Consumer Price Index Forecasting Based on Univariate Time Series and a Deep Neural Network

Reynaldo Rosado, Aldis Joan Abreu, José C. Arencibia,
Hector Gonzalez(✉), and Yanio Hernandez

Universidad de las Ciencias Informaticas (UCI), La Habana, Cuba
{rrosado,ajabreu,jcarencibia,hglez,yhernandezh}@uci.cu

Abstract. The global Consumer Price Index (CPI) is a monthly record, which allows measuring the variation of the final consumer prices of a given set of goods and services of households living in a given geographic region, city or country. The present work addresses the problem of CPI forecasting using different Long Short-Term Memory (LSTM) neural network architectures according to state of the art in time series forecasting. Univariate time series data are mapped by a multivariate spatiotemporal representation using a set of Box-Jenkins functions and a time window. Next, a Convolutional Neural Network (CNN) with a specific droop out function combines the feature set to make a more discriminative representation space of the multivariate data. Finally, a LSTM exploits the temporality relationship among the data. The pipeline results, by combining a CNN with a LSTM, showed an improvement in forecasting CPI time series over Ecuador available dataset with respect to other LSTM-based architectures and models.

Keywords: Consumer Price Index · Time series forecasting · Long Short-Term Memory

1 Introduction

The Consumer Price Index (CPI) reflects the variation in the prices of household products and services in a given period of time. It constitutes one of the most important economic indicators for any country. Through it, the inflationary processes of economies can be measured. The CPI is systematically way taken as a reference for decision-making regarding monetary policies by governments and financial entities. It is also used for various aspects of social finance, such as retirement, unemployment and government financing [18].

Both the prices of the products and services that give origin to the CPI estimate, as well as the CPI itself, are calculated systematically, so they are time

This work has been partially funded by FONCI through project: Plataforma para el análisis de grandes volúmenes de datos y su aplicación a sectores estratégicos.

Y. Hernández Heredia et al. (Eds.): IWAIPR 2021, LNCS 13055, pp. 33–42, 2021.
https://doi.org/10.1007/978-3-030-89691-1_4

series data type. As CPI forecast helps to estimate future trends, it is key for decision making. Moreover, it allows the application of price stabilization policies to reduce the economic impact on the prices of products and services demanded by consumers. In those economies that present instability, CPI data fluctuate over time, which translates into a non-linear and non-stationary behavior [13].

Traditionally, time series forecasting is performed using well-established statistical tools such as the Box-Jenkins Auto-Regressive models or the Holt-Winters approach to exponential smoothing [4,8]. Since the computational point of view, the problem of measuring, estimating or forecasting CPI, not the CPI problem inself, has been approached as a univariate time series for the study of the global metric, or as a multivariate problem when the study is extended to the set of products or services that are included in the basic family basket. The most popular statistical method used for CPI forecasting, as a univariate time series problem, has been the autoregressive family of algorithms known as ARIMA [1,2]. Recently, there are several approaches that introduce modern techniques with better accuracy results, however, it is still a tendency to compare the results with ARIMA [12]. The use of machine learning techniques for CPI forecasting has emerged in recent references. Artificial Neural Networks (ANN) [6] and Support Vector Machines (SVM) [15] stand out as learning techniques in the CPI forecast.

The introduction of Deep Neural Networks in the time series forecasting has been widely studied in recent years with results that establish the state of the art in several problems. Particularly the use of ANN architectures with the presence of recurrent mechanisms such as RNN or LSTM shows the best results [9,10,16]. These neural network models have the capability of capturing the temporal dependence in the data, and, at the same time, they are flexible to the forecast of more than one output variable, which corresponds to the multivariate problem. According to the study consulted, this models are more widely used to study the CPI forecast as a univariate problem, as compared to deep networks and recurrent models. Two examples to use a simple LSTM model, are the comparative empirical study of Ecuador with temporal data between 2005–2020 CPI [14] and Indonesia CPI where several optimization approach with LSTM are compared [18].

In the empirical CPI studies of Indonesia and Ecuador mentioned above, the models used are based on classical regression approaches in which the parameters are not tuned, so that possible improvements in accuracy should be expected. Likewise, there are limitations in the representation of the set of attributes, so that the possible temporal non-linearity inherent in the data is not handled, or the attributes are not transformed into spaces with greater discriminative power. Finally, the regularization approach is not clear in either paper. Especially in the LSTM approach, the authors do not combine several interesting architectures such as stacked LSTMs, a CNN combined with LSTMs or, bidirectional LSTMs among other approaches.

The aim of this paper is to develop an empirical evaluation of different models and architectures of LSTM-based deep neural networks, that allow solving the CPI forecasting problem with higher accuracy on the available data set from

Ecuador. In our proposal, the set of attributes must be represented with an adequate treatment of the non-linearity in the time series relationship. On the other hand, the model should consider a droop out schemes that ensure generalization in the learning process and select the most relevant features.

2 General Notation and State of the Art

A univariate time series is defined as a collection of values of a given variable, ordered chronologically and sampled at constant time intervals. Whenever a variable is spatially related and individually shows a temporal relationship, we say that the problem is a multivariate time series. Classical statistical or machine learning models need to consider the univariate or multivariate problem differently, however deep learning models can handle both indistinctly with high accuracy. Time series are usually characterized by three components: trend, seasonality and residuals [16]. In real-world time series and, in particular the CPI problem, seasonality can be affected by external agents such as the economic and financial crisis, prices of the main products in the world market, and emerging situations such as the COVID-19 pandemic.

Let $X = \{x_1, \ldots, x_T\} \in \mathbb{R}$, be a chronological ordered value. For a temporal window of size h, which considers a low seasonality of the problem, each training instance is written as $\left(\boldsymbol{x}_j, \boldsymbol{y}_j\right) \in \mathbb{R}^p \times \mathbb{R}^q$, where the input variables are $\boldsymbol{x}_j = \{x_{j-1}, x_{j-2}, \ldots, x_{j-h}, \boldsymbol{g}(x_{j-1}, \ldots, x_{j-h})\}$, with $\boldsymbol{g} = \{g_1, \ldots, g_r\}$ a family of the Box-Jenkins non-linear functions [1], and the outputs variables $\boldsymbol{y}_j = \{x_j, x_{j+1}, \ldots, x_{j+q}\}$. Finally, the corresponding time series forecasting problem consists of the estimating a predictor $F : \mathbb{R}^p \rightarrow \mathbb{R}^q$ in such a way that the expected deviation between true and predicted outputs is minimized for all possible inputs.

2.1 LSTM in Time Series Forecasting

In a recent review article [10], long-term time series forecasting based on LSTM models is discussed in more detail. The main contribution of this model to recurrent architectures such as RNNs [11,17] is in the solution of the optimization problem, where classical activation functions tend to gradient vanishing in interactive propagation to capture long-term dependence. The Gated Recurrent Unit (GRU) [3] is the newest generation of RNNs and is quite similar to an LSTM. The main difference between a GRU and an LSTM is that a GRU has gate, an update, and reset gate; while an LSTM has three gates: an input, a forget, and an output gate, which allow for changes in the state vector of a cell while capturing the long-term temporal relationship. When the time series is small, GRU is suggested; on the other hand, if the series is large, it must be LSTM. GRU checks in each iteration and can be updated with short-term information, however LSTM limits the change gradient in each iteration and in this way does not allow the past information to be completely discarded, this is why LSTM is mostly used for the long-term dependency modeling. In [10] he states that there are no significant advantages with respect to the computation time of GRU over

LSTM, although it has a smaller number of parameters in the cells. An additional advantage of the use of LSTM cells is in the incorporation of filters in the input that allows removing unnecessary information. For this reason, the present work proposes the use of cellular architectures based on LSTM.

2.2 Conventional LSTM Architectures

The most simple LSTM model is Vanilla, that has a single hidden layer of units of this type and an output layer used to make a prediction. One of its advantages is its application in time series, given by the fact that its sequence prediction is a function of the previous steps. The use of a simple architecture with an input layer, a hidden cell, and an output layer is effective for prediction problems with short sequences.

Bidirectional LSTM

There are problems in the field of Natural Language Processing (NLP) where, in order to predict a value of a sequence of data at a given time instant, information is needed from the sequence both before and after that instant. Bidirectional Recurrent Neural Networks (BRNN) address this point to solve this type of problem. Their main limitation with BRNNs is that the entire data set is needed beforehand to make the prediction, unlike standard networks that compute the activation values of the hidden units using a one-way feedforward procedure. In a BRNN, information from the past, present and future is used as input for prediction by means of a forward and backward process. Figure 1a show an example of this architecture.

Stacked LSTM

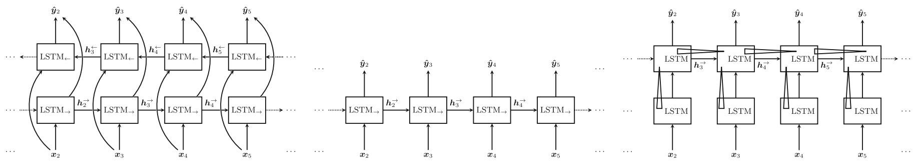
This extension has LSTM layers where each layer contains multiple memory cells. Stack-type architecture is composed of several hidden layers of LSTM memory blocks, and, in some cases MLP layers. For this type of deep architecture good results are recognized for solving problems of high level of complexity. In this type of network, each layer gradually solves a part of the prediction and then passes it to the next layer, until the output information is obtained. A simple example of this architecture is shown in Fig. 1b.

CNN LSTM

Convolutional Neural Networks (CNNs) are one of the most common architectures used in image processing and computer vision. At the same time, convolutional LSTM networks are also suitable for modeling multiple quantities, e.g., spatially and temporally distributed relationships, due to their characteristic properties.

CNNs have three types of layers: convolutional, clustering, and fully connected. The core work of the convolution layers is the learning of features from the input data. For this purpose, filters of a predefined size are applied to the data using the convolution maneuver between matrices. Convolution is the addition of all the products of the features.

Pooling reduces the size of the input, which speeds up the computation and avoids overfitting. The most popular pooling methods are average pooling and maximum pooling, which summarize the values using the average or maximum value, respectively. Once the features have been extracted by the convolutional layers, the prediction is carried out using fully connected layers. The input data for the latter fully connected layers are the attenuated features resulting from the convolutional layers and the dense layers. See example in Fig. 1c.



(a) Bidirectional LSTM (b) Stacked LSTM (c) Convolutional LSTM

Fig. 1. Architectures of the LSTMs consider in our proposal.

3 Results and Discussion

Results of time series forecasting of the Ecuador CPI and main goods and services are compared in terms of graphs and different classification metrics. Forecasted general CPI and the most ten relevant products are graphically compared in the next sections. Comparison is driven with respect to Stacked, Bi-directional and two Convolutional LSTMs. Additionally, the experimental setup is described, and the main results of the LSTMs models discussed. In the first place, we present technical details related to the datasets, parameter setup and implementations. Finally, a detailed comparison between the LSTMs approach and the machine learning algorithms proposal [14]. Also, the empirical evaluation of the forescasted LSTMs models over the ten critical goods and services gives a fine multivariate solution to CPI problem.

Datasets

The dataset used for the experiments are taken from the Ecuador CPI in the official governmental website https://www.ecuadorencifras.gob.ec. Our study is limited to using LSTMs architectures in univariate ahead forecasting considering only single seasonality, to be able to straightforwardly compare against automatic standard benchmark methods over mentioned datasets as an update[14]. In Table 1, the basic statistics information of general CPI and the ten goods and services considered in the experimentation are shown. The time interval considered for the general CPI is from January 2005 to March 2021, and for goods and services from January 2015 to March 2021. A more detailed characterization of the dataset related to trend, seasonality, and residuals measures [14].

Table 1. Basic information of datasets

Datasets	Mean	Std
General	91,59	13,42
Food and non-alcoholic drinks	105,95	1,51
Alcoholic drinks, tobacco and drugs	125,65	8,33
Clothing and shoes	88,89	5,64
Accommodation, water, electricity, gas and other fuels	113,20	3,21
Furniture and household items	102,64	1,71
Health	108,84	3,98
Transportation	106,31	1,52
Communications	94,95	0,95
Recreation and culture	98,48	1,77
Education	109,11	4,65
Restaurants and hotels	108,18	1,40
Goods and other services	106,34	1,44

Implementation Details

For the experiment, we design two strategies to split the data consequently with the range of dataset employed [14]. In the first one, the predictive model was trained with all CPI data except for the last 12 months, selecting data in the range since January 2005 to June 2019 for model training, and data from July 2019 to June 2020 as the test dataset. The goal of this approach was to compare our results to the machine learning algorithms used in the previous work. Secondly, all data was employed to compare the performance of the LSTMs approaches. The experiment was implemented and run in Python using available libraries to work with deep neural network.

The optimization of the models will determine their quality, and it must be performed based on the adjustment of its hyper-parameters, obtaining the best fit of the model that provides the most accurate results. In deep learning, there are two types of hyper-parameters: model parameters and optimization parameters. Model parameters must be adjusted in the model definition to obtain optimal performance. Optimization parameters are adjusted during the training phase of the model by using the dataset. The number of hyper-parameters used in a predictive model depends on the architecture and layers used in the implementation of each algorithm. One of the most common hyper-parameters used on LSTM algorithms is the number of nodes or neurons that LSTM layer has, therefore we choose this parameter for searching its best fit in order to obtain the best model optimization. For searching the best hyper-parameter optimization, we used a two-level heuristic search. In the first level of search, values between 100 and 1000 nodes were used, with an increase of 100 nodes per iteration, obtaining the range of nodes with the lowest RMSE as the first level search results. For the second search level, values in the best range obtained in the previous level were chosen, with an increment of 10 per iteration, finally obtaining the number of nodes with the lowest RMSE and best precision as the result of the complete search.

Performance Measures

As in other similar works, we use two common metrics, the Root Mean Squared Error (RMSE) and Mean absolute Percentage Error(MAPE). Given a test set, D_{test}, and a predictor, h, this measures are given as

$$RMSE(h; D_{test}) = \sqrt{\frac{\sum_{(x,y)\in D_{test}} (h(\boldsymbol{x}) - \boldsymbol{y})^2}{N}} \tag{1}$$

$$MAPE(h; D_{test}) = \frac{1}{N} \sum_{(x,y)\in D_{test}} \frac{|\,(h(\boldsymbol{x}) - \boldsymbol{y})\,|}{|\boldsymbol{y}|} \tag{2}$$

In the second experiment, we use the Friedman test procedure with the corresponding post-hoc to compare algorithms over multiple datasets as recommended by Demsar's method [5] and its extensions [7].

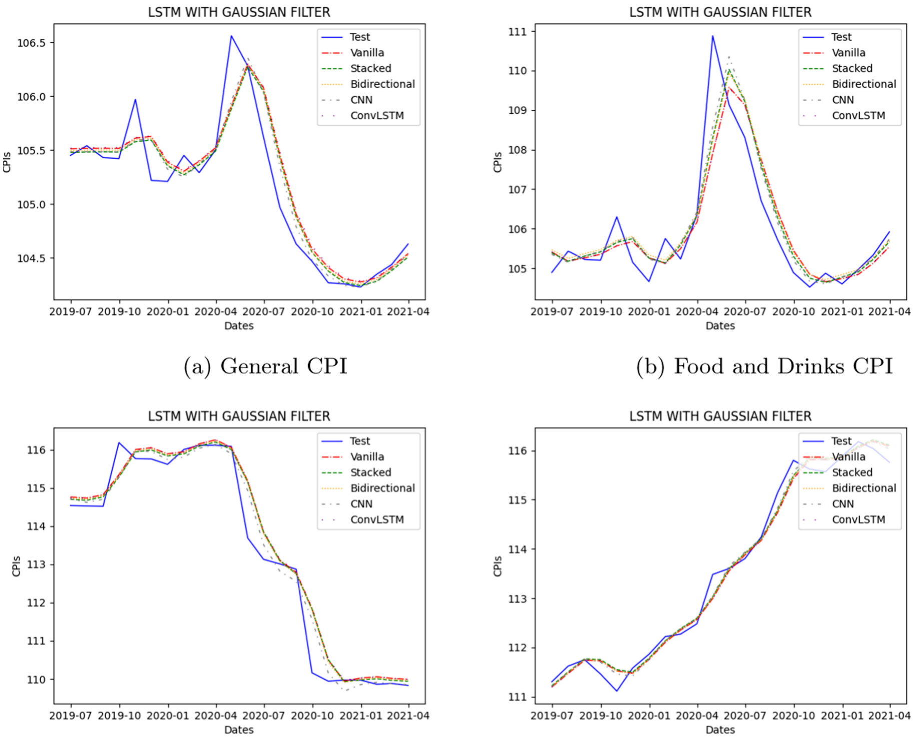
Benchmarks

In Table 2, we indicate the results of comparing the predictions in the July 2019 to June 2020 range to previous experimentation. The best performance with the MAPE metric is found in the Bidirectional LSTM (marked in boldface), and the poorest LSTMs results is better than the best of the previous research. These results suggest that the use of an LSTM approach to the general CPI problem in Ecuador is an accurate one and improves the results obtained in previous research.

Table 2. Mean MAPE results and last year estimation for each LSTMs and benchmark models.

Month	Test	SVR Poly	SVR RBF	LSTM	SARIMA (MAPE)	Exp. Smt	Prophet	LSTM Vanilla	LSTM Stacked	LSTM Bid	LSTM CNN	LSTM Conv
2019-07	105,54	105,35	105,26	105,51	105,92	105,37	105,98	105,52	105,54	105,48	105,51	105,63
2019-08	105,43	105,36	105,20	105,57	105,98	105,40	105,89	105,52	105,54	105,48	105,51	105,63
2019-09	105,42	105,36	105,13	105,61	106,00	105,59	105,99	105,51	105,54	105,48	105,51	105,63
2019-10	105,97	105,36	105,06	105,61	106,09	105,55	105,91	105,63	105,66	105,60	105,63	105,74
2019-11	105,22	105,36	104,99	105,61	106,24	105,47	105,89	105,63	105,66	105,60	105,63	105,75
2019-12	105,21	105,35	104,91	105,64	106,39	105,61	105,89	105,34	105,37	105,31	105,33	105,49
2020-01	105,45	105,34	104,82	105,67	106,47	105,87	106,14	105,27	105,30	105,24	105,26	105,41
2020-02	105,29	105,34	104,73	105,64	106,51	105,89	106,30	105,41	105,43	105,37	105,40	105,52
2020-03	105,50	105,32	104,64	105,64	106,58	106,01	106,48	105,55	105,58	105,52	105,55	105,65
2020-04	106,56	105,31	104,55	105,67	106,71	106,28	106,71	106,00	106,04	105,99	106,01	106,06
2020-05	106,28	105,29	104,45	105,68	106,86	106,21	106,57	106,42	106,46	106,41	106,43	106,47
2020-06	105,62	105,27	104,35	105,69	106,95	106,01	106,41	106,14	106,16	106,10	106,13	106,25
MAPE		3,2E−03	7,4E−03	3,0E−03	7,3E−03	2,9E−03	5,4E−03	2,1E−03	2,2E−03	**2,0E−03**	2,1E−03	2,6E−03

In the LSTMs comparative models, we consider a Gaussian filter in the input layer for all models in order to capture the non linearity of the data. As shown in Table 3, the Gaussian filter in the input layer increase, the performance of the LSTMs models for the General CPI Ecuador dataset, and particularly for each good and service. Figure 2 shows the forecasting of General CPI and other three datasets over test set.



(a) General CPI

(b) Food and Drinks CPI

(c) Education CPI

(d) Health CPI

Fig. 2. Examples of the CPI forecasting of general dataset and three goods and services.

Besides, we compare the results of different algorithms, and, in general, the combination of CNN with LSTM takes advantage of the filter. We use the Friedman test to compare the relative performance of the different LSTMs architectures against each other across all the datasets in terms of RMSE error metric. The relative goodness of each of the five variants considered can be graphically observed in Fig. 3 that corresponds to the result of the Friedman test with Shaffer post-hoc. Even though we have significance differences in the combination of CNN-LSTM neural network respect to rest of LSTMs variants. These results can be seen as very encouraging specially taking into account that we consider in input layer a non-linear transformation with Gaussian filter.

Table 3. Mean RMSE results with Gaussian filters (right columns) and without Gaussian filters (left columns) in the LSTMs models.

Goods and Services	LSTM Vanilla	LSTM Stacked	LSTM Bid	LSTM CNN	LSTM Conv	LSTM Vanilla	LSTM Stacked	LSTM Bid	LSTM CNN	LSTM Conv
General	0,399	0,391	0,397	0,410	0,400	0,250	0,244	0,243	**0,223**	0,257
Food and non-alcoholic drinks	1,305	1,306	1,325	1,376	1,302	0,779	0,726	0,756	**0,708**	0,809
Alcoholic drinks, tobacco and drugs	0,524	0,561	0,568	0,524	0,544	0,415	0,360	0,393	**0,274**	0,481
Clothing and shoes	0,511	0,491	0,515	0,551	0,517	0,223	0,222	0,222	**0,221**	0,224
Accommodation, water, electricity, gas and other fuels	0,159	0,158	0,152	0,156	0,156	0,124	**0,076**	0,099	0,083	0,140
Furniture and household items	0,303	0,297	0,305	0,311	0,304	0,162	0,156	0,159	**0,155**	0,168
Health	0,360	0,357	0,362	0,366	0,363	0,220	0,216	0,216	**0,202**	0,228
Transportation	1,165	1,159	1,185	1,264	1,171	0,592	0,593	0,593	**0,566**	0,605
Communications	0,575	0,574	0,580	0,638	0,577	0,334	0,362	0,330	**0,297**	0,343
Recreation and culture	0,925	0,915	0,934	0,961	0,923	0,553	0,560	0,548	**0,521**	0,552
Education	0,928	0,912	0,909	0,931	0,928	0,561	0,567	0,553	**0,488**	0,573
Restaurants and hotels	1,093	1,091	1,104	1,227	1,080	0,601	0,592	0,592	**0,559**	0,601
Goods and other services	0,406	0,395	0,400	0,404	0,420	0,259	0,256	0,262	**0,236**	0,291

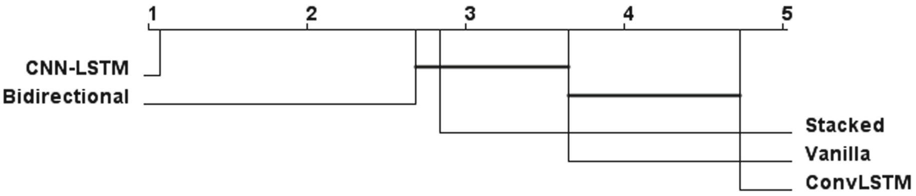


Fig. 3. Results of the Friedman test with Shaffer correction: $p = 1.33e - 7 < 0.05$

4 Concluding Remarks and Further Work

An attempt is made to improve CPI forecasting based on different LSTM architectures of Neural Networks. The application of a Gaussian filter in the input of these architectures has led to competitive results in the preliminary experimentation performed. The proposal was compared using a General CPI Database and specific goods and services of Ecuador, available in previous works. The results of the five LSTM architectures used improve the previous machine learning models, with respect to MAPE and RMSE metrics. The proposal was enriched with the application of a Gaussian filter in the input of all the implemented LSTM variants. The most efficient result is the combination of LSTM with CNN, with an RMSE of 0.243. Future work is being planned in several directions. On the one hand, work is being done to compare this experiment with CPI forecasts from other countries. On the other hand, different optimization schemes can be adopted to improve efficiency and performance. In addition, it is possible to build a multivariate model for CPI forecasting that has as input the prices of different products and services, and to do a simultaneous estimation of those prices.

References

1. Box, G.E.P., Jenkins, G.M.: Time Series Analysis: Forecasting and Control. Holden-Day, San Francisco (1976)
2. Brockwell, P.J., Davis, R.A.: Time Series: Theory and Methods (1987)
3. Chung, J., Gulcehre, C., Cho, K., Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling. In: NIPS 2014 Workshop on Deep Learning, December 2014 (2014)
4. Collins, S.: Prediction techniques for box-cox regression models. J. Bus. Econ. Stat. **9**(3), 267–277 (1991)
5. Demšar, J.: Statistical comparisons of classifiers over multiple data sets. J. Mach. Learn. Res. **7**, 1–30 (2006)
6. Fauzan, M., et al.: Epoch analysis and accuracy 3 ANN algorithm using consumer price index data in Indonesia. In: 3rd International Conference of Computer, Environment, Agriculture, Social Science, Health Science, Engineering and Technology, pp. 1–7 (2018)
7. García, S., Fernández, A., Luengo, J., Herrera, F.: A study of statistical techniques and performance measures for genetics-based machine learning: accuracy and interpretability. Soft Comput. **13**(10), 959–977 (2009)
8. Granger, C.W.J., Newbold, P.: Forecasting transformed series. J. Roy. Stat. Soc.: Ser. B (Methodol.) **38**(2), 189–203 (1976)
9. Lim, B., Zohren, S.: Time series forecasting with deep learning: a survey. arXiv preprint arXiv:2004.13408 (2020)
10. Lindemann, B., Müller, T., Vietz, H., Jazdi, N., Weyrich, M.: A survey on long short-term memory networks for time series prediction. Proc. CIRP **99**, 650–655 (2021)
11. Mikolov, T., Karafiát, M., Burget, L., Černockỳ, J., Khudanpur, S.: Recurrent neural network based language model. In: Eleventh Annual Conference of the International Speech Communication Association (2010)
12. Mohamed, J.: Time series modeling and forecasting of Somaliland consumer price index: a comparison of ARIMA and regression with ARIMA errors. Am. J. Theoret. Appl. Stat. **9**(4), 143–153 (2020)
13. Qin, X., Sun, M., Dong, X., Zhang, Y.: Forecasting of China consumer price index based on EEMD and SVR method. In: 2018 2nd International Conference on Data Science and Business Analytics (ICDSBA), pp. 329–333. IEEE (2018)
14. Riofrıo-Valarezo, J., Peluffo-Ordónez, D.H., Chang, O.: Forecasting consumer price index (CPI) of Ecuador: a comparative study of predictive models. Int. J. Adv. Sci. Eng. Inf. Technol. **10**(3), 1078–1084 (2020)
15. Rohmah, M.F., Putra, I.K.G.D., Hartati, R.S., Ardiantoro, L.: Predicting consumer price index cities and districts in East Java with the Gaussian-radial basis function kernel. J. Phys.: Conf. Ser. **1456**, 012026 (2020)
16. Torres, J.F., Hadjout, D., Sebaa, A., Martinez-Alvarez, F., Troncoso, A.: Deep learning for time series forecasting: a survey. Big Data **9**(1), 3–21 (2021)
17. Williams, R.J., Zipser, D.: A learning algorithm for continually running fully recurrent neural networks. Neural Comput. **1**(2), 270–280 (1989)
18. Zahara, S., Ilmiddaviq, M.B., et al.: Consumer price index prediction using long short term memory (LSTM) based cloud computing. J. Phys.: Conf. Ser. **1456**, 012022 (2020)