IJCCS (Indonesian Journal of Computing and Cybernetics Systems)
Vol.13, No.3, July 2019, pp. 293~300
ISSN (print): 1978-1520, ISSN (online): 2460-7258
DOI: https://doi.org/10.22146/ijccs.47802

Extended Kalman Filter In Recurrent Neural Network: USDIDR Forecasting Case Study

Muhammad Asaduddin Hazazi[1], Agus Sihabuddin*[2]
[1]Master Program of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
[2]Department of Computer Science and Electronics, FMIPA UGM, Yogyakarta, Indonesia
e-mail: [1]muhammad.asaduddin.h@mail.ugm.ac.id, ***[2]a_sihabudin@ugm.ac.id**

Abstrak

Jaringan Syaraf Tiruan (JST) khususnya reccurent neural network (RNN) telah banyak digunakan untuk memprediksi nilai tukar mata uang. Algoritma pembelajaran yang umum digunakan dalam JST adalah Stochastic Gradient Descent (SGD). Salah satu kelebihan SGD adalah waktu komputasi yang dibutuhkan relatif singkat. Tetapi SGD juga memiliki kelemahan, termasuk SGD yang membutuhkan beberapa hiperparameter seperti parameter regularisasi. Selain itu SGD relatif membutuhkan banyak zaman untuk mencapai konvergensi.

Extended Kalman Filter (EKF) sebagai algoritma pembelajaran pada RNN digunakan untuk menggantikan SGD dengan harapan tingkat akurasi dan konvergensi yang lebih baik. Penelitian ini menggunakan data nilai tukar IDR / USD dari 31 Agustus 2015 hingga 29 Agustus 2018 dengan data 70% sebagai data pelatihan dan 30% data sebagai data uji. Penelitian ini menunjukkan bahwa RNN-EKF menghasilkan kecepatan konvergen yang lebih baik dan akurasi yang lebih baik dibandingkan dengan RNN-SGD.

Kata kunci— *mata uang, peramalan, recurrent neural network, stochastic gradient descent, extended Kalman Filter*

Abstract

Artificial Neural Networks (ANN) especially Recurrent Neural Network (RNN) have been widely used to predict currency exchange rates. The learning algorithm that is commonly used in ANN is Stochastic Gradient Descent (SGD). One of the advantages of SGD is that the computational time needed is relatively short. But SGD also has weaknesses, including SGD requiring several hyperparameters such as the regularization parameter. Besides that SGD relatively requires a lot of epoch to reach convergence. Extended Kalman Filter (EKF) as a learning algorithm on RNN is used to replace SGD with the hope of a better level of accuracy and convergence rate. This study uses IDR / USD exchange rate data from 31 August 2015 to 29 August 2018 with 70% data as training data and 30% data as test data. This research shows that RNN-EKF produces better convergent speeds and better accuracy compared to RNN-SGD.

Keywords— *exchange rates, forecasting, recurrent neural network, stochastic gradient descent, extended Kalman Filter.*

Received July 18th,2019; Revised July 22th, 2019; Accepted July 25th, 2019

1. INTRODUCTION

There are two main approaches to forecast exchange rates: fundamentally and technically approach. The fundamental approach looks at the situation the economy of a region and conduct relationship analysis, whereas technical approach is an attempt to estimate prices with observing changes in prices in the past, including time-series method and neural network. Exchange rate data has many variables, with two most used variables are value variables and time variables. Exchange rate data is included in time-series data which are classified as data sequences and forecasting in this field is well developed. Recently, soft computing (especially *artificial neural network (ANN)*) is used as an approach for technical analysis. The advantage of artificial neural networks is to produce better results when applied to non-linear data than other non-linear methods such as *bilinear* and *regression tree models* [1].

Conducted a system test on the implementation of the Arima method and Artificial Neural Network and merged Arima and ANN [2]. Data that is not stationary will be differed using *ACF (Autocorrelation Function)* and *PACF (Partial Autocorrelation Function)*. The results of the calculation of *MSE (Mean Square Error)* and *MAPE (Mean Absolute Percentage Error)* errors with ANN are the largest. The values of MSE and MAPE when using the merge of Arima and ANN show mixed results for each input. Did the *Hybrid Arima* model with a Neural Network to predict time-series data [3]. The Arima is used for predicting time-series data containing linear components, while NN for time-series data with non-linear components. The hybrid model from Arima-NN has a more accurate prediction accuracy than the traditional Arima model.

One form of artificial neural network that is widely used in modelling data sequences is a *recurrent neural network (RNN)*. Slightly different from ordinary ANN, RNN has units in the hidden layer that are interconnected and can be used to be inserted into the hidden layer again [4]. Therefore, the network can use previous data history for sequence data, and it is suitable for exchange rate data using previous data (lag) as input [5].

Apply the RNN with the *Backpropagation Through Time (BPTT)* learning algorithm to forecast stock prices [6]. From the results of the trials conducted on forecasting stock prices using RNN-BPTT produces different error values. Prediction by applying RNN was also carried out by [7]. The focus of this study is an online prediction, where the tasks are done are far more difficult than gratuitous inference with neural networks offline. This study uses discrete-time RNN to see the RNN's ability to predict the next symbol in the sequence.

The common learning algorithm used in conjunction with RNN is *Stochastic Gradient Descent (SGD)*. SGD works by minimizing the value of a function by making changes to parameter values that affect the function value. One of the advantages of SGD is that the computational time needed in one iteration is relatively short. This is because SGD changes parameter values after evaluating only one or several pairs of training data compared to all available data [8]. SGD itself has experienced various developments to improve the performance of ANN on exchange rate forecasting, for example like Adam [9] and RMS Prop. SGD has weaknesses, including SGD requiring several *hyperparameters* such as the *regularization parameter*. Besides that SGD relatively requires a lot of epochs to reach convergence [10].

Cernansky, Benuskova [11] and Trebaticky [12] managed to prove that the performance of the RNN can be improved by adopting the *Extended Kalman Filter (EKF)* algorithm as a learning algorithm. With EKF, the RNN model obtained has a quicker convergent level, and better performance compared to RNN which uses the SGD learning algorithm. EKF itself is widely applied to the other field technology. Generally, EKF is applied for instructions, navigation, and control of vehicles, especially aeroplanes and spacecraft. In addition, EKF is a concept commonly used for time series analysis such as

signal processing and econometrics, so the EKF models generated from training data can be used to predict future exchange rates [5].

The application of Neural Network with EKF has been carried out [13] to predict flooding by using ANN and EKF. This study emphasizes ANN to get the best model in the accuracy of flood prediction. After that, in the following year [14] conducted a classification using the Feedforward Neural Network with EKF. The results of the study provide a comparison with the classification using gradient descent. EKF gives better results in the classification problem.

2. METHODS

This research has two main objectives: to create an RNN EKF forecasting model with a faster convergence rate than RNN SGD and to discover the EKF RNN model has better accuracy compared to RNN SGD.

2.1 Data

Historical data on currency exchange rates are obtained from *https://www.ofx.com/en-gb/forex-news/historical-exchange-rates/*. The data obtained consists of 2 variables, namely time and exchange rate at that time. Table 4.1 is a data snippet of IDR / USD exchange rates from August 31, 2015, to August 29, 2018. Exchange rate data is stored in a file with CSV format.

2.2 Method

This research was conducted to make a forecasting model of currency exchange rates. In this case, the exchange rate of the currency to be used is the Indonesian rupiah (IDR) exchange rate against the US dollar (USD).

Normalization in the form of [-1,1] is carried out on exchange rate data. This range of values is adjusted to the activation function that will be used, namely *tanh*.

In the RNN architecture used there are 4 types of layers. These layers are the input layer, hidden layer, an output layer, and a context layer. The neurons in the input layer amount to the size of the sliding window used. An experiment will be conducted on the number of neurons in the hidden layer. The number of neurons 1 until 10 is used in the experiment to see the number of neurons that provide the best results. Neurons in the output layer are one because they are a regression problem. The number of neurons in the context layer will follow the number of neurons in the hidden layer for *Elman Networks*. The context layer itself is used to store the value of the hidden layer on timestep t-1 for reuse as input to the hidden layer on timestep t. The value in the context layer will continue to be updated until the training process is complete.

Training data is the data used during the training process in the RNN-EKF system. From the total data available, the percentage of training data is 70% of the total data. Training data that enters the network is 760 data points.

The learning algorithm that will be used in ANN is EKF. The equation for measurement updates from EKF can be simplified to [10]:

$$K_t = P_{t-1}H_t^T(H_tP_{t-1}H_t^T + R_t)^{-1} \quad \text{.....(1)}$$

$$\hat{x}_t = \widehat{x_{t-1}} + K_t(z_t - h_t(\hat{x}_t u_t)) \quad \text{.....(2)}$$

$$P_t = P_{t-1} - K_tH_tP_{t-1} + Q_t \quad \text{.....(3)}$$

With K_t is the gain, P_t is automatic covariance error, H_t is the Jacobian matrix, R_t is the measurement noise covariance matrix, $\widehat{x_t}$ is the new weight prediction, Z_t is the desired output, Q_t is the process noise covariance matrix.

The learning parameters in the RNN in this study are the value of the *Q noise covariance process*, *measurement noise covariance R*, and the number of epochs (up to 1000 epochs).

Process noise covariance Q is a matrix of size $n_W * n_W$ where n_w is the sum of all weights on ANN. Measurement noise covariance R is a variable size matrix where it is the number of neurons in the output layer. Q and R have a value of $n * I$ where n is a scalar and I is an identity matrix. Will be tested against the value of n to determine the effect of the value of Q and R on the results of predictions. Data for testing needs to be done normalizing the forecasting data. Testing is done by calculating the value of RMSE and Dstat.

3. RESULT AND DISCUSSION

In ANN that uses the EKF learning algorithm, the process noise Q covariant value, and measurement noise R covariance value replace the learning rate function in the ANN that uses the SGD learning algorithm. Therefore, it is necessary to know the right Q and R values so that the training process can run optimally.

The test is carried out with values of 1, 0.1, 0.01, and 0.001 for the values of Q and R. Other parameters will be regulated constantly, a namely sliding window of 1, 2 units of neurons in hidden layers, and maximum epochs of 1000. Figure 1 is a training loss graph for Q variation values.

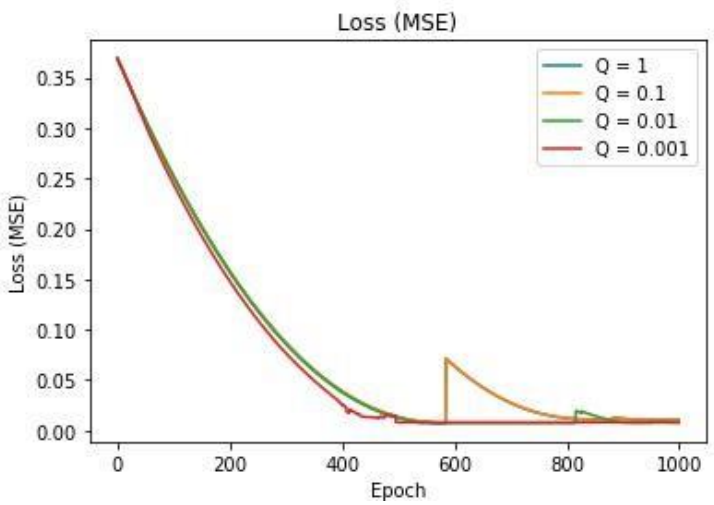

Figure 1 Loss training graph for each Q value

It can be seen that there are some instabilities in each Q value but the loss is stable to converge when it approaches the epoch 1000 value. By looking at the loss graph, the Q value of 1 is chosen as the EKF parameter because it produces a more stable loss graph compared to 3 other Q values.

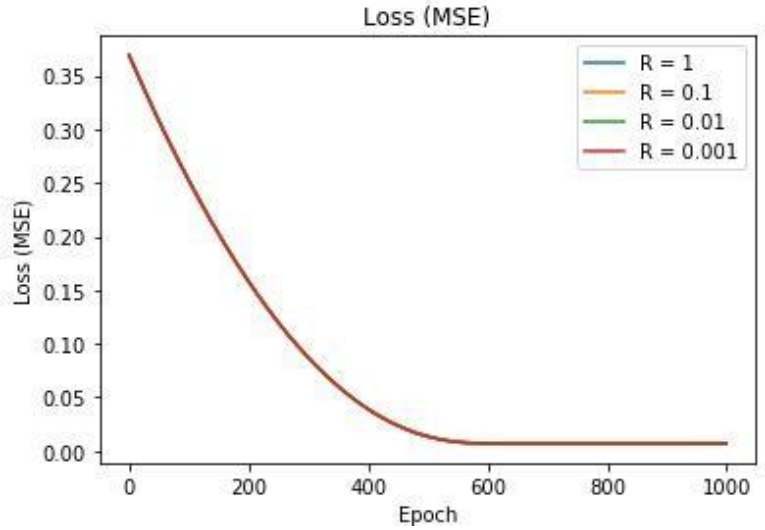

Figure 2 Loss training graph for each R-value

Figure 2 is a training loss graph for the R-value of the training loss value for each R-value, and it can be seen that the R-value does not have a significant effect on network loss when Q is 1. Therefore the value is determined R is 1 as the EKF parameter.

Tests are carried out with fixed parameters Q and R and with the variation of sliding windows from 1 to 8. Dstat results from 10 trials for each architectural test are presented in Table 1. We use Dstat to observe the accuracy primarily because the data is a financial instrument so this parameter is the most appropriate one.

Table 1. Results of the Dstat for variation of the sliding window

Experiment	Sliding Window							
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	60,62	58,95	58,20	58,69	55,45	53,75	51,72	54,08
2	59,70	58,64	68,11	60,55	58,88	58,44	57,05	59,43
3	60,92	55,86	60,99	64,91	58,78	59,38	60,18	56,28
4	59,07	57,09	53,25	57,45	53,27	51,56	56,74	57,54
5	59,07	57,40	55,73	55,28	61,06	53,44	63,22	58,18
6	60,61	61,11	55,11	59,01	60,44	56,56	60,82	62,57
7	62,46	58,95	65,94	62,11	59,50	61,25	58,62	61,32
8	60,30	68,20	64,09	60,56	61,06	56,25	60,82	58,80
9	62,15	52,77	58,20	52,17	51,09	56,25	59,56	58,81
10	65,53	61,11	64,70	59,68	56,07	60,31	59,56	63,20
Maximum	**65,53**	68,20	**68,11**	64,91	61,06	61,25	63,22	63,20
Minimum	59,07	52,77	53,25	52,17	51,09	51,56	51,72	54,08
Average	**61,04**	59,01	**60,43**	59,04	57,56	56,72	58,83	59,02

The summary for the sliding windows testing is presented in Table 2.

Table 2 Test results for sliding windows

Parameter	Sliding Window 1			Sliding Window 3		
	Min	Max	Mean	Min	Max	Mean
RMSE	49,82	74,04	62,09	57,36	88,20	72,39
Dstat (%)	59,07	65,53	61,04	53,25	68,11	60,43

The number of neurons tested is 1 to 10 neuron units in one hidden layer. Testing is the same as testing the size of a sliding window, where 10 attempts will be used for each neuron value. The Dstat results for each neuron are presented in Table 3.

Table 3 Results of Dstat(%) on sliding windows 1 and 3

Number of Neurons	Sliding window 1			Sliding window 3		
	Min	Max	Mean	Min	Max	Mean
1	59,38	63,08	60,52	56,04	66,56	61,24
2	59,07	65,53	61,04	53,25	68,73	60,83
3	59,07	63,08	60,58	56,66	70,58	62,48
4	59,07	63,38	60,89	52,63	67,49	62,14
5	56,01	63,69	60,77	58,51	72,76	65,08
6	58,77	63,38	61,44	60,99	73,06	**65,42**
7	54,46	**67,38**	61,57	52,63	63,16	58,73
8	59,08	63,08	60,99	51,08	**73,07**	61,30
9	54,10	66,46	**61,62**	52,63	72,45	61,12
10	59,08	63,38	60,71	52,32	68,73	60,22

The best results for the use of 1 sliding window were found in the application of 7 units of neurons at Dstat of 67.38%. Whereas for the average Dstat the best is obtained by applying 9 units of neurons at 61.62%. The best results for the use of sliding window with size 3 were found in the application of 8 units of neurons by 73.07% but the value was not much different when applying neurons as much as 6 units of 73.06%. Whereas for the best average Dstat obtained on the application of neurons as much as 6 units of 65.42%. So the best architecture for the RNN-EKF, in this case, is 3-6-1. The summary accuracy data for a number of neurons for 1 and 3 sliding windows, with RMSE data, are presented in Table 4.

Table 4 Accuracy data for the number of neurons with RMSE data

Parameter	Sliding window 1 with 6 neurons			Sliding window 3 with 8 neurons		
	Min	Max	Mean	Min	Max	Mean
RMSE	49,82	74,04	62,09	57,36	88,20	72,39
Dstat (%)	54,46	65,53	61,57	51,08	73.08	61,30

The model produced will be compared with models trained using RNN-SGD. This test is conducted to find out which model is better used to forecast currency exchange rates.

The architecture that will be used is the 3-6 - 1 architecture, similar to the architecture in the RNN-EKF model. From 10 times the experiment with a maximum epoch of 1000, the results obtained as shown in Table 4.

Table 5 Results of RNN SGD accuracy

Experiment	RNN - SGD	
	Dstat (%)	RMSE
1	43,65	141,30
2	45,51	222,01
3	44,58	167,38
4	43,96	225,75
5	46,74	**92,63**
6	**50,77**	107,97
7	46,43	200,27
8	47,36	123,26
9	43,65	104,57
10	45,82	180,70
Minimum	**43,65**	**92,63**
Maximum	**50,77**	**225,75**
Average	**45,85**	**156,58**

In Table 5, the maximum Dstat of 50,77% is achieved with 6 nodes in the hidden neurons. The average Dstat accuracy is 45,85%. This Dstat accuracy is less than accuracy achieved by RNN-EKF with a maximum accuracy of 73.08% and average 61.30 %.

The minimum RMSE for RNN-SGD is 92,63 with an average of 156,58. This RMSE accuracy is much higher than achieved from RNN-EKF model which the minimum RMSE is 57,36 and the average is 72,39. From the Dstat and RMSE values achieved by RNN-EKF and RNN_SGD, it can be seen that RNN-EKF model having higher accuracy than RNN-SGD.

4. CONCLUSIONS

The results of this study indicate that the EKF RNN provides high accuracy, and higher accuracy compared to RNN-SGD with a considerable increase in accuracy in Dstat and a considerable reduction in RMSE. Further research can be done on other learning algorithms on Kalman Filters such as the *Unscented Kalman Filter* or other learning algorithms.

REFERENCES

[1] A. Hector, M. Claudio dan S. Rodrigo, 2002, Artificial Neural Networks in Time series Forecasting: A Comparative Analysis, *Kybernetika*, Vol. 38, No. 6, pp. 685-707.

[2] S.E. Rumagit, 2012, Prediksi Pemakaian Listrik dengan Menggunakan Jaringan Syaraf Tiruan dan ARIMA di Wilayah Suluttenggo, *Tesis*, Program Studi S2 Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta.

[3] E. Munarsih, 2011, Penerapan Model ARIMA - Neural Network Hybrid Untuk Peramalan Data Time Series, *Tesis*, Program Studi S2 Matematika, Universitas Gadjah Mada, Yogyakarta.

[4] D.U. Wutsqa, R. Kusumawati dan R. Subekti, 2014, The Application of Elman Recurrent Neural Network Model for Forecasting Consumer Price Index of Education, Recreation, and Sports in Yogyakarta, *IEEE. 10th International Conference on Natural Computation*, 192-196.

[5] A. Nandury dan L. Sherry, 2016, Anomaly Detection in Aircraft Data Using Recurrent Neural Network (RNN), *Integrated Communications,Navigation, and Surveillance (ICNS), Herndon*, VA, pp. 5C2-1-5C2-8.

[6] L. Susanti, A. Fariza, Setiawardhana, 2011, Peramalan Harga Saham Menggunakan Recurrent Neural Network dengan Algoritma Backpropagation Through Time (BPTT), *Skripsi*,Politeknik Elektronika Negeri Surabaya, Surabaya.

[7] J.A. Perez-Ortiz, J. Calera-Rubio dan M.L. Forcada, 2001, Online Text Prediction with Recurrent Neural Network, *Neural Processing Letter 12: 127-140*, Kluwer Academic Publisher, Netherlands.

[8] F. Syahrian, 2016, Perbandingan Metode Optimasi Stochastic Gradient Descent, Adadelta, Dan Adam Pada Jaringan Saraf Tiruan Dalam Klasifikasi Data Aritmia, *Tesis*, Program Studi S2 Ilmu Komputer, Universitas Gadjah Mada, Yogyakarta.

[9] A.S. Prabowo, A. Sihabuddin, A.S. Nugrahito, 2019, Adaptive Moment Estimation On Deep Belief NetworkFor Rupiah Currency Forecasting, *Indonesian Journal of Computing and Cybernetics Systems (IJCCS)*, Vol.13, No.1 pp. 31~42, Yogyakarta.

[10] W. Xu, 2011, Towards Optimal One Pass Large Scale Learning with Averaged Stochastic Gradient Descent, *arxiv*, July 13

[11] M. Cernansky dan L. Benuskova, 2003, Simple Recurrent Neural Network Trained By RTRL and Extended Kalman Filter Algorithm, *Neural Network World*, 13(3), pp. 223-234.

[12] P. Trebaticky, 2005, Recurrent Neural Network Training with Extended Kalman Filter, M.Bielikova (Ed.), *IIT.SRC 2005*, April 27, 2005, pp. 57-64.

[13] R. Adnan, F.A. Ruslan, A.M. Samad dan Z.M. Zain, 2013, New Artificial Neural Network and Extended Kalman Filter Hybrid Model of Flood Prediction System, *IEEE 9th International Colloquium on Signal Processing and its Applications*, 8 - 10 Maret 2013, Kuala Lumpur, Malaysia.

[14] A.N. Cernodub, 2014, Training Neural Network for Classification Using The Extended Kalman Filter: A Comparative Study, SSN 1060 992X, *Optical Memory and Neural Networks (Information Optics), Vol. 23, No. 2, pp. 96- 103*, Allerton Press, Inc.