Jurnal Matematika Vol. 14, No.1, Juni 2024, pp. 37-51 ISSN: 1693-1394
Article DOI: 10.24843/JMAT.2024.v14.i01.p170

A Comparative Analysis of Deep Autoregressive, Deep State Space, Simple Feed Forward, and Seasonal Naive in Forecasting Indonesia's Inflation Rate

I Wayan Sumarjaya
Department of Mathematics, FMIPA, Universitas Udayana, Indonesia
e-mail: sumarjaya@unud.ac.id

Made Susilawati
Department of Mathematics, FMIPA, Universitas Udayana, Indonesia
e-mail: mdsusilawati@unud.ac.id

Abstract**:** *Information about inflation plays an important role in economic policy. The government of the Republic of Indonesia has put a great deal of effort into controlling the inflation rate. This research aims to forecast Indonesia's inflation rate using deep* autoregressive *and to compare it with other models such as deep state space, simple feed forward, and seasonal naive. This study compares eighteen deep autoregressive network models. Each model differs only in its hyperparameter settings such as the number of epochs, the number of layers, the number of cells, and the number of batch sizes. In order to check for consistency each model was replicated ten times. In total there are 180 runs for each configuration including the replication. Similarly, each of the deep state space and simple feed forward model was replicated ten times to ensure consistency. The seasonal naive, however, did not need this replication. The results showed that the deep autoregressive model with 50 epochs, 4 layers, 40 cells, 32 batch sizes produced the smallest root mean squared error at 0.218565. This root mean squared error was also the smallest among the other models such as deep state space (0.28734), simple feed forward (0.350449), and seasonal naive (0.336056). Hence, deep autoregressive was the preferred forecasting model. In conclusion, using deep autoregressive the median forecasts fluctuated but below 1 percent.*

Keywords*: deep autoregressive networks, deep learning, deep state space, simple feed forward, seasonal naive, Indonesia inflation rate forecasting*

1. Introduction

One of the key indicators in the economic stability of a country is inflation. According to Oner (2022) "inflation has plunged countries into long periods of instability". This suggests that information about inflation is important not only because of its role in understanding the economy, but also the key to understand the future economic stability of a country.

Forecasting Indonesia's inflation rate has been an active research area during the last decade. The time series methodology used in forecasting the inflation rate can be classified into three: statistical-based methods, fuzzy logic-based methods, and machine learning methods. Popular statistical-based methods for Indonesia's inflation rate forecasting include autoregressive integrated moving average (ARIMA) such as the works of Hartati (2017), Yusnitasari (2020), Melyani et al. (2021), Qalbi et al. (2021), Amaly et al., (2022), Saragih & Sembiring (2022), Asmarani (2023), Nanlohy & Loklomin (2023), and Muslihin & Ruchjana (2023) among others; exponential smoothing such as the works of Rismawanti & Darsyah (2018), Purwanti & Purwadi (2019), Sudibyo et al., (2020), and Saragih & Sembiring (2022) among others. In the ARIMA methodology the steps usually include routine such as specifying the order of the ARIMA, model selection, model diagnostic, and forecasting. The specification of the order of the ARIMA models can be problematic since there may be many candidate models whereas in exponential smoothing the selection of optimal smoothing coefficient can be obtained through trial and error.

The second popular method used in forecasting the inflation rate in Indonesia is the fuzzy time series method. Researchers in this area include the works of Udin et al. (2020), Qalbi et al. (2021), Kadry et al. (2022), and Fireza & Ahmad (2023). Fuzzy time series rely on correct fuzzy logical relationship.

The third popular method is machine learning which usually deploys conventional neural networks such as the works of Estiko & S. (2019), Rifa'i (2021), Amaly et al. (2022), and Wiranto & Setiawan (2023) among others. The conventional neural networks require authors to manually set the activation functions, hidden layers, and other settings to obtain optimum results.

Machine learning is a computer algorithm that runs on a computer that can learn patterns from data. One of the purposes of machine learning is to make a prediction. This prediction is commonly done through artificial neural networks (or just neural networks).

Neural networks try to mimic how the human brain works. The conventional machine learning does this by constructing pattern recognition. This conventional machine learning usually involves data transformation, selecting activation functions, determining the number of hidden layers, and determining the architecture of neural networks. Domain expertise is often required in this case. In the conventional machine learning, the activation functions give nonlinearity to neural networks, while the hidden layers serve as input for the next layers. The architectures of neural networks provide means of various types of learning.

Recent advances in the machine learning, especially deep learning, have allowed researchers to make use of a variety of data types such as images, text, audio, and video to be predicted. The popularity of deep learning stems from its ability to learn from data with minimum human intervention. The various studies cited in Ao & Fayek (2023) suggest the superiority of the deep learning over the conventional machine learning. This superiority is achieved through deploying hundreds of thousands of hidden layers. One particular feature that distinguishes the deep learning from the conventional machine learning is automatization. The conventional machine learning techniques have limitations in processing natural data in their raw form (LeCun et al., 2015). In the deep learning the focus is on tuning the hyperparameters of the proposed model.

Due to its nature, the order of time series data is preserved, for instance the data is not exchangeable. One particular neural networks architecture that is suitable for time series data is recurrent neural networks (RNN). However, RNN suffers from the vanishing gradient problem (Van Houdt et al., 2020). Modern RNN architectures that can fix the vanishing gradient problem are long short-term memory (LSTM) and gated recurrent unit (GRU). LSTM has been proven to be more effective than conventional RNN (LeCun et al., 2015). Furthermore, LSTM is the RNN architecture of choice for probabilistic deep learning models such as deep autoregressive (DeepAR) and deep state space (DeepSS).

The deep autoregressive (DeepAR) of Salinas et al. (2020) aims at producing accurate probabilistic forecast. The DeepAR methodology relies on training autoregressive recurrent networks. These recurrent networks are usually of LSTM-type. However, a simpler version of LSTM called gated recurrent unit (GRU) is also possible. Thus, the type of recurrent networks in DeepAR is either LSTM or GRU. Free and open-source software such as Python-based GluonTS of Alexandrov et al. (2019) and Alexandrov et al. (2020) provides full support for the two types of RNN.

The article is organized as follows. The first section provides motivation for forecasting monthly Indonesia's rate of inflation and surveys current research about methodologies for forecasting inflation. The next section discusses DeepAR methodology and some other competing models. Section three provides results and discussions. Section four concludes the article.

2. Research Methodology

Salinas et al. (2020) proposed deep autoregressive (DeepAR) that aims at producing accurate probabilistic forecast. The DeepAR methodology relies on training autoregressive recurrent networks. The model learns a global model from historical data of all time series data in the dataset (Salinas et al., 2020). More specifically, the model utilizes RNN architecture such as LSTM or GRU. In this research LSTM is preferred over GRU for its ability in capturing long-term dependency. See also studies in Irie et al. (2016) and Cahuantzi et al. (2023).

Figure 1 shows an LSTM cell with input gate (i_t), forget gate (f_t), and output gate (o_t).

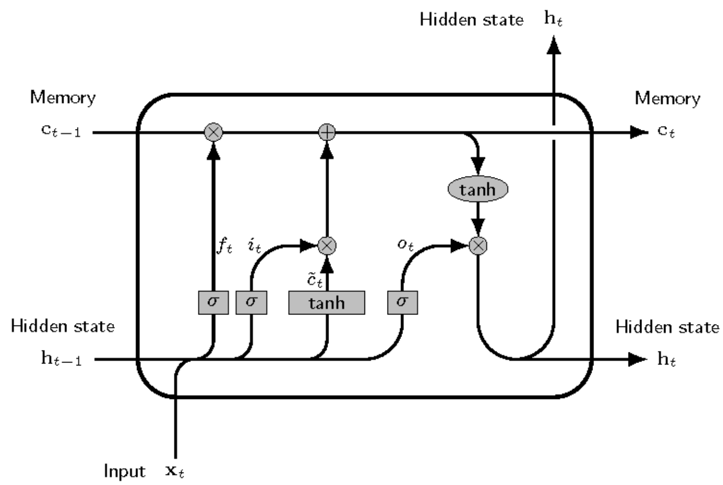

Figure 1. An LSTM cell with input gate, forget gate, and output gate.

As can be seen from Figure 1, the values of input gate (i_t), forget gate (f_t), and output gate (o_t) are computed by three connected layers with sigmoid activation functions σ (Zhang et al., 2023). The tanh activation function ensures the hidden state h_t takes the value $(-1,1)$. For more details about LSTM see for example Van Houdt et al. (2020), Lindemann et al. (2021), and Zhang et al. (2023).

In the following discussion about DeepAR we follow Salinas et al. (2020) with slight modification on symbols. Suppose that the value of time series i at time t is denoted by $y_{i,t}$. The goal is to model conditional distribution

$$P\left(\boldsymbol{y}_{i,t_0:T} \middle| \boldsymbol{y}_{i,1:t_0-1}, \boldsymbol{x}_{i,1:T}\right) \tag{1}$$

of the future of each time series $\boldsymbol{y}_{i,t_0:T} \coloneqq \left[y_{i,t_0}, y_{i,t_0+1}, \dots, y_{i,T}\right]$ given its past $\boldsymbol{y}_{i,1:t_0-1} \coloneqq \left[y_{i,1}, y_{i,0}, \dots y_{i,t_0-2}, y_{i,t_0-1}\right]$. In equation (1), t_0 denotes the time point from which it is assumed that $y_{i,t}$ is unknown at prediction time and $\boldsymbol{x}_{i,1:T}$ are covariates that are assumed to be known for all time points. It is assumed that the distribution of $Q_\Theta\left(\boldsymbol{y}_{i,t_0:T} \middle| \boldsymbol{y}_{i,1:t_0-1}, \boldsymbol{x}_{i,1:T}\right)$ can be written as product of likelihood of the form

$$\begin{aligned} Q_\Theta\left(\boldsymbol{y}_{i,t_0:T} \middle| \boldsymbol{y}_{i,1:t_0-1}, \boldsymbol{x}_{i,1:T}\right) &= \prod_{t=t_0}^{T} Q_\Theta\left(y_{i,t} \middle| y_{i,1:t-1}, \boldsymbol{x}_{i,1:T}\right) \\ &= \prod_{i}^{n} p\left(y_{i,t} \middle| \theta\left(\boldsymbol{h}_{i,t}, \Theta\right)\right) \end{aligned} \tag{2}$$

which is parametrized by the output $\boldsymbol{h}_{i,t}$ of the form $\boldsymbol{h}_{i,t} = h\left(\boldsymbol{h}_{i,t-1}, y_{i,t-1}, x_{i,t}, \Theta\right)$. Here h is a function that is a multilayer RNN, for instance LSTM (see Figure 1).

Steps in forecasting the monthly Indonesia's inflation rate are as follows.

1. Plot the monthly inflation rate
   The plot of the inflation rate usually contains information about trend, seasonality, outliers, and any unusual patterns in the data. This information then can be used to make further analysis on the forecasts.
2. Descriptive statistics
   Describing some descriptive statistics such as minimum, maximum, mean, median, and quantiles are often helpful to get better insight from the data.
3. Feature engineering
   These steps include splitting the data into training and testing, say 70% and 30%, and data normalization if needed. The main goal of this splitting, as recommended by Gholamy et al. (2018), is to avoid overfitting.
4. Setting up hyperparameters
   At this stage some hyperparameters might need to be setup such as the number of epochs, the number of layers, the number of cells, batch sizes, and distribution output. Since there are no hard-and-fast rules for determining the hyperparameters, the hyperparameters are chosen with the aim to minimize overfitting and reduce computational cost. Thus, we arrived at the following hyperparameter:
   i. number of epochs: 50, 100, 150, and 200;
   ii. number of layers: 2, 3, 4;
   iii. number of cells: 40, 50, 60;
   iv. batch sizes: 32, 64.

   Note that there are 18 hyperparameter settings for each epoch. This can be explained as follows. Suppose the number of epochs is 50, there are three layers (2, 3, 4), there are three number of cells (40, 50, 60), and two batch sizes (32, 64). From this epoch there are $1 \times 3 \times 3 \times 2 = 18$ combination of hyperparameter settings. This combination also applies to other epochs. The name of each configuration corresponds to DeepAR (abbreviated DAR) model i. For example, for the number of epochs equals 50, the DAR1 corresponds to DeepAR model with the number of layers 2, the number of cells 40, and the batch sizes 32. Similarly, the DAR18 consist of number of layers 4, the number of cells 60, and the batch sizes 64. This naming convention also applies to other epochs. Thus, for the number of epochs equal 200, the DAR1 corresponds to the DeepAR model with the number of layers 2, the number of cells 40, and the batch sizes 32. Table 1 shows the hyperparameter settings for the eighteen DAR models. Again, note that these models differ only in the number of epochs.

Table 1. DeepAR Models with Hyperparameter Settings

Model	Number of layers	Number of cells	Batch sizes
DAR 1	2	40	32
DAR 2	2	40	64
DAR 3	2	50	32
DAR 4	2	50	64
DAR 5	2	60	32
DAR 6	2	60	64

DAR 7	3	40	32
DAR 8	3	40	64
DAR 9	3	50	32
DAR 10	3	50	64
DAR 11	3	60	32
DAR 12	3	60	64
DAR 13	4	40	32
DAR 14	4	40	64
DAR 15	4	50	32
DAR 16	4	50	64
DAR 17	4	60	32
DAR 18	4	60	64

Here the number of layers refers the number of LSTM layers, the number of cells refers to the number of LSTM cells for each layer, and the batch sizes refer to the size of batches used in the training and prediction.

5. Training and testing the model
   For each of hyperparameters settings, the training and testing are replicated ten times to ensure consistency.
6. Forecasting
   The root mean square error (RMSE) is used as a measure of forecast accuracy. The RMSE of the deep autoregressive models is compared with the other competing models such deep state space, simple feed forward, dan seasonal naive.

In this research the performance of DeepAR is compared with other models such as deep state space, simple feed forward, and seasonal naive. The deep state space of (Rangapuram et al., 2018), another probabilistic deep learning method, models the data according to

$$p\left(\boldsymbol{y}_{1:T}^{(i)} \middle| \boldsymbol{x}_{1:T_i}^{(i)}, \Phi\right) = p_{SS}\left(\boldsymbol{y}_{1:T_i}^{(i)} \middle| \Theta_{1:T_i}^{(i)}\right) \tag{3}$$

where p_{SS} denotes the marginal likelihood under a linear state space which is defined as

$$p_{SS}\left(\boldsymbol{y}_{1:T_i}^{(i)} \middle| \Theta_{1:T_i}^{(i)}\right) = p\left(y_i^{(i)} \middle| \Theta_1^{(i)}\right) \prod_{t=2}^{T} p\left(y_t^{(i)} \middle| y_{1:t-1}^{(i)}, \Theta_{1:t}^{(i)}\right). \tag{4}$$

In equation (3), $\Theta_t^{(i)}$ is time varying parameters. Furthermore, the parameters of the state space model $\Theta_t^{(i)}$ is computed via the recurrent neural network function $\boldsymbol{h}_t^{(i)} = h\left(\boldsymbol{h}_{t-1}^{(i)}, \boldsymbol{x}_t^{(i)}, \Phi\right)$.

Another model that was considered in this research was the simple feed-forward networks (FFN). This FFN is basically a simple multi-layer perceptron (MLP) that was trained using whole dataset (Makridakis et al., 2018).

The last model that was considered in this research was seasonal naive that was implemented in Alexandrov et al. (2020) . This models the series as

$$\tilde{y}(T+l) = y(T+l-s) \tag{5}$$

where T is the forecast time, s is the season length, and $l = 0, \dots, n-1$. Here n is the prediction length. If $n > s$, then the season is repeated multiple times.

In order to compare the accuracy of those four models the root mean squared error (RMSE) was used. This RMSE is defined as

$$\mathrm{RMSE} = \sqrt{\tfrac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}\,. \tag{6}$$

Here n denote the number of observations, y denote observations, and $\hat{y}$ denote the predicted values. The advantage of using RMSE is that it has the same scale with the series and it is robust to outliers (Koutsandreas et al., 2022). This study follows Salinas et al. (2020) in that $\hat{y}$ denote the predicted median values.

3. Results and Discussions

In this research the GluonTS of Alexandrov et al. (2020) was used in modelling and forecasting the Indonesian inflation rate. The software ran both on localhost and on Google Colaboratory. Jupyter Notebook was used as the computing platform that allows users to code, to manipulate, and to convert various file formats.

Figure 2 shows the monthly Indonesia inflation rate from January 1978 to April 2022. As can be seen from the figure, the rate has extrema at some points, but otherwise look stationary.

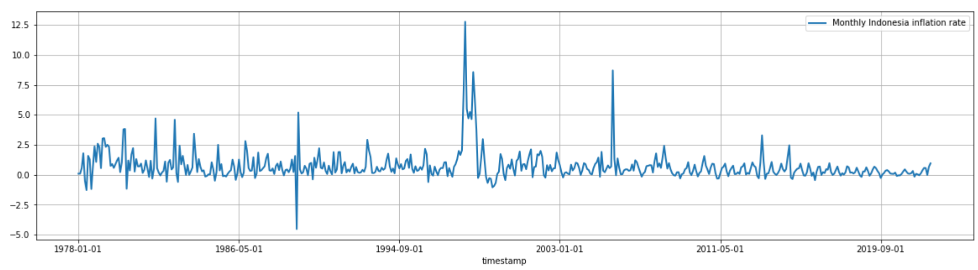

Figure 2. Plot of the Indonesian inflation rate from January 1978 to April 2022

The inflation reached a peak at 12.76 percent in February 1998 which was strongly related to the economic crisis during that time. There are two more peaks worthy to note: the one on July 1998 (at 8.56 percent) and the other one on October 2005 (at 8.7%). Looking in more details at the data, the 1989 saw the lowest rate of inflation at -4.53 percent.

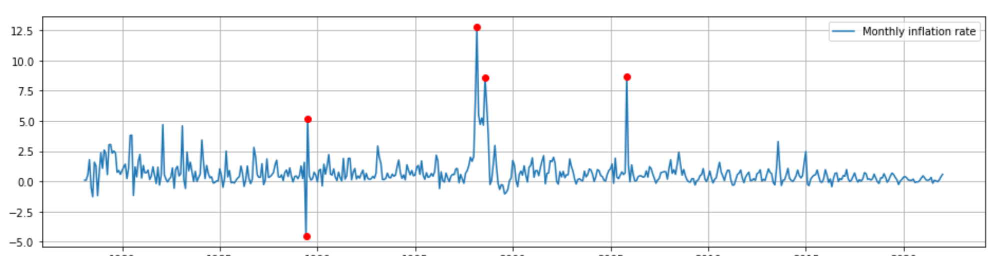

Figure 3. Plot of the Indonesian inflation rate with extrema points shown in red dots.

Table 2 summaries the descriptive statistics of the inflation rate throughout the four decades.

Table 2. Descriptive Statistics

Mean	Standard Deviation	Minimum	25%	50%	75%	Maximum
0.718985	1.210475	-4.53	0.13	0.455	0.93	12.76

As can be seen from Table 2, the mean is larger than the median. This suggests that the inflation rate is rightly skewed. The median of inflation rate is approximately 0.5 percent.

The next step was feature engineering. Based on study in Gholamy et al. (2018), the data was split into training (70 percent) and testing (30 percent). This splitting aimed to obtain better accuracy and avoid overfitting. Specifically, the inflation rate from January 1978 to December 2008 were the training data set, while the testing data started from January 2009 to April 2022. Figure 3 shows the training (dark blue) and testing data (orange).

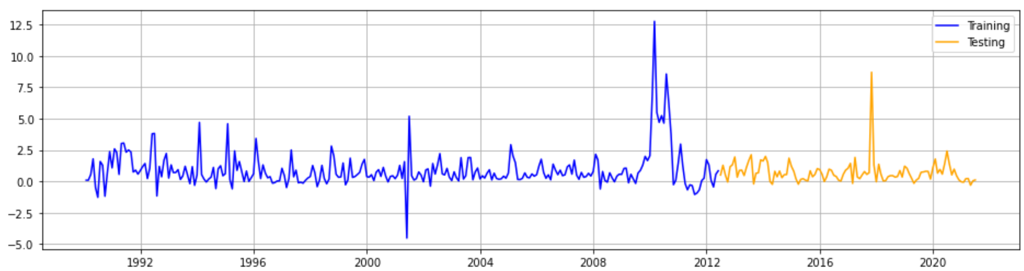

Figure 3. Plot of training data (in blue) and testing data (in orange).

Based on hyperparameter configurations in Table 1, the next step was to training and testing the models. Each model was executed both in localhost and in cloud, i.e., Google Colaboratory, and was repeated ten times to ensure consistency. It took 5—7 hours for one run. Table 3—5 list the RMSE values for each DeepAR model.

Table 3 shows RMSE values for DeepAR (DAR 1 to DAR 18) models.

Table 3. RMSE Values for Epoch Equals 50

Model	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
DAR 1	0.317614	0.309161	0.306674	0.293830	0.305772	0.300959	0.314762	0.302108	0.262321	0.281506
DAR 2	0.306219	0.297990	0.316872	0.304903	0.303254	0.288397	0.304205	0.274916	0.288743	0.308438
DAR 3	0.337765	0.357402	0.339177	0.381367	0.339900	0.371025	0.302200	0.292312	0.343702	0.276909
DAR 4	0.289336	0.337862	0.351306	0.430639	0.364866	0.390863	0.337978	0.342361	0.317310	0.424000
DAR 5	0.332866	0.376746	0.320159	0.339308	0.419357	0.392557	0.290768	0.278509	0.343095	0.380025
DAR 6	0.364176	0.389930	0.306796	0.354915	0.370278	0.361754	0.286701	0.413559	0.301474	0.320430
DAR 7	0.499842	0.273018	0.386497	0.400895	0.309781	0.318686	0.315459	0.298293	0.287979	0.344904
DAR 8	0.372474	0.382282	0.308242	0.329491	0.358737	0.380598	0.349630	0.369443	0.359362	0.341940

Jurnal Matematika Vol. 14, No.1, Juni 2024, pp. 37-51 ISSN: 1693-1394
Article DOI: 10.24843/JMAT.2024.v14.i01.p170

DAR 9	0.331416	0.405092	0.362811	0.290188	0.366699	0.414145	0.351266	0.355339	0.318569	0.468836
DAR 10	0.492864	0.352287	0.279800	0.406719	0.244334	0.364978	0.360313	0.388407	0.354070	0.402968
DAR 11	0.436598	0.434068	0.394674	0.347134	0.429010	0.574953	0.435929	0.276156	0.375429	0.271837
DAR 12	0.426436	0.381947	0.409521	0.411895	0.474065	0.611994	0.402436	0.485465	0.476788	0.533566
DAR 13	0.498284	0.415508	0.500602	0.354785	0.330303	**0.218565**	0.323817	0.523003	0.382354	0.527157
DAR 14	0.474622	0.392635	0.469889	0.308384	0.320830	0.469451	0.451374	0.430991	0.493335	0.531682
DAR 15	0.451387	0.433600	0.454015	0.286293	0.293743	0.432763	0.450927	0.402470	0.523399	0.532556
DAR 16	0.448691	0.451255	0.386928	0.390762	0.448700	0.691879	0.430478	0.468398	0.457985	0.289701
DAR 17	0.478482	0.285666	0.470660	0.590003	0.435568	0.607273	0.352392	0.407704	0.411419	0.717300
DAR 18	0.599063	0.609363	0.666988	0.575343	0.273620	0.560479	0.590278	0.313811	0.378752	0.529570

As can be seen from Table 3, the smallest RMSE is achieved by DAR 13 (see Table 1 for the details of hyperparameter settings) on the sixth run at 0.218565 (highlighted in bold). Table 4 shows RMSE values for epoch equals 100. The smallest RMSE value is achieved by DAR 17 at 0.229062 (highlighted in bold).

Table 4. RMSE Values for Epoch Equals 100

Model	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
DAR 1	0.265950	0.308572	0.279576	0.265090	0.274590	0.296468	0.281116	0.259341	0.291985	0.314305
DAR 2	0.470014	0.522559	0.236752	0.510035	0.300995	0.450139	0.435761	0.424059	0.510408	0.378726
DAR 3	0.332606	0.287613	0.411581	0.349135	0.337759	0.361104	0.367371	0.307404	0.673508	0.322393
DAR 4	0.379688	0.374468	0.434132	0.382588	0.318243	0.324781	0.394430	0.426091	0.400175	0.305051
DAR 5	0.380918	0.483723	0.520154	0.350029	0.317199	0.307455	0.506411	0.444640	0.330239	0.413108
DAR 6	0.341298	0.287421	0.381445	0.342654	0.429770	0.412719	0.397891	0.352922	0.444793	0.719359
DAR 7	0.342331	0.474262	0.475847	0.451400	0.275051	0.400920	0.640682	0.427831	0.375531	0.439820
DAR 8	0.405394	0.338952	0.436858	0.316741	0.392914	0.422532	0.403969	0.439470	0.407749	0.501722
DAR 9	0.742244	0.502284	0.470180	0.322253	0.412216	0.431448	0.409925	0.575907	0.389968	0.408164
DAR 10	0.378908	0.373765	0.284617	0.421750	0.397964	0.327392	0.427432	0.296213	0.492612	0.280997
DAR 11	0.344448	0.369279	0.277906	0.385458	0.336327	0.443025	0.284452	0.258659	0.409194	0.377493
DAR 12	0.719623	0.499944	0.583190	0.618173	0.261763	0.380859	0.515033	0.655327	0.261816	0.464304
DAR 13	0.538125	0.405359	0.357253	0.435073	0.341867	0.454693	0.379356	0.403762	0.380581	0.703169
DAR 14	0.321088	0.320173	0.313014	0.371480	0.318690	0.557548	0.422981	0.356628	0.333968	0.350448
DAR 15	0.316245	0.350573	0.314890	0.396257	0.304519	0.560910	0.466060	0.355940	0.368064	0.354497
DAR 16	0.241868	0.452229	0.423942	0.366509	0.450344	0.398044	0.372857	0.471954	0.376209	0.317370
DAR 17	0.346805	0.275832	0.430380	0.286376	0.289424	0.436569	**0.229062**	0.526409	0.296551	0.451998
DAR 18	0.512824	0.379425	0.563748	0.361984	0.456919	0.775952	0.417182	0.647197	0.623466	0.335106

Table 5 contains RMSE values for epoch equals 150. In this setting, the smallest RMSE is achieved by DAR 9 on the tenth run at 0.257854 (highlighted in bold).

Table 5. RMSE Values for Epoch Equals 150

Model	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
DAR 1	0.285575	0.358732	0.306646	0.325979	0.338432	0.362360	0.351053	0.358920	0.283742	0.308268
DAR 2	0.288758	0.272763	0.384886	0.396732	0.371763	0.341018	0.328804	0.301975	0.305865	0.315157
DAR 3	0.575138	0.401380	0.277330	0.340559	0.346982	0.435525	0.331567	0.349752	0.429843	0.425787
DAR 4	0.306759	0.400804	0.481443	0.365940	0.370672	0.366338	0.352784	0.650164	0.535620	0.339668
DAR 5	0.336779	0.332993	0.309706	0.370935	0.563885	0.333553	0.447668	0.410626	0.324740	0.306009
DAR 6	0.402613	0.358640	0.363700	0.313064	0.400371	0.451577	0.349704	0.370639	0.316810	0.345606
DAR 7	0.545996	0.380983	0.381072	0.372448	0.271845	0.372693	0.305008	0.331316	0.270528	0.611340
DAR 8	0.337631	0.482281	0.264475	0.517108	0.325671	0.434251	0.326594	0.451940	0.418313	0.297580
DAR 9	0.481815	0.576095	0.432489	0.532070	0.374078	0.275887	0.560564	0.412469	0.305170	**0.257854**
DAR 10	0.377458	0.470343	0.431058	0.441781	0.473640	0.598524	0.335916	0.456994	0.408120	0.351306
DAR 11	0.554044	0.368573	0.726049	0.524560	0.318058	0.341357	0.527009	0.425026	0.431565	0.334513
DAR 12	0.421421	0.601397	0.652900	0.431712	0.433326	0.465526	0.311294	0.512543	0.441739	0.474546
DAR 13	0.367051	0.403070	0.444820	0.329255	0.474392	0.432492	0.378855	0.442000	0.320936	0.404724
DAR 14	0.304039	0.463083	0.319265	0.487654	0.400917	0.382977	0.483388	0.430346	0.385051	0.342318

DAR 15	0.302302	0.462505	0.321894	0.430761	0.405400	0.393552	0.484181	0.405028	0.370600	0.343288
DAR 16	0.391861	0.362476	0.437432	0.365363	0.282134	0.436781	0.613258	0.463808	0.452265	0.330510
DAR 17	0.430347	0.611273	0.366050	0.448697	0.431357	0.548656	0.649802	0.415294	0.405516	0.483858
DAR 18	0.329232	0.433340	0.382667	0.515018	0.323156	0.747460	0.488332	0.445409	0.565922	0.785081

Table 6 lists RMSE values for epoch equals 200. The smallest RMSE in this setting is achieved by DAR 15 on the third run at 0.222143 (highlighted in bold).

Table 6. RMSE Values for Epoch Equals 200

Model	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
DAR 1	0.447461	0.372106	0.341088	0.315047	0.328073	0.345760	0.305854	0.382979	0.457037	0.413096
DAR 2	0.330191	0.341704	0.377589	0.330375	0.383521	0.331075	0.317915	0.453882	0.352808	0.345809
DAR 3	0.450878	0.278263	0.395717	0.347092	0.473111	0.316858	0.395291	0.340395	0.340667	0.352289
DAR 4	0.388730	0.336464	0.367751	0.348395	0.593076	0.372872	0.462732	0.348219	0.401593	0.437111
DAR 5	0.349545	0.264744	0.366815	0.550055	0.330749	0.321232	0.538661	0.305216	0.356934	0.361641
DAR 6	0.449242	0.348516	0.598830	0.456327	0.486678	0.429122	0.422985	0.442061	0.463316	0.479412
DAR 7	0.446407	0.457296	0.339763	0.282373	0.424506	0.417358	0.340753	0.506834	0.300659	0.437505
DAR 8	0.309921	0.548189	0.601084	0.614135	0.455836	0.481137	0.557362	0.337543	0.336549	0.328104
DAR 9	0.385897	0.397578	0.338701	0.379585	0.269196	0.236563	0.375584	0.343175	0.389407	0.489726
DAR 10	0.723735	0.454514	0.390780	0.361191	0.281158	0.419942	0.444579	0.304734	0.621242	0.434477
DAR 11	0.504088	0.438947	0.449609	0.432153	0.462658	0.406054	0.678947	0.487215	0.344998	0.336521
DAR 12	0.435170	0.303224	0.493280	0.528651	0.505379	0.385246	0.427239	0.292252	0.368242	0.614583
DAR 13	0.562600	0.396963	0.656181	0.706909	0.337236	0.253804	0.399090	0.448885	0.415895	0.460445
DAR 14	0.323853	0.357540	0.222547	0.352882	0.444174	0.455539	0.429200	0.382352	0.514199	0.473389
DAR 15	0.329255	0.353904	**0.222143**	0.351385	0.441663	0.453026	0.434622	0.386044	0.545503	0.474050
DAR 16	0.483481	0.426216	0.330604	0.415785	0.363181	0.437443	0.343265	0.459599	0.431832	0.444365
DAR 17	0.372719	0.491972	0.307150	0.422434	0.428813	0.369715	0.469807	0.632944	0.246757	0.255241
DAR 18	0.334982	0.567418	0.371870	0.406682	0.349036	0.442105	0.350631	0.430492	0.485384	0.382346

As can be seen from Tables 3—6, the smallest RMSE is achieved by DAR 13 on the sixth run at 0.218565 (highlighted in bold). This DAR 13 corresponds to the number of epochs equals 50, the number of layers equals 4, the number of cells equals 40, and the batch size equals 32. This RMSE also the smallest among other competing models such as deep state space (Table 7), simple feed-forward (Table 8), and seasonal naive.

Table 7. RMSE Values for Deep State Space (DSS)

Epochs	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
50	0.359365	0.301276	0.301872	0.365233	0.424525	**0.287340**	0.301276	0.478826	0.417870	0.396166
100	0.357950	0.434968	0.394559	0.358319	0.358607	0.532757	0.571228	0.386906	0.439717	0.497993
150	0.447518	0.363252	0.372764	0.404966	0.411262	0.290104	0.432599	0.413087	0.503382	0.371575
200	0.661428	0.312687	0.432330	0.475475	0.495799	0.451847	0.325782	0.604501	0.499192	0.629215

As with the DeepAR, the deep state space (DSS) models were run ten times to ensure consistency. As can be seen from Table 7, the smallest RMSE value is at run 6 with the number of epochs equals 50. Table 8 summarizes RMSE values for simple feed-forward (SFF). The SFF models compete quite well with DeepAR and DSS.

Table 8. RMSE Values for Simple Feed-Forward (SFF)

Epochs	Run 1	Run 2	Run 3	Run 4	Run 5	Run 6	Run 7	Run 8	Run 9	Run 10
50	0.365199	0.368222	0.365288	0.363144	0.361732	0.361983	0.368896	0.369558	0.361296	0.363355
100	0.354422	0.360085	0.352488	0.368279	0.354707	0.358639	0.362032	**0.350449**	0.365157	0.355787

Jurnal Matematika Vol. 14, No.1, Juni 2024, pp. 37-51 ISSN: 1693-1394
Article DOI: 10.24843/JMAT.2024.v14.i01.p170

150	0.365013	0.38156	0.369951	0.376117	0.363430	0.361013	0.354662	0.370218	0.369914	0.353486
200	0.363377	0.361456	0.353084	0.357774	0.357841	0.360260	0.361795	0.358244	0.355145	0.360406

As can be seen from Table 8, the RMSE value at 0.350449 is achieved by SFF with the number of epochs equal 100 on the eight runs. Finally, the seasonal naïve (SN) performs quite well. It is slightly better (RMSE at 0.336056) than SFF in this case. Since the DeepAR has the smallest RMSE among the other competing models, we then proceeded to forecast the monthly inflation rate.

Table 9. Median predictions of inflation rate forecast

May 2022	June 2022	July 2022	August 2022	Septem-ber 2022	October 2022	Novem-ber 2022	December 2022	January 2023	February 2023
0.25	0.30	0.75	0.8125	0.7	0.6875	0.5	0.375	0.125	0.125

The median inflation rate forecasts from May 2022 to February 2023 can be seen in Table 9 and plot graphically in Figure 4.

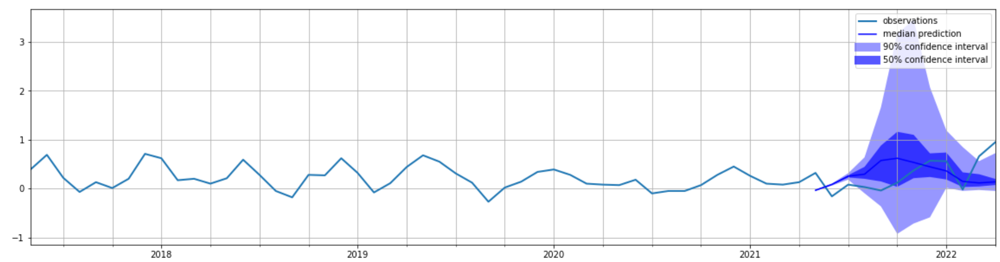

Figure 4. The median inflation forecast for May 2022 to February 2023 (blue line).

The median forecast of the rate of inflation for May 2022 to February 2023 suggests that the rate fluctuates below 1 percent, but otherwise stationary.

In this research the authors note the following. First, while there is no fix rule on how to set hyperparameters optimally, increasing the number of epochs does not guarantee smaller RMSE. Next, the other models such as deep state space performs quite well which is indicate by the RMSE value close to the selected deep autoregressive models. Finally, the authors also note that the inflation rate forecasts fluctuate which is in agreement in the authors' previous study. However, due to its stochastic nature during the sampling, the forecasts results obtained by probabilistic deep learning model such as deep autoregressive and deep state space may differ considerably. For future work, different hyperparameters settings are possible to improve the prediction of deep autoregressive networks.

4. Conclusions

This study concludes the following. First, deep autoregressive (DeepAR) outperforms the other competing models (DeepSS, SFF, and SN) in terms of RMSE. More spesifically

DeepAR with the following hyperparameter settings: the number of epochs equals 50, the number of layers equal 4, the number of cells equals 40, and the batch size equals 32. Second, as can be seen from Table 9 and Figure 4, the forecasts of inflation rate fluctuate below 1 percent, but otherwise look stationary. This results also inline with our previous research, see for example Sumarjaya & Susilawati (2023)

Acknowledgment

We would like to thank Universitas Udayana for funding this research. This research is under the DIPA PNBP Universitas Udayana year 2022 based on Research Grant Contract number B/78.820/UN14.4.A/PT.01.03/2022, on 19 April 2022. The authors would like to thank two anonymous reviewers for their helpful comments.

References

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., Stella, L., Türkmen, A. C., & Wang, Y. (2019). *GluonTS: Probabilistic time series models in Python*. http://arxiv.org/abs/1906.05264

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., Stella, L., Türkmen, A. C., & Wang, Y. (2020a). GluonTS: Probabilistic and neural time series modeling in Python. *Journal of Machine Learning Research*, *21*(116), 1–6. http://jmlr.org/papers/v21/19-820.html

Alexandrov, A., Benidis, K., Bohlke-Schneider, M., Flunkert, V., Gasthaus, J., Januschowski, T., Maddix, D. C., Rangapuram, S., Salinas, D., Schulz, J., Stella, L., Türkmen, A. C., & Wang, Y. (2020b). GluonTS: Probabilistic and Neural Time Series Modeling in Python. In *Journal of Machine Learning Research* (Vol. 21). https://github.com/awslabs/gluon-ts

Amaly, M. H., Hirzi, R. H., & Basirun, B. (2022). Perbandingan metode ANN backpropagation dan ARMA untuk peramalan inflasi di Indonesia. *Jambura Journal of Probability and Statistics*, *3*(2). https://doi.org/10.34312/jjps.v1i1.15440

Ao, S. I., & Fayek, H. (2023). Continual deep learning for time series modeling. *Sensors*, *23*(16), 7167–7167. https://doi.org/10.3390/s23167167

Asmarani, T. E. (2023). Peramalan inflasi dengan menggunakan metode ARIMA: Studi di Indonesia. *Journal on Education*, *5*(2), 4684–4692. https://jonedu.org/index.php/joe/article/view/1200/944

Cahuantzi, R., Chen, X., & Güttel, S. (2023). A comparison of LSTM and GRU networks for learning symbolic sequences. *Lecture Notes in Networks and Systems*, *739 LNNS*. https://doi.org/10.1007/978-3-031-37963-5_53

Estiko, F. I., & S., W. (2019). Analysis of Indonesia's inflation using ARIMA and artificial neural network. *Economics Development Analysis Journal*, *8*(2). http://journal.unnes.ac.id/sju/index.php/edaj

Fireza, A. F., & Ahmad, D. (2023). Implementasi fuzzy time series logika Lee untuk peramalan inflasi di Indonesia. *Lebesgue: Jurnal Ilmiah Pendidikan Matematika, Matematika Dan Statistika*, *4*(2), 1084–1092. https://doi.org/10.46306/lb.v4i2.386

Gholamy, A., Kreinovich, V., & Kosheleva, O. (2018). Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. *International Journal of Intelligent Technologies and Applied Statistics*, *11*(2), 105–111. https://doi.org/10.6148/IJITAS.201806_11(2).0003

Hartati, H. (2017). Penggunaan metode ARIMA dalam meramal pergerakan inflasi. *Jurnal Matematika Sains Dan Teknologi (JMST)*, *18*(1), 1–10. https://doi.org/10.33830/jmst.v18i1.163.2017

Irie, K., Tüske, Z., Alkhouli, T., Schlüter, R., & Ney, H. (2016). LSTM, GRU, highway and a bit of attention: An empirical overview for language modeling in speech recognition. *Proceedings of the Annual Conference of the International Speech Communication Association, INTERSPEECH*, *08-12-September-2016*. https://doi.org/10.21437/Interspeech.2016-491

Kadry, I. R. Al, Massalesse, J., & Nur, Muh. (2022). Forecasting inflation in Indonesia using the modified fuzzy time series Cheng. *Jurnal Matematika, Statistika Dan Komputasi*, *19*(1), 210–222. https://doi.org/10.20956/j.v19i1.21868

Koutsandreas, D., Spiliotis, E., Petropoulos, F., & Assimakopoulos, V. (2022). On the selection of forecasting accuracy measures. *Journal of the Operational Research Society*, *73*(5), 937–954. https://doi.org/10.1080/01605682.2021.1892464

LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *Nature*, *521*(7553), 436–444. https://doi.org/10.1038/nature14539

Lindemann, B., Müller, T., Vietz, H., Jazdi, N., & Weyrich, M. (2021). A survey on long short-term memory networks for time series prediction. *Procedia CIRP*, *99*, 650–655. https://doi.org/10.1016/j.procir.2021.03.088

Makridakis, S., Spiliotis, E., & Assimakopoulos, V. (2018). Statistical and machine learning forecasting methods: Concerns and ways forward. *PLOS ONE*, *13*(3), e0194889. https://doi.org/10.1371/journal.pone.0194889

Melyani, C. A., Nurtsabita, A., Shafa, G. Z., & Widodo, E. (2021). Peramalan inflasi di Indonesia menggunakan metode autoregressive moving average (ARMA). *Journal of Mathematics Education and Science*, *4*(2), 67–74. https://doi.org/10.32665/james.v4i2.231

Muslihin, K. R. A., & Ruchjana, B. N. (2023). Model autoregressive moving average (ARMA) untuk peramalan tingkat inflasi di Indonesia. *Limits: Journal of Mathematics and Its Applications*, *20*(2), 209–218. https://doi.org/10.12962/limits.v20i2.15098

Nanlohy, Y. W. A., & Loklomin, S. B. (2023). Model autoregressive integrated moving average (ARIMA) untuk meramalkan inflasi Indonesia. *Variance: Journal of Statistics and Its Applications*, *5*(2), 201–208. https://doi.org/10.30598/variance-vol5iss2page201-208

Oner, C. (2022). Inflation: prices on the rise. *Finance & Development*, 30–31. https://www.imf.org/-/media/Files/Publications/Fandd/Back-to-Basics/oner-inflation.ashx

Purwanti, D., & Purwadi, J. (2019). Metode Brown's double exponential smoothing dalam peramalan laju inflasi di Indonesia. *Jurnal Ilmiah Matematika*, *6*(2), 54. https://doi.org/10.26555/konvergensi.v6i2.19548

Qalbi, A., Nurfadilah, K., & Alwi, W. (2021). Comparison of fuzzy time series methods and autoregressive integrated moving average (ARIMA) for inflation data. *Eigen Mathematics Journal*, *4*(2), 40–50. https://doi.org/10.29303/emj.v4i2.122

Rangapuram, S. S., Seeger, M., Gasthaus, J., Stella, L., Wang, Y., & Januschowski, T. (2018). Deep state space models for time series forecasting. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*. Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2018/file/5cf68969fb67aa6082363a6d4e6468e2-Paper.pdf

Rifa'i, A. (2021). Prediksi inflasi Indonesia berdasarkan fuzzy ANN menggunakan algoritma genetika. *Jurnal Eltikom: Jurnal Teknik Elektro, Teknologi Informasi Dan Komputer*, *5*(1), 12–24. https://doi.org/10.31961/eltikom.v5i1.215

Rismawanti, Y., & Yamin Darsyah, M. (2018). Perbandingan peramalan metode moving average dan exponential smoothing Holt Winter untuk menentukan peramalan inflasi di Indonesia. In D. Cahyandari, H. D. Santoso, A. H. Saptadi, A. Yanto, & E. Yuliyanto (Eds.), *Prosiding Seminar Nasional Mahasiswa Unimus* (Vol. 1, pp. 330–335). Lembaga Penelitian dan Pengabdian Masyarakat Universitas Muhammadiyah Semarang. https://prosiding.unimus.ac.id/index.php/mahasiswa/article/download/167/170

Salinas, D., Flunkert, V., Gasthaus, J., & Januschowski, T. (2020). DeepAR: Probabilistic forecasting with autoregressive recurrent networks. *International Journal of Forecasting*, *36*(3), 1181–1191. https://doi.org/10.1016/j.ijforecast.2019.07.001

Saragih, S. M., & Sembiring, P. (2022). Analisis perbandingan metode ARIMA dan double exponential smoothing dari Brown pada peramalan inflasi di Indonesia. *Journal of Fundamental Mathematics and Applications (JFMA)*, *5*(2), 176–191. https://doi.org/10.14710/JFMA.V5I2.15312

Sudibyo, N. A., Iswardani, A., Septyanto, A. W., & Wicaksono, T. G. (2020). Prediksi inflasi di Indonesia menggunakan metode moving average, single exponential smoothing dan double exponential smoothing. *Lebesgue: Jurnal Ilmiah Pendidikan Matematika, Matematika Dan Statistika*, *1*(2), 123–129. https://doi.org/10.46306/lb.v1i2

Sumarjaya, I. W., & Susilawati, M. (2023). Forecasting monthly inflation rate in Denpasar using long short-term memory. *Jurnal Matematika*, *13*(1), 11–24. https://doi.org/10.24843/JMAT.2023.v13.i01.p157

Udin, A. C., & Jatipaningrum, M. T. (2020). Peramalan inflasi di Indonesia menggunakan metode fuzzy time series based average dan fuzzy time series Saxena-Easo (Studi kasus: data inflasi di Indonesia). *Jurnal Statistika Industri Dan Komputasi*, *05*(2), 1–10. https://ejournal.akprind.ac.id/index.php/STATISTIKA/article/view/2871/2183

Jurnal Matematika Vol. 14, No.1, Juni 2024, pp. 37-51 ISSN: 1693-1394
Article DOI: 10.24843/JMAT.2024.v14.i01.p170

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A review on the long short-term memory model. *Artificial Intelligence Review*, *53*(8), 5929–5955. https://doi.org/10.1007/s10462-020-09838-1

Wiranto, A. R., & Setiawan, E. (2023). Implementasi metode backpropagation neural network dalam meramalkan tingkat inflasi di Indonesia. *MATHunesa: Jurnal Ilmiah Matematika*, *II*(1), 8–16.

Yusnitasari, A. (2020). Peramalan inflasi Indonesia dengan menggunakan metode ARIMA Box-Jenkins. *Jurnal Paradigma Multidisipliner (JPM)*, *1*(2), 152–159. https://smkn1magelang.sch.id/jpm/index.php/jpm/article/view/25

Zhang, A., Lipton, Z. C., Li, M., & Smola, A. J. (2023). *Dive into deep learning*. Cambridge University Press. https://d2l.ai/d2l-en.pdf