UNIVERSITY OF ECONOMICS IN BRATISLAVA
FACULTY OF ECONOMIC INFORMATICS

Evidence number: 103003/D/2024/421000078076

FORECASTING MACROECONOMIC VARIABLES USING MACHINE LEARNING

Dissertation thesis

2024 **Ing. Ádám Csápai**

UNIVERSITY OF ECONOMICS IN BRATISLAVA
FACULTY OF ECONOMIC INFORMATICS

FORECASTING MACROECONOMIC VARIABLES USING MACHINE LEARNING

Dissertation thesis

Study program: Data Science in Economics

Field of study: 6258 Quantitative Methods in Economics – Data Science in Economics

Department: Department of operations research and econometrics

Thesis advisor: prof. Ing. Martin Lukáčik, PhD.

Bratislava, 2024 **Ing. Ádám Csápai**

Ekonomická univerzita v Bratislave
Fakulta hospodárskej informatiky

ZADANIE ZÁVEREČNEJ PRÁCE

Meno a priezvisko študenta: Ing. Ádám Csápai
Študijný program: data science v ekonómii (Jednoodborové štúdium, doktorandské III. st., denná forma)
Študijný odbor: ekonómia a manažment
Typ záverečnej práce: Dizertačná záverečná práca
Jazyk záverečnej práce: slovenský
Sekundárny jazyk: anglický

Názov: Prognózovanie makroekonomických premenných pomocou metód strojového učenia

Literatúra:
1. BOX, G. E. P., JENKINS, G. M. Time Series Analysis: Forecasting and Control. 1st ed. San Francisco: Holden-Day, 1970.
2. CHAKRABORTY, C., JOSEPH, A. Machine Learning at Central Banks. In Bank of England Staff Working Papers [online]. London: Bank of England, 9.2017, issue 674.
3. CLEMENTS, M. P., HENDRY, D. F. The Oxford Handbook of Economic Forecasting. 1st ed. Oxford: Oxford University Press, 2011.
4. GIANNONE, D., LENZA, M., PRIMICERI, G. E. Economic predictions with big data: the illusion of sparsity. In European Central Bank Working Paper Series. Frankfurt: European Central Bank, 4.2021, issue 2542.
5. MAKRIDAKIS, S., SPILIOTIS, E., ASSIMAKOPOULOS, V. Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward. In PLOS One, 2018, vol. 13, issue 3, pp. 1–35.
6. MASINI, R. P., MEDEIROS, M. C., MENDES, E. F. Machine Learning Advances for Time Series Forecasting. In The Journal of Economic Surveys. 2020, vol. 37, issue 1, pp. 76–111.

Cieľ: Hlavným cieľom práce je zhodnotenie predikčného výkonu rôznych vybraných modelov strojového učenia a porovnanie týchto výsledkov s vhodným referenčným modelom. Prognózujeme reprezentantov kľúčových slovenských makroekonomických časových radov; konkrétne priemyselnú produkciu reprezentujúcu vývoj HDP a infláciu. Práca by mala prispieť k rozvoju oblasti ekonomického prognózovania, najmä v kontexte aplikácie pokročilých výpočtových techník na analýzu makroekonomických údajov.

Anotácia: Prognózovanie zohráva dôležitú úlohu pri hodnotení ekonomickej situácie a usmerňovaní tvorby hospodárskej politiky. Je základom pre zostavovanie štátnych rozpočtov a nevyhnutné pre tvorcov politík, ako sú centrálne banky, aby mohli efektívne načasovať zásahy na základe predpovedí kľúčových ekonomických ukazovateľov. S príchodom veľkých dát, zlepšenými výpočtovými schopnosťami a pokrokmi v oblasti štatistického učeniamajú ekonómovia prístup k rôznym novým metódam, vrátane tých založených na strojovom učení. Táto práca predstavuje zhodnotenie možností modelov strojového učenia pre prognózovanie makroekonomických údajov na Slovensku. Cieľom práce je poukázať na relatívnu účinnosť modelov strojového učenia v tomto ohľade v porovnaní s konvenčnými metódami.

Ekonomická univerzita v Bratislave
Fakulta hospodárskej informatiky

Školiteľ:	prof. Ing. Martin Lukáčik, PhD.
Katedra:	KOVE FHI - Katedra operačného výskumu a ekonometrie
Vedúci katedry:	prof. Mgr. Juraj Pekár, PhD.
Dátum zadania:	12.03.2021

Dátum schválenia: 01.09.2022

prof. Mgr. Juraj Pekár, PhD.
osoba zodpovedná za realizáciu študijného programu

Acknowledgments

First and foremost, I extend my deepest gratitude to the supervisor of my dissertation thesis, Prof. Ing. Martin Lukáčik, PhD., for his invaluable guidance, assistance, expert advice, and constructive feedback throughout the preparation of my work. Additionally, I would like to thank Mr. Professor Dr. Jesús Crespo Cuaresma for his mentorship, support, and expert insights. I am also deeply grateful to my family for their unwavering support throughout my studies.

ABSTRAKT

CSÁPAI, Ádám: *Prognózovanie Makroekonomických Premenných Použitím Metód Strojového Učenia: Prípad Slovenska.* – Ekonomická Univerzita v Bratislave. Fakulta hospodárskej informatiky; Katedra operačného výskumu a ekonometrie. – Školiteľ: prof. Ing. Martin Lukáčik, PhD. – Bratislava: FHI EU, 2024, 148 s.

Táto dizeračná práca sa zameriava na skúmanie účinnosti modelov strojového učenia (ML) pri zlepšovaní makroekonomických prognóz, konkrétne zameraných na ukazovatele priemyselnej produkcie a inflácie v Slovenskej republike. Porovnaním rôznych modelov ML, ako sú regularizované metódy najmenších štvorcov, techniky strojového učenia založené na ensemble, neurónové siete a podporné vektory, s tradičnými metódami, výskum zdôrazňuje, že ML je nadradené v zachytávaní zložitých, nelineárnych vzťahov, ktoré tradičné modely často prehliadajú. Významná pozornosť štúdie je venovaná prognózovaniu po COVIDe, s dôrazom na odolnosť a adaptabilitu modelov na náhle ekonomické zmeny. Zistenia ukazujú, že modely ML, najmä tie, ktoré využívajú techniky ensemble a regularizácie, konzistentne predčia tradičné metódy prognózovania, čo naznačuje významný potenciál ML pri zvyšovaní presnosti makroekonomických predikcií. Štúdia nielenže poskytuje tvorcom politík a ekonomickým analytikom presvedčivé dôvody pre integráciu ML do ekonomického prognózovania, ale tiež prispieva do akademického diskurzu tým, že kládne silný empirický základ pre budúci výskum aplikácie ML v ekonomickom prognózovaní.

Kľúčové slová: prognózovanie, strojové učenie, Slovensko, inflácia, priemyselná produkcia

ABSTRACT

CSÁPAI, Ádám: *Forecasting Macroeconomic Variables Using Machine Learning: The case of Slovakia.* – Unviersity of Economics in Bratislava. Faculty of Economic Informatics; Department of operations research and econometrics. – Thesis Advisor: prof. Ing. Martin Lukáčik, PhD. – Bratislava: FHI EU, 2024, 148 p.

This study aims to explore the effectiveness of machine learning (ML) models in improving macroeconomic forecasting, specifically targeting industrial production and inflation indicators within Slovakia. By comparing various ML models, such as Regularized Least Squares, Ensemble Machine Learning techniques, Neural Networks, and Support Vector Machines, against traditional methods, the research highlights ML's superior ability to capture complex, nonlinear relationships that conventional models often miss. A notable focus of the study is on post-COVID economic forecasting, emphasizing the models' resilience and adaptability to sudden economic shifts. The findings reveal that ML models, particularly those using ensemble and regularization techniques, consistently outperform traditional forecasting methods, suggesting a significant potential for ML to enhance the accuracy of macroeconomic predictions. The study not only offers policymakers and economic analysts compelling reasons to integrate ML into economic forecasting but also contributes to the academic discourse by laying a strong empirical foundation for future research in ML's application to economics.

Key words: forecasting, machine learning, Slovakia, industrial production, inflation

Contents

Introduction

Forecasting plays a vital role in assessing the economic condition and guiding economic policymaking. It is foundational for crafting government budgets and crucial for policymakers, like Central Banks, to time interventions effectively based on forecasts of key economic indicators such as Gross Domestic Product (GDP), inflation, and unemployment. Current models, however, often fail to capture the true dynamics between economic variables. For example, Medeiros et al. (2019) illustrate how governments and international bodies, especially the ECB, tend to consistently overestimate inflation projections. Such discrepancies can lead to significant welfare losses and skew inflation expectations, underlining the need for more precise forecasting models.

With the emergence of big data, enhanced computational capabilities, and advances in statistical learning, economists now have access to a variety of new methods, including those based on machine learning. These methods have become increasingly popular in macroeconomic applications over the last decade, particularly in the last five to six years.

This thesis presents a rigorous assessment of machine learning models for forecasting in Slovakia. We aim to determine the relative efficacy of machine learning models in capturing the complexities of the Slovak economy compared to conventional methods.

Our contributions to macroeconomic forecasting are varied and significant. Firstly, we propose a hybrid approach, inspired by Medeiros et al. (2019), that effectively captures nonlinearities and variable interactions. This is particularly useful in post-socialist economies of Eastern Europe, where datasets are often short. Regularizing these datasets before applying nonlinear methods significantly improves the performance of nonlinear models. Secondly, we compare the effectiveness of regularization and principal component analysis (PCA) in handling dimensionally reduced datasets. Our findings indicate that regularization, a machine learning (ML) technique, yields more accurate forecasts. Thirdly, we explore the use of ML methods as tools for combining forecasts, demonstrating that these can enhance the accuracy of individual forecasts. Fourthly, we demonstrate that regularization can significantly enhance forecasting capabilities compared to traditional benchmarks. Fifthly, we highlight and capture nonlinearities in the data using Ensemble ML models. To our knowledge, this is the first application of these two techniques using ML in a small, open industrialized economy within a monetary union with a short dataset. Sixthly, we are the pioneers in assessing the directional accuracy of ML models, as existing studies typically focus only on error magnitudes. However,

the directions of change are equally crucial. Seventhly, we provide a comprehensive overview of ML model performance in both pre-COVID and post-COVID periods, showing that ML models perform even better during periods of increased economic volatility and uncertainty. Although other studies involve periods of crisis as well, they have decades worth of data that can smooth out the effects of crises. Our sample has two crises despite having only 16 years data, both at the beginning and end. One model even completely accurately forecasts the highly uncertain inflation one to three months ahead. Eighthly, we demonstrate how ML methods can more effectively capture trends based on soft indicators. Ninthly, all of this should serve as a response to the critique of Makridakis et al. (2018), further elaborated on in Chapter 1.

The thesis is organized as follows. The first chapter reviews the existing literature on macroeconomic forecasting with machine learning methods. It provides a general overview, highlights the key findings in the field and justifies our benchmark. The second chapter states our primary and partial aims, along with the scientific hypotheses. The third chapter provides an exhaustive overview of our methodology, including the mathematical formulation of the models, the codes we utilize, the forecast evaluation process and the data preprocessing steps. The fourth chapter presents the results for both the pre-COVID and post-COVID samples. The fifth chapter summarizes and interprets the results, while also giving recommendations. The final chapter concludes.

1 Current state of the field at home and abroad

Breiman (2001) promotes the use of models driven by data that don't rely on any fixed structures. He emphasizes that these models should not make any assumptions about the Data Generating Process (DGP), which makes them more adaptable to changes in the DGP. Moreover, Breiman (2001) argues that models should not assume any specific relationships between predictors and target variables, making them more robust to errors in model design. Machine learning models fit these requirements well.

Before moving on to machine learning, it is important to explain why we chose our benchmark models and how machine learning compares with traditional methods found in academic studies. Stock and Watson (2007) analyze a model that evolves over time based on unobserved components. Then, Stock and Watson (2010) examine models based on Random Walks and univariate time series, finding that these models are very challenging to improve upon for macroeconomic forecasting. Faust and Wright (2013) review previous research and agree with these findings. Therefore, we choose Random Walk and ARIMA models as our benchmarks, as these are commonly used standards in economic forecasting literature.

Vargas (2020) observes that advancements in computing power, statistical learning theory, and the increased availability of big data over the past decade have led to the adoption of machine learning methods in economic forecasting. Historically, these methods were primarily applied to classification tasks, such as predicting loan delinquencies or consumer purchasing decisions, where they surpassed the performance of traditional models. Although these applications focus on discrete variables, Vargas (2020) points out that machine learning can also effectively predict continuous variables such as GDP or inflation. According to Vargas (2020), machine learning's ability to capture nonlinear relationships in the data enhances our understanding of complex economic phenomena like asymmetric business cycles, stock market volatility, and regime switches.

The conclusions of Vargas (2020) are supported by Athey (2019), who notes that over the past decade, economists have increasingly turned to machine learning for forecasting, especially when large datasets are available. These methods are mostly used for forecasting financial variables such as stock prices, as there is ample data. An example case study of machine learning forecasting is provided by Cibuľa and Tkáč (2023), who use machine learning methods to predict bitcoin and other cryptocurrency spot prices. They compare the prediction capacity of multiple supervised learning algorithms, including ensemble machine learning and

neural networks. They conclude that their machine learning algorithms are capable of beating the benchmark. While Cibuľa and Tkáč (2023) focus on cryptocurrencies and Athey (2019) primarily discusses the use of microdata for forecasting, Mullainathan and Spiess (2017) investigate the application of these methods in macroeconomic forecasting. All of these authors argue that macroeconomists should add machine learning tools to their forecasting arsenal.

Bolhuis and Rayner (2020) analyze the theoretical aspects of Ordinary Least Squares (OLS) in comparison to factor models and machine learning. They point out that OLS predictions may be unsuitable due to issues like predictor relevance, nonlinearity, collinearity, and high dimensionality. While factor models have been central in data-driven forecasting for decades and can mitigate problems related to dimensionality and collinearity, they fall short when it comes to addressing nonlinearity and predictor relevance, often resulting in less accurate forecasts. In addition to theoretical limitations, Shintani (2005) and Maehashi and Shintani (2020) provide empirical evidence showing that factor models often underperform. These authors not only use dynamic factor models for forecasting, but they also employ principal component analysis to extract common factors. They then apply machine learning techniques using these factors as a basis. Their findings support that while methods based on common factors can surpass the benchmarks, as noted by Stock and Watson (2002), they generally fall short when compared to machine learning models that utilize hard data. With regards to theory, Bolhuis and Rayner (2020) state that in contrast to factor models, machine learning methods excel at capturing nonlinear dynamics within high-dimensional datasets. They adeptly learn from complex historical data relationships without making unwarranted future projections. Thus, despite the effectiveness of simple data-driven models, their drawbacks have prompted researchers to explore machine learning as a viable alternative for macroeconomic forecasting.

To elaborate further, we describe these methods within a macroeconomic framework. Masini et al. (2021) review recent progress in the field and note that machine learning encompasses various approaches. Specifically for macroeconomic forecasting, they describe machine learning as the integration of automated algorithms and statistical methods to identify patterns in large datasets. They also summarize the differences between the three widely used categories of machine learning, namely supervised, unsupervised, and reinforcement learning.

Like Masini et al. (2021), we utilize supervised learning techniques for our forecasting purposes. Supervised learning involves training the model on data that is structured as input-output pairs. For example, in linear regression models, the inputs (or predictors) and the outputs

(target variable forecasts) are paired clearly. The model learns a function that maps the input to the output, aiming to predict the value of the target variable effectively. The specifics of individual machine learning methods employed are discussed in the methodological section of this thesis.

The empirical evidence on the macroeconomic forecasting capacity of machine learning methods is scarce but compelling. Chakraborty and Joseph (2017) apply machine learning to predict medium-term inflation rates and report that their selected methods can outperform traditional benchmarks by as much as 29%. Similarly, Jung et al. (2018) employ methods like Elastic Net, SuperLearner, and Recurrent Neural Networks to forecast macroeconomic variables for seven advanced and emerging economies, achieving results that surpass the benchmark World Economic Outlook (WEO) forecasts.

The most often cited study claiming that machine learning models are inadequate forecasting tools is that of Makridakis et al. (2018). Makridakis et al. (2018) conduct a comparative examination of machine learning (ML) and traditional statistical approaches to forecasting, utilizing data from the M3 Competition. They employ various models, including basic neural networks (such as the Multilayer Perceptron and Bayesian variant), advanced neural networks (such as Recurrent and Long-Short Term Memory), Support Vector Machine, a single CART Regression Tree, and basic ensemble models. Their sample spans 14 years and encompasses 3003 time series. They discuss how traditional statistical methods like ARIMA and ETS often outperform more complex ML techniques, noting that ML methods entail higher computational costs and tend to overfit. However, they acknowledge that their findings may be specific to the dataset used, which comprises a maximum of 126 monthly observations for 3003 variables. In our thesis, we demonstrate that with a dataset of comparable length but fewer variables, regularization can significantly enhance forecasting performance. Additionally, recent papers by Masini et al. (2021) and Coulombe et al. (2022), which offer updated perspectives on ML forecasting, largely disregard eight of the ten methods proposed by Makridakis et al. (2018), except for RNN and LSTM networks, as Medeiros et al. (2019) argue that the remaining methods are too simplistic. Similarly, we conclude that employing more advanced methods leads to substantial improvements in forecasting over general benchmarks like ARIMA and its variants, employed by Makridakis et al. (2018).

In response to the critique by Makridakis et al. (2018), Medeiros et al. (2019) demonstrate that machine learning can improve forecasting accuracy by up to 30% for U.S. inflation data. They utilize a large dataset and employ various methods, analyzing data from periods of both

high and low economic uncertainty. According to Medeiros et al. (2019), the main criticisms against machine learning are unfounded when more advanced methods are applied.

Although these studies recognize that machine learning methods can outperform traditional benchmarks at both micro and macro levels, they often overlook the underlying reasons for this superior performance. A second significant limitation, as pointed out by Coulombe et al. (2022), is their limited scope. These studies typically rely on relatively small datasets, forecast only a single target variable across a few time horizons, and generally use between one and three models. The notable exception is Medeiros et al. (2019), which responds directly to the critique of Makridakis et al. (2018). Furthermore, except for Medeiros et al. (2019), the primary goal of these papers is to conduct a "forecasting horserace" using hard macroeconomic data, focusing predominantly on one accuracy metric, such as mean squared error or root mean squared error. They neglect the performance of machine learning methods based on common factors, an aspect only explored by Shintani (2005) and Maehashi and Shintani (2020). Additionally, these studies often concentrate solely on minimizing forecast error, overlooking directional accuracy. They also generally fail to explain why these methods perform well or what might cause one method to outperform another.

These shortcomings are identified by Coulombe et al. (2022), who conduct a forecasting exercise using US data from the past four decades to explore the factors contributing to the superior performance of machine learning methods. They forecast multiple target variables across five different time horizons using various models, pinpointing the critical success factors for machine learning in forecasting. Firstly, they find that nonlinearities significantly impact the data generating process in the US, enhancing the predictive power of machine learning models. Secondly, they demonstrate that regularization techniques greatly improve forecasting accuracy, outperforming traditional factor models. Lastly, they establish that B-fold cross-validation is more effective for model selection than the traditional information criteria typically used in econometrics.

Medeiros et al. (2019) also delve into why regularization and nonlinearities are critical in macroeconomic forecasting. Starting with regularization: in recent years, as big data becomes more prevalent, macroeconomists are reevaluating the significance of each variable within these large datasets. In addition to factor models, there has been an increased use of shrinkage and variable selection techniques. Notably, Medeiros and Mendes (2016), and Giannone et al. (2021) recommend Lasso models for macroeconomic forecasting, which have shown to outperform traditional benchmarks. These studies primarily focus on US inflation data and

short-term forecasting. The most extensive study by Medeiros et al. (2019) reveals that although shrinkage methods are slightly less precise than tree-based methods in the US, they significantly enhance forecasting accuracy.

Regarding nonlinearities, Medeiros et al. (2019) also provide a theoretical backdrop. For example, the relationship between inflation and employment is nonlinear and depends on economic slackness. Additionally, nonlinearity arises from uncertainty; for instance, the nonlinear hiring practices due to the high costs associated with firing employees. Furthermore, the zero lower bound (ZLB) on nominal interest rates introduces nonlinearity between inflation, employment, and interest rates, especially under unconventional monetary policies. The role of houses as collateral and their interaction with monetary policy and financial intermediation further complicates this nonlinearity. As evidenced during the Great Recession, the burst of a housing bubble can precipitate severe credit downturns, marked by nonlinear dynamics. Our dataset for Slovakia encompasses the Great Recession and a prolonged period of near-ZLB interest rates, suggesting that nonlinearities are significant in the timeframe under study. What is more, Obradović and Lojanica (2022) provide further evidence of nonlinearities in inflation. They test the presence of a unit root in the inflation of the selected Western Balkan countries. The authors find that inflation in Serbia and Bosnia and Herzegovina are best described by nonlinear mean reverting behavior.

To summarize the theoretical section, we first highlight the importance of providing accurate macroeconomic forecasts. We then outline the key advantages of data-driven models that rely on statistical and machine learning techniques. We also explain our rationale for selecting specific benchmark models. Additionally, we highlight the limitations of the main forecasting models commonly cited in the literature and discuss how machine learning can address these shortcomings. Moving forward, we delve into the application of machine learning within the macroeconomic framework, noting that while empirical results are promising, they often have a limited scope. Most studies focus on a small number of models over few forecasting horizons, are predominantly conducted on large economies, and largely neglect directional accuracy. Finally, we conclude this section by emphasizing the importance of considering regularization techniques and nonlinear relationships in the data to enhance forecasting accuracy and model robustness. In the next chapter we describe our aims. dataset, the forecasting setup and the methods selected for forecasting.

2 Aims of the thesis

In this chapter, we outline the objectives of our research. Our principal aim is to rigorously evaluate the forecasting performance of various selected machine learning models and compare these results against an econometric benchmark model. To the authors' knowledge, there is no published study at the time of writing this thesis that dwells on the applicability and performance of these methods in a small open industrialized economy in a monetary union.

We forecast key Slovakian macroeconomic time series; specifically, industrial production and inflation. Medeiros et al. (2019), Coulombe et al. (2022), and Maehashi and Shintani (2020) also forecast these variables, deemed to be representative indicators of a nation's economic health, to assess the performance of machine learning models.

To fulfill the principal objective of our study, we have established several specific goals:

- **Methodological detailing**. In Chapter 3, we describe the machine learning methods used in our research in detail. We also outline our three key methodological contributions. These consist of combining regularized and nonlinear methods, using machine learning methods as forecast combination tools, and comparing the performance of principal component analysis with regularization as a dimensional reduction technique.
- **Data description.** We provide a detailed description of the macroeconomic database provided by the National Bank of Slovakia. This part also includes an overview of the data gathering and preprocessing steps taken to prepare the data for analysis.
- **Performance analysis.** We conduct a thorough analysis of the performance of various machine learning models in forecasting Slovakian macroeconomic variables. This involves testing specific hypotheses related to the effectiveness of these models compared to traditional forecasting methods.
- **Summarization and Recommendations.** Finally, we aim to summarize the results of our study and provide policy and methodological recommendations based on our findings. We also suggest areas for further research that could build on our work, potentially leading to improved forecasting methods or applications in other economic contexts.

By systematically addressing these aims, we hope to contribute significantly to the field of economic forecasting, particularly in the context of applying advanced computational techniques to the analysis of macroeconomic data.

To achieve our objectives, we need to adopt the following primary and secondary scientific hypotheses for each forecasted variable. The primary scientific 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 true, then when employing observable "hard" data, at least one machine learning model from either the Regularized Least Squares or the Ensemble Machine Learning categories, as specified later in Chapter 3, should statistically significantly outperform the benchmark at every forecast horizon.

Our primary aim is a rigorous assessment of model performance, which lacks an exact definition. However, based on Coulombe (2022) we believe that, in addition to the main hypotheses, our secondary scientific hypotheses can be considered comprehensive. These secondary scientific hypotheses are:

H3	Hybrid models can enhance the forecast accuracy of nonlinear methods.
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	Regularization based methods based on hard macroeconomic data deliver better performance than dimensional reduction based on PCA.

To summarize, this chapter presents the principal aim of the thesis along with the partial aims, and both the primary and secondary hypotheses. In the following section, we describe the tools used to achieve these aims.

3 Methodology and methods

In this chapter, we outline the methods and methodology used to obtain our results. Firstly, we briefly describe the R statistical programming software used for our calculations. The packages used for modeling are listed in Table 1. Secondly, we introduce our benchmark models. As we automate the exploration of functional space, Table 2 presents different benchmark possibilities.

Thirdly, we discuss the first group of methods, known as Ensemble Machine Learning. Fourthly, we provide a theoretical description of Regularized Least Squares methods. We selected multiple models from each category to provide a thorough overview of each category's performance and to highlight the main benefits and limitations of each method. We categorize the models following Coulombe et al. (2022), grouping them by their ability to either regularize or capture nonlinear relationships. We chose three methods from each category to ensure our results are robust and not merely due to chance from selecting a single method.

Fifthly, we present one of our key methodological contributions: the potential to combine these two model categories to enhance results. Sixthly, we outline the two types of neural networks considered for forecasting. Seventhly, we characterize the Support Vector Machine Regression. Eighthly, we describe our forecasting setup, focusing on the rolling window approach, extracting and applying common factors, and the methods by which we prepare composite forecasts using machine learning.

The ninth subchapter presents the forecast evaluation procedures. The tenth subchapter explains stationarization, describes the macroeconomic database of the National Bank of Slovakia, and outlines our database and observation periods.

In summary, this chapter provides a comprehensive overview of our methodology, machine learning methods, the evaluation of their performance, and data description.

3.1 RStudio

RStudio[1] is an integrated development environment (IDE) tailored specifically for the R programming language, which is extensively utilized in scientific research and data analysis. It

[1] https://posit.co/download/RStudio-desktop/

features a variety of packages and offers a user-friendly interface for composing, executing, and debugging R code, making it an indispensable tool for researchers engaged in statistical analysis, data visualization, and scientific modeling.

RStudio enhances productivity through tools like syntax highlighting, code completion, and built-in data visualization capabilities, which help streamline data-driven workflows. This makes it a highly efficient, intuitive, and user-friendly platform with extensive resources for implementing machine learning methods. The list of the packages, along with their respective machine learning methods is provided in Table 1. The code snippets of each method along with the CV and manual hyperparameter setup are presented in the subchapters of the particular method's description.

Table 1: R packages

Method	Package
Benchmark	"forecasting" by Hyndman and Khandakar (2008)
Ridge	"glmnet" by Friedman et al. (2010)
Lasso	"glmnet" by Friedman et al. (2010)
Elastic Net (EN)	"glmnet" by Friedman et al. (2010)
Random Forest (RF)	"randomForest" by Liaw and Wiener (2002)
Boosting	"caret" by Max (2008)
Bagging	"caret" by Max (2008)
Support Vector Machine (SVM)	https://cran.r-project.org/web/packages/e1071/index.html[2]
Feedforward Neural Network (FFNN)	"nnet" by Venables and Ripley (2002)
Long Short-Term Memory Neural Network (LSTM)	https://tensorflow.RStudio.com/[3]

Source: authors' own work

Moreover, except for a few hyperparameters which we have to manually specify, we utilize these packages to automatically adjust the hyperparameters during each iteration of the forecasting procedure. This approach ensures that we achieve the optimal values for each individual forecast. In summary, RStudio serves as an effective and straightforward tool for macroeconomic forecasting with machine learning, so we use it.

[2] This package lists no citation info at the website so I can only provide the website link.
[3] This package lists no citation info at the website so I can only provide the website link.

3.2 Benchmark models

In this subchapter we detail the forecasting methods employed in our study. Our computational work is conducted using RStudio. As benchmarks, we employ models such as Autoregressive Integrated Moving Average (ARIMA) and Random Walk (RW), the latter being a simpler subset of ARIMA models. These models are selected based on their fit as determined by the Akaike Information Criterion (AIC), which RStudio calculates automatically.

$$y_{t+h} = y_t + \epsilon_{t+h}, \tag{1}$$

where y_{t+h} represents the forecast of the target variable at time $t+h$, y_t is the last observed value of the target variable at time t, and ϵ_{t+h} is the white noise error term at time $t+h$. Besides Stock and Watson (2010), Atkenson and Ohanian (2001) also show that despite being a simple model, the Random Walk is hard to beat in forecasting exercises. Pratap and Sengupta (2019), Mahajan and Srinivasan (2019) and Medeiros et al. (2019) also use this model as one of their benchmarks.

Additionally, Pratap and Sengupta (2019) and Chakraborty and Joseph (2017) also incorporate the ARIMA(p,d,q) model of Box and Jenkins (1970) as another benchmark, underlining its prevalent use for comparative analysis in time series forecasting. We write an ARIMA model as

$$\phi(B)(1 - B^d)y_t = c + \theta(B)\varepsilon_t, \tag{2}$$

where ε_t is a white noise error term, d is the order of integration, 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.

As the model selection procedure is automated in R with the auto.arima function from the forecasting package of Hyndman and Khandakar (2008), other possible benchmark models, nested in the ARIMA framework, are presented in Table 2.

Table 2: Possible benchmark specifications

White noise	ARIMA(0,0,0) with no constant
Random walk	ARIMA(0,1,0) with no constant
Random walk with drift	ARIMA(0,1,0) with a constant
Autoregressive	ARIMA(p,0,0)
Moving average	ARIMA(0,0, q)

Source: Box and Jenkins (1970)

3.3 Ensemble Machine Learning

This subchapter presents the Ensemble Machine Learning methods, namely Random Forest, Boosting and Bagging. It is largely based on Masini et al. (2021). Before describing the methods, we provide a brief note on the symbols used. An uppercase letter like X represents a random variable, whereas a lowercase letter x indicates a fixed (non-random) value. Bold type, used for both $\boldsymbol{X}$ and $\boldsymbol{x}$, signifies multivariate entities like vectors or matrices. The notation $\|\cdot\|_q$ for $q \geq 1$ stands for the ℓ_q norm of a vector. For any set S we use $|S|$ to denote its size.

The objective is to forecast Y_{T+h}using T observations of the random $(Y_t, \boldsymbol{Z}_t')'$, where $h = 1, \dots, H$. The following derivations of machine learning methods assume that $\{(Y_t, \boldsymbol{Z}_t')'\}_{t=1}^{\infty}$ is a covariance-stationary stochastic process on $\mathbb{R}^{d+1}$. As a result, we stationarize the data prior to training.

To continue with, we define an n-dimensional vector of predictors $\boldsymbol{X}_t := \left(Y_{t-1}, \dots, Y_{t-p}, \boldsymbol{Z}_t', \dots, \boldsymbol{Z}_{t-r}'\right)'$ using fixed integers $p \geq 1$ and $r \geq 0$, where $n = p + d(r+1)$. In the rest of the thesis, we consider the direct forecasting model

$$Y_{t+h} = f_h(\boldsymbol{X}_t) + U_{t+h},\ h = 1, \dots, H,\ t = 1, \dots, T. \tag{3}$$

Here, $f_h: \mathbb{R}^n \to \mathbb{R}$ is an undefined (measurable) function, and $U_{t+h} := Y_{t+h} - f_h(\boldsymbol{X}_t)$ is presumed to have zero mean and finite variance.

In equation (3), the function $f_h(\boldsymbol{X}_t)$ is unknown, and for many applications, the assumption of linearity proves too limiting, necessitating more flexible forms. Assuming a quadratic loss function, the estimation problem becomes the minimization of the functional \($S(f)$, defined as

$$S(f) := \sum_{t=1}^{T-h} [Y_{t+h} - f(\boldsymbol{X}_t)]^2, \tag{4}$$

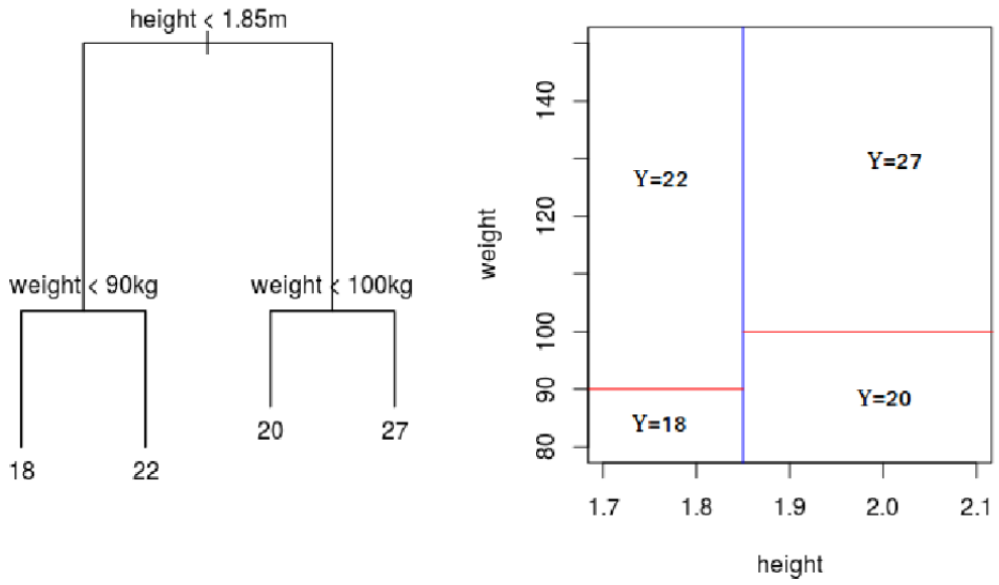
where $f \in \mathcal{G}$, a generic function space.

3.3.1 Decision Tree

Masini et al. (2021) explain the basic concept of a decision tree as follows. A decision tree is a nonparametric approach that through local approximations estimates an unknown nonlinear function $f_h(\boldsymbol{X}_t)$ from (3). This is achieved by recursively splitting the covariate space.

Masini et al. (2021) provide a graphical example. Figure 1 presents a decision tree on the left side, which corresponds to the spatial partitioning shown on the right side for a two-dimensional case. For instance, consider predicting basketball players' scores based on their height and weight. The tree's initial split divides players based on a height threshold of 1.85 meters, separating taller players from shorter ones. Following this, the tree further divides each group based on weight: the left branch continues with shorter players, and the right with taller ones. The scores are then predicted at each terminal node by calculating the average score within each group. The process of developing the tree involves determining the optimal point for splitting at each node, which involves selecting the best variable and the specific value for the split, such as height at 1.85 meters in the given example.

Figure 1: Decision Tree



Source: Masini et al. (2021)

The decision tree approximates the function $f_h(\boldsymbol{X}_t)$ using the formula

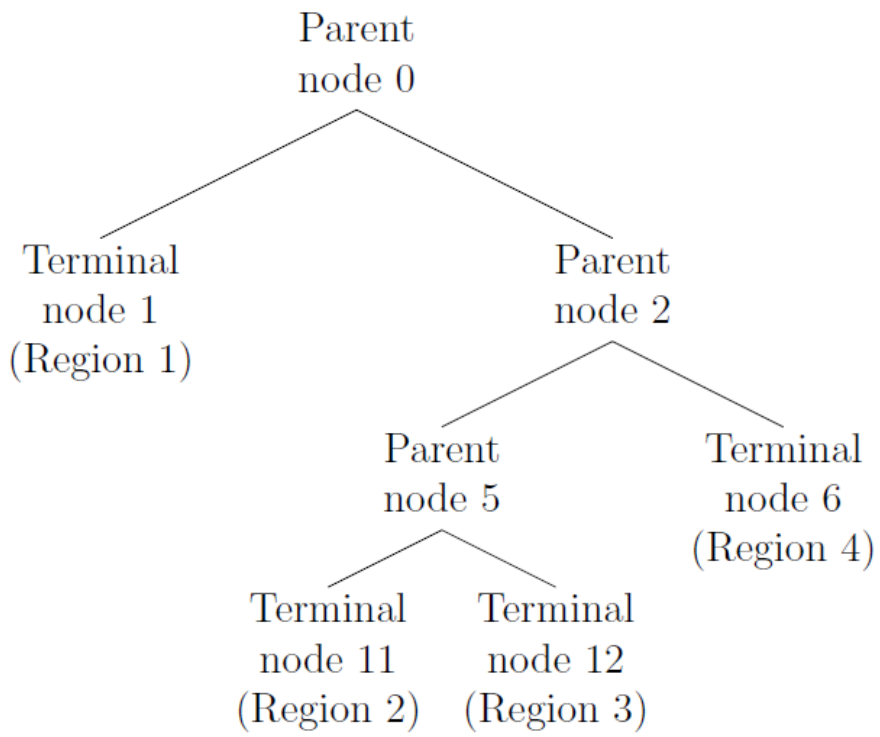
$$h_D(\boldsymbol{X}_t) = \sum_{j=1}^{J_T} \beta_j I_j(\boldsymbol{X}_t), \text{ where } I_k(\boldsymbol{X}_t) = \begin{cases} 1 & \text{if } \boldsymbol{X}_t \in \mathcal{R}_j, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

This model essentially represents a linear regression on J_T dummy variables, with each $I_j(\boldsymbol{X}_t)$ being a product of indicator functions that activate based on whether $\boldsymbol{X}_t \in \mathcal{R}_j$ defined by the tree's splits. Let $J := J_T$ denote the set of indices for parent nodes (where a leaf begins) and $N := N_T$ for terminal nodes (where a leaf ends). The tree's structure has regions labeled as $\mathcal{R}_1, \ldots, \mathcal{R}_j$, with the root node (first parent node) located at position 0. Each parent node at position j has

two child (split) nodes located at positions $2j + 1$ and $2j + 2$. Each parent node is associated with a threshold (split) variable $X_{s_j t}$, where $s_j \in \mathbb{S} = \{1,2, \dots, p\}$.

Figure 2 presents an example based on Masini et al. (2021). On Figure 2 the parent nodes are indexed by $\mathbb{J} = \{0,2,5\}$ and the terminal nodes, which are the leaf ends where predictions occur, are indexed by $\mathbb{T} = \{1,6,11,12\}$. This example shows how different regions and splits are organized within a regression tree to categorize data points based on their covariate values, thus facilitating the stepwise prediction process modeled by $h_D(\boldsymbol{X}_t)$.

Figure 2: Parent and terminal nodes of a decision tree



Source: Masini et al. (2021)

The approximating model can be written as

$$h_D(\boldsymbol{X}_t) = \sum_{i \in \mathbb{T}} \beta_i B_{\mathbb{J}i}(\boldsymbol{X}_t; \boldsymbol{\theta}_i), \tag{6}$$

where

$$B_{\mathbb{J}i}(\boldsymbol{X}_t;\boldsymbol{\theta}_i) = \prod_{j\in\mathbb{J}} I\left(X_{s_j,t};c_j\right)^{\frac{n_{i,j}(1+n_{i,j})}{2}} \times \left[1-I\left(X_{s_j,t};c_j\right)\right]^{(1-n_{i,j})(1+n_{i,j})},$$

$$I\left(X_{s_j,t};c_j\right) = \begin{cases} 1 & \text{if } X_{s_j,t} \le c_j \\ 0 & \text{otherwise,} \end{cases}$$

$$n_{i,j} = \begin{cases} -1 & \text{if the path to leaf } i \text{ does not include parent node } j; \\ 0 & \text{if the path to leaf } i \text{ include the right-hand child of parent node } j; \\ 1 & \text{if the path to leaf } i \text{ include the left-hand child of parent node } j. \end{cases}$$

Here, the notation $\mathbb{J}_i$ refers to the indices of parent nodes that are included in the path leading to leaf node i. Each leaf node i, which belongs to the set $\mathbb{T}$, is associated with a set of conditions $\boldsymbol{\theta}_i = \{c_k\}$ where k belongs to $\mathbb{J}_i$. These conditions c_k are thresholds or decision criteria at each parent node that must be satisfied for the path to proceed towards the terminal node i.

The expression $\sum_{j\in\mathbb{J}} B_{\mathbb{J}i}(\boldsymbol{X}_t;\boldsymbol{\theta}_j) = 1$ describes a fundamental property of the decision tree structure. Here, $B_{\mathbb{J}i}(\boldsymbol{X}_t;\boldsymbol{\theta}_j)$ is a binary function that evaluates to 1 if the input $\boldsymbol{X}_t$ meets the criteria specified by $\boldsymbol{\theta}_i$ for the path to node i, and 0 otherwise. This implies that for any given input $\boldsymbol{X}_t$, exactly one path through the tree will satisfy all the conditions from the root to a terminal node, ensuring that each input is uniquely classified into one region $\mathcal{R}_i$.

3.3.2 Random Forest

Breiman (2001) introduces Random Forest (RF), while Wager and Athey (2018) prove consistency and asymptotic normality of the RF estimator of $f_h(\boldsymbol{X}_t)$. Masini et al. (2021) provide a brief overview. Random Forest consists of multiple regression trees, each developed from a bootstrap sample of the initial dataset. Considering the context of time series data, a block bootstrap approach is utilized. Let's assume there are B bootstrap samples. For each sample $b, b = 1, \dots, B$, a tree containing K_b regions is constructed using a randomly chosen subset of the original regressors. The parameter $K_b$is selected to ensure a minimum number of observations in each region. The ultimate prediction is derived by averaging the outputs from each tree when applied back to the original data in the following way

$$\hat{Y}_{t+h|t} = \frac{1}{B}\sum_{b=1}^{B}\left[\sum_{i=1}^{\mathbb{T}_b}\hat{\beta}_{i,b}B_{]\,]i,b}(\boldsymbol{X}_t;\widehat{\boldsymbol{\theta}}_{i,b})\right]. \quad (7)$$

Besides the work of Wager and Athey (2018) on inference from random forests, Davis and Nielsen (2020) also demonstrate a uniform concentration inequality for regression trees

based on nonlinear autoregressive stochastic processes and establishes the consistency for a broad range of random forests.

Figure 3 depicts the R code for the Random Forest method, utilizing the "train" function from the caret package for optimal training outcomes. We restrict the number of variables randomly sampled as candidates at each split (known as "mtry") to 60 or fewer. This is a reduction from the default setting, which samples about one-third of the total variables, approximately 80 variables, and would significantly increase computational costs. However, we allow for fewer than 60 variables at splits if the data indicates that this is optimal. Other hyperparameters of the Random Forest method, such as the number of trees ("ntree"), the size of terminal nodes ("nodesize"), and their maximum number in a forest ("maxnodes"), are determined through 5-fold cross-validation at each iteration of the rolling window process[4].

Figure 3: R code of the Random Forest method

```
# Limit the range for mtry to make it more efficient
limited_mtry <- seq(1, min(60, ncol(x)), by = 1)

# Set up the Random Forest model
rf_model <- train(x, y,
                  method = "rf",
                  trControl = trainControl(method = "cv", number = 5),
                  tuneGrid = expand.grid(mtry = limited_mtry))
```

Source: based on Max (2008)

3.3.3 Boosting

Masini et al. (2021) reviews the Boosting algorithm. Boosting is a method that progressively refines the approximation of nonlinear functions using simple base learners, through a process called sequential approximation. Specifically, the version known as Gradient Boosting, introduced by Schapire (1990) and Friedman (2001), operates like a Gradient Descent in the space of functions.

This approach is iterative, where boosted decision trees aim to adjust the gradient of the loss function using small trees in each cycle. For a quadratic loss as in (4), which is the focus of this discussion, the algorithm effectively refits the residuals left from the previous round. The described boosting algorithm, particularly for quadratic losses, employs a shrinkage parameter $v \in (0,1]$ within the range of $(0,1]$ to moderate the learning speed. When v

[4] For more information on our forecasting setup, please refer to Subchapter 3.8.1

approaches 1, the model converges faster and fits better to the training data, though it risks overfitting and potentially poor performance on new, unseen data. Overfitting can significantly skew derivative estimates, even when only considering training data. Therefore, a learning rate between 0.1 and 0.2 is suggested to achieve a balance between convergence speed and overfitting mitigation.

The boosting algorithm works as follows:

1. Initialize:

$$\phi_{i0} = \bar{Y} := \frac{1}{T}\sum_{t=1}^{T} Y_t.$$

This value serves as the initial approximation of the response variable.

2. For each iteration $m = 1, \dots, M$:

(a) Compute residuals as

$$U_{tm} = Y_t - \phi_{tm-1}.$$

This step involves calculating the residuals between the actual data points and the predictions from the previous iteration.

(b) Fit a tree model as

$$\hat{u}_{tm} = \sum_{i \in \mathbb{T}_m} \hat{\beta}_{im} B_{\mathbb{J}_m i}(\boldsymbol{X}_t; \widehat{\boldsymbol{\theta}}_{im}).$$

Here, a tree model is grown to fit these residuals. Each tree makes a prediction $\hat{u}_{tm}$ which is a sum over the leaves i of the tree $\mathbb{T}_m$, weighted by coefficients $\hat{\beta}_{im}$, and evaluated using basis functions $B_{\mathbb{J}_m i}$ parameterized by $\widehat{\boldsymbol{\theta}}_{im}$.

(c) Optimize the contribution of the tree

$$\rho_m = \arg\min_{\rho} \sum_{t=1}^{T} [u_{tm} - \rho \hat{u}_{tm}]^2.$$

This step involves finding the optimal scaling factor ρ_m that minimizes the squared error between the actual residuals and the scaled predictions from the current tree.

(d) Update the model

$$\phi_{tm} = \phi_{tm-1} + v\rho_m \hat{u}_{tm}.$$

The predictions are updated by adding a fraction (controlled by the shrinkage parameter v of the scaled tree predictions to the previous predictions.

3. Final Model:

The final prediction for a future value $\hat{Y}_{t+h}$ is given by

$$\begin{aligned} \hat{Y}_{t+h} &= \bar{Y} + \sum_{m=1}^{M} v\rho_m \hat{u}_{tm} \\ &= \bar{Y} + \sum_{m=1}^{M} v\hat{\rho}_m \sum_{k\in\mathbb{T}_m} \hat{\beta}_{km} B_{\mathbb{J}_m k}(\boldsymbol{X}_t; \hat{\boldsymbol{\theta}}_{km}) \end{aligned} \tag{8}$$

This expression represents the initial average plus the accumulated contributions of all trees, adjusted by their respective optimal scaling ρ_m and the shrinkage factor v.

This sequential building and updating process is designed to iteratively reduce errors in the predictions, using trees to model residuals at each step and adjusting the contribution of each tree to prevent overfitting.

Figure 4: R code of the Boosting method

```
# Define hyperparameter grid for GBM
gbmGrid <-  expand.grid(n.trees = 100,
                        interaction.depth = c(1, 3, 5),
                        shrinkage = c(0.01, 0.1),
                        n.minobsinnode = c(10, 20))

# Train GBM model using the selected features
gbm_model <- train(x_selected, y,
                   method = "gbm",
                   trControl = trainControl(method = "cv", number = 5),
                   tuneGrid = gbmGrid)
```

Source: based on Max (2008)

In the case of Boosting, we specify additional values for the hyperparameters, either for computational efficiency or based on theoretical considerations. We continue to utilize the "train" function from the caret package of Max (2008). Initially, we set the shrinkage parameter to either 0.01 or 0.1, following the recommendations of Masini et al. (2021) to strike a balance between fit quality and computational expense. Furthermore, we determine the number of trees to grow ("n.trees"), and we set potential values for interaction depth as well as for the minimum number of observations required in each node ("n.minobsinnode"). Subsequently, we employ 5-fold cross-validation to compute the remaining hyperparameters at each iteration of the rolling window process and to evaluate our specified values, ultimately selecting the best-performing model.

3.3.4 Bagging

Breiman (1996) introduces the concept of bagging, short for Bootstrap Aggregating, Bagging is a method to decrease the variance of unstable predictors. It gained traction within the time series analysis community through the work of Inoue and Kilian (2008). They applied it to generate predictions using multiple regression models that have nearly zero regression coefficients and potentially correlated or conditionally heteroscedastic errors. In the context of time series, the bagging algorithm must consider the temporal aspect when creating bootstrap samples. Bagging is particularly useful when the predictor count is large compared to the sample size, as in our case. Masini et al. (2021) describes the bagging algorithm for times series models as follows:

1. Organize the set of tuples $(y_{t+h}, \boldsymbol{x}_t'), t = h+1, \dots, T$ into a matrix $\boldsymbol{V}$ which has dimensions $(T-h) \times n$. This matrix contains the target variable y_{t+h} and the predictors $\boldsymbol{x}_t'$ from each corresponding time period.

2. Generate bootstrap samples. For $i = 1, \dots, B$ (where B is the number of bootstrap samples we create), draw blocks of M rows from matrix $\boldsymbol{V}$ with replacement. These blocks form the bootstrap samples $\{(y^*_{(i)2}, \boldsymbol{x}'^*_{(i)2}), \dots, (y^*_{(i)T}, \boldsymbol{x}'^*_{(i)T})\}$.

3. Compute Bootstrap Forecasts. For each bootstrap sample, calculate the forecast $\hat{y}^*_{(i)t+h|t}$ as

$$\hat{y}^*_{(i)t+h|t} = \begin{cases} 0 & \text{if } |t^*_j| < c \forall j, \\ \hat{\lambda}^*_{(i)} \tilde{\boldsymbol{x}}^*_{(i)t} & \text{otherwise,} \end{cases}, \tag{9}$$

where $\tilde{\boldsymbol{x}}^*_{(i)t} := \boldsymbol{S}^*_{(i)t} \boldsymbol{z}^*_{(i)t}$ and $\boldsymbol{S}_t$ is a diagonal selection matrix with the j-th diagonal element given by

$$\mathbb{I}_{\{|t_j| > c\}} = \begin{cases} 1 & \text{if } |t_j| > c, \\ 0 & \text{otherwise} \end{cases}.$$

Here c is a critical value predefined in our analysis, and $\hat{\lambda}^*_{(i)}$ is the Ordinary Least Squares (OLS) estimator recalculated for each bootstrap sample.

4. Average the forecasts. Finally, compute the average forecast $\tilde{y}_{t+h|t}$ over all bootstrap samples as

$$\tilde{y}_{t+h|t} = \frac{1}{B}\sum_{i=1}^{B} \hat{y}^{*}_{(i)t|t-1}.$$

This structured approach iteratively assesses the robustness and stability of the model predictions by using the block bootstrap method, accounting for temporal dependencies and potential heteroscedasticity in the time series data.

The bagging algorithm outlined above necessitates the ability to estimate and analyze in a linear model. However, this becomes impractical when the number of predictors exceeds the sample size ($n > T$), necessitating modifications to the algorithm. Garcia et al. (2017) and Medeiros et al. (2019) implement the following modifications to the algorithm:

0. Perform n univariate regressions of y_{t+h} on each covariate in x_t. Calculate the t-statistics and retain only those that are significant at a predetermined level. Label this new set of regressors as $\check{\boldsymbol{x}}_t$.

1-4. Follow the same steps as previously, but replace $\boldsymbol{x}_t$ with $\check{\boldsymbol{x}}_t$.

3.4 Regularized Least Squares

Regularized Least Squares (RLS) encompasses penalized regression methods. Here we consider the Lasso, Ridge and Elastic Net. Before we dive deep into the methods, we outline our notation. This subchapter is based on the work of Kock et al. (2020).

For any vector $x \in \mathbb{R}^n$, the ℓ_2-norm of $\boldsymbol{x}$ is defined as $\| x \| = \sqrt{\sum_{i=1}^{n} x_i^2}$, which represents the Euclidean norm. The ℓ_1-norm of x, denoted as $\| x \|_{\ell_1}$, is the sum of the absolute values of the entries of x, or $\sum_{i=1}^{n} |x_i|$. The ℓ_0-"norm" of $\boldsymbol{x}$, although technically not a norm, counts the number of non-zero elements in x, given by $\| x \|_{\ell_0} = \sum_{i=1}^{n} 1_{\{x_i \neq 0\}}$.

For any subset $A \subseteq \mathbb{R}^n$, the cardinality of A is represented as $|A|$. The vector $\boldsymbol{x}_A$ is derived from $\boldsymbol{x}$ and contains only those entries of $\boldsymbol{x}$ whose indices are in A, resulting in a vector of length $| \overline{A} |$. Similarly, for an $n \times n$ matrix $\boldsymbol{B}$ denotes the submatrix consisting of rows and columns indexed by A, and is thus of dimension $|A| \times |A|$.

Next, this chapter explores variants of the classical linear regression model, written as

$$y_t = \beta' x_t + \epsilon_t,\ t = 1, \ldots, T, \tag{10}$$

where $T \in \mathbb{N}$ and β is a $k \times 1$ vector of unknown parameters. The subscript t indicates that the observations are sequential, reflecting the time series nature of the data. Furthermore, β^0 denotes the true parameter vector, which is assumed to be unique within the context of this chapter. Define $\mathcal{A} = \{i: \beta_i^0 \neq 0\}$. The cardinality of $\mathcal{A}$, denoted $s_0 = |\mathcal{A}|$, reflects the number of non-zero coefficients. When s_0 is substantially smaller than k, β^0 is described as sparse. This sparsity is crucial for certain statistical methods and interpretations within time series analysis.

Consider the equation $\boldsymbol{y} = \boldsymbol{X\beta} + \boldsymbol{\epsilon}$ represented in the standard matrix format, where $\boldsymbol{Z} = (\boldsymbol{y}, \boldsymbol{X})$. A penalized regression estimator $\hat{\boldsymbol{\beta}}$ is derived by solving the following optimization problem:

$$\widehat{\boldsymbol{\beta}} \in \underset{\boldsymbol{\beta} \in \mathbb{R}^k}{\operatorname{argmin}} [\| \boldsymbol{y} - \boldsymbol{X\beta} \|^2 + \lambda p(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{Z})]. \tag{11}$$

Here, $\lambda \geq 0$ acts as a penalty or tuning parameter, and $\mathrm{p}: \mathbb{R}^k \times \mathbb{R}^d \times \mathbb{R}^{T \times (1+k)} \to [0, \infty)$ is the penalty function, which imposes a cost on the elements of $\boldsymbol{\beta}$ for being non-zero. The parameter $\boldsymbol{\alpha}$ is a d-dimensional tuning parameter chosen by the user. Although the penalty function p generally does not depend on any tuning parameter, it can be influenced by the observed data Z.

In all of the following forms of p, for any $(\boldsymbol{\alpha}, \boldsymbol{B}) \in \mathbb{R}^d \times \mathbb{R}^{T \times (1+k)}$, $p(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{B}) = 0$ if and only if $\boldsymbol{\beta} = 0$. This condition means p penalizes $\boldsymbol{\beta}$ when it is not zero, thereby the estimator $\hat{\boldsymbol{\beta}}$ from equation (11) is termed penalized as it minimizes both the regular least squares objective and the penalty function p. The value of λ, when increased, assigns more importance to the penalty function, leading to $\hat{\boldsymbol{\beta}}$ having generally smaller (in absolute terms) entries compared to the least squares estimator. Moreover, Kock et al. (2020) emphasize that the penalties discussed lead to estimators that are viable even when $k > T$.

3.4.1 Ridge

The first penalty term, Ridge, is presented in Hoerl and Kennard (1970) and concisely described in Kock et al. (2020). It uses

$$\mathrm{p}(\boldsymbol{\beta}, \alpha, \boldsymbol{Z}) = \sum_{i=1}^{k} \beta_i^2. \tag{12}$$

The Ridge Regression estimator, $\hat{\boldsymbol{\beta}}_{\text{ridge}}(\lambda)$, is derived from the closed-form solution $\hat{\boldsymbol{\beta}}_{\text{ridge}}(\lambda) = (\boldsymbol{X}'\boldsymbol{X} + \lambda \boldsymbol{I}_k)^{-1}$. Note that $\hat{\boldsymbol{\beta}}_{\text{ridge}}(0) = \hat{\boldsymbol{\beta}}_{\text{OLS}}$ when well-defined. Ridge Regression proves particularly useful when $X'X$ is (nearly) singular, where the ordinary least squares (OLS) estimator faces issues such as non-uniqueness and high variance. Although unbiased in cross-sectional data, the high variance of the OLS estimator leads to a significant mean square error (MSE). In contrast, $\hat{\beta}_{\text{ridge}}$ is biased for any $\lambda > 0$; however, there is always some $\lambda > 0$ for which the MSE of $\hat{\beta}_{\text{ridge}}(\lambda)$ is strictly lower than that of the OLS estimator, as detailed in Hoerl and Kennard (1970). This lower variance underpins the effectiveness of Ridge Regression even in time series analysis. However, it's important to note that although Ridge Regression reduces the magnitude of the parameter estimates (i.e., $\|\hat{\boldsymbol{\beta}}_{\text{ridge}}(\lambda)\|^2 < \|\hat{\boldsymbol{\beta}}_{OLS}\|^2$ for $\lambda > 0$), it does not produce coefficients that are exactly zero, thus limiting its utility for variable selection.

Figure 5 illustrates the R code for implementing the Ridge regression method. We optimize the parameter values for λ as suggested by Friedman et al. (2010), while setting $\alpha = 0$ to configure a Ridge model. We proceed with the model training using 5-fold cross-validation and utilize the "train" function from the care package of Max (2008) to select the optimal λ during each iteration of the rolling window process. Upon determining the best λ value, we estimate the Ridge model using the "glmnet" function from the glmnet package by Friedman et al. (2010). Subsequently, we use the estimated model for predictions and calculate the forecast errors.

Figure 5: R code of the Ridge method

```
# Set up the parameter grid for lambda (alpha is fixed to 0 for Ridge)
lambda_grid <- 10^seq(-3, 1, length.out = 100)
param_grid <- expand.grid(alpha = 0, lambda = lambda_grid)

# Perform nested cross-validation
inner_cv <- trainControl(method = "cv", number = 5)
outer_cv <- trainControl(method = "cv", number = 5)

model <- train(x, y, method = "glmnet", trControl = inner_cv, tuneGrid = param_grid)

# Retrieve the optimal hyperparameters
optimal_lambda[i] <- model$bestTune$lambda

# Make predictions on the test data using the optimal hyperparameters
ridge_model <- glmnet(x, y, family = "gaussian", alpha = 0, lambda = optimal_lambda[i])
```

Source: based on Friedman (2010) and Max (2008)

3.4.2 Least Absolute Shrinkage and Selection Operator (LASSO)

The second penalty term is Lasso, introduced by Tibshirani (1996) and briefly outlined in Kock et al. (2020). It uses

$$\mathrm{p}(\boldsymbol{\beta}, \alpha, \boldsymbol{Z}) = \sum_{i=1}^{k} |\beta_i|, \tag{13}$$

within the framework described by equation (11). This contrasts with Ridge Regression, which employs the squared ℓ_2-norm; the Lasso uses the ℓ_1-norm. A distinctive feature of the Lasso is its ability to produce coefficient estimates that are exactly zero when the penalty parameter λ is sufficiently large, effectively performing variable selection.

The Lasso combines estimation and variable selection in a single step, starkly contrasting traditional methods which typically estimate parameters first and then conduct hypothesis tests to determine which parameters are non-zero. Traditional model testing procedures heavily depend on the sequence and type of tests conducted, such as sequential t-tests or a combination of joint and individual tests, leading to final models that vary based on the testing approach. Alternatives like using information criteria (AIC, BIC, HQ) for model selection also exist. Both of these variable selection methods are generally applicable only to least squares estimates for $k < T$ and become computationally burdensome as k increases. The Lasso, therefore, is particularly popular in high-dimensional settings where the number of predictors $k < T$.

With regards to inference, van de Geer et al. (2014) introduces the desparsified LASSO to create (asymptotically) valid confidence intervals for each $\beta_{j,0}$ by adjusting the original LASSO estimate $\widehat{\boldsymbol{\beta}}$. Van de Geer et al. (2014), however, do not explore time series analysis. Adámek et al. (2020) further develop the foundational work by van de Geer et al. (2014), applying it to time-series models. They adapt the desparsified LASSO for time series analysis, even when the number of regressors may increase faster than the sample size.

The code for optimizing Lasso, as illustrated in Figure 5, remains identical except for the modification where α is set to 1. This adjustment configures the model for Lasso regression, which enables variable selection by shrinking certain parameter coefficients to zero.

3.4.3 Elastic Net

The third penalty term, introduced by Zou and Hastie (2005), is the Elastic Net. The method is briefly characterized in Kock et al. (2020). The penalty function is

$$\mathrm{p}(\boldsymbol{\beta}, \boldsymbol{\alpha}, \boldsymbol{Z}) = \alpha \sum_{i=1}^{k} \beta_i^2 + (1-\alpha) \sum_{i=1}^{k} |\beta_i| \quad (14)$$

for $\alpha \in [0,1]$. This penalty represents a convex combination of the Ridge Regression penalty ($\alpha = 1$) and the Lasso penalty ($\alpha = 0$), effectively harnessing the benefits of both approaches. Particularly in scenarios where there is high correlation among explanatory variables—common in time series applications—the Lasso tends to select only one variable from a group of correlated variables. This selection might not be ideal depending on the specific application requirements. For instance, if the objective is to consider all relevant variables within a highly correlated group, the elastic net becomes preferable as it tends to include multiple correlated variables rather than selecting just one. This makes the elastic net particularly useful in applications where the representation of all related variables is critical.

Figure 6 presents the R code for the Elastic Net method. This code is nearly identical to that shown in Figure 5, with the key distinction being the optimization of both λ and α at each iteration. This dual optimization ensures that we assign the optimal weight to both the squared ℓ_2-norm (Ridge) and the ℓ_1-norm (Lasso) at each iteration, effectively balancing shrinkage and variable selection.

Figure 6: R code of the Elastic Net method

```
# Set up the parameter grid for alpha and lambda
alpha_grid <- seq(0, 1, by = 0.1)
lambda_grid <- 10^seq(-3, 1, length.out = 100)
param_grid <- expand.grid(alpha = alpha_grid, lambda = lambda_grid)

# Perform nested cross-validation
inner_cv <- trainControl(method = "cv", number = 5)
outer_cv <- trainControl(method = "cv", number = 5)

model <- train(x, y, method = "glmnet", trControl = inner_cv, tuneGrid = param_grid)

# Retrieve the optimal hyperparameters
optimal_alpha[i] <- model$bestTune$alpha
optimal_lambda[i] <- model$bestTune$lambda

# Make predictions on the test data using the optimal hyperparameters
elastic_net_model <- glmnet(x, y, family = "gaussian", alpha = optimal_alpha[i], lambda = optimal_lambda[i])
```

Source: based on Friedman (2010) and Max (2008)

3.4.4 Selection of penalty parameters

Kock et al. (2020) show an example of cross-validation applied to RLS models. Cross-validation (CV) is one of the most commonly used methods for selecting models and variables. In penalized regressions, CV is utilized to choose the penalty parameters. This technique involves dividing the sample into two separate groups: a training set (known as "in-sample") and a validation set (referred to as "out-of-sample"). The model parameters are determined only

using the data from the training set, and the model's performance is then evaluated on the validation set. Consider Λ and A as the sets of possible values for the parameters λ and α, respectively. Also, let $\mathrm{V} \subseteq \{1, \dots, T\}$ be the indices for the validation set observations, and $\mathrm{T} \subseteq \{1, \dots, T\}$ the indices for the training set observations. Typically, but not always, the training set indices are the complement of the validation set indices as $\mathrm{T} := \mathrm{V}^c$. The estimated parameter, $\hat{\beta}_{\mathrm{T}}(\lambda, \boldsymbol{\alpha})$, is calculated from the training data T for each pair of tuning parameters $(\lambda, \alpha) \in \Lambda \times A$.

For each pair of tuning parameters (λ, α), the function

$$CV(\lambda, \alpha, \mathrm{V}) = \sum_{t \in \mathrm{V}} \left(y_t - x_t' \hat{\beta}_{\mathrm{T}}(\lambda, \alpha) \right)^2$$

represents the summation of squared prediction errors over the validation set V, where y_t and x_t' are the observed outcomes and predictor values, respectively, and $\hat{\beta}_{\mathrm{T}}(\lambda, \alpha)$ is the parameter estimate from the training set. The variable V refers to a pre-defined collection of validation sets, denoted as $V = \{\mathrm{V}_1, \dots, \mathrm{V}_B\}$, each paired with corresponding training sets $\{\mathrm{T}_1, \dots, \mathrm{T}_B\}$.

The cross-validation error for a given combination of tuning parameters (λ, α) is computed as

$$CV(\lambda, \alpha) = \sum_{i=1}^{B} CV(\lambda, \alpha, \mathrm{V}_i),$$

where each $CV(\lambda, \alpha, \mathrm{V}_i)$ represents the prediction error for a specific validation set V_i. To identify the optimal set of tuning parameters, one selects

$$(\hat{\lambda}, \hat{\alpha}) \in \underset{(\lambda, \alpha) \in \Lambda \times A}{\operatorname{argmin}} CV(\lambda, \alpha).$$

The final estimate of the parameters, $\hat{\beta}(\hat{\lambda}, \hat{\alpha})$ is then calculated using all observations $1, \dots, T$. The determination of B and the corresponding sets $\{\mathrm{V}_1, \dots, \mathrm{V}_B\}$ involves choosing between two primary types of cross-validation (CV) methods: exhaustive and non-exhaustive.

In the exhaustive category, the most prevalent method is the leave-v-out CV. This approach involves using v observations as the validation set and the remaining observations to estimate the model parameters. This process is repeated for all possible combinations of v observations chosen from the total T observations. Consequently, $B = \binom{T}{v}$, where each V_i contains exactly v observations, and $\mathrm{T}_i = \mathrm{V}_i^c, i = 1, \dots, B$. A frequently used configuration within this framework is the leave-one-out CV, where $v = 1$. This results in $B = T$, and each

validation set $\mathrm{V_i} = \{i\}$ consists of a single observation indexed by i. This method is especially thorough as it tests each data point individually as a validation set, ensuring that the model is robustly validated across all data points.

Non-exhaustive CV methods offer a practical solution to the high computational demands of exhaustive CV by reducing the number of sample splits. One of the most common non-exhaustive methods is B -fold cross-validation. In this approach, the sample is divided into B subsamples, or "folds," each containing approximately the same number of observations. This partitioning is typically done randomly to ensure that each subset is representative of the whole. In B -fold CV, the B validation groups $\mathrm{V}_1, \ldots, \mathrm{V}_B$ are disjoint, meaning each observation is included in exactly one validation group. Consequently, the training sets are defined as $\mathrm{T}_i = \mathrm{V}_i^c, i = 1, \ldots, B$. This setup allows for each subsample to be used as a validation set once, while the remaining $B - 1$ subsamples are used as the training set. The results from each fold are then averaged to produce a final model estimate. Common choices for B are 5 or 10, balancing between computational efficiency and the reliability of the validation process. This method is widely used due to its robustness and relatively lower computational load compared to exhaustive methods. In this thesis we use 5-fold cross-validation for both linear and nonlinear models.

3.5 Combined RLS and EML methods

One of the main methodological contributions of this thesis is the proposal of machine learning method combination. To be more precise, we use RLS methods for data preparation and then apply nonlinear methods to the pre-selected or shrinked variables.

According to our knowledge, the only other paper that proposes something similar is Medeiros et al. (2019). Their methodology involves two distinct specifications that integrate Random Forests, adaLASSO (adaptive Least Absolute Shrinkage and Selection Operator), and Ordinary Least Squares (OLS) regression.

Their first specification begins with the use of a Random Forest to select relevant variables from a dataset. Random Forests, being robust and capable of handling nonlinearities and interactions between variables, effectively determine which variables are most influential. The variables selected through this process are then used in a traditional OLS regression. The

OLS model, known for its simplicity and interpretability, provides a linear relationship between these selected variables and the target variable, in this case, inflation.

Their second specification starts with variable selection using adaLASSO, a modification of the LASSO technique that adapts the penalty terms based on preliminary estimates of the coefficients. This method is particularly effective in reducing the bias in variable selection, promoting a more refined set of predictors. Following the selection of variables through adaLASSO, these are then utilized within a Random Forest model. The idea here is to explore how the selected variables perform in a nonlinear modeling context, provided by Random Forest, thus assessing the impact of including or excluding certain predictors.

The overarching goal of both methodologies is to dissect and understand the roles of variable selection and the handling of nonlinear relationships in economic forecasting. By experimenting with these hybrid approaches, Medeiros et al. (2019) aim to ascertain which aspects—variable selection or the ability to model nonlinear dynamics—play a more crucial role in accurately forecasting inflation.

The results of Medeiros et al. (2019) show that they cannot improve upon a simple Random Forest benchmark, which still performs best in their dataset. In contrast to their methodology and results, we employ 9 different specifications (combining each RLS and EML method with the other) and show that the combination of these methods can lead to substantial improvements in forecasting accuracy.

Figure 7: R code of the combined Elastic Net and Random Forest methods

```
# Retrieve the optimal hyperparameters
optimal_alpha <- model$bestTune$alpha
optimal_lambda <- model$bestTune$lambda

# Fit the Elastic Net model with optimal hyperparameters
elastic_net_model <- glmnet(x, y, family = "gaussian", alpha = optimal_alpha, lambda = optimal_lambda)

# Extract the coefficients and identify the non-zero features
enet_coefs <- coef(elastic_net_model, s = optimal_lambda)
selected_features <- which(enet_coefs != 0)

# Filter the features for the Random Forest model
x_selected <- x[, selected_features, drop = FALSE]


# Train the Random Forest model using selected features
limited_mtry <- seq(1, min(60, ncol(x_selected)), by = 1)

rf_model <- train(x_selected, y,
                  method = "rf",
                  trControl = trainControl(method = "cv", number = 5),
                  tuneGrid = expand.grid(mtry = limited_mtry))
```

Source: based on Friedman (2010) and Max (2008)

Figure 7 presents an example of a hybrid specification combining the Elastic Net and Random Forest methods. The initial part of the Elastic Net code mirrors that shown in Figure 6. At each iteration, we select the optimal λ and α values using 5-fold cross-validation. We then estimate the Elastic Net model and select variables corresponding to non-zero coefficients. These selected variables are subsequently used in training the Random Forest model, also through 5-fold cross-validation. Finally, we use the Random Forest model, which has been trained on the variables identified by the Elastic Net, for forecasting. This approach leverages the strengths of both methods—variable selection or shrinkage from Elastic Net and the predictive power of Random Forest.

3.6 Neural Networks

In this subchapter, we introduce the Feedforward and Long Short -Term Memory (LSTM) neural networks that we employ, starting with the former. LeCun et al. (2015) provide a detailed examination of the Feedforward Neural Network (FFNN), offering a comprehensive review of neural network theory. The FFNN is described as a predictive model featuring a single hidden layer, written as

$$y_{t+h} = f(\boldsymbol{X}_t) + \varepsilon_{t+h}, \tag{15}$$

where

$$f(X_t) = \sum_{j=1}^{q} \boldsymbol{\theta}_j \sigma\left(w_j' \boldsymbol{X}_t + b_j\right) + b. \tag{16}$$

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 $(\boldsymbol{\theta}_j, w_j', b_j, b)$ for $j = 1, \ldots, q$ by minimizing the least square criterion.

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

$$\sigma(z) = \frac{1}{1+e^{-z}}, \tag{17}$$

and the ReLU (rectified linear unit) function written as

$$\sigma(z) = \begin{cases} 0 & \text{if } z < 0 \\ z & \text{otherwise} \end{cases}, \tag{18}$$

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). The geometric pyramid rule of Masters (1993) is a method used in the training of neural networks. The rule suggests a geometric progression in the size of each layer of the network. Typically, this means that each successive hidden layer in a neural network should have fewer neurons than the one preceding it, forming a pyramid shape if you were to graph the number of neurons per layer. This can aid in preventing overfitting and improving the generalization capabilities of the network. By systematically reducing the number of neurons, the network is forced to capture the most essential features in the early layers, which can be more effective for learning complex patterns with fewer parameters.

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.

Continuing, we introduce the LSTM network, a variant of Recurrent Neural Network (RNN). The forecasting model of a recurrent neural network with one hidden layer can be formulated as

$$y_{t+h} = \sum_{j=1}^{q} \theta_j h_{jt} + b + \varepsilon_{t+h}, \tag{19}$$

where

$$h_{jt} = \sigma\left(w_j' X_t + \sum_{k=1}^{q} \theta_{jk} h_{kt-1} + b_j\right), \tag{20}$$

where b stands for bias, $h_{1t}, h_{2t}, \dots, h_{qt}$ 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 receives inputs from both the preceding layer and from its own outputs in the previous time step, via an internal loop referenced in (17). This structure allows it to capture the serial dependencies inherent in time series data. As Hochreiter and Schmidhuber (1997) noted, this type of simple recurrent neural network is prone to the exploding or vanishing gradient problem, which complicates the learning of long-term dependencies. To address this issue, a specific version of the RNN, such as the LSTM, is necessary to effectively capture and maintain these long-term dependencies within the data.

Hochreiter and Schmidhuber (1997) introduce the Long Short-Term Memory (LSTM) network, a specialized form of the Recurrent Neural Network (RNN) that is particularly prevalent in the field of macroeconomic forecasting. The LSTM network incorporates a unique component known as the memory cell within its RNN framework. This memory cell is crucial for capturing long term information, as it primarily relies on the memory from the previous time step. Additionally, the LSTM network includes two essential gating mechanisms: the input gate and the forget gate. The input gate regulates the entry of new information into the memory cell, whereas the forget gate determines when to remove outdated information. These features make the LSTM network exceptionally suitable for analyzing time series data due to its capability to maintain relevant historical information over extended periods, thereby enhancing its predictive accuracy.

Because our computer resources are limited and we don't have access to advanced computing, we cannot use grid search[5] to choose the best settings for our neural network. Instead, we follow the advice of Maehashi and Shintani (2020) and manually set the number of parts in the network to 10. For the parts of the network that need activation functions, we use the hyperbolic tangent function, which is a common choice. For the special parts of the network called gates, we use the sigmoid function. This setup helps us manage our limited resources while still following established methods in the field.

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 4.

Figure 8: R code of the LSTM neural network

```
# Define LSTM neural network model
model2 <- keras_model_sequential()
model2 %>%
  layer_dense(units = 32, activation = "relu", input_shape = c(window_size * (ncol(data)-1))) %>%
  layer_dense(units = 16, activation = "relu") %>%
  layer_dense(units = 1)  # Output layer
model2 %>% compile(optimizer = "adam", loss = "mean_squared_error")

# Train the LSTM model
model2 %>% fit(X_train, y_train, epochs = 50, batch_size = 1, verbose = 0)
```

Source: based on "keras" and "tensorflow" packages in R

Figure 8 depicts the code snippet for an LSTM network. We must predefine all critical hyperparameters, including the activation function, optimizer, batch size, and epochs. The section titled "Train the LSTM model" is normally part of a rolling window loop; however, it

[5] Grid search for neural networks involves systematically exploring multiple combinations of model parameters to determine which configuration performs best for a given task.

is presented separately here to illustrate its functional form in R. Ideally, all previously mentioned hyperparameters should be optimized within the forecasting loop of the rolling window scheme to enhance model performance and accuracy in predictions, but this comes at a high computational cost.

3.7 Support Vector Machine

Vapnik (1995) presents the Support Vector Machine (SVM) method and Fiszeder and Orzeszko (2021) provide a brief review. This subchapter is based on the latter paper. Consider the regression model defined as

$$y = r(\mathbf{x}) + \delta \tag{21}$$

Here, $r(\mathbf{x})$ represents the regression function, y is the dependent variable, $\mathbf{x}$ is the set of predictors, and δ is zero-mean noise with a variance of σ^2. Using a training dataset $\{(\mathbf{x}_t, y_t)\}_{t=1,\dots T}$, our goal is to estimate the unknown regression function with a function $f(\mathbf{x})$, such that $f(\mathbf{x})$ varies from the actual outputs y_t by no more than ε and is as smooth as possible.

In SVM, the input $\mathbf{x}$ is initially transformed into a high-dimensional feature space through a fixed (nonlinear) mapping. Following this transformation, a linear model is established within this new feature space, written as

$$f(\mathbf{x}) = \sum_{i=1}^{d} \omega_i \varphi_i(\mathbf{x}) + b. \tag{22}$$

The dimension d of the feature space, along with functions $\varphi_i(\mathbf{x})$ representing nonlinear transformations, coefficients ω_i, and a bias term b are crucial components. The $\varphi_i(\mathbf{x})$ functions transform the input $\mathbf{x}$ into the feature space. It's important to highlight that the capacity of the SVM model to approximate a smooth input-output mapping is influenced by the dimension d of this space. Essentially, a larger value of d results in a more precise approximation.

According to (22), to determine the function $f(\mathbf{x})$, one needs to estimate the coefficients $\boldsymbol{\omega} = (\omega_1, \omega_2, \dots, \omega_d)'$ and the bias term b. To assess the quality of this estimation, Vapnik (1995) proposes the usage of the ε-insensitive loss function of the form

$$L_\varepsilon(y, f(\mathbf{x})) = \begin{cases} 0, & |y - f(\mathbf{x})| \le \varepsilon, \\ |y - f(\mathbf{x})| - \varepsilon, & \text{otherwise} \end{cases}. \tag{23}$$

This function implies that errors smaller than ε incur no penalty. SVM employs this ε-insensitive loss function to perform linear regression in the d-dimensional feature space, while simultaneously striving to reduce model complexity by minimizing the squared norm of $\boldsymbol{\omega}$, represented as $\| \boldsymbol{\omega} \|^2 = \boldsymbol{\omega}'\boldsymbol{\omega}$.

The optimal regression function in SVM is identified by minimizing the functional

$$\Phi(\boldsymbol{\omega}, \boldsymbol{\xi}) = \frac{1}{2} \| \boldsymbol{\omega} \|^2 + C \sum_{t=1}^{n} (\xi_t + \xi_t^*), \tag{24}$$

where C is a predetermined positive constant and ξ_t and ξ_t^* are nonnegative slack variables. These variables represent the upper and lower bounds on the deviations from the predicted outputs relative to the actual outputs expressed as

$$\begin{matrix} y_t - f(\mathbf{x}_t) \leq \varepsilon + \xi_t^* \\ f(\mathbf{x}_t) - y_t \leq \varepsilon + \xi_t \end{matrix}, \tag{25}$$

for all $t = 1,2, \ldots, T$. The parameter C serves to control the penalty applied to observations that fall outside the ε-margin, which helps in mitigating the risk of overfitting. The values for both ε and C are set by the user, influencing the model's complexity and its sensitivity to deviations beyond the ε-threshold.

The optimization problem in SVM can be converted into a dual problem, where the solution is characterized by

$$f(\mathbf{x}) = \sum_{t=1}^{T_{SV}} (\alpha_t - \alpha_t^*) K(\mathbf{x}_t, \mathbf{x}) \text{ s.t. } 0 \leq \alpha_t \leq C, 0 \leq \alpha_t^* \leq C. \tag{26}$$

Here, α_t and α_t^* are Lagrange multipliers, T_{SV} denotes the number of support vectors, and K is the kernel function, which is defined as

$$K(\mathbf{x}_t, \mathbf{x}) = \sum_{i=1}^{d} \varphi_i(\mathbf{x}) \varphi_i(\mathbf{x}_t). \tag{27}$$

The kernel function K enables the computation of the inner product in the feature space without directly calculating the potentially complex and high-dimensional transformations φ_i. This approach simplifies the mathematical operations by replacing the explicit computation of the feature vectors $\varphi(\mathbf{x}) = (\varphi_1(\mathbf{x}), \varphi_2(\mathbf{x}), \ldots, \varphi_d(\mathbf{x}))'$ with the kernel function K, which efficiently captures the necessary dimensional interactions. This transformation into the dual problem and the use of a kernel function significantly reduce the computational complexity compared to solving the primal problem directly, especially in scenarios involving high-dimensional data spaces.

The most popular kernel functions are:

- Linear (dot product): $K(\mathbf{x}_t, \mathbf{x}) = \mathbf{x}_t'\mathbf{x}$,
- Gaussian / Radial: $K(\mathbf{x}_t, \mathbf{x}) = \exp\left(-\|\mathbf{x}_t - \mathbf{x}\|^2\right)$,
- Polynomial: $K(\mathbf{x}_t, \mathbf{x}) = (1 + \mathbf{x}_t^T\mathbf{x})^p; p = 2,3, \ldots$

out of which the latter two allows for nonlinear SVM computation. In this thesis we use the Gaussian kernel.

Figure 9 illustrates the setup of an SVM (Support Vector Machine) in R using the e1071 package. In this setup, we partially define the parameter grid, conduct cross-validation, optimize the hyperparameters, and estimate the forecasting model using these optimal hyperparameter values.

Figure 9: R code of the SVM method

```
# Define the search space and use correct parameter names for svmRadial
tune_grid <- expand.grid(C = c(0.1, 1, 10), sigma = c(0.1, 1, 10))

# Cross-validation
ctrl <- trainControl(method = "cv", number = 5)

# Hyperparameter optimization
tuned_svm <- train(x, y, method = "svmRadial", tuneGrid = tune_grid, trControl = ctrl)
best_params <- tuned_svm$bestTune

# Train the SVM model using the best hyperparameters
svm_model <- svm(x, y, kernel = "radial", cost = best_params$C, gamma = 1 / (2 * best_params$sigma))
```

Source: based on the e1071 R package

3.8 Forecasting setup

3.8.1 General setup

This subchapter outlines our forecasting setup. We forecast two key economic indicators: industrial production (IP), measured by the Index of Industrial Production, and inflation (Inf), measured by the harmonized index of consumer prices. We project these measures for 5 different periods: 1, 3, 6, 9, and 12 months ahead. The Index of Industrial Production is commonly used as a proxy for GDP, as discussed and applied by Coulombe et al. (2022) and Maehashi and Shintani (2020).

Our dataset starts during the Great Recession, a period marked by significant economic uncertainty and volatility. The early observations from this period could potentially skew the overall dataset. To mitigate this issue, we utilize a rolling window scheme for forecasting,

which allows the parameter estimates to adjust over time. This approach is also adopted by Pratap and Sengupta (2019), Mahajan and Srinivasan (2019), and Maehashi and Shintani (2020) for similar reasons. Recently, McCracken (2020) demonstrate that using a fixed estimation window can cause the Diebold and Mariano (1995) statistic to potentially diverge under the null hypothesis. This finding also underscores the importance of employing a rolling window scheme for forecasting.

Specifically, we maintain a constant window size, always using 6 years of observations for each estimation point during the period of 2008M01–2019M12 and 10 years during the period of 2008M01–2023M12, adjusting the initial point of estimation as needed. Any adjustments to the window size of our estimations lead to consistent conclusions as those presented in Chapters 4 and 5.

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.

Additionally, considering that each machine learning model requires specific parameter selection, we follow the guidance of Coulombe et al. (2022) to counter the overfitting issue that often arises when dealing with large datasets and complex models. To do this, we employ B-fold cross-validation – explained in the Lasso subchapter for RLS methods – a method whose statistical validity is affirmed by Bergmeir (2018). In our implementation, B is set to 5. This means that the in-sample dataset is randomly divided into five distinct subsets, each accounting for approximately 20% of the in-sample observations.

For each of these five subsets, and for every set of hyperparameters under consideration, four subsets are used for model training, while the fifth subset is reserved as a test subset to evaluate forecast accuracy. The mean squared error (MSE) is utilized as the evaluation metric. This approach not only helps in fine-tuning the models but also ensures that the training process is robust against overfitting.

In summary, we use a rolling window scheme to continuously adapt our forecasts to new data, while B-fold cross-validation assists in the careful selection and validation of model hyperparameters, ensuring that our forecasting models are both accurate and generalizable.

3.8.2 Common factors

Moreover, we utilize Principal Component Analysis (PCA), as proposed by Stock and Watson (2002), to extract common factors from our dataset. The purpose of using these common factors is to capture the latent forces that drive the data-generating processes across a large number of variables, and then use a more condensed set of factors as predictors for our forecasting efforts. In this framework, a potential predictor x_{it} is generated by the equation

$$x_{it} = \lambda_i' F_t^k + e_{it}, \tag{28}$$

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

The forecasting process using these common factors involves two main steps. First, we minimize the sum of squared differences $\sum_{i=1}^{N}\sum_{t=1}^{T}\left(x_{it} - \lambda_i' F_t^k\right)^2$ 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. We set the maximum number of principal components to 20. To determine the optimal number of factors to use for each forecast, we apply the Bai and Ng (2002) information criterion:

$$IC(k) = \ln V(k) + k\left(\frac{N+T}{NT}\right)\ln C_{NT}^2, \tag{29}$$

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

Secondly, once the factors are derived, we use them to construct forecasts h-periods ahead utilizing machine learning models. In summary, the PCA-derived common factors serve as critical input variables for the machine learning models, enhancing the robustness and accuracy of our forecasting approach.

3.8.3 Composite forecasts

To complete our discussion on our forecasting setup, we delve into the methods used for combining forecasts. Bates and Granger (1969) highlight that combining individual forecasts can enhance forecast accuracy and potentially yield better outcomes than any single model alone. Clements and Hendry (2011) affirm the benefits of combining forecasts from different models to improve accuracy, though they recognize that in some cases, individual models might outperform combined forecasts. Bürgi (2015) contends that achieving more accurate forecasts through any method other than combining individual models using equal weights is challenging; therefore, he recommends the arithmetic mean for combining forecasts.

Adding a contemporary perspective, Araujo and Gaglianone (2023) explore the use of machine learning techniques for forecast combination. They employ methods like adaLasso and Random Forest, which uniquely use individual forecasts as inputs, rather than traditional predictors. Their findings suggest that in their specific context, using machine learning for combining forecasts results in improvements over the simpler methods of averaging, such as the arithmetic mean or median.

Motivated by these findings, our study also utilizes machine learning methods as forecast combination tools to evaluate whether they can produce more accurate forecast combinations than individual model forecasts. In summary, our approach involves combining individual forecasts using various weighting procedures, including more complex machine learning-driven combinations, to identify the most effective method in enhancing forecast accuracy.

3.9 Performance evaluation

We evaluate the forecasting performance of our models using literature standard measures, namely the Mean Squared Error (MSE), Mean Absolute Error (MAE), and Directional Accuracy (DA). To facilitate model comparisons, we relativize these metrics by dividing the metric of the evaluated model by that of the benchmark model, normalizing the benchmark to 1, which simplifies comparisons.

The Mean Absolute Error is defined as

$$MAE = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}. \tag{30}$$

This formula calculates the average of the absolute differences between the forecasted values and the observed values, providing a measure of prediction accuracy that is not sensitive to extreme values.

The Mean Squared Error is expressed as

$$MSE = \frac{\sum_{i=1}^{N} (x_i - \hat{x}_i)^2}{N}. \tag{31}$$

This metric computes the quadratic mean of the forecast errors. Using both MSE and MAE is crucial. MSE is a common standard in machine learning as most models are trained to minimize this or the Root Mean Squared Error (RMSE), and MAE provides a robustness check. Consistent results across these metrics confirm that the model's accuracy is not solely influenced by a few large errors, ensuring the reliability of the model's predictive performance.

In addition to the classical measures of forecasting performance such as Mean Squared Error (MSE) and Mean Absolute Error (MAE), we also incorporate Directional Accuracy (DA), a metric ignored so far in the machine learning forecasting literature. We argue that this omission is problematic because accurately predicting the direction of change in a variable is as critical as the magnitude of errors. A model that fails to capture the correct trend direction essentially provides incorrect forecasts, regardless of how small the error might be. Constantini et al. (2016) write Directional Accuracy as follows

$$DA_{th} = I\big(sgn(y_t - y_{t-h}) = sgn(\hat{y}_t - y_{t-h})\big), \tag{32}$$

where $I_{(.)}$ is the indicator function that equals 1 if the signs of the actual change $(y_t - y_{t-h})$ and the forecasted change $(\hat{y}_t - y_{t-h})$ are the same, and 0 otherwise. This metric assesses whether the forecasted values are moving in the same direction as the actual values, thereby evaluating the model's ability to capture trends in the data effectively. Using DA along with MSE and MAE provides a comprehensive view of a method's performance, assessing its accuracy in both magnitude and direction of predicted changes. This multi-faceted evaluation helps in ensuring that the forecasts are not only close in value to the actuals but also aligned in their directional movement.

To conclude our evaluation, we compare the Mean Squared Error (MSE) of each model against our benchmark model. A model is considered to outperform the benchmark if it has a lower MSE. To determine whether this improvement is statistically significant, we employ the

Diebold-Mariano (DM) test statistic, formulated by Diebold and Mariano (2002), which is designed to compare forecasting accuracies. The *DM*-statistic is calculated as

$$DM = \frac{\bar{d}}{\sqrt{\frac{\hat{\sigma}_{\bar{d}}^2}{T}}} \tag{33}$$

Here, $\bar{d}$ represents the sample mean of the loss differentials between the two forecasts over T periods. The term $\hat{\sigma}_{\bar{d}}^2$ is an estimate of the variance of these loss differentials, which is often calculated using Newey-West standard errors to adjust for autocorrelation and heteroskedasticity. In this thesis we present the *p*-values of this test to determine the statistical significance of our results. We test whether our forecasts are better against the null hypothesis that the two forecasts from two different models have the same accuracy. If $p \leq 0.05$, we reject the null.

The DM test is flexible in that it can be applied regardless of the specific loss function used to measure forecast errors, and it adjusts for potential issues like autocorrelation in the forecast error series.

By applying the DM test, we ensure that observed improvements in MSE are not due to random fluctuations in the data but are statistically significant, reflecting true advancements in forecasting capability.

3.10 Data

3.10.1 Data preparation and preprocessing

In this subchapter we describe the Augmented Dickey-Fuller test for stationarity and the data standardization procedure. We begin with the former.

If a time series is nonstationary, meaning its mean and variance change over time, traditional methods of hypothesis testing, constructing confidence intervals, and making forecasts can be highly inaccurate. One reason for nonstationarity might be a trend in how the data is generated. As highlighted in Subchapter 3.3, Masini et al. (2021) outline that machine learning methods are built on the assumption that inputs are stationary, leading us to formally examine – and adjust if needed – the input time series.

Dickey and Fuller (1979) introduce a specific test for checking if a series is stationary,

known as the Dickey-Fuller test. This test's null hypothesis assumes that the time series is nonstationary, possessing a unit root. The alternative hypothesis, on the other hand, suggests that the time series is stationary. We are particularly interested in the enhanced version of this test, called the Augmented Dickey-Fuller (ADF) test.

Consider the linear regression model

$$\Delta Y_t = \beta_0 + \delta Y_{t-1} + \gamma_1 \Delta Y_{t-1} + \gamma_2 \Delta Y_{t-2} + \cdots + \gamma_p \Delta Y_{t-p} + u_t. \quad (34)$$

The Augmented Dickey-Fuller (ADF) test for a unit root examines 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 number of lags in the test is determined by selecting the lag length that minimizes the Akaike Information Criterion (AIC), written as

$$AIC = 2k - 2\ln(\hat{L}), \quad (35)$$

where k represents the number of estimated parameters in the model, and $\hat{L}$ is the maximum likelihood estimate for the model.

If a series is nonstationary, there are several methods to transform it into a stationary series. These methods include converting the data into logarithms to stabilize the variance, differencing the data, or combining these techniques. We employ the adequate methods for our input data.

What is more, Zhang and Qi (2005) assert that machine learning methods perform optimally when the input data exhibit a Gaussian distribution. Consequently, we standardize the time series data to have zero mean and unit variance prior to modeling to adhere to this requirement. We standardize the predictors by transforming them to a variable with a zero mean and unit standard deviation. This is done as

$$\text{Standardized value} = \frac{\text{Actual value of the predictor - Mean of the time series}}{\text{Standard deviation of the time series}}. \quad (36)$$

In summary, in the previous paragraphs we present the methods by which we preprocess the data. The next subchapters contain the data description.

3.10.2 Macroeconomic database of the National Bank of Slovakia

In this chapter, we detail the data utilized in our forecasting experiments. Firstly, we introduce the database, which is a primary factor in our decision to focus on Slovakia. The National Bank of Slovakia (NBS) has compiled and maintains an innovative, accessible, and exemplary public macroeconomic database.[6] To our knowledge, a public macroeconomic database of this scope is currently unmatched among the V4 countries.

The data is presented in a clean, consistent, and concise format, encompassing multiple periods, formats, and categories. This comprehensive availability and quality of data significantly enhance the reliability and breadth of our analyses. In addition to observed macroeconomic variables, the database includes "soft" indicators. These are referred to as soft because they predominantly consist of survey-based data that gauge the expectations of economic participants, rather than measuring actual economic activity. Coulombe et al. (2022) recommend using these indicators as input data for forecasting with machine learning methods. Unlike observed macroeconomic variables, these indicators are available with minimal time lag, enabling policymakers to initiate discussions without waiting for hard macroeconomic data[7] to become available. Although forecasting methods based on soft indicators are not expected to achieve lower forecast errors than those based on hard data, their primary function is to indicate the potential direction of the economy at a given time. Therefore, we can utilize measures of directional accuracy to assess whether these methods are meeting their intended purpose. To sum up, our forecasting approach includes not only real macroeconomic variables but also leading indicators. For soft indicators, our primary focus is on the directional accuracy of the forecasts, which is crucial for understanding the potential future trends rather than precise values.

A limitation of the database is that it only contains actual data after revisions; hence, older data vintages before revisions are not available. This constraint makes it challenging to compare the forecast performance of machine learning methods with historical forecasts made by entities like central banks, which had access to different data at the time of their forecasts. Furthermore, to our knowledge, no country or organization maintains a publicly available revisions database that includes sufficient data to reliably train machine learning methods. Therefore, data-driven

[6] https://nbs.sk/statisticke-udaje/vybrane-makroekonomicke-ukazovatele/makroekonomicka-databaza/
[7] By hard macroeconomic data we mean actually observed macroeconomic data of the time series in Tables A1.1-A1.8 in Appendix 1.

models such as ARIMA and Random Walk remain the literature standard for benchmarking, as discussed in Chapter 1.

In summary, the database is well-suited for our purposes as it contains ample data to reliably train machine learning methods. We employ data-driven models, specifically Random Walk or ARIMA, as benchmarks for the reasons previously mentioned. This setup ensures a robust framework for evaluating the effectiveness of machine learning in macroeconomic forecasting.

3.10.3 Our dataset

Given the breadth of the database, we divide our dataset into two segments to use machine learning for forecasting our chosen variables: industrial production, measured by the Index of Industrial Production (IP), and inflation, measured by the Harmonized Index of Consumer Prices (Inf). We select these target variables because they are frequently used in machine learning forecasting literature, as noted in studies made by Chakraborty and Joseph (2017), Jung et al. (2018), Medeiros et al. (2019), Maehashi and Shintani (2020), and Coulombe et al. (2022).

We document the observation period as utilizing the time series from the database, captured monthly and adjusted for seasonality when applicable, spanning from January 2008 (2008M01) to December 2019 (2019M12). This period encapsulates the economic fluctuations of the 2008–2009 Great Recession and a subsequent extended phase of economic stability. Later we extend the observation period to include data up to September 2023 (2023M09) to also cover the economic impacts of the COVID-19 pandemic. We use variables from the following categories: industrial production, prices, revenue, wages, employment, foreign trade, current account, exchange rates and indicators. [8]

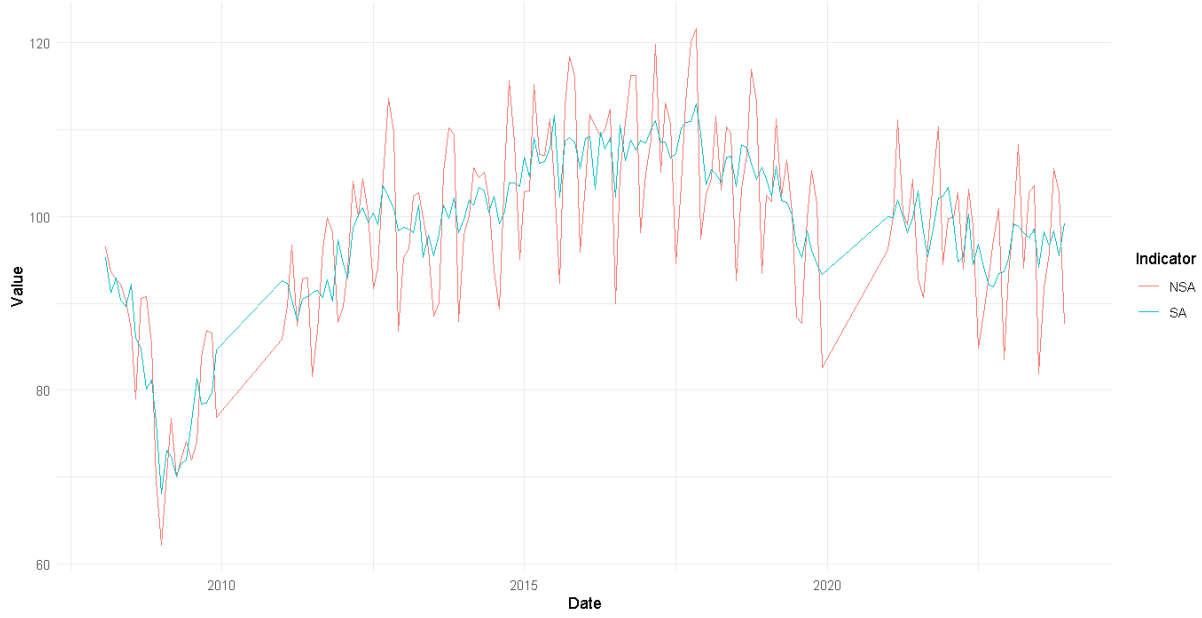
Regrettably, splitting the data into distinct segments to compare model performance across different economic conditions is impractical. Specifically, the dataset from the crisis period alone would be insufficiently large for effective training of machine learning or other statistical models, rendering any derived conclusions unreliable. Consequently, we choose to utilize the entire dataset without segmenting it into multiple periods. To address potential

[8] The full list of variables and their specific categories is presented in Appendix 1

temporal variations in the regression parameters, we apply a rolling-window approach, which is detailed in Subchapter 3.8.1.

To continue with, we detail the data preprocessing steps, focusing on stationarization and standardization, along with the extraction of common factors. As emphasized in Subchapter 3.2.1, ensuring the stationarity of time series is crucial before they are used in forecasting models. To achieve stationarity, macroeconomists typically eliminate the seasonal and trend components of the data. Fortunately, our macroeconomic database offers both seasonally adjusted and unadjusted series, allowing us to directly utilize the seasonally adjusted data, thereby bypassing the need for manual seasonal adjustments. A few series are not available in the seasonally adjusted form, so we stationarize them accordingly.

Figure 10: Seasonally adjusted and non-adjusted series of IP



Source: NBS macroeconomic database

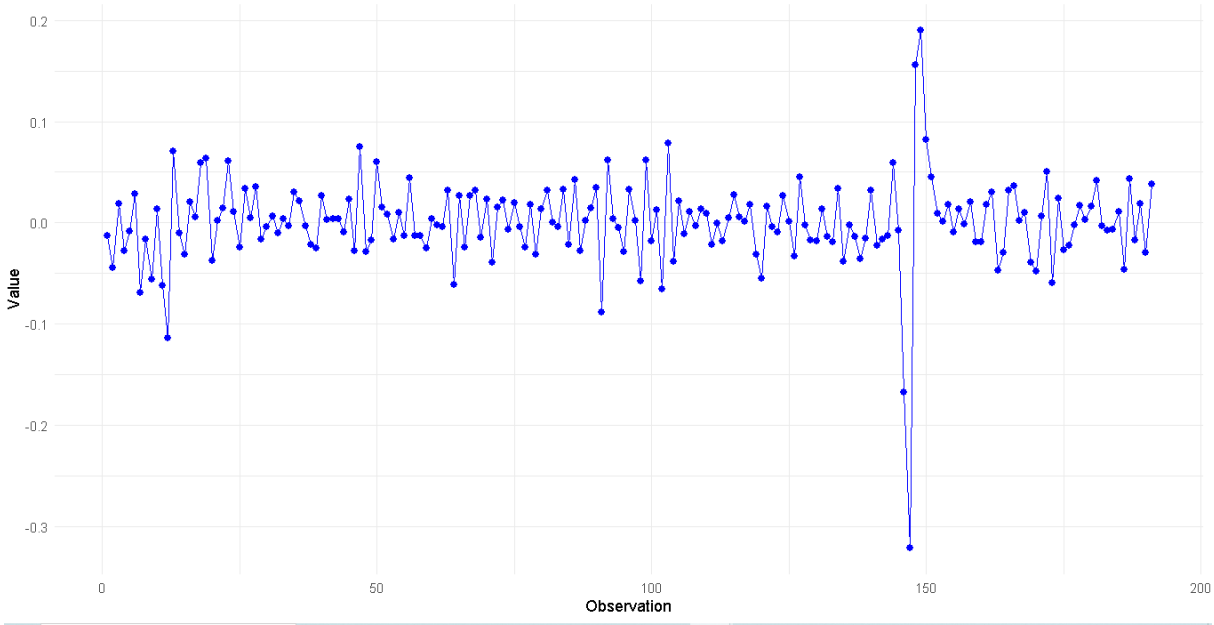
For illustration, Figure 10 showcases the effect of seasonal adjustments on industrial production series, highlighting the smoother nature of the seasonally adjusted series, which is more suitable for forecasting purposes. Figure 10 also indicates that the data exhibits an upward trend over time. Thus, to prepare the series for modeling, we remove this trend component. Trend removal involves taking the first differences of the variables in percentages like the unemployment rate, thereby following the standard approach in literature; and of soft indicators, which range between -100 and 100 and cannot be logarithm-transformed due to the presence of non-positive values. These series achieve stationarity post-differencing. For other variables, we apply logarithmic transformation to stabilize variance and then apply

differencing. Upon completion of these adjustments, each time series is confirmed to be stationary, as verified by the Augmented Dickey-Fuller (ADF) test.

Figure 11 displays the industrial production series post-trend removal, demonstrating a series that oscillates around a constant mean with stable variance. Despite this, a few influential observations may exist that could potentially increase forecasting errors, suggesting that the use of Mean Absolute Error (MAE) might be appropriate for assessing the robustness of our forecasting results.

Lastly, we conclude our preprocessing by mean-variance standardizing the regressors. This normalization process is essential for aligning the scales of the variables, facilitating more effective analysis and comparisons across the data set, as detailed in Subchapter 3.10.1.

Figure 11: Stationary series of IP



Source: NBS macroeconomic database

Last but not least, we extract common factors from the data, aligning with one of our objectives outlined in Chapter 2. Our goal is to evaluate whether data that has been dimensionally reduced through factor analysis can yield forecast errors and accuracy comparable to those obtained using the original time series from which the factors were extracted. We employ Principal Component Analysis (PCA), as detailed in Subchapter 3.8.2, utilizing a rolling window scheme to systematically apply common factors at each step in the forecasting process. Once extracted, these factors are then used as regressors in various machine learning methods. This approach, while established, has been infrequently tested in practical scenarios. We aim to explore whether machine learning methods can effectively forecast our

selected variables using both the original, complete time series and the dimensionally reduced information set derived through PCA.

To summarize, this chapter introduces the data preprocessing steps, the NBS macroeconomic database and outlines the specific data we utilize for forecasting. We detail the preprocessing steps, including stationarization, standardization, and factor extraction. Following these, the data is fed into the methods, and the results are subsequently discussed in the following chapter. This methodology ensures that we comprehensively prepare the data to accurately evaluate the effectiveness of machine learning methods in economic forecasting.

4 Results

4.1 Pre-COVID

4.1.1 Forecasting industrial production using hard macroeconomic data

This subchapter presents the results of industrial production (IP) forecasts based on hard macroeconomic data. Table 3 displays the relative MSE values for the RLS machine learning methods, with the MSE value of the benchmark model set as 1 for comparison. It is evident that each machine learning method surpasses the performance of the benchmark model.

Table 3: Relative MSE and MAE of RLS models - IP

Relative MSE	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	1	1	1	1	1
Lasso	0.25376	0.25854	0.25077	0.25699	0.26991
Ridge	0.57640	0.58032	0.59155	0.58178	0.57804
Elastic Net	0.25925	0.25221	0.26200	0.27318	0.25893
Relative MAE					
Benchmark	1	1	1	1	1
Lasso	0.45058	0.44097	0.45073	0.44895	0.45683
Ridge	0.80952	0.80917	0.82727	0.82146	0.81080
Elastic Net	0.45879	0.44100	0.46591	0.46079	0.44607

Source: authors' calculations

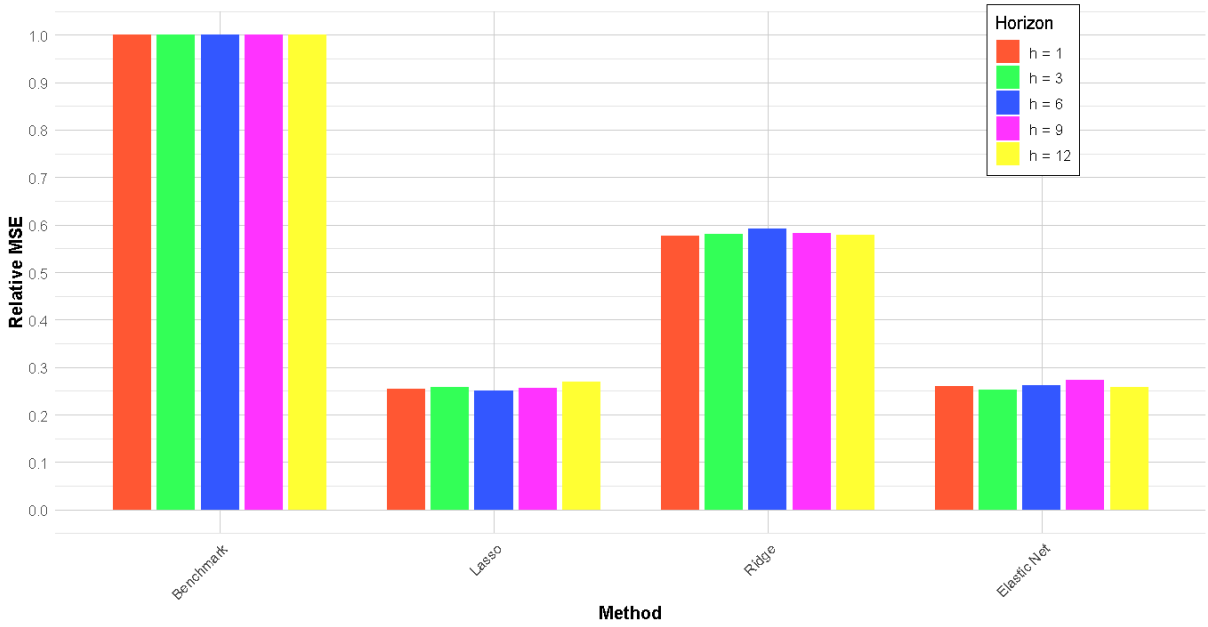
In analyzing MSE, we find that all RLS models perform better than the benchmark. Specifically, both Lasso and Elastic Net models exhibit a relative MSE of around 0.25. This suggests that in the Elastic Net model, the Lasso component carries more weight compared to the Ridge component. Referring to Subchapter 3.4, we observe that in this scenario, selecting variables plays a more crucial role than shrinking coefficients. The Ridge model also surpasses the benchmark, showing a relative MSE between 0.57 and 0.59 for every forecast horizon. Consequently, all RLS models outshine the benchmark model.

Furthermore, the relative MAE results reinforce our earlier conclusion, being smaller than those of the benchmark. This reassures us that the RLS models' superior performance isn't simply due to a few large forecast errors.

To summarize, the RLS models consistently outperform the benchmark across every horizon. This robust performance is expected due to the presence of numerous correlated variables in the dataset, which tend to convey similar information. Additionally, the shorter time series used in our study compared to other machine learning research might prompt Lasso and Elastic Net methods to disregard less critical variables due to the limited number of

observations available. Moreover, the magnitude of the outperformance remains consistent across all models and forecast horizons, as depicted in Figure 12.

Figure 12: Relative MSE of RLS models over time - IP



Source: authors' calculations

The machine learning methods demonstrate robust superiority as evidenced by their outperformance in both MSE and MAE metrics. This consistent outperformance across metrics allows us to evaluate the statistical significance of the results using the DM test, as detailed in Subchapter 3.9. The DM test focuses on analyzing the original forecast errors rather than the derived MSE or MAE values. Table 4 displays the outcomes of this test represented in terms of *p*-values. Notably, numbers marked with two asterisks (**) indicate horizons where the forecasts of machine learning methods statistically significantly outperform the forecasts of the benchmark model at the 5% significance level. The forecasts generated by most RLS models exhibit significant improvement at the 5% level of confidence. Only the forecasts for the 9 and 12-month horizons of the Ridge model show significance at the 10% level.

Table 4: DM test *p*-values of RLS methods - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Lasso	0.00639**	0.01358**	0.01289**	0.04393**	0.00794**
Ridge	0.00575**	0.03273**	0.04754**	0.06779*	0.05317*
Elastic Net	0.00664**	0.01321**	0.01174**	0.04661**	0.00935**

Source: authors' calculations

Since the outperformance is statistically significant, we can express the magnitude of outperformance in percentage terms. Table 5 indicates that the Lasso model's outperformance ranges between 73.01% and 74.92%. Ridge consistently outperforms the benchmark by 40.84%

to 42.20%, while Elastic Net closely mirrors Lasso, ranging between 73.80% at $h = 6$ and 74.78% at $h = 3$. The results suggest that by fine-tuning hyperparameters at each step, the methods effectively select the most relevant variables for the forecasts to surpass the benchmark model. Shrinkage also seems to work, but to a lesser extent.

Table 5: % improvement of RLS forecasts over the benchmark - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Lasso	74.62%	74.15%	74.92%	74.30%	73.01%
Ridge	42.36%	41.97%	40.84%	41.82%	42.20%
Elastic Net	74.07%	74.78%	73.80%	72.68%	74.11%

Source: authors' calculations

Continuing from the previous points, it is crucial to establish that machine learning methods not only yield lower forecast errors but also accurately capture the direction of change in the forecast variable. This aspect is often as important as the magnitude of the errors because a method that predicts the correct trend but is off in magnitude can still be very useful, especially in macroeconomic planning and policy making. To assess this capability, we compute the directional accuracy of each machine learning method and compare it to that of the benchmark model. Directional accuracy specifically measures the percentage of times the predicted change in direction (increase or decrease) of the forecast variable aligns with the actual observed change. The comparative results of this analysis are presented in Table 6.

Table 6: Directional accuracy of RLS models - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	41.43%	31.37%	26.67%	21.51%	22.18%
Lasso	90.00%	88.24%	86.92%	86.74%	85.59%
Ridge	72.86%	74.02%	74.10%	73.66%	74.01%
Elastic Net	90.00%	86.76%	87.44%	85.48%	85.45%

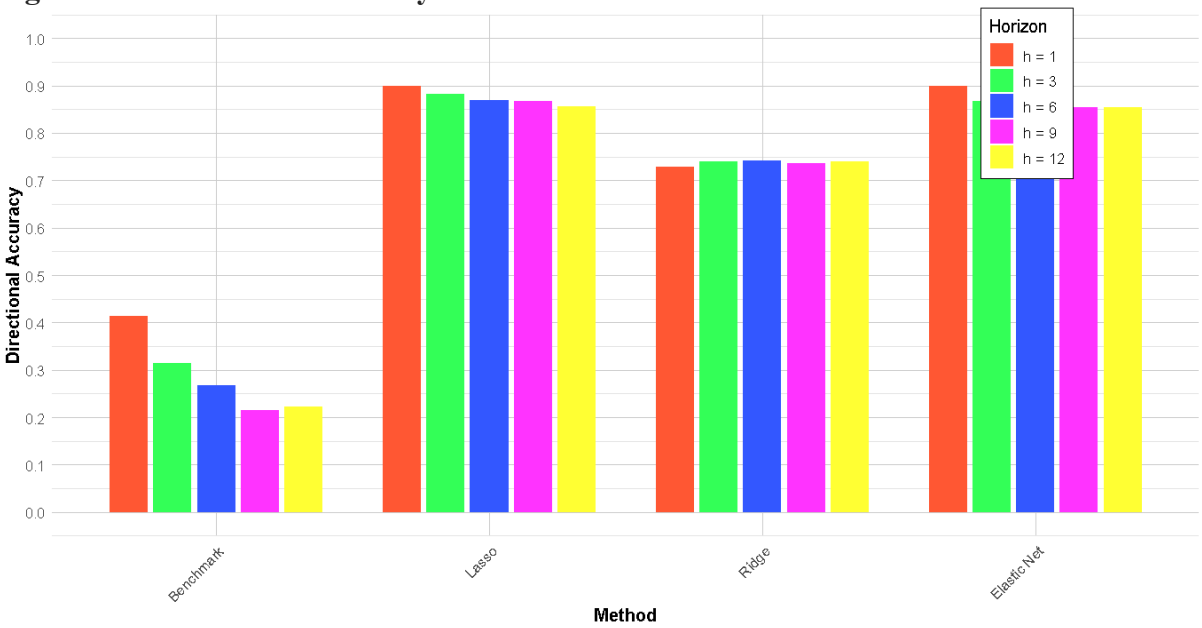
Source: authors' calculations

Each RLS model is more likely to correctly predict the direction of change in the target variable compared to the benchmark model, as demonstrated in Table 6. The DA of both Lasso and Elastic Net ranges between 85.45% and 90.00%, while the DA of Ridge falls between 72.86% and 74.10%. In contrast, the benchmark model exhibits a DA ranging from 21.51% to 41.43%, depending on the forecast horizon.

Furthermore, this enhancement remains consistent over time, as illustrated in Figure 13. Another noteworthy observation emerges: as the forecast horizon extends, the DA of the benchmark model decreases significantly, contrasting with the marginal decreases observed in Lasso and Elastic Net models. This discrepancy arises because our benchmark model inherently

integrates only short-term information. The DA of the Ridge model maintains consistency, albeit at a lower level compared to Lasso or Elastic Net, yet still notably surpassing that of the benchmark.

Figure 13: Directional Accuracy of RLS models over time - IP



Source: authors' calculations

We proceed by presenting the results of the EML models in Table 7.

Table 7: Relative MSE and MAE of EML models - IP

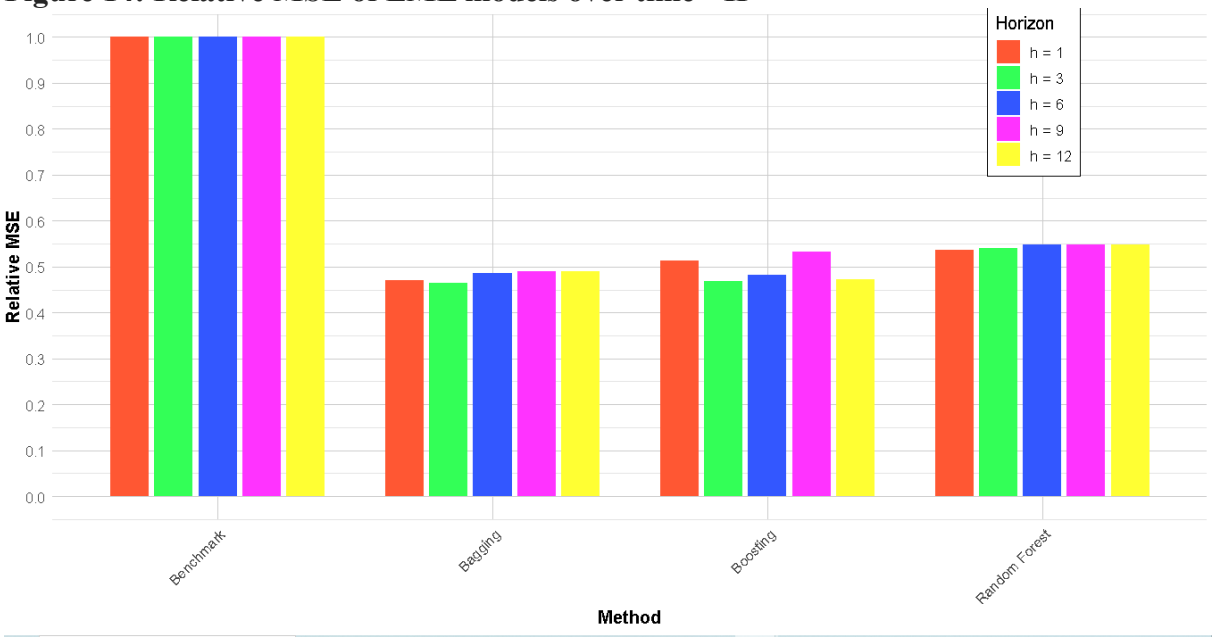
Relative MSE	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	1	1	1	1	1
Bagging	0.47081	0.46523	0.48503	0.49009	0.48902
Boosting	0.51369	0.46769	0.48275	0.53275	0.47255
Random Forest	0.53703	0.54068	0.54897	0.54848	0.54870
Relative MAE					
Benchmark	1	1	1	1	1
Bagging	0.60385	0.59082	0.61838	0.61948	0.60656
Boosting	0.62291	0.60607	0.63971	0.67181	0.60336
Random Forest	0.67807	0.67354	0.68552	0.66990	0.66560

Source: authors' calculations

Firstly, in terms of MSE, every EML model outperforms the benchmark. Bagging exhibits a MSE ranging between 0.47 and 0.49, Boosting between 0.46 and 0.54, and Random Forest between 0.53 and 0.55. This indicates that Bagging performs best for every horizon, except for $h = 12$, where Boosting holds a slight advantage. Secondly, the relative MAE results corroborate our earlier conclusion, as they are also smaller than those of the benchmark. This assures us that the outperformance of the EML models is not attributed to a few large forecast errors. One notable observation from the analysis is that, in terms of MAE, the nonlinear methods show a

smaller increase compared to the relative MSE than the RLS methods. This suggests that nonlinear methods might be less susceptible to the impact of large forecast errors. Additionally, the magnitude of the outperformance remains consistent for every model and across every horizon, as illustrated in Figure 14.

Figure 14: Relative MSE of EML models over time - IP



Source: authors' calculations

Table 8 presents the *p*-values from the DM test. The forecasts from Bagging are significantly better at the 5% level for every horizon. For the Boosting model, the forecast at $h = 1$ is significant at the 10% level, while the forecasts for the remaining horizons are significant at the 5% level. Similarly, the forecasts from the Random Forest model show significant outperformance at the 5% level for all horizons except for $h = 1$.

Table 8: DM test *p*-values of EML methods - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Bagging	0.03742**	0.00904**	0.01138**	0.04915**	0.02405**
Boosting	0.05308*	0.02725**	0.01276**	0.02942**	0.02178**
Random Forest	0.06095*	0.04542**	0.02695**	0.02935**	0.03348**

Source: authors' calculations

Since the outperformance is statistically significant, we can quantify the magnitude of outperformance in percentage terms. Table 9 presents the results. The table reveals that the outperformance of Bagging ranges between 50.99% and 53.48%, which is impressive, although slightly less than the best RLS models. Boosting demonstrates outperformance ranging from 46.72% to 53.23%, while Random Forest shows outperformance between 45.10% and 46.30%. Generally, Bagging emerges as the best-performing model, except for $h = 12$, where Boosting

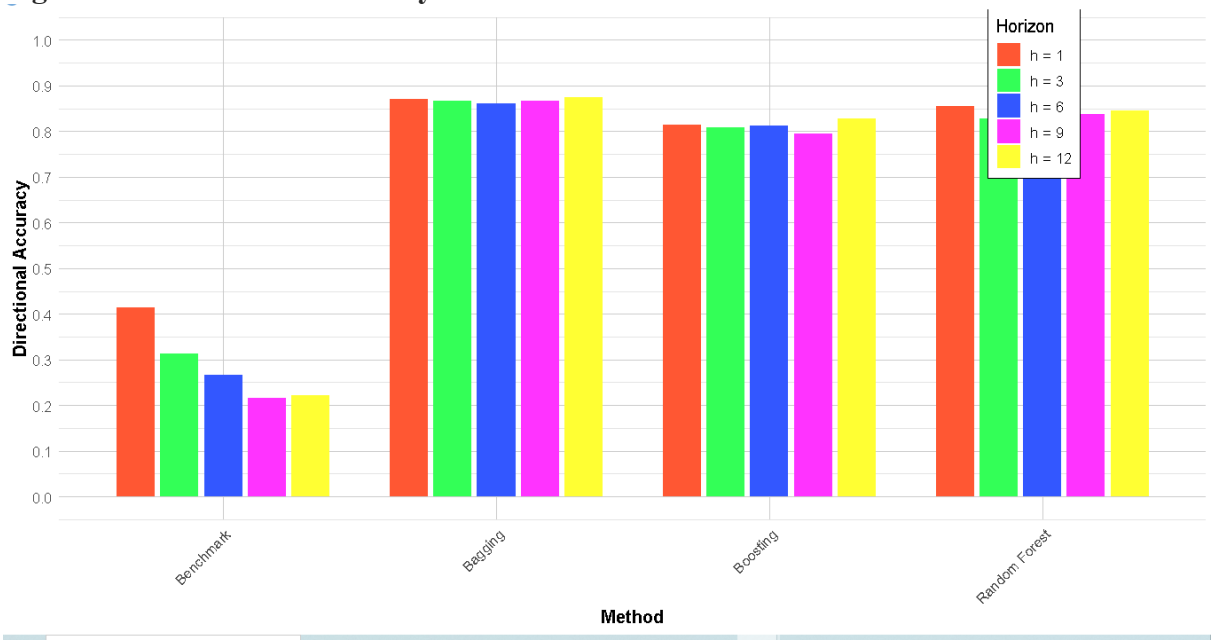
holds a slight advantage. Among the EML models, Random Forest exhibits the highest consistency.

Table 9: % improvement of EML forecasts over the benchmark - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Bagging	52.92%	53.48%	51.50%	50.99%	51.10%
Boosting	48.63%	53.23%	51.72%	46.72%	52.74%
Random Forest	46.30%	45.93%	45.10%	45.15%	45.13%

Source: authors' calculations

Figure 15: Directional accuracy of EML models over time - IP



Source: authors' calculations

Moving on to directional accuracy, every EML model is more likely to correctly predict the direction of change in the target variable compared to the benchmark model. The results are presented in Table 10. The DA of Bagging ranges between 86.15% and 87.43%, Boosting between 79.39% and 82.77%, and Random Forest between 82.56% and 85.51%. In comparison, the benchmark model exhibits a DA ranging from 21.51% to 41.43%, depending on the forecast horizon. What is more, this improvement is consistent over time, as shown in Figure 15.

Table 10: Directional accuracy of EML models - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	41.43%	31.37%	26.67%	21.51%	22.18%
Bagging	87.14%	86.76%	86.15%	86.74%	87.43%
Boosting	81.43%	80.88%	81.28%	79.39%	82.77%
Random Forest	85.51%	82.84%	82.56%	83.69%	84.46%

Source: authors' calculations

In Table 11, we present the results of the remaining models, namely the Support Vector Machine (SVM), the Feedforward Neural Network (FFNN), and the Long-Short Term Memory

Neural Network (LSTM). Due to their less compelling performance, we summarize their results in one table.

Table 11: Forecasts with SMV, FFNN and LSTM - IP

Relative MSE	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Support Vector Machine	0.77375	0.85289	0.75612	0.74687	0.75215
Feedforward NN	7.42215	7.02437	6.87417	6.60251	6.42844
LSTM NN	128.124	18.7122	7.54591	4.03824	3.68138
Relative MAE					
Benchmark	1	1	1	1	1
Support Vector Machine	0.92173	0.9459	0.90444	0.89306	0.90576
Feedforward NN	3.71541	3.56651	3.56545	3.48494	3.40986
LSTM NN	13.2955	4.81009	3.17909	2.20504	2.24366
DM test					
Support Vector Machine	0.20681	0.05121	0.03532	0.13286	0.00035
Feedforward NN	1	1	1	1	1
LSTM NN	1	1	0.99996	0.99192	0.99998
DA					
Benchmark	41.43%	31.37%	26.67%	21.51%	22.18%
Support Vector Machine	65.71%	67.65%	69.23%	71.33%	71.89%
Feedforward NN	54.29%	55.39%	56.15%	57.17%	57.77%
LSTM NN	45.71%	48.04%	48.97%	53.41%	54.38%

Source: authors' calculations

Starting with the relative MSE, these models either show results closer to that of the benchmark, as observed in the SVM model, or higher, particularly in the case of the NN models. Notably, both NN models exhibit substantial underperformance. However, this can be attributed to the inability to optimize hyperparameter search using R, unlike the other models. This limitation is evident in the overall results.

Furthermore, the relative MAE further confirms the underperformance of the FFNN and LSTM models, as neither of them produces statistically significantly better forecasts compared to the benchmark model. Although their DA values are slightly higher than those of the benchmark, given the magnitude of errors and the superior performance of other models, we advise against drawing meaningful conclusions from these models for now. Once we have access to appropriate computational capacity, we plan to delve deeper into exploring their

performance. Until then, we do not give them further attention in this study.

Moving on to the SVM model, as previously mentioned, it manages to outperform the benchmark, albeit to a lesser extent compared to other models. This outperformance is statistically significant at the 5% level for $h = 6$ and $h = 12$, and at the 10% level for $h = 3$, but it is insignificant for $h = 1$ and $h = 9$. The DA values of the SVM model range between 65.71% and 71.89%, which are lower than those of the other models, despite showing an increase in accuracy with the forecast horizon.

To summarize, in this subchapter we forecast industrial production using hard macroeconomic data. Most of the machine learning models exhibit significant outperformance compared to the benchmark model, with the exception of the neural network models due to computational limitations. The models, sorted from best to worst in terms of % improvement over the benchmark, are presented in Table 12. There are some models that are really close to each other in terms of performance, and one model might be the best on one horizon, while another on a different horizon (e.g. Lasso is better than Elastic Net at *h = 9*, while the latter slightly outperforms the former at *h = 12*).

Table 12: Forecasting models from best to worst - IP

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
1.	Lasso	Elastic Net	Lasso	Lasso	Elastic Net
2.	Elastic Net	Lasso	Elastic Net	Elastic Net	Lasso
3.	Bagging	Bagging	Boosting	Bagging	Boosting
4.	Boosting	Boosting	Bagging	Boosting	Bagging
5.	RF	RF	RF	RF	RF
6.	Ridge	Ridge	Ridge	Ridge	Ridge
7.	SVM	SVM	SVM	SVM	SVM
8.	FFNN	FFNN	FFNN	LSTM	LSTM
9.	LSTM	LSTM	LSTM	FFNN	FFNN

Source: authors' calculations

Firstly, the top-performing models are Lasso and Elastic Net, which demonstrate statistically significant outperformance of the benchmark model at every forecasting horizon at the 5% level. Additionally, Lasso and Elastic Net models boast the highest directional accuracy values among all models, with consistent results over time. Although the Ridge model's performance is slightly weaker, it still significantly outperforms the benchmark, albeit not as effectively as Lasso and Elastic Net.

Secondly, EML models display slightly higher errors and lower directional accuracy compared to Lasso and Elastic Net. However, they are still capable of statistically significant outperformance of the benchmark model, especially Bagging, which emerges as the top

performer among the EML models.

Thirdly, the Support Vector Machine (SVM) model yields mixed results, leading us to recommend prioritizing models from the previous two categories, particularly Lasso and Elastic Net for regularization, and Bagging for capturing nonlinear relationships in the data.

Moving forward, we delve into exploring the forecasting performance of machine learning models based on common factors in the next subchapter.

4.1.2 Forecasting industrial production using common factors

In this subchapter, we explore the application of machine learning methods to forecasting based on common factors extracted from the available dataset, drawing inspiration from the works of Shintani (2005) and Maehashi and Shintani (2020). Our aim is to ascertain whether this approach outperforms the benchmark model and, if so, how it compares to forecasting performance based on hard macroeconomic data without dimensional reduction.

Table 13: Relative MSE of forecasts based on common factors - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	1	1	1	1	1
Lasso	0.701524	0.700192	0.681737	0.852603	0.698305
Ridge	0.689632	0.705956	0.641994	0.640785	0.687538
Elastic Net	0.715571	0.832926	0.705255	0.667444	0.670397
Bagging	0.873563	0.831503	0.840161	0.812082	0.813511
Boosting	0.729903	0.681461	0.676986	0.663042	0.688355
Random Forest	0.837169	0.873106	0.832147	0.831806	0.825432
SVM	0.739667	0.714448	0.720253	0.712005	0.710229

Source: authors' calculations

To commence, Table 13 presents the relative MSE values of the forecasts based on common factors. The results indicate that forecasts based on common factors continue to outperform the benchmark model, with no relative MSE exceeding one. Starting with RLS methods, both Lasso and Elastic Net yield similar results, as does the Ridge model, despite being the least effective in forecasting using hard macroeconomic data. The cause of this might be the reduced dimension of the dataset, as if there is not too many observables, coefficient shrinkages is more pronounced while Lasso loses its edge. In terms of EML methods, Boosting outperforms both Bagging and Random Forest, which exhibit comparable results. Additionally, the performance of the Support Vector Machine (SVM) model remains consistent with the findings in the previous subchapter, meaning it is still the weakest performing model.

Table 14: % difference between forecasts based on factors and hard data - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Lasso	176.45%	170.83%	171.85%	231.76%	158.72%
Ridge	19.64%	21.65%	8.53%	10.14%	18.94%
Elastic Net	176.01%	230.24%	169.18%	144.32%	158.91%
Bagging	85.55%	78.73%	73.22%	65.70%	66.35%
Boosting	42.09%	45.71%	40.23%	24.46%	45.67%
Random Forest	55.89%	61.48%	51.58%	51.66%	50.44%
SVM	-4.40%	-16.23%	-4.74%	-4.67%	-5.57%

Source: authors' calculations

Continuing, we present the percentage difference between the relative MSE values of forecasts based on hard macroeconomic data and common factors in Table 14. Positive values indicate that the model based on hard data is more accurate, while negative values suggest the opposite.

To add, Table 14 reveals that in most cases, models based on hard data exhibit greater accuracy. Moreover, their forecasts demonstrate higher consistency across multiple horizons. For instance, while Table 3 highlights the consistency of Lasso forecasts, Table 14 demonstrates that the magnitude of underperformance for forecasts based on common factors fluctuates significantly, ranging between 158% and 232% for the Lasso model. This indicates that forecasts based on hard macroeconomic data are more consistent over multiple horizons compared to those based on common factors. Similar observations can be made for the Ridge model. On the other hand, the SVM model displays lower errors when forecasting based on common factors. However, the differences are relatively small, ranging between 4.40% and 5.57%, except for $h = 3$, which shows a larger difference at 16.23%.

Table 15: Directional accuracy of forecasts based on common factors - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	41.43%	31.37%	26.67%	21.51%	22.18%
Lasso	65.71%	68.63%	67.95%	68.64%	67.66%
Ridge	65.71%	66.67%	68.21%	68.82%	69.07%
Elastic Net	64.29%	67.65%	68.46%	68.10%	68.64%
Bagging	68.57%	68.14%	69.74%	71.51%	72.74%
Boosting	62.86%	69.12%	66.67%	67.03%	67.66%
Random Forest	67.14%	64.22%	67.95%	69.71%	69.77%
SVM	67.14%	65.69%	67.18%	68.46%	70.06%

Source: authors' calculations

To conclude, we present the directional accuracy of forecasts based on common factors in Table 15. This table reinforces our previous findings. Directional accuracy values based on common factors are significantly higher than those of the benchmark model. However, they are

lower compared to the results based on hard macroeconomic data, as shown in Tables 6 and 10. Nonetheless, their values remain relatively stable across multiple horizons.

In conclusion, forecasts based on common factors outperform the benchmark model but underperform forecasting based on hard macroeconomic data. The exception is the SVM model, which, despite displaying lower errors with common factors, is generally the worst-performing model.

4.1.3 Forecasting industrial production using soft indicators

In Subchapter 3.10, we establish that soft indicators are unlikely to yield forecasts with lower errors compared to those based on hard data. However, their primary purpose is not to precisely capture the magnitude of change in macroeconomic variables. Instead, they serve to reflect the sentiments of economic participants regarding the general direction of the economy and its components. Consequently, they offer an approximate overview of where the economy is headed. The key advantage of soft indicators lies in their availability, as there is no substantial lag in obtaining data, unlike macroeconomic variables where decision-makers often face delays of weeks or even months. In summary, due to their design and availability, soft indicators can serve as a valuable tool for forecasting the direction of the economy, as acknowledged by Coulombe et al. (2022).

Table 16: Relative MSE of forecasts based on indicators - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	1	1	1	1	1
Lasso	1.32312	1.34674	1.39162	1.51222	1.48444
Ridge	1.12521	1.14370	1.22579	1.35124	1.37424
Elastic Net	1.29585	1.16898	1.24093	1.58505	1.50255
Bagging	1.31483	1.38907	1.46642	1.61058	1.63486
Boosting	1.28823	1.24934	1.38982	1.48275	1.52916
Random Forest	1.27363	1.29351	1.36195	1.52735	1.51519
SVM	1.21398	1.22918	1.31444	1.42896	1.43386

Source: authors' calculations

Given this premise, we are interested in exploring the potential of combining input data from soft indicators (listed in Table A1.9) and utilizing machine learning methods for forecasting. Our objective is to demonstrate that by combining indicators in this manner, we can increase the likelihood of forecasting the direction of change in our target variable, which

in this case is industrial production. We conduct the forecasting exercise and report the relative MSE values in Table 16.

As anticipated, Table 16 confirms that indicators are not the most suitable source of information when we seek the most precise forecasts in terms of magnitude. Generally, the benchmark model outperforms machine learning models based on indicators in forecasting industrial production, as indicated by their relative MSE values consistently exceeding 1. Our results thus underscore the superiority of using hard data to minimize relative MSE in forecasts. Alternatively, if access to a large dataset is limited, relying on the benchmark data-driven model proves to be more reliable than using indicators, if we are interested in the magnitude of errors.

Table 17: Directional accuracy of forecasts based on indicators - IP

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	41.43%	31.37%	26.67%	21.51%	22.18%
Lasso	61.43%	61.27%	58.46%	56.27%	56.50%
Ridge	55.71%	57.84%	57.18%	57.53%	57.77%
Elastic Net	58.57%	59.80%	62.31%	58.42%	59.18%
Bagging	57.14%	55.39%	56.67%	56.63%	56.92%
Boosting	58.57%	61.27%	55.13%	58.24%	59.46%
Random Forest	58.57%	61.27%	60.26%	58.96%	60.59%
SVM	61.43%	60.29%	59.49%	59.32%	59.04%

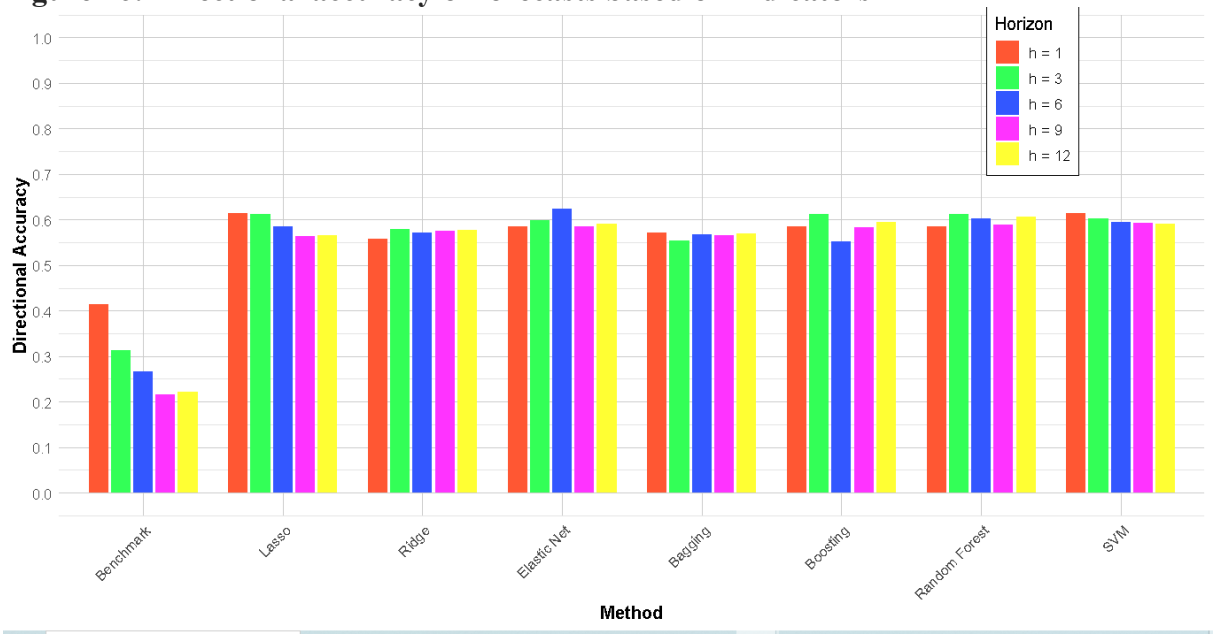
Source: authors' calculations

Continuing, we analyze directional accuracy, our primary metric of interest in the case of indicator data. The DA values are presented in Table 17, revealing several noteworthy observations. Firstly, every machine learning model significantly outperforms the benchmark model across all horizons, validating our assumptions and demonstrating the effectiveness of utilizing indicators for forecasting. Secondly, while the outperformance is smaller compared to forecasts based on hard macroeconomic data, this outcome is anticipated. Forecasts based on indicator data inherently yield lower DA values than those based on hard macroeconomic data due to the nature of indicators measuring sentiments and their availability. Thirdly, this outperformance remains consistent across horizons, as depicted in Figure 16. Fourthly, there are minimal differences between models, indicating that we can select the model with the least computational requirements (such as Lasso in our case) and utilize it for computing directional accuracy.

In conclusion, indicators serve their intended purpose effectively. By leveraging an extensive dataset of soft indicators and employing machine learning models, we can predict the

direction of change in industrial production with a higher likelihood than if we solely relied on the benchmark model.

Figure 16: Directional accuracy of forecasts based on indicators - IP



Source: authors' calculations

4.1.4 Forecasting inflation using hard macroeconomic data

Continuing our analysis, we present the relative MSE and MAE results of the forecasts of inflation in Table 18.

Table 18: Relative MSE and MAE of RLS models - Inf

Relative MSE	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Lasso	0.85183	0.84309	0.84579	0.82543	0.82269
Ridge	0.65684	0.66673	0.65922	0.65645	0.63384
Elastic Net	0.57095	0.62576	0.62040	0.54723	0.59771
Relative MAE					
Benchmark	1	1	1	1	1
Lasso	0.87313	0.86684	0.86351	0.86148	0.86148
Ridge	0.79810	0.81195	0.80224	0.79797	0.79797
Elastic Net	0.74610	0.75678	0.76212	0.71168	0.71168

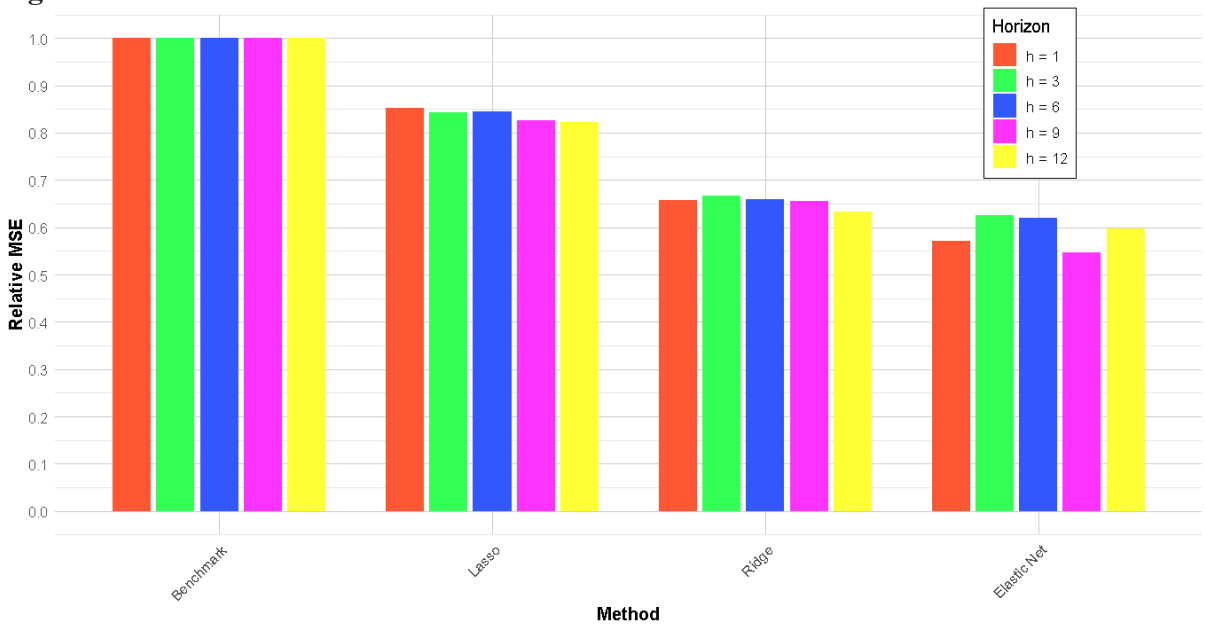
Source: authors' calculations

Firstly, when we look at MSE, we find that all RLS models perform better than the benchmark. The Lasso model has a relative MSE ranging from 0.82 to 0.86. Similarly, the Ridge model shows a relative MSE of around 0.63 to 0.65 for each forecast period, consistently outperforming the benchmark. The most effective among these is the Elastic Net model, with a

relative MSE between 0.57 and 0.63. Notably, since the Ridge model exhibits a lower relative MSE compared to Lasso, the Elastic Net model incorporates more information from Ridge, indicating that in this scenario, shrinkage is favored over variable selection. This is in contrast to Table 3, when variable selection was preferred. Overall, all RLS models surpass the benchmark model in terms of performance.

Secondly, the relative MAE results further support our earlier conclusion, as they are also lower than those of the benchmark. In opposition to Table 3, the differences between relative MSE and MAE values are relatively miniscule in Table 18. This suggests that RLS models are more robust to large outlier forecast errors in this case. Furthermore, the extent of outperformance remains consistent across all models and forecast horizons, as demonstrated in Figure 17.

Figure 17: Relative MSE of RLS models over time - Inf



Source: authors' calculations

To ascertain the statistical significance of this outperformance, we present the DM test values in Table 19. It is evident that each forecast generated by the RLS models exhibits statistically significant improvement at the 5% significance level.

Table 19: DM test *p*-values of RLS methods - Inf

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Lasso	0,00622**	0,00121**	0,00415**	0,00525**	0,00168**
Ridge	0,01335**	0,01527**	0,01263**	0,01498**	0,03328**
Elastic Net	0,00197**	0,00430**	0,00565**	0,00775**	0,02192**

Source: authors' calculations

Table 20: % improvement of RLS forecasts over the benchmark - Inf

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Lasso	14.82%	15.69%	15.42%	17.46%	17.73%
Ridge	34.32%	33.33%	34.08%	34.35%	36.62%
Elastic Net	42.91%	37.42%	37.96%	45.28%	40.23%

Source: authors' calculations

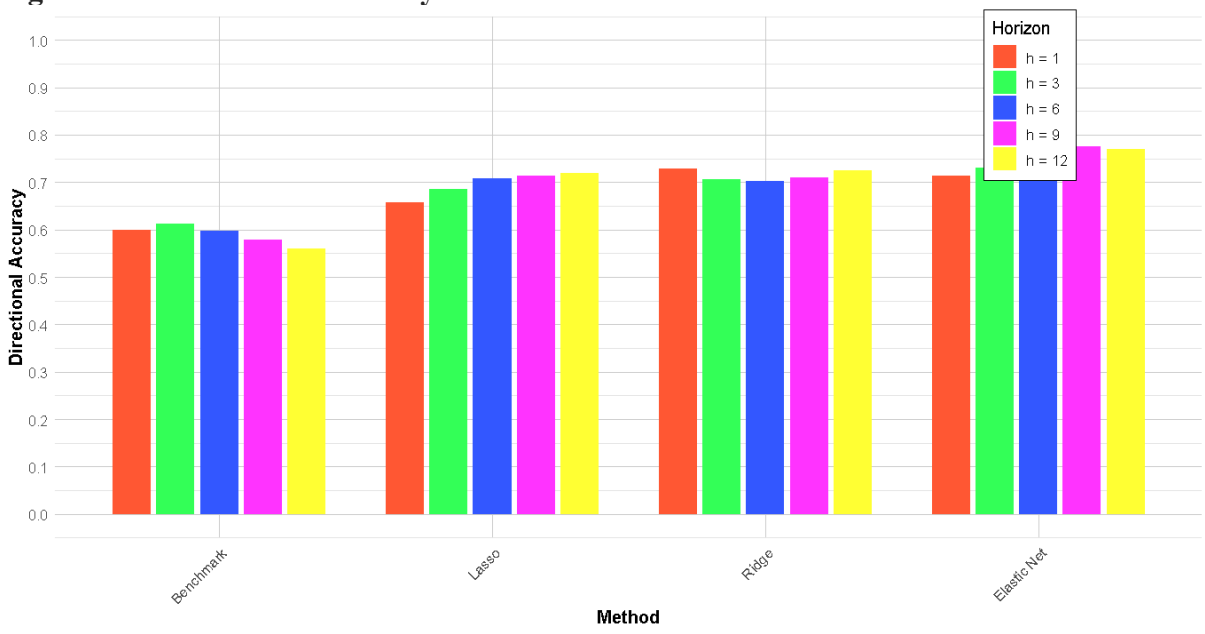
Given the statistical significance of the outperformance, we can express the extent of improvement in percentage figures. Table 20 presents the results. The Lasso model's outperformance ranges between 14.82% and 17.73%, with an increase as the forecast horizon increases. The Ridge model consistently outperforms the benchmark by 33.33% to 36.63%, while the Elastic Net model yields the highest improvement, ranging from 37.42% at $h = 3$ to 45.28% at $h = 9$.

Table 21: Directional accuracy of RLS models - Inf

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

Source: authors' calculations

Figure 18: Directional Accuracy of RLS models over time - IP



Source: authors' calculations

Transitioning to DA, each RLS model demonstrates a greater likelihood of correctly capturing the direction of change in the target variable compared to the benchmark model. The findings are summarized in Table 21. The directional accuracy of the Lasso model falls within the range of 65.71% to 71.89%, while that of Ridge ranges from 70.26% to 72.86%. The Elastic

Net performs best in terms of DA as well. In contrast, the benchmark model's directional accuracy varies from 56.07% to 61.27%, contingent upon the forecast horizon.

Furthermore, this enhancement maintains consistency over time, as depicted in Figure 18. The DA values of Lasso and Elastic Net exhibit a slight growth as *h* increases. However, the DA values of the Ridge model do not display a discernible pattern of growth. Another noteworthy observation is that, with an increase in the forecast horizon, the DA of the benchmark model experiences only a marginal decrease.

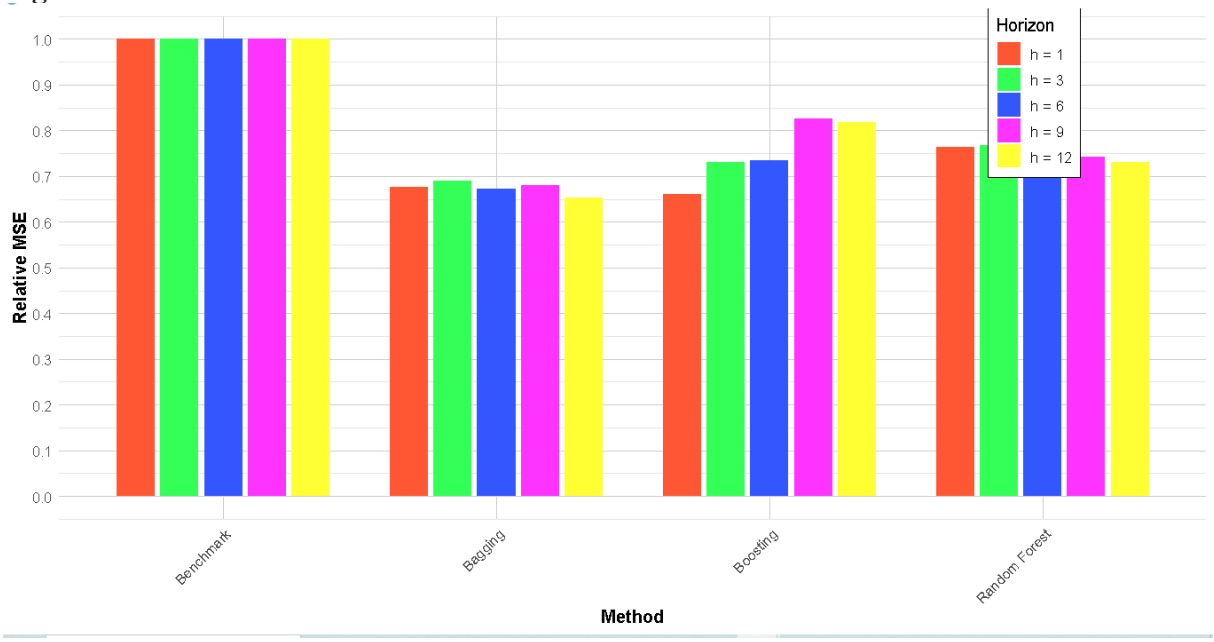
We proceed by presenting the results of the EML models in Table 22.

Table 22: Relative MSE and MAE of EML models - Inf

Relative MSE	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Bagging	0.67586	0.69034	0.67162	0.67980	0.65210
Boosting	0.66167	0.73106	0.73522	0.82561	0.81767
Random Forest	0.76327	0.76775	0.74941	0.74185	0.73045
Relative MAE					
Benchmark	1	1	1	1	1
Bagging	0.80201	0.80335	0.79988	0.81326	0.81326
Boosting	0.7671	0.83876	0.82218	0.87872	0.82839
Random Forest	0.8565	0.83859	0.82778	0.82819	0.80148

Source: authors' calculations

Figure 19: Relative MSE of EML models over time - IP



Source: authors' calculations

To start, when evaluating MSE, each EML model demonstrates superior performance compared to the benchmark. Bagging exhibits an MSE ranging from 0.67 to 0.70, Boosting

ranges from 0.66 to 0.83, and Random Forest ranges from 0.73 to 0.77. Notably, Bagging emerges as the top-performing model, except for $h = 1$, where Boosting marginally outperforms it.

Secondly, the relative MAE results reinforce our earlier findings, as they also indicate lower errors compared to the benchmark. Moreover, the magnitude of outperformance remains consistent for Bagging, as depicted in Figure 19. Figure 19 also illustrates an increase in forecast errors for Boosting and a slight decrease for Random Forest as the forecast horizon increases.

Table 23 presents the *p*-values from the DM test. It's noteworthy that every individual EML model exhibits statistically significant outperformance compared to the benchmark at the 5% significance level across all forecast horizons.

Table 23: DM test *p*-values of EML methods - Inf

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Bagging	0.00318**	0.00246**	0.00215**	0.00228**	0.00701**
Boosting	0.00155**	0.01021**	0.01181**	0.04125**	0.01602**
Random Forest	0.01956**	0.02505**	0.01219**	0.02601**	0.04646**

Source: authors' calculations

Table 24 highlights the outperformance of Bagging, ranging between 30.02% and 34.79%. This achievement is particularly noteworthy as it surpasses the performance of Lasso and closely rivals that of Ridge, falling just short of the Elastic Net model's superiority. Among the other EML models, Boosting demonstrates an outperformance ranging from 17.44% to 33.83%, while Random Forest shows a narrower range of improvement, between 23.22% and 26.96%. Interestingly, Boosting's strongest outperformance occurs at $h = 1$ but diminishes as h increases, whereas Random Forest exhibits a slight increase in outperformance over the forecast horizons.

Table 24: % improvement of EML forecasts over the benchmark - Inf

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Bagging	32.41%	30.97%	32.84%	32.02%	34.79%
Boosting	33.83%	26.89%	26.48%	17.44%	18.23%
Random Forest	23.67%	23.22%	25.06%	25.81%	26.96%

Source: authors' calculations

Progressing to DA, each EML model demonstrates a greater likelihood of correctly capturing the direction of change in the target variable compared to the benchmark model. The findings are presented in Table 25. The directional accuracy of Bagging ranges from 70.48% to 73.04%, while that of Boosting ranges from 65.71% to 70.42%, and Random Forest ranges from 68.57% to 70.00%. In contrast, the benchmark model's directional accuracy varies from

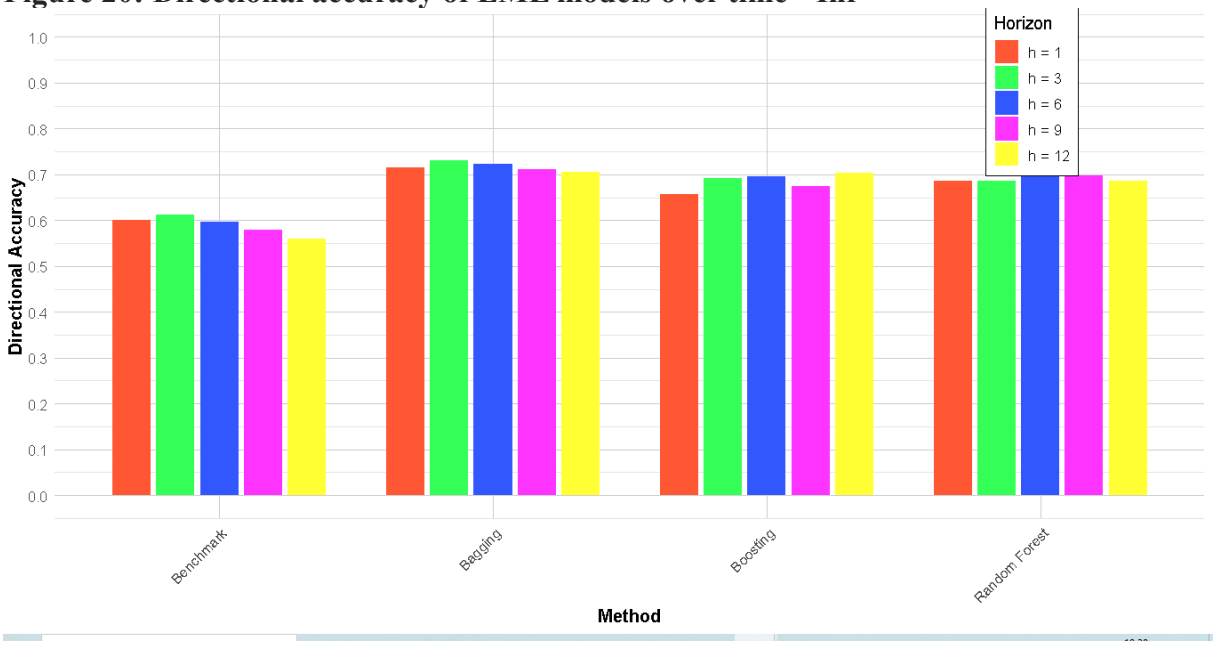
56.07% to 61.27%, depending on the forecast horizon. Moreover, this outperformance is consistent over time, as depicted in Figure 20.

Table 25: Directional accuracy of EML models - Inf

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	60.00%	61.27%	59.74%	57.89%	56.07%
Bagging	71.43%	73.04%	72.31%	71.15%	70.48%
Boosting	65.71%	69.12%	69.49%	67.38%	70.42%
Random Forest	68.57%	68.63%	70.00%	69.71%	68.64%

Source: authors' calculations

Figure 20: Directional accuracy of EML models over time - Inf



Source: authors' calculations

In Table 26, we present the results of the SVM, FFNN and LSTM models. Firstly, we acknowledge that computational resource constraints rendered the Neural Network models impractical for our analysis, as elaborated in Subchapter 5.1.1, thus we won't delve into further details here. Moving forward, we direct our attention to the SVM model, which exhibits notably stronger performance in forecasting inflation compared to industrial production. While its relative MSE is higher compared to previously described models in this subchapter, the near-identical values of relative MSE and MAE indicate that the model adeptly captures significant data patterns and can accommodate potentially large forecast errors without substantial impact. Additionally, these results hold high statistical significance at the 5% level across all horizons.

Regarding DA, the SVM model achieves values ranging from 68.77% to 74.15%, indicating its outperformance compared to the benchmark model. Its performance generally aligns with that of the Ridge and Bagging models, slightly trailing behind the Elastic Net model

while surpassing Lasso, Boosting, and Random Forest.

Table 26: Forecasts with SMV, FFNN and LSTM - Inf

Relative MSE	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Support Vector Machine	0.83966	0.85146	0.85869	0.85031	0.82470
Feedforward NN	458.15	451.18	438.00	420.98	418.35
LSTM NN	5716.14	1617.63	1219.65	477.68	637.59
Relative MAE					
Benchmark	1	1	1	1	1
Support Vector Machine	0.86623	0.87089	0.87021	0.86015	0.86015
Feedforward NN	27.9724	27.7361	27.2919	26.7198	26.7198
LSTM NN	76.7502	42.9465	35.6332	21.4885	21.4885
DM test					
Support Vector Machine	0.00709	0.01365	0.01686	0.01555	0.00413
Feedforward NN	1	1	1	1	1
LSTM NN	1	1	0.99997	0.99998	0.99905
DA					
Benchmark	60.00%	61.27%	59.74%	57.89%	56.07%
Support Vector Machine	68.57%	70.59%	73.08%	73.48%	74.15%
Feedforward NN	44.29%	45.10%	45.90%	46.42%	46.75%
LSTM NN	48.57%	37.75%	54.62%	57.89%	53.25%

Source: authors' calculations

To summarize, in this subchapter we forecast inflation using hard macroeconomic data. Most of the machine learning models exhibit significant outperformance compared to the benchmark model, with the exception of the neural network models due to computational limitations. The outperformance is lesser in magnitude compared to industrial production, but it is highly statistically significant at each horizon. The models, sorted from best to worst in terms of % improvement over the benchmark, are presented in Table 27.

Firstly, among the models evaluated, Ridge and Elastic Net stand out as the top performers, with Elastic Net being number one on each horizon, demonstrating statistically significant outperformance of the benchmark model at every forecasting horizon at the 5% significance level. Additionally, both Ridge and Elastic Net models exhibit the highest directional accuracy values across all models, with consistent performance over time. Notably, in contrast to the good performance of Lasso in the case of industrial production, its

performance deteriorates substantially and it remains one of the worst performing models, surpassing only the SVM. The outperformance of the Lasso model is also statistically significant on every *h* at the 5% level though.

Secondly, EML models, while displaying slightly higher errors and lower directional accuracy compared to Ridge and Elastic Net, are still capable of achieving statistically significant outperformance of the benchmark model. What is more, Bagging surpasses Lasso in terms of DA. Bagging emerges as the overall top performer except for $h = 1$, showcasing notable effectiveness in capturing nonlinear relationships in the data.

Thirdly, the SVM model yields far more compelling results compared to industrial production. It is still the weakest model, but the outperformance becomes statistically significant on every *h*. Moreover, it is the most robust model in terms of influential forecast errors, as it produces really similar relative MSE and MAE values.

Table 27: Forecasting models from best to worst - Inf

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
1.	Elastic Net	Elastic Net	Elastic Net	Elastic Net	Elastic Net
2.	Ridge	Ridge	Ridge	Ridge	Ridge
3.	Boosting	Bagging	Bagging	Bagging	Bagging
4.	Bagging	Boosting	Boosting	RF	RF
5.	RF	RF	RF	Lasso	Boosting
6.	Lasso	Lasso	Lasso	Boosting	Lasso
7.	SVM	SVM	SVM	SVM	SVM
8.	FFNN	FFNN	FFNN	FFNN	FFNN
9.	LSTM	LSTM	LSTM	LSTM	LSTM

Source: authors' calculations

4.1.5 Forecasting inflation using common factors

Table 28 presents the relative MSE values of the inflation forecasts derived from common factors. Table 28 underscores that forecasts derived from common factors continue to outperform the benchmark model, with no relative MSE exceeding one. Within the RLS methods, the performance of the Lasso model experiences a decline compared to industrial production forecasting, while Elastic Net maintains satisfactory results. Notably, the Ridge model demonstrates a significant improvement and emerges as the top-performing RLS specification. This is likely the result of the reduced information set, where shrinkage becomes more pronounced in contrast to variable selection.

Table 28: Relative MSE of forecasts based on common factors - Inf

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	1	1	1	1	1
Lasso	0.91935	0.92376	0.93345	0.93323	0.91158
Ridge	0.52143	0.51825	0.52280	0.51764	0.50053
Elastic Net	0.60710	0.61172	0.60056	0.59217	0.56605
Bagging	0.82478	0.83360	0.85232	0.84480	0.80954
Boosting	0.93111	0.93141	0.87634	0.86724	0.94901
Random Forest	0.82074	0.84776	0.81986	0.76227	0.71753
SVM	0.70042	0.71128	0.71516	0.70507	0.69529

Source: authors' calculations

Turning to EML methods, Bagging and Random Forest exhibit nearly identical performances, with the exception of $h = 12$, where the latter marginally outperforms the former by almost 0.08. Boosting, however, fares less favorably compared to these two. Furthermore, the Support Vector Machine (SVM) model's results demonstrate a notable improvement, compared to its performance on hard data.

Table 29: % difference between forecasts based on factors and hard data - Inf

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Lasso	7.93%	9.57%	10.37%	13.06%	10.81%
Ridge	-20.62%	-22.27%	-20.69%	-21.15%	-21.03%
Elastic Net	6.33%	-2.24%	-3.20%	8.21%	-5.30%
Bagging	22.03%	20.75%	26.90%	21.03%	22.67%
Boosting	40.72%	27.41%	19.19%	5.04%	16.06%
Random Forest	7.53%	10.42%	9.40%	2.75%	-1.77%
SVM	-16.58%	-16.46%	-16.72%	-17.08%	-15.69%

Source: authors' calculations

Table 29 presents the percentage difference between the relative MSE values of forecasts based on hard macroeconomic data and those based on common factors. Primarily, the table reveals that in most instances, models based on hard data exhibit greater accuracy. Nonetheless, there are a few exceptions. Firstly, the Ridge model based on common factors surpasses its counterpart based on hard macroeconomic data. Additionally, Elastic Net demonstrates superior performance at certain horizons compared to forecasts based on hard data, albeit with small percentage differences ranging between 2.24% and 5.30%. Furthermore, the Support Vector Machine (SVM) model consistently yields lower errors compared to the model forecast based on hard data, indicating its robust performance. This result is in line with the SVM result of Table 14.

In Table 30, we present the directional accuracy of forecasts based on common factors. This table further reinforces our conclusion that hard macroeconomic data should be prioritized

for forecasting purposes. While directional accuracy values based on common factors generally exceed the benchmark, they typically fall short compared to results derived from hard macroeconomic data. Notably, there are instances, such as from $h = 1$ to $h = 9$, where the benchmark outperforms Boosting in terms of directional accuracy, a scenario not observed when using real data.

Table 30: Directional accuracy of forecasts based on common factors - Inf

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	60.00%	61.27%	59.74%	57.89%	56.07%
Lasso	64.29%	65.69%	66.15%	66.13%	66.10%
Ridge	67.14%	67.16%	67.44%	67.56%	67.09%
Elastic Net	68.57%	67.16%	68.72%	68.64%	68.50%
Bagging	68.57%	71.57%	70.77%	69.53%	69.49%
Boosting	58.57%	60.29%	59.74%	57.53%	62.29%
Random Forest	64.29%	64.22%	62.56%	62.19%	65.11%
SVM	60.00%	61.27%	59.74%	57.89%	56.07%

Source: authors' calculations

4.1.6 Forecasting inflation using soft indicators

The results of machine learning and benchmark model forecasts based on these indicators are presented in Table 31. As anticipated, the table reveals that indicators are not the most suitable source of information when aiming to capture the magnitude of changes in macroeconomic variables. This is primarily because they yield forecasts similar to the benchmark model, which necessitates significantly less data and computational resources. It's worth noting that the SVM and Elastic Net models have no relative MSE over 1. However, all values are so close to 1 that they are practically indistinguishable.

Table 31: Relative MSE of forecasts based on indicators - Inf

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Lasso	0.99920	1.00930	1.02142	1.01731	0.97916
Ridge	1.00141	1.00944	1.02160	1.01633	0.97927
Elastic Net	0.94591	0.95314	0.96228	0.95526	0.91147
Bagging	1.16871	1.20947	1.18994	1.18321	1.14751
Boosting	1.16607	1.11865	1.06712	1.09477	1.07865
Random Forest	1.00079	0.99324	1.00011	1.00889	0.99492
SVM	0.94591	0.95053	0.96228	0.95526	0.91147

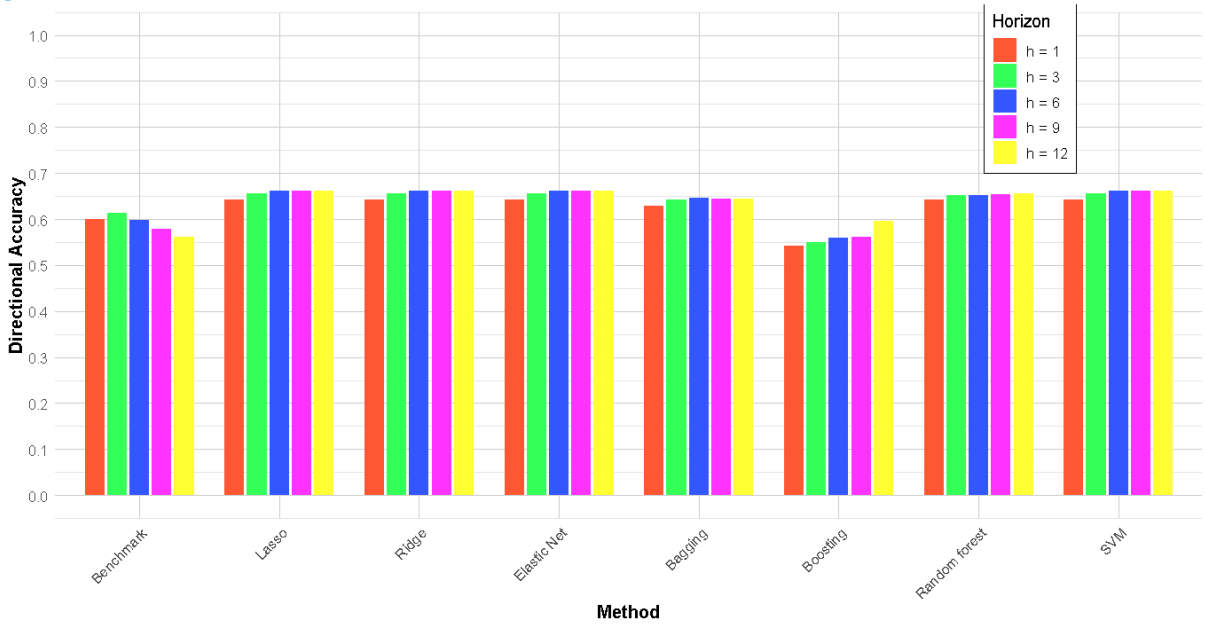
Source: authors' calculations

Table 32: Directional accuracy of forecasts based on indicators - Inf

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	60.00%	61.27%	59.74%	57.89%	56.07%
Lasso	64.29%	65.69%	66.15%	66.13%	66.10%
Ridge	64.29%	65.69%	66.15%	66.13%	66.10%
Elastic Net	64.29%	65.69%	66.15%	66.13%	66.10%
Bagging	62.86%	64.22%	64.62%	64.52%	64.41%
Boosting	54.29%	54.90%	55.90%	56.09%	59.60%
Random Forest	64.29%	65.20%	65.13%	65.41%	65.68%
SVM	64.29%	65.69%	66.15%	66.13%	66.10%

Source: authors' calculations

Figure 21: Directional accuracy of forecasts based on indicators - Inf



Source: authors' calculations

Continuing our analysis, we delve into directional accuracy, our primary measure of interest when utilizing indicator data. The directional accuracy values are presented in Table 32. Several notable observations emerge. Firstly, every machine learning model consistently outperforms the benchmark model across all forecast horizons, except for Boosting. Within the RLS models, there is no discernible difference in directional accuracy values, rendering the choice between them irrelevant. Similarly, EML models generally outperform the benchmark, except for Boosting, which exhibits lower directional accuracy values. While one model underperforms, the others align with expectations. Secondly, the level of outperformance is smaller compared to when using hard macroeconomic data. This outcome was expected, as forecasts based on indicator data typically yield lower directional accuracy values. However, it's crucial to acknowledge that indicators offer the advantage of earlier availability and capturing sentiments. Thirdly, the consistent outperformance is evident, as depicted in Figure

21, albeit to a lesser extent than in Subchapter 4.1.3. Fourthly, there are no substantial differences between models, suggesting that selecting the model with the least computational capacity (in this case, Lasso) suffices for computing directional accuracy.

In conclusion, indicators effectively serve their purpose. Through leveraging an extensive dataset of soft indicators and employing machine learning models, we can predict the direction of change in inflation with a higher likelihood than if we relied solely on the benchmark model.

4.2 Post-COVID

4.2.1 Forecasting industrial production using hard macroeconomic data – post-COVID

In this subchapter, we utilize our revised dataset, which now encompasses data from the COVID era, hence referred to as the post-COVID dataset. We commence our analysis by delineating the relative MSE and MAE outcomes of the forecasts pertaining to industrial production, derived from hard macroeconomic data. Table 33 presents the results of the RLS models. Firstly, in terms of MSE, all RLS models demonstrate superior performance compared to the benchmark. Both Lasso and Elastic Net models exhibit a relative MSE of approximately 0.02, indicating that the Lasso component within the Elastic Net model carries a higher weight in comparison to the Ridge component. On the other hand, the Ridge model also surpasses the benchmark by maintaining a relative MSE ranging between 0.15 to 0.16 for every forecast horizon. This indicates that variable selection emerges as a crucial factor, surpassing coefficient shrinkage in significance. To highlight the differences between shrinkage and selection, we present the selected variables from Tables A1.1 – A1.8. on Figures 22-24.

Table 33: Relative MSE and MAE of RLS models - IP

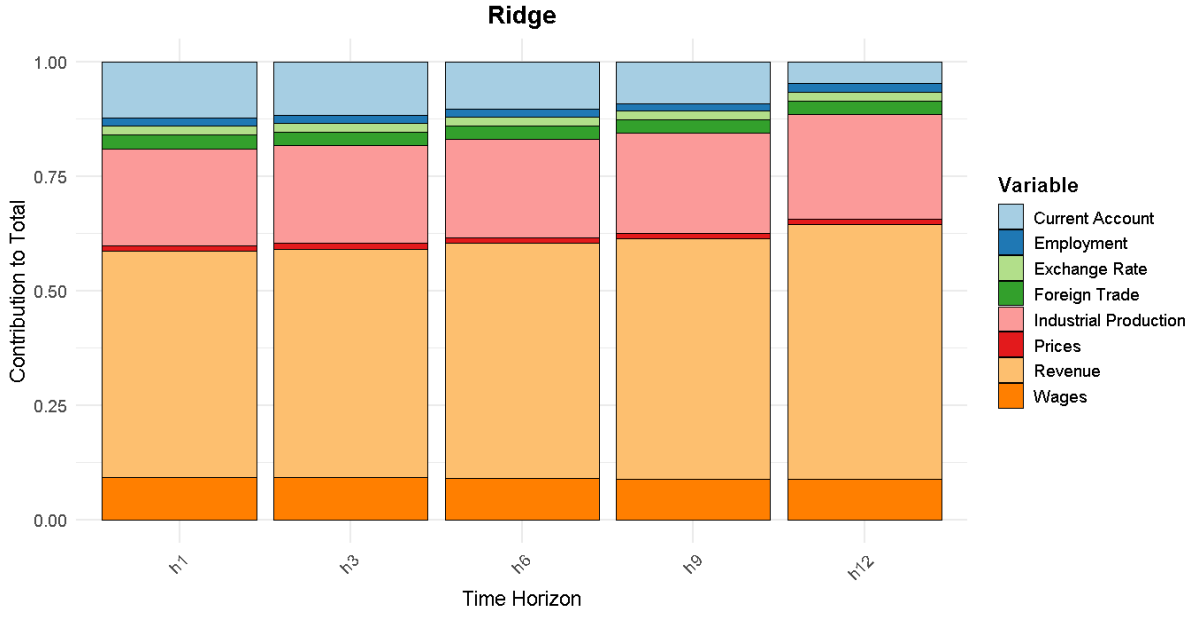
Relative MSE	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	1	1	1	1	1
Lasso	0.01708	0.01690	0.01651	0.01485	0.01421
Ridge	0.16493	0.15828	0.15719	0.15798	0.15508
Elastic Net	0.02293	0.02321	0.02341	0.02309	0.02210
Relative MAE					
Benchmark	1	1	1	1	1
Lasso	0.19054	0.19100	0.18811	0.17607	0.17197
Ridge	0.53807	0.53016	0.53287	0.52925	0.51732
Elastic Net	0.21986	0.22374	0.22300	0.22138	0.21563

Source: authors' calculations

Variable importance is calculated according to Kock et al. (2020). As there are a high number of predictors, we group them. For each forecasting horizon, the Figures 22-24 display the average estimated coefficient across the rolling windows for each group. Before averaging, the coefficients are adjusted by multiplying them by the standard deviation of the variables to ensure comparability. The resulting importance measures are then rescaled to sum up to one. Since the number of variables in each group varies significantly, the importance measures are further divided by the number of variables in the respective group for normalization. The best models show remarkable simplicity, as expected. Both Lasso and Elastic Net place the highest weight on the 32 variables from the category of industrial production, presented in Table A1.1. This pattern is remarkably consistent over the different horizons, keeping the relative importance of other variable groups at a low level. In contrast, Ridge places the highest weight on the revenue category, which slightly increases with *h*. It also places some weight on wages and the current account, with the latter showing a decreasing tendency. While the other models also incorporate variables from the revenue category, they do it to a much lesser extent.

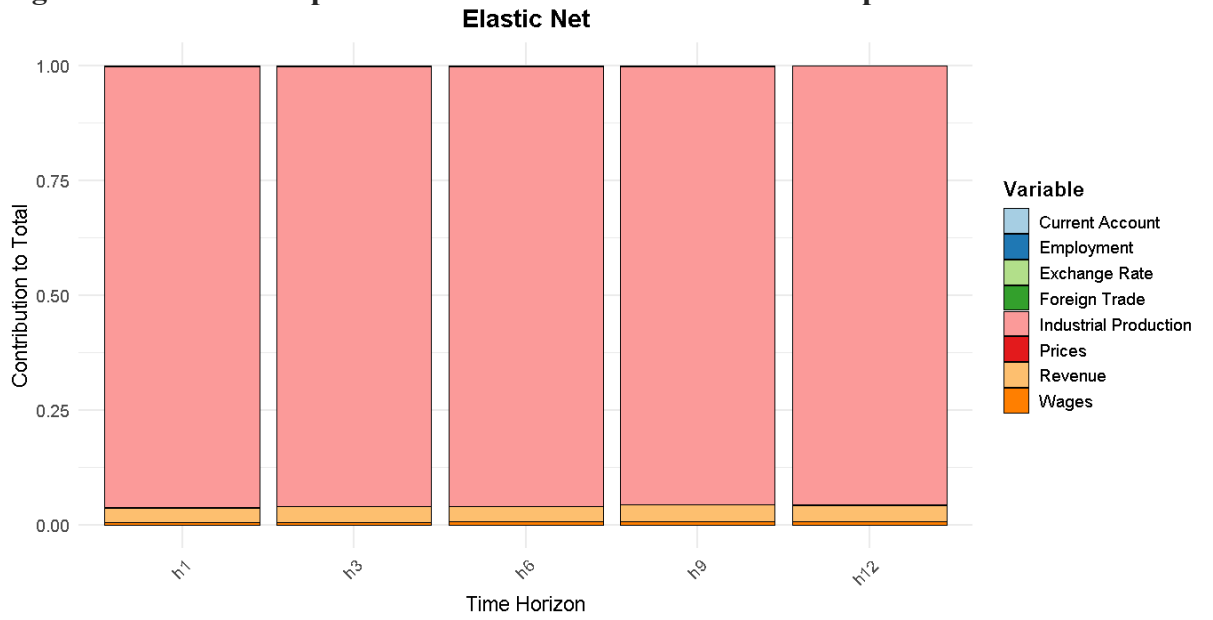
Secondly, the relative MAE outcomes reinforce our prior deduction, showcasing smaller errors compared to the benchmark model. This also signifies the higher performance of the variable selection models, for in absolute terms, Ridge exhibits the largest difference between relative MSE and MAE values, indicating its sensitivity to larger forecast errors. This signifies that all RLS models outshine the benchmark, with errors notably smaller than those observed in the pre-COVID dataset, presented in Table 3.

Figure 22: Variable importance of the Ridge model – IP – post-COVID



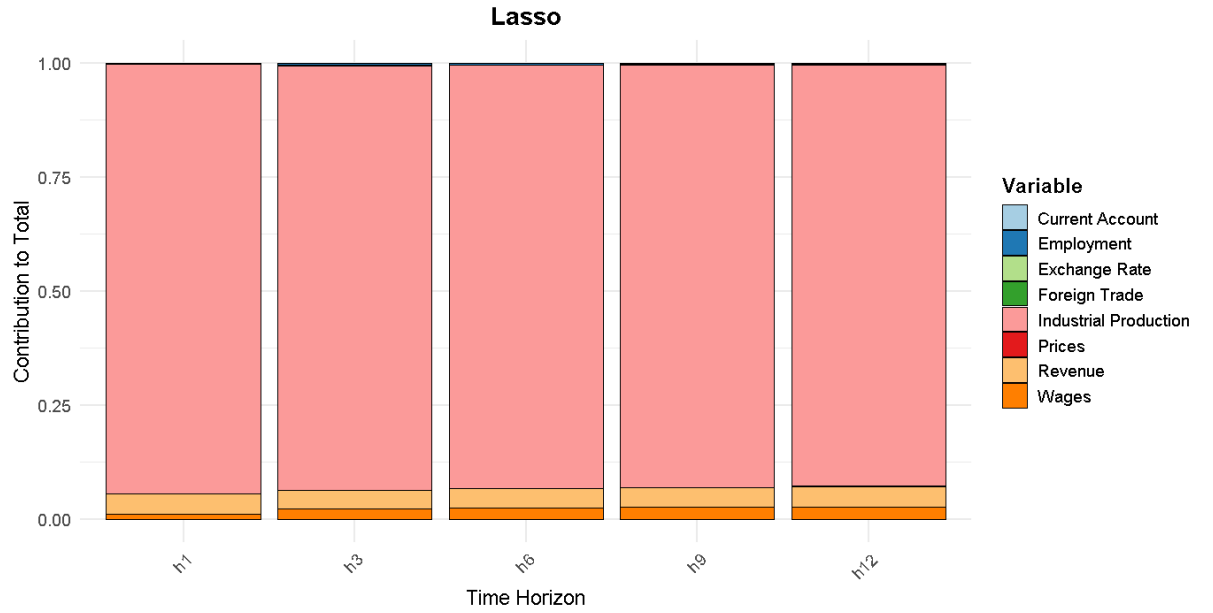
Source: authors' calculations

Figure 23: Variable importance of the Elastic Net model – IP – post-COVID



Source: authors' calculations

Figure 24: Variable importance of the Lasso model – IP – post-COVID



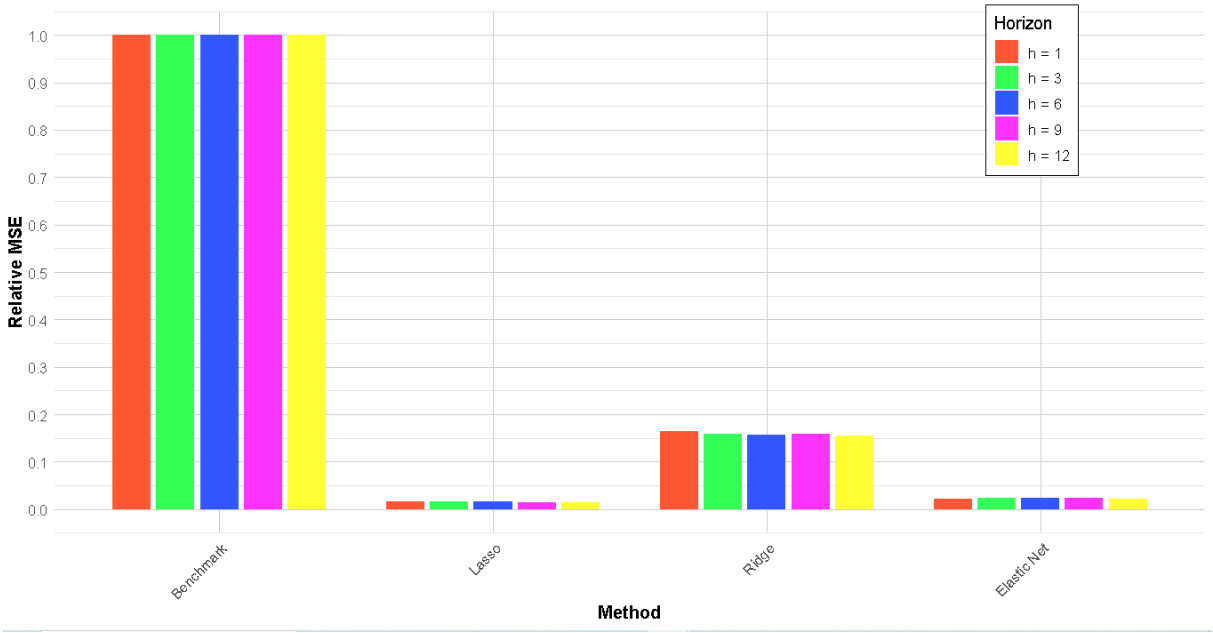
Source: authors' calculations

The magnitude of outperformance, presented on Figure 25, shows remarkable consistency similar to of Figure 12.

To shed further light on the precision of these methods, we compare the out-of-sample actual and forecasted values on Figure 26 for $h = 1$ and Figure 27 for $h = 12$. Regardless of time horizon, both the Lasso and Elastic Net models demonstrate a remarkable match in terms of both general direction and magnitude. This includes accurately capturing large spikes in the

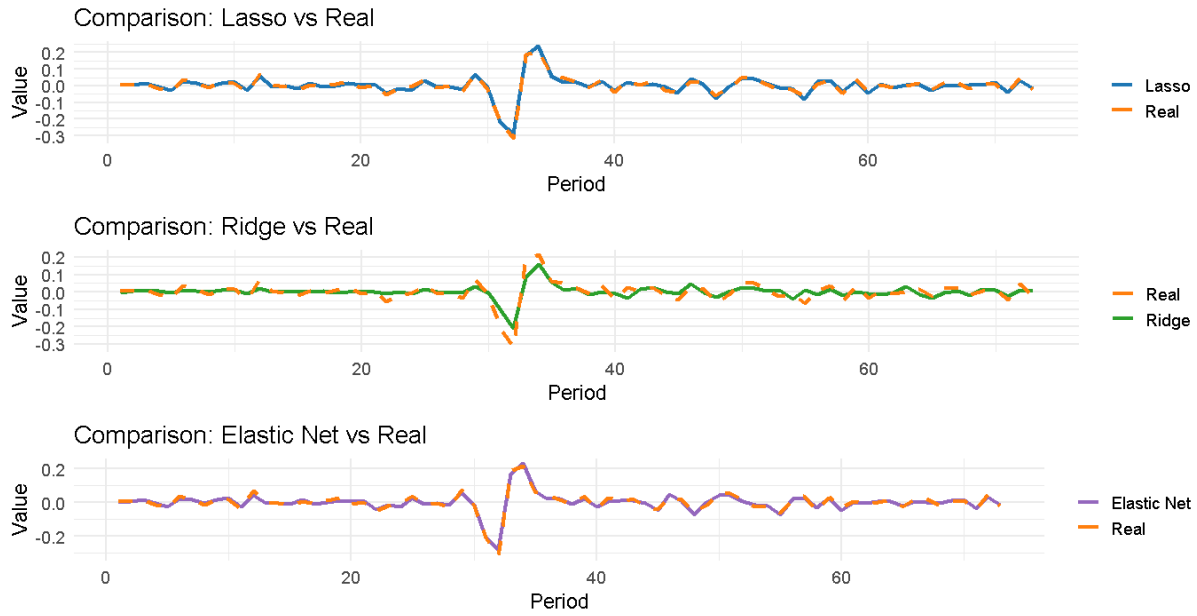
data. While the Ridge model correctly predicts the direction, it tends to underperform in terms of magnitude, particularly evident in slightly overestimating the impact of large spikes.

Figure 25: Relative MSE of RLS models over time – IP – post-COVID



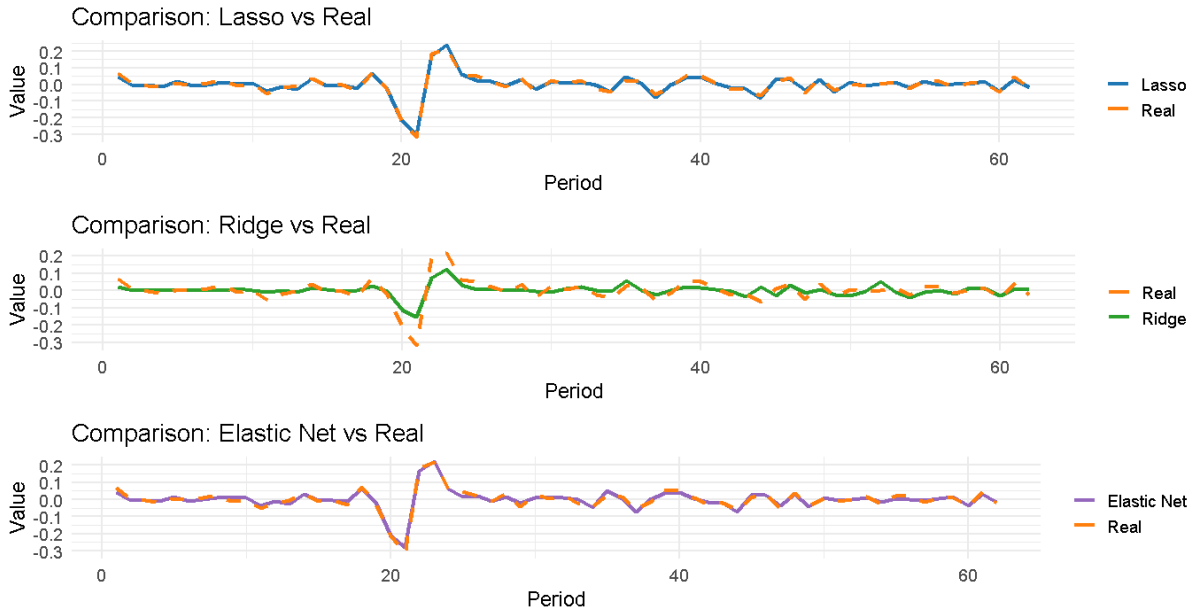
Source: authors' calculations

Figure 26: Actual and forecasted values of RLS models on $h = 1$ – IP – post-COVID



Source: authors' calculations

Figure 27: Actual and forecasted values of RLS models on $h = 12$ – IP – post-COVID



Source: authors' calculations

To add, Table 34 presents the DM statistic *p*-values, showcasing that all RLS models significantly outperform the benchmark at the 5% level.

Table 34: DM test *p*-values of RLS methods – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Lasso	0.01547**	0.01637**	0.0167**	0.01672**	0.01698**
Ridge	0.02481**	0.02529**	0.02625**	0.02648**	0.02591**
Elastic Net	0.00664**	0.01321**	0.01174**	0.04661**	0.00935**

Source: authors' calculations

Since the outperformance is statistically significant, we can express the magnitude of outperformance in percentage terms. Table 35 illustrates that the Lasso model consistently outperforms the benchmark by approximately 98.00% across all *h*. Similarly, the Ridge model generally surpasses the benchmark by 84.00%, while the Elastic Net model, not surprisingly, exhibits a very close performance to Lasso, with an outperformance of about 97.00% across all horizons. Notably, both Lasso and Elastic Net models achieve an increase in performance of approximately 20 percentage points compared to the pre-COVID sample. Multiple reasons can be given for this outperformance. Generally, the models have more data to learn with the prolongation of the time series by 4 years. To add, the benchmark model's performance can deteriorate substantially as the simple model might not be able to capture an abrupt and unexpected shock such as COVID. In conclusion, regardless of the model under consideration, the improvement in percentage terms is substantial, highlighting the efficacy of the chosen modeling approaches.

Table 35: % improvement of RLS forecasts over the benchmark – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Lasso	98.29%	98.31%	98.35%	98.51%	98.58%
Ridge	83.51%	84.17%	84.28%	84.21%	84.49%
Elastic Net	97.71%	97.68%	97.66%	97.69%	97.79%

Source: authors' calculations

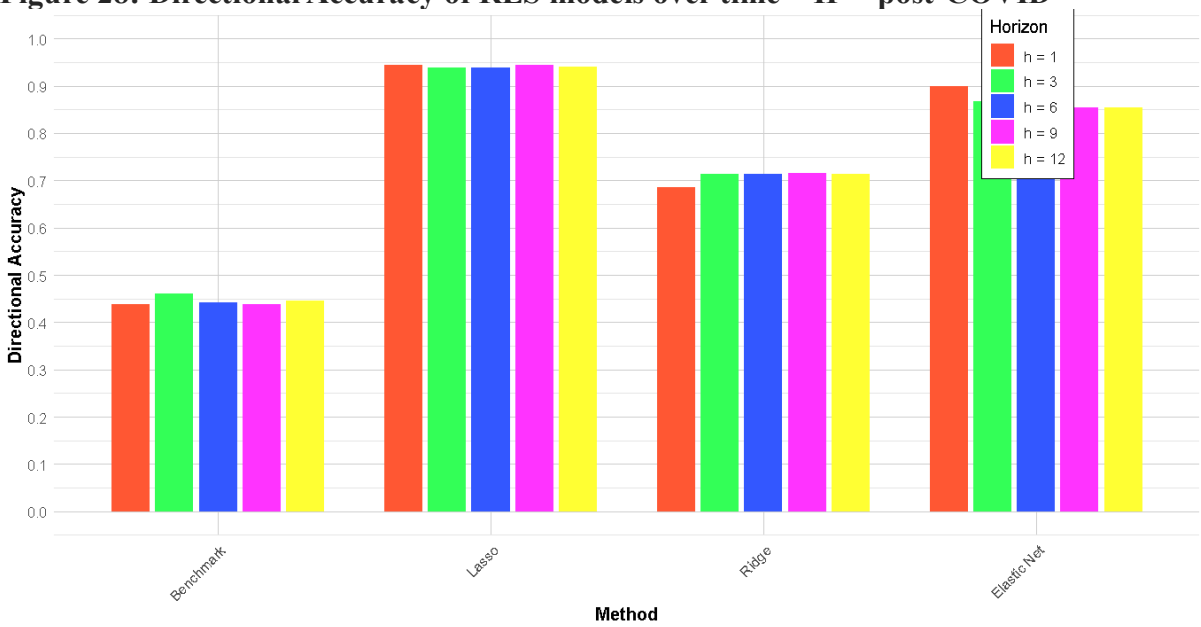
To finish the analysis of the RLS model errors, DA values are presented in Table 36. Lasso is the best performing with DA values between 93.87% and 94.52% compared to the benchmark, which is accurate between 43.76% and 44.62% of the time. Elastic Net performs slightly worse than Lasso, but still has relative MSE values between 85.45% and 90.00%, which is a more than twofold increase compared to the benchmark, while Ridge is the weakest model with DA values between 68.49% and 71.62%.

Table 36: Directional accuracy of RLS models – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	43.84%	46.01%	44.12%	43.76%	44.62%
Lasso	94.52%	93.90%	93.87%	94.36%	94.09%
Ridge	68.49%	71.36%	71.32%	71.62%	71.37%
Elastic Net	90.00%	86.76%	87.44%	85.48%	85.45%

Source: authors' calculations

Figure 28: Directional Accuracy of RLS models over time – IP – post-COVID



Source: authors' calculations

Furthermore, the results are consistent irrespective of h, as shown in Figure 28. In contrast to the pre-COVID sample, this time the benchmark model is consistent as well.

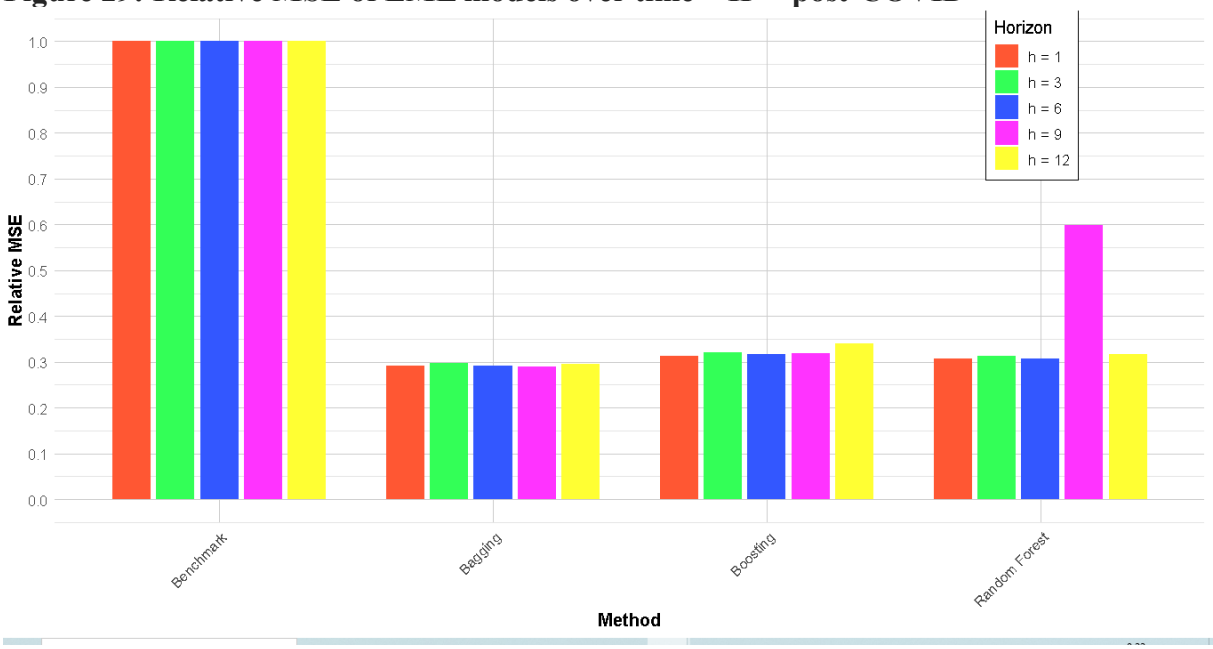
We proceed by presenting the results of the EML models in Table 37.

Table 37: Relative MSE and MAE of EML models – IP – post-COVID

Relative MSE	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Bagging	0.29142	0.29817	0.29136	0.28997	0.29515
Boosting	0.31343	0.32121	0.31778	0.31932	0.34098
Random Forest	0.30690	0.31272	0.30703	0.59819	0.31621
Relative MAE					
Benchmark	1	1	1	1	1
Bagging	0.41735	0.42114	0.43231	0.41821	0.42341
Boosting	0.46399	0.48096	0.47895	0.47221	0.48446
Random Forest	0.48173	0.48949	0.49292	0.80020	0.49943

Source: authors' calculations

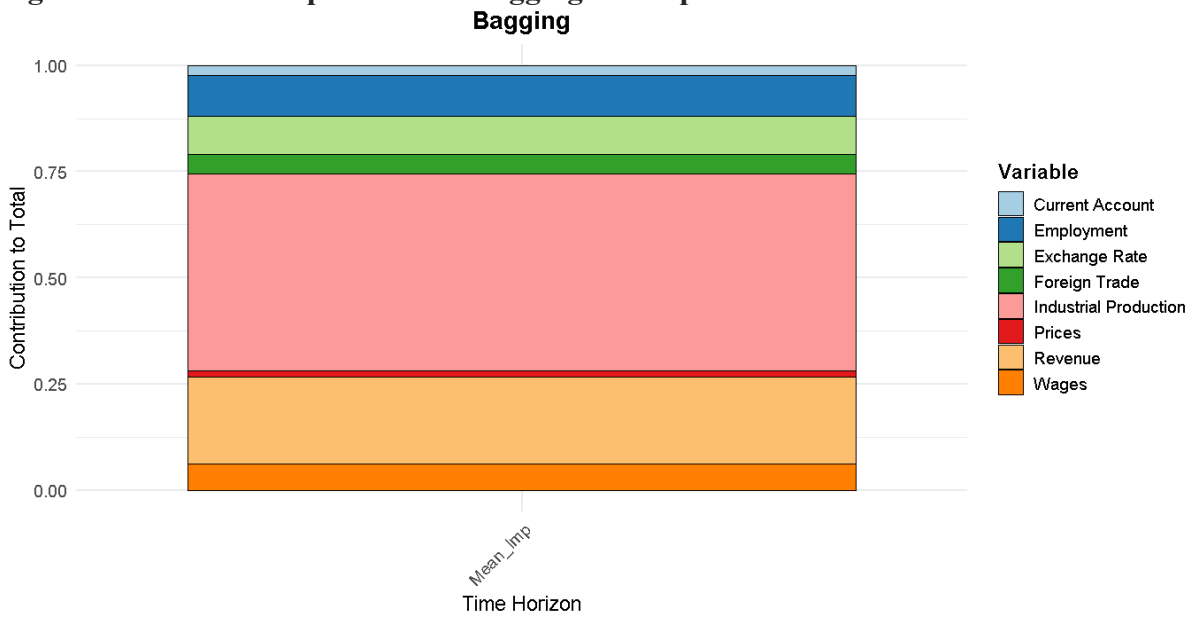
Figure 29: Relative MSE of EML models over time – IP – post-COVID



Source: authors' calculations

Firstly, in terms of MSE, every EML model outperforms the benchmark. Bagging demonstrates an MSE of around 0.29, with Boosting ranging between 0.31 and 0.34, and Random Forest between 0.30 and 0.32, except at $h = 9$, where it increases to 0.60. This indicates that Bagging performs best for every horizon, closely followed by both Random Forest and Boosting. Random Forest performs better than Boosting at every horizon except for $h = 9$. The relative MAE values support this conclusion and Figure 29 shows the consistency of relative MSE across different *h*. For EML methods, variable importance is computed similarly than for RLS methods, but it is averaged across horizons as well. Figure 30 presents the importance of each category for Bagging. The rest of the EML models – as expected based on Table 37 – assign virtually identical importances to variable categories, hence we only present Bagging.

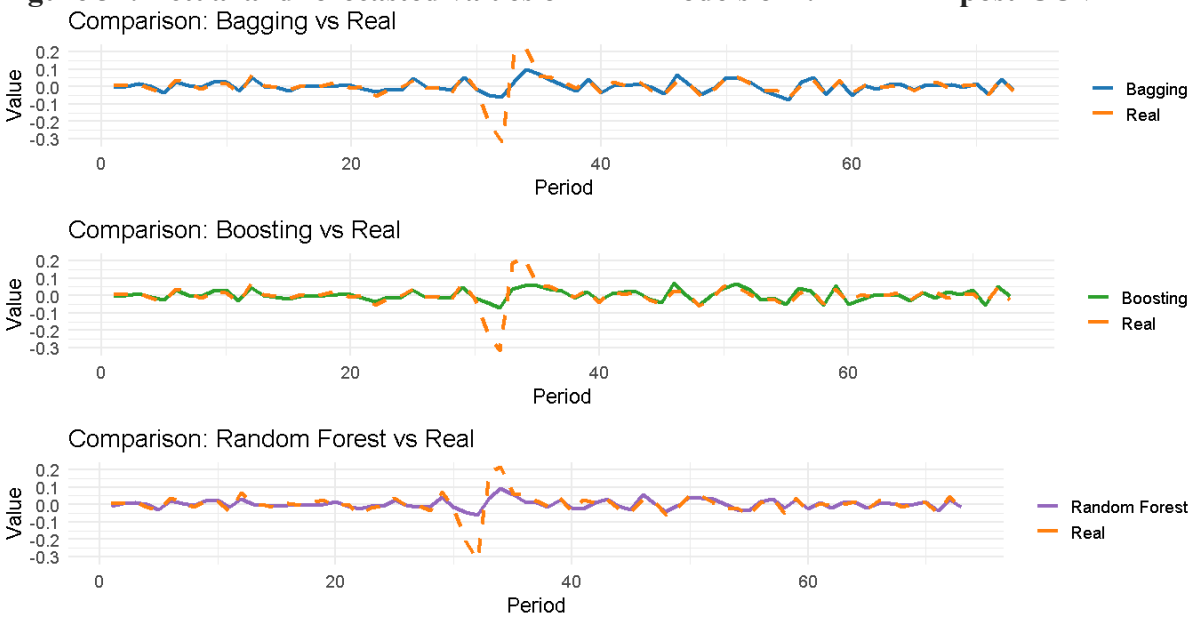
Figure 30: Variable importance in Bagging – IP – post-COVID



Source: authors' calculations

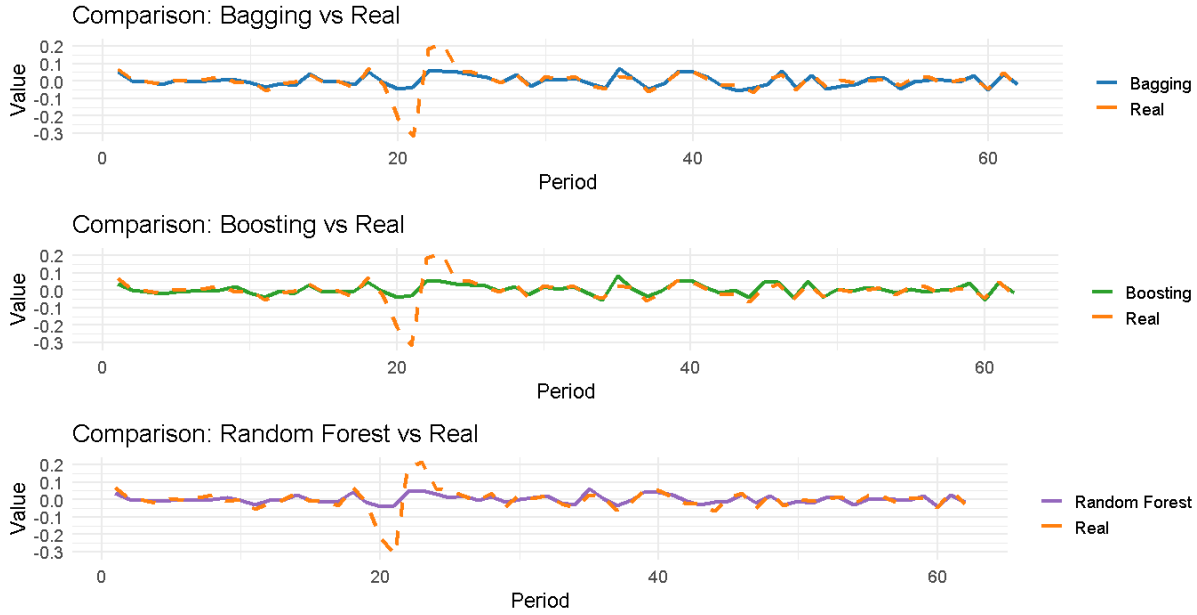
In line with RLS models (with the exception of Ridge), Bagging assigns the highest weight to the variables in Table A1.1. In contrast to Elastic Net and Lasso, it also weighs variables from other categories, with revenue being the second most important category, followed by employment and wages. Foreign trade and exchange rates play a bit larger, while current account plays a bit smaller role. The effect of prices on the outcome of forecast is minimal.

Figure 31: Actual and forecasted values of EML models on $h = 1$ – IP – post-COVID



Source: authors' calculations

Figure 32: Actual and forecasted values of EML models on $h = 12$ – IP – post-COVID



Source: authors' calculations

To provide additional insight into the accuracy of these methods, we compare the out-of-sample actual and forecasted values in Figure 31 for $h = 1$ and Figure 32 for $h = 12$. Across both time horizons, all three models closely align with the actual values, demonstrating a good match in terms of both general direction and magnitude, at least when the data is smooth. The models correctly capture the direction of the large spikes in the real data, but they cannot exactly capture their magnitude.

To continue with, Table 38 shows that every model outperforms the benchmark statistically significantly at the 5% level.

Table 38: DM test *p*-values of EML methods – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Bagging	0,01136**	0,01248**	0,01299**	0,01333**	0,01385**
Boosting	0,01544**	0,01726**	0,01865**	0,01690**	0,01941**
Random Forest	0,01311**	0,01382**	0,01533**	0,04373**	0,01501**

Source: authors' calculations

With the statistically significant outperformance established, we proceed to quantify the magnitude of outperformance in percentage terms, stating them in Table 39. Table 39 reveals that the outperformance of Bagging ranges between 70.18% and 71.00%, still impressive but slightly less than the best RLS models. Boosting's outperformance ranges between 65.90% and 68.66%, and Random Forest's between 68.38% and 69.30%, except for $h = 9$. Generally, Bagging emerges as the best-performing model, closely followed by both Boosting and Random Forest. The outperformance remains consistent for every model, except for Random

Forest at $h = 9$. In conclusion, while the improvement in percentage terms is substantial regardless of the model considered, it is smaller when compared to the RLS methods. However, the performance of the models has significantly improved compared to the pre-COVID sample, by approximately 20 percentage points on average.

Table 39: % improvement of EML forecasts over the benchmark – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Bagging	70.86%	70.18%	70.86%	71.00%	70.48%
Boosting	68.66%	67.88%	68.22%	68.07%	65.90%
Random Forest	69.31%	68.73%	69.30%	40.18%	68.38%

Source: authors' calculations

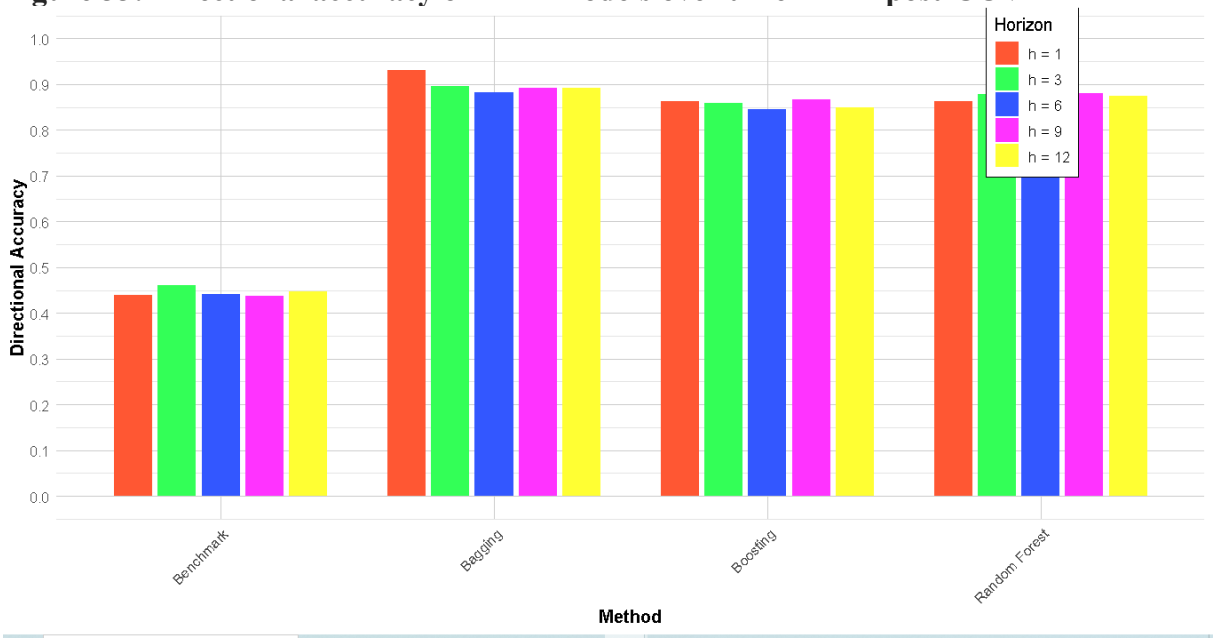
Table 40: Directional accuracy of EML models – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	43.84%	46.01%	44.12%	43.76%	44.62%
Bagging	93.15%	89.67%	88.24%	89.23%	89.25%
Boosting	86.30%	85.92%	84.56%	86.67%	84.95%
Random Forest	86.30%	87.79%	87.99%	88.03%	87.50%

Source: authors' calculations

Moving forward to DA, every EML model demonstrates a higher likelihood of capturing the correct direction of change in the target variable compared to the benchmark model. The results are presented in Table 40. Bagging achieves DA percentages between 88.24% and 93.15%, Boosting ranges from 84.56% to 86.67%, and Random Forest ranges from 86.30% to 88.03%. In contrast, the benchmark model exhibits a DA between 43.76% and 46.01%, depending on the forecast horizon. Figure 33 outlines the consistency of DA at different *h*.

Figure 33: Directional accuracy of EML models over time – IP – post-COVID



Source: authors' calculations

In Table 41, we exclusively present the results of the Support Vector Machine (SVM) model. The Feedforward Neural Network and the Long-Short Term Memory Neural Network are omitted due to their poor performance in the previous sample, which has not significantly improved with the introduction of new data. SVM remains superior even in this new sample post-COVID, with smaller errors compared to the pre-COVID sample. This observation is corroborated by the relative MAE values. However, the DA values of the SVM model range between 53% and 55%, lower than those of the other models.

Table 41: Forecasts with SMV, FFNN and LSTM – IP – post-COVID

Relative MSE	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Support Vector Machine	0.60012	0.600908	0.59867	0.31037	0.59862
Relative MAE					
Benchmark	1	1	1	1	1
Support Vector Machine	0.80242	0.80509	0.80283	0.49318	0.80266
DA					
Benchmark	43.84%	46.01%	44.12%	43.76%	44.62%
Support Vector Machine	54.79%	54.46%	54.17%	53.85%	53.36%

Source: authors' calculations

Table 42: Combined RLS and EML methods – IP – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Lasso + Bagging	60.68%	59.74%	61.27%	63.24%	62.53%
Lasso + Boosting	53.20%	55.63%	52.93%	51.00%	51.57%
Lasso + Random Forest	65.38%	64.50%	64.52%	66.76%	66.40%
EN + Bagging	59.54%	70.98%	63.46%	63.73%	60.12%
EN + Boosting	51.10%	67.91%	51.34%	52.85%	52.39%
EN + Random Forest	68.17%	71.63%	65.20%	65.37%	63.06%
Ridge + Bagging	70.94%	59.77%	71.72%	72.00%	72.14%
Ridge + Boosting	64.83%	52.44%	68.18%	65.90%	64.12%
Ridge + Random Forest	70.91%	66.64%	71.48%	70.67%	70.52%

Source: authors' calculations

In addition to the previous analysis, we also explore the performance of combining regularization and nonlinear methods, as summarized in Table 42. Similarly to the case of inflation, where such combinations often lead to performance improvements, we observe that

in the context of industrial production forecasting, combining regularization and nonlinear methods improves the performance of nonlinear models. This might be due to the fact that $k > T$, although EML methods should not be affected by this too much, as stated in Chapter 3.

To summarize, in this subchapter we forecast industrial production using hard macroeconomic data. The machine learning models exhibit significant outperformance. The models, sorted from best to worst in terms of % improvement over the benchmark, are presented in Table 43.

Table 43: Forecasting models from best to worst – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
1.	Lasso	Lasso	Lasso	Lasso	Lasso
2.	Elastic Net	Elastic Net	Elastic Net	Elastic Net	Elastic Net
3.	Ridge	Ridge	Ridge	Ridge	Ridge
4.	Ridge + Bagging	EN + RF	Ridge + Bagging	Ridge + Bagging	Ridge + Bagging
5.	Ridge + RF	EN + Boosting	Ridge + RF	Bagging	Ridge + RF
6.	Bagging	Bagging	Bagging	Ridge + RF	Bagging
7.	RF	RF	RF	Boosting	RF
8.	Boosting	Boosting	Boosting	Lasso + RF	Lasso + RF
9.	EN + RF	EN + Boosting	Ridge + Boosting	Ridge + Boosting	Boosting
10.	Lasso + RF	Ridge + RF	EN + RF	EN + RF	Ridge + Boosting
11.	Ridge + Boosting	Lasso + RF	Lasso + RF	EN + Bagging	EN + RF
12.	Lasso + Bagging	Ridge + Bagging	EN + Bagging	Lasso + Bagging	Lasso + Bagging
13.	SVM	Lasso + Bagging	Lasso + Bagging	SVM	EN + Bagging
14.	EN + Bagging	Lasso + Boosting	SVM	EN + Boosting	SVM
15.	Lasso + Boosting	SVM	Lasso + Boosting	Lasso + Boosting	EN + Boosting
16.	EN + Boosting	Ridge + Boosting	EN + Boosting	RF	Lasso + Boosting
17.	FFNN	FFNN	FFNN	LSTM	LSTM
18.	LSTM	LSTM	LSTM	FFNN	FFNN

Source: authors' calculations

To start, the standout performers are the Lasso and Elastic Net models, showcasing statistically significant outperformance of the benchmark model across all forecasting horizons at a 5% significance level. Notably, both Lasso and Elastic Net models exhibit the highest directional accuracy values among all models, maintaining consistent performance over time.

Although the Ridge model's performance is slightly weaker, it consistently ranks as the third-best model across all horizons. In contrast, the performance of EML models presents a less clear picture. Neither of them outperforms their combination with either Ridge or Elastic Net, positioning these latter models at the 4th and 5th places, respectively. This underscores the effectiveness of our hybrid approach in substantially improving the forecasting performance of nonlinear models by selecting appropriate variables or parameter values for training.

Looking ahead, we delve into exploring the forecasting performance of machine learning models based on common factors in the next subchapter.

4.2.2 Forecasting industrial production using common factors – post-COVID

To commence, Table 44 presents the relative MSE values of the forecasts based on common factors. This table reaffirms that forecasts derived from common factors continue to outperform the benchmark model, with no relative MSE exceeding one. Firstly, focusing on RLS methods, both Lasso and Elastic Net yield comparable results to each other. Additionally, the performance of the Ridge model notably improves compared to the forecasts based on hard data, emerging as the best-performing model among the three. Regarding EML methods, Bagging demonstrates superior performance compared to both Boosting and Random Forest, with the latter two yielding fairly similar results. The results of the SVM model remain largely consistent with those in the previous subchapter, with a slight deterioration observed at $h = 9$. Overall, these findings underscore the continued effectiveness of forecasts based on common factors.

Table 44: Relative MSE of forecasts based on common factors – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	1	1	1	1	1
Lasso	0.094219	0.091859	0.091085	0.088662	0.081859
Ridge	0.086534	0.080003	0.079782	0.08465	0.070691
Elastic Net	0.089579	0.086746	0.085341	0.080882	0.085165
Bagging	0.374125	0.374765	0.371353	0.374969	0.368122
Boosting	0.47972	0.472179	0.458188	0.443167	0.45353
Random Forest	0.399788	0.367635	0.428995	0.416814	0.379344
SVM	0.599592	0.600849	0.598138	0.597554	0.597642

Source: authors' calculations

Continuing, we present the percentage difference between the relative MSE values of the forecasts based on hard macroeconomic data and common factors in Table 45. Positive values indicate that the model based on hard data is more accurate than the model based on common

factors, while negative values signify the opposite. Table 45 reveals that in most cases, the models based on hard data exhibit greater accuracy. The only exceptions are the SVM model, where the forecasts are nearly identical, and the Ridge model, which notably performs better based on common factors.

Table 45: % difference between forecasts based on factors and hard data – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Lasso	81.86%	81.59%	81.87%	83.24%	82.64%
Ridge	-90.60%	-97.85%	-97.04%	-86.58%	-119.39%
Elastic Net	74.40%	73.24%	72.57%	71.45%	74.05%
Bagging	22.10%	20.44%	21.54%	22.67%	19.82%
Boosting	34.66%	31.97%	30.64%	27.94%	24.82%
Random Forest	23.23%	14.94%	28.43%	-43.52%	16.64%
SVM	-0.09%	-0.01%	-0.09%	48.06%	-0.16%

Source: authors' calculations

To conclude, we present the directional accuracy of forecasts based on common factors in Table 46. This table further corroborates our findings. Directional accuracy values derived from common factors significantly surpass the benchmark, albeit they are lower compared to the results based on hard macroeconomic data. Nevertheless, these values remain relatively stable across multiple horizons.

Table 46: Directional accuracy of forecasts based on common factors – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	43.84%	46.01%	44.12%	43.76%	44.62%
Lasso	79.45%	79.81%	81.62%	80.34%	79.17%
Ridge	79.45%	81.22%	82.84%	81.20%	81.59%
Elastic Net	78.08%	78.40%	77.70%	77.61%	77.15%
Bagging	60.27%	60.56%	62.01%	61.71%	61.02%
Boosting	58.90%	65.26%	64.46%	62.74%	64.52%
Random Forest	64.38%	60.56%	61.52%	62.05%	59.81%
SVM	54.79%	54.46%	54.17%	53.85%	53.36%

Source: authors' calculations

4.2.3 Forecasting industrial production using soft indicators – post-COVID

In this subchapter, we explore the potential of combining input data from the soft indicator database of NBS and employing machine learning models for forecasting.

Table 47 presents the relative MSE values. Contrary to our initial expectations and the results observed in our pre-COVID sample, we find that the relative MSE of indicator data is

lower than that of the benchmark model. This unexpected outcome could potentially be attributed to the larger sample size, as we have more data available to train the models. While the results are compelling, they do not match the accuracy achieved by forecasts based on hard macroeconomic data. However, it's important to note that indicators are more readily accessible and do not entail a time lag, making them a valuable tool for providing a broad overview of the direction of the economy. If these indicators can not only sketch the direction but also provide insight into the magnitudes of change to some extent, then they offer even greater utility.

Table 47: Relative MSE of forecasts based on indicators – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	1	1	1	1	1
Lasso	0.52173	0.48680	0.49512	0.51500	0.48728
Ridge	0.52504	0.52168	0.54900	0.55666	0.53364
Elastic Net	0.50156	0.48783	0.53762	0.53910	0.47824
Bagging	0.61925	0.61314	0.61790	0.61249	0.61285
Boosting	0.59182	0.59197	0.59054	0.58101	0.59214
Random Forest	0.57262	0.60631	0.58691	0.60898	0.56609
SVM	0.59995	0.60063	0.59872	0.59778	0.59849

Source: authors' calculations

Continuing our analysis, we turn our attention to DA. Table 48 reveals some notable observations. Firstly, every machine learning model outperforms the benchmark model at every horizon, with the exception of Ridge and Random Forest at $h = 9$, although the extent of outperformance is smaller than in the pre-COVID sample. Nonetheless, this reaffirms our initial assumptions. Secondly, while the outperformance is smaller compared to using hard macroeconomic data, this outcome is anticipated. Forecasts based on indicator data typically exhibit lower DA values than those based on hard macroeconomic data, but this compromise is justified by the advantage of speed. Thirdly, this outperformance remains consistent across every forecasting horizon.

Table 48: Directional accuracy of forecasts based on indicators – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	43.84%	46.01%	44.12%	43.76%	44.62%
Lasso	54.79%	48.83%	50.49%	49.06%	48.66%
Ridge	43.84%	45.54%	44.85%	45.47%	43.41%
Elastic Net	45.21%	46.48%	44.85%	43.42%	43.68%
Bagging	53.42%	48.83%	50.00%	52.14%	51.08%
Boosting	56.16%	51.17%	50.98%	48.03%	50.54%
Random Forest	58.90%	48.83%	50.49%	5.81%	49.46%
SVM	54.79%	54.46%	54.17%	53.85%	53.36%

Source: authors' calculations

4.2.4 Composite forecasts of industrial production – post-COVID

In this thesis, we explore the use of ML models not only as individual forecasters but also as tools for combining forecasts, specifically focusing on data from Slovakia. This methodology is relatively novel in existing research, with only Araujo and Gaglianone (2023) briefly employing it. We generate composite forecasts by employing outputs from individual ML models as inputs (regressors) in subsequent applications of these models.

Table 49: Composite forecasts including all models – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	43.84%	46.01%	44.12%	43.76%	44.62%
Lasso	89,31%	90,56%	86,04%	94,09%	93,18%
Ridge	76,23%	76,61%	71,68%	82,90%	83,30%
Elastic Net	90,04%	90,65%	86,86%	92,91%	91,25%
Bagging	88,94%	70,38%	35,44%	66,16%	90,19%
Boosting	51,74%	47,45%	13,73%	45,56%	50,98%
SVM	28,00%	27,63%	11,10%	34,86%	30,32%

Source: authors' calculations

We conduct composite forecasts across two different categories. For the first composite forecast, we incorporate all model types: individual and combined models utilizing hard data, models based on soft indicators, and models derived from common factors. Additionally, a final composite includes all combined model forecasts from each category as regressors. In this chapter, to conserve space, we present only the percentage improvement of the composite forecasts compared to a benchmark model.

To start, we discuss the results of the composite forecasts for industrial production using all available models, as shown in Table 49. This table presents intriguing findings. On one hand, the performance of individual models such as Lasso and Elastic Net decreases. On the other hand, the performance of Ridge increases slightly. More notably, Bagging exhibits a significant improvement in forecast accuracy, achieving an outperformance of over 90.00% at $h = 12$.

Table 50: Composite forecasts including combined models – IP – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	43.84%	46.01%	44.12%	43.76%	44.62%
Lasso	76,18%	91,01%	87,08%	93,68%	92,11%
Ridge	72,39%	92,21%	80,62%	92,82%	93,55%
Elastic Net	79,06%	90,99%	87,08%	94,03%	93,20%
Bagging	63,58%	50,10%	13,22%	29,14%	90,97%
Boosting	25,60%	43,40%	-27,27%	64,24%	73,82%
SVM	27,62%	35,10%	11,77%	34,34%	35,49%

Source: authors' calculations

The composite forecasts that rely solely on the combined models are detailed in Table 50. From Table 50, it is evident that only the results from the Ridge model show improvement through the second composite forecast setup compared to the previous one.

4.2.5 Forecasting inflation using hard macroeconomic data – post-COVID

Table 51 presents the relative MSE and MAE values of inflation forecasts of the RLS models.

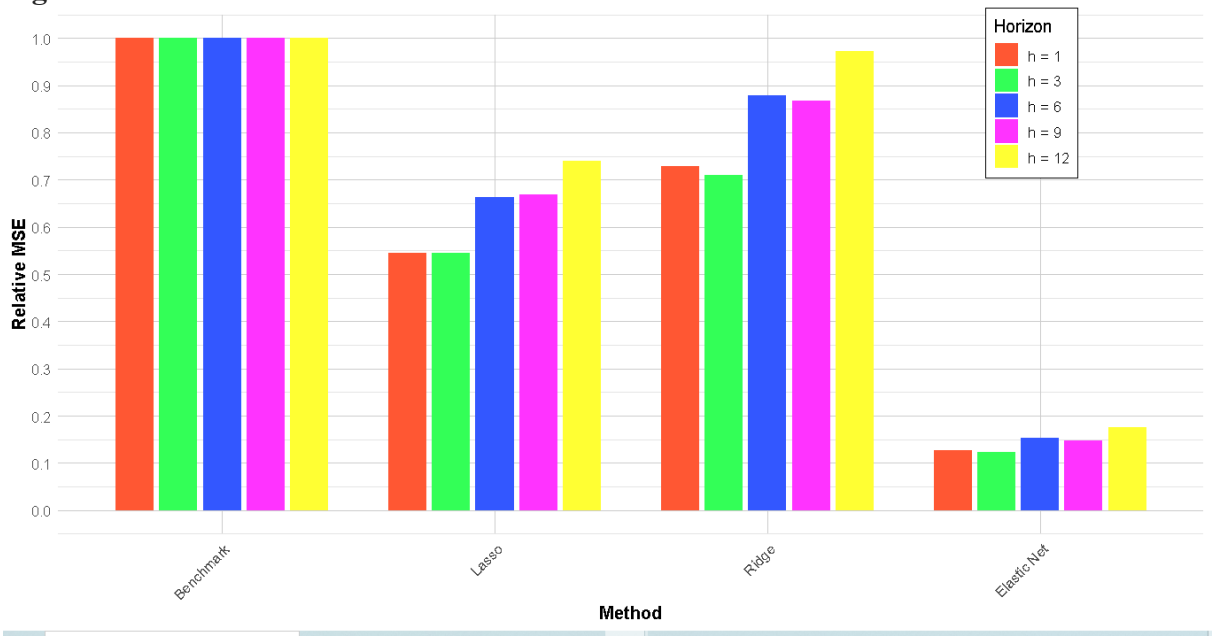
Table 51: Relative MSE and MAE of RLS models - inf

Relative MSE	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Lasso	0.545497	0.545653	0.663038	0.669495	0.740914
Ridge	0.727910	0.710609	0.879189	0.868040	0.972999
Elastic Net	0.126798	0.123258	0.152947	0.148391	0.175345
Relative MAE					
Benchmark	1	1	1	1	1
Lasso	0.766725	0.768022	0.851964	0.853420	0.898265
Ridge	0.912467	0.887733	0.993053	0.975330	1.002049
Elastic Net	0.389653	0.379065	0.423948	0.411235	0.438947

Source: authors' calculations

Firstly, concerning relative MSE, every RLS model surpasses the benchmark. The Lasso and Elastic Net models show slight and substantial improvements, respectively. Contrary to the results in Table 18, the relative MSE for Lasso ranges from 0.54 to 0.74, while for Elastic Net it is between 0.12 and 0.18—indicating a significant enhancement when compared to the previous dataset. However, the performance of the Ridge model deteriorates. Although it still outperforms the benchmark, it does so marginally, and the relative MAE values indicate that the results lack robustness, leading us to deem the Ridge model's superiority as unreliable. Ridge has a similarly high difference between MAE and MSE values when forecasting industrial production, but the problem is greatly magnified here. The Elastic Net model, which effectively incorporates aspects of both Lasso and Ridge, emerges as the most reliable, confirming that in this scenario, a strategy combining variable selection (zero coefficients) and coefficient shrinkage is more effective than focusing solely on shrinkage as in the Ridge model. Considering the relative MAE values, the superior performance of the Lasso and Elastic Net models is consistent and not skewed by a few large forecast errors. While the magnitude of Lasso's outperformance diminishes over *h*, the Elastic Net's remains relatively steady across all forecast horizons. We present this on Figure 34.

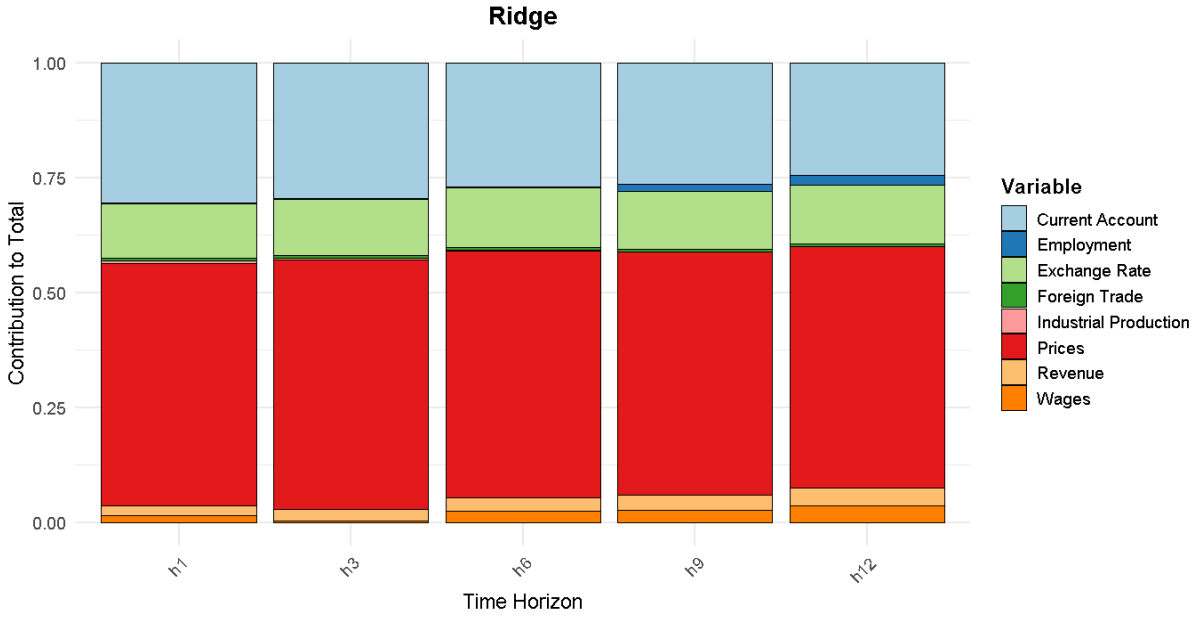
Figure 34: Relative MSE of RLS models over time – Inf– Post-COVID



Source: authors' calculations

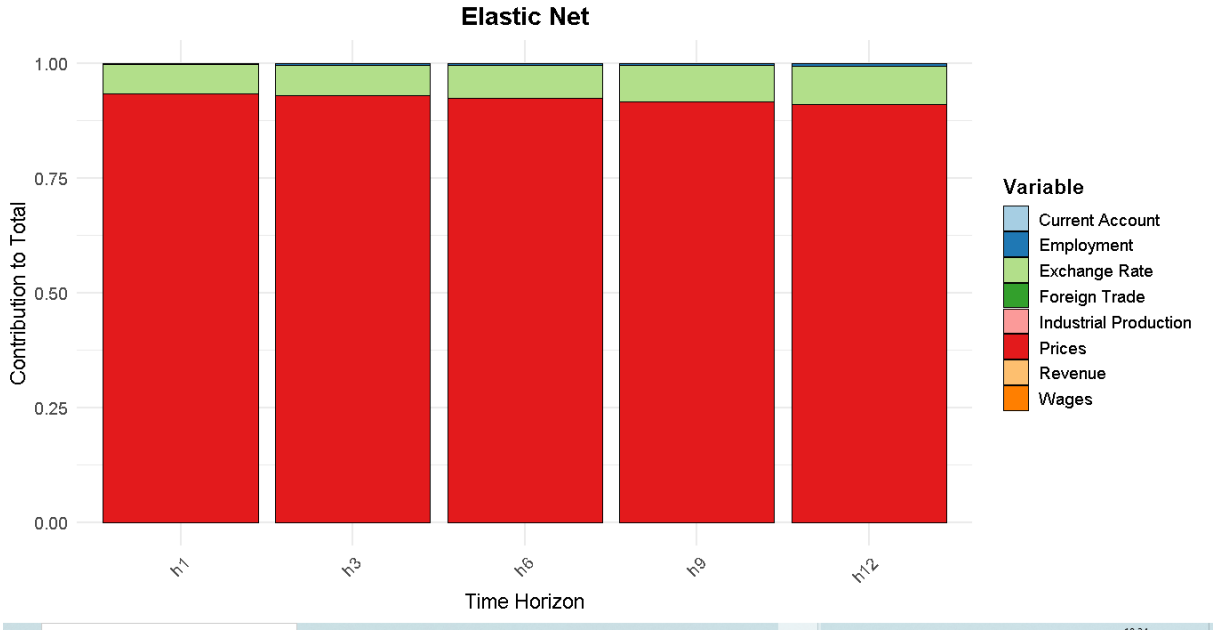
The results indicate that variable selection plays a pivotal role, proving more crucial than coefficient shrinkage. To illustrate the differences between shrinkage and selection more clearly, we present the variables selected by each model in Figures 35 to 37, which are based on the data from Tables A1.1 to A1.8. Variable importance is calculated according to Kock et al. (2020), expanded on in Subchapter 4.2.1. These figures highlight the specific variable categories that each model prioritizes, helping to understand their impact on the forecast accuracy and the effectiveness of each approach.

Figure 35: Variable importance of the Ridge model – Inf – post-COVID



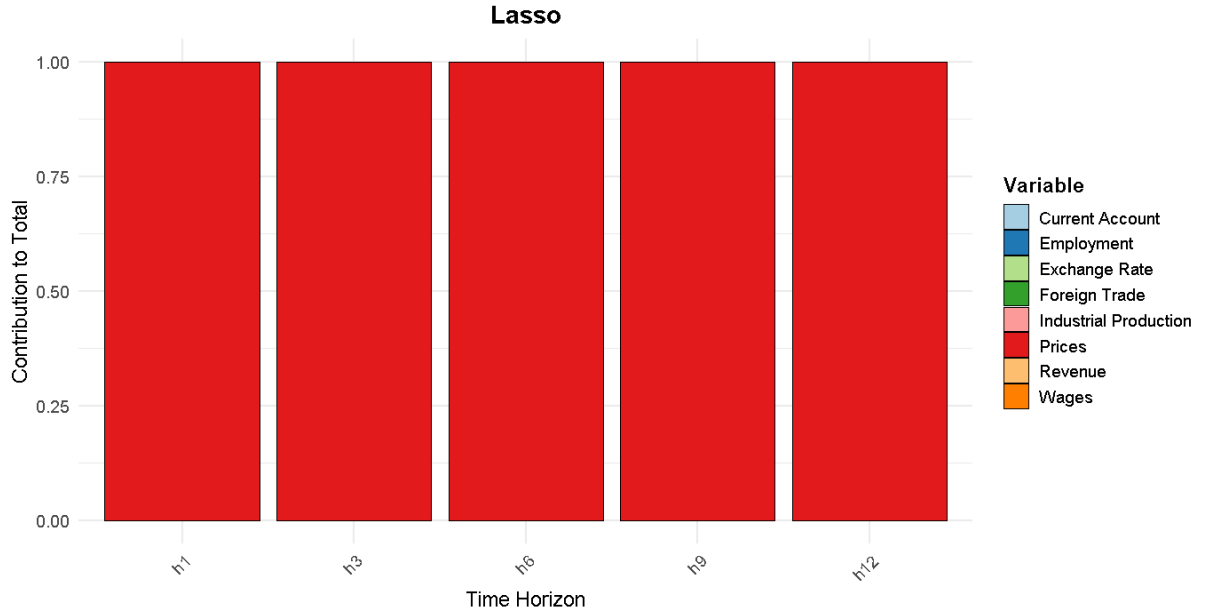
Source: authors' calculations

Figure 36: Variable importance of the Elastic Net model – Inf – post-COVID



Source: authors' calculations

Figure 37: Variable importance of the Lasso model – Inf – post-COVID



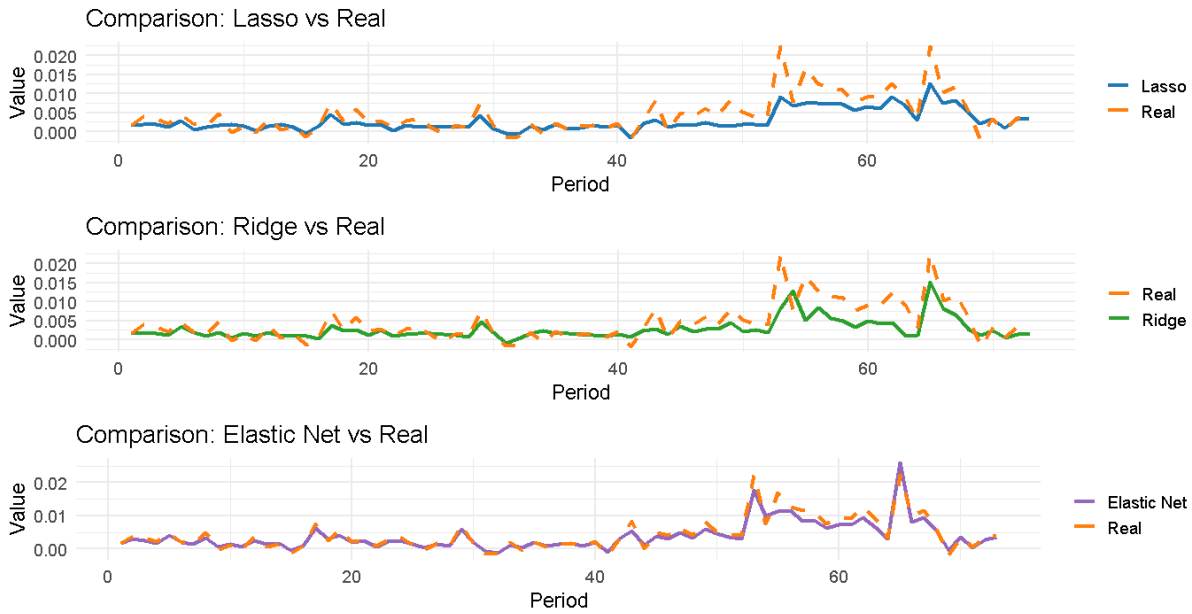
Source: authors' calculations

In this analysis, both Ridge and Elastic Net models emphasize the significance of inflation and open economy variables in forecasting inflation. Ridge, unable to perform variable selection and only capable of shrinking coefficients, places greater emphasis on variables like the current account and the exchange rate. Conversely, Elastic Net tends to prioritize various price indexes while assigning a limited effect to exchange rates on inflation. Lasso takes the most selective approach, effectively reducing every coefficient to zero except those within the price category.

Given that Elastic Net yields the best performance, we infer that its strategy—incorporating a broad array of variables like Ridge, while focusing tightly on the price category like Lasso—represents the most effective general approach for inflation forecasting. Thus, we conclude that both domestic price factors and open economy variables are critical in accurately forecasting inflation. Elastic Net's balanced methodology particularly demonstrates the importance of integrating comprehensive variable inclusion with focused selection.

To shed further light on the precision of these methods, we compare the out-of-sample actual and forecasted values on Figure 38 for $h = 1$ and Figure 39 for $h = 12$. Figures 38 and 39 provide crucial insights into the performance of our models over different forecast horizons. Initially, at $h = 1$, the models closely match the actual inflation values until the onset of COVID. Although the models continue to capture the general pattern of inflation post-COVID, they underpredict the volatility and the magnitudes of change. The notable exception is the Elastic Net model, which not only accurately predicts the direction of changes but also the magnitudes, fitting almost perfectly with the actual data. This high level of accuracy leads us to conclude that in our case the Elastic Net model is the most precise predictor of inflation, even in highly volatile environments.

Figure 38: Actual and forecasted values of RLS models on $h = 1$ – Inf – post-COVID

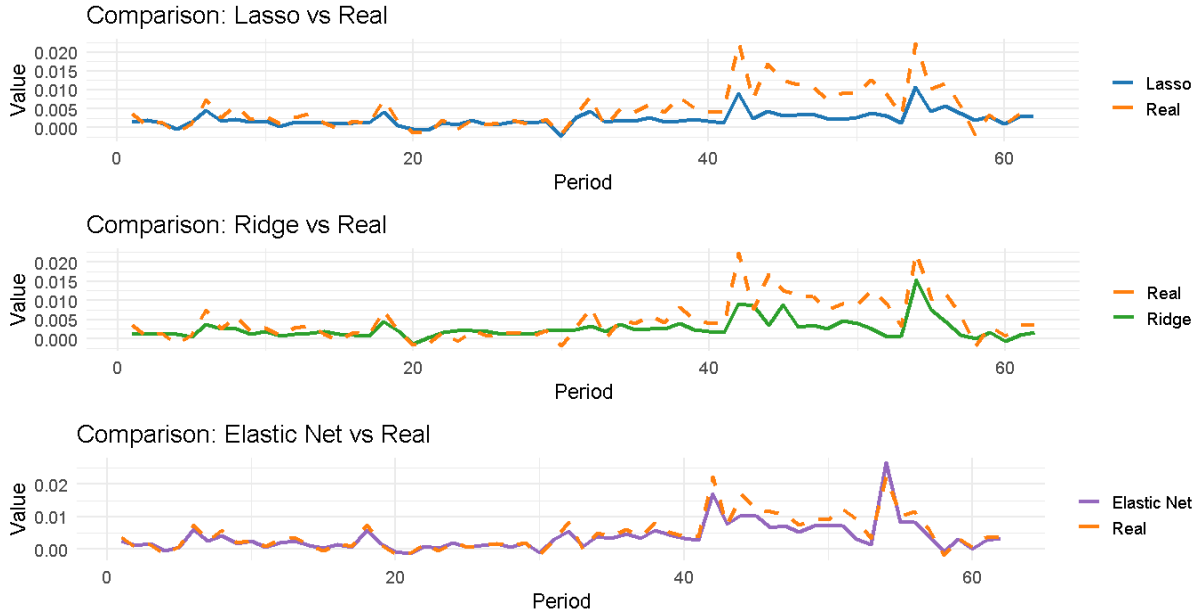


Source: authors' calculations

While the model's performance in capturing the magnitudes of change slightly deteriorates at $h = 12$, the ability to forecast inflation with such precision for one and three months ahead remains an exceptional achievement. Given that the values in Table 51 for $h = 1$

and $h = 3$ are nearly identical, the results displayed in Figures 38 and 39 would effectively be the same for these two horizons, illustrating the robustness and reliability of the Elastic Net model across short-term forecasts.

Figure 39: Actual and forecasted values of RLS models on $h = 12$ – Inf – post-COVID



Source: authors' calculations

To add, Table 52 presents the DM statistic *p*-values. It reveals some variations in the results when comparing the current dataset, which includes COVID, to the previous sample that excluded it. Firstly, the Elastic Net model demonstrates robust performance, being highly statistically significant at every forecast horizon and maintaining significance at the 5% level throughout. This indicates strong and consistent predictive power across various forecasting periods. On the other hand, the Lasso model starts strong, showing statistical significance for the initial four horizons, but its significance wanes by $h = 12$. This suggests that while the Lasso model can effectively predict inflation in the short to medium term, its reliability decreases as the forecast horizon extends to a year.

Table 52: DM test *p*-values of RLS methods – Inf – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Lasso	0,01657**	0,01664**	0,04022**	0,04660**	0,122705
Ridge	0,13071	0,11557	0,303495	0,292095	0,456085
Elastic Net	0,0029**	0,00281**	0,00675**	0,00822**	0,02141**

Source: authors' calculations

Table 53 provides a detailed comparison of model performances, highlighting the differences brought about by including COVID data in the dataset. For the Lasso model, there is a notable improvement in its outperformance, ranging from 25.91% to 45.45% over the

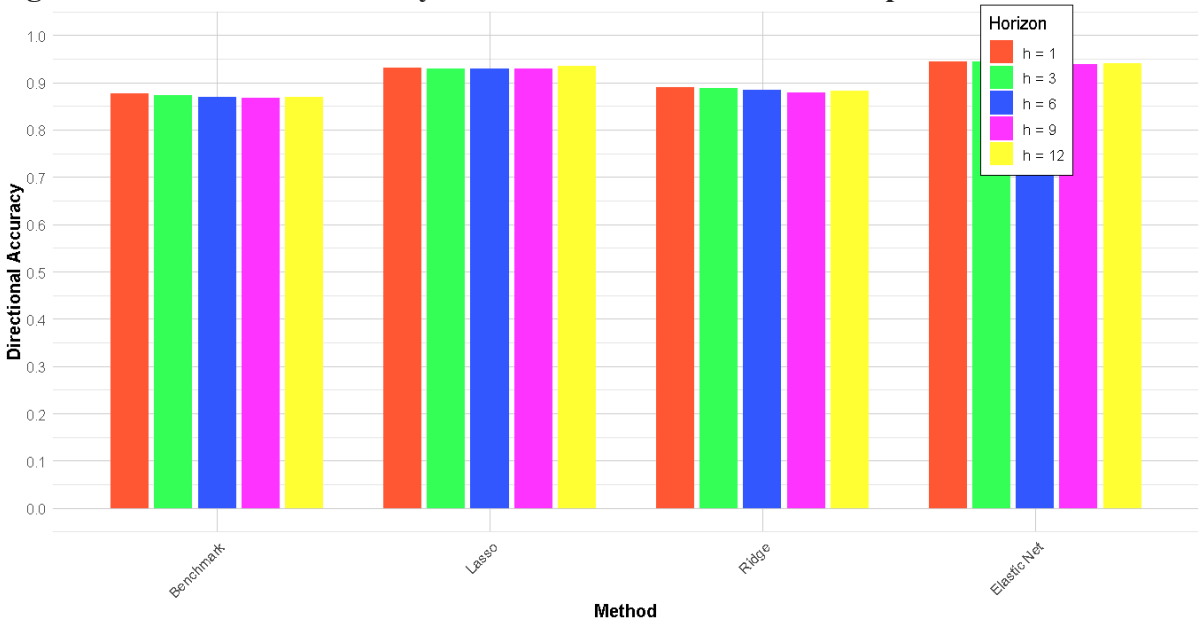
benchmark, which represents an almost two-fold increase from previous data, particularly at shorter horizons. However, this outperformance diminishes as *h* extends. The Ridge model consistently outperforms the benchmark across various horizons, although never reaching statistical significance. Its performance, similar to that of the Lasso model, tends to deteriorate as *h* increases, suggesting a weakening in predictive power over longer durations. The Elastic Net model stands out with the highest degree of outperformance, ranging from 82.47% to 87.32%. This is a significant enhancement, nearly doubling its relative performance compared to the previous sample without COVID data. This consistent outperformance across horizons underscores the robustness of the Elastic Net model, especially in handling the complexities introduced by the volatile economic environment during the COVID era.

Table 53: % improvement of RLS forecasts over the benchmark – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Lasso	45.45%	45.43%	33.70%	33.05%	25.91%
Ridge	27.21%	28.94%	12.08%	13.20%	2.70%
Elastic Net	87.32%	87.67%	84.71%	85.16%	82.47%

Source: authors' calculations

Figure 40: Directional Accuracy of RLS models over time – Inf – post-COVID



Source: authors' calculations

Table 54 focuses on the DA of RLS models. The results show that all RLS models consistently predict the direction of change more accurately than the benchmark. Specifically, the DA of the Lasso model ranges from 92.89% to 93.41%, indicating a high level of consistency in predicting the correct direction. The Elastic Net model performs even better, with DA ranging from 93.85% to 94.52%. In contrast, the benchmark model's DA ranges from

86.84% to 87.67% across different horizons, significantly lower than both the Lasso and Elastic Net models. This superior performance in directional accuracy is not only an improvement over the benchmark but also represents an enhancement compared to the previous data, as shown in Table 21. Figure 40 shows that the results are consistent irrespective of *h.* In contrast to the pre-COVID sample, this time the benchmark model is consistent as well.

Table 54: Directional accuracy of RLS models – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	87,67%	87,32%	87,01%	86,84%	86,96%
Lasso	93,15%	92,96%	92,89%	92,99%	93,41%
Ridge	89,04%	88,73%	88,48%	87,86%	88,17%
Elastic Net	94,52%	94,37%	94,12%	93,85%	94,09%

Source: authors' calculations

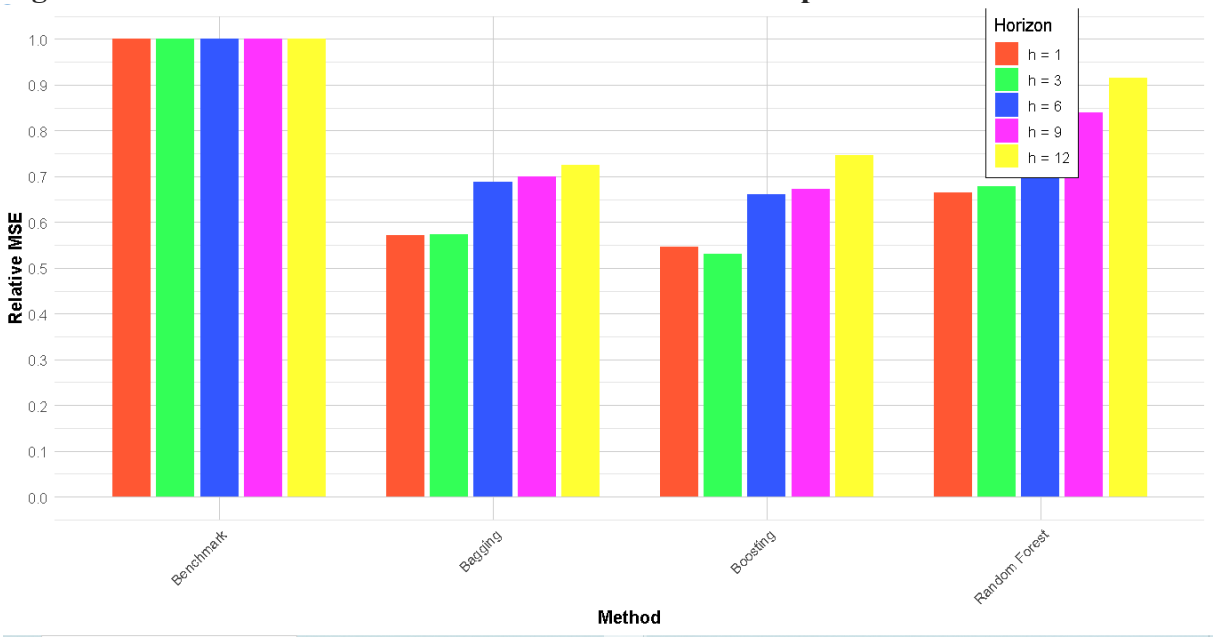
We proceed by presenting the results of the EML models in Table 55. In this analysis, every EML model surpasses the benchmark when assessed using relative MSE. Specifically, Bagging achieves a relative MSE between 0.57 and 0.73, Boosting ranges from 0.54 to 0.75, and Random Forest varies between 0.66 and 0.92. The most effective model is Boosting from $h = 1$ to $h = 9$, and Bagging takes the lead at $h = 12$, although the performance differences between these two are minimal. Their results are notably consistent with their performances in the pre-COVID sample. In contrast, the performance of the Random Forest model shows a decline compared to the previous sample but still manages to outperform the benchmark. Secondly, the relative MAE results corroborate our earlier observations, as they too are smaller than the benchmark, not to mention that the relative MSE and MAE values are relatively similar, showcasing a greater robustness that those of the RLS models. Bagging and Boosting maintain stable performances over time, similar to those observed in previous samples, while the performance of Random Forest declines slightly as *h* increases.

Table 55: Relative MSE and MAE of EML models – Inf – post-COVID

Relative MSE	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Bagging	0.57115	0.57272	0.68839	0.70001	0.72526
Boosting	0.54712	0.53020	0.66037	0.67262	0.74603
Random Forest	0.66392	0.67879	0.82230	0.83953	0.91612
Relative MAE					
Benchmark	1	1	1	1	1
Bagging	0.655411	0.648945	0.698539	0.704609	0.697229
Boosting	0.679309	0.667017	0.766165	0.753878	0.786163
Random Forest	0.76334	0.76933	0.838285	0.837562	0.87526

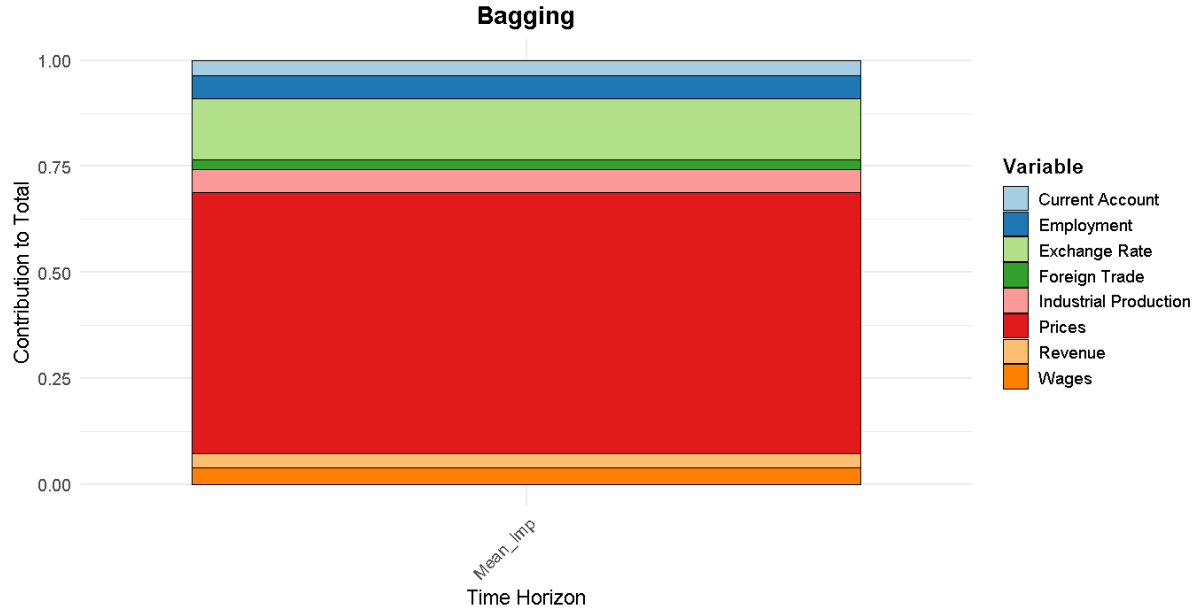
Source: authors' calculations

Figure 41: Relative MSE of EML models over time – Inf – post-COVID



Source: authors' calculations

Figure 42: Variable importance in Bagging – Inf – post-COVID



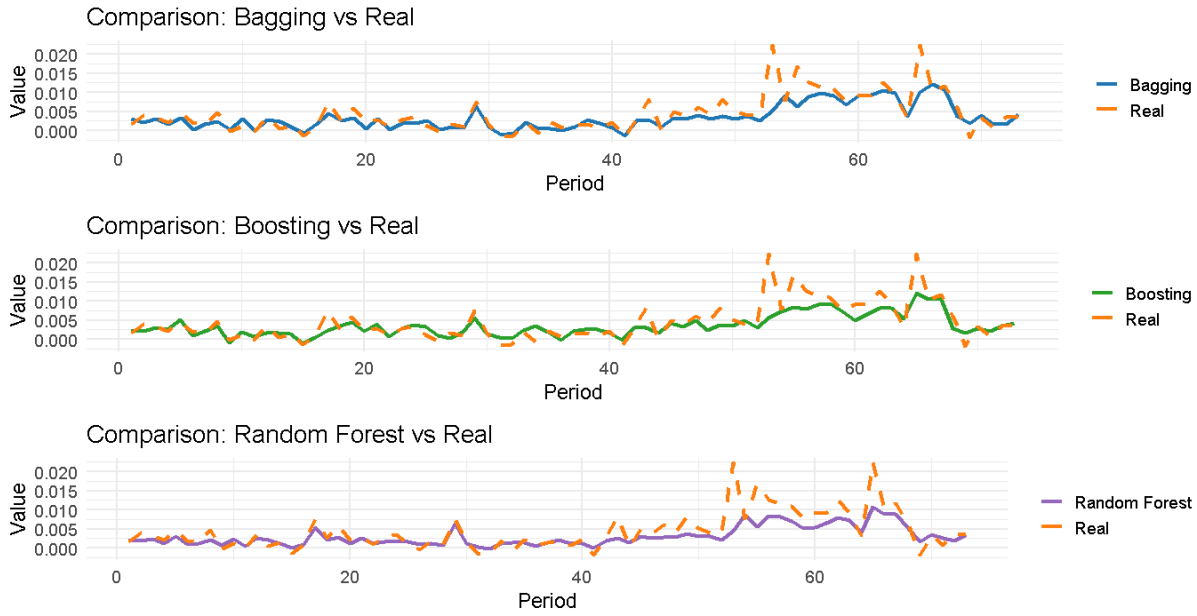
Source: authors' calculations

Like RLS models, Bagging places significant emphasis on the variables listed in Table A1.2. Unlike the Lasso model, which primarily focuses on a specific category of variables, Bagging also considers variables from other categories. It assigns substantial weight to open economy variables including exchange rates, indicating their perceived importance in the forecasting model. Additionally, industrial production and employment variables are notably represented, suggesting their relevance in the model's predictive accuracy. However, variables such as wages, revenue, and foreign trade are given lesser weight in the Bagging model's

forecast preparations. This distribution of weights suggests that while these factors are considered, they are deemed less critical to the forecast outcome compared to the primary variables like prices. This strategic allocation of variable importance helps tailor the Bagging model's predictions to the most influential economic indicators.

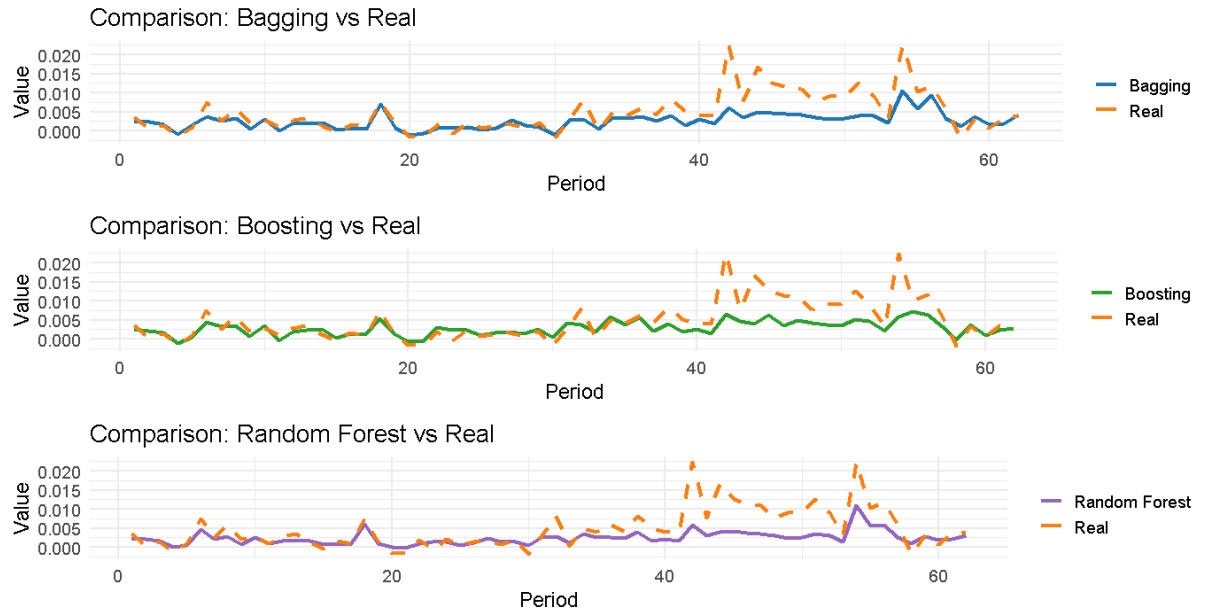
To further evaluate the accuracy of these forecasting methods, we examine the out-of-sample actual versus forecasted values presented in Figure 43 for $h = 1$ and Figure 44 for $h = 12$. At $h = 1$, while the models successfully predict the direction of change, they fail to accurately capture the magnitudes of these changes. This indicates a limitation in their short-term predictive precision regarding the scale of economic shifts. For longer-term forecasts at $h = 12$, the models tend to predict a smoother inflation trajectory than what is observed in reality. This smoother prediction suggests that the models may be underestimating the volatility or the extremes of inflation changes over longer periods. In conclusion, when compared to the performance of RLS models, particularly the Elastic Net model, these methods exhibit weaker performance. The Elastic Net model, with its ability to accurately capture both the direction and magnitude of changes, outperforms these other methods, especially in terms of capturing more complex economic dynamics over varying time horizons.

Figure 43: Actual and forecasted values of EML models on $h = 1$ – Inf – post-COVID



Source: authors' calculations

Figure 44: Actual and forecasted values of EML models on h = 12 – Inf – post-COVID



Source: authors' calculations

To continue with, Table 55 shows that Both Bagging and Boosting demonstrate statistically significant outperformance over the benchmark at the 5% level across all forecast horizons, indicating strong and consistent effectiveness in their predictions. On the other hand, the performance of the Random Forest model shows a slight deterioration but remains statistically significant at shorter horizons h = 1 and h = 3.

Table 56: DM test *p*-values of EML methods – Inf – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Bagging	0,00767**	0,00812**	0,00668**	0,00563**	0,03183**
Boosting	0,00537**	0,00584**	0,00352**	0,00564**	0,01953**
Random Forest	0,03415**	0,04004**	0,10103	0,13071	0,29610

Source: authors' calculations

With the statistically significant outperformance established, we proceed to quantify the magnitude of outperformance in percentage terms, stating them in Table 57. The Bagging model shows an outperformance ranging from 27.34% to 42.88% and the Boosting model ranges from 25.40% to 45.29%. These models not only perform similarly to the Lasso model in terms of percentage improvement but also maintain statistical significance at the 5% level across all horizons. Notably, only the Elastic Net model surpasses these two in terms of percentage improvement over the benchmark. The performance of the Random Forest model is comparatively weaker, with outperformance declining from 33% to 8% as the forecast horizon increases. This trend indicates a clear drop in effectiveness over longer horizons.

Table 57: % improvement of EML forecasts over the benchmark – Inf – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Bagging	42.88%	42.73%	31.16%	30.00%	27.47%
Boosting	45.29%	46.98%	33.96%	32.74%	25.40%
Random Forest	33.61%	32.12%	17.77%	16.05%	8.39%

Source: authors' calculations

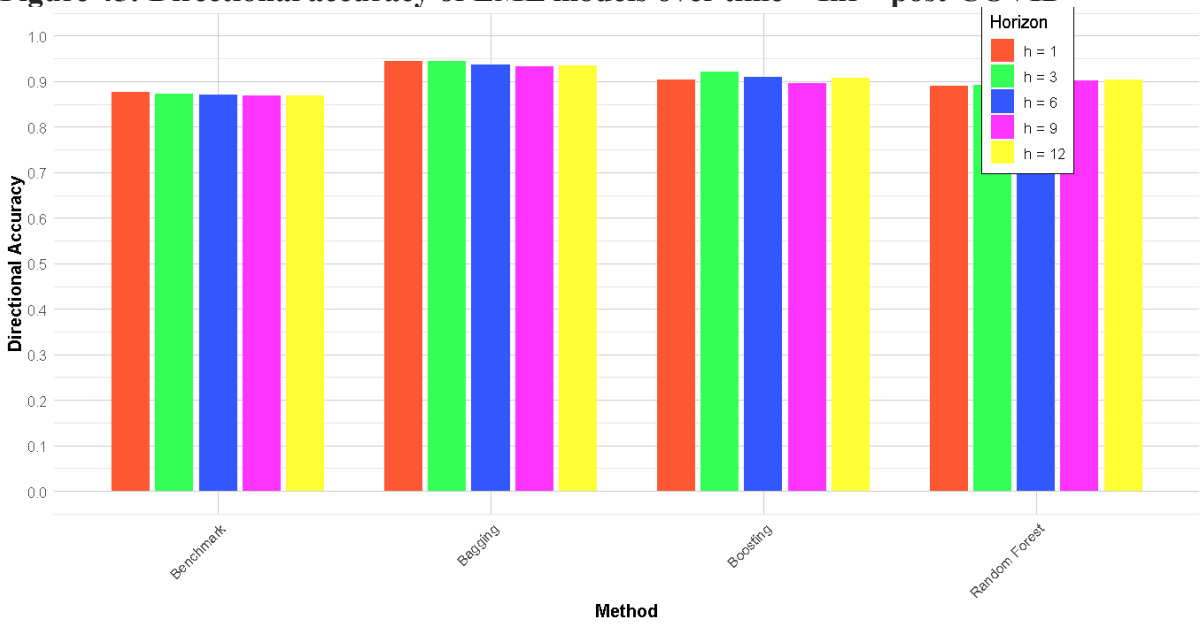
Table 58: Directional accuracy of EML models – Inf – post-COVID

	$h = 1$	$h = 3$	$h = 6$	$h = 9$	$h = 12$
Benchmark	87.67%	87.32%	87.01%	86.84%	86.96%
Bagging	94.52%	94.37%	93.63%	93.33%	93.55%
Boosting	90.41%	92.02%	90.93%	89.57%	90.73%
Random Forest	89.04%	89.20%	89.22%	90.09%	90.32%

Source: authors' calculations

Moving on to DA, every EML model more accurately predicts the direction of change in the target variable compared to the benchmark model. The details are presented in Table 58. When compared to the pre-COVID sample, the DA for each model has significantly increased. These models perform on par with RLS models, including the Elastic Net model. The DA for the Bagging model ranges between 93.33% and 94.52%. For the Boosting model, it ranges from 87.57% to 92.02%, while for the Random Forest model, it is between 89.04% and 90.32%. This is in contrast to the benchmark model, which shows a DA between 86.96% and 87.67%, depending on the forecast horizon. In conclusion, the EML models enhance DA, with a particular focus on the Bagging model, which achieves the highest DA, surpassing even the Boosting model. Figure 45 shows that this improvement is consistent over time.

Figure 45: Directional accuracy of EML models over time – Inf – post-COVID



Source: authors' calculations

In Table 59, we present the results for the SVM model. We observe a significant deterioration in its performance in the new sample post-COVID compared to the pre-COVID sample. The model reports higher relative MSE values than the benchmark across all forecast horizons. The relative MAE values support this finding, and the DM test results confirm the benchmark model's superiority. DA values for the SVM model are the lowest among all models and worsen with increasing *h*.

Table 59: Forecasts with SMV, FFNN and LSTM – IP – post-COVID

Relative MSE	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Support Vector Machine	1.81410	1.81913	2.19781	2.14399	2.22832
Relative MAE					
Benchmark	1	1	1	1	1
Support Vector Machine	1.39671	1.38662	1.54470	1.49922	1.53964
DA					
Benchmark	43.84%	46.01%	44.12%	43.76%	44.62%
Support Vector Machine	87.67%	29.11%	14.46%	9.57%	7.39%

Source: authors' calculations

As with industrial production, we are also interested in exploring how the combination of regularization and nonlinear methods performs. Given that the Elastic Net is the clear frontrunner among the three regularization methods, we use it for variable selection and apply nonlinear methods to these selected variables at each stage of the process. We aim to determine if the performance of nonlinear methods improves when applied to a more parsimonious sample. To the best of the authors' knowledge, no published study has combined these two approaches before. This absence may be due to other studies, which work with much larger datasets and longer time series from the world's largest economies, not needing to address the issue of limited degrees of freedom. Although nonlinear methods can effectively handle models with even negative degrees of freedom, their performance might be enhanced by a more parsimonious approach. Several notable results are highlighted in Table 59 below.

Table 59: Combined RLS and EML methods – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
EN + Bagging	39.83%	45.08%	24.19%	26.01%	17.63%
EN + Boosting	49.82%	47.72%	38.38%	39.58%	27.47%
EN + Random Forest	46.31%	48.94%	36.62%	34.77%	30.23%

Source: authors' calculations

We leave out the Lasso and Ridge combinations from Table 59 intentionally, as they

produce either equal or inferior results to the Elastic Net combinations. To start, combining Elastic Net with Bagging does not lead to any improvement. In fact, the performance of Bagging slightly worsens when compared to its performance using the entire sample. Thus, this change is not statistically significant. However, when Boosting is combined with EN, there is a modest improvement—about 7% in some instances. The most notable improvement occurs with the Random Forest model, which, when combined with the Elastic Net, significantly exceeds its performance without variable selection.

To summarize, in this subchapter we forecast inflation using hard macroeconomic data. The machine learning models exhibit significant outperformance in most cases. The models, sorted from best to worst in terms of % improvement over the benchmark, are presented in Table 60.

To begin, the Elastic Net model stands out as the top performer. Lasso ranks in the middle, while Ridge is fourth to last, only outperforming the SVM and neural networks. The performance of EML models is more ambiguous. None of them surpass their combination with the Elastic Net model, but these combined models do outperform both the standalone nonlinear and linear methods, except for the Elastic Net itself. These hybrid models hold the 2nd and 3rd positions, respectively. This highlights the effectiveness of our hybrid approach in significantly enhancing the forecasting performance of nonlinear models by selecting the most suitable variables or parameter values for training.

Table 60: Forecasting models from best to worst – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
1.	Elastic Net	Elastic Net	Elastic Net	Elastic Net	Elastic Net
2.	EN + Boosting	EN + RF	EN + Boosting	EN + Boosting	EN + RF
3.	EN + RF	EN + Boosting	EN + RF	EN + RF	EN + Boosting
4.	Lasso	Boosting	Boosting	Lasso	Bagging
5.	Boosting	Lasso	Lasso	Boosting	Boosting
6.	Bagging	EN + Bagging	Bagging	Bagging	Lasso
7.	RF	Bagging	EN + Bagging	EN + Bagging	EN + Bagging
8.	EN + Bagging	RF	RF	RF	RF
9.	Ridge	Ridge	Ridge	Ridge	Ridge
16.	SVM	SVM	SVM	SVM	SVM
17.	FFNN	FFNN	FFNN	LSTM	LSTM
18.	LSTM	LSTM	LSTM	FFNN	FFNN

Source: authors' calculations

4.2.6 Forecasting inflation using common factors – post-COVID

Table 61 indicates that forecasts based on common factors perform worse in the post-COVID sample compared to the pre-COVID period. Most models no longer outperform the benchmark. The performance of the Ridge and SVM models improve, and the Elastic Net model also remains a notable exception; it still significantly outperforms the benchmark by a large margin. In contrast, other RLS and EML methods show poor performance relative to the benchmark.

Continuing, we analyze the percentage difference between the relative MSE values of forecasts based on hard macroeconomic data and those based on common factors, as shown in Table 62. In most cases, models relying on hard data are more accurate. The performance of the Ridge model is almost identical to its counterpart, while the Support Vector Machine (SVM) model outperforms the version based on hard macroeconomic data, though it is generally the least effective model overall.

Table 61: Relative MSE of forecasts based on common factors – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Lasso	1.623955	1.622663	2.018820	2.003048	1.987983
Ridge	0.726089	0.712190	0.869869	0.887938	0.894660
Elastic Net	0.685868	0.678498	0.833812	0.828000	0.884142
Bagging	1.091421	1.104612	1.357364	1.352660	1.430018
Boosting	1.069080	1.128786	1.400039	1.345368	1.386928
Random Forest	1.623955	1.622663	2.018820	2.003048	1.987983
SVM	0.726089	0.712190	0.869869	0.887938	0.894660

Source: authors' calculations

Table 62: % difference between forecasts based on factors and hard data – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Lasso	66.41%	66.37%	67.16%	66.58%	62.73%
Ridge	-0.25%	0.22%	-1.07%	2.24%	-8.76%
Elastic Net	81.51%	81.83%	81.66%	82.08%	80.17%
Bagging	47.67%	48.15%	49.28%	48.25%	49.28%
Boosting	48.82%	53.03%	52.83%	50.00%	46.21%
Random Forest	41.15%	37.67%	42.24%	39.04%	37.90%
SVM	-3.68%	-4.18%	-2.49%	-2.91%	-4.54%

Source: authors' calculations

To conclude, we examine the DA of forecasts based on common factors as presented in Table 63. This table corroborates our previous findings that hard macroeconomic data should

be preferred. The directional accuracy values for models based on common factors are, in some cases, effectively identical to those of the benchmark, with none of the models managing to confidently outperform the benchmark. This evidence further supports our conclusion that in situations characterized by uncertainty and volatility, forecasts based on hard macroeconomic data are more reliable and should be utilized.

Table 63: Directional accuracy of forecasts based on common factors – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	87.67%	87.32%	87.01%	86.84%	86.96%
Lasso	87.67%	87.32%	87.01%	86.84%	86.96%
Ridge	86.30%	87.32%	87.01%	86.84%	86.96%
Elastic Net	87.67%	87.32%	87.01%	86.84%	86.96%
Bagging	87.67%	84.04%	83.82%	83.42%	83.47%
Boosting	86.30%	86.38%	86.27%	85.47%	86.96%
Random Forest	89.04%	87.32%	87.01%	86.84%	86.96%
SVM	87.67%	87.32%	87.01%	86.84%	86.96%

Source: authors' calculations

4.2.7 Forecasting inflation using soft indicators – post-COVID

The outcomes of machine learning and benchmark model forecasts based on soft indicators are displayed in Table 64. As anticipated, Table 64 indicates that soft indicators are not the most suitable source of information when the focus is on the magnitude of changes in macroeconomic variables, particularly in the case of inflation.

Table 64: Relative MSE of forecasts based on indicators – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	1	1	1	1	1
Lasso	1.76030	1.75710	2.15862	2.08513	2.11772
Ridge	1.76850	1.76980	2.17129	2.09983	2.16164
Elastic Net	1.70017	1.69383	2.09359	2.00069	1.99589
Bagging	1.71610	1.71215	2.10642	2.01600	2.00236
Boosting	1.72136	1.71904	2.14139	1.99844	1.98146
Random Forest	1.73032	1.66970	2.08791	1.96987	1.99995
SVM	1.81310	1.79238	2.21362	2.14523	2.18999

Source: authors' calculations

Continuing our analysis, we focus DA, which is our main measure of interest when evaluating indicator data. The DA values are presented in Table 65. Unlike the pre-COVID sample, the DA values for models based on indicators, similar to those using common factors, align closely with those of the benchmark model in the current sample.

This similarity suggests that while indicator-based models do not necessarily improve accuracy in terms of the magnitude of changes, they can still match the benchmark model's ability to predict the direction of changes in macroeconomic variables, even slightly outperforming it, under the conditions examined.

Table 65: Directional accuracy of forecasts based on indicators – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	87.67%	87.32%	87.01%	86.84%	86.96%
Lasso	87.67%	85.92%	86.03%	85.47%	85.89%
Ridge	87.67%	85.45%	86.76%	87.18%	88.44%
Elastic Net	87.67%	87.32%	87.50%	88.72%	88.58%
Bagging	83.56%	87.32%	87.50%	87.18%	87.10%
Boosting	86.30%	84.98%	83.82%	85.30%	85.89%
Random Forest	87.67%	87.79%	86.76%	87.18%	86.96%
SVM	87.67%	87.32%	87.01%	86.84%	86.96%

Source: authors' calculations

4.2.8 Composite forecasts of inflation – post-COVID

In this subchapter, we explore the composite forecast methods that have been previously introduced. We begin by presenting the results of composite forecasts for inflation, utilizing all available models, as detailed in Table 66. Table 66 present the outperformance, in percentage terms, as measured by the relative MSE values. Initially, there is a significant improvement in the performance of the Lasso model in terms of error reduction and consistency. The composite Lasso model is approximately twice as accurate as the individual model in the short term, and this advantage becomes even more pronounced as the forecast horizon h increases. Similarly, the composite Ridge model shows substantial improvement over its individual counterpart, although its performance drops dramatically at $h = 12$. The performance of the Bagging model also sees considerable enhancement, with the exception of $h = 12$. This pattern indicates that while composite methods generally enhance forecasting accuracy, their efficacy can vary significantly across different time horizons.

Table 66: Composite forecasts including all models – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Lasso	83.20%	82.49%	77.70%	62.16%	70.66%
Ridge	74.11%	80.44%	59.57%	73.38%	24.39%
Elastic Net	78.98%	82.82%	78.66%	65.19%	70.76%
Bagging	62.17%	62.71%	53.89%	62.67%	25.00%
Boosting	34.63%	51.36%	7.56%	24.31%	36.40%
SVM	-51.97%	-52.07%	-83.48%	-149.77%	-107.42%

Source: authors' calculations

Continuing our discussion, we present the results of the composite forecasts of inflation based on individual models using hard macroeconomic data, as detailed in Table 67. Comparing these results to those previously discussed, we find that the Lasso composite model produces forecasts that are almost identical to those in the prior case, suggesting consistent performance across different data scenarios. Interestingly, the Ridge model shows improvement at the longer forecast horizon of $h = 12$, contrasting with its earlier performance drop at the same horizon. This suggests that the Ridge model may be more effective when utilizing hard macroeconomic data for longer-term forecasts. The performance of the Bagging model remains similar to its previous version, indicating a stable forecasting ability regardless of the specific data set used. These findings underscore the nuanced effects that different types of macroeconomic data can have on the performance of composite forecasting models.

Table 67: Composite forecasts of individual models based on hard macroeconomic data – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	83.13%	82.35%	77.43%	62.42%	70.66%
Lasso	80.79%	85.36%	78.34%	58.49%	73.36%
Ridge	83.32%	85.39%	80.18%	62.69%	72.53%
Elastic Net	63.99%	61.92%	51.46%	63.75%	18.79%
Bagging	34.86%	52.15%	7.61%	28.72%	44.45%
Boosting	-46.59%	-44.32%	-72.86%	-138.23%	-98.03%
SVM	83.13%	82.35%	77.43%	62.42%	70.66%

Source: authors' calculations

To conclude, the results from the composite forecasts based on combined models generally mirror the performance of the individual models themselves; therefore, we do not include them here. Additionally, the performance of composites based on indicators is weak, reflecting the limited effectiveness of individual models that use these indicators.

Table 68: Composite forecasts of individual models based on factors – Inf – post-COVID

	***h* = 1**	***h* = 3**	***h* = 6**	***h* = 9**	***h* = 12**
Benchmark	49.68%	50.89%	35.80%	29.24%	-27.78%
Lasso	44.67%	50.96%	45.88%	26.91%	6.88%
Ridge	46.10%	58.76%	55.55%	30.09%	-12.06%
Elastic Net	30.50%	46.31%	23.89%	0.27%	-26.68%
Bagging	34.97%	58.28%	24.27%	31.44%	30.73%
Boosting	-60.31%	-59.21%	-83.25%	-170.52%	-119.89%
SVM	49.68%	50.89%	35.80%	29.24%	-27.78%

Source: authors' calculations

However, what is more striking is the performance of composites based on common factors, as shown in Table 68. It is clear from this table that common factor-based composites

outperform the benchmark model in most cases, with the exceptions of the SVM model and at the forecast horizon of $h = 12$. This outcome suggests that although individual models based on common factors may perform poorly, leveraging the aggregated information from each of these models through machine learning techniques can lead to more accurate forecasts.

5 Discussion

This chapter summarizes the main findings of our study. Our principal aim is to rigorously evaluate the forecasting performance of various selected machine learning models and compare these results against an econometric benchmark model. We forecast industrial production and inflation for both pre-COVID and post-COVID periods. Our contributions to the literature are both methodological and empirical.

To begin with, we outline our methodological contributions, which introduce three approaches not widely discussed in current literature, or where evidence remains limited. First, we evaluate the effectiveness of dimensionality reduction techniques, drawing parallels to the work by Maehashi and Shintani (2020). PCA is considered one of the best ways for reducing the dimension of a dataset. Our findings suggest that, in general, regularization techniques are preferable to PCA reduction in Slovakia. The observed underperformance of forecasts based on common factors compared to those based on hard data suggests a potential avenue for exploration in the empirical transition from factor-based dimensionality reduction techniques to regularization methods using hard data in Slovakia. Both strategies are designed to distill essential information while minimizing the number of predictors; however, regularization using hard data shows greater accuracy in our sample.

Secondly, we explore the combination of regularization and nonlinear methods, inspired by Medeiros et al. (2019). However, our approach differs as we use alternative methods and conduct a more thorough performance evaluation. Medeiros et al. (2019) find that combination is not particularly beneficial. In contrast, our analysis reveals that in Slovakia, hybrid models are more accurate than simple nonlinear models, as shown in Tables 43 and 60. We enhance the performance of nonlinear methods by first applying regularization to streamline the information set, followed by the nonlinear estimation and prediction. This efficiency likely stems from the reduction in the number of variables, which allows for achieving comprehensive coverage of critical nonlinearities and variable interactions with fewer trees. Furthermore, as noted by Medeiros et al. (2019), when the Random Forest model is applied to a set of regularized variables, its capacity to discern important variable interactions is notably amplified, but it becomes harder to capture nonlinear relationships. In such case, if the goal is to explore the effects of nonlinearities further, one might consider other nonlinear methods like Boosting, which are specifically designed to handle such complexities effectively.

Thirdly, forecast combination is a critical topic in forecasting literature. Machine learning methods are inherently suitable for combining forecasts, yet there is a notable absence of studies testing this approach, aside from Araujo and Gaglianone (2023). We address this gap by demonstrating that using individual ML models as tools for combining forecasts can significantly enhance the performance of certain models. We find that some composite models, which integrate outputs from multiple individual models, can be twice as accurate as their counterparts that solely rely on hard macroeconomic data. This improvement is particularly evident in forecasting inflation and, to a lesser extent, in forecasting industrial production.

We now proceed to discuss our empirical contributions, beginning with the industrial production and inflation forecasts using the pre-COVID sample.

Firstly, every RLS model consistently outperforms the benchmark, with statistical significance at the 5% level. Only in two instances, relating to industrial production, is the outperformance significant at the 10% level. In the case of inflation, every outperformance is significant at the 5% level. Moreover, every RLS forecast model achieves a substantially higher DA than the benchmark for industrial production. Although the DA for machine learning models is somewhat lower for inflation, they still outperform the benchmark. These results identify the Elastic Net model as the most effective in this category. Furthermore, for industrial production, variable selection by Lasso plays a more crucial role than the coefficient shrinkage by Ridge. This situation is reversed for inflation forecasts. To summarize, the RLS models consistently outperform the benchmark across every horizon. This robust performance is expected due to the presence of numerous correlated variables in the dataset, which tend to convey similar information. Additionally, the shorter time series used in our study compared to other machine learning research might prompt Lasso and Elastic Net methods to disregard less critical variables due to the limited number of observations available.

Secondly, regarding the accuracy of the EML models, each model significantly surpasses the benchmark for predicting both industrial production and inflation at a 5% significance level, except for the Boosting and Random Forest models for industrial production at $h = 1$, which achieve significance only at the 10% level. Furthermore, all EML forecast models display higher DA values than the benchmark model. Except for Boosting, these DA values remain stable over time. Bagging emerges as the most effective overall. Given its significant performance for both targets across all horizons at the 5% level and generally the lowest forecast errors, it is the optimal choice for capturing nonlinearities in Slovakia's data. This

finding is particularly relevant as there are no other published studies addressing potential nonlinearities in Slovakia's data in their data driven forecasting models.

Thirdly, we examine three additional model types outside the main categories. The Support Vector Machine (SVM) model surpasses the benchmark at every forecast horizon for both variables, showing compelling DA values. However, its performance for industrial production is inconsistent, generally ranking it among the weaker models in our suite, not counting neural networks. Concerning neural network models, our computational resources were limited, preventing us from fine-tuning these models effectively. As a result, the outcomes from the neural networks are not reliable for our analysis.

Continuing our analysis, we evaluate the forecasting ability of the models using common factors for both industrial production and inflation. Each model surpasses its benchmark; however, they demonstrate significantly better performance in terms of relative MSE and DA values when forecasts are grounded in hard macroeconomic data.

Moreover, we assess the performance of machine learning models trained on soft indicator data. We anticipate that forecasts based on soft indicators would not yield lower errors than those based on hard macroeconomic data. However, we expect them to surpass the benchmark in predicting the direction of change in the target variable, and our assumption is confirmed**.** In our conclusion, indicators serve their intended purpose effectively. By leveraging an extensive dataset of soft indicators and employing machine learning models, we can predict the direction of change with a higher likelihood than if we solely relied on the benchmark model.

We continue with by summarizing the results involving the post-COVID sample. In our exploration of post-COVID results, we find patterns somewhat similar to earlier findings. In addition, the introduction of COVID-19 data into the sample enhances the performance of machine learning models. Notably, in the case of industrial production, both Lasso and Elastic Net models achieve an increase in performance of approximately 20 percentage points compared to the pre-COVID sample. In the case of inflation forecasting, the Elastic Net model stands out with the highest degree of outperformance. This is a significant enhancement, nearly doubling its relative performance compared to the previous sample without COVID data.

Moreover, the performance of EML models presents a less clear picture. They perform better compared to the pre-COVID sample. On the other hand, neither of them outperforms their combination with either Ridge or Elastic Net. This underscores the effectiveness of our

hybrid approach in substantially improving the forecasting performance of nonlinear models by selecting appropriate variables or optimizing parameter values for training.

The results of models based on common factors is similar to the pre-COVID sample. Additionally, soft indicators demonstrate similar or greater forecasting accuracy than the benchmark model during this period. In the case of industrial production, they outperform the benchmark even in terms of errors.

In this sample, we also evaluate the efficacy of machine learning models as tools for forecast combination. We find that some composite models, which integrate outputs from multiple individual models, can be twice as accurate as their counterparts that solely rely on hard macroeconomic data. This advantage is consistent and becomes even more pronounced as h increases. Moreover, the performance of composite models based on common factors is notably impressive. These common factor-based composites generally outperform the benchmark model in most scenarios. This finding indicates that while individual models relying on common factors might underperform on their own, aggregating information from each through machine learning techniques can significantly enhance overall forecast accuracy. This approach effectively harnesses the collective strength of various models, demonstrating the power of composite forecasting in machine learning applications.

In the post-COVID sample, we also present how each category from Tables A1.1-A1.8 affects the forecasts. Additionally, we show that RLS forecasts almost perfectly match the out-of-sample trajectory of the actual values of the forecasted series. The best performance is delivered by the Elastic Net model, which not only accurately predicts the direction of changes but also the magnitudes, fitting almost perfectly with the actual data. This high level of accuracy leads us to conclude that in our case the Elastic Net model is the most precise predictor of inflation, even in highly volatile environments. While the model's performance in capturing the magnitudes of change slightly deteriorates at $h = 12$, the ability to forecast inflation in a highly uncertain and volatile period with such precision for one and three months ahead remains an exceptional achievement.

To conclude, most of the machine learning models outperform the benchmark model by a huge margin, and they perform substantially better than when using the pre-COVID sample. Multiple reasons can be given for this outperformance. Generally, the models have more data to learn with the prolongation of the time series by 4 years. To add, the benchmark model’s

performance can deteriorate substantially as the simple model might not be able to capture an abrupt and unexpected shock such as COVID.

After summarizing our results, it is clear that none of the scientific hypotheses from Chapter 2 can be rejected.

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	Hybrid models can enhance the forecast accuracy of nonlinear methods.	✓
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	Regularization based methods based on hard macroeconomic data deliver better performance than dimensional reduction based on PCA.	✓

To end this chapter with, we formulate recommendations based on our results. First of all, it is important to acknowledge that machine learning methods at present largely remain "black boxes", as highlighted by Masini et al. (2021). Masini et al. (2021) also state that although various interpretative techniques are available, there is no consensus within the academic community on their adequacy, especially when compared to more traditional models like VAR, which allow for straightforward interpretation through impulse responses or variance decompositions. Masini et al. (2021), however, highlight that this area of research is booming at the time of writing this thesis. It is important to note, therefore, that the main contributions are not in the field of inference but on the field of forecasting capacity.

Firstly, the general high statistical significance at the 5% level across these models indicates that regularization significantly enhances forecast accuracy, by more than 90% in some cases involving Lasso and Elastic Net, despite their previous non-utilization in Slovakia. Regularization thus provides more accurate and robust estimates during both pre- and post-COVID periods. As such, it seems advantageous to explore further the implementation of regularization techniques in macroeconomic forecasting by applied macroeconomists. This shift could potentially improve the precision of economic forecasts by leveraging the detailed

and direct measurements of economic activity provided by hard data, thus enhancing the overall reliability of the forecasting models in various economic conditions.

Secondly, the robust performance of various models highlights the importance of nonlinear relationships, briefly discussed in Chapter 1, in the Slovakian data generating process. These factors should be taken into account by macroeconomists when preparing forecasts. The significance of nonlinearities and interactions among variables is further emphasized by the successful use of nonlinear machine learning methods. For instance, the relative Mean Squared Error (MSE) values for the Boosting model in the case of inflation are comparable to those of the Lasso model in the case of inflation. This similarity suggests that even in smaller datasets, typical of smaller economies, the lack of regularization does not hinder these methods from identifying crucial relationships within the data. Our findings strongly support the inclusion of nonlinear data-driven models in forecasting frameworks, given their demonstrated capability to effectively capture significant nonlinear effects. This approach could enhance the accuracy and relevance of economic forecasts by incorporating the complex dynamics often present in real-world data. To further improve these nonlinear methods, one should apply regularization first and then use EML techniques on the regularized inputs.

Finally, it is advisable for policymakers to have machine learning models set up in advance. When soft indicator data becomes available, it can be quickly fed into these models to more reliably predict the direction of the economy. This approach provides policymakers with a proactive tool to effectively gauge future economic trends. By utilizing real-time data inputs, these models can offer timely insights, allowing for more informed and responsive economic decision-making. This method underscores the importance of readiness and the strategic use of technology in economic forecasting.

In summary, our principal recommendation is that macroeconomists in Slovakia should begin to utilize the models in the two main categories, which demonstrate strong performance in forecasting. These models are effective at capturing nonlinearities in the data generating process, and also highlight the critical role of regularization. As evidenced in our analysis, both elements – nonlinearities and regularization – are crucial for accurately predicting economic trends. This approach ensures that macroeconomic forecasts are not only robust but also reflective of the complex dynamics that characterize the economic landscape.

Conclusion

In this thesis, we use machine learning (ML) methods to forecast industrial production and inflation in Slovakia. Our study is the first to apply machine learning (ML) methods for macroeconomic forecasting in Slovakia, and more broadly, in any small open industrialized economy in a monetary union. We utilize a comprehensive dataset from the National Bank of Slovakia, which is publicly available. This dataset is divided into pre-COVID and post-COVID periods and we apply a rolling window scheme, allowing us to account for time-variance in parameter estimates and enhance the robustness of our results.

We focus on two categories of ML models, one to address nonlinearities in the data, and one to examine the impact of regularization on macroeconomic forecast outcomes. Specifically, we also employ Support Vector Machine models to counter the generalized critique by Makridakis et al. (2018), as discussed in Chapter 1, demonstrating that their criticisms do not hold for Slovakia.

Our contributions to macroeconomic forecasting are both diverse and impactful. First, we introduce a hybrid method, inspired by Medeiros et al. (2019), which is adept at capturing nonlinearities and variable interactions. This approach is especially beneficial in post-socialist Eastern European economies where datasets tend to be limited in length. By regularizing these datasets prior to applying nonlinear methods, we significantly boost the performance of these models. Second, we assess the efficacy of regularization versus principal component analysis (PCA) in managing datasets with reduced dimensions. Our results show that regularization, a machine learning (ML) technique, provides more accurate forecasts. Third, we investigate the utility of ML methods in combining forecasts, finding that they improve the accuracy of individual predictions. Fourth, we show that regularization markedly improves forecasting capabilities when compared to conventional benchmarks. Fifth, we use Ensemble ML models to identify and model nonlinearities in the data. To our knowledge, this marks the first time these techniques have been applied using ML in a small, open industrialized economy within a monetary union, characterized by a brief dataset. Sixth, we break new ground by evaluating the directional accuracy of ML models, a vital aspect often overlooked in favor of focusing solely on the magnitude of errors. Seventh, we provide an exhaustive analysis of ML model performance across both pre-COVID and post-COVID periods, noting superior performance during times of heightened economic uncertainty and volatility. Despite a dataset spanning only 16 years with crises at both ends, one model notably forecasts inflation one to three months

ahead with complete accuracy. Eighth, we illustrate how ML methods more effectively discern trends from soft indicators. Ninth, our findings collectively address and counter the criticisms posed by Makridakis et al. (2018), which we discuss in detail in Chapter 1. We show that simple methods, such as the SVM do not perform well, but complex methods are capable of delivering statistically significant outperformance.

In conclusion, institutions responsible for official macroeconomic forecasts in Slovakia should start testing the practical applications of ML models to integrate new information into their forecasts. Additionally, researchers need to focus more on combining linear and nonlinear machine learning approaches, directional accuracy measures, the use of soft indicators, the potential of ML for creating composite forecasts, and favoring regularization over PCA. Further exploration into demystifying the ML black box and conducting inference should also be a priority for future research.

Resume

Prognózovanie zohráva kľúčovú úlohu pri hodnotení ekonomického stavu a usmerňovaní ekonomickej politiky. Je základom pre tvorbu vládnych rozpočtov a je nevyhnutné pre tvorcov politiky, ako sú centrálne banky, aby mohli načasovať intervencie na základe predpovedí kľúčových ekonomických ukazovateľov, ako sú hrubý domáci produkt (HDP), inflácia a nezamestnanosť. Súčasné modely však často nedokážu zachytiť skutočnú dynamiku medzi ekonomickými premennými. Napríklad Medeiros et al. (2019) ukazujú, ako vlády a medzinárodné orgány, najmä ECB, majú tendenciu neustále nadhodnocovať projekcie inflácie. Takéto rozdiely môžu viesť k významným stratám v oblasti blahobytu a skresliť očakávania inflácie, čo podčiarkuje potrebu presnejších predikčných modelov.

S príchodom big data, vylepšených výpočtových schopností a pokrokov v oblasti štatistického učenia majú ekonómovia teraz prístup k rôznym novým metódam, vrátane tých založených na strojovom učení. Tieto metódy sa stali v poslednom desaťročí čoraz populárnejšími v makroekonomických aplikáciách, najmä v posledných piatich až šiestich rokoch.

V tejto práci používame metódy strojového učenia (ML) na prognózovanie priemyselnej výroby a inflácie na Slovensku. Naša štúdia je prvou, ktorá aplikuje metódy strojového učenia (ML) na makroekonomické prognózy na Slovensku a všeobecnejšie, v akejkoľvek malej otvorenej industrializovanej ekonomike v menovej únii. Využívame komplexnú databázu Národnej banky Slovenska, ktorá je verejne dostupná. Naša vzorka je rozdelená na pre-COVID a post-COVID obdobia, na ktoré aplikujeme schému posuvného okna, čo nám umožňuje zohľadniť časovú variabilitu v odhadoch parametrov a zvýšiť robustnosť našich výsledkov.

Zameriavame sa na dve kategórie ML modelov: jednu na riešenie nelinearít v dátach a druhú na skúmanie vplyvu regularizácie na výsledky makroekonomických prognóz. Tiež používame model Support Vector Machine, aby sme vyvrátili všeobecnú kritiku Makridakisa et al. (2018), a ukázali, že ich kritika neplatí pre Slovensko.

Práca je usporiadaná nasledovne. Prvá kapitola sa venuje prehľadu existujúcej literatúry o makroekonomickom prognózovaní pomocou metód strojového učenia. Poskytuje všeobecný prehľad, zdôrazňuje kľúčové zistenia v tejto oblasti a odôvodňuje použitie nášho východiskového modelu (benchmark). Druhá kapitola uvádza naše hlavné a čiastkové ciele spolu s vedeckými hypotézami. Tretia kapitola poskytuje vyčerpávajúci prehľad našej

metodológie, vrátane matematického formulovania modelov, kódov, ktoré používame, procesu hodnotenia predpovedí a krokov predspracovania údajov. Štvrtá kapitola predstavuje výsledky pre vzorky pred a po COVIDe. Piata kapitola sumarizuje a interpretuje výsledky a zároveň poskytuje odporúčania. Posledná kapitola obsahuje záver.

Naším hlavným cieľom je komplexne zhodnotiť predikčnú výkonnosť rôznych vybraných modelov strojového učenia a porovnať tieto výsledky s ekonometrickým benchmark modelom. Prognózujeme priemyselnú produkciu a infláciu pre obdobia pred a po COVIDe. Naše prínosy k literatúre sú metodologické aj empirické.

Najprv predstavíme naše metodologické prínosy, ktoré zavádzajú tri prístupy, ktoré nie sú v súčasnej literatúre široko diskutované, alebo kde je dôkazov obmedzené množstvo. Po prvé, hodnotíme účinnosť techník redukcie dimenzionality, pričom sa inšpirujeme prácou Maehashiho a Shintaniho (2020). Principal Component Analysis (PCA) sa považuje za jeden z najlepších spôsobov na redukciu dimenzie dátovej sady. Naše zistenia naznačujú, že všeobecne sú regularizačné techniky na Slovensku preferované pred PCA redukciou. Pozorovaná slabšia výkonnosť predpovedí založených na spoločných faktoroch v porovnaní s tými, ktoré sú založené na tvrdých dátach, naznačuje možný prechod v prognózovaní od techník redukcie dimenzií založených na faktoroch k regularizačným metódam využívajúcim tvrdé dáta na Slovensku. Obe stratégie sú navrhnuté tak, aby extrahovali esenciálne informácie pri minimalizovaní počtu prediktorov; avšak regularizácia používajúca tvrdé dáta ukazuje v našej vzorke väčšiu presnosť.

Po druhé, skúmame kombináciu regularizácie a nelineárnych metód, inšpirovanú Medeirosom et al. (2019). Naša metóda sa však líši, pretože používame alternatívne metódy a vykonávame dôkladnejšie hodnotenie výkonnosti. Medeiros et al. (2019) zistili, že kombinácia nie je obzvlášť prospešná. Naopak, naša analýza odhaľuje, že na Slovensku sú hybridné modely presnejšie ako jednoduché nelineárne modely, čo je zobrazené v tabuľkách 43 a 60. Výkonnosť nelineárnych metód zlepšujeme tým, že najprv aplikujeme regularizáciu na zjednodušenie informačnej sady, nasledovanú nelineárnym odhadom a predikciou. Táto efektivita pravdepodobne vyplýva zo zníženia počtu premenných, čo umožňuje dosiahnuť komplexné pokrytie kľúčových nelinearít a interakcií premenných s menším počtom stromov. Navyše, ako poznamenávajú Medeiros et al. (2019), keď je model Random Forest aplikovaný na sadu regularizovaných premenných, jeho schopnosť rozpoznať dôležité interakcie premenných je výrazne zosilnená, ale stáva sa ťažším zachytiť nelineárne vzťahy. V takom prípade, ak je

cieľom skúmať vplyvy nelinearít podrobnejšie, možno zvážiť iné nelineárne metódy ako Boosting, ktoré sú špecificky navrhnuté na efektívne zvládnutie takýchto komplexít.

Po tretie, kombinácia prognóz je kľúčovou témou v literatúre o prognózovaní. Metódy strojového učenia sú prirodzene vhodné na kombinovanie prognóz, napriek tomu je výrazný nedostatok štúdií testujúcich tento prístup, okrem Arauja a Gaglianone (2023). Riešime túto medzeru tým, že ukazujeme, že použitie individuálnych modelov strojového učenia ako nástrojov na kombinovanie predpovedí môže výrazne zlepšiť výkonnosť niektorých modelov. Zistili sme, že niektoré kombinované modely, ktoré integrujú výstupy z viacerých individuálnych modelov, môžu byť dvakrát presnejšie ako ich náprotivky, ktoré sa spoliehajú iba na tvrdé makroekonomické údaje. Toto zlepšenie je obzvlášť zrejmé pri predpovedaní inflácie a v menšej miere pri predpovedaní priemyselnej produkcie.

Teraz prejdeme k diskusii o našich empirických prínosoch, počnúc prognózami priemyselnej produkcie a inflácie pomocou vzorky pred COVIDom.

Po prvé, každý RLS model neustále prekonáva benchmark model, so štatistickou významnosťou na 5%-nej hladine významnosti. Len v dvoch prípadoch, týkajúcich sa priemyselnej produkcie, je tento rozdiel významný na 10%-nej hladine významnosti. V prípade inflácie je každý rozdiel významný na 5%-nej hladine významnosti. Navyše, každý RLS predikčný model dosahuje podstatne vyššie DA hodnoty ako benchmark model pre priemyselnú produkciu. Hoci je DA pre modely strojového učenia o niečo nižšia pre infláciu, stále prekonávajú benchmark. Tieto výsledky identifikujú model Elastic Net ako najefektívnejší v tejto kategórii. Ďalej, pre priemyselnú produkciu zohráva výber premenných pomocou Lasso dôležitejšiu úlohu ako zmenšovanie koeficientov pomocou Ridge. Táto situácia je obrátená pri predikciách inflácie. Zhrnuté, RLS modely neustále prekonávajú benchmark model v každom horizonte. Tento robustný výkon je očakávaný kvôli prítomnosti mnohých korelovaných premenných v dátovej sade, ktoré majú tendenciu poskytnúť podobné informácie pre odhad. Navyše, kratšie časové rady použité v našej štúdii v porovnaní s iným výskumom strojového učenia môžu viesť k tomu, že metódy Lasso a Elastic Net ignorujú menej dôležité premenné kvôli obmedzenému počtu dostupných pozorovaní.

Po druhé, čo sa týka presnosti EML modelov, každý model výrazne prekonáva benchmark model pri prognózovaní priemyselnej produkcie aj inflácie na 5%-nej hladine významnosti s výnimkou modelov Boosting a Random Forest pre priemyselnú produkciu pri $h = 1$, kde dosahujú štatistickú významnosť len na 10%-nej hladine významnosti. Navyše, všetky EML

predikčné modely vykazujú vyššie hodnoty DA ako benchmark. S výnimkou Boosting, tieto hodnoty DA zostávajú stabilné v čase. Bagging sa ukazuje ako celkovo najúčinnejší. Vzhľadom na jeho významný výkon pre oba ciele naprieč všetkými horizontmi na úrovni 5 % a všeobecne najnižšie predikčné chyby, je to optimálna voľba pre zachytenie nelinearít v dátach Slovenska. Tento nález je obzvlášť relevantný, pretože neexistujú iné publikované štúdie, ktoré by riešili potenciálne nelinearity na slovenských dátach.

Po tretie, skúmame tri ďalšie typy modelov mimo hlavných kategórií. Model Support Vector Machine (SVM), ktorý prekonáva benchmark model v každom predikčnom horizonte pre obe premenné, pričom vykazuje presvedčivé hodnoty DA. Jeho výkon pre priemyselnú produkciu je však nekonzistentný, čo ho zaraďuje medzi slabšie modely v našom súbore, neberúc do úvahy neurónové siete. Čo sa týka modelov neurónových sietí, naše výpočtové možnosti boli obmedzené, čo nám bránilo efektívne doladiť tieto modely. Výsledkom je, že výsledky z neurónových sietí nie sú spoľahlivé pre našu analýzu.

Pokračujúc v našej analýze, hodnotíme predikčnú schopnosť modelov pomocou spoločných faktorov pre priemyselnú produkciu a infláciu. Každý model prekonáva svoj benchmark model; avšak modely vykazujú výrazne lepšiu výkonnosť z hľadiska relatívneho MSE a hodnôt DA, keď sú predikcie založené na tvrdých makroekonomických dátach.

Ďalej posudzujeme výkonnosť modelov strojového učenia, ktoré boli trénované na mäkkých indikátorových dátach. Očakávame, že predikcie založené na mäkkých indikátoroch nebudú mať nižšie chyby ako tie, ktoré sú založené na tvrdých makroekonomických dátach. Avšak predpokladáme, že prekonajú benchmark model v prognózovaní smeru zmeny cieľovej premennej, čo sa potvrdzuje. Na záver môžeme povedať, že indikátory efektívne plnia svoj zamýšľaný účel. Využitím rozsiahlej dátovej sady mäkkých indikátorov a použitím modelov strojového učenia môžeme predpovedať smer zmeny s vyššou pravdepodobnosťou, ako keby sme sa spoliehali len na benchmark model.

Pokračujeme so sumarizáciou výsledkov týkajúcich sa post-COVID obdobia. Pri skúmaní post-COVID výsledkov nachádzame vzory čiastočne podobné skorším zisteniam. Okrem toho, zahrnutie COVID-19 údajov do vzorky zlepšuje výkon modelov strojového učenia. Významné je, že v prípade priemyselnej výroby dosahujú modely Lasso a Elastic Net zvýšenie výkonu približne o 20 percentuálnych bodov v porovnaní s pre-COVID vzorkou. V prípade predikcie inflácie vyniká model Elastic Net s najlepšími výsledkami. Toto je významné

zlepšenie, ktoré takmer zdvojnásobuje jeho relatívny výkon v porovnaní s predchádzajúcou vzorkou bez COVID údajov.

Navyše, výkon EML modelov predstavuje menej jasný obraz. Dosahujú lepší výkon v porovnaní s pre-COVID vzorkou. Na druhej strane, žiadny z nich neprekonáva svoju kombináciu s Ridge alebo Elastic Net. Toto podčiarkuje efektívnosť nášho hybridného prístupu pri podstatnom zlepšení predikčného výkonu nelineárnych modelov výberom vhodných premenných alebo optimalizáciou hodnôt parametrov pre tréning.

Výsledky modelov založených na spoločných faktoroch sú podobné ako pri pre-COVID vzorke. Okrem toho, soft indikátory vykazujú podobnú alebo vyššiu presnosť predikcií ako benchmark model počas tohto obdobia. V prípade priemyselnej výroby prekonávajú benchmark aj z hľadiska chýb.

V tejto vzorke tiež hodnotíme účinnosť modelov strojového učenia ako nástrojov na kombináciu predikcií. Zistili sme, že niektoré kompozitné modely, ktoré integrujú výstupy z viacerých individuálnych modelov, môžu byť dvakrát tak presné ako ich náprotivky, ktoré sa spoliehajú výlučne na tvrdé makroekonomické údaje. Táto výhoda je konzistentná a stáva sa ešte výraznejšou s rastúcim h. Navyše, výkon kompozitných modelov založených na spoločných faktoroch je pozoruhodne pôsobivý. Tieto modely založené na spoločných faktoroch vo všeobecnosti prekonávajú benchmark model vo väčšine scenárov. Toto zistenie naznačuje, že zatiaľ čo individuálne modely spoliehajúce sa na spoločné faktory môžu samostatne dosahovať horšie výsledky, agregácia informácií z každého modelu prostredníctvom techník strojového učenia môže výrazne zvýšiť celkovú presnosť predikcií. Tento prístup efektívne využíva kolektívnu silu rôznych modelov, čím demonštruje silu kompozitného predikovania v aplikáciách strojového učenia.

Vo vzorke post-COVID tiež uvádzame, ako každá kategória z tabuliek A1.1-A1.8 ovplyvňuje predikcie. Okrem toho ukazujeme, že predikcie RLS takmer dokonale zodpovedajú trajektórii skutočných hodnôt predikovaných radov. Najlepší výkon dosahuje model Elastic Net, ktorý nielen presne predpovedá smer zmien, ale aj ich rozsah, pričom takmer dokonale zodpovedá skutočným údajom. Táto vysoká úroveň presnosti nás vedie k záveru, že v našom prípade je model Elastic Net najpresnejšou metódou predikcie inflácie, dokonca aj v prostredí s vysokou volatilitou. Hoci výkon modelu pri zachytávaní rozsahu zmien mierne klesá pri h = 12, schopnosť predpovedať infláciu v období vysokej neistoty a volatility s takou presnosťou na jeden a tri mesiace dopredu zostáva výnimočným úspechom.

Na záver, väčšina modelov strojového učenia prekonáva benchmark model o veľký rozdiel a dosahujú podstatne lepšie výsledky ako pri použití pre-COVID vzorky. Pre túto výkonnosť možno uviesť viacero dôvodov. Vo všeobecnosti majú modely viac údajov na učenie vďaka predĺženiu časovej rady o 4 roky. Navyše, výkon benchmark modelu môže výrazne klesnúť, pretože jednoduchý model nemusí byť schopný zachytiť náhly a neočakávaný šok, ako je COVID.

Na záver formulujeme odporúčania na základe našich výsledkov. V prvom rade je dôležité spomenúť, že metódy strojového učenia v súčasnosti zostávajú vo veľkej miere „čiernymi skrinkami“, ako zdôrazňujú Masini et al. (2021). Masini et al. (2021) tiež uvádzajú, že hoci sú dostupné rôzne interpretačné techniky, neexistuje konsenzus v rámci akademickej obce o ich primeranosti, najmä v porovnaní s tradičnejšími modelmi ako VAR, ktoré umožňujú priamočiaru interpretáciu. Masini et al. (2021) však zdôrazňujú, že táto oblasť výskumu v čase písania tejto práce prudko rastie. Preto je dôležité poznamenať, že hlavné prínosy nie sú v oblasti identifikácie štrukturálnych súvislostí, ale v oblasti predikčnej kapacity.

Po prvé, všeobecne vysoká štatistická významnosť na úrovni 5 % naprieč týmito modelmi naznačuje, že regularizácia výrazne zvyšuje presnosť predikcií, a to o viac ako 90 % v niektorých prípadoch zahŕňajúcich Lasso a Elastic Net, napriek ich predchádzajúcemu nevyužívaniu na Slovensku. Regularizácia tak poskytuje presnejšie a robustnejšie odhady počas pre-COVID aj post-COVID období. Preto sa zdá byť výhodné ďalej skúmať implementáciu regularizačných techník v makroekonomických predikciách aplikovanými makroekonómami. Tento posun by mohol potenciálne zlepšiť presnosť ekonomických prognóz využitím podrobných a priamych meraní ekonomickej aktivity poskytovaných tvrdými dátami, čím by sa zvýšila celková spoľahlivosť predikčných modelov v rôznych ekonomických podmienkach.

Po druhé, robustný výkon rôznych modelov zdôrazňuje význam nelineárnych vzťahov, stručne diskutovaných v Kapitole 1. Tieto faktory by mali brať makroekonómovia do úvahy pri príprave predikcií. Význam nelinearít a interakcií medzi premennými je ďalej zdôraznený úspešným použitím nelineárnych metód strojového učenia. Napríklad relatívne hodnoty MSE pre model Boosting sú porovnateľné s hodnotami modelu Lasso v prípade inflácie. Táto podobnosť naznačuje, že aj pri menších dátových výberových súboroch, typických pre menšie ekonomiky, nedostatok regularizácie nebráni týmto metódam v identifikácii kľúčových vzťahov v rámci dát. Naše zistenia silne podporujú zahrnutie nelineárnych dátovo riadených modelov do predikčných rámcov, vzhľadom na ich preukázanú schopnosť efektívne zachytiť významné nelineárne efekty. Tento prístup by mohol zvýšiť presnosť a relevantnosť

ekonomických prognóz tým, že by do nich zahrnul komplexnú dynamiku, často prítomnú v reálnych dátach. Na ďalšie zlepšenie týchto nelineárnych metód by sa mala najprv aplikovať regularizácia a potom použiť techniky EML na regularizované vstupy.

Nakoniec je vhodné, aby mali tvorcovia politík vopred pripravené modely strojového učenia. Keď sa sprístupnia údaje o mäkkých indikátoroch, môžu byť rýchlo zadané do týchto modelov na spoľahlivejšie predpovedanie smeru ekonomiky. Tento prístup poskytuje tvorcom politík proaktívny nástroj na efektívne hodnotenie budúcich ekonomických trendov. Využitím vstupov v reálnom čase môžu tieto modely ponúknuť včasné postrehy, čo umožňuje informovanejšie a pružnejšie ekonomické rozhodovanie. Tento spôsob zdôrazňuje dôležitosť pripravenosti a strategického využitia technológií v ekonomických predikciách.

Zhrnutím je naše hlavné odporúčanie, aby makroekonómovia na Slovensku začali využívať modely v dvoch hlavných kategóriách, ktoré preukazujú silný výkon v predikciách. Tieto modely sú účinné pri zachytávaní nelinearít v procese generovania dát a tiež zdôrazňujú kľúčovú úlohu regularizácie. Ako sme ukázali v našej analýze, oba prvky – nelinearity a regularizácia – sú rozhodujúce pre presné predpovedanie ekonomických trendov. Tento prístup zaisťuje, že makroekonomické predpovede sú nielen robustné, ale aj odrážajú komplexnú dynamiku, ktorá charakterizuje ekonomiku krajiny.

Aby sme to zhrnuli, naše prínosy k makroekonomickému prognózovaniu sú viaceré a sú zároveň dôležité. Po prvé, zavádzame hybridnú metódu inšpirovanú Medeirosom et al. (2019), ktorá je úspešná v zachytávaní nelinearít a interakcií premenných. Tento prístup je obzvlášť prospešný v postsocialistických východoeurópskych ekonomikách, kde sú dátové súbory obvykle krátke. Regularizáciou týchto dátových súborov pred aplikovaním nelineárnych metód výrazne zvyšujeme výkon týchto modelov. Po druhé, hodnotíme účinnosť regularizácie v porovnaní s analýzou hlavných komponentov (PCA). Naše výsledky ukazujú, že regularizácia, technika strojového učenia (ML), poskytuje presnejšie prognózy. Po tretie, skúmame užitočnosť ML metód pri kombinovaní prognóz a zistíme, že zlepšujú presnosť individuálnych modelov. Po štvrté, ukazujeme, že regularizácia výrazne zlepšuje predikčné schopnosti v porovnaní s konvenčnými benchmark modelmi. Po piate, používame Ensemble ML modely na identifikáciu a modelovanie nelinearít v dátach. Pokiaľ je nám známe, toto je prvýkrát, čo boli tieto techniky aplikované pomocou ML v malej otvorenej industrializovanej ekonomike v menovej únii. Po šieste, prichádzame s novým prístupom hodnotenia smerovej presnosti ML modelov, čo je dôležitý aspekt, ktorý sa často prehliada v prospech zamerania sa iba na veľkosť chýb. Po siedme, poskytujeme vyčerpávajúcu analýzu výkonu ML modelov v pre-COVID a

post-COVID obdobiach, pričom zaznamenávame lepší výkon počas období zvýšenej ekonomickej neistoty a volatility. Napriek krátkemu obdobiu, ktoré pokrýva iba 16 rokov s krízami na oboch koncoch, jeden model prognózuje infláciu na jeden až tri mesiace dopredu s úplnou presnosťou. Po ôsme, ilustrujeme, ako ML metódy efektívnejšie rozlišujú trendy z mäkkých indikátorov. Po deviate, naše zistenia kolektívne adresujú a vyvracajú kritiku Makridakisa et al. (2018), ktorú podrobne diskutujeme v Kapitole 1. Ukazujeme, že jednoduché metódy, ako napríklad SVM, nefungujú dobre, ale komplexné metódy sú schopné dosiahnuť štatisticky významnú výkonnosť.

References

1. ADÁMEK, Robert - SMEEKES, Stephan - WILMS, Ines. LASSO inference for high-dimensional time series. In *arXiv Technical Report* [online]. New York: Cornell University, 7.2020, vol. 2007, p. 20 [15.4.2024]. Available at: https://arxiv.org/abs/2007.10952
2. ARAUJO, Gustavo Silva - GAGLIANONE, Wagner Piazza. Machine learning methods for inflation forecasting in Brazil: New contenders versus classical models. In *Latin American Journal of Central Banking* [online]. Elsevier, 6.2023, vol. 4, issue 2, pp. 1–29 [15.7.2023]. ISSN 2666-1438. Available at: https://www.sciencedirect.com/science/article/pii/S2666143823000042
3. ARMSTRONG, Scott J. Combining Forecasts. In *University of Pennsylvania Scholarly Commons Marketing Papers* [online]. Philadelphia: University of Pennsylvania, 6.2001, vol. 34 [12.2.2023]. Available at: https://repository.upenn.edu/entities/publication/2d255989-3e75-46af-9922-8b1649254bc7
4. ATHEY, Susan. The impact of machine learning on economics. In *The Economics of Artificial Intelligence*. Washington: National Bureau of Economic Research, 2019, pp. 507–547. ISBN 978-0-226-61347-5.
5. ATKESON, Andrew - OHANIAN, Lee E. Are Phillips curves useful for forecasting inflation? In *Federal Reserve Bank of Minneapolis Quarterly Review* [online]. Minneapolis: Federal Reserve Bank of Minneapolis, 2001, vol. 25, issue 11, pp. 2–11 [4.1.2023]. ISSN 0271-5287. Available at: https://www.minneapolisfed.org/research/quarterly-review/are-phillips-curves-useful-for-forecasting-inflation
6. BAI, Jushan - NG, Serena. Determining the number of factors in approximate factor models [online]. In *Econometrica*. Econometric Society, 12 .2003, vol. 70, issue 1, pp. 191–221 [5.9.2022]. ISSN 1468-0262. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1111/1468-0262.00273
7. BATES, John M.- GRANGER, Clive W.J. The combination of forecasts. In *Journal of the Operational Research Society* [online]. Taylor & Francis, 12.1969, vol. 20, pp. 451–468 [7.11.2022]. ISSN 1476-9360. Available at: https://link.springer.com/article/10.1057/jors.1969.103

8. BERGMEIR, Christoph - HYNDMAN, Rob. J. - KOO, Bonsoo. A note on the validity of cross-validation for evaluating autoregressive time series prediction [online]. In *Computational Statistics and Data Analysis*. Elsevier, 4.2018, vol. 120, pp. 70–83 [6.8.2023]. ISSN 0167-9473. Available at: https://robjhyndman.com/publications/cv-time-series/

9. BOLHUIS, Marjin A. - RAYNER, Brett. Deus ex Machina? A Framework for Macro Forecasting with Machine Learning. In *IMF Working Paper* [online]. Washington: International Monetary Fund, 2.2020, vol. 45, 25 p. [8.1.2023]. ISSN 1018-5941. Available at: https://www.imf.org/en/Publications/WP/Issues/2020/02/28/Deus-ex-Machina-A-Framework-for-Macro-Forecasting-with-Machine-Learning-49094

10. BOX, George E. P. - JENKINS, Gwilym M. *Time Series Analysis: Forecasting and Control*. 1st ed. San Francisco: Holden-Day, 1970. p. 553. ISBN 978-08-1621-094-7.

11. BREIMAN, Leo. Bagging predictors. In *Machine Learning* [online]. Springer, 8.1996, vol. 24, issue 2, pp. 123–140 [7.12.2023]. ISSN 1573-0565. Available at: https://link.springer.com/article/10.1007/BF00058655

12. BREIMAN, Leo. Statistical Modeling: The Two Cultures. In *Statistical Science* [online]. Institute of Mathematical Statistics, 8.2001, vol. 16, issue 3, pp. 199–231 [15.11.2023]. ISSN 2168-8745. Available at: https://www.jstor.org/stable/2676681

13. BREIMAN, Leo. Random forest. In *Machine Learning* [online]. Springer, 10.2001. vol. 45, issue. 1, pp. 5–32 [7.12.2023]. ISSN 1573-0565. Available at: https://link.springer.com/article/10.1023/A:1010933404324

14. BÜRGI, Constantin. Can a subset of forecasters beat the simple average in the SPF?. In *Research Program on Forecasting Working Paper* [online]. Washington: The George Washington University, 3.2015, issue 001, 15 p. [5.6.2023]. Available at: https://www2.gwu.edu/~forcpgm/2015-001.pdf

15. CHAKRABORTY, Chiranjit - JOSEPH, Andreas. Machine Learning at Central Banks. In *Bank of England Staff Working Papers* [online]. London: Bank of England, 9.2017, issue 674, 89 p. [11.6.2023]. Available at: https://www.bankofengland.co.uk/-/media/boe/files/working-paper/2017/machine-learning-at-central-banks.pdf

16. CIBUĽA, Milan - TKÁČ, Michal. Comparison of Machine Learning Algorithms for Creation of a Bitcoin Price Prediction Model. In *Politická ekonomie* [online]. Vysoká

škola ekonomická v Praze, 8.2023, vol. 71, issue 5, pp. 496–517 [2.4.2024]. ISSN 2336-8225. Available at: https://doi.org/10.18267/j.polek.1397

17. CLEMENTS, Michael P. - HENDRY, David F. *The Oxford Handbook of Economic Forecasting*. 1st ed. Oxford: Oxford University Press, 2011. 744 p. ISBN 978-0-195-39864-9.

18. CONSTANTINI, Mauro - Cuaresma, Jesús C. - Hlouskova, Jaroslava. Forecasting Errors, Directional Accuracy and Profitability of Currency Trading: The case of EUR/USD Exchange Rate. In *Journal of Forecasting* [online]. Wiley, 2016, vol. 35, pp. 652–668 [17.12.2023]. ISSN 1099-131X. Available at: https://ideas.repec.org/a/wly/jforec/v35y2016i7p652-668.html

19. COULOMBE, Philippe Goulet - LEROUX, Maxime - STEVANOVIC, Dalibor - SURPRENANT, Stéphane. How is machine learning useful for macroeconomic forecasting? In *Journal of Applied Econometrics* [online]. Wiley, 5.2022, vol. 37, issue 5, pp. 920–964 [15.7.2023]. ISSN 1099-1255. Available at: https://onlinelibrary.wiley.com/doi/abs/10.1002/jae.2910

20. DAVIS, Richard A. - MIKKEL, Nielsen S. Modeling of time series using random forests: Theoretical developments. In *Electronic Journal of Statistics* [online]. Institute of Mathematical Statistics, 8.2020, vol. 14, issue 2, pp. 3644–3671 [2.4.2024]. ISSN 1935–7524. Available at: https://doi.org/10.1214/20-EJS1758

21. DICKEY, David A. - FULLER, Wayne A. Distribution of the Estimators for Autoregressive Time Series with a Unit Root. In *Journal of the American Statistical Association* [online]. Taylor & Francis, 6.1979, *v*ol. 74, issue 366, pp. 427–431 [15.12.2023]. ISSN 1537-274X. Available at: https://www.jstor.org/stable/2286348

22. DIEBOLD, Francis X. - MARIANO, Robert. S. Comparing Predictive Accuracy [online]. In *Journal of Business and Economic Statistics.* Taylor & Francis, 1.2002, vol. 20, issue 1, pp. 134–144 [3.3.2023]. ISSN 1537–2707. Available at: https://doi.org/10.1198/073500102753410444

23. FAUST, Jon - WRIGHT, Jonathan. Forecasting Inflation. In *Handbook of Economic Forecasting*. 2nd ed. Amsterdam: Elsevier, 2013, pp. 2–56. ISBN 978-0-444-53683-9

24. FISZEDER, Piotr - ORESZKO, Witold. Covariance matrix forecasting using support vector regression. In *Applied Intelligence* [online]. Springer, 2.2021, vol. 51, issue 1,

pp. 7029–7042 [19.4.2024]. ISSN 1573-7497. Available at: https://link.springer.com/article/10.1007/s10489-021-02217-5

25. FRIEDMAN, Jerome H. Greedy function approximation: a gradient boosting machine. In *Annals of Statistics* [online]. Institute of Mathematical Statistics, 2001, vol. 29, issue 5, pp. 1189–1232 [3.3.2024]. ISSN 0090-5364. Available at: https://www.jstor.org/stable/2699986
26. FRIEDMAN Jerome H. - TIBSHIRANI, Robert - HASTIE, Trevor. Regularization Paths for Generalized Linear Models via Coordinate Descent. *Journal of Statistical Software* [online]. Foundation for Open Access Statistics, 2.2010, vol. 33, issue 1, pp. 1–22 [15.4.2024]. ISSN 1548-7660. Available at: https://www.jstatsoft.org/article/view/v033i01
27. GARCIA, Márcio G.P - MEDEIROS, Marcelo P. – VASCONCELOS, Gabrielo. Real-time inflation forecasting with high-dimensional models: The case of Brazil. In *International Journal of Forecasting* [online]. International Institute of Forecasters, 9.2017, vol. 33, issue 3, pp. 679–693 [6.4.2024]. ISSN 0169-2070. Available at: https://www.sciencedirect.com/science/article/abs/pii/S0169207017300262
28. GIANNONE, Domenico - LENZA, Michele - PRIMICERI, Giorgio E. Economic predictions with big data: the illusion of sparsity. In *European Central Bank Working Paper Series* [online]. Frankfurt: European Central Bank, 4.2021, issue 2542, 50 p. [11.2.2022]. Available at: https://www.ecb.europa.eu/pub/pdf/scpwps/ecb.wp2542~a12964e712.en.pdf
29. GU, Shihao - KELLY, Bryan - XIU, Dacheng. Empirical asset pricing via machine learning. In *NBER Working Paper Series* [online]. Washington: National Bureau of Economic Research, 9.2019, issue 25398, 46 p. [20.3.2023]. Available at: https://www.nber.org/system/files/working_papers/w25398/w25398.pdf
30. HOCHREITER, Sepp. - SCHMIDHUBER, Jürgen. Long short-term memory. In *Neural Computation* [online]. Boston: MIT Press, 1997, vol. 9, issue 8, pp. 1735–1780 [7.8.2023]. ISSN 1530-888X. Available at: https://direct.mit.edu/neco/article-abstract/9/8/1735/6109/Long-Short-Term-Memory?redirectedFrom=fulltext
31. HOERL, Arthur E. - KENNARD, Robert W. Ridge regression: biased estimation for nonorthogonal problems. In *Technometrics* [online]. Taylor & Francis, 2.1970, vol. 12, issue 1, pp. 55–67 [12.12.2022]. ISSN 1537-2723. Available at:

https://www.jstor.org/stable/1267351

32. HYNDMAN, Rob J. - KHANDAKAR, Yeasmin. Automatic time series forecasting: The forecast package for R. In *Journal of Statistical Software* [online]. Foundation for Open Access Statistics, 7.2008, vol. 26, issue 3, pp. 1–22 [18.4.2024]. ISSN 1548-7660. Available at: https://www.jstatsoft.org/article/view/v027i03
33. INOU, Atsushi - KILIAN, Lutz. How useful is bagging in forecasting economic time series? A case study of U.S. consumer price inflation. In *Journal of American Statistical Association* [online]. Taylor & Francis, 6.2008, vol. 103, issue 482, pp. 511–522 [20.10.2023]. ISSN 1537-274X. Available at: https://www.jstor.org/stable/27640075
34. JOSEPH, Andreas - POTJAGAILO, Galina - KALAMARA, Eleni. - CHAKRABORTY, Chiranjit - KAPETANIOS, George. Forecasting UK inflation bottom up. In *Bank of England Staff Working Papers* [online]. London: Bank of England, 3.2021, issue 915, 38 p. [22.11.2023]. Available at: https://www.bankofengland.co.uk/working-paper/2021/forecasting-uk-inflation-bottom-up
35. JUNG, Jin-Kyu - PATNAM, Manasa - TER-MARTIROSYAN, Anna. An Algorithmic Crystall Ball: Forecasts Based on Machine Learning. In *IMF Working Papers* [online]. Washington: International Monetary Fund, 11.2018, issue 230, pp. 1–34 [4.12.2023]. Available at: https://www.imf.org/en/Publications/WP/Issues/2018/11/01/An-Algorithmic-Crystal-Ball-Forecasts-based-on-Machine-Learning-46288
36. KOCK, Anders B. – MEDEIROS, Marcelo – VASCONCELOS, Gabriel. Penalized Time Series Regression. In *Macroeconomic Forecasting in the Era of Big Data: Theory and Practice.* Springer, 2020. Pp. 193–229. ISBN 978-3-030-31149-0.
37. KUHN, Max. Building Predictive Models in R Using the caret Package. In *Journal of Statistical Software* [online]. Foundation for Open Access Statistics, 11.2008, vol. 28, issue 5, pp. 1–26 [17.4.2024]. ISSN 1548-7660. Available at: https://www.jstatsoft.org/article/view/v028i05
38. LECUN, Yann - BENGIO, Yoshua - HILTON, Geoffrey. Deep learning. In *Nature* [online]. Springer Nature, 5.2015, vol. 521, pp. 436–444 [6.12.2022]. ISSN 1476-4687. Available at: https://www.nature.com/articles/nature14539
39. LIAW, Andy - WIENER, Matthew. Classification and Regression by randomForest. In *R News* [online]. CRAN project, 12.2002, vol. 2, issue 3, pp. 18–22 [12.4.2024]. ISSN

1609-3631. Available at: https://journal.r-project.org/articles/RN-2002-022/RN-2002-022.pdf

40. MAEHASHI, Kohei - SHINTANI, Mototsugu. Macroeconomic forecasting using factor models and machine learning: an application to Japan. In *Journal of The Japanese and International Economies* [online]. Elsevier, 6.2020, vol. 58, pp. 1–17 [3.1.2022]. ISSN 0889-1583. Available at: https://ideas.repec.org/a/eee/jjieco/v58y2020ics0889158320300411.html

41. MAHAJAN, Kriti - SRINIVASAN, Anand. Inflation Forecasting in Emerging Markets: A Machine Learning Approach. In *CAFRAL Working Paper* [online]. Reserve Bank of India, 2019, pp. 1-45 [13.11.2023]. Available at: https://ifrogs.org/PDF/CONF_2019/Mahajan_Srinivasan_2019.pdf

42. MAKRIDAKIS, Spryos - SPILIOTIS, Evangelos - ASSIMAKOPOULOS, Vassilios. Statistical and Machine Learning Forecasting Methods: Concerns and Ways Forward. In *PLOS One* [online]. PLOS, 3.2018, vol. 13, issue 3, pp. 1–35 [12.5.2023]. ISSN 1932-6203. Available at: https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0194889

43. MASINI, Ricardo P. - MEDEIROS, Marcelo C. - MENDES, Eduardo F. Machine Learning Advances for Time Series Forecasting. In *The Journal of Economic Surveys* [online]. Wiley, 12.2020, vol. 37, issue 1, pp. 76–111 [7.10.2022]. ISSN 1467-6419. Available at: https://www.semanticscholar.org/paper/Machine-Learning-Advances-for-Time-Series-Masini-Medeiros/67f8e32beb6495579652a8a8317105066799ae1c

44. MASTERS, Timothy. *Practical Neural Network Recipes in C++*. 1st ed. Sand Diego: Academic Press, 1993, p. 493. ISBN 978-0-12-479040-7.

45. MCCRACKEN, Michael. Diverging tests of equal predictive ability. In *Econometrica* [online]. The Econometric Society, 7.2020, vol. 88, issue 4, pp. 1753-1754 [11.4.2024]. ISSN 1468-0262. Available at: https://www.econometricsociety.org/publications/econometrica/2020/07/01/diverging-tests-equal-predictive-ability

46. MEDEIROS, Marcelo P. - MENDES, Eduardo F. Regularization of High-Dimensional Time-Series Models with Non-Gaussian and Heteroskedastic Errors. In *Journal of Econometrics* [online]. Elsevier, 3.2016, vol. 191, pp. 255–271 [11.2.2023]. ISSN 1872-6895. Available at:

https://www.sciencedirect.com/science/article/abs/pii/S0304407615002638

47. MEDEIROS, Marcelo C. - VASCONCELOS, Gabriel - VEIGA, Alvaro - ZILBERMAN, Eduardo. Forecasting Inflation in a Data-Rich Environment: The Benefits of Machine Learning Methods. In *Journal of Business and Economic Statistics* [online]. Taylor & Francis, 8.2019, vol. 39, issue 1, pp. 1–45 [6.6.2023]. ISSN 1537-2707. Available at: https://www.tandfonline.com/doi/full/10.1080/07350015.2019.1637745
48. MULLAINATHAN, Sendhil - SPIESS, Jann. Machine learning: An applied econometric approach [online]. In *Journal of Economic Perspectives*. American Economic Association, 2017, vol. 31, issue 2, pp. 87–106 [15.8.2023]. ISSN 1944-7465. Available at: https://www.aeaweb.org/articles?id=10.1257/jep.31.2.87
49. OBRADOVIĆ, Saša - LOJANICA, Nemanja. Are Inflation Rates Stationary in the Western Balkan Countries? Evidence from Unit Root Tests. In *Politická ekonomie*. Vysoká škola ekonomická v Praze, 9.2022, Vol. 70., Issue 4., pp. 421–439 [2.4.2024]. ISSN 2336-8225. Available at: https://doi.org/10.18267/j.polek.1362
50. SCHAPIRE, Robert E. The strength of weak learnability. In *Machine Learning* [online]. Springer, 6.1990, vol. 5, issue 2, pp. 197–227 [15.12.2023]. ISSN 1573-0565. Available at: https://link.springer.com/article/10.1007/BF00116037
51. SHINTANI, Mototsugu. Nonlinear forecasting analysis using diffusion indexes: an application to Japan. In *Journal of Money, Credit and Banking* [online]. Ohio State University Press, 6.2005, vol. 37, issue 3, pp. 517–538 [11.9.2023]. ISSN 0022-2879. Available at: https://www.jstor.org/stable/3839166
52. STOCK, James H. - WATSON, Mark W. Macroeconomic Forecasting With Diffusion Indexes. In *Journal of Business and Economic Statistics* [online]. American Statistical Association, 4.2002, vol. 20, issue 2, pp. 147–162 [11.5.2023]. ISSN 1537-2707. Available at: https://www.princeton.edu/~mwatson/papers/Stock_Watson_JBES_2002.pdf
53. STOCK, James H. - WATSON, Mark W. Why Has U.S. Inflation Become Harder to Forecast? In *Journal of Money, Credit and Banking* [online]. Ohio State University Press, 2007, vol. 39, pp. 3–33 [23.6.2023]. ISSN 0022-2879. Available at: https://www.princeton.edu/~mwatson/papers/Stock_Watson_JMCB_2007.pdf
54. STOCK, James H. - WATSON, Mark W. Modeling Inflation After the Crisis? In *Technical Report 1*: *National Bureau of Economic Research* [online]. Washington:

National Bureau of Economic Research, 2010, issue 1, 61 p. [12.7.2023]. Available at: https://www.nber.org/system/files/working_papers/w16488/w16488.pdf

55. TIBSHIRANI, Robert. Regression shrinkage and selection via lasso. In *Journal of the Royal Statistical Society: Series B* [online]. Royal Statistical Society, 1996, vol. 58, issue 1, pp. 267–288 [26.3.2022]. ISSN 1467-9868. Available at: https://www.jstor.org/stable/2346178
56. VAN DE GEER, Sara - BÜHLMANN, Peter - RITOV, Ya'acov - DEZEURE, Rube. On asymptotically optimal confidence regions and tests for high-dimensional models. In *Annals of Statistics* [online]. Institute of Mathematical Statistics, 2014, vol. 42, issue 3, pp. 1166–1202 [3.3.2024]. ISSN 0090-5364. Available at: https://www.jstor.org/stable/43556319
57. VAPNIK, Vladimir N. *The nature of statistical learning theory*. 1st ed. Springer: New York, 1995. p. 188. ISBN 978-1-4757-2440-0.
58. VARGAS, Adolfo Rodriguez. Forecasting Costa Rican inflation with machine learning methods. In *Latin American Journal of Central Banking* [online]. Elsevier, 2020, vol. 1, issues 1-4, pp. 1–21 [14.9.2023]. Available at: https://www.sciencedirect.com/science/article/pii/S2666143820300120?via%3Dihub
59. VENABLES, William N. - RIPLEY, Barnabas D. *Modern Applied Statistics with S*. 4th ed. Springer: New York, 2002. p. 495. ISBN 0-387-95457-0.
60. WAGER, Stefan - ATHEY, Susan. Estimation and inference of heterogeneous treatment effects using random forests. In *Journal of the American Statistical Association* [online]. Taylor & Francis, 6.2018, vol. 113, issue 523, pp. 1128–1242 [17.4.2024]. ISSN 0162-1459. Available at: https://www.tandfonline.com/doi/full/10.1080/01621459.2017.1319839
61. ZHANG, Peter G. - QI, Min. Neural network forecasting for seasonal and trend time series. In *European Journal of Operational Research* [online]. Elsevier, January 2005, vol. 160, issue 2, pp. 501–514 [19.4.2024]. ISSN 0377-2217. Available at: https://doi.org/10.1016/j.ejor.2003.08.037
62. ZOU, Hui. and HASTIE, Trevor. Regularization and variable selection via the elastic net. In *Journal of the Royal Statistical Society: Series B* [online]. Royal Statistical Society, 2005, vol. 67, issue 2, pp. 301–320 [11.2.2022]. ISSN 1467-9868. Available at: https://academic.oup.com/jrsssb/article/67/2/301/7109482

Appendix

Appendix 1: List of variables

Table A1.1: Industrial production

Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Industrial manufacturing
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of food, beverages, and tobacco products
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of textiles, clothing, leather, and leather products
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of wood and paper products, printing
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of coke and refined petroleum products
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of chemicals and chemical products
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of basic pharmaceutical products and pharmaceutical preparations
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of rubber and plastic products and other non-metallic mineral products
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of metals and metal structures excluding machinery and equipment
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of computers, electronic, and optical products
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of electrical equipment
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of machinery and equipment n.e.c.
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Production of transport equipment
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Other manufacturing, repair, and installation of machinery and equipment
Industrial production, Industrial production, NACE Rev. 2, seasonally adjusted, Supply of electricity, gas, steam, and air conditioning
Industrial production, Construction production, constant prices, seasonally unadjusted, Construction production
Industrial production, Construction production, constant prices, seasonally unadjusted, Domestic construction production
Industrial production, Construction production, constant prices, seasonally unadjusted, New construction, reconstruction, and modernization
Industrial production, Construction production, constant prices, seasonally unadjusted, Repairs and maintenance
Industrial production, Construction production, constant prices, seasonally unadjusted, Construction production abroad
Industrial production, Construction production, constant prices, seasonally unadjusted, New construction, reconstruction, and modernization - residential buildings
Industrial production, Construction production, constant prices, seasonally unadjusted, New construction, reconstruction, and modernization - non-residential buildings

Industrial production, Construction production, constant prices, seasonally unadjusted, New construction, reconstruction, and modernization - civil engineering works
Industrial production, Construction production, constant prices, seasonally unadjusted, Repairs and maintenance - residential buildings
Industrial production, Construction production, constant prices, seasonally unadjusted, Repairs and maintenance - non-residential buildings
Industrial production, Construction production, constant prices, seasonally unadjusted, Repairs and maintenance - civil engineering works
Industrial production, Construction production, constant prices, seasonally unadjusted, Other works - domestic
Industrial production, Construction production, constant prices, seasonally unadjusted, Domestic construction production - residential buildings
Industrial production, Construction production, constant prices, seasonally unadjusted, Domestic construction production - non-residential buildings
Industrial production, Construction production, constant prices, seasonally unadjusted, Domestic construction production - civil engineering works
Industrial production, Construction production, constant prices, seasonally unadjusted, Domestic construction production - buildings

Source: NBS macroeconomic database

Table A1.2: Prices

HICP, HICP - basic structure of NBS, seasonally adjusted, All items HICP
HICP, HICP - basic structure of NBS, seasonally adjusted, Net inflation excluding fuels
HICP, HICP - basic structure of NBS, seasonally adjusted, Energy
HICP, HICP - basic structure of NBS, seasonally adjusted, Food
HICP, HICP - basic structure of NBS, seasonally adjusted, Administered prices excluding energy prices
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Total industry
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Mining and quarrying
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Industrial production
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of food, beverages, and tobacco
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of textiles, clothing, leather, and leather products
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of wood and paper products, printing
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of coke and refined petroleum products
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of chemicals and chemical products
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of basic pharmaceutical products and pharmaceutical preparations
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of rubber and plastic products and other non-metallic mineral products
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of metals and metal structures excluding machinery and equipment

Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of computers, electronic, and optical products
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of electrical equipment
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of machinery and equipment n.e.c.
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of transport equipment
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Other manufacturing, repair, and installation of machinery and equipment
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Supply of electricity, gas, and cold air
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of electricity, transmission, and distribution
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Production of gas: distribution of gas fuels by pipeline
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Steam supply and distribution of cold air
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Water supply, cleaning, and waste water disposal, waste
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Collection, treatment, and supply of water
Industrial producer price index (ICPV), Domestic ICPV by sections and subsections NACE Rev. 2, Cleaning and waste water disposal
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Total industry
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Mining and quarrying
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Industrial production
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of food, beverages, and tobacco
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of textiles, clothing, leather, and leather products
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of wood and paper products, printing
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of coke and refined petroleum products
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of chemicals and chemical products
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of basic pharmaceutical products and pharmaceutical preparations
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of rubber and plastic products and other non-metallic mineral products
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of metals and metal structures excluding machinery and equipment
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of computers, electronic, and optical products

Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of electrical equipment
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of machinery and equipment n.e.c.
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Production of transport equipment
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Other manufacturing, repair, and installation of machinery and equipment
Industrial producer price index (ICPV), Export ICPV by sections and subsections NACE Rev. 2, Supply of electricity, gas, and cold air

Source: NBS macroeconomic database

Table A1.3: Revenue

Revenue, Total revenue, Total, constant prices, seasonally adjusted
Revenue, Total revenue, Total, domestic trade, constant prices, seasonally adjusted
Revenue, Total revenue, Total, selected sectors, constant prices, seasonally adjusted
Revenue, Domestic trade, constant prices, seasonally adjusted, Sale and repair of motor vehicles
Revenue, Domestic trade, constant prices, seasonally adjusted, Wholesale excluding motor vehicles
Revenue, Domestic trade, constant prices, seasonally adjusted, Retail excluding motor vehicles
Revenue, Domestic trade, constant prices, seasonally adjusted, Accommodation
Revenue, Domestic trade, constant prices, seasonally adjusted, Food and beverage service activities
Revenue, Selected sectors, constant prices, seasonally adjusted, Industry
Revenue, Selected sectors, constant prices, seasonally adjusted, Construction
Revenue, Selected sectors, constant prices, seasonally adjusted, Selected market services
Revenue, Selected sectors, constant prices, seasonally adjusted, Information and communication
Revenue, Selected sectors, constant prices, seasonally adjusted, Transportation and storage
Revenue, Revenue, MIG, constant prices, seasonally adjusted, Mining and quarrying
Revenue, Revenue, MIG, constant prices, seasonally adjusted, Mining and quarrying; industrial production
Revenue, Revenue, MIG, constant prices, seasonally adjusted, Industrial production
Revenue, Revenue, MIG, constant prices, seasonally adjusted, Investment goods
Revenue, Revenue, MIG, constant prices, seasonally adjusted, Consumer goods
Revenue, Revenue, MIG, constant prices, seasonally adjusted, Consumer goods excluding food, alcohol, and tobacco products

Source: NBS macroeconomic database

Table A1.4: Wages

Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, All sectors
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Industry total
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Industry total, of which: mining and quarrying

Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Industry total, of which: industrial production
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Industry total, of which: energy
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Industry total, of which: water and waste management
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Construction
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Sale and repair of motor vehicles
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Wholesale, excluding motor vehicles
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Retail, excluding motor vehicles
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Accommodation
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Restaurants and catering
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Transportation and storage
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Transportation and storage, of which: postal services and courier services
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Information and communication
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Selected market services
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Services total
Wages, compensations, Monthly average wages in selected sectors, seasonally adjusted, Trade total

Source: NBS macroeconomic database

Table A1.5: Employment

Job vacancies, Monthly job vacancies, UPSVAR, seasonally adjusted, Job vacancies
Job vacancies, Monthly job vacancies, UPSVAR, seasonally adjusted, Job vacancies - inflow
Job vacancies, Monthly job vacancies, UPSVAR, seasonally adjusted, Job vacancies - outflow
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, All sectors
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Industry total, of which: mining and quarrying
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Industry total, of which: energy
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Industry total, of which: water and waste management
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Construction
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Sale and repair of motor vehicles

Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Wholesale, excluding motor vehicles
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Retail, excluding motor vehicles
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Accommodation
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Restaurants and catering
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Transportation and storage
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Transportation and storage, of which: postal services and courier services
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Information and communication
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Selected market services
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Services total
Employment (hours, persons), Monthly employment in selected sectors, seasonally adjusted, Trade total
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Mining and quarrying
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Manufacturing of food, beverages and tobacco products
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Manufacturing of wood and paper products, printing
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Manufacturing of coke and refined petroleum products
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Manufacturing of chemicals and chemical products
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Manufacturing of basic pharmaceutical products and pharmaceutical preparations
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Manufacturing of rubber and plastic products and other non-metallic mineral products
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Manufacturing of computer, electronic and optical products
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Manufacturing of electrical equipment
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Manufacturing of machinery and equipment not elsewhere classified
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Manufacturing of transport equipment
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Energy
Employment (hours, persons), Monthly employment in industry, seasonally adjusted, Water and waste management
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Number of unemployed persons
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Number of available unemployed persons

Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Inflow of job seekers
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Inflow of job seekers, graduates
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Outflow of job seekers
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Outflow of job seekers, placed in the labor market
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Outflow of job seekers, excluded due to non-cooperation
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Outflow of job seekers, others
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Non-available job seekers
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Temporary incapacity for work and OČR
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Unemployment rate as a percentage of total applicants
Unemployment, NAIRU, Unemployment, UPSVAR, seasonally adjusted, Registered unemployment rate

Source: NBS macroeconomic database

Table A1.6: Foreign trade

Foreign Trade, Foreign Trade-Export, Total Export, seasonally adjusted
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Countries outside the eurozone
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Countries in the eurozone
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - EU 28
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - V3 (Poland, Hungary, Czechia)
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Germany
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Czechia
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - France
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Poland
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Austria
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Italy
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Hungary
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Russia
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - United Kingdom of Great Britain and Northern Ireland
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - China
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Netherlands
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Spain
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Sweden
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Romania
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - USA
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Belgium
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Turkey

Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Switzerland
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Korea
Foreign Trade, Foreign Trade-Export, Territorial structure of exports - Unspecified
Foreign Trade, Foreign Trade-Export, Commodity structure of exports - final consumption
Foreign Trade, Foreign Trade-Export, Commodity structure of exports - raw materials
Foreign Trade, Foreign Trade-Export, Commodity structure of exports - intermediates
Foreign Trade, Foreign Trade-Export, Commodity structure of exports - machinery, apparatus, equipment
Foreign Trade, Foreign Trade-Import, Total Import, seasonally adjusted
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Countries outside the Eurozone
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Countries in the Eurozone
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - EU 28
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - V3 (Poland, Hungary, Czechia)
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Germany
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Czechia
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - France
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Poland
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Austria
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Italy
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Hungary
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Russia
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - United Kingdom of Great Britain and Northern Ireland
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - China
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Netherlands
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Spain
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Sweden
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Romania
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - USA
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Belgium
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Turkey
Foreign Trade, Foreign Trade-Import, Territorial structure of imports - Switzerland

Source: NBS macroeconomic database

Table A1.7: Current account

Current account (BPM6), Current account (BPM6), seasonally unadjusted, Current account - assets
Current account (BPM6), Current account (BPM6), seasonally unadjusted, Current account - liabilities
Current account (BPM6), Current account (BPM6), seasonally unadjusted, Goods - credit
Current account (BPM6), Current account (BPM6), seasonally unadjusted, Goods - debit
Current account (BPM6), Current account (BPM6), seasonally unadjusted, Services - credit
Current account (BPM6), Current account (BPM6), seasonally unadjusted, Services - debit
Current account (BPM6), Current account (BPM6), seasonally unadjusted, Primary income - credit

Current account (BPM6), Current account (BPM6), seasonally unadjusted, Primary income - debit
Current account (BPM6), Current account (BPM6), seasonally unadjusted, Secondary income - credit
Current account (BPM6), Current account (BPM6), seasonally unadjusted, Secondary income - debit
Current account (BPM6), Current account (BPM6), seasonally unadjusted, Capital account - debit

Source: NBS macroeconomic database

Table A1.8: Exchange rates

Exchange rates, Bilateral exchange rates, Bilateral exchange rate CZK/EUR
Exchange rates, Bilateral exchange rates, Bilateral exchange rate HUF/EUR
Exchange rates, Bilateral exchange rates, Bilateral exchange rate PLN/EUR
Exchange rates, Bilateral exchange rates, Bilateral exchange rate USD/EUR
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Germany
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Czech Republic
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Italy
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), France
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Austria
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Poland
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Hungary
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), United Kingdom
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), United States
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Netherlands
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Belgium
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Spain
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Russian Federation
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), China
Exchange rates, Bilateral real exchange rates based on CPI (15 main trading partners of the Slovak Republic), Republic of Korea
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Germany
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Czech Republic

Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Italy
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), France
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Austria
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Poland
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Hungary
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), United Kingdom
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), United States
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Netherlands
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Belgium
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Spain
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Russian Federation
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), China
Exchange rates, Bilateral real exchange rates based on PPI (15 main trading partners of the Slovak Republic), Republic of Korea
Exchange rates, Effective exchange rates - monthly (15 main trading partners of the Slovak Republic), Nominal effective exchange rate
Exchange rates, Effective exchange rates - monthly (15 main trading partners of the Slovak Republic), Real effective exchange rate based on CPI
Exchange rates, Effective exchange rates - monthly (15 main trading partners of the Slovak Republic), Real effective exchange rate based on PPI

Source: NBS macroeconomic database

Table A1.9: Indicators

The Economic Sentiment Indicator, The Economic Sentiment Indicator, Long-term average, Economic Sentiment Indicator
Economic Sentiment Indicator, Industrial Confidence Indicator, Seasonally adjusted, Industry
Economic Sentiment Indicator, Industrial Confidence Indicator, Seasonally adjusted, Industrial Production Trend (last 3 months)
Economic Sentiment Indicator, Industrial Confidence Indicator, Seasonally adjusted, Current Level of Overall Demand for Production
Economic Sentiment Indicator, Industrial Confidence Indicator, Seasonally adjusted, Current Level of Demand for Production Abroad
Economic Sentiment Indicator, Industrial Confidence Indicator, Seasonally adjusted, Current Stocks of Finished Products
Economic Sentiment Indicator, Industrial Confidence Indicator, Seasonally adjusted, Expected Industrial Production (next 3 months)

Economic Sentiment Indicator, Industrial Confidence Indicator, Seasonally adjusted, Expected Product Prices (next 3 months)
Economic Sentiment Indicator, Industrial Confidence Indicator, Seasonally adjusted, Expected Number of Employees (next 3 months)
Economic Sentiment Indicator, Services Confidence Indicator, Seasonally adjusted, Services
Economic Sentiment Indicator, Services Confidence Indicator, Seasonally adjusted, Business Situation Trend (last 3 months)
Economic Sentiment Indicator, Services Confidence Indicator, Seasonally adjusted, Demand for Services Development (last 3 months)
Economic Sentiment Indicator, Services Confidence Indicator, Seasonally adjusted, Expected Demand for Services (next 3 months)
Economic Sentiment Indicator, Services Confidence Indicator, Seasonally adjusted, Number of Employees (last 3 months)
Economic Sentiment Indicator, Services Confidence Indicator, Seasonally adjusted, Expected Number of Employees (next 3 months)
Economic Sentiment Indicator, Services Confidence Indicator, Seasonally adjusted, Expected Service Prices (next 3 months)
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Consumers
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Expected Household Financial Situation (next 12 months)
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Past Household Financial Situation (last 12 months)
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Past Economic Situation of Slovakia (last 12 months)
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Expected Economic Situation Development in Slovakia (next 12 months)
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Past Inflation (last 12 months)
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Expected Inflation (next 12 months)
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Expected Unemployment (next 12 months)
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Conditions for Major Purchases in Slovakia
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Household Plans for Major Purchases (last 12 months)
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Saving Conditions in Slovakia
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Expected Household Savings Development (next 12 months)
Economic Sentiment Indicator, Consumer Confidence Indicator, Seasonally adjusted, Opinion on Household Financial Situation (last 12 months)
Economic Sentiment Indicator, Retail Confidence Indicator, Seasonally adjusted, Retail
Economic Sentiment Indicator, Retail Confidence Indicator, Seasonally adjusted, Business Situation Trend (last 3 months)
Economic Sentiment Indicator, Retail Confidence Indicator, Seasonally adjusted, Current Inventory Levels

Economic Sentiment Indicator, Retail Confidence Indicator, Seasonally adjusted, Expected Supplier Requirements (next 3 months)
Economic Sentiment Indicator, Retail Confidence Indicator, Seasonally adjusted, Expected Business Situation (next 3 months)
Economic Sentiment Indicator, Retail Confidence Indicator, Seasonally adjusted, Expected Number of Employees (next 3 months)
Economic Sentiment Indicator, Retail Confidence Indicator, Seasonally adjusted, Expected Product Prices (next 3 months)
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Construction
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Construction Activity Trend (last 3 months)
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Factors Limiting Construction Production (%): none
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Factors Limiting Construction Production (%): insufficient demand
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Factors Limiting Construction Production (%): weather conditions
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Factors Limiting Construction Production (%): lack of employees
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Factors Limiting Construction Production (%): lack of material and/or equipment
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Factors Limiting Construction Production (%): other
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Factors Limiting Construction Production (%): financial constraints
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Current Level of Demand for Construction Production
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Expected Number of Employees (next 3 months)
Economic Sentiment Indicator, Construction Confidence Indicator, Seasonally adjusted, Expected Construction Production Prices (next 3 months)
Economic Sentiment Indicator, Expected Employment Development, Expected Employment Development Together
Economic Sentiment Indicator, Industrial Confidence Indicator, Contributions, Seasonally adjusted, Industry
Economic Sentiment Indicator, Industrial Confidence Indicator, Contributions, Seasonally adjusted, Current Level of Overall Demand for Production
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Food Production
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Beverage Production
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Textile Production
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Clothing Production
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Leather and Leather Goods Production

Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Wood Processing and Wood and Cork Products Manufacturing except Furniture
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Paper and Paper Products Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Printing and Reproduction of Recorded Media
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Chemicals and Chemical Products Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Pharmaceutical Products and Preparations Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Rubber and Plastic Products Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Other Non-metallic Mineral Products Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Metals Processing and Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Fabricated Metal Products Manufacturing except Machinery and Equipment
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Computer, Electronic and Optical Products Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Electrical Equipment Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Machinery and Equipment Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Motor Vehicles, Trailers and Semi-Trailers Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Other Transport Equipment Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Furniture Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Other Manufacturing
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Electricity, Gas, Steam and Air Conditioning Supply
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Water Supply, Sewerage, Waste Management and Remediation Activities
Economic Sentiment Indicator, Industrial, Sectors, Industrial Production Trend (last 3 months), Seasonally adjusted, Mining and Quarrying

Source: NBS macroeconomic database