Prague Economic Papers
2025, Volume 34, Issue 4
https://doi.org/10.18267/j.pep.898

VŠE / PRAGUE UNIVERSITY OF ECONOMICS AND BUSINESS

# Do Machine Learning Techniques Outperform Autoregressive Distributed Lag Models in Inflation Forecasting?

Bogdan Oancea, Mihaela Simionescu, Richard Pospisil

Bogdan Oancea, Department of Applied Economics and Quantitative Analysis, University of Bucharest, Bd. Regina Elisabeta no. 4–12, Sector 3, 030108 Bucharest, Romania, National Institute of Research and Development for Biological Sciences, Splaiul Independenței 296, 060031 Bucharest, Romania, E-mail: bogdan.oancea@faa.unibuc.ro, bogdan.oancea@incdsb.ro

Mihaela Simionescu, Department of Applied Economics and Quantitative Analysis, University of Bucharest, Bd. Regina Elisabeta no. 4–12, Sector 3, 030108 Bucharest, Romania, Institute for Economic Forecasting, Romanian Academy, Bucharest, Romania, Academy of Romanian Scientists, Ilfov 3, 050044 Bucharest, Romania, E-mail: mihaela.simionescu@faa.unibuc.ro

Richard Pospisil (Corresponding author), Tomas Bata University in Zlin, Faculty of Logistics & Crisis Management, Zlin, Czech Republic, E-mail: r1pospisil@utb.cz

**Abstract**:

Following the COVID-19 pandemic, Romania and other Central and Eastern European (CEE) countries faced some of the highest inflation rates in the European Union, creating a pressing need for accurate short-term forecasts to guide monetary policy. This study compares modern machine learning (ML) methods - Long Short-Term Memory (LSTM) neural networks, Random Forests (RF) and Support Vector Regression (SVR) - with traditional Autoregressive Distributed Lag (ARDL) models in forecasting Harmonised Index of Consumer Prices. Using quarterly data for Romania (2006Q1–2023Q4) and monthly data for nine CEE economies (2006M1–2025M3), we incorporate unemployment and sentiment indicators derived from the Romanian Central Bank reports and the European Commission's Economic Sentiment Indicator (ESI). We further evaluate model performance through simulation experiments that include high persistence, moving-average non-invertibility, nonlinear regimes, and structural breaks. Across both empirical and LSTM and SVR models - they frequently deliver lower forecast errors than ARDL, with LSTM achieving up to 53% reductions in mean squared error relative to naïve benchmarks. However, ARDL remains competitive when sentiment indices

are the main predictor. These findings highlight that while advanced ML models can capture nonlinear dynamics and regime changes, traditional econometric tools still provide valuable robustness, particularly in sentiment-driven contexts. Overall, integrating ML, econometric approaches, and sentiment analysis offers a more reliable toolkit for short-horizon inflation forecasting under economic uncertainty.

**Keywords**: Inflation, Long Short-Term Memory neural networks, Random Forests, Support Vector Regression, Autoregressive Distributed Lag models.

**JEL classification:** C51, C53, E31

## 1. Introduction

This paper investigates whether modern machine-learning (ML) methods, specifically Long Short-Term Memory neural networks (LSTM), Random Forests (RF), and Support Vector Regression (SVR), can produce more accurate short-term inflation forecasts than traditional Autoregressive Distributed Lag (ARDL) models, especially during periods of high volatility and policy uncertainty. Focusing on Romania and the Central and Eastern European (CEE) economies after the COVID-19 pandemic, we developed and compared forecasting models that incorporate macroeconomic variables such as Harmonised Indices of Consumer Prices (HICP) as a proxy for inflation and unemployment, along with sentiment indicators derived from central-bank reports and the European Commission's Economic Sentiment Indicator (ESI). Our results highlight the situations in which each approach performs best, provide practical advice for policymakers on model choice, and emphasize the importance of combining traditional econometrics with advanced ML tools to improve real-time policy decisions.

Inflation in Romania has been a major concern in 2023, reaching double-digit levels due to rising energy and food prices, supply chain disruptions caused by the pandemic, and increased government spending. In the first quarter of 2024, Romania recorded the highest inflation in the EU at 6.7% in March 2024. This economic situation underscores the need for accurate forecasts. Recent macro models for Romania show that fiscal transmission is delayed and history-dependent, with "memory" in income dynamics affecting short-run adjustment paths. This strengthens the case for short-horizon inflation forecasts that accommodate distributed lags and nonlinearity (Panzaru, Belea, and Jianu, 2025).

Inflation forecasting is crucial for policymakers, economists, and financial advisors. Accurate predictions of inflation can greatly influence monetary policy and economic decisions. When forecasting economic time series data, choosing the best algorithmic approach is essential for obtaining accurate predictions and strong performance. Two common methodologies

in this field are traditional econometric methods like autoregressive integrated moving average (ARIMA) models and machine learning (ML) techniques.

Econometric techniques have been widely used in analyzing and forecasting time series data because they effectively capture time and seasonal patterns. Additionally, ML techniques provide a more adaptable framework for representing interconnected dynamics and nonlinear relationships in sequential data. The choice between these models depends on various factors, such as the features of the time series, the complexity of the temporal patterns, and the primary goals of the forecasting task. While econometric methods often assume certain conditions like stationarity, machine learning models do not offer greater flexibility in explaining connected dynamics and nonlinear relationships in series.

This research assesses the effectiveness of three machine learning techniques - RF, LSTM, and SVR - for predicting HICP during 2023Q1 to 2023Q4, based on historical data from 2006 to 2022 for Romania. The results are compared with traditional econometric methods, specifically autoregressive distributed lag models (ARDL), which address endogeneity issues related to the connection between inflation and unemployment in Romania. Romania was selected because it experienced one of the highest rates of inflation in the European Union during the post-pandemic period. Additionally, to evaluate the broader applicability of the findings, the analysis was extended to all CEE countries using monthly seasonally adjusted data from 2006M1 to 2024M12, with a forecast horizon of three months: 2025M1 to 2025M3. The core research question examines whether machine learning methods can outperform econometric models in short-term inflation forecasting within an economically vulnerable context, characterized by high. Inflation is modeled based on unemployment rates and a sentiment index derived from the National Bank's quarterly reports for Romania and the ESI for other CEE countries. While the interest rate might better explain inflation since central banks use it to control prices, it is not included here because the National Bank of Romania promotes a self-regulating market and typically maintains stable interest rates, with only significant adjustments in extreme situations.

On the other hand, according to the Phillips curve theory, there is an indirect relationship between inflation and unemployment. When unemployment rates are low, inflation tends to be high. The theory suggests that in a strong economy with low unemployment, workers have more purchasing power and can demand higher wages. This can lead to increased prices, which result in inflation. This describes the current situation of the Romanian economy. Ahead of the 2024/2025 elections, the government has announced a salary increase, including an increase in the minimum wage. Additionally, recent employee protests driven by social tensions

in recent years have put pressure on the labor market and linked it more closely with inflation.

The novelty of this study lies in constructing the most appropriate models and identifying the best forecasting method for a short-term horizon characterized by high inflation and for a country that gradually increased interest rates to address the phenomenon.

Additionally, this study investigates the ability of sentiment techniques based on natural language processing (NLP) to analyze large volumes of text provided by official experts for improving inflation forecasts within specific spatial and temporal frameworks. Economic sentiment, which reflects the overall mood of the economy and future expectations, is a key indicator in macroeconomics. It is relevant across various sectors - such as trade, services, and industry - as well as for managers and investors in financial markets, including commodities and stocks. In mainstream economic thought, the aggregate confidence indicator, often called the Economic Sentiment Indicator (ESI) or Basic Index, combines both business and consumer confidence into a single measure.

The sentiment analysis remains an underexplored research area with real potential to enhance prediction accuracy (Dang et al., 2025; Eugster and Uhl, 2024), and the use of NLP falls under the category of modern artificial intelligence techniques. Additionally, this pioneering research conducted in Romania uses complex machine learning techniques to forecast inflation over short-term horizons and during periods of high inflation that could not last long due to central bank interventions.

The inflationary period affecting all developed countries after the Covid-19 pandemic, especially many EU economies, calls for new and advanced methods of inflation forecasting. Achieving a high level of price stability is the main goal of central banks (Stock and Watson, 1999). As a result, inflation forecasting is a key starting point for inflation targeting in central bank monetary policy. Besides central banks, commercial banks also forecast inflation to estimate the prices of their resources and assets in the short and medium term (Hong et al., 2024). Additionally, inflation forecasting interests' academia and supranational institutions such as the International Monetary Fund, the World Bank, and others. Nominal output data and forecast comments are primary sources for rational decision-making by consumers, households, and companies (Liu et al., 2024).

Most studies and sources on inflation forecasting primarily predict inflation trends in isolation without thoroughly examining the interrelationships among economic variables. Many forecasts also focus on longer economic cycles that could be significantly influenced by common monetary and fiscal policy tools. Additionally, non-economic factors such as pandemics, international events, wars, and others are increasingly recognized as important (Groen et al.,

2012). This complexity makes inflation forecasting more challenging and requires reliance on shorter observation periods (Fulton and Hubrich, 2021). Furthermore, the results and recommendations from inflation forecasts tend to have shorter time horizons and lower informational value (Eugster and Uhl, 2024).

A correct estimate of the future inflation rate is extremely important, even dominant, for the economy. The inflation rate is directly related to the level of short-term interest rates, and thus directly affects the prices of assets, deposits, mortgages, debts, and more (Grothe and Meyer, 2015). Broadly speaking, it influences the country's economic performance, economic growth, tax collection, and the real value of wages and pensions.

Most of the current inflation forecasting models are well documented in many papers (Stock and Watson, 2003; Zhu et al., 2024), but they are not very suitable for today's economic reality, which is burdened with high uncertainty. The research gap involves developing inflation forecasting methods using ML models, comparing their performances, and creating a method that achieves the highest possible accuracy (Medeiros et al., 2021). The presence of high inflation rates across all advanced economies highlights the need for a new approach to inflation forecasting, one that incorporates models considering various economic and potential new non-economic indicators (Benett and Owyang, 2022; Faust and Wright, 2013).

The paper follows the traditional structure by including a literature review, methodology, empirical results, and conclusions. All these sections contribute to the novelty of this research.

## 2. Literature review

In the era of inflation targeting since the 1990s, short-term inflation forecasts have become a vital input for central banks and economic policy (Rygh, 2025). This is especially true in Europe, where price stability is formally defined in terms of the Harmonised Index of Consumer Prices (HICP) and where continuously monitoring and predicting inflation is deemed "indispensable" for monetary strategy (Vicente, 2005). A rich literature has accordingly developed a range of forecasting approaches - from simple time-series extrapolations to structural economic models - to anticipate near-term inflation dynamics. These approaches and findings in Europe are often compared with those from other regions to distil general insights into inflation forecast performance.

Conventional econometric methods have long underpinned short-term inflation forecasting. Univariate time-series models like ARIMA (and seasonal ARIMA) often provide robust benchmarks, sometimes matching or outperforming more complex specifications (Rygh, 2025). Multivariate models, such as Vector Autoregressions (VARs) that incorporate broader econom-

ic information, and structural Phillips curve models linking inflation to measures of slack (unemployment or output gaps), have been extensively applied. However, many studies find that these elaborate models do not consistently outperform simpler ones out of sample. For example, traditional Phillips-curve forecasts frequently failed to beat naive predictions based on past inflation. More broadly, decades of research have documented the challenge of improving upon basic extrapolative benchmarks (random walks or autoregressive trends) in forecasting inflation (Beck and Wolf, 2025). This has motivated ongoing refinements of classical models (e.g. Bayesian VARs or factor models) as researchers seek incremental gains in predictive accuracy.

Short-term inflation forecasting in Europe - typically expressed in terms of HICP remains challenging, with simple extrapolative benchmarks often difficult to beat. A comprehensive ECB assessment of Phillips-curve specifications from 1994 to 2018 finds that while some variants do outperform a univariate benchmark at times, gains are modest, and model performance is episodic. Allowing for a time-varying inflation trend and carefully chosen slack measures helps, whereas adding external drivers generally does not improve out-of-sample accuracy. These results underscore why forecasters combine tools and emphasize short horizons (3–12 months) where policy relevance is highest (Bańbura and Bobeica, 2020).

Within traditional econometric approaches, a key European question is whether to forecast aggregate HICP directly or to model and aggregate components. Recent ECB work comparing direct ("aggregate") versus bottom-up (component) models reveal small average differences overall, but notable short-horizon episodes where disaggregation proves beneficial. For example, at h = 1month, relative RMSEs (component vs. aggregate; values < 1 favour component) were 0.53 for Italy and 0.38 for the Netherlands (i.e., component better), while by h = 12 months, relative RMSEs were near unity across the big five economies (e.g., 0.94–1.01). Density (probabilistic) scores often favour the aggregate route at most horizons, with statistically significant but small gains (Chalmovianský et al., 2020).

The past decade has seen growing interest in ML methods for inflation forecasting, both in Europe and internationally. Techniques such as SVR (Zhang and Li, 2012), tree-based ensembles (random forests and boosting), and deep neural networks (especially LSTMs) have been applied to capture potential non-linearities and complex data patterns that traditional models might miss (Beck and Wolf, 2025). In some cases, ML models have delivered improved accuracy: for example, studies using data-rich environments report that ensemble methods like random forests can outperform standard autoregressive or random-walk benchmarks in predicting inflation. Nevertheless, the evidence is mixed. Advanced ML approaches do not consistently dominate well-tuned econometric models, especially at short horizons. Recent research

finds that LSTM-based forecasts for both U.S. CPI and euro-area inflation achieved accuracy roughly comparable to simpler models (such as SARIMA or regularized regressions), offering at best marginal gains (Almosov and Andresen, 2023). Similarly, while SVR has shown promise in specific instances (outperforming neural networks or OLS in one CPI forecasting study), its advantages have not proven universal. Given the tendency of complex ML models to overfit and their relative lack of transparency, a notable trend is the rise of hybrid approaches - combining machine learning with traditional techniques. For example, researchers have explored blending LSTM networks with feature selection or linear components and using regularized regression (like LASSO) alongside non-linear learners, to improve interpretability and robustness (Rygh, 2025; Paranhos, 2025).

ML methods have also been incorporated recently into the European toolbox. An ECB Quantile Regression Forest (QRF) built on 60 Phillips-curve-inspired predictors delivered point-forecast RMSEs for euro-area headline/core inflation that are broadly comparable to institutional benchmarks at short horizons: for headline, RMSEs were 0.58 (QRF) vs. 0.47 (BMPE) at h = 3, 0.92 vs. 0.94 at h = 6; for core, 0.21 vs. 0.22 at h = 3, 0.36 vs. 0.38 at h = 6. Over the sample, the QRF tended to perform better for core than headline and showed competitive density forecasts against both linear BVAR combinations and survey densities. These results suggest non-linear methods can match state-of-the-art linear systems for short-run HICP forecasts, especially for core (Lenza et al., 2023).

The factors influencing short-term performance in Europe align with macroeconomic theory and practice. In the QRF, SHAP analysis emphasizes short-term interest rates (Euribor 3-month), survey-based price expectations (consumer and industry), unemployment, government bond yields, building permits (as a measure of real activity), producer price indices, and negotiated wages as the top contributors at six-month horizons. In Phillips-curve models, time-varying inflation trends, filter-based output gaps, and various measures of labor market slack are important; meanwhile, adding external variables (such as terms of trade and commodity prices) did not consistently improve forecast accuracy in euro-area studies, likely due to their poor forecastability (Lenza et al., 2023).

Comparative evidence beyond Europe shows similar trends. In the United States, data-rich ML models, especially RF, often reduce RMSE compared to random-walk/AR benchmarks across different horizons, supporting their use alongside traditional models. Recent research on “hedged random forests” reports RMSE and MAE ratios below 1 compared to standard forests across multiple inflation measures and horizons, with many Diebold–Mariano tests indicating significant improvements. Studies using disaggregated data in emerging markets like

Brazil show RMSE reductions of approximately 20-27% at several horizons. These findings, along with European results, suggest that while linear models are still effective, ML methods that can identify mild non-linearities and interactions improve short-term HICP/CPI forecasts - especially for core inflation (Medeiros et al., 2021; Beck and Wolf, 2025; Boaretto and Medeiros, 2023).

In summary, while most studies used econometric models to predict inflation, including popular examples like random walk, Dynamic Stochastic General Equilibrium (DSGE) models, various extensions of Vector Autoregressions (VAR), and other econometric model types (Petropoulos et al., 2022), only a few recent papers have explored modern forecasting methods such as ML techniques. As expected, these new methods aim to improve forecasting by focusing on accuracy as the main measure of forecast performance. However, the potential of ML methods to yield more accurate predictions than traditional econometric models remain promising, and empirical evaluation is needed to verify this. Therefore, in this section, we introduce the ML techniques used in the study: RF, LSTM, and SVR.

RF is a supervised ensemble learning method that constructs multiple decision trees and combines their predictions (Breiman, 2001). They are known for their robustness and strong performance in both classification and regression tasks. This algorithm trains numerous decision trees and then merges their predictions to make a final decision. This approach enhances predictive accuracy and helps prevent overfitting.

Each decision tree is created using a random subset of the training data and a random subset of features, which helps reduce overfitting and improve generalization. The final prediction is made by a majority vote (for classification) or by averaging (for regression) the outputs of the individual trees.

The ability of RF to handle nonlinear relationships and noisy inputs makes it particularly suitable for a wide range of real-world applications, including economics, finance, bioinformatics, and remote sensing (Svoboda et al., 2022; Wang and Liu, 2025; Zhou et al., 2025; Raza et al., 2025). The algorithm's parallelizability and scalability enable efficient computation on large datasets, making it feasible for applications even in domains with extensive data resources, such as genomics and climate modeling (Cutler et al., 2007; Strobil et al., 2009; Kumuda and Panigrahy, 2025; Sevgin, 2025). However, RF is not immune to certain weaknesses and considerations. While RF generally delivers competitive performance across a range of tasks, its predictive accuracy may plateau or diminish on extremely imbalanced datasets or those characterized by highly correlated features. Careful tuning of hyperparameters (number of trees and maximum tree depth) is crucial to maximize model performance and prevent overfitting (Cutler et al., 2007; Biau and Scornet, 2016).

SVR extends the principles of Support Vector Machines to regression problems, aiming to find a function that approximates the data within a certain margin while keeping the model simple (Drucker et al., 1997). Unlike traditional regression methods that minimize prediction errors directly, SVR focuses on minimizing deviations outside a specified margin, called the epsilon-insensitive tube. This margin defines a range where prediction errors are acceptable, and deviations beyond them are penalized proportionally to their size. SVR achieves this by mapping input data into a higher-dimensional space using a kernel function and considering a hyperplane that maximizes the margin while minimizing errors. By adding a regularization term to the objective function, SVR balances maximizing margin width and minimizing prediction errors, resulting in a robust and generalizable regression model (Drucker et al., 1997; Smola and Scholkopf, 2004).

A key feature of SVR is its ability to handle non-linear relationships between input variables and target outputs through the use of kernel functions. By implicitly mapping data into a higher-dimensional feature space, SVR can identify more complex patterns and relationships that may be hidden in the original input space. This flexibility allows SVR to perform well across various regression tasks, including time series forecasting, financial modeling, and bioinformatics (Ngwaba, 2025; Lin et al., 2021). Additionally, SVR's principle of structural risk minimization, which focuses on maximizing the margin while controlling model complexity, helps reduce the risk of overfitting and improves its ability to generalize to new data (Smola and Scholkopf, 2004; Vapnik, 1995).

Despite its versatility and effectiveness, SVR also has some limitations. The choice of a specific kernel function and its associated parameters can influence SVR's performance, requiring careful tuning to achieve the best results. Additionally, the computational cost of SVR increases with both the size of the training dataset and the dimensionality of the feature space, often making it less suitable for large-scale problems compared to simpler regression methods. Furthermore, the interpretability of SVR predictions can be difficult in high-dimensional spaces, which limits its use in applications where clear model insights are essential. Nonetheless, with proper parameter selection and regularization, SVR remains a valuable tool for regression tasks that need robustness to non-linearities and flexibility in model representation (Smola and Scholkpopf, 2004; Ito and Nakano, 2003; Basak et al., 2007).

LSTM networks, a type of recurrent neural networks (RNNs) specifically designed to address the vanishing gradient problem and to handle long-range connections in sequential data, have become important tools for time series prediction tasks (Hochreiter and Schmidhuber, 1997). By leveraging their ability to incorporate long-range links and manage sequential data,

LSTM-based methods have achieved significant success across various fields, including economics, finance, healthcare, and climate forecasting (Park and Yang, 2022; Saadauui and Rabbouch, 2024; Cissoko et al., 2025; Nitesh et al., 2023).

Basic RNNs have difficulty learning from long sequences because of vanishing gradients. In contrast, LSTMs use a more sophisticated gating system that allows them to selectively add or remove information based on input signals. This system, which includes input, forget, and output gates, helps LSTMs learn and hold onto long-term dependencies while reducing the impact of vanishing gradients (Hochreiter and Schnidhuber, 1997; Graves et al., 2009).

At the core of LSTM networks are memory cells that act as information carriers, equipped with self-loop connections to support the circulation of information over time. Each memory cell maintains a cell state, functioning like a conveyor belt along the sequence, and is controlled by three types of gates: the input, forget, and output gates. The input gate regulates the flow of new information into the cell state; the forget gate decides whether to retain or discard existing information; and the output gate filters the information passed to the next step or output layer. By combining these gates with non-linear activation functions, LSTMs can preserve and transmit important information across long sequences, enabling the learning of complex temporal patterns and relationships (Graves et al., 2009; Gers et al., 2000).

The main advantage of LSTMs is their ability to model and predict sequences with variable lengths and temporal dynamics. Unlike traditional fixed-length window methods, which require predefined segment sizes and often struggle with capturing long-range dependencies, LSTMs can dynamically adjust their memory cells to handle sequence tasks such as speech recognition, where utterance durations vary, and natural language processing, where sentence structures and lengths differ. Additionally, LSTMs can incorporate contextual information and semantic relationships within sequences, leading to coherent and relevant predictions in applications like language translation, sentiment analysis, and time series forecasting (Sutskever and Vinyals, 2014; Bahdanau et al., 2015).

LSTMs also face limitations and challenges. Training LSTM networks requires many calculations, especially for large-scale data sets and complex architectures with many parameters. Additionally, LSTMs may encounter overfitting, particularly with small data sets or when noisy inputs are present. Overfitting can be mitigated through dropout and weight decay, which enhance generalization. However, with careful design of architecture, regularization, and training, LSTMs remain a versatile and strong tool for modeling sequential data and capturing complex temporal connections in various tasks (Salton and Kelleher, 2019; Mienye et al., 2024; Srivastava et al., 2014; Zaremba et al., 2014).

All three methods are used to produce short-term inflation forecasts for Romania and other CEE countries, and these predictions are compared to those from ARDL models. The findings could guide future forecasts in similar economic settings.

Specifically, we intended to check the following hypotheses:

H1. (Short Horizon): ML models with macro covariates (unemployment rate, sentiment index/ESI) reduce RMSE/MAPE compared to ARDL baselines for HICP short-horizon forecasts (1-3 steps).

H2. (Sentiment-only): When sentiment is the sole extra covariate, ARDL remains competitive with ML.

H3. (Data-generating conditions): Under MA non-invertibility or regime shifts, ML methods (LSTM and/or SVR) show larger gains.

## 3. Materials and Methods

A short description of the datasets and methods is given below.

The quarterly data used for Romania cover the period 2006Q1-2023Q4 (forecasts made for 2023Q1-2023Q4) and refer to the following indicators:

Q1. Harmonized Index of Consumer Prices (HICP) as a proxy for inflation. We used the quarterly HICP data provided by Eurostat.

Q2. Sentiment index. Derived from textual data sources (abstracts of Inflation Reports released by the National Bank of Romania), calculated using NLP techniques in IntelliDocker to quantify economic sentiment.

Q3. Unemployment rate. Collected from official statistics sources (Romanian National Institute of Statistics), it provides a vital indicator of economic health.

The monthly data for the CEE countries in the sample (Romania, Slovakia, Slovenia, Lithuania, Latvia, Estonia, Poland, Hungary, Czechia) cover the period from 2006M1 to 2025M03 (with forecasts for 2025M1 to 2025M03) and pertain to the following indicators:

M1. Harmonized Index of Consumer Prices (HICP) with monthly frequency provided by Eurostat.

M2. Unemployment rate provided by Eurostat.

M3. European Commission's Economic Sentiment Indicator (ESI). It is provided by Eurostat for all CEE countries and reflects confidence in the economic outlook of consumers and businesses.

Thus, we forecast HICP index level (2015 = 100), and we report short-horizon rates only in month-over-month (MoM) or quarter-over-quarter (QoQ) terms: $\pi^{MoM}t = 100x\left(\frac{HICP_t}{HICPt-1} - 1\right)$, $\pi^{QoQ}t = 100x\left(\frac{HICP_t}{HICPt-1} - 1\right)$. We do not report year-over-year rates.

First, we will provide some technical details on calculating the sentiment index for Romania. Using document classification and RNN, sentiment analysis produces indexes based on the sentiment (positive, negative, neutral) associated with words in a text. This involves scoring words as positive (+1), negative (-1), or neutral (0), summing these scores, and normalizing the results to a scale between 0 and 1. Although Clements and Reade (2020) also used a system of increments and decrements, our methodology introduces automated sentiment index computation via IntelliDocker and expands the analysis to cover the entire Romanian dictionary, unlike their 3,000-word limit. The evolution of the sentiment index (together with the HICP and unemployment rate) is shown in the Appendix Figure A3, showing successive increases and decreases. When we tested the generalizability of the results to CEE countries, we used the standard ESI calculated by Eurostat.

Second, we will provide some details on the similar and different characteristics of the sentiment index and ESI used in the models with quarterly and monthly data, respectively.

Both the ESI and the Romanian Inflation-Report Sentiment Index act as "early-warning" systems, signaling changes in economic trends before they appear in official data. However, they differ significantly in their construction.

The Romanian Inflation-Report Sentiment Index is created using machine-learning-based sentiment analysis on the unstructured text of the National Bank's quarterly inflation reports. Its goal is to measure the tone of central bank communications, mainly reflecting expert opinions about future inflation. Since similar narrative reports are not available from other CEE central banks, we use the ESI as our second measure.

The ESI, compiled by the European Commission's Directorate-General for Economic and Financial Affairs (DG ECFIN), is a composite index based on structured monthly surveys of firms (in industry, services, retail, construction) and households. It reflects broader business and consumer expectations about the current and future economic situation—and is designed to track overall economic confidence rather than just inflation. The ESI is constructed in two main steps. First, for each survey component, country-level balances of positive versus negative responses are aggregated using moving-average country weights based on sectoral value-added shares. Second, these country-aggregated component series are combined using fixed survey weights to produce the overall ESI.

In summary, the Romanian index assesses expert sentiment in central-bank statements, while the ESI gauges perceived sentiment among a broad range of economic participants. Together, they provide complementary early signals of inflationary pressures and overall economic confidence.

The Appendix (tables A1-A4) presents descriptive statistics for the main variables and shows that only the unemployment variable follows a normal distribution.

The analysis starts with the econometric approach based on ARDL models with three specifications:

$$\mathrm{HICP}_t = \alpha_1 + \beta_1 \times \mathrm{HICP}_{t-1} + \gamma_1 \times si_t + \varepsilon_{1t}, \quad (1)$$

$$\mathrm{HICP}_t = \alpha_2 + \beta_2 \times \mathrm{HICP}_{t-1} + \gamma_2 \times si_t + \delta_1 \times u_t + \varepsilon_{2t}, \quad (2)$$

$$\mathrm{HICP}_t = \alpha_3 + \beta_3 \times \mathrm{HICP}_{t-1} + \delta_2 \times u_t + \varepsilon_{3t}, \quad (3)$$

where:

HICP is the quarterly harmonised index of consumer prices.

$si$ is the sentiment index.

$u$ is the unemployment rate.

$\alpha_1, \alpha_2, \alpha_3, \beta_1, \beta_2, \beta_3, \gamma_1, \gamma_2, \delta_1, \delta_2$ are the parameters of the model.

$\varepsilon_{1t}, \varepsilon_{2t}, \varepsilon_{3t}$ are the errors.

$t$ is the time index.

ARDL models were chosen for this research because they can reduce endogeneity caused by the connection between inflation and unemployment. Similar models are built for all CEE countries using monthly ESI, HICP, and unemployment rates provided by Eurostat.

We explored multiple settings, including univariate and multivariate series, adding extra predictors: the sentiment index and unemployment rate. Specifically, we investigate the following configurations:

- Univariate Series: we used only past values of HICP to forecast future values.
- Multivariate Series setting I: we combined past values of HICP and the unemployment rate.
- Multivariate Series setting II: we combined past values of HICP and the sentiment index / ESI.
- Multivariate Series setting III: we combined past values of HICP, the sentiment index / ESI, and the unemployment rate.

The ML models are run on Python, while the ARDL models in EViews. To implement our ML models for time series forecasting, we proceeded as follows:

- pre-processed the data, structuring it into suitable input-output pairs for supervised learning.
- split the data sets into training and testing subsets.
- defined the RF, SVR, and LSTM model architecture.
- trained the model using historical data, performing a grid search for the best hyperparameter values.
- used the trained model to provide forecasts on unseen future data points and assessed the prediction accuracy.

We first prepared our datasets for use with ML methods. Preparing time series data for ML involves specific steps to ensure the data is properly structured for modeling. One essential step is creating lag features, which are basically previous time steps of the target variable or related variables used as inputs for the model. Lag features help incorporate the temporal relationships inherent in time series data by including past values to predict future ones.

In the pre-processing step, we created a dataset suitable for a supervised learning method, consisting of pairs in the form (X, Y) where **X** is the input vector and **Y** is the output value. In this context, **X** is a vector made up of several past values, and Y is the next value in the time series that we aim to predict.

In the univariate setting, if we denote by $\boldsymbol{X} = (X_0, X_1, X_2, ..., X_T)$, our time series, the training and test sets are structured as follows:

$$\left(X_{t-nlags-1}, X_{t-nlags-2}, \ldots X_{t-1}, X_t\right), \left(X_{t+1}\right) \tag{4}$$

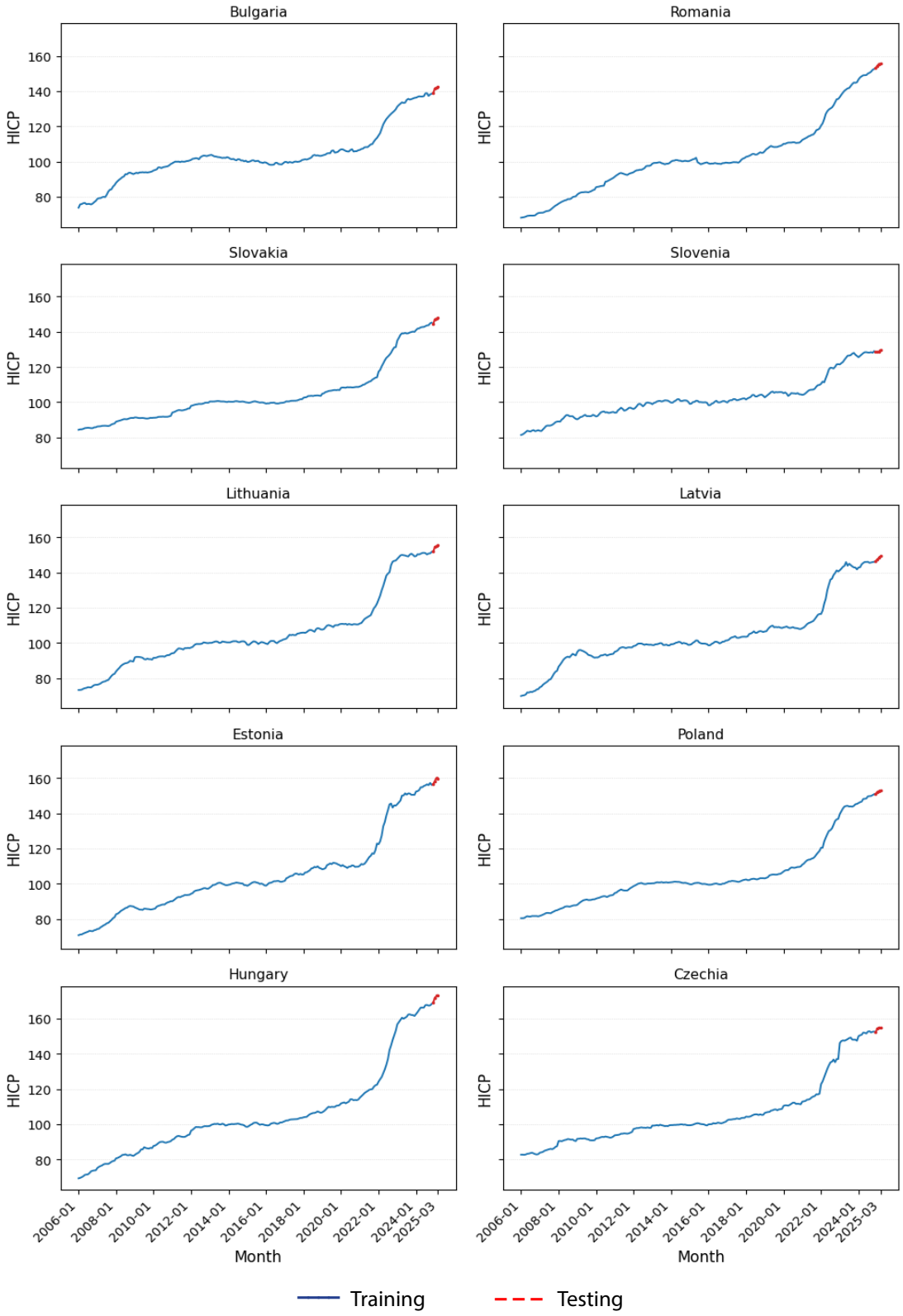
$$\left(X_{t-nlags-2}, X_{t-nlags-3}, \ldots X_t, X_{t+1}\right), \left(X_{t+2}\right) \tag{5}$$

$$\left(X_{t-nlags-3}, X_{t-nlags-4}, \ldots X_{t+1}, X_{t+2}\right), \left(X_{t+3}\right) \tag{6}$$

where *nlags* is the number of lags (past values) used to build sequences of data points to predict the value at $t+1$. In our experiments, we used a lag of 6-time steps based on an autocorrelation analysis.

Scaling features are another crucial preprocessing step for many ML methods because it ensures all characteristics contribute equally to the model's learning process. Without scaling, features with larger numerical ranges tend to dominate the learning algorithm, causing biased results and subpar model performance. This step is especially important for algorithms based on distance metrics. For SVR, we applied Min-Max scaling, bringing all values into the [0,1] range. As a result, we achieved faster convergence during training, improved the model's stability and performance, and made sure each feature was properly weighted in the prediction process (Ahsan et al., 2021; Han et al., 2014).

**Figure 1. Monthly HICP for CEE countries (2015 = 100). The train-test split.**




Source: Eurostat database

Regarding the split of the quarterly data sets for Romania (training and testing subsets), we kept the last 4 values from the time series as a test set and used the other 62 values as a training set. For the monthly data for CEE countries, we kept the last 3 values (2025M1, 2025M2, 2025M3) as a test set and used the rest (2006M1-2024M12) as a training data set. The monthly HICP for all CEE countries is shown in Figure 1, while the ESI and unemployment rate are shown in the Appendix, Figures A1 and A2.

For the concrete implementation of the ML methods, we used the scikit-learn (Pedregosa et al., 2011), Keras (Chollet, 2015), and TensorFlow (Abadi et al., 2016) libraries. For RF and SVR, we relied on scikit-learn, while for LSTM, we used Keras and TensorFlow. Scikit-learn provides extensive customization options for RF and SVR models through various hyperparameters. For RF, users can adjust the number of trees, maximum depth, and criteria for splitting nodes, among other settings. Similarly, for SVR, users can select different kernels (linear, polynomial, RBF), set the regularization parameter (C), and define the epsilon-insensitive tube width. This flexibility allows fine-tuning of models to optimize performance for specific tasks.

Using LSTM networks with Keras, a user-friendly neural network API, and TensorFlow, a robust open-source ML framework, provides numerous benefits such as ease of use, flexibility, and scalability. Keras offers a high-level interface for constructing neural network architectures, enabling users to develop and train complex models with minimal code. Meanwhile, TensorFlow supplies efficient computation and optimization tools, making it ideal for training large-scale deep learning models.

While RF and SVR models do not need a specific architecture, LSTM, on the other hand, requires careful design. We built an LSTM network with two stacked LSTM layers and a final Dense layer to generate forecasts. Recurrent dropout was used in both LSTM layers, along with L2 regularization on the Dense layer to prevent overfitting. The network was optimized with Adam.

Hyperparameter tuning is essential for optimizing machine learning models used in economic time series prediction. These methods, ranging from traditional statistical techniques to advanced deep learning architectures, rely heavily on hyperparameters to achieve optimal performance. Selecting the right hyperparameters greatly impacts the model's accuracy, robustness, and ability to capture complex patterns unique to economic data. Proper hyperparameter choice influences the model's capacity to learn from data, generalize to new cases, and avoid overfitting. However, manually tuning hyperparameters can be time-consuming and biased, making automated methods necessary. Among these, grid search with cross-validation is a popular and effective approach. Grid search tests a set of predefined hyperparameter combinations

to find the best configuration to maximize performance. Cross-validation is a dependable validation method that splits data into multiple parts, allowing thorough evaluation of the model's performance across different data splits. Combining grid search with cross-validation helps practitioners find optimal hyperparameters, evaluate the model's ability to generalize, and reduce overfitting risks.

Research in economic forecasting has shown that grid search with cross-validation effectively improves the accuracy and consistency of predictive models. For example, studies by Hyndman and Athanasopoulos (2018) highlight how hyperparameter tuning through grid search with cross-validation enhances the forecasting ability of traditional time series methods like ARIMA and exponential smoothing. Additionally, in recent years, machine learning techniques such as ensemble methods, SVMs, and deep learning models like RNNs and LSTMs have become more popular in economic time series forecasting. Grid search with cross-validation has been key in fine-tuning hyperparameters of machine learning models to achieve better predictive accuracy and stability, as demonstrated in a study on stock price prediction (Hoque and Aljamaan, 2021). Similarly, in computer vision, this approach has helped fine-tune hyperparameters for convolutional neural networks, resulting in higher accuracy and robustness in image recognition (Szegedy et al., 2016). These results highlight the important role of hyperparameter tuning techniques in reaching top-tier performance across various applications.

Therefore, hyperparameter tuning - by exploring the hyperparameter space and thoroughly assessing model performance - is a vital step in optimizing ML models. As the field of economic forecasting advances, the significance of effective hyperparameter tuning techniques like grid search with cross-validation cannot be overstated.

Our approach uses a grid search to find the best parameters for RF, SVR, and LSTM. We apply 3-fold cross-validation to ensure reliable performance evaluation. Because of the sequential nature of the data, we use the *TimeSeriesSplit* class from the scikit-learn library, which preserves the order of data points during cross-validation. This method divides the data into training and testing sets so that each training set precedes its corresponding test set, mimicking a real-world forecasting situation.

For the RF, we searched for the best values for the following parameters:

- *n_estimators* - specifies the number of trees in the forest;
- *max_depth* - sets the maximum depth of each tree in the forest;
- *min_samples_split* - defines the minimum number of samples required to split an internal node;
- *max_features* - specifies the number of features to consider when looking for the best split (as fraction of the total number of features.

For SVR, we used the following parameters to find the best model:

- *C* – this parameter determines the strength of regularization, also called the regularization parameter;
- *gamma* - specifies the kernel coefficient for 'rbf', and 'poly';
- *coef0* - is the independent term in the kernel function. It is only significant in 'poly' and 'sigmoid' kernels;
- *Epsilon* - defines the epsilon-tube where errors are not penalized. It controls the margin's width in the ε-insensitive loss function;
- *kernel* - specifies the type of kernel to use in the algorithm (we used 'poly', and 'rbf' in our experiments);
- *degree* - the degree of the polynomial kernel function ('poly').

For LSTM, we considered the following hyperparameters for tuning.

- *recurrent dropout (layer1, layer2)* - specifies the dropout rate for recurrent connections in the LSTM units during training, helping to prevent overfitting;
- *dropout (layer 1, layer2)* - specifies the dropout rate for the forward connections. Used to prevent overfitting;
- *number of neurons* - the number of LSTM units or neurons in the hidden layers;
- *batch size* - the number of samples processed before updating the model's parameters during training;
- *L2* for the kernel_regularizer - allows us to apply L2 regularization to the weights of the final Dense layer. L2 regularization penalizes large weights by adding a term to the loss function that is proportional to the square of the weights, helping to prevent overfitting.

The values used for these parameters are presented in Table 1.

After selecting the best parameters, we trained the model and assessed its performance on both the training and testing datasets. During the training of the LSTM models, we used the Mean Squared Error (MSE) as the loss function, while also monitoring the Mean Absolute Percentage Error (MAPE) and Mean Absolute Error (MAE). MSE calculates the average squared difference between forecasted and actual values, reflecting the model's variance (James et al., 2013). MAPE, on the other hand, measures the average percentage difference between predicted and actual values, making it useful for understanding errors in relation to the true value (Hyndman and Koehler, 2006). In regression tasks, MSE, MAPE, and MAE serve to evaluate different aspects of model performance. Although MSE penalizes larger errors more heavily because of squaring, MAPE expresses errors as a percentage of the actual values, offering insight into the relative size of errors. MAPE is especially useful in forecasting and business settings where understanding the magnitude of error matters (Hyndman and Koehler, 2006). MSE, MAPE, and MAE are calculated as follows:

**Table 1. The values for parameters in SVR, RF and LSTM networks**

| Grid parameters | | | | | |
|---|---|---|---|---|---|
| **SVR** | | **RF** | | **LSTM** | |
| **Parameter** | **Value** | **Parameter** | **Value** | **Parameter** | **Value** |
| **C** | 1, 2, 3, 4, 5, 6, 7, 8, 9 | n_estimators | 50, 75, 100, 150, 200 | Recurrent dropout layer 1 | 0.0, 0.1, 0.2 |
| **Gamma** | 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9 | max_depth | None, 5, 10, 20, 30, 40, 50 | Recurrent dropout layer 2 | 0.0, 0.1, 0.2 |
| **Coef0** | 0, 0.01, 0.5, 0.1, 1.0, 2.0, 2.5 | min_samples_split | 2, 5, 10, 12 ,15, 20 | Dropout layer1 | 0.0, 0.1, 0.2 |
| **epsilon** | 0.01, 0.05, 0.1, 0.2, 0.3 | max_features | 0.1, 0.3, 0.5, 0.7, 0.9 | Dropout layer2 | 0.0, 0.1, 0.2 |
| **kernel** | rbf, poly | | | L2 regularization | 0.0, 0.0001, 0.001, 0.003 |
| **degree** | 1, 2, 3, 4, 5 | | | Number of units | 128, 256, 512 |
| | | | | Batch size | 1, 8, 16 |

Source: own construction

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y}_i\right)^2 \tag{7}$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \widehat{y}_i}{y_i}\right| \times 100 \tag{8}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}\left|y_i - y_i\right| \tag{9}$$

where:

n – the number of data points;

$y_i$ – actual observed value for a certain $i^{\text{th}}$ data point;

$\widehat{y}_i$ – predicted value for a particular $i^{\text{th}}$ data point.

All preprocessing and model selection are performed within the training data only to avoid look-ahead bias. Standardization/scaling parameters are fit on each training window and then applied to the corresponding validation/test data; no information from the validation/test sets

enters the fit. Hyperparameters are chosen by time-series cross-validation with expanding windows (scikit-learn TimeSeriesSplit), preserving temporal order. For neural networks, early stopping monitors the within-fold validation loss only. After selection, each model is refit on the entire training window and evaluated once on the held-out test observations.

Finally, using the trained model, we predicted the next three HICP values and calculated the performance metrics.

# 4. Results

## 4.1 Results based on simulations

To assess whether our models outperform a naïve forecast, we conducted two simulation studies.

First, we conducted a basic simulation study by generating two series with 100 observations each, which were used to fit ARDL and ML models (RF, SVR, and LSTM) and to make forecasts.

We considered two data-generating processes (DGPs):

$$x_t = 0.3 \times x_{t-1} + e_{1t} \tag{10}$$

$$y_t = 0.2 \times |x_t| + e_{2t} \tag{11}$$

*where* $e_{1t}$, $e_{2\,t}$ are Gaussian white noise (random numbers).

We generated two data series, each with 100 values, and split them into training and testing sets—using the first 80 for training and the remaining for testing. Using these datasets, we fitted four models: ARDL, SVR, RF, and LSTM, based on the feature vector provided by:

$$X_t = \{y_{t-1},\ y_{t-2},\ y_{t-3},\ x_t,\ x_{t-1}\} \tag{12}$$

After fitting the models, we performed one-step-ahead predictions on the test set and compared the performance metrics of the four prediction models mentioned above with the naïve forecast provided by:

$$\widehat{y_t^{\text{naive}}} = y_{t-1} \tag{13}$$

The ARDL model was estimated using OLS with an intercept and the previously mentioned regressor. The RF was built with 200 trees and the following parameters:

min_samples_split = 10

min_samples_leaf = 1

max_features = 0.2

max_depth = 10

The SVR model used a one-degree polynomial kernel with C = 1 and $\varepsilon$ = 0.01 and the LSTM network had a single layer with 16 units, trained for 50 epochs with a batch size of 8.

The loss per test point was the squared error, needed for Diebold-Mariano tests: $l_t = (y_t - \hat{y}_t)^2$ . We reported the aggregated metrics on the test set: RMSE, MSE, MAE, and their standard deviations in Table 2 for 200 replications of the experiment.

**Table 2. Performance metrics for the test sample (one-step forecasts)**

| Method | RMSE | sd(RMSE) | MSE | sd(MSE) | MAE | sd(MAE) |
|---|---|---|---|---|---|---|
| **ARDL** | 0.8059 | 0.1701 | 0.6495 | 0.3342 | 0.6943 | 0.1491 |
| **RF** | 0.8069 | 0.1631 | 0.6510 | 0.3061 | 0.6725 | 0.1409 |
| **SVR** | 0.7931 | 0.1804 | 0.6291 | 0.3396 | 0.6775 | 0.1570 |
| **LSTM** | 0.7316 | 0.2043 | 0.5353 | 0.4070 | 0.5969 | 0.1720 |
| **Naïve** | 1.0671 | 0.2363 | 1.1388 | 0.6600 | 0.8761 | 0.2063 |

Source: own computations

We performed paired comparisons (each method versus Naïve) and computed Diebold–Mariano (DM) test on $\bar{d}$ squared-error loss, which 1are shown in Table 3. For $h$ = 1, DM reduces to a t-type statistic on where $d_t = l_{M,t} - l_{\text{naive},t}$ We also computed the Wilcoxon signed-rank test on paired $l_{M,t}$ versus $l_{\text{naive},t}$ . The 1-sided p-values are converted when the observed statistic has the direction consistent with the alternative $E[d] < 0$ (all four methods produced negative $\bar{d}$. For h-step-ahead forecasts the error sequence is overlapping, so we estimate Var(dt) with a Newey–West HAC estimator using a Bartlett kernel and truncation *lag* $q = h - 1$.

**Table 3. Paired tests, DM, and Wilcoxon p-values**

| Comparison | Mean(d) | DM stat | DM p-value (2-sided) | DM p-value (1-sided) | Wilcoxon p-value (2-sided) | Wilcoxon p-value (1-sided) |
|---|---|---|---|---|---|---|
| **ARDL vs Naïve** | −0.4892 | −2.3148 | 0.0320 | 0.0160 | 0.0266 | 0.0133 |
| **RF vs Naïve** | −0.4877 | −2.0313 | 0.0564 | 0.0282 | 0.0296 | 0.0148 |
| **SVR vs Naïve** | −0.5097 | −2.4419 | 0.0246 | 0.0123 | 0.0266 | 0.0133 |
| **LSTM vs Naïve** | −0.6035 | −2.1993 | 0.0404 | 0.0202 | 0.0362 | 0.0181 |

Source: own computations

Table 3 shows paired comparisons of replication losses for all four methods versus the naïve forecast. All four fitted methods produce significantly lower one-step-ahead MSE than the naïve hold-last benchmark in this experiment. In the test sample, the naïve MSE was 1.1388; LSTM achieves the greatest reduction (MSE = 0.5353, about 53% lower), while ARDL and SVR decrease MSE by approximately 43% (MSE around 0.6495 and 0.6291, respectively), and RF by about 45% (MSE = 0.6510). These effect sizes are practically large and consistent across replications (see mean ± SD tables). We assessed significance with paired one-sided Diebold–Mariano tests ($H_1$: method loss < naïve loss) and one-sided Wilcoxon signed-rank tests as a robust check. Converting the reported two-sided DM p-values to one-sided p-values (and using Wilcoxon one-sided directly) yields DM one-sided $p \approx 0.0160$ (ARDL), 0.0282 (RF), 0.0123 (SVR), and 0.0202 (LSTM), and Wilcoxon one-sided $p \approx 0.0133$, 0.0148, 0.0133, 0.0181, respectively. After Benjamini-Hochberg FDR correction across the four comparisons, all adjusted p-values remain below 0.05, confirming that each method outperforms the naïve benchmark even after controlling for multiple testing. We observed that RF is marginal on the parametric DM test but significant with the Wilcoxon test, indicating minor departures from DM's normality assumptions (such as skewness or outliers) in RF's paired differences. Reporting both tests and the effect sizes above provides the clearest and most transparent account.

The second simulation study examined more complex DGPs that better resemble real processes.

Specifically, we generated a monthly series of log CPI and two exogenous covariates (unemployment $u_t$ and sentiment index $s_t$) under a small set of controlled scenarios. Each Monte Carlo replication produces a panel:

$$\left(\text{date}_t, \ln CPI_t, u_t, s_t\right)_{t=1}^{T} \tag{14}$$

with a monthly index *t*. The core DGP for log-CPI is a persistent autoregressive process with exogenous effects, optional MA(1) shocks, an optional regime nonlinearity on *st*, and an optional structural break. Specifically, we simulate:

$$u_t = \mu_u + \phi_u\left(u_{t-1} - \mu_u\right) + \varepsilon_t^{(u)}, \quad \varepsilon_t^{(u)} \sim \mathcal{N}(0, \sigma_u^2) \tag{15}$$

$$s_t = \mu_s + \phi_s(s_{t-1} - \mu_s) + \varepsilon_t^{(s)}, \quad \varepsilon_t^{(s)} \sim \mathcal{N}(0, \sigma_u^2) \tag{16}$$

$$w_t \sim \mathcal{N}(0, \sigma_w^2) \tag{17}$$

$\eta_t = w_t + \theta w_{t-1}$, MA(1) innovation; if $\theta = 0$, reduces to *w_t* (18)

$$\ln CPI_t = \mu + \phi \ln CPI_{t-1} + \beta_u u_{t-1} + \beta_s s_{t-1} + \gamma u_{t-1} s_{t-1} + f_{NL}\left(s_{t-1}\right) + \eta_t + \Delta_t^{(\text{break})} \tag{19}$$

where:

- $f_{\text{NL}}(s_{t-1})$ is an optional nonlinear/regime term activated when sentiment crosses a threshold $s_{t-1} > s^*$ In the default scenarios, we set

$$f_{\text{NL}}\left(s_{t-1}\right) = \begin{cases} \hat{e}\left(s_{t-1} - s^*\right) & s_{t-1} > s^* \\ 0 & otherwise \end{cases}$$

So, sentiment can increase persistence or level, conditional on a high-sentiment regime.

- $\Delta_t^{(\text{break})}$ is an additive structural break applied from a break time $t \geq t_{\text{break}}$

$$\Delta_t^{(\text{break})} = \begin{cases} \delta & t \geq t_{\text{break}} \\ 0 & t < t_{\text{break}} \end{cases}$$

The simulator enforces that $t_{\text{break}}$ is placed only when there is sufficient pre- and post-break data.

We generated 5 datasets, each representing a different scenario:

- *baseline_linear*: $\phi = 0.95$, $\beta_u = -0.05$, $\beta_s = 0.10$, $\theta = 0$
- *high_persistence*: $\phi = 0.99$
- *nonlinear_regime:* $\phi = 0.90$, $s^* = 0.5$, $\kappa = 0.4$, $\gamma = 0.0$
- *structural_break*: additive $\Delta = 0.08$ in $\ln CPI_t$ at safe breaks
- *ma_noninvertible*: $\theta = -1.2$ to probe non-invertible shocks

Our MA non-invertible scenario and regime/break experiments are included to address concerns that annual-rate constructions can induce non-invertibility and complicate inference; by working in the $CPI_t$ domain and testing different DGP regimes, we ensure models are evaluated under the pathologies that can appear in real conditions.

We used R = 200 replications with T = 240 months. The simulation script enforces $T \geq N_{\text{lags}} + H_{max} + B$, where $N_{\text{lags}} = 6$, is the lag window used by all models, $H_{max} = 6$ is the maximal forecast horizon evaluated, and $B$ is a buffer for stable training (currently we set B = 12) which prevents invalid lag construction. The break position $t_{\text{break}}$ is chosen so that $t_{\text{break}} \geq N_{\text{lags}} + 12$, $t_{\text{break}} \leq \text{T} - H_{max} - 6$ to ensure sufficient pre/post samples for estimation and evaluation; if no valid break is possible, no break is applied.

We fitted the same four models on each replication: (1) an ARDL-style linear model estimated by OLS; (2) an RF regressor; (3) an SVR model with a polynomial kernel; and (4) a small LSTM network. All methods use the same input data: the last $N_{\text{lags}} = 6$ lags of ln$CPI$ and the same lagging convention for covariates (unemployment and sentiment).

For the ARDL OLS we built regressors:

$$X_t = \left\{\ln CPI_{t-1}, \ldots, \ln CPI_{t-6},\ u_{t-1}, \ldots,\ u_{t-6},\ s_{t-1},\ \ldots, s_{t-6}\right\} \tag{20}$$

and estimated

$$\ln CPI_t = \alpha + X_t\beta + \varepsilon_t \tag{21}$$

For RF, SVR, and LSTM we constructed a vector following the same lag-window convention. Before SVR and LSTM, we standardized features (zero mean, unit variance). For each replication, the last $H_{max} = 6$ months are reserved for evaluation. Let $T$ be the series length; the training sample ends at index $t_{\text{train}} = T - H_{max} - 1$. Forecasts are generated recursively for horizons $h \in \{1, 3, 6\}$.

Besides the performance metrics used in the first simulation study, for each scenario and forecast horizon, we computed the *win_rate* of a forecasting method as the fraction of Monte-Carlo replications in which that method attains the smallest RMSE. Denote by $L_{r,m}$ the loss (RMSE) in replication $r$ for method $m$ and let $M$ be the set of methods. For replication $r$ we defined $w_{r,m} := \begin{cases} \frac{1}{k_r} & if m \in \arg\min_{j \in M} L_{r,j} \\ 0 & otherwise \end{cases}$ where $k_r$ is the number of methods tied for the minimum loss in replication $r$ (so ties are split equally). The reported win-rate is:

$$\text{win\_rate}_{\text{m}} = \frac{1}{R}\sum_{r=1}^{R} w_{r,m}$$

with $R$ the number of replications. We reported bootstrap 95% confidence intervals for win-rates (nonparametric resampling of replications) and complemented win-rates with mean ± sd of RMSE and paired statistical tests (paired t-test and Wilcoxon signed-rank) on replication RMSE differences to assess whether observed ranking differences are statistically significant.

Table 4 outlines the parameters used in the simulations.

## Table 4. Default parameters used in the simulation

| Parameters | Values |
|---|---|
| **Replications *R*** | 200 |
| **Series length *T*** | 240 months |
| **Lags** | 6 |
| **Forecast horizons** | {1, 3, 6} months |
| **RF: n_estimators** | 200 |
| **SVR** | polynomial kernel (degree 1), $C = 1.0$, $\varepsilon = 0.01$ |
| **LSTM** | 16 units, 50 epochs, batch size 8 |

Source: own construction

The performance metrics for each scenario, forecasting horizon, and method are presented in Tables 5, 6, 7, 8, and 9, while the paired comparisons (each method versus Naïve) and the DM and Wilcoxon one-sided tests are presented in Tables A6-A7.

## Table 5. Performance metrics for the baseline scenario

| Method | Mean(RMSE) | sd(RMSE) | Mean(MAPE) | sd(MAPE) |
|---|---|---|---|---|
| **H = 1** | | | | |
| **ARDL** | 0.3959 | 0.3289 | 3.4890 | 2.9674 |
| **LSTM** | 0.4724 | 0.3636 | 4.2088 | 3.4600 |
| **Naive** | 0.3972 | 0.3228 | 3.4811 | 2.9046 |
| **RF** | 0.4445 | 0.3446 | 3.8817 | 3.0782 |
| **SVR** | 0.4117 | 0.3297 | 3.5934 | 2.9375 |
| **H = 3** | | | | |
| **ARDL** | 0.4257 | 0.3037 | 3.6753 | 2.6657 |
| **LSTM** | 0.4636 | 0.3504 | 4.0336 | 3.1793 |
| **Naive** | 0.6877 | 0.5119 | 5.8805 | 4.2978 |
| **RF** | 0.4560 | 0.3441 | 3.9508 | 2.9681 |
| **SVR** | 0.4245 | 0.2968 | 3.6648 | 2.6094 |
| **H = 6** | | | | |
| **ARDL** | 0.4350 | 0.3308 | 3.8381 | 3.0583 |
| **LSTM** | 0.5184 | 0.3755 | 4.6114 | 3.5766 |
| **Naive** | 1.0008 | 0.7415 | 8.8593 | 6.8600 |
| **RF** | 0.5016 | 0.3913 | 4.4790 | 3.7828 |
| **SVR** | 0.4470 | 0.3302 | 3.9573 | 3.0982 |

Source: own computations

## Table 6. Performance metrics for the MA noninvertible scenario

| Method | Mean(RMSE) | sd(RMSE) | Mean(MAPE) | sd(MAPE) |
|---|---|---|---|---|
| | | **H = 1** | | |
| **ARDL** | 0.4831 | 0.3725 | 4.0363 | 3.0791 |
| **LSTM** | 0.6640 | 0.5451 | 5.5748 | 4.6241 |
| **Naive** | 0.6026 | 0.4373 | 5.0275 | 3.6783 |
| **RF** | 0.4755 | 0.3974 | 4.0028 | 3.4231 |
| **SVR** | 0.4739 | 0.3773 | 3.9748 | 3.1883 |
| | | **H = 3** | | |
| **ARDL** | 0.5106 | 0.3800 | 4.2899 | 3.2413 |
| **LSTM** | 0.6813 | 0.5163 | 5.7253 | 4.4557 |
| **Naive** | 0.6967 | 0.5199 | 5.8242 | 4.3998 |
| **RF** | 0.5528 | 0.4252 | 4.6538 | 3.7072 |
| **SVR** | 0.4913 | 0.3845 | 4.1407 | 3.3147 |
| | | **H = 6** | | |
| **ARDL** | 0.5760 | 0.4276 | 4.8668 | 3.6972 |
| **LSTM** | 0.7484 | 0.5446 | 6.3034 | 4.7679 |
| **Naive** | 0.7676 | 0.6375 | 6.5006 | 5.5594 |
| **RF** | 0.5349 | 0.4476 | 4.5549 | 3.9959 |
| **SVR** | 0.5310 | 0.4182 | 4.4912 | 3.6875 |

Source: own computations

## Table 7. Performance metrics for the high persistence scenario

| Method | Mean(RMSE) | sd(RMSE) | Mean(MAPE) | sd(MAPE) |
|---|---|---|---|---|
| | | **H = 1** | | |
| **ARDL** | 0.3953 | 0.3289 | 0.7412 | 0.6234 |
| **LSTM** | 0.7007 | 0.5272 | 1.3038 | 0.9786 |
| **Naive** | 0.3953 | 0.3266 | 0.7413 | 0.6204 |
| **RF** | 0.5371 | 0.4174 | 0.9992 | 0.7765 |
| **SVR** | 0.5611 | 0.4248 | 1.0460 | 0.7947 |
| | | **H = 3** | | |
| **ARDL** | 0.4204 | 0.3045 | 0.7803 | 0.5618 |
| **LSTM** | 0.7013 | 0.5418 | 1.2943 | 0.9876 |
| **Naive** | 0.7148 | 0.5520 | 1.3283 | 1.0178 |
| **RF** | 0.6682 | 0.6217 | 1.2334 | 1.1352 |
| **SVR** | 0.5458 | 0.4414 | 1.0106 | 0.8100 |
| | | **H = 6** | | |
| **ARDL** | 0.4382 | 0.3345 | 0.8144 | 0.6262 |
| **LSTM** | 0.4344 | 0.4574 | 1.3516 | 1.0029 |
| **Naive** | 1.0754 | 0.8007 | 2.0079 | 1.4916 |
| **RF** | 0.8543 | 0.7403 | 1.5807 | 1.3437 |
| **SVR** | 0.6595 | 0.4566 | 1.2316 | 0.8649 |

Source: own computations

## Table 8. Performance metrics for the non linear regime scenario

| Method | Mean(RMSE) | sd(RMSE) | Mean(MAPE) | sd(MAPE) |
|---|---|---|---|---|
| | | **H = 1** | | |
| **ARDL** | 0.4133 | 0.3380 | 43.5098 | 119.5565 |
| **LSTM** | 0.3787 | 0.4556 | 31.2507 | 77.2534 |
| **Naive** | 0.5028 | 0.3712 | 53.7913 | 144.6628 |
| **RF** | 0.4995 | 0.3941 | 45.0396 | 112.6434 |
| **SVR** | 0.4064 | 0.3325 | 41.0670 | 104.4457 |
| | | **H = 3** | | |
| **ARDL** | 0.5349 | 0.4012 | 63.9732 | 254.8887 |
| **LSTM** | 0.3353 | 0.3732 | 41.1261 | 199.6160 |
| **Naive** | 1.0265 | 0.7700 | 118.9202 | 688.5146 |
| **RF** | 0.6306 | 0.4323 | 69.2605 | 262.5694 |
| **SVR** | 0.5373 | 0.4017 | 64.7472 | 263.4395 |
| | | **H = 6** | | |
| **ARDL** | 0.6536 | 0.4471 | 47.6812 | 85.8452 |
| **LSTM** | 0.4801 | 0.4587 | 36.0240 | 54.8546 |
| **Naive** | 1.4918 | 1.1382 | 120.6361 | 271.2341 |
| **RF** | 0.7062 | 0.5161 | 50.0000 | 95.5245 |
| **SVR** | 0.6772 | 0.4611 | 48.5897 | 85.7147 |

Source: own computations

**Table 9. Performance metrics for the structural break regime scenario**

| Method | Mean(RMSE) | sd(RMSE) | Mean(MAPE) | sd(MAPE) |
|---|---|---|---|---|
| | | **H = 1** | | |
| **ARDL** | 0.4034 | 0.3301 | 4.1605 | 3.5618 |
| **LSTM** | 0.3620 | 0.3599 | 3.7435 | 2.9520 |
| **Naive** | 0.3972 | 0.3228 | 4.0664 | 3.4248 |
| **RF** | 0.4493 | 0.3503 | 4.6229 | 3.8844 |
| **SVR** | 0.4157 | 0.3290 | 4.2514 | 3.5201 |
| | | **H = 3** | | |
| **ARDL** | 0.4310 | 0.3044 | 4.3693 | 3.2208 |
| **LSTM** | 0.3642 | 0.2721 | 3.7325 | 2.8881 |
| **Naive** | 0.6877 | 0.5118 | 6.8433 | 5.0316 |
| **RF** | 0.4475 | 0.3498 | 4.5557 | 3.6537 |
| **SVR** | 0.4275 | 0.2971 | 4.3448 | 3.1650 |
| | | **H = 6** | | |
| **ARDL** | 0.4334 | 0.3291 | 4.5163 | 3.7019 |
| **LSTM** | 0.3144 | 0.3378 | 3.3968 | 2.9580 |
| **Naive** | 1.0008 | 0.7415 | 10.3928 | 8.2422 |
| **RF** | 0.5165 | 0.3963 | 5.5304 | 4.8589 |
| **SVR** | 0.4455 | 0.3311 | 4.6617 | 3.8107 |

Source: own computations

Across all simulated scenarios and forecast horizons, the learning-based models (ARDL, RF, SVR, LSTM) consistently outperform the naïve benchmark. Aggregating results over replications shows lower errors for our models on RMSE and MAE, with improvements that become more pronounced as the horizon lengthens. Formal Diebold–Mariano tests against the naïve forecast corroborate these gains, indicating that the error reductions are not due to sampling variability but reflect genuine predictive improvements.

The performance edge is robust to the underlying data-generating mechanism. ARDL provides stable, across-the-board accuracy, SVR excels when short-memory or moving-average features dominate, and LSTM delivers competitive (often best-in-class) long-horizon forecasts when nonlinearities or regime changes are present. While the naïve forecast can be competitive at h=1 in some scenarios, it deteriorates rapidly with horizon, whereas our models either maintain their advantage or widen it. This pattern is consistent across the scenario-specific win-rates and the paired tests versus the benchmark.

From a practical standpoint, these findings justify replacing the naïve approach with our modeling toolkit in applications where accuracy beyond the next step matters. The combination of lower average errors, statistically significant improvements, and complementary strengths across methods suggests that practitioners can expect reliable gains over the naïve baseline, with ARDL as a robust default and SVR/LSTM offering additional benefits when the data exhibit MA structure or nonlinear/regime-shift dynamics.

## 4.2 ARDL models for Romania using quarterly data and sentiment index

The ADF test on seasonally adjusted datasets indicated that, except for the sentiment index, which is stationary in level at a 1% significance level, the other series are integrated of order one. The ARDL models used to develop HICP forecasts are presented in Table 10. Multicollinearity is not an issue in this case. Additionally, there is no significant correlation between predictors, as Pearson's coefficient is 0.202.

**Table 10. The ARDL estimates**

| Indicator | Coef./stat. with prob in brackets | | |
|---|---|---|---|
| | Model 1 | Model 2 | Model 3 |
| **$HICP_{t-1}$** | 0.836***<br>(0.000) | 0.833***<br>(0.000) | 0.828***<br>(0.066) |
| **$si_t$** | −0.060*<br>(0.081) | −0.059*<br>(0.081) | – |
| **$unemployment_t$** | – | −0.071*<br>(0.061) | −0.079*<br>(0.057) |
| **constant** | 16.884**<br>(0.016) | 17.710**<br>(0.015) | 18.515<br>(0.011) |
| **Breusch-Godfrey test for one lag** | 1.923<br>(0.1655) | 1.852<br>(0.173) | 2.334<br>(0.126) |
| **White test** | 4.155<br>(0.5272) | 7.536<br>(0.581) | 3.935<br>(0.558) |
| **Shapiro-Wilk test** | 0.899<br>(0.184) | 0.807<br>(0.197) | 0.903<br>(0.179) |
| **Ramsey Reset Test** | 2.456<br>(0.117) | 2.356<br>(0.124) | 1.315<br>(0.251) |

Source: own computations

Note: *,**,*** suggest significance at 10%,5% and 1% level, respectively. p-values in brackets

These ARDL models form the basis of our HICP forecasting method, and their prediction accuracy is evaluated. Table 11 shows that Model 1, which includes the previous period's HICP and the current period's sentiment index, outperforms the other ARDL models based on MAPE and MSE.

**Table 11. Inflation forecasts based on ARDL models and their accuracy (horizon 2023Q1-2023Q4)**

| Time | Model 1 | Model 2 | Model 3 |
|---|---|---|---|
| **2023Q1** | 114.19 | 114.23 | 114.46 |
| **2023Q2** | 112.33 | 112.42 | 112.87 |
| **2023Q3** | 110.81 | 110.94 | 111.57 |
| **2023Q4** | 109.72 | 109.89 | 110.49 |
| **MAPE** | 1.73% | 1.81% | 2.15% |
| **MSE** | 4.53 | 5.02 | 7.36 |

Source: own computations

## 4.3 Machine learning techniques for quarterly data for Romania

The best-performing models are identified by the hyperparameter values listed in Table 12. We trained the models using these parameters and then used them to predict the next four HICP values.

## Table 12. The optimal values for the hyperparameters of the machine learning methods for quarterly data, for Romania

| Method | Hyper-parameters | Values of the hyperparameters for | | | |
|---|---|---|---|---|---|
| | | Univariate setting | Multivariate setting | | |
| | | | HICP-Unemployment | HICP-Sentiment index | HICP-Unemployment-Sentiment Index |
| RF | n_estimators | 75 | 200 | 50 | 50 |
| | max_depth | None | None | None | None |
| | min samples split | 15 | 15 | 5 | 2 |
| | max_features | 1.0 | 1.0 | 1.0 | 1.0 |
| SVR | C | 4 | 1 | 1 | 6 |
| | gamma | 0.7 | 0.6 | 0.4 | 0.1 |
| | Coef0 | 1.0 | 0.01 | 0.01 | 2.5 |
| | epsilon | 0.01 | 0.01 | 0.01 | 0.01 |
| | kernel | poly | poly | poly | poly |
| | degree | 1 | 1 | 1 | 1 |
| LSTM | Recurrent dropout layer 1 | 0.1 | 0.2 | 0.0 | 0.1 |
| | Recurrent dropout layer 2 | 0.1 | 0.1 | 0.1 | 0.0 |
| | Dropout layer 1 | 0.2 | 0.2 | 0.2 | 0.2 |
| | Dropout layer 2 | 0.0 | 0.1 | 0.0 | 0.2 |
| | Number of neurons | 512 | 512 | 512 | 512 |
| | Batch size | 1 | 1 | 1 | 1 |
| | L2 for the kernel_ regularizer | 0.0001 | 0.003 | 0.003 | 0.003 |

Source: own computations

While Table 11 reports three ARDL forecasts for 2023Q1–2023Q4 with overall accuracy metrics: MAPE = 1.73% (Model 1), 1.81% (Model 2), and 2.15% (Model 3) with the corresponding MSE values approximately 4.53, 5.02, and 7.36, Table 13 presents the performance of RF, SVR, and LSTM under four feature settings. In the univariate setting, the machine learning methods significantly outperform the ARDL models by achieving lower test errors: RF has a test MSE of 8.13 with a MAPE of 2.20%, SVR has a test MSE of 1.40 with a MAPE of 0.91%, and LSTM has a test MSE of 0.55 with a MAPE of 0.65%. Both SVR and LSTM improved on ARDL Model 1 (with a MAPE of 1.73%) in the univariate test sample, with LSTM providing the largest reduction in MAPE. In the multivariate HICP-unemployment setup, SVR (test MSE = 3.05, MAPE 1.28%) and LSTM (test MSE = 3.86, MAPE 1.46%) again achieve lower test MAPE than ARDL Model 1, which has a MAPE of 1.73%. Conversely, RF (2.29% MAPE) performs worse than ARDL. In the HICP–sentiment multivariate case, RF and LSTM perform poorly on the test set, with RF showing a MAPE of 2.30% and LSTM 2.51%, while SVR's test MAPE (1.75%) is roughly equal to ARDL Model 1's (1.73%). Here, ARDL remains competitive and, in some comparisons (Model 1 versus RF/LSTM), even slightly better. Finally, in the comprehensive HICP–unemployment–sentiment model, machine learning methods show mixed results: RF has a test MAPE of 2.56% (worse than ARDL), SVR scores 1.72% (similar or slightly better), and LSTM achieves 1.12% (better than ARDL Model 1). In summary, ML methods - particularly SVR and LSTM - can outperform ARDL in several contexts, notably the univariate and HICP-unemployment-sentiment cases. However, ARDL remains competitive and sometimes superior when only sentiment indexes are included. The ranking principally depends on the covariates chosen.

On the quarterly Romanian series, the LSTM model that incorporates HICP and unemployment fits the training data very well but shows a clear drop in out-of-sample accuracy, indicating that its high complexity leads it to capture noise rather than the underlying signal. In contrast, the SVR approach, due to its built-in margin maximization and relatively simple parameterization, maintains consistent performance between the training and test sets, demonstrating stronger generalization on unseen quarterly data. When we experimented with reducing the LSTM's size and increasing its regularization, the gap between training and test performance narrowed, confirming that the original architecture was too flexible for the limited volume of quarterly data. These findings emphasize that even at lower frequencies, models with effective capacity control can provide more stable forecasts than over-parameterized deep learning networks.

**Table 13. The performance metrics in the case of RF, SVR and LSTM network**

| Setting | Data set | RF | | | SVR | | | LSTM | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | MAE | MSE | MAPE | MAE | MSE | MAPE | MAE | MSE | MAPE |
| **Univariate setting** | Train data set | 1.13 | 2.48 | 1.06% | 1.23 | 4.47 | 1.16% | 0.98 | 1.83 | 0.92% |
| | Test data set | 2.40 | 8.13 | 2.20% | 1.00 | 1.40 | 0.91% | 0.70 | 0.55 | 0.65% |
| **Multivariate setting: HICP-Unemployment** | Train data set | 1.12 | 2.43 | 1.06% | 1.12 | 4.33 | 1.12% | 0.60 | 0.67 | 0.57% |
| | Test data set | 2.51 | 7.88 | 2.29% | 1.40 | 3.05 | 1.28% | 1.59 | 3.86 | 1.46% |
| **Multivariate setting: HICP-Sentiment Index** | Train data set | 0.74 | 1.18 | 0.70% | 1.17 | 3.97 | 1.11% | 0.35 | 0.23 | 0.33% |
| | Test data set | 2.51 | 9.63 | 2.30% | 1.93 | 3.98 | 1.75% | 2.78 | 11.33 | 2.51% |
| **Multivariate setting: HICP-Unemployment-Sentiment Index** | Train data set | 0.68 | 0.97 | 0.62% | 1.13 | 3.88 | 1.07% | 0.36 | 0.26 | 0.34% |
| | Test data set | 2.79 | 11.05 | 2.56% | 1.88 | 4.23 | 1.72% | 1.22 | 2.02 | 1.12% |

Source: own computations

The results highlight the importance of using multiple economic indicators in forecasting models. LSTM networks, with their ability to handle complex, nonlinear relationships and interactions among predictors, consistently outperformed RF and SVR in this study. However, when only the sentiment index is included, the ARDL models demonstrated a greater capacity to provide more accurate inflation forecasts over a short-term horizon.

## 4.4 ARDL models for CEE countries using monthly data and ESI

The results of the unit root test show that the data series for HICP and unemployment are I(1), while the series for ESI are I(0) for CEE countries. This enables the use of ARDL models. The inflation rate in levels is I(1) at a 1% significance level, which meets the condition for the dependent variable to be non-stationary.

The results shown in Table 14 display ARDL estimations for CEE countries from 2006M1 to 2024M12. Unemployment significantly reduced HICP only in Bulgaria, the Czech Republic, Slovakia, Slovenia, Latvia, and Lithuania, while ESI positively and significantly affected HICP in Lithuania, Bulgaria, Czechia, Estonia, and Slovenia.

## Table 14. The results of estimations for ARDL models for CEE countries (2006M1- 2024M12)

| Variable | Countries | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | BG | CZ | EE | HU | LV | LT | PL | RO | SK | SI |
| $HICP_{t-1}$ | 1.343*** (0.000) | 1.086*** (0.000) | 1.277*** (0.000) | 1.428*** (0.000) | 1.331*** (0.000) | 1.529*** (0.000) | 1.358*** (0.0606) | 1.425*** (0.000) | 1.445*** (0.000) | 1.169*** (0.000) |
| $HICP_{t-2}$ | −0.391*** (0.0005) | 0.089 (0.366) | −0.005 (0.958) | −0.264** (0.025) | −0.052 (0.638) | −0.648*** (0.000) | −0.210* (0.0659) | −0.423*** (0.000) | −0.672*** (0.000) | 0.133 (0.206) |
| $HICP_{t-3}$ | 0.261** (0.018) | −0.177*** (0.009) | −0.267*** (0.0001) | −0.029 (0.808) | −0.281*** (0.000) | 0.344*** (0.005) | −0.008 (0.942) | – | 0.225*** (0.0007) | −0.150 (0.155) |
| $HICP_{t-4}$ | −0.213*** (0.0012) | – | – | −0.132* (0.051) | – | −0.224*** (0.0009) | −0.138** (0.040) | – | – | −0.150** (0.033) |
| $ESI_t$ | 0.001* (0.076) | 0.023** (0.013) | 0.010* (0.066) | – | −0.0003 (0.953) | 0.004* (0.0696) | – | −0.005* (0.205) | 0.004 (0.993) | 0.017* (0.053) |
| $ESI_{t-1}$ | – | −0.029* (0.060) | – | – | – | – | −0.025** (0.016) | – | – | −0.034*** (0.010) |
| $ESI_{t-4}$ | – | – | – | 0.028*** (0.005) | – | – | – | – | – | – |
| $unemployment_t$ | −0.030** (0.036) | −0.078* (0.075) | 0.003 (0.817) | −0.015 (0.532) | −0.023* (0.099) | −0.009* (0.084) | −0.017 (0.185) | −0.059** (0.039) | −0.588* (0.078) | −0.020* (0.092) |
| constant | 0.375 (0.562) | 1.273 (0.285) | −1.350 (0.105) | 0.215 (0.748) | 0.615 (0.444) | −0.263 (0.677) | 0.161 (0.771) | 0.934 (0.143) | 0.452 (0.639) | −0.114 (0.804) |
| Breusch-Godfrey Serial Correlation LM Test for lag=1: stat. | 0.009 (0.9207) | 1.132 (0.2873) | 5.668 (0.0173) | 1.147 (0.2246) | 0.461 (0.4971) | 3.678 (0.055) | 2.437 (0.1185) | 1.153 (0.2829) | 2.502 (0.113) | 2.858 (0.0909) |
| Breusch-Godfrey Serial Correlation LM Test for lag=2: stat. | 0.741 (0.6902) | 3.766 (0.1521) | 6.478 (0.0393) | 2.475 (0.2901) | 0.591 (0.7441) | 2.174 (0.140) | 2.409 (0.122) | 1.153 (0.5618) | 3.567 (0.168) | 3.388 (0.1837) |
| ARCH heteroskedasticity test: stat. | 0.584 (0.4445) | 0.211 (0.6459) | 5.924 (0.014) | 0.573 (0.4488) | 4.740 (0.0295) | 2.901 (0.0885) | 5.780 (0.0162) | 0.008 (0.925) | 0.005 (0.9433) | 6.616 (0.0101) |
| Jarque-Bera test: stat. (p-value in brackets) | 8.629 (0.00001) | 8.786 (0.000) | 4.167 (0.124) | 3.456 (0.063) | 3.879 (0.143) | 1.270 (0.529) | 4.556 (0.102) | 2.997 (0.102) | 3.613 (0.164) | 3.374 (0.185) |
| Ramsey Reset Test: stat. | 0.0104 (0.9174) | 0.205 (0.6506) | 0.175 (0.6754) | 0.785 (0.3755) | 0.040 (0.8408) | 2.307 (0.1288) | 0.248 (0.6178) | 0.043 (0.8349) | 0.039 (0.1279) | 0.529 (0.4667) |

Source: own computations in EViews

Note: *,**,*** suggest significance at 10%, 5%, 1% level respectively. p-value in brackets

The forecasting performance metrics in Table 15 show that the most accurate HICP forecasts using ARDL models are for Slovakia during 2025M1-2025M3.

**Table 15. The evaluation of HICP forecast accuracy by ARDL for CEE countries (2025M1-2025M3)**

| Country | RMSE | MSE | MAE | MAPE(%) |
|---|---|---|---|---|
| **BG** | 2.005 | 4.020 | 2.004 | 1.411 |
| **CZ** | 1.677 | 2.812 | 1.664 | 1.076 |
| **EE** | 2.332 | 5.438 | 2.234 | 1.399 |
| **HU** | 1.560 | 2.433 | 1.502 | 0.870 |
| **LV** | 1.417 | 2.007 | 1.220 | 0.819 |
| **LT** | 2.467 | 6.086 | 2.460 | 1.585 |
| **PL** | 0.888 | 0.788 | 0.887 | 0.581 |
| **RO** | 1.114 | 1.240 | 1.099 | 0.707 |
| **SK** | 0.328 | 0.107 | 0.304 | 0.235 |
| **SI** | 2.514 | 6.320 | 2.501 | 1.695 |

Source: own computations in EViews

## 4.5 Machine learning techniques for monthly data for CEE countries with ESI

We use the same grid search process to determine the best hyperparameter values for the machine learning methods used, specifically RF, SVM, and LSTM. These values are listed for all countries in Tables A8-A10. The search space was identical to the one previously shown in Table 1.

We ran forecasting models using optimal hyperparameter values and calculated their performance metrics. The results are shown in Tables 16-18.

**Table 16. The performance metrics for RF model under different settings**

| Setting | Metric | BG | RO | SK | SI | LT | LV | EE | PL | HU | CZ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Univariate setting** | MAPE Train | 0.16% | 0.13% | 0.18% | 0.44% | 0.34% | 0.46% | 0.18% | 0.11% | 0.33% | 0.17% |
| | MSE Train | 0.04 | 0.03 | 0.06 | 0.34 | 0.20 | 0.46 | 0.09 | 0.02 | 0.17 | 0.09 |
| | MAPE Test | 2.68% | 1.64% | 1.94% | 0.67% | 2.57% | 1.80% | 1.76% | 1.19% | 2.35% | 1.43% |
| | MSE Test | 14.73 | 6.90 | 8.42 | 1.05 | 16.44 | 8.20 | 8.59 | 3.48 | 16.90 | 493 |
| **HICP-Unemployment** | MAPE Train | 0.30% | 0.14% | 0.09% | 0.40% | 0.33% | 0.40% | 0.17% | 0.10% | 0.16% | 0.16% |
| | MSE Train | 0.15 | 0.03 | 0.02 | 0.30 | 0.20 | 0.30 | 0.08 | 0.02 | 0.06 | 0.09 |
| | MAPE Test | 2.58% | 2.19% | 1.96% | 1.34% | 2.68% | 1.77% | 1.85% | 1.44% | 2.40% | 1.49% |
| | MSE Test | 13.66 | 12.05 | 8.62 | 3.34 | 17.71 | 8.01 | 9.54 | 5.05 | 17.91 | 5.38 |
| **HICP-ESI** | MAPE Train | 0.16% | 0.16% | 0.11% | 0.42% | 0.33% | 0.54% | 0.42% | 0.26% | 0.32% | 0.17% |
| | MSE Train | 0.04 | 0.05 | 0.03 | 0.31 | 0.20 | 0.58 | 0.54 | 0.12 | 0.20 | 0.12 |
| | MAPE Test | 2.64% | 2.38% | 2.43% | 0.65% | 3.53% | 2.74% | 2.96% | 1.70% | 2.92% | 1.33% |
| | MSE Test | 14.34 | 14.72 | 13.23 | 1.02 | 30.50 | 18.22 | 22.46 | 6.87 | 26.24 | 4.26 |
| **HICP-Unemployment-ESI** | MAPE Train | 0.14% | 0.16% | 0.10% | 0.41% | 0.34% | 0.41% | 0.41% | 0.11% | 0.35% | 0.17% |
| | MSE Train | 0.03 | 0.05 | 0.03 | 0.36 | 0.21 | 0.27 | 0.31 | 0.03 | 0.35 | 0.13 |
| | MAPE Test | 2.67% | 2.83% | 2.15% | 1.07% | 3.71% | 2.29% | 2.35% | 1.82% | 2.93% | 1.48% |
| | MSE Test | 14.75 | 21.33 | 10.33 | 2.18 | 34.85 | 13.34 | 14.29 | 7.81 | 26.25 | 5.39 |

Source: own computations

## Table 17. The performance metrics for SVR model under different settings

| Setting | Metric | BG | RO | SK | SI | LT | LV | EE | PL | HU | CZ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Univariate setting** | MAPE Train | 0.39% | 0.32% | 0.24% | 0.41% | 0.43% | 0.43% | 0.46% | 0.32% | 0.42% | 0.45% |
| | MSE Train | 0.30 | 0.24 | 0.20 | 0.32 | 0.35 | 0.45 | 0.57 | 0.22 | 0.34 | 0.79 |
| | MAPE Test | 0.74% | 0.35% | 0.59% | 0.42% | 0.49% | 0.41% | 0.97% | 0.18% | 0.60% | 0.52% |
| | MSE Test | 1.93 | 0.38 | 1.67 | 0.34 | 1.62 | 0.40 | 2.50 | 0.12 | 1.56 | 1.33 |
| **HICP-Unemployment** | MAPE Train | 0.35% | 0.30% | 0.23% | 0.39% | 0.37% | 0.41% | 0.47% | 0.34% | 0.42% | 0.36% |
| | MSE Train | 0.25 | 0.20 | 0.20 | 0.31 | 0.34 | 0.43 | 0.62 | 0.19 | 0.34 | 0.76 |
| | MAPE Test | 0.58% | 0.31% | 0.61% | 0.34% | 0.40% | 0.39% | 0.52% | 0.23% | 0.61% | 0.51% |
| | MSE Test | 1.63 | 0.33 | 1.94 | 0.27 | 0.75 | 0.40 | 1.02 | 0.18 | 1.65 | 1.42 |
| **HICP-ESI** | MAPE Train | 0.34% | 0.26% | 0.33% | 0.41% | 0.37% | 0.45% | 0.44% | 0.32% | 0.35% | 0.36% |
| | MSE Train | 0.24 | 0.21 | 0.23 | 0.32 | 0.35 | 0.44 | 0.53 | 0.22 | 0.29 | 0.76 |
| | MAPE Test | 0.59% | 0.34% | 0.64% | 0.28% | 0.73% | 0.22% | 0.92% | 0.22% | 0.67% | 0.47% |
| | MSE Test | 1.88 | 0.39 | 1.58 | 0.25 | 2.39 | 0.15 | 2.23 | 0.14 | 1.55 | 1.25 |
| **HICP-Unemployment-ESI** | MAPE Train | 0.34% | 0.26% | 0.32% | 0.39% | 0.40% | 0.45% | 0.46% | 0.31% | 0.44% | 0.35% |
| | MSE Train | 0.24 | 0.20 | 0.22 | 0.31 | 0.30 | 0.43 | 0.52 | 0.19 | 0.36 | 0.76 |
| | MAPE Test | 0.59% | 0.34% | 0.64% | 0.34% | 0.68% | 0.27% | 0.97% | 0.25% | 0.64% | 0.49% |
| | MSE Test | 1.88 | 0.29 | 1.22 | 0.26 | 1.98 | 0.24 | 2.45 | 0.20 | 1.66 | 1.20 |

Source: own computations

**Table 18. The performance metrics for the LSTM model under different settings**

| Setting | Metric | BG | RO | SK | SI | LT | LV | EE | PL | HU | CZ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| **Univariate setting** | MAPE Train | 1.86% | 6.98% | 4.63% | 5.13% | 3.55% | 5.24% | 7.75% | 2.84% | 2.60% | 8.55% |
| | MSE Train | 5.26 | 74.36 | 22.30 | 34.33 | 15.68 | 48.71 | 60.94 | 9.47 | 12.25 | 104.85 |
| | MAPE Test | 6.64% | 0.58% | 0.89% | 1.05% | 0.31% | 0.43% | 2.31% | 0.79% | 0.22% | 2.99% |
| | MSE Test | 89.33 | 0.87 | 1.74 | 1.91 | 0.27 | 0.58 | 14.13 | 1.47 | 0.27 | 21.53 |
| **HICP-Unemployment** | MAPE Train | 1.50% | 6.40% | 4.00 | 4.60 | 3.10 | 4.70 | 7.00 | 2.50 | 2.20 | 7.90 |
| | MSE Train | 4.20 | 61.17 | 18.00 | 28.00 | 12.50 | 41.00 | 53.00 | 7.60 | 10.00 | 90.00 |
| | MAPE Test | 5.80 | 0.54% | 0.85% | 1.16% | 0.33% | 0.42% | 2.51% | 0.75 | 0.22% | 2.52% |
| | MSE Test | 71.20 | 0.85 | 1.60 | 1.70 | 0.25 | 0.50 | 11.00 | 1.30 | 0.22 | 17.00 |
| **HICP-ESI** | MAPE Train | 1.40% | 0.85% | 7.1% | 3.09% | 3.16% | 3.08% | 2.97% | 3.19% | 3.2% | 2.94% |
| | MSE Train | 2.81 | 1.13 | 53.69 | 19.12 | 23.88 | 15.45 | 12.32 | 25.47 | 24.76 | 21.11 |
| | MAPE Test | 2.55% | 0.10% | 6.43% | 3.16% | 3.22% | 3.21% | 3.06% | 3.18% | 3.22% | 3.50% |
| | MSE Test | 14.23 | 0.04 | 70.23 | 34.82 | 35.2 | 35.13 | 31.99 | 34.92 | 35.2 | 29.28 |
| **HICP-Unemployment-ESI** | MAPE Train | 1.21% | 0.73% | 6.50% | 0.95% | 1.54% | 2.03% | 2.82% | 1.01% | 1.82% | 3.85% |
| | MSE Train | 2.23 | 0.90 | 50.01 | 2.54 | 3.22 | 5.40 | 6.56 | 1.30 | 2.21 | 12.32 |
| | MAPE Test | 3.50% | 0.31% | 5.55% | 0.95% | 0.45% | 0.92% | 4.21% | 0.80% | 0.93% | 4.50% |
| | MSE Test | 23.21 | 0.44 | 45.03% | 1.75 | 0.32 | 0.91 | 16.57 | 1.21% | 1.04% | 25.25 |

Source: own computations

In the experiments reported, modern ML methods - particularly the LSTM network and SVR - offer clear, practically meaningful improvements over the ARDL baselines in many country × feature-set combinations. Overall, the LSTM often achieves the lowest out-of-sample error in the univariate designs and several multivariate configurations, while SVR frequently matches or slightly exceeds ARDL when unemployment or sentiment are used as predictors. RFs, once properly tuned, also provide useful gains in various settings, especially when additional covariates are available. In summary, Tables 16-18 demonstrate that ML methods are not

only competitive but also consistently enhance short-horizon HICP forecasting accuracy in our panel.

Two facts in the results make the ML advantage particularly credible. First, the monthly experiments use longer training sequences (nearly two decades), which gives data-hungry models like LSTM the sample size they need to learn nonlinear lag structures - the observed reductions in test error therefore reflect genuine pattern learning rather than chance. Second, improvements are not limited to in-sample fit: in many country-setting combinations, the test MAPEs are lower (or only modestly higher) than training MAPEs for ML methods when unemployment is included, indicating stable generalization rather than pure overfitting. In the HICP-Unemployment configuration, SVR and LSTM both show consistent test MAPE reductions relative to ARDL (SVR test MAPE often ≈ 0.20–0.6%, LSTM often ≈ 0.5–1.5%), and RF shows meaningful gains after tuning in several countries as well (test MAPEs commonly falling into the low single digits).

In summary, the strong agreement between Romania's quarterly results and the full CEE monthly LSTM outcomes, along with the gap compared to ARDL benchmarks, confirms that these deep-learning models reliably generalize across different frequencies and economies, providing a clear forecasting benefit for CEE countries.

## 5. Conclusions

This study provides a valuable contribution to empirical forecasting. It shows the effectiveness of using LSTM networks for inflation forecasting, consistent with traditional economic theory. Additionally, including sentiment analysis results in ARDL models can improve forecast accuracy. Overall, these findings indicate that multivariate approaches, especially those incorporating sentiment indices and unemployment rates, significantly enhance forecast precision. The use of *TimeSeriesSplit* for cross-validation maintains the temporal integrity of our models, offering reliable performance assessments. These results suggest that advanced machine learning techniques, combined with comprehensive data inputs, can greatly improve economic forecasting capabilities.

Despite the importance of these empirical findings, the research has several limitations that highlight areas for future study. First, a key limitation of this paper is its focus on a short forecasting horizon - covering four quarters with high inflation in Romania due to the international and domestic economic context, and three months for CEE countries. The study aimed to analyze short-term periods of high inflation under extreme economic conditions. This situation occurs over brief periods because of interventions by the central bank. Specifically, the Na-

tional Bank of Romania managed inflation by gradually raising interest rates, making the high inflation episode temporary. Therefore, applying the proposed forecasting methods over a short horizon is justified, and they could be useful in similar future scenarios. From this perspective, the paper deserves praise for employing quantitative methods that are well-suited to the economic context of the countries analyzed. The short-term forecasting horizon is closely related to the longer-term transmission of monetary policy. This process is complex and involves a time lag, which affects how short-term interest rates influence economic growth and inflation. The actual transmission occurs through various channels, including, besides the interest rate channel, the exchange rate channel and the asset price channel, all of which play important roles. Changes in the exchange rate and asset prices immediately impact the inflation rate once the transmission cycle ends. This process typically spans about 12-18 months, which is much longer than the forecasting horizon considered in this paper.

Second, the study is limited to CEE countries because they experienced the highest inflation in the EU, and it was conducted within a limited time frame. We considered limited computing resources when choosing the number of countries to include. Nevertheless, optimizing the LSTM took more than a week of computing time on a computer with an Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz processor, 16 cores, and 192 GB of RAM.

Third, the methodology is limited to a few methods and economic variables. Besides the economic variables mentioned earlier, there are influences that are less certain and hard to quantify but also play a significant role. This could include the structure of the consumer basket, which, according to the harmonized index of consumer prices, changes in EU countries roughly every five years. Since inflation rates can be accelerated by various influences within months, reach a peak, and then gradually decline, the structure of the consumer basket may already seem inappropriate and outdated. It is also important to consider political influences like the political cycle, natural factors, and others. These limitations, by their nature, could not and have not been addressed in the paper.

Fourth, the machine learning techniques are usually employed on long time series, but in this case, the series are quite short due to limited data availability in inflation reports, which encourages overfitting. Therefore, future studies should consider other types of econometric models and machine learning techniques, longer time series, and more explanatory variables to predict inflation (such as various types of interest rates used by central banks to control inflation). Additionally, future research could also include more countries with similar characteristics and a longer forecasting horizon.

**Funding:** Bogdan Oancea gratefully acknowledges funding from the MRID, project PNRR-I8 no 842027778, contract no 760096. Mihaela Simionescu gratefully acknowledges funding from the Academy of Romanian Scientists, in the "AOȘR-TEAMS-III" Project Competition EDITION 2024-2025, project name "Improving forecasts inflation rate in Romania using sentiment analysis and machine learning".

**Data Availability Statement:** Data are freely available: 10.13140/RG.2.2.14225.95846

**Conflicts of Interest:** The authors declare no conflicts of interest.

## References

Abadi, M.; et al. (2016). TensorFlow: A System for Large-Scale Machine Learning. In *Proc. 12th USENIX Symp. Oper. Syst. Des. Implement.*; pp 265–283. https://www.usenix.org/system/files/conference/osdi16/osdi16-abadi.pdf

Ahsan, M. D. Manjurul; Mahmud, M. A. Parvez; Saha, P. K.; Gupta, K. D.; Siddique, Z. (2021). Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. *Technologies, 9*(3), 52. https://doi.org/10.3390/technologies9030052.

Almosova, A., Andresen, N. (2023). *Nonlinear inflation forecasting with recurrent neural networks*. J. Forecasting, 42(3), 240–259. https://doi.org/10.1002/for.2901

Bahdanau, D.; Cho, K.; Bengio, Y. (2015). Neural Machine Translation by Jointly Learning to Align and Translate. In *Proc. 3rd Int. Conf. Learn. Representations (ICLR)*. https://arxiv.org/pdf/1409.0473

Bańbura, M., Bobeica, E. (2020). *Does the Phillips curve help to forecast euro area inflation?* ECB Working Paper Series, No 2471. European Central Bank. https://www.ecb.europa.eu/pub/pdf/scpwps/ecb.wp2471~fc87caada8.en.pdf

Basak, D.; Pal, S.; Patranabis, D. C. (2007). Support Vector Regression. *Neural Inf. Process. Lett. Rev.*, *11*(10), 203–224. https://www.researchgate.net/publication/228537532_Support_Vector_Regression

Beck, E., Wolf, M. (2025). *Forecasting Inflation with the Hedged Random Forest*. Policy Brief No. 1250, SUERF – The European Money and Finance Forum. https://www.suerf.org/wp-content/uploads/2025/09/SUERF-Policy-Brief-1250-_Beck.Wolf_.pdf

Benett, J., Owyang, M. (2022). *On the Relative Performance of Inflation Forecasts*. Federal Reserve Bank of St. Louis. Staff Rep. No. 214/2022. https://doi.org/10.20955/r.104.131-48

Boaretto, G., Medeiros, M. C. (2023). *Forecasting inflation using disaggregates and machine learning,* arXiv preprint arXiv:2308.11173. https://doi.org/10.48550/arXiv.2308.11173.

Breiman, L. (2001). Random Forests. *Machine Learning*, *45*(1), 5–32. https://doi.org/10.1023/A:1010933404324

Biau, G., Scornet, E. (2016). Random Forest Guided Tour. *TEST*, 25, 197–227. https://doi.org/10.1007/s11749-016-0481-725 (2).

Chalmovianský, J., Porqueddu, M., Sokol, A. (2020). *Weigh(t)ing the basket: aggregate and component-based inflation forecasts for the euro area.* ECB Working Paper Series, No 2501, European Central Bank. https://www.ecb.europa.eu/pub/pdf/scpwps/ecb.wp2501~8797484f4b.en.pdf

Chollet, F. (2015). *Keras: The Python Deep Learning Library.* [Retrieved 2024-10-01]Available at: https://keras.io.

Cissoko, M. B. H., Castelain, V., Lachiche, N. (2025). Multi-Way Adaptive Time Aware LSTM for Irregularly Collected Sequential ICU Data. *Expert Syst. Appl. 261*, 125548. https://doi.org/10.1016/j.eswa.2024.125548.

Clements, M.P., Reade, J.J., 2020. Forecasting and forecast narratives: The Bank of England inflation reports. *International Journal of Forecasting*, 36(4), 1488-1500. https://doi.org/10.1016/j.ijforecast.2019.08.013

Cutler, D. R.; Edwards, T. C., Jr.; Beard, K. H.; Cutler, A.; Hess, K. T.; Gibson, J.; Lawler, J. J. (2007). Random Forests for Classification in Ecology. *Ecology*, *88*(11), 2783–2792.

https://doi.org/10.1890/07-0539.1

Dang, P. N., Dang, T. V. D., Nguyen, T. M. T., Vu, Q. K., Nguyen, A. H. (2025). *Sentiment index as a predictor of CPI: A lexicon-based approach using economic news data in Vietnam*. *Journal of Open Innovation: Technology, Market, and Complexity, 11*(3), 100620. https://doi.org/10.1016/j.joitmc.2025.100620.

Drucker, H.; Burges, C. J. C.; Kaufman, L.; Smola, A.; Vapnik, V. (1997). Support Vector Regression Machines. In *Adv. Neural Inf. Process. Syst., 9,* 155–161. https://proceedings.neurips.cc/paper_files/paper/1996/file/d38901788c533e8286cb6400b40b386d-Paper.pdf

Eugster, P., Uhl, M. (2024). Forecasting Inflation Using Sentiment. *Economics Letters.* Vol. 236(3), 111575. https://doi.org/10.1016/j.econlet.2024.111575

Faust, J., Wright, J. (2013). Forecasting Inflation. *Handbook of Economic Forecasting.* 1(2), p. 2-56. https://doi.org/10.1016/B978-0-444-53683-9.00001-3

Fulton, Ch., Hubrich, K. (2021). Forecasting Real Inflation in Real Time. *Econometrics.* 9 (4), 36. https://doi.org/10.3390/econometrics9040036

Gers, F. A., Schmidhuber, J., Cummins, F. (2000). Learning to Forget: Continual Prediction with LSTM. *Neural Comput. 12*(10), 2451–2471. https://doi.org/10.1162/089976600300015015.

Graves, A., Liwicki, M., Fernandez, S., Bertolami, R., Bunke, H., Schmidhuber, J. (2009). A Novel Connectionist System for Unconstrained Handwriting Recognition. *IEEE Trans. Pattern Anal. Mach. Intell. 31*(5), 855–868. https://doi.org/10.1109/TPAMI.2008.137.

Groen, J., Paap, R., Ravazzolo, F. (2012). *Real Time Inflation Forecasting in Changing World*. Federal Reserve Bank of New York. Staff Rep. No. 388. https://resources.newyorkfed.org/medialibrary/media/research/staff_reports/sr388.pdf

Grothe M., Meyler, A. (2015). *Inflation Forecasts: Are Marked Based and Survey Based Measures Informative?* ECB: Working Paper Series. 1865(11). https://www.ecb.europa.eu/pub/pdf/scpwps/ecbwp1865.en.pdf

Han, J., Kamber, M., Pei, J. (2011). *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann: Burlington, MA.

Hochreiter, S., Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Comput. 9*(8), 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hong, Y., Jiang, F., Meng, L., Xue, B. (2024). Forecasting Inflation Using Economic Narratives. *Journal of Business and Economic Statistics.* 43(1), 216-231. https://doi.org/10.1080/07350015.2024.2347619

Hoque, K. E., Aljamaan, H. (2021). Impact of Hyperparameter Tuning on Machine Learning Models in Stock Price Forecasting. *IEEE Access, 9*, 163815–163830. https://doi.org/10.1109/ACCESS.2021.3134138.

Hyndman, R. J., Koehler, A. B. (2006). Another Look at Measures of Forecast Accuracy. *Int. J. Forecast. 22*(4), 679–688. https://doi.org/10.1016/j.ijforecast.2006.03.001

Hyndman, R. J., Athanasopoulos, G. (2018). *Forecasting: Principles and Practice*; OTexts.

Ito, K.; Nakano, R. (2003). Optimizing Support Vector Regression Hyperparameters Based on Cross-Validation. In *Proc. Int. Joint Conf. Neural Networks*; Portland, OR, USA, Vol. 3, pp 2077–2082. https://doi.org/10.1109/IJCNN.2003.1223728.

James, G., Witten, D., Hastie, T.; Tibshirani, R. (2013). *An Introduction to Statistical Learning*; Springer Science & Business Media: New York.

Kumuda, A., Panigrahy, S. K. (2025). Design of an Efficient Model for Psychological Disease Analysis and Prediction Using Machine Learning and Genomic Data Samples. *Big Data Cogn. Comput. 9*(3), 49. https://doi.org/10.3390/bdcc9030049.

Lenza, M., Moutachaker, I., Paredes, J. (2023). *Density forecasts of inflation: A quantile regression forest approach*. ECB Working Paper No. 2830. European Central Bank. https://www.ecb.europa.eu/pub/pdf/scpwps/ecb.wp2830~81049ee58f.en.pdf

Lin, E., Kuo, P.-H., Lin, W.-Y., Liu, Y.-L., Yang, A. C.; Tsai, S. J. (2021). Prediction of Probable Major Depressive Disorder in the Taiwan Biobank: An Integrated Machine Learning and Genome-Wide Analysis Approach. *J. Pers. Med. 11*(7), 597. https://doi.org/10.3390/jpm11070597.

Liu, Y., Pan, R., Xu, R. (2024). *Mending the Crystal Ball: Enhanced Inflation Forecasts with Machine Learning.* International Monetary Fund. WP/24/206, (9).

https://doi.org/10.5089/9798400285387.001

Medeiros, M., Vasconcelos, G., Veiga, A., Zilberman, E. (2021). Forecasting Inflation in a Data Rich Environment: The Benefits of Machine Learning Methods. *Journal of Business & Economic Statistics*, 39(1), 98-119. https://doi.org/10.1080/07350015.2019.1637745

Mienye, I. D., Swart, T. G.; Obaido, G. (2024). Recurrent Neural Networks: A Comprehensive Review of Architectures, Variants, and Applications. *Information 15*(9), 517. https://doi.org/10.3390/info15090517.

Ngwaba, C. A. (2025). Forecasting Covered Call Exchange-Traded Funds (ETFs) Using Time Series, Machine Learning, and Deep Learning Models. *J. Risk Financ. Manag. 18*(3), 120. https://doi.org/10.3390/jrfm18030120.

Nitesh, K., Abhiram, Y., Teja, R. K., Kavitha, S. (2023). Weather Prediction Using Long Short Term Memory (LSTM) Model. In *Proc. 2023 5th Int. Conf. Smart Syst. Invent. Technol. (ICSSIT)*; Tirunelveli, India, pp 1–6. https://doi.org/10.1109/ICSSIT55814.2023.10061039.

Panzaru, C., Belea, S., & Jianu, L. (2025). Delayed Taxation and Macroeconomic Stability: A Dynamic IS–LM Model with Memory Effects. *Economies*, 13(7), 208. https://doi.org/10.3390/economies13070208.

Paranhos, L. (2025). Predicting inflation with recurrent neural networks. *Intl. J. of Economic Forecasting*, in press. https://doi.org/10.1016/j.ijforecast.2024.07.010.

Park, S., Yang, J. S. (2022). Interpretable Deep Learning LSTM Model for Intelligent Economic Decision-Making. *Knowledge-Based Systems, 248,* 108907. https://doi.org/10.1016/j.knosys.2022.108907.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E. (2011). Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res. 12,* 2825–2830.

https://www.jmlr.org/papers/volume12/pedregosa11a/pedregosa11a.pdf

Petropoulos, F., Apiletti, D., Assimakopoulos, V., Babai, M. Z., Barrow, D. K., Taieb, S. B., Ziel, F. (2022). Forecasting: Theory and Practice. *Int. J. Forecast. 38*(3), 705–871. https://doi.org/10.1016/j.ijforecast.2021.11.001

Raza, A., Shahid, M. A., Zaman, M., Miao, Y., Huang, Y., Safdar, M., Maqbool, S., Muhammad, N. E. (2025). Improving Wheat Yield Prediction with Multi-Source Remote Sensing Data and Machine Learning in Arid Regions. *Remote Sensing 17*(5), 774. https://doi.org/10.3390/rs17050774.

Rygh, T., Vaage, C., Westgaard, S., de Lange, P. E. (2025). Inflation Forecasting: LSTM Networks vs. Traditional Models for Accurate Predictions. *J. Risk Financial Manag*., 18(7), 365. https://doi.org/10.3390/jrfm18070365

Saâdaoui, F., Rabbouch, H. (2024). Financial Forecasting Improvement with LSTM-ARFIMA Hybrid Models and Non-Gaussian Distributions. *Technol. Forecast. Soc. Change, 206*, 123539. https://doi.org/10.1016/j.techfore.2024.123539.

Salton, G. D.; Kelleher, J. D. (2019). Persistence Pays Off: Paying Attention to What the LSTM Gating Mechanism Persists. In *Proc. Recent Adv. Nat. Lang. Process. (RANLP 2019)*; Varna, Bulgaria, pp 1052–1059. https://doi.org/10.26615/978-954-452-056-4_121.

Sevgin, F. (2025). Machine Learning-Based Temperature Forecasting for Sustainable Climate Change Adaptation and Mitigation. *Sustainability, 17*(5), 1812. https://doi.org/10.3390/su17051812.

Smola, A. J.; Schölkopf, B. (2004). A Tutorial on Support Vector Regression. *Stat. Comput.,14*, 199–222. https://doi.org/10.1023/B:STCO.0000035301.49549.88.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R. (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *J. Mach. Learn. Res., 15*(1), 1929–1958. http://jmlr.org/papers/v15/srivastava14a.html

Stock, J., M. Watson. (1999). Forecasting Inflation. *Journal of Monetary Economics*. 44(2), p. 293-335. https://doi.org/10.1016/S0304-3932(99)00027-6

Stock, J., M. Watson. (2003). Forecasting Output and Inflation: the Role of Assets Prices. *Journal of Economic Literature.* 41(3), p. 788-829. https://www.aeaweb.org/articles/pdf/doi/10.1257/002205103322436197

Strobl, C., Boulesteix, A. L., Zeileis, A., Hothorn, T. (2009). Bias in Random Forest Variable Importance Measures: Illustrations, Sources and a Solution. *BMC Bioinformatics*, *10* (1), 213. https://doi.org/10.1186/1471-2105-8-25

Sutskever, I., Vinyals, O., Le, Q. V. (2014). Sequence to Sequence Learning with Neural Networks. In *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*; MIT Press, pp. 3104–3112. https://proceedings.neurips.cc/paper_files/paper/2014/file/5a18e133cbf9f257297f410bb7eca942-Paper.pdf

Svoboda, J., Štych, P., Laštovička, J., Paluba, D., Kobliuk, N. (2022). Random Forest Classification of Land Use, Land-Use Change and Forestry (LULUCF) Using Sentinel-2 Data—A Case Study of Czechia. *Remote Sensing, 14*(5), 1189. https://doi.org/10.3390/rs14051189.

Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J.; Wojna, Z. (2016). Rethinking the Inception Architecture for Computer Vision. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, pp. 2818–2826. https://doi.org/ 10.1109/CVPR.2016.308.

Vapnik, V. (1995). *The Nature of Statistical Learning Theory*; Springer: New York.

Vicente, J. (2005). *Forecasting inflation through a bottom-up approach: The Portuguese case*. Banco de Portugal Working Paper No. 5-05. https://www.bportugal.pt/sites/default/files/anexos/papers/wp200502.pdf

Wang, H., Liu, M. (2025). Credit Risk Assessment of Green Supply Chain Finance for SMEs Based on Multi-Source Information Fusion. *Sustainability*, *17*(4), 1590. https://doi.org/10.3390/su17041590.

Zaremba, W., Sutskever, I., Vinyals, O. (2014). Recurrent Neural Network Regularization. In *Proc. 2014 Conf. Empir. Methods Nat. Lang. Process. (EMNLP)*; Association for Computational Linguistics, pp 1524–1534. https://doi.org/10.48550/arXiv.1409.2329

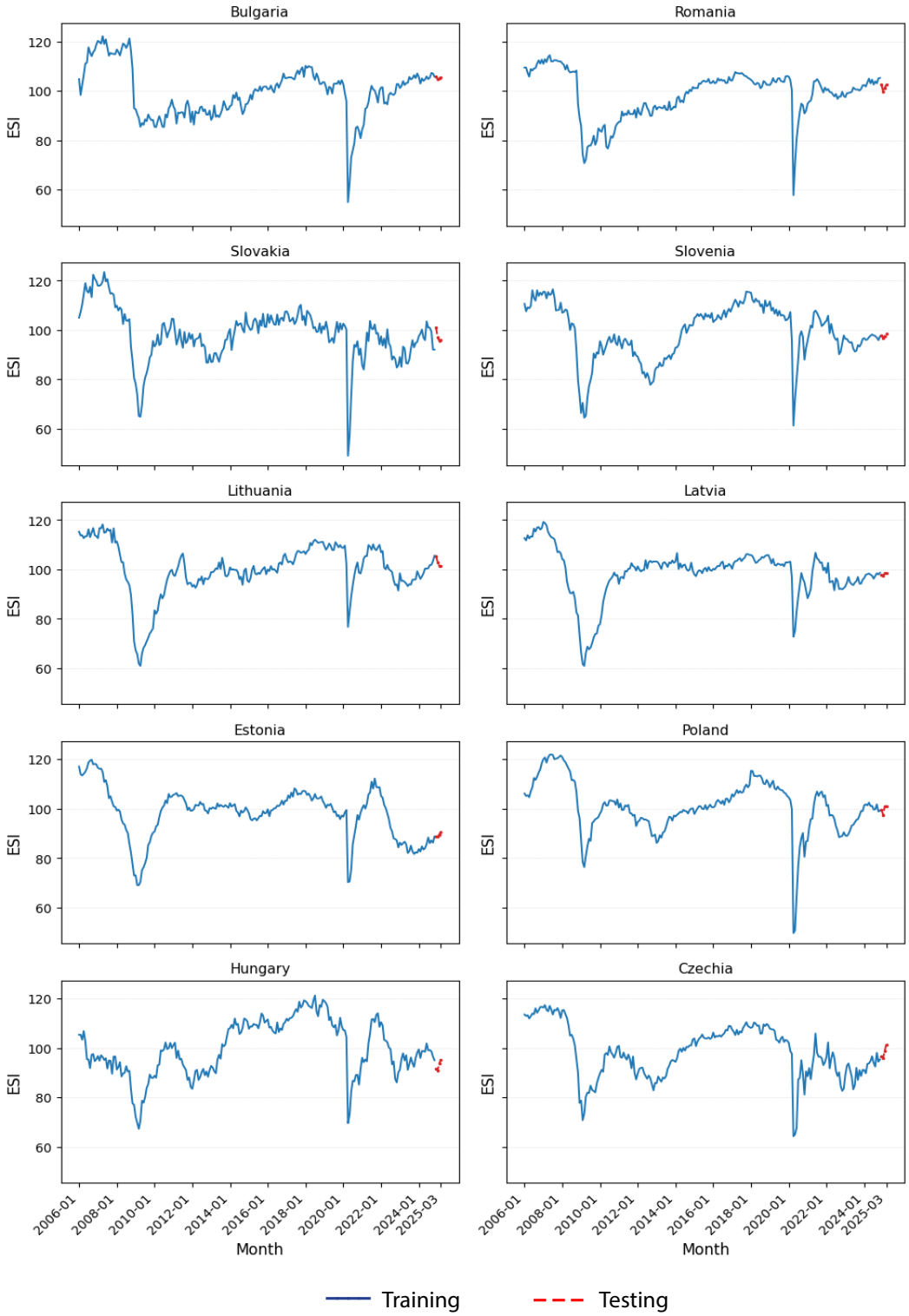
Zhang, L., Li, J. (2012). Inflation Forecasting Using Support Vector Regression, *Fourth International Symposium on Information Science and Engineering*, Shanghai, China, 2012, pp. 136-140. https://doi.org/ 10.1109/ISISE.2012.37

Zhou, J.; Li, C.; Kim, Y. K.; Park, S. (2025). Bioinformatics and Deep Learning Approach to Discover Food-Derived Active Ingredients for Alzheimer's Disease Therapy. *Foods*, *14*(1), 127. https://doi.org/10.3390/foods14010127.

Zhu, P. Zhou, O., Zhang, Y. (2024). Investor Attention and Consumer Price Index Inflation rate: an Evidence from the United States. *Human and Social Studies Communications,* 11 (541). https://doi.org/10.1057/s41599-024-03036-y

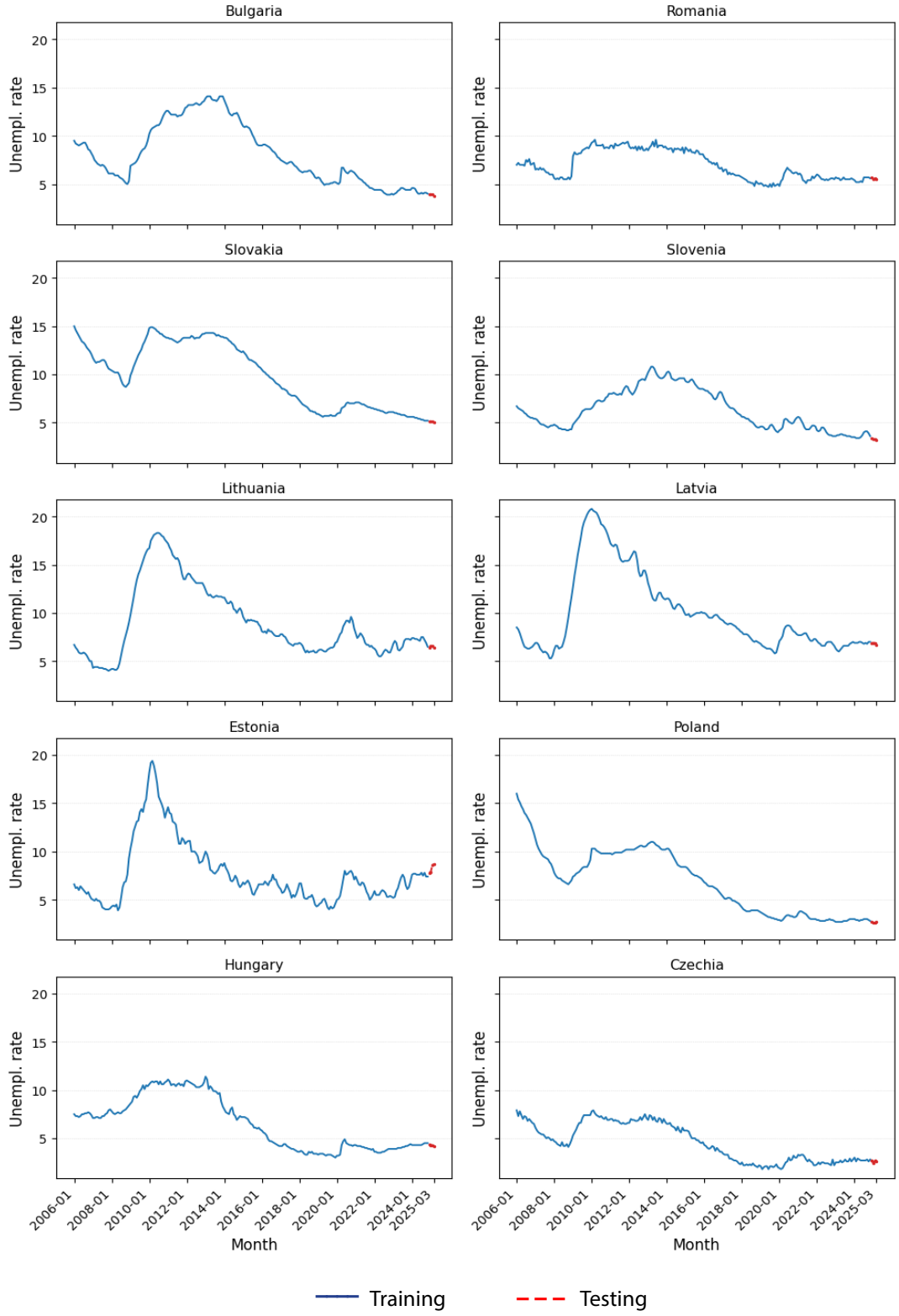
# Appendix A

## Figure A1. ESI for CEE countries



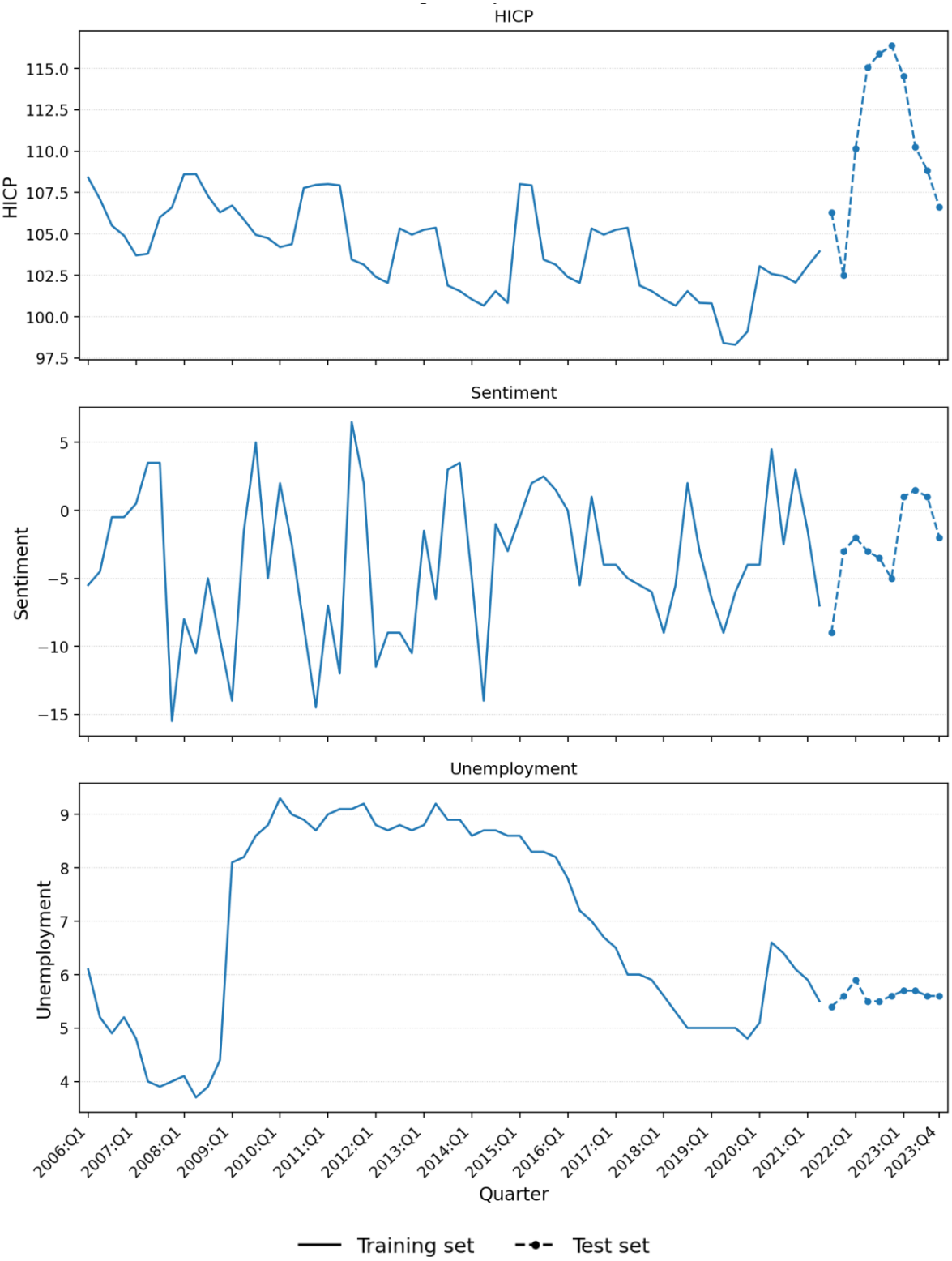

Source: own computations

## Figure A2. Unemployment rates (%) for CEE countries




Source: own computations

**Figure A3. HICP (index, 2015=100), Sentiment index, and Unemployment (%) for Romania – quarterly data**




Source: own computations

## Table A1. Descriptive statistics for quarterly Unemployment rates and HICP (2015=100) data series for Romania

| Descriptive statistics | Unemployment (%) | HICP |
|---|---|---|
| **Average** | 6.68 | 104.92 |
| **Lowest value** | 3.7 | 98.3 |
| **Median** | 6.05 | 104.82 |
| **Highest value** | 9.3 | 116.37 |
| **Standard dev.** | 1.77 | 3.78 |
| **Jarque-Bera stat** | 6.93 | 17.03 |

Source: own computations

## Table A2. CEE countries

| Country | Abbreviation |
|---|---|
| **Bulgaria** | BG |
| **Romania** | RO |
| **Slovakia** | SK |
| **Slovenia** | SI |
| **Lithuania** | LT |
| **Latvia** | LV |
| **Estonia** | EE |
| **Poland** | PL |
| **Hungary** | HU |
| **Czechia** | CZ |

## Table A3. Descriptive statistics for monthly data for ESI for CEE countries

| | BG | RO | SK | SI | LT | LV | EE | PL | HU | CZ |
|---|---|---|---|---|---|---|---|---|---|---|
| **Average** | 99.70 | 98.66 | 98.52 | 99.14 | 100.11 | 98.60 | 98.07 | 100.94 | 99.60 | 98.37 |
| **Lowest value** | 54.9 | 57.7 | 49.1 | 61.3 | 60.9 | 60.9 | 69 | 49.6 | 67.2 | 64.2 |
| **Median** | 100.6 | 101.2 | 99.4 | 99.5 | 100.3 | 100.2 | 100.1 | 100.9 | 98.7 | 98 |
| **Highest value** | 122.1 | 114.4 | 123.5 | 116.5 | 118.2 | 119.2 | 119.8 | 121.9 | 121.2 | 117.3 |
| **Standard dev.** | 10.30 | 9.52 | 10.51 | 10.99 | 10.67 | 10.11 | 10.23 | 10.61 | 11.11 | 10.53 |
| **Jarque-Bera stat** | 22.34 | 64.02 | 125.98 | 28.40 | 119.32 | 151.36 | 15.73 | 184.76 | 3.68 | 5.17 |

Source: own computations

## Table A4. Descriptive statistics for monthly data for HICP (2015=100) for CEE countries

| | BG | RO | SK | SI | LT | LV | EE | PL | HU | CZ |
|---|---|---|---|---|---|---|---|---|---|---|
| **Average** | 103.30 | 101.92 | 101.78 | 104.07 | 105.65 | 104.32 | 104.56 | 104.46 | 105.91 | 105.82 |
| **Lowest value** | 73.81 | 68.12 | 81.44 | 84.35 | 73.29 | 69.88 | 70.73 | 80.40 | 69.36 | 82.70 |
| **Median** | 100.99 | 99.38 | 100.55 | 100.28 | 100.69 | 99.90 | 100.41 | 100.50 | 100.25 | 100 |
| **Highest value** | 138.99 | 152.96 | 128.98 | 145.18 | 151.95 | 146.54 | 157.20 | 150.90 | 169.30 | 152.90 |
| **Standard dev.** | 14.85 | 20.78 | 11.40 | 15.64 | 20.34 | 18.41 | 21.80 | 17.92 | 24.77 | 18.66 |
| **Jarque-Bera stat** | 25.18 | 17.62 | 25.90 | 70.80 | 36.75 | 28.88 | 36.02 | 62.26 | 58.13 | 73.42 |

Source: own computations

## Table A5. Descriptive statistics for monthly data for unemployment rates (%) for CEE countries

| | BG | RO | SK | SI | LT | LV | EE | PL | HU | CZ |
|---|---|---|---|---|---|---|---|---|---|---|
| **Average** | 8.03 | 6.92 | 9.82 | 6.24 | 8.96 | 9.94 | 7.68 | 6.78 | 6.51 | 4.55 |
| **Lowest value** | 3.8 | 4.7 | 5 | 3.2 | 4 | 5.3 | 3.9 | 2.6 | 3 | 1.8 |
| **Median** | 7.1 | 6.5 | 10 | 5.6 | 7.6 | 8.3 | 6.7 | 6.9 | 6.1 | 4.3 |
| **Highest value** | 14.1 | 9.6 | 15 | 10.8 | 18.3 | 20.8 | 19.4 | 16 | 11.4 | 7.9 |
| **Standard dev.** | 3.22 | 1.50 | 3.34 | 2.12 | 3.75 | 4.12 | 3.27 | 3.38 | 2.70 | 1.97 |
| **Jarque-Bera stat** | 20.70 | 24.85 | 24.25 | 19.89 | 36.29 | 55.01 | 140.96 | 12.71 | 23.03 | 24.82 |

Source: own computations

## Table A6. DM and the Wilcoxon tests for ARDL/RF/SVR/LSTM versus naïve forecast

| Scenario | Horizon | Method | Mean(d) | DM stat | DM p (one_sided) | Wilcoxon p (one_sided) |
|---|---|---|---|---|---|---|
| **Baseline linear** | 1 | ARDL | 0.0029 | 0.2399 | 0.5947 | 0.6011 |
| | | RF | 0.0543 | 3.1106 | 0.9989 | 0.9992 |
| | | SVR | 0.0162 | 1.2308 | 0.8901 | 0.8872 |
| | | LSTM | 0.0932 | 3.4888 | 0.9997 | 0.9985 |
| | 3 | ARDL | -0.4607 | -6.6451 | 0.0000 | 0.0000 |
| | | RF | -0.4080 | -6.4946 | 0.0000 | 0.0000 |
| | | SVR | -0.4658 | -6.9860 | 0.0000 | 0.0000 |
| | | LSTM | -0.3966 | -5.5153 | 0.0000 | 0.0000 |
| | 6 | ARDL | -1.2506 | -7.3427 | 0.0000 | 0.0000 |
| | | RF | -1.1448 | -6.9299 | 0.0000 | 0.0000 |
| | | SVR | -1.2404 | -7.3811 | 0.0000 | 0.0000 |
| | | LSTM | -1.1396 | -6.8622 | 0.0000 | 0.0000 |
| **High persistence** | 1 | ARDL | 0.0014 | 0.1237 | 0.5492 | 0.5398 |
| | | RF | 0.1994 | 5.6820 | 1.0000 | 1.0000 |
| | | SVR | 0.2319 | 5.9683 | 1.0000 | 1.0000 |
| | | LSTM | 0.5050 | 7.7522 | 1.0000 | 1.0000 |
| | 3 | ARDL | -0.5450 | -6.8334 | 0.0000 | 0.0000 |
| | | RF | 0.0170 | 0.2337 | 0.5923 | 0.1434 |
| | | SVR | -0.3223 | -5.9496 | 0.0000 | 0.0000 |
| | | LSTM | -0.0302 | -0.3245 | 0.3730 | 0.4442 |
| | 6 | ARDL | -1.4910 | -7.5974 | 0.0000 | 0.0000 |
| | | RF | -0.5193 | -3.5409 | 0.0002 | 0.0002 |
| | | SVR | -1.1520 | -6.8669 | 0.0000 | 0.0000 |
| | | LSTM | -0.9459 | -4.7525 | 0.0000 | 0.0000 |
| **MA noninvertible** | 1 | ARDL | -0.1819 | -3.3872 | 0.0004 | 0.0000 |
| | | RF | -0.1701 | -3.1801 | 0.0009 | 0.0000 |
| | | SVR | -0.1872 | -3.6338 | 0.0002 | 0.0001 |
| | | LSTM | 0.1831 | 2.0641 | 0.9798 | 0.9243 |
| | 3 | ARDL | -0.3499 | -5.7643 | 0.0000 | 0.0000 |
| | | RF | -0.2688 | -4.3489 | 0.0000 | 0.0000 |
| | | SVR | -0.3659 | -6.1720 | 0.0000 | 0.0000 |
| | | LSTM | -0.0249 | -0.3079 | 0.3792 | 0.2693 |
| | 6 | ARDL | -0.4799 | -4.7069 | 0.0000 | 0.0000 |
| | | RF | -0.5080 | -5.4918 | 0.0000 | 0.0000 |
| | | SVR | -0.5377 | -5.5876 | 0.0000 | 0.0000 |
| | | LSTM | -0.1384 | -1.2111 | 0.1136 | 0.2381 |

| Scenario | Horizon | Method | Mean(d) | DM stat | DM p (one_sided) | Wilcoxon p (one_sided) |
|---|---|---|---|---|---|---|
| **Non linear regime** | 1 | ARDL | -0.1054 | -3.4225 | 0.0004 | 0.0004 |
| | | RF | 0.0141 | 0.3733 | 0.6453 | 0.2139 |
| | | SVR | -0.1148 | -3.4592 | 0.0003 | 0.0001 |
| | | LSTM | 0.2521 | 2.5389 | 0.9941 | 0.9261 |
| | 3 | ARDL | -1.1974 | -7.8564 | 0.0000 | 0.0000 |
| | | RF | -1.0601 | -7.3643 | 0.0000 | 0.0000 |
| | | SVR | -1.1944 | -7.8352 | 0.0000 | 0.0000 |
| | | LSTM | -0.9132 | -5.0088 | 0.0000 | 0.0000 |
| | 6 | ARDL | -2.8885 | -8.3053 | 0.0000 | 0.0000 |
| | | RF | -2.7508 | -8.0069 | 0.0000 | 0.0000 |
| | | SVR | -2.8445 | -8.1756 | 0.0000 | 0.0000 |
| | | LSTM | -2.4743 | -6.6978 | 0.0000 | 0.0000 |
| **Structural break** | 1 | ARDL | 0.0097 | 0.7550 | 0.7744 | 0.7319 |
| | | RF | 0.0625 | 2.9599 | 0.9983 | 0.9837 |
| | | SVR | 0.0190 | 1.3919 | 0.9172 | 0.9489 |
| | | LSTM | 0.0809 | 3.0657 | 0.9988 | 0.9814 |
| | 3 | ARDL | -0.4556 | -6.4634 | 0.0000 | 0.0000 |
| | | RF | -0.4116 | -6.2130 | 0.0000 | 0.0000 |
| | | SVR | -0.4630 | -6.7251 | 0.0000 | 0.0000 |
| | | LSTM | -0.3803 | -5.1908 | 0.0000 | 0.0000 |
| | 6 | ARDL | -1.2531 | -7.3477 | 0.0000 | 0.0000 |
| | | RF | -1.1257 | -6.7087 | 0.0000 | 0.0000 |
| | | SVR | -1.2411 | -7.3415 | 0.0000 | 0.0000 |
| | | LSTM | -1.0934 | -6.3340 | 0.0000 | 0.0000 |

Source: own computations

## Table A7. Win rates for different simulation scenarios and their 95% CIs

| Scenario | Horizon | Method | Win_rate | Win_rate_ci_low | Win_rate_ci_high |
|---|---|---|---|---|---|
| **Baseline linear** | 1 | ARDL | 0.225 | 0.170 | 0.285 |
| | | LSTM | 0.200 | 0.145 | 0.255 |
| | | Naive | 0.235 | 0.180 | 0.295 |
| | | RF | 0.165 | 0.115 | 0.220 |
| | | SVR | 0.175 | 0.125 | 0.230 |
| | 3 | ARDL | 0.215 | 0.155 | 0.275 |
| | | LSTM | 0.230 | 0.175 | 0.290 |
| | | Naive | 0.190 | 0.140 | 0.245 |
| | | RF | 0.205 | 0.145 | 0.260 |
| | | SVR | 0.160 | 0.110 | 0.215 |
| | 6 | ARDL | 0.265 | 0.205 | 0.325 |
| | | LSTM | 0.230 | 0.170 | 0.285 |
| | | Naive | 0.160 | 0.110 | 0.210 |
| | | RF | 0.185 | 0.135 | 0.240 |
| | | SVR | 0.160 | 0.110 | 0.210 |
| **High persistence** | 1 | ARDL | 0.295 | 0.235 | 0.360 |
| | | LSTM | 0.140 | 0.095 | 0.190 |
| | | Naive | 0.255 | 0.195 | 0.320 |
| | | RF | 0.180 | 0.130 | 0.235 |
| | | SVR | 0.130 | 0.085 | 0.180 |
| | 3 | ARDL | 0.310 | 0.250 | 0.370 |
| | | LSTM | 0.185 | 0.130 | 0.240 |
| | | Naive | 0.120 | 0.075 | 0.165 |
| | | RF | 0.205 | 0.150 | 0.265 |
| | | SVR | 0.180 | 0.130 | 0.235 |
| | 6 | ARDL | 0.390 | 0.320 | 0.460 |
| | | LSTM | 0.170 | 0.115 | 0.225 |
| | | Naive | 0.110 | 0.070 | 0.155 |
| | | RF | 0.145 | 0.100 | 0.195 |
| | | SVR | 0.185 | 0.130 | 0.240 |

| Scenario | Horizon | Method | Win_rate | Win_rate_ci_low | Win_rate_ci_high |
|---|---|---|---|---|---|
| **MA noninvertible** | 1 | ARDL | 0.200 | 0.145 | 0.260 |
| | | LSTM | 0.215 | 0.160 | 0.275 |
| | | Naive | 0.195 | 0.140 | 0.250 |
| | | RF | 0.190 | 0.135 | 0.250 |
| | | SVR | 0.200 | 0.145 | 0.255 |
| | 3 | ARDL | 0.230 | 0.170 | 0.290 |
| | | LSTM | 0.210 | 0.155 | 0.265 |
| | | Naive | 0.200 | 0.145 | 0.255 |
| | | RF | 0.185 | 0.135 | 0.240 |
| | | SVR | 0.175 | 0.125 | 0.230 |
| | 6 | ARDL | 0.195 | 0.140 | 0.255 |
| | | LSTM | 0.220 | 0.160 | 0.280 |
| | | Naive | 0.195 | 0.145 | 0.250 |
| | | RF | 0.250 | 0.190 | 0.310 |
| | | SVR | 0.140 | 0.095 | 0.190 |
| **Non linear regime** | 1 | ARDL | 0.225 | 0.170 | 0.285 |
| | | LSTM | 0.170 | 0.120 | 0.225 |
| | | Naive | 0.180 | 0.130 | 0.235 |
| | | RF | 0.175 | 0.125 | 0.230 |
| | | SVR | 0.250 | 0.190 | 0.310 |
| | 3 | ARDL | 0.195 | 0.140 | 0.250 |
| | | LSTM | 0.245 | 0.185 | 0.305 |
| | | Naive | 0.175 | 0.125 | 0.230 |
| | | RF | 0.170 | 0.120 | 0.225 |
| | | SVR | 0.215 | 0.160 | 0.275 |
| | 6 | ARDL | 0.185 | 0.135 | 0.235 |
| | | LSTM | 0.230 | 0.175 | 0.290 |
| | | Naive | 0.175 | 0.125 | 0.225 |
| | | RF | 0.230 | 0.175 | 0.290 |
| | | SVR | 0.180 | 0.130 | 0.235 |

| Scenario | Horizon | Method | Win_rate | Win_rate_ci_low | Win_rate_ci_high |
|---|---|---|---|---|---|
| **Structural break** | 1 | ARDL | 0.215 | 0.160 | 0.270 |
| | | LSTM | 0.200 | 0.145 | 0.255 |
| | | Naive | 0.225 | 0.170 | 0.285 |
| | | RF | 0.205 | 0.150 | 0.265 |
| | | SVR | 0.155 | 0.105 | 0.205 |
| | 3 | ARDL | 0.185 | 0.130 | 0.240 |
| | | LSTM | 0.255 | 0.200 | 0.315 |
| | | Naive | 0.170 | 0.120 | 0.225 |
| | | RF | 0.230 | 0.175 | 0.290 |
| | | SVR | 0.160 | 0.110 | 0.215 |
| | 6 | ARDL | 0.195 | 0.145 | 0.250 |
| | | LSTM | 0.260 | 0.200 | 0.320 |
| | | Naive | 0.175 | 0.125 | 0.230 |
| | | RF | 0.195 | 0.140 | 0.250 |
| | | SVR | 0.175 | 0.120 | 0.230 |

Source: own computations

## Table A8. The optimal values for the hyperparameters of the RF method

| Country | Hyperparameters | Values of the hyperparameters for | | | |
|---|---|---|---|---|---|
| | | Univariate setting | Multivariate setting | | |
| | | | HICP-Unemployment | HICP-ESI | HICP-Unemployment-ESI |
| BG | n_estimators | 150 | 50 | 50 | 150 |
| | max_depth | None | 5 | 10 | None |
| | min_samples_split | 2 | 2 | 2 | 2 |
| | max_features | 0.9 | 0.7 | 0.9 | 0.7 |
| RO | n_estimators | 200 | 100 | 50 | 50 |
| | max_depth | 10 | 10 | 10 | None |
| | min_samples_split | 2 | 2 | 2 | 2 |
| | max_features | 1.0 | 0.9 | 0.5 | 0.9 |
| SK | n_estimators | 200 | 50 | 100 | 50 |
| | max_depth | 5 | 10 | 10 | None |
| | min_samples_split | 2 | 2 | 2 | 2 |
| | max_features | 0.9 | 0.9 | 0.7 | 1.0 |
| SI | n_estimators | 75 | 50 | 100 | 75 |
| | max_depth | 5 | 5 | 5 | None |
| | min_samples_split | 10 | 10 | 10 | 12 |
| | max_features | 1.0 | 1.0 | 1.0 | 0.7 |
| LT | n_estimators | 200 | 75 | 75 | 75 |
| | max_depth | 5 | 5 | 5 | 5 |
| | min_samples_split | 2 | 2 | 2 | 2 |
| | max_features | 1.0 | 1.0 | 0.7 | 0.5 |

| | | | | | |
|---|---|---|---|---|---|
| **LV** | n_estimators | 50 | 100 | 200 | 100 |
| | max_depth | None | 5 | 5 | 5 |
| | min_samples_split | 10 | 5 | 10 | 2 |
| | max_features | 1.0 | 0.9 | 1.0 | 0.5 |
| **EE** | n_estimators | 100 | 150 | 200 | 200 |
| | max_depth | 10 | 10 | None | 5 |
| | min_samples_split | 2 | 2 | 10 | 2 |
| | max_features | 0.9 | 1.0 | 1.0 | 1.0 |
| **PL** | n_estimators | 200 | 150 | 150 | 150 |
| | max_depth | None | 10 | 5 | None |
| | min_samples_split | 2 | 2 | 2 | 2 |
| | max_features | 1.0 | 1.0 | 0.5 | 0.5 |
| **HU** | n_estimators | 50 | 75 | 50 | 75 |
| | max_depth | 5 | 10 | 5 | 5 |
| | min_samples_split | 2 | 2 | 2 | 2 |
| | max_features | 0.5 | 0.9 | 1.0 | 0.5 |
| **CZ** | n_estimators | 150 | 50 | 50 | 75 |
| | max_depth | 10 | 10 | None | None |
| | min_samples_split | 2 | 2 | 2 | 2 |
| | max_features | 0.5 | 0.9 | 0.9 | 1.0 |

Source: own computations

## Table A9. The optimal values for the hyperparameters of the SVR method

| Country | Hyperparameters | Values of the hyperparameters for | | | |
|---|---|---|---|---|---|
| | | Univariate setting | Multivariate setting | | |
| | | | HICP-Unemployment | HICP-ESI | HICP-Unemployment-ESI |
| BG | C | 3 | 1 | 8 | 7 |
| | coef0 | 0.5 | 1.0 | 0.01 | 0.01 |
| | degree | 2 | 1 | 1 | 1 |
| | epsilon | 0.01 | 0.01 | 0.01 | 0.01 |
| | gamma | 0.4 | 0.7 | 0.8 | 0.3 |
| | kernel | poly | poly | poly | poly |
| RO | C | 1 | 9 | 7 | 6 |
| | coef0 | 2.5 | 2.5 | 2.5 | 2.0 |
| | degree | 2 | 1 | 1 | 1 |
| | epsilon | 0.01 | 0.01 | 0.01 | 0.01 |
| | gamma | 0.6 | 4 | 0.9 | 0.9 |
| | kernel | poly | poly | poly | poly |
| SK | C | 5 | 9 | 9 | 2 |
| | coef0 | 0.5 | 0.5 | 0.5 | 1.0 |
| | degree | 1 | 1 | 1 | 1 |
| | epsilon | 0.01 | 0.01 | 0.01 | 0.01 |
| | gamma | 0.9 | 0.2 | 0.8 | 0.7 |
| | kernel | poly | poly | poly | poly |
| SI | C | 8 | 2 | 4 | 7 |
| | coef0 | 0.5 | 0.5 | 0.01 | 4 |
| | degree | 1 | 1 | 1 | 1 |
| | epsilon | 0 | 0.01 | 0.01 | 0.01 |
| | gamma | 0.8 | 0.9 | 0.1 | 0.6 |
| | kernel | poly | poly | poly | poly |
| LT | C | 9 | 4 | 7 | 5 |
| | coef0 | 0.5 | 0.5 | 1.0 | 1.0 |
| | degree | 1 | 1 | 1 | 1 |
| | epsilon | 0.01 | 0.01 | 0.01 | 0.01 |
| | gamma | 0.6 | 0.8 | 0.6 | 0.8 |
| | kernel | poly | poly | poly | poly |

| | | | | | |
|---|---|---|---|---|---|
| **LV** | C | 9 | 9 | 6 | 3 |
| | coef0 | 0.01 | 0.01 | 2.0 | 0.01 |
| | degree | 1 | 1 | 1 | 1 |
| | epsilon | 0.01 | 0.01 | 0.01 | 0.01 |
| | gamma | 0.9 | 0.4 | 0.9 | 0.6 |
| | kernel | poly | poly | poly | poly |
| **ES** | C | 9 | 8 | 9 | 9 |
| | coef0 | 0.5 | 2.5 | 2.0 | 2.0 |
| | degree | 1 | 1 | 1 | 1 |
| | epsilon | 0.01 | 0.01 | 0.01 | 0.01 |
| | gamma | 0.9 | 0.7 | 0.5 | 0.4 |
| | kernel | poly | poly | poly | poly |
| **PL** | C | 7 | 4 | 2 | 6 |
| | coef0 | 2.0 | 2.0 | 2.5 | 1.0 |
| | degree | 1 | 1 | 1 | 1 |
| | epsilon | 0.01 | 0.01 | 0.01 | 0.01 |
| | gamma | 0.8 | 0.7 | 0.2 | 0.9 |
| | kernel | poly | poly | poly | poly |
| **HU** | C | 3 | 9 | 9 | 2 |
| | coef0 | 1.0 | 0.01 | 0.5 | 1.0 |
| | degree | 2 | 1 | 1 | 1 |
| | epsilon | 0.01 | 0.01 | 0.01 | 0.01 |
| | gamma | 0.9 | 0.5 | 0.6 | 0.6 |
| | kernel | poly | poly | poly | Poly |
| **CZ** | C | 9 | 9 | 3 | 6 |
| | coef0 | 0.01 | 0.01 | 1.0 | 0.5 |
| | degree | 1 | 1 | 1 | 1 |
| | epsilon | 0.01 | 0.01 | 0.01 | 0.01 |
| | gamma | 0.9 | 0.9 | 0.1 | 0.5 |
| | kernel | poly | poly | poly | poly |

Source: own computations

## Table A10. The optimal values for the hyperparameters of the LSTM method

| Country | Hyperparameters | Values of the hyperparameters for | | | |
|---|---|---|---|---|---|
| | | Univariate setting | Multivariate setting | | |
| | | | HICP-Unemployment | HICP-ESI | HICP-Unemployment-ESI |
| BG | batch_size | 8 | 1 | 1 | 8 |
| | Dropout layer 1 | 0.2 | 0.1 | 0.1 | 0.1 |
| | Dropout layer 2 | 0.0 | 0.1 | 0.1 | 0.1 |
| | L2 reg | 0.0 | 0.2 | 0.1 | 0.2 |
| | recurrent_dropout layer 1 | 0.1 | 0.2 | 0.1 | 0.0 |
| | recurrent_dropout layer 1 | 0.1 | 0.1 | 0.1 | 0.1 |
| | units | 128 | 512 | 512 | 512 |
| RO | batch_size | 1 | 1 | 1 | 1 |
| | Dropout layer 1 | 0.2 | 0.1 | 0.2 | 0.1 |
| | Dropout layer 2 | 0.1 | 0.1 | 0.2 | 0.0 |
| | L2 reg | 0.1 | 0.2 | 0.2 | 0.2 |
| | recurrent_dropout layer 1 | 0.1 | 0.2 | 0.0 | 0.2 |
| | recurrent_dropout layer 1 | 0.1 | 0.2 | 0.1 | 0.1 |
| | units | 128 | 512 | 128 | 512 |
| SK | batch_size | 1 | 1 | 1 | 1 |
| | Dropout layer 1 | 0.2 | 0.1 | 0.1 | 0.1 |
| | Dropout layer 2 | 0.1 | 0.0 | 0.2 | 0.2 |
| | L2 reg | 0.1 | 0.0 | 0.2 | 0.2 |
| | recurrent_dropout layer 1 | 0.1 | 0.0 | 0.2 | 0.2 |
| | recurrent_dropout layer 1 | 0.1 | 0.2 | 0.1 | 0.1 |
| | units | 128 | 512 | 512 | 512 |

| | | | | | |
|---|---|---|---|---|---|
| **SI** | batch_size | 1 | 8 | 4 | 1 |
| | Dropout layer 1 | 0.2 | 0.1 | 0.2 | 0.1 |
| | Dropout layer 2 | 0.2 | 0.1 | 0.1 | 0.1 |
| | L2 reg | 0.1 | 0.2 | 0.1 | 0.2 |
| | recurrent_dropout layer 1 | 0.1 | 0.0 | 0.1 | 0.1 |
| | recurrent_dropout layer 1 | 0.1 | 0.2 | 0.1 | 0.1 |
| | units | 128 | 512 | 512 | 512 |
| **LT** | batch_size | 1 | 1 | 1 | 1 |
| | Dropout layer 1 | 0.1 | 0.0 | 0.1 | 0.2 |
| | Dropout layer 2 | 0.2 | 0.2 | 0.1 | 0.0 |
| | L2 reg | 0.0 | 0.1 | 0.1 | 0.1 |
| | recurrent_dropout layer 1 | 0.1 | 0.2 | 0.1 | 0.1 |
| | recurrent_dropout layer 1 | 0.0 | 0.1 | 0.2 | 0.1 |
| | units | 512 | 512 | 512 | 512 |
| **LV** | batch_size | 1 | 1 | 1 | 1 |
| | Dropout layer 1 | 0.0 | 0.1 | 0.1 | 0.1 |
| | Dropout layer 2 | 0.2 | 0.1 | 0.0 | 0.1 |
| | L2 reg | 0.2 | 0.1 | 0.1 | 0.2 |
| | recurrent_dropout layer 1 | 0.0 | 0.2 | 0.1 | 0.1 |
| | recurrent_dropout layer 1 | 0.2 | 0.2 | 0.2 | 0.1 |
| | units | 512 | 512 | 512 | 512 |
| **EE** | batch_size | 1 | 1 | 1 | 1 |
| | Dropout layer 1 | 0.1 | 0.1 | 0.1 | 0.2 |
| | Dropout layer 2 | 0.2 | 0.1 | 0.2 | 0.2 |
| | L2 reg | 0.0 | 0.1 | 0.1 | 0.1 |
| | recurrent_dropout layer 1 | 0.1 | 0.2 | 0.2 | 0.1 |
| | recurrent_dropout layer 1 | 0.2 | 0.2 | 0.2 | 0.1 |
| | units | 512 | 512 | 512 | 512 |

| | | | | | |
|---|---|---|---|---|---|
| **PL** | batch_size | 1 | 1 | 1 | 1 |
| | Dropout layer 1 | 0.0 | 0.1 | 0.0 | 0.1 |
| | Dropout layer 2 | 0.0 | 0.1 | 0.1 | 0.1 |
| | L2 reg | 0.1 | 0.2 | 0.2 | 0.2 |
| | recurrent_dropout layer 1 | 0.2 | 0.1 | 0.2 | 0.1 |
| | recurrent_dropout layer 1 | 0.2 | 0.2 | 0.2 | 0.1 |
| | units | 512 | 512 | 512 | 512 |
| **HU** | batch_size | 1 | 1 | 1 | 1 |
| | Dropout layer 1 | 0.1 | 0.2 | 0.1 | 0.2 |
| | Dropout layer 2 | 0.1 | 0.0 | 0.2 | 0.0 |
| | L2 reg | 0.1 | 0.0 | 0.2 | 0.2 |
| | recurrent_dropout layer 1 | 0.1 | 0.1 | 0.2 | 0.0 |
| | recurrent_dropout layer 1 | 0.1 | 0.2 | 0.1 | 0.1 |
| | units | 512 | 512 | 512 | 512 |
| **CZ** | batch_size | 1 | 1 | 1 | 8 |
| | Dropout layer 1 | 0.2 | 0.1 | 0.2 | 0.2 |
| | Dropout layer 2 | 0.2 | 0.1 | 0.1 | 0.1 |
| | L2 reg | 0.0 | 0.1 | 0.1 | 0.1 |
| | recurrent_dropout layer 1 | 0.1 | 0.2 | 0.2 | 0.1 |
| | recurrent_dropout layer 1 | 0.2 | 0.0 | 0.0 | 0.2 |
| | units | 512 | 512 | 512 | 512 |

Source: own computations