in the data generating process of Slovakia. In addition, both categories have individual models that outperform the category average. On the other hand, neural networks cannot produce reliable results due to computational limitations and the performance of the Support Vector Machine model is relatively good but still has the poorest performance among all the models. Secondly, forecasts based on common factors outperform their benchmarks both in terms of MSE and DA values but, as expected based on Shintani (2005) and Maehashi and Shintani (2020), their performance is not as good as the performance of models based on hard macroeconomic data. Thirdly, indicator data produces higher DA values than the benchmark, as expected. Based on these facts, we accept the hypotheses formulated in chapter 3 for the forecasts of inflation.

5.3. COMPOSITE FORECASTS

In this subchapter we analyze the performance of composite forecasts. As mentioned in subchapter 3.8., we use machine learning models for combining individual forecasts as shown by Araujo and Gaglianone (2023). We use all machine learning methods described in chapter 3 and we use the individual forecasts as inputs instead of the predictor variables. We compare them to the trimmed mean of all forecasts as advised by Armstrong (2001) and Burgi (2015). We leave out neural networks from the mean, for they would distort the results. The results of forecasting industrial production are presented in the 29. Table.

29. Table: Composite forecasts of IP

Directional accuracy	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Trimmed mean	0,44083	0,43735	0,44043	0,4378	0,43511
Lasso	0,27252	0,29045	0,28669	0,32022	0,31707
Ridge	0,28593	0,3212	0,27647	0,32703	0,31233
Elastic Net	0,30582	0,30701	0,28009	0,31747	0,29678
Bagging	0,31726	0,32532	0,27547	0,28584	0,27467
Boosting	0,57966	0,54257	0,58092	0,58916	0,64293
Random Forest	0,33085	0,33675	0,31563	0,29577	0,28865
Support Vector Machine	0,33088	0,35622	0,36441	0,40139	0,38146

Source: authors' calculations

In the 29. Table we see that except for Boosting, every composite forecast using machine learning methods manages to outperform the trimmed mean, which is very hard to do, according to the general consensus in the forecasting literature presented by, for example, Burgi (2015) and Armstrong (2001). Another interesting thing stands out. Except for Lasso and Elastic Net, where the individual models based on hard data slightly beat the composite forecast, every other model manages to produce lower forecast errors than it produces when it is trained on hard data. On the other hand, it is important to highlight that to the authors' knowledge, only Araujo and Gaglianone (2023) tried to use machine learning methods as forecast combination tools so far and they have only used two methods, namely adaLasso and Random Forest and nobody has done the combination using so many models as we do in this study. Because of this, based on the preliminary results we can conclude that machine learning models can be useful tools for forecast combination in Slovakia for forecasting industrial production, but further exploration of this weighting scheme is advised.

It is also advised because in the case of inflation, presented in the 30. Table, machine learning models prove themselves to be less accurate than the trimmed mean of forecasts. Because of this, we cannot provide a definitive conclusion about the performance of machine

learning methods as forecast combination tools in Slovakia. In the next chapter we provide a holistic summary of our results.

30. Table: Composite forecasts of inf

Directional accuracy	h = 1	h = 3	h = 6	h = 9	h = 12
<i>Trimmed mean</i>	0,67346	0,68505	0,69728	0,68977	0,68630
<i>Lasso</i>	0,89061	0,96500	0,96675	0,91257	1,01434
<i>Ridge</i>	0,74474	0,87802	0,76659	0,79503	0,78751
<i>Elastic Net</i>	0,76111	0,82509	0,77153	0,74990	0,79821
<i>Bagging</i>	0,81391	0,93277	0,77861	0,75734	0,74510
<i>Boosting</i>	0,91323	0,92965	0,92797	0,90487	0,86995
<i>Random Forest</i>	0,83277	0,89174	0,74641	0,71348	0,79087
<i>Support Vector Machine</i>	0,88922	1,01558	0,92117	0,83955	0,94612

Source: authors' calculations

6. DISCUSSION

This chapter summarizes our key findings. The principal aim of this paper is to thoroughly analyze the forecasting performance of the selected machine learning models compared to a benchmark model using a novel large dataset consisting of all the important macroeconomic variables of the Slovak economy. We forecast industrial production and inflation.

To the authors' knowledge, we are the first to provide a thorough overview of the macroeconomic forecasting performance of machine learning models for Slovakia. In addition, as mentioned in chapter 1, there are multiple papers employing these techniques in large economies, but the evidence on their performance in developed small open industrialized economies is scarce. What is more, most papers only focus on the magnitude of errors while ignoring the directional accuracy of the forecasts. Our results are as follows.

To begin with, we make rolling window forecasts of the index of industrial production and harmonized index of consumer prices using a benchmark model and machine learning models based on stationarized and standardized hard macroeconomic data.

Firstly, every RLS model statistically significantly outperforms the benchmark at the 5% level and there are only two instances when the outperformance is significant only at the 10% level in the case of industrial production. The outperformance is consistent for every model and at every horizon. In addition, the Lasso and Elastic Net models, which are almost identical, outperform even the category average for industrial production. The Elastic Net model outperforms the category average for inflation as well. Moreover, every RLS forecast model has a substantially higher DA than the benchmark model for industrial production. Even though the DA values of machine learning models are somewhat lower for inflation, they still outperform the benchmark. Based on these findings, we consider the Elastic Net model as the best of this category. We also highlight that for industrial production, variable selection introduced by Lasso plays a more important role than coefficient shrinkage introduced by Ridge. This tendency is reversed for inflation. What is more, the fact that the models are generally highly statistically significant at the 5% level signals that regularization can substantially improve forecast accuracy, more precisely by more than 70% in the case of Lasso and Elastic Net, yet these methods have not been used so far in Slovakia. We hope that our findings point out its important place and we advise macroeconomists to add forecasting models incorporating regularization to their toolkit. By incorporating this into the macroeconomic forecasting framework, we can arrive to even more accurate forecasts in the future.

Secondly, regarding the errors of the EML models, every model statistically significantly outperforms the benchmark for both industrial production and inflation at the 5% level, except for Boosting and Random Forest for industrial production at $h = 1$, which are only significant at the 10% level. The best overall model is the Bagged model, which is statistically significant at the 5% level at every horizon for both industrial production and inflation and consistently outperforms the category average for both target variables. Moreover, every EML forecast model have higher DA values than the benchmark model and except for Boosting, the DA values are consistent over time. It is clear from our analysis that the Bagged model fits our dataset best, both for industrial production and inflation, but the other models are close to it in terms of performance. Given that the Bagged model is significant for both target variables for all horizons at the 5% level and generally has the lowest forecast errors, we conclude that it best fits our data and should be used to capture nonlinearities in Slovakia. What is more, given the good performance of the other two models as well, we conclude that nonlinearities, for example those mentioned in chapter 1, play a crucial role in the Slovakian DGP and should be accounted for by macroeconomists when preparing forecasts.

Thirdly, there are three other types of models outside of the two main categories. The Support Vector Machine model outperforms the benchmark at every horizon for both of the variables and the DA values are compelling. On the other hand, the SVM delivers mixed performance for industrial production and generally it is one of the weakest performing models in the model suite, if we do not consider neural networks. Makridakis et al (2018) also formulate their criticism of machine learning methods partially based on SVM. As we show, however, even though SVM performs generally worse, other models deliver strong performance. Regarding the neural network models, we have already mentioned multiple times that our computational capacity was limited and we could not fine tune them, so the results are not useful.

In summary, our main recommendation is that macroeconomists in Slovakia should stick to the models in the main categories which do a good job at forecasting and thus capturing nonlinearities in the data generating process and pointing to the importance of regularization, which, as shown in this work, are important. What is more, nonlinear methods perform well, but their results are hard to interpret, as shown by Joseph et al. (2022). The results of RLS models, however, are more easily interpretable. Given that Elastic Net delivers the best overall performance, we can do variable selection and calculate Shapely ratios to interpret the results. In general, based on our results, it might be possible to open the so-called machine learning

black box. This, however, should be a subject of further research because entangling these relationships and variable interactions can take another paper in and of itself.

To continue with, we test the forecasting capacity of the models based on common factors for both industrial production and inflation. Every model outperforms their benchmark, but the models perform far better in terms of relative MSE and DA values when the forecasts are based on hard macroeconomic data. In the case of industrial production, the Support Vector Machine does better based on common factors, but the improvements are very small. In the case of inflation, both the Ridge and Support Vector Machine models do better when the forecasts are based on common factors. SVM, however, is the worst performing model when the forecasts are based on hard data. In conclusion, every model is capable of beating the benchmark based on common factors, but most of them cannot perform as well as they do when they are trained on a complete macroeconomic dataset. Our results are thus similar to Maehashi and Shintani (2020), who show that machine learning forecasts based on common factors can beat the benchmark but are generally worse performing than the models trained on the whole dataset. Because of these findings, we advise macroeconomist to do not use machine learning models based on common factors in Slovakia, but stick to using hard macroeconomic data for forecasting macroeconomic variables.

Moreover, we explore how well the machine learning models perform when they are trained on soft indicator data. As stated in the previous chapters, we do not expect forecast based on soft indicators to have lower errors than forecasts based on hard macroeconomic data, but they should be able to beat the benchmark at predicting the direction of change in the target variable. Our assumption is proven to be correct. We combine more than 250 time series of different soft indicators and show that machine learning models, given enough information, can forecast the correct direction of change in the variable with a higher likelihood than the benchmark model. The main implication of this is that policymakers can have machine learning models prepared and as soon as the indicator data arrives, it can be fed to these models to see with a higher likelihood which way the economy is going.

To end with, we experiment with the ability of machine learning methods to formulate composite forecasts. Our results are mixed. On the one hand, for industrial production, there are only two individual models, namely Lasso and Elastic Net which are more precise than the composite forecasts of machine learning methods. These forecasts can also beat the trimmed mean. On the other hand, the performance of these composite forecasts for inflation is poor. To the authors' knowledge, our study is the second which tries to use machine learning methods as tool of making composite forecasts. The previous study of Araujo and Gaglianone (2023),

however, is limited in scope as stated in subchapter 3.8. Even though the results for industrial production are compelling, the poor performance for inflation does not allow us to reach a definitive conclusion. We can only state that, given the good results for industrial production, using machine learning models as forecast combination tools should be explored in far more depth. Again, this study could fill a paper in and of itself.

H1	Nonlinearities play a statistically significant role in the data generating process of the Slovak macroeconomic time series.	✓
H2	Regularization can statistically significantly improve the quality of the macroeconomic forecasts of our target variables.	✓
H3	There is at least one individual model from each of the two categories that is more accurate and has lower forecast errors than the category average.	✓
H4	When using indicator data, forecasting models based on machine learning are more likely to forecast the correct direction of the change in the variable than the benchmark model.	✓
H5	Machine learning models based on common factors have lower forecast errors and are more likely to forecast the correct direction of the change in the forecasted variable than the benchmark model.	✓
H6	Machine learning models generally outperform the benchmark model for the selected target variables in Slovakia.	✓

After summarizing our results, it is clear that none of the hypotheses from chapter 2 can be rejected. Both RLS and EML categories have at least one model for both target variables that is statistically significant at the 5% level at every horizon, meaning that nonlinearities play an important role in the data generating process of the Slovak macroeconomic time series and regularization, which provides the best models in general, can substantially increase forecasting accuracy. To be more precise, it increases accuracy by more than 70% for industrial production and 37%-46% for inflation. The best models are the Elastic Net model and Bagged model, respectively. These models are also more accurate and have lower forecast errors than the category average. The hypothesis regarding the soft indicators cannot be rejected and we show that common factor based forecasts can outperform the benchmark model. In addition, 94% of forecasts are statistically significant at the 5% level and 100% at the 10% level for industrial production and 100% and 100% for inflation, respectively. Based on this we conclude that machine learning models generally outperform the benchmark.

7. CONCLUSION

In this study we forecast industrial production and inflation using machine learning methods. We use the publicly available all-encompassing dataset of the National Bank of Slovakia. The dataset is 12 years long with multiple periods and we allow for time-variance of the parameter estimates, lending robustness to our results. We select two categories of machine learning models to capture nonlinearities in the data and test the effects of regularization on the results of macroeconomic forecasts. We also select the Support Vector Machine to show that the generalized critique of Makridakis et al. (2018), mentioned in chapter 1, is unfounded in the case of Slovakia.

Our study is the first to use machine learning methods for macroeconomic forecasting in Slovakia. Our results have very important implications both for applied macroeconomics in Slovakia in particular, and the forecasting literature in general. Firstly, methods based on regularization have been ignored so far in Slovakia, despite improving the forecasting performance by more than 70% in some cases. Macroeconomists should incorporate these methods into their forecasting toolkit. Secondly, the good performance of Ensemble Machine Learning methods shows the importance of nonlinearities - of which we show some examples in chapter 1 - in the DGP of Slovakia. Because of this, methods capable of capturing nonlinear relationships should also be incorporated in the macroeconomic toolkit. After extensive testing, both regularization and EML methods can become a part of the toolkit used for preparing official macroeconomic forecasts of Slovakia. What is more, macroeconomists should dive extensively into opening the machine learning black box, as in Joseph et al. (2022), to make the results interpretable. Thirdly, we use the Support Vector Machine and show that yes, this method performs comparatively poorly, as in Makridakis et al. (2018). We also show, however, that the forecasts are statistically significantly outperform the benchmark and more complex methods deliver far better performance. Because of this, the critique of machine learning models proposed by Makridakis et al. (2018) is unfounded in Slovakia. Fourthly, forecasts based on common factors outperform the benchmark but perform worse than forecasts based on hard macroeconomic data. Due to this, macroeconomists should explore the possibility of switching to regularization from PCA to achieve the same goal (dimensionally reduced information set) but increase accuracy. Fifthly, we show that soft indicators have higher directional accuracy values. Because of this, macroeconomists should prepare machine learning methods so they can get a better intuition about the direction of the economy as soon as new indicator data becomes available. This can help them provide more sound policy advise in a

timely manner. In addition, directional accuracy has largely been ignored in the machine learning literature. We advise fellow researches to explore the performance of these models in different countries. Sixthly, we show that machine learning models can perform well as forecast combination tools for industrial production but so well for inflation. Because of this, this field should be the subject of further testing.

In conclusion, the institutions preparing official macroeconomic forecasts in Slovakia should begin the testing of the empirical applications of machine learning models to capture new information in official macroeconomic forecast, while researchers should devote more attention to directional accuracy measures, soft indicators and regularization instead of PCA. Opening the machine learning black box and rigorously testing the capability of machine learning models as tools of forecast combination should be the subject of further research.

REFERENCES

- Araujo, G.S. – Gaglianone, W. P. (2023): Machine learning methods for inflation forecasting in Brazil: New contenders versus classical models. *Latin American Journal of Central Banking*, Vol. 4., pp. 1-29.
- Armstrong, J. S. (2001): Combining Forecasts. *University of Pennsylvania Scholarly Commons Marketing Papers* 34. <https://core.ac.uk/download/pdf/76379527.pdf> (downloaded: 3.2.2019).
- Athey, S. (2019): The impact of machine learning on economics. *The Economics of Artificial Intelligence*. NBER, Washington.
- Atkeson, A. - Ohanian, L. E. (2001): Are Phillips curves useful for forecasting inflation? *Federal Reserve Bank of Minneapolis Quarterly Review*, Vol. 25., No. 1., pp. 2–11. <https://www.minneapolisfed.org/research/quarterly-review/are-phillips-curves-useful-for-forecasting-inflation> (downloaded: 16.9.2023).
- Bai, J. - Ng, S. (2002): Determining the number of factors in approximate factor models. *Econometrica*, Vol. 70., No. 1., pp. 191–221.
- Bates, J.M.- Granger, C.W.J. (1969): The combination of forecasts. *Journal of the Operational Research Society*, Vol. 20., pp. 451–468.
- Bergmeir, C.- Hyndman, R. J. - Koo, B. (2018): A note on the validity of cross-validation for evaluating autoregressive time series prediction. *Computational Statistics and Data Analysis*, Vol. 120., pp. 70–83.
- Bolhuis, A. M. – Rayner, B. (2020): Deus ex Machina? A Framework for Macro Forecasting with Machine Learning. *IMF Working Paper WP/20/45*. <https://www.imf.org/en/Publications/WP/Issues/2020/02/28/Deus-ex-Machina-A-Framework-for-Macro-Forecasting-with-Machine-Learning-49094> (downloaded: 6.5.2023).
- Breiman, L. (1996): Bagging predictors. *Machine Learning*, Vol. 24., No. 2., pp. 123–140.
- Breiman, L. (2001): Statistical Modeling: The Two Cultures. *Statistical Science*, Vol. 16., No. 3., pp. 199-231.
- Breiman, L. (2001): Random forest. *Machine Learning*, Vol. 45., No. 1., pp. 5–32.
- Breiman, L. - Friedman, J. - Stone, C.J. - Olshen, R.A. (1984): Classification and Regression Trees. *Chapman and Hall/CRC, New York, NY*.

- Bühlmann, P. - Yu, B. (2002): Analyzing bagging. *Annals of Statistics*, Vol. 30., No. 4., pp. 927–961.
- Bürgi, C. (2015): Can a subset of forecasters beat the simple average in the SPF? *RPF Working Paper No. 2015-001*. <https://www2.gwu.edu/~forcpgm/2015-001.pdf> (downloaded: 1.2.2019).
- Chakraborty, C. - Joseph, A. (2017): Machine Learning at Central Banks. *Bank of England Working Paper No. 674*. https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3031796 (downloaded: 11.7.2023).
- Clements, M. P. - Hendry, D.F. (2011): *The Oxford Handbook of Economic Forecasting*. Oxford University Press, Oxford.
- Constantini, M. – Cuaresma, J. C. – Hlouskova, J. (2016): Forecasting Errors, Directional Accuracy and Profitability of Currency Trading: The case of EUR/USD Exchange Rate. *Journal of Forecasting*. Vol. 35., pp. 652 – 668.
- Coulombe, P. G. – Leroux, M. – Stevanovic, D. (2022). How is machine learning useful for macroeconomic forecasting? *Journal of Applied Econometrics*, Vol. 37., No. 5., pp. 920–964. <https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.2910> (downloaded: 30.4.2023).
- Dickey, D. A. – Fuller, W. A. (1979): Distribution of the Estimators for Autoregressive Time Series with a Unit Root. *Journal of the American Statistical Association*, Vol. 74., No. 366., pp. 427–431. <https://www.jstor.org/stable/2286348> (downloaded: 15.9.2023).
- Faust, J. - Wright, J. (2013): Forecasting Inflation. In: *Handbook of Economic Forecasting*. Elsevier, Amsterdam.
- Gu, S. - Kelly, B. - Xiu, D. (2019): Empirical asset pricing via machine learning. *NBER Working Paper* 25398, pp. 1–46. https://www.nber.org/system/files/working_papers/w25398/w25398.pdf (25.7.2023).
- Giannone, D.- Lenza, M. - Primiceri, G. (2018): Economic Predictions with Big Data: The Illusion of Sparsity. *Northwestern University Working Paper*. <https://faculty.wcas.northwestern.edu/gep575/illusion5-1-colors.pdf> (downloaded: 7.8.2023).
- Hochreiter, S. - Schmidhuber, J. (1997): Long short-term memory. *Neural Computation*, Vol. 9., No. 8., pp. 1735–1780.
- Hoerl, A.E. - Kennard, R.W. (1970): Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, Vol. 12., No. 1., pp. 55–67.

- Inoue, A. - Kilian, L. (2008): How useful is bagging in forecasting economic time series? A case study of U.S. consumer price inflation. *Journal of American Statistical Association*, Vol. 103., No. 482., pp. 511–522.
- Joseph, A. – Potjagailo, G. – Kalamara, E. – Chakraborty, Ch. – Kapetanios, G. (2022): Forecasting UK inflation bottom up. *Bank of England Staff Working Paper No. 915*.
- Jung, J. – Patnam, M. – Ter-Martirosyan, A. (2018): An Algorithmic Crystall Ball: Forecasts Based on Machine Learning. *IMF Working Papers*, Vol. 230., No. 18., pp. 1-34. <https://www.imf.org/en/Publications/WP/Issues/2018/11/01/An-Algorithmic-Crystal-Ball-Forecasts-based-on-Machine-Learning-46288> (downloaded: 20.6.2023).
- LeCun, Y. - Bengio, Y. - Hilton, G. (2015): Deep learning. *Nature*, Vol. 521., pp. 436-444.
- Maehashi, K. – Shintani, M. (2020): Macroeconomic forecasting using factor models and machine learning: an application to Japan. *Journal of The Japanese and International Economies*, Vol. 58., pp. 1-17.
- Mahajan, K. – Srinivasan, A. (2019): Inflation Forecasting in Emerging Markets: A Machine Learning Approach. *CAFRAL Working Paper*, pp. 1-45. https://ifrogs.org/PDF/CONF_2019/Mahajan_Srinivasan_2019.pdf (downloaded: 13.6.2023)
- Makridakis, S. - Spiliotis, E. - Assimakopoulos, V. (2018): Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward. *PLoS One*, No. 13. <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0194889> (downloaded: 2.6.2023)
- Masini, R. P. – Medeiros, M. C. – Mendes, Eduardo F. (2021): Machine Learning Advances for Time Series Forecasting. *The Journal of Economic Surveys*, Vol. 37., No. 1., pp. 76-111.
- Masters, T. (1993). *Practical Neural Network Recipes in C++*. Sand Diego: Academic Press.
- Medeiros, M. - Mendes, E. (2016): Regularization of High-Dimensional Time-Series Models with Non-Gaussian and Heteroskedastic Errors. *Journal of Econometrics*, Vol. 191., pp. 255–271.
- Medeiros, M. C. – Vasconcelos, G. – Veiga, A. (2019): Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods. *Journal of Business and Economic Statistics*, Vol. 39., No. 1., pp. 1-45.
- Mullainathan, S. and Spiess, J. (2017). Machine learning: An applied econometric approach. *Journal of Economic Perspectives*, Vol. 31., No. 2., pp. 574–589.
- Pescatori, A., - Zaman, S. (2011): Macroeconomic models, forecasting, and policymaking. *Economic Commentary of the Federal Reserve Bank of Cleveland 2011-19*.

- https://www.researchgate.net/publication/227437211_Macroeconomic_Models_Forecasting_and_Policymaking (downloaded: 20.7.2023).
- Sanyal, A. - Roy, I. (2014): Forecasting major macroeconomic variables in India: Performance comparison of linear, non-linear models and forecast combinations. *RBI Working Paper Series* No. 11/2014.
https://www.researchgate.net/publication/308718137_Forecasting_Major_Macroeconomic_Variables_in_India_-_Performance_Comparison_of_Linear_Non-linear_Models_and_Forecast_Combinations (downloaded: 25.7.2023).
- Schapire, R.E. (1990): The strength of weak learnability. *Machine Learning*, Vol. 5., No. 2., s. 197–227.
- Shintani, M. (2005). Nonlinear forecasting analysis using diffusion indexes: an application to Japan. *Journal of Money, Credit and Banking*, Vol. 37., No. 3., pp. 517–538.
- Stock, J. - Watson, M. (2002): Macroeconomic Forecasting With Diffusion Indexes. *Journal of Business and Economic Statistics*, Vol. 20., pp. 147–162.
- Stock, J. - Watson, M. (2007): Why Has U.S. Inflation Become Harder to Forecast?. *Journal of Money, Credit and Banking*, Vol. 39., pp. 3–33.
- Stock, J. - Watson, M. (2010): Modeling Inflation After the Crisis. *Technical Report 1, National Bureau of Economic Research*.
- Tibshirani, R. (1996). Regression shrinkage and selection via lasso. *Journal of the Royal Statistical Society: Series B*, Vol. 58., No. 1., pp. 267–288.
- Vargas, A. R. (2020): Forecasting Costa Rican inflation with machine learning methods. *Latin American Journal of Central Banking*, No. 1., pp. 1–21.
- Zou, H. and Hastie, T. (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B*, Vol. 67., No. 2., pp. 301–320.