STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **11**, September 2023, pp 1041–1050.
Published online in International Academic Press (www.IAPress.org)

Comparing the Accuracy of Classical, Machine Learning, and Deep Learning Methods in Time Series Forecasting: A Case Study of USA Inflation

Youness Jouilil [1,*], M'barek Iaousse [2]

[1] *REIS Laboratory, Department of Economics, Faculty of Economics and Social Sciences of Mohammedia, Hassan II University of Casablanca, Morocco. Orcid:0000-0002-0394-8329*
[2] *C3S Laboratory, Hassan II University of Casablanca, Morocco. Orcid:0000-0002-4949-5556*

Abstract This paper compares some selected statistical classical methods, machine learning, and deep learning algorithms for time series forecasting. In particular, we have selected the Exponential Smoothing, Prophet, hybrid ARIMA-GARCH model, K-Nearest Neighbors (KNN), and Long-Short Term Memory (LSTM). The comparison concerns univariate time series in short-run forecasting. The data set used in this study is related to US inflation and covers the period from 1965 to 2021. The performance of the models was evaluated using different metrics especially Mean Squared Error (MSE), Mean Absolute Error (MAE), Median Absolute Error (Median AE), and Root Mean Squared Error (RMSE). The results of the numerical comparison show that the best performance was achieved by Exponential Smoothing, followed closely by KNN. The results indicate that these two models are well-suited for forecasting inflation in the US. ARIMA-GARCH, LSTM, and Prophet performed relatively poorly in comparison. Overall, the findings of this study can be useful for practitioners in choosing the most suitable method for forecasting inflation in the US in the short-term period.

Keywords Machine learning, Time series forecasting, Classical approaches, Forecasting.

AMS 2010 subject classifications C22, C53.

DOI: 10.19139/soic-2310-5070-1767.

1. Introduction

Inflation, defined as a persistent increase in the general price level of goods and services in an economy over a period of time [1], is a crucial macroeconomic indicator that has significant implications for individuals, businesses, and governments. Inflation affects the purchasing power of consumers, the competitiveness of businesses, and the stability of the economy as a whole. For these reasons, inflation has been the subject of economic research for many years and continues to be a central concern for policymakers and economists.

Over time, inflation in the United States has experienced both periods of stability and periods of significant change. During the 20th century, for example, the United States experienced periods of high inflation during and after World War I, as well as during the 1970s, when the country experienced the so-called "stagflation" [2]. However, in recent decades, inflation in the United States has been relatively stable, with inflation rates generally remaining below 4%.

The importance of accurately forecasting inflation cannot be overstated. Inflation expectations play a crucial role in shaping economic behavior, and accurate inflation forecasts are essential for the formulation of monetary and fiscal policies [3]. Accurate inflation forecasts can also help businesses and individuals make more informed decisions about spending, saving, and investing [4].

*Correspondence to: Youness Jouilil (Email: y.jouilil@gmail.com). REIS Laboratory, Department of Economics Faculty of Law Economics and Social Sciences of Mohammedia, Hassan II University of Casablanca, Morocco, Rabat Road, Bd Hassan II BP 145 CP: 20650 - 20650, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

Given the importance of inflation and the need for accurate inflation forecasts, it is not surprising that economists and policymakers have developed various methods for forecasting inflation, including the use of time series analysis. Time series analysis involves the use of statistical techniques to analyze the patterns in historical data and make predictions about future values [5]. As we will see in this paper, time series analysis has proven to be a powerful tool for inflation forecasting, with several studies demonstrating the efficacy of various time series models, including classical models like ARIMA and exponential smoothing, as well as more recent machine learning and deep learning methods [6]. Time series analysis is a powerful tool that has various applications in fields such as economics, finance, and meteorology. The main objective of time series analysis is forecasting, which involves predicting future values of a time series based on its past values. There are several classical time series models that can be used for forecastings, such as univariate Moving Average, univariate Autoregressive, and Autoregressive Integrated Moving Average (ARIMA) models.

ARIMA models are widely used for time series forecasting due to their ability to capture both short-term and long-term patterns in the data [7]. They are a combination of autoregression and moving average models, and can be made stationary through the use of differencing. Several studies have compared the performance of ARIMA models with other time series models and have found that ARIMA models generally provide more accurate and precise predictions. For example, Hyndman and Koehler compared the performance of various time series models on a large dataset and found that ARIMA models performed the best [5].

Another popular method for time series forecasting is exponential smoothing. Exponential smoothing is a method that uses a weighted average of past observations to predict future observations, where the weights are calculated using an exponential function that gives more weight to more recent observations. Exponential smoothing has been widely used in practice and has been found to perform well in a variety of settings. A study by Goodman showed that exponential smoothing methods outperformed other time series models in terms of accuracy and precision [4].

In recent years, the field of time series forecasting has witnessed a significant transformation with the integration of deep learning methods[18, 19], representing a paradigm shift in the way complex temporal patterns are extracted and utilized for predictions [3, 8, 9, 10]. This evolution is particularly evident through the application of Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks, which have garnered substantial attention due to their remarkable capacity to capture intricate patterns within time series data.

RNNs, known for their sequential modeling capabilities, have proven highly effective in handling time-dependent data like time series [16]. Their recurrent structure enables them to incorporate historical information into predictions, allowing them to capture the temporal dependencies that classical models might overlook.

Similarly, LSTM networks, a specialized type of RNN designed to mitigate the vanishing gradient problem, have shown remarkable proficiency in capturing long-range dependencies[17].

A study by [6] provides an overview of the use of deep learning techniques in time series forecasting. The authors survey the current literature on the topic and discuss the various deep-learning architectures and methods used for time series forecasting. The study concludes by summarizing the strengths and limitations of these methods and suggesting potential avenues for future research.

The objective of this paper is to provide an overview of the time series methods used for inflation forecasting and compare their performance. To achieve this goal, the paper will present a comprehensive survey of the recent literature, focusing on the classical time series methods, ML methods, and DL methods.

The present paper is structured as follows: Section 2 describes the data source used for the comparison, section 3 presents an overview of the methods used in this study, which include ARIMA [11, 9], GARCH [12], Exponential Smoothing [11], KNN [13], Prophet [14], and LSTM [7, 15]. Each method is briefly explained and defined. Section 4 compares the performance of the different techniques using evaluation metrics such as MSE, MAE, Median Absolute Error, and R-squared. Finally, section 5 summarizes the findings of the study, discusses the limitations of the research, and outlines perspectives for further improvement.

2. Data

The data used in this study is obtained from the World Bank database †. The World Bank is an international organization that collects, compiles, and disseminates economic, financial, and social data from various sources such as national governments, international organizations, and other institutions. The data set consists of the inflation rate in the USA, collected on a yearly basis. The inflation rate data was collected and processed to create a time series data set that was used for the analysis.

The period of the data used in this study covers the years 1965 to 2021, a total of 57 years of data. The data was divided into two parts, the first part was used for training the models, and the second part was used for testing the models. The training data set consists of data from 1965 to 2014, which represents a proportion of 1% of the total number of observations due to the small number of observations in the data set. The testing data set consists of data from 2015 to 2021. The aim of dividing the data into two parts was to evaluate the performance of the models developed in this study.

The data pre-processing step was applied to the data set to clean and prepare the data for analysis. The pre-processing step included checking for missing values (none), outliers, and the consistency of the data. The data was also checked for stationarity to ensure that the data met the assumptions of the time series analysis. The data was transformed if necessary to meet the assumptions of the models used in this study. The pre-processed data was then divided into the training and testing data sets, as mentioned earlier, and used for the analysis.

In this study, we will use Python and Anaconda to analyze the inflation data. Anaconda is a distribution of Python that comes with a lot of useful libraries and tools for data science, such as `NumPy`, `pandas`, and `scikit-learn`. By using Anaconda, we can easily access and manipulate the data and perform various statistical analyses on it.

3. Methodology

3.1. Exponential Smoothing Algorithm

Exponential smoothing is a time series forecasting method that uses a weighted average of past observations to predict future points. The weight assigned to each past observation decreases exponentially as the observation gets older. The method is defined mathematically as:

$$\hat{y}_t = \alpha y_{t-1} + (1-\alpha)\hat{y}_{t-1} \tag{1}$$

Where $\hat{y}_t$ is the predicted value at time t, y_{t-1} is the observed value at time $t-1$, and α is the smoothing parameter, a value between 0 and 1 that determines the weight assigned to the previous observation.

Exponential smoothing is useful in time series analysis because it can handle both trend and seasonality in the data, and it is easy to implement. It is particularly useful for forecasting short-term trends in data with no clear underlying patterns.

3.2. ARIMA-GARCH algorithm

The Auto-Regressive Integrated Moving Average, or ARIMA, models can be defined mathematically as follows:

$$\Phi_p(B)(1-B)^d X_t = \Theta_q(B)\eta_t \tag{2}$$

Where X_t is the time series data at time t, B is the backshift operator, d is the order of differencing, θ_i and ϕ_i are the parameters of the model, and η_t is the error term.

ARIMA models are particularly useful when the time series data has a clear underlying pattern, such as seasonality or a trend. One of the algorithms of ARIMA is the Box-Jenkins algorithm, It can be summarized as follows:

† The World Bank inflation data can be accessed online via the World Bank database website https://data.worldbank.org/.

Algorithm 1 Box-Jenkins Algorithm

Require: Time series data X_t
Ensure: ARIMA model parameters

1: Perform preliminary analysis on the time series data to identify the order of differencing d, the number of autoregressive terms p, and the number of moving average terms q
2: Fit the model using the identified values of d, p, and q
3: Use diagnostic checking to check the model fit
4: Refine the model as necessary
5: Return the estimated model parameters

Meanwhile, the GARCH process or even Generalized ARCH process is commonly implied to predict the volatility of a given time series (Bollerslev, T., 1986). The process η_t is a GARCH(p,q) process if it is stationary and if it satisfies, for all t and some strictly positive-valued process σ_t, the equations :

$$\begin{cases} y_t & = f(t, y_{t-1}, y_{t-2}, \ldots) + \eta_t \\ \eta_t & = \sigma_t z_t \\ \sigma_t^2 & = \alpha_0 + \sum_{j=0}^{q} \alpha_j a_{t-j}^2 + \sum_{j=0}^{q} \beta_j \sigma_{t-j}^2 \\ z_t & \sim N(0,1) \end{cases} \tag{3}$$

Where $\alpha_0 \in \mathbf{R}^{+*}$, $\alpha_i; i \in \{1, 2, \ldots, q\}$, $\beta_j; j \in \{1, 2, \ldots, p\}$ and the function y_t refers to a deterministic information process.

It should be noticed that the ARIMA estimates the conditional mean, whereas a GARCH model estimates the conditional variance in the residuals of the ARIMA estimation. In other words, the ARIMA-GARCH models are popular forecasting methods in the presence of volatility.

3.3. K-nearest Neighbors Algorithm

The K-nearest Neighbors (KNN) algorithm stands as a non-parametric technique utilized for both classification and regression tasks. This algorithm operates by identifying the K instances within the training dataset that exhibit the closest proximity to the target observation in question. Subsequently, it attributes the prevailing output value, observed most frequently among these K instances, to the aforementioned target observation.

When provided with a novel observation denoted as x_{new}, along with a training dataset labeled as X_{train}, the KNN algorithm undertakes the task of locating the K nearest observations within X_{train} concerning x_{new}. These K closest observations are denoted as X_{KNN}. Following this, the algorithm proceeds to ascribe the most prevalent output value among these K proximate observations to x_{new}.

Algorithm 2 K-nearest Neighbors Algorithm

```
procedure KNN(x_new, X_train, y_train, K)
    X_KNN ← FindKNN(x_new, X_train, K)
    y_new ← MostCommonValue(y_train[X_KNN])
    return y_new
end procedure
```

3.4. Long Short Term Memory Algorithm

Long Short-term Memory (LSTM) Networks, are a special kind of Recurrent Neural Networks (RNNs). They are designed to work with sequential data, such as speech or video. LSTMs are particularly useful for tasks where the input sequences can be very long.

Although it well-known that LSTM networks require a large amount of data to be trained effectively and may not be the best choice for small datasets, we did implement it to compare it with the other methods. The mathematical equations for an LSTM unit can be represented as follows:

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$$

$$f_t = \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$$

$$o_t = \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$$

$$g_t = \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g)$$

$$c_t = f_t * c_{t-1} + i_t * g_t$$

$$h_t = o_t * \tanh(c_t)$$

The LSTM algorithm is as follows:

Algorithm 3 LSTM Algorithm

$i_t \leftarrow \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i)$
$f_t \leftarrow \sigma(W_{xf}x_t + W_{hf}h_{t-1} + b_f)$
$o_t \leftarrow \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o)$
$g_t \leftarrow \tanh(W_{xg}x_t + W_{hg}h_{t-1} + b_g)$
$c_t \leftarrow f_t * c_{t-1} + i_t * g_t$
$h_t \leftarrow o_t * \tanh(c_t)$

3.5. Facebook Prophet algorithm

Facebook Prophet is a time series forecasting library developed by Facebook's Core Data Science team. It is based on an additive model framework, which means it models the trend and seasonality of the data as separate components and fits them using regression. Prophet is designed for forecasting time series with a strong seasonal component and multiple sources of growth.

Prophet was first released in 2017 and has since become a widely used tool for time series forecasting in various industries, including finance, economics, and retail. Its popularity is due to its ease of use, as well as its ability to handle non-linear trends, seasonality, and holidays.

At its core, Prophet models the time series as an additive combination of three components: a trend, a seasonal component, and a holiday component.

The trend component is modeled using a piecewise linear function, where the breakpoints are determined automatically. The seasonal component is modeled using a Fourier series, and the holiday component is modeled using an indicator variable. The overall model can be represented as follows:

$$y(t) = g(t) + s(t) + h(t) + \epsilon_t \tag{4}$$

Where:

$y(t)$ is the observed time series $g(t)$ is the trend component $s(t)$ is the seasonal component $h(t)$ is the holiday component ϵ_t is the error term

Prophet's algorithm can be summarized as follows:

It should be noted that the previous algorithm is only a simplified version of the fbprophet algorithm and does not necessarily include all detailed steps. It is recommended to consult the fbprophet documentation for a more complete and accurate implementation.

Algorithm 4 fbprophet Algorithm

procedure FBPROPHET(df)
  df = data frame of historical time-series data
  Fit a curve to the historical data using a Prophet model
  Define and fit a linear regression model for the log-transformed data
  Remove the trend and seasonality from the data using the linear regression model
  Fit the residuals from the linear regression to a Gaussian process regression model
  Add the trend and seasonality back to the data using the linear regression model
  Make future predictions based on the fitted Gaussian process regression model
  return the future predictions
end procedure

4. Results & comparison

In this section, we will derive the previous methods for time series forecasting using a real data set that comes from the World Bank. We will choose the smaller criterion. As can be shown from the Figure below, the ARIMA(p,1,q) is the appropriate one. It should be noted that for a MA(q) process the ACF and PACF are given by :

$$PACF(i) \neq 0 \ ; \forall \ \ i \in \mathbb{N} \tag{5}$$

$$PACF(i) = \begin{cases} \neq 0 & ; \forall \ \ i = 1, \ldots, p \\ = 0 & ; \forall \ \ i > p \end{cases} \tag{6}$$

Similarly, for an AR(p) process the ACF and PACF are given by :

$$ACF(i) \neq 0 \ ; \forall \ \ i \in \mathbb{N} \tag{7}$$

$$ACF(i) = \begin{cases} \neq 0 & ; \forall \ \ i = 1, \ldots, q \\ = 0 & ; \forall \ \ i > q \end{cases} \tag{8}$$

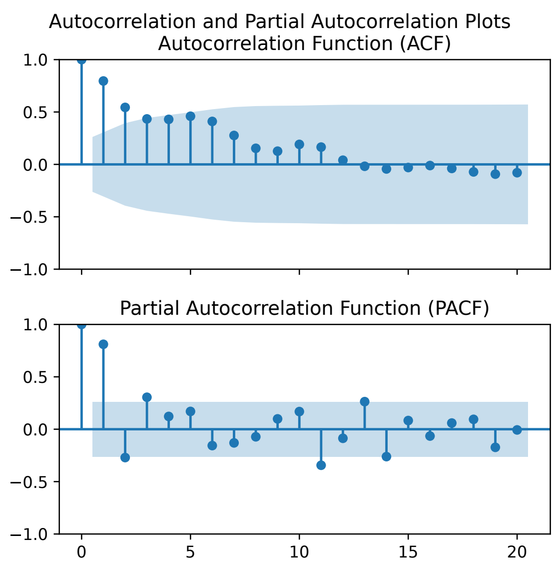

Figure 1. Autocorelation and partial autocorrelation function of the time series

We have used Python programming language to estimate the ARIMA parameters, shown in the program upper. To identify the ARMA parameters, we have to implement the Box-Jenkins methodology. In the Python language, the package “auto.arima” has been used to estimate the parameters of our ARIMA model.

We, therefore, checked several models to detect the most appropriate one for our target series. Our manipulations show that the adopted model parameter is ARIMA(0,1,2).

Table 1. ARIMA-GARCH modeling of inflation in the USA

ARIMA model of time series	coef	std err	z	$P > \|z\|$	[0.025	0.975]
ma.L1	0.1432	0.132	1.081	0.280	-0.117	0.403
ma.L2	-0.4807	0.155	-3.105	0.002	-0.784	-0.177
sigma2	2.5410	0.435	5.845	0.000	1.689	3.393

Garch model of residuals (Mean Model)	coef	std err	t	$P > \|t\|$	95.0% Conf. Int.
mu	-0.0592	0.249	-0.238	0.812	[-0.547, 0.429]

4.1. Baseline comparison

Figure 2 shows the comparison of the classical approach and machine learning techniques over the test period and Figure 3 visualizes the comparison in terms of the different metrics used.

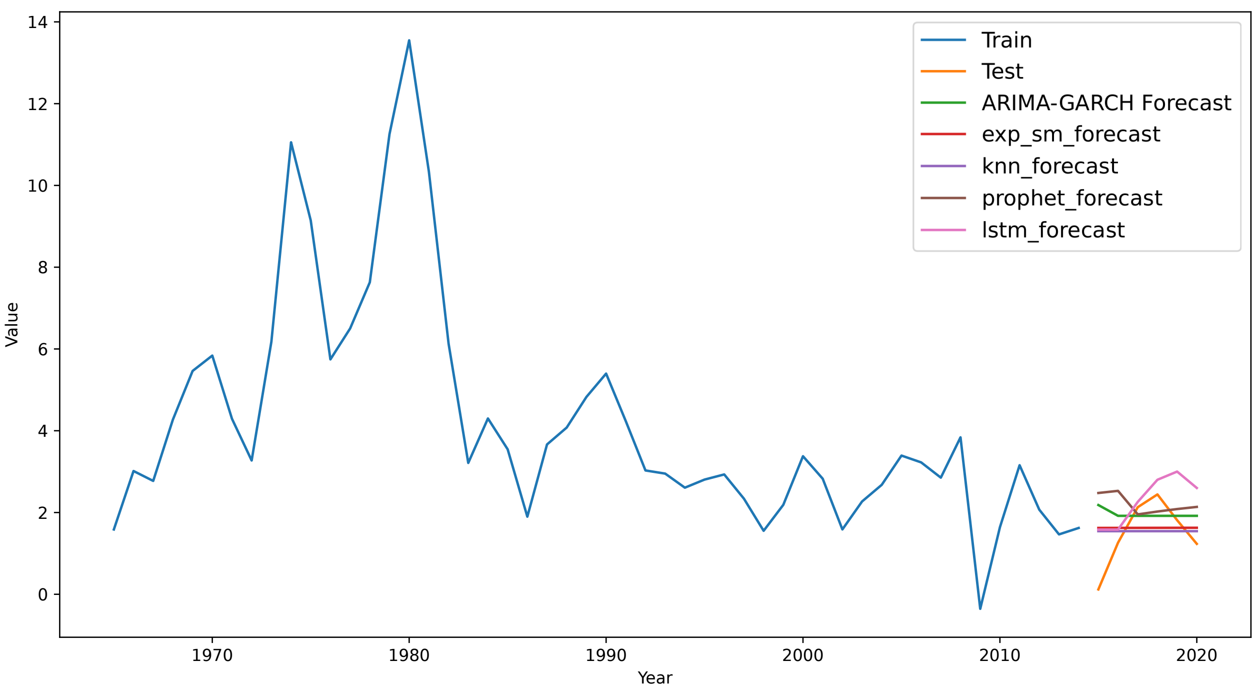

Figure 2. Classical methods vs ML algorithms

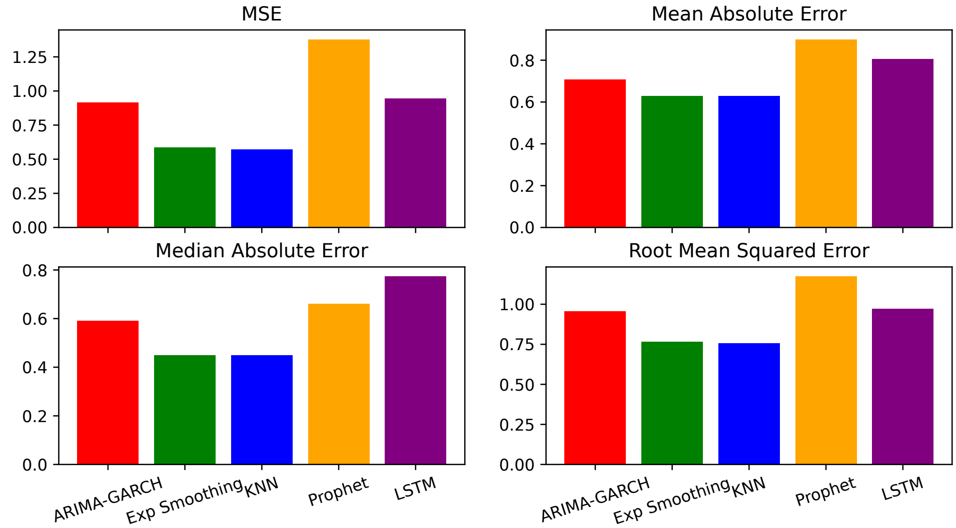

Figure 3. Evaluation score of the prediction algorithms

4.2. Discussion & analysis

In this section, we will evaluate the performance of five-time series forecasting models: ARIMA-GARCH, Exponential Smoothing, k-Nearest Neighbors (KNN), Prophet, and Long Short-Term Memory (LSTM). To assess the performance of our models, we will use four measures of accuracy, namely Mean Squared Error (MSE), Mean Absolute Error (MAE), Median Absolute Error (Median AE), and Root Mean Squared Error (RMSE).

The Mean Squared Error measures the average of the squares of the errors between the predicted and actual values. In terms of MSE, KNN and Exponential Smoothing perform the best, with values of 0.5717 and 0.5845, respectively. On the other hand, Prophet has the highest MSE, with a value of 1.3783.

The Mean Absolute Error measures the average of the absolute differences between the predicted and actual values. In terms of MAE, KNN performs the best, with a value of 0.6285, followed closely by Exponential Smoothing, with a value of 0.6281. Prophet has the highest MAE, with a value of 0.8997.

In terms of Median Absolute Error, which measures the median of the absolute differences between the predicted and actual values, the KNN and Exponential Smoothing perform the best, with values of 0.4483 and 0.4494, respectively. Prophet has the highest Median AE, with a value of 0.6611.

The Root Mean Squared Error measures the square root of the average of the squares of the errors between the predicted and actual values. In terms of RMSE, Exponential Smoothing performs the best, with a value of 0.7645, followed by KNN, with a value of 0.7561. Prophet has the highest RMSE, with a value of 1.1740.

Based on these results, KNN and Exponential Smoothing perform the best in terms of accuracy, as they have the lowest MSE, MAE, Median AE, and RMSE values. On the other hand, Prophet performs the worst among the five models.

In terms of overall performance, KNN and Exponential Smoothing appear to be the best models, as they perform well in all four accuracy measures.

To further improve the performance of these models, it may be beneficial to fine-tune their hyperparameters, as well as explore other methods for preprocessing the data. Additionally, it may be worth considering other machine learning models or ensembles of models, as these may perform better than any of the models considered here. Ultimately, the best model for a particular use case will depend on the specific requirements and characteristics of the data, and the choice of model should be guided by a thorough evaluation of its accuracy using relevant metrics.

Table 2. Performance comparison of different models

	MSE	Mean AE	Median AE	RMSE
ARIMA-GARCH	0.9155	0.7081	0.5905	0.9568
Exp Smoothing	0.5845	0.6281	0.4494	0.7645
KNN	0.5717	0.6285	0.4483	0.7561
Prophet	1.3783	0.8997	0.6611	1.1740
LSTM	0.9454	0.8047	0.7737	0.9723

5. Conclusion & Future works

In this study, we compared the performance of several time series forecasting methods for predicting the inflation rate in the USA. The methods included ARIMA, Exponential Smoothing, KNN, Prophet, and LSTM. The comparison was based on evaluation metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), Median Absolute Error (Median AE), and Root Mean Squared Error (RMSE).

The results showed that the KNN algorithm had the lowest MSE and RMSE, indicating that it performed the best in terms of accuracy. On the other hand, the Prophet model had the highest MSE and RMSE, making it the worst performing method among the ones tested.

Based on the results, we can conclude that KNN is the best method for short-term forecasting accuracy among the methods tested. However, it is important to keep in mind the limitations of this study, such as the limited data used and the possible lack of generalizability to other countries or economic indicators.

Future work can expand the scope of the study by applying these methods to other countries and other economic indicators, and comparing their performance. Additionally, other time series forecasting methods can be considered and compared to determine the most suitable method for predicting economic indicators.

It is also important to consider the limitations of this study. First, the data used in this study is limited to the USA inflation rate, and the results may not be generalizable to other countries or other economic indicators. Second, the methods used in this study may not be suitable for predicting inflation in other countries or for other economic indicators, as different methods may perform better or worse for different types of data.

In order to further improve the results, it is recommended to perform fine-tuning of the hyperparameters for each algorithm and to consider using pipelines and grid search techniques for model selection. An extension of this work to study the violation of homoscedasticity by the errors of the model obtained would also be appropriate and represent another field of analysis.

In addition to that, an extension of this work to study the violation of homoscedasticity by the errors of the model obtained would be appropriate and represent another field of analysis. In addition, we highly recommended further research to compare the advanced machine learning algorithms with the classical algorithms, especially with the ARIMA approach in order to choose the best program in terms of results and predictive performance.

Acknowledgement

We thank the anonymous reviewers and the editorial team for their careful reading of our manuscript and their many insightful comments and suggestions.

REFERENCES

1. N. G. Mankiw, Principles of microeconomics, 8th ed. CENGAGE Learning Custom Publishing, 2016.
2. M. Friedman, The Counter-Revolution in Monetary Theory. In: Wood, G.E. (eds) Explorations in Economic Liberalism. Palgrave Macmillan, London, 1996.

3. J. Youness, M. Driss, An ARIMA Model for Modeling and Forecasting the Dynamic of Univariate Time Series: The case of Moroccan Inflation Rate, International Conference on Intelligent Systems and Computer Vision (ISCV), Fez, Morocco, 2022.
4. L. S. Goodman, Book Review: Smoothing, Forecasting, and Prediction of Discrete Time Series. Journal of Marketing Research, 2(3), 314–315, 1965.
5. Rob J. Hyndman and Anne B. Koehler, Another look at measures of forecast accuracy, International Journal of Forecasting, Volume 22, Issue 4, Pages 679-688, 2006.
6. José F. Torres, et al. Deep Learning for Time Series Forecasting: A Survey. Big Data.3-2.3-21, 2021.
7. J. Youness and M. Driss, LSTM Deep Learning vs ARIMA Algorithms for Univariate Time Series Forecasting: A case study, 2022 8th International Conference on Optimization and Applications (ICOA), Genoa, Italy, pp. 1-4, 2022.
8. Jouilil Y and Mentagui D. Comparing The Forecasting Accuracy Metrics Of Support Vector Regression And ARIMA Algorithms For Non-Stationary Time Proces. Mathematics and Statistics, 2023.
9. Abdullayeva, F., and Imamverdiyev, Y. Development of Oil Production Forecasting Method based on Deep Learning. Statistics, Optimization & Information Computing, 7(4), 826-839, 2019.
10. Thormann, M.-L., Farchmin, J., Weisser, C., Kruse, R.-M., Säfken, B., and Silbersdorff, A. *Stock Price Predictions with LSTM Neural Networks and Twitter Sentiment*. Statistics, Optimization & Information Computing, 9(2), 268-287, 2021.
11. Achmad Muchayan, Comparison of Holt and Brown's Double Exponential Smoothing Methods in The Forecast of Moving Price for Mutual Funds. Journal of Applied Science, Engineering, Technology, and Education Vol. 1 No. 2, 2019.
12. Bollerslev, T. Generalized autoregressive conditional heteroskedasticity. Journal of Econometrics, 31, 307–327, 1986.
13. Ionut-Cristian Baciu, Stochastic models for forecasting inflation rate. Empirical evidence from Romania, 7th International Conference on Globalization and Higher Education in Economics and Business Administration, GEBA 2013, 2015.
14. Board of Governors of the Federal Reserve System. Inflation Data, 2021.
15. Livia Paranhos, Predicting Inflation with Recurrent Neural Networks. R&R at the International Journal of Forecasting, 2022.
16. Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," arXiv preprint arXiv:1412.3555, 2014.
17. Martin Längkvist, Lars Karlsson, and Amy Loutfi, "A Review of Unsupervised Feature Learning and Deep Learning for Time-Series Modeling," Pattern Recognition Letters, vol. 42, pp. 11-24, 2014. DOI: 10.1016/j.patrec.2014.01.025.
18. Lipton, Zachary C., John Berkowitz, and Charles Elkan, "A Critical Review of Recurrent Neural Networks for Sequence Learning," arXiv preprint arXiv:1506.00019, 2015.
19. Sezer, O. B., Gudelek, M. U., and Ozbayoglu, A. M., "Financial time series forecasting with deep learning: A systematic literature review: 2005–2019," Applied soft computing, 90, 106181, 2020. DOI: 10.1016/j.asoc.2020.106181.
20. Youness, J., Driss, M. Mathematical Modeling of Financial Time Series Volatility: A GARCH Model. In: Farhaoui, Y., Rocha, A., Brahmia, Z., Bhushab, B. (eds) Artificial Intelligence and Smart Environment. ICAISE 2022. Lecture Notes in Networks and Systems, vol 635. Springer, Cham, 2023.
21. Iaousse, M., Jouilil, Y., Bouincha, M., and Mentagui, D. A Comparative Simulation Study of Classical and Machine Learning Techniques for Forecasting Time Series Data. International Journal of Online and Biomedical Engineering (iJOE), 19(08), pp. 56–65, 2023.