A NEURO FUZZY SYSTEM BASED INFLATION PREDICTION OF AGRICULTURAL COMMODITIES.

1st Abhishek Dutta, 2nd Abhisek Nayak, 3rd Aditya, 4th Rama Ranjan Panda, 5th Naresh Kumar Nagwani
Department of Computer Science and Engineering.
National Institute of Technology
Raipur, India
abhi97.dutta@gmail.com, abknayak26@gmail.com, adityachandrakar1998@gmail.com,
rrpanda.phd2018.cs@nitrr.ac.in, nknagwani.cs@nitrr.ac

***Abstract*—Predictions based on Sequential Data such as time-series data of agricultural product prices play a crucial role in agriculture-based business. Determination of inflation in prices help farmers and associated businesses to take corrective measures for higher returns. However, unavailability of enough collective and accurate data for Indian Markets challenges accuracy. This paper captures the advantage of NN (Neural Networks) and FZ (Fuzzy Systems) for predictions based on time series analysis with limited data. NN learn by adjusting the weights between connecting neurons. This helps in pattern recognition of similar data points. Recent developments in DL (Deep Learning) such as the RNN (Recurrent Neural Network) variant, LSTM (Long Short Term Memory) dominates the trade market predictions. LSTM solves the gradient descent problem of traditional NN and remembers temporal patterns. FZ, on the other hand, helps in making inference about human cognition through membership functions. Learning capabilities of NN and Fuzzy rules form the novel Neuro-Fuzzy system termed as FLSTM (Fuzzy-LSTM). Further, the data set contains monthly wholesale prices published by the Ministry of Commerce and Industry, Govt. of India for essential agricultural commodities. The evaluation based on the proposed work shows decent improvement than some standard DL model for various entities when subject to limited records.**

***Index Terms*—Time-Series analysis, Fuzzy Logic, Deep Learning, LSTM, RNN, Neuro-Fuzzy.**

I. INTRODUCTION

India being an agrarian economy and one of the largest exporters of agriculture-based products involves many businesses related to it. The growth of crops depends on various factors from government policies to climate conditions. These directly or indirectly affect the prices of agricultural products in the market. The government issues certain indexes for regulating prices of agricultural-based products per month. Their are handful of such indicators, one such index is the WPI (Wholesale Price Index) published by the Ministry of Commerce and Industry, Govt. of India [1]. These indexes are useful for marketing decisions and predicting uncertainty in the market especially for medium-sized business. Global investors also track WPI as one of the key macro indicators for their investment decisions. WPI estimates the wholesale price before commodities can be used for commercial purposes. However, these estimates consider a base year that gets revised regularly for a period of 5-10 years. Revision of values for such limited time frames generates limited data. Predictions based on such limited data set results to low accuracy. Such data however, shows certain correlations of prices depending on the season or month. This work focuses in extracting those correlations and predicting inflation with minimal data set. The study inspires from recent developments in time series analysis as discussed in [2–6].

This paper is structured as follows. Section-II presents prior findings in this domain and motivation to overcome the drawback of present approaches. Section-III illustrates the model structure and working through flow diagram along with algorithms. Section-IV presents a brief example to develop understanding of the model and study. Section-V shows the experimental results and extracts the necessary findings from it. Further, Section-V determines the conclusion derived from the study and marks further improvements in the same.

II. RELATED WORK

Prior works suggest that NN are good at pattern recognition and establishing non-linear relationships through adjusting the weights between the decision entities called neurons [7]. Further, advancements in DL has significantly improved their performance and are used widely for time-series analysis. These standard DL models generates good results but require sufficient data points to make proper predictions. These models also neglect certain temporal changes to favour major changes . Further, they make predictions computation intensive and are often time consuming.

Fuzzy logic on the other hand, exploits the partial membership to a category [8]. This helps to map ill-defined relations to structured rule. Further, through these rules closest values can be predicted [9]. The Fuzzy based systems hence helps to make prediction of unstructured data fast even with limited data.

Both the approaches being used extensively for time-series analysis has certain advantages and limitations as mentioned in [10]. These techniques however lacks accuracy when the data is scarce and often requires more processing. Reducing computation time and provide good performance on small data set is the key motivation behind this work. Further, for data with seasonal dependencies through out the year like the WPI with limited data values, a new method has to be devised using the combination of both approaches to overcome the drawbacks.

III. Model Design and Problem Evaluation

Utilizing the advantages of both the standard concepts as mentioned in Section-II, this work introduces a novel sequential LSTM and Fuzzy based model termed as FLSTM. In this section, we introduce the design of this proposed FLSTM model to determine the inflation in prices for agricultural commodities through the WPI data set. The predictions are first evaluated using the fuzzy rule based system and then passed to an LSTM model. The fuzzy rule base system extracts the temporal behaviour from the smaller data set. The LSTM model further enriches the predictions and generates accurate results. The flow diagram 1 describes the structure of the overall model. The following sub-section illustrates the detailed sub-parts and working of it.

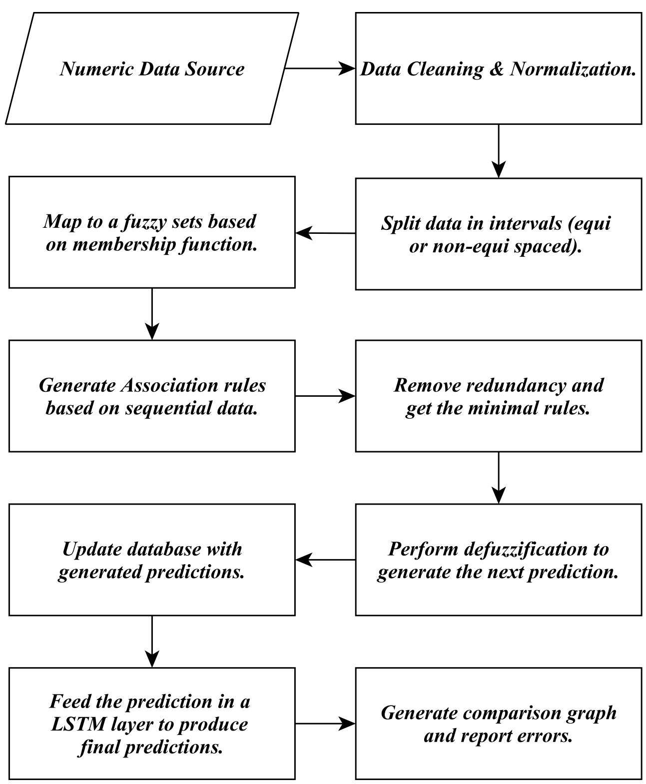

Fig. 1. Flow Diagram

A. Numeric Data Source

Monthly commodity prices in terms of WPI gives the input data for processing. Correlation of the function with it's delayed copy gives this analysis. This filters data values with correlation of certain threshold for a lag of about 12. Observations show that the prices hike in certain months of year and hence this lag captures the trend for a year. Such calculation of linear correlation follows Pearson correlation. This gives values between $(0,1)$. A good correlation produce values closer to 1 and vice versa. Following equations 1-3 explain its processing for the series $y_1, y_2, y_3, ..., y_n$. The equation 1 gives the mean of the series.

$$\overline{y} = \frac{1}{n}\Sigma_{i=1}^{n} y_i \tag{1}$$

For a time lag of k where $k \geq 0$, the equation 2 shows the variance of the series denoted by s_k.

$$s_k = \frac{1}{n}\Sigma_{i=1}^{n-k}(y_i - \overline{y})(y_{i+k} - \overline{y}) \tag{2}$$

For the time lag of k and where s_0 gives the variance of entire series, the equation 3 shows the correlation.

$$r_k = \frac{s_k}{s_0} \tag{3}$$

The graphical representation of this correlations r_k is called Correlogram. The figure 2 shows the correlation for a sample input of onion prices. This filters data greater than threshold values for input to the next stage. This also requires extrapolation of data for about 20% to keep predictions in the universe of discourse U.

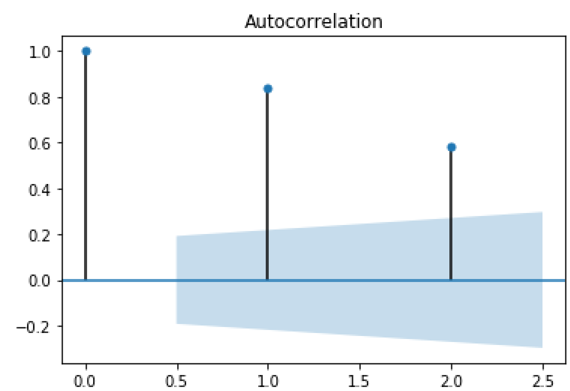

Fig. 2. Flow Diagram

B. Data Normalization

This part improves the input data by scaling the values to standard values. Values may take wide range which are narrowed down to scale of 0 to 1. This work uses decimal scaling to scale the data.

C. Data split & Membership Function

Classification of entities to ill-defined categories helps generate relationships rules. This breaks the data to certain number of sets. Each data point is assigned to a particular set based on membership function [11]. This study consider equi-spaced partitions technique called the grid based partition scheme. A partition factor of 12 is used to map the prices for 12 months. Membership functions such as Triangular, Trapezoidal or Gaussian can map the entities [12]. However, Triangular membership function shows best performance over SD. The equation 4 states the triangular membership scheme. The notation $\mu(x)_i$ determines the membership of an element x to the fuzzy set i, where a and b are lower and upper limit. m is average of the two limits.

$$\mu(x)_i = \begin{cases} 0, & x \leq a \\ \frac{x-a}{m-a}, & a \leq x \leq m \\ \frac{b-x}{b-m}, & m < x < b \\ 0, & x \geq b \end{cases} \tag{4}$$

The figure 3 shows a sample grid partition with triangular membership function where data is divided into 6 categories namely, VL (Very Low), L (Low), NL (Normal Low), NH (Normal High), H (High), VH (Very High).

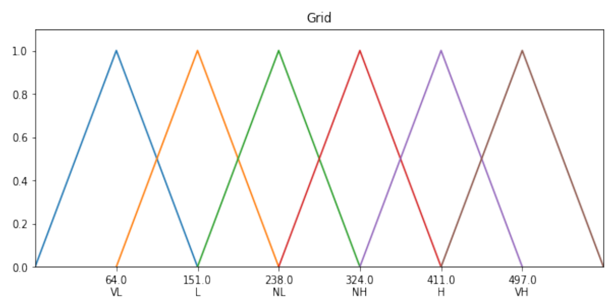

Fig. 3. Grid Partition Scheme with Triangular Membership Function.

D. Rule Generation & Minimization

This sub-section generates the relationship rules between various categories based on the data set. For example, a relation $NH \rightarrow H$ states that the determinant of NH value produces a H as consequent. The predictions uses these generated rules. These rules are determined by the relation between the data points. Two major factors affect the predictions. These are order and method applied. For this study time-series analysis, order of 2 produced best results. The order determines the number of input values presented before the prediction is made. Further, HOFTS (Higher Ordered Fuzzy Time Series resulted greater performance than the other methods such as Weighted-HOFTS (WHOFTS), Probabilistic-WHOFTS (PWHOFTS), Multivariate Fuzzy Time Series (MVFTS), Weighted-MVFTS (WMVFTS) and Granularity-WMVFTS (GWMVFTS) as in [13–18]. Table I shows different measures such as RMSE (Root Mean Squared Error), MAPE (Mean Absolute Percentage Error) and U (Uncertainty coefficient) for different Uni-variate methods. Also fig. 6 shows comparison for multi-variate methods.

E. Update Database

The relationship rules generate prediction against the test cases. Hence, this adds the prediction for time $t+1$ to the data set. Also, the least used data value is discarded. For summary, Algorithm 1 shows the working of FZ in brief. Further, the updated data set works as input to the LSTM layer. That is, this transfers the predictions for $(t+1)^{th}$ time, generated by the Fuzzy unit. For example, if the data set contains 1 to n values, updated data set contains 2 to $n+1$ data points. LSTM requires large data set to determine temporal behaviours. However, this updated data set transferred to LSTM part already contains the temporal relations. Hence, can result to good accuracy even with limited data points.

TABLE I
COMPARISON BETWEEN DIFFERENT METHODS AND ORDER

Method	Order	RMSE	MAPE	U
HOFTS	1	63.62	23.39	1.05
	2	**47.70**	**19.82**	**0.81**
	3	51.27	20.19	0.88
	4	53.15	20.14	0.98
WHOFTS	1	63.19	20.2	1.04
	2	**48.1**	**17.31**	**0.82**
	3	51.61	18.54	0.89
	4	52.36	18.59	0.96
PWHOFTS	**1**	**71.28**	**19.72**	**1.16**
	2	162.84	33.16	2.7
	3	168.09	36.07	2.79
	4	173.26	39.84	3.07

Algorithm 1: Working of Fuzzy unit of the proposed system.

Output: Prediction for input to LSTM unit.
Input: Time series data.
while *true* **do**
- Determine correlation between data point ;
- Normalize the data ;
- Determine the partition Scheme;
- Apply membership function for partition;
- Generate time variant relationship rules;
- Remove redundancy;
- Make predictions for time $t+1$;
- Update database & transfer it to the LSTM unit;

end

F. LSTM Analysis

This sub-section describe the working of the LSTM unit as in [19]. The time series analysis performed using LSTM have internal mechanisms known as gates that regulates the flow of information.

$$f_t = \sigma(W_{vf} * v_t + W_{hf} * h_{t-1} + b_f) \tag{5}$$

$$i_t = \sigma(b_i + W - vi * v_t + W_{hi} * h_{t-1}) \tag{6}$$

$$\tilde{c_t} = \tanh(b_c + W_{vc} * v_t + w_{hc} * h_{t-1}) \tag{7}$$

$$c_t = i_t * \tilde{c_t} + f_t * c_{t-1} \tag{8}$$

$$o_t = \sigma(b_o + W_{vo} * v_t + W_{ho} * h_{t-1}) \tag{9}$$

$$h_t = \tanh(c_t) * o_t \tag{10}$$

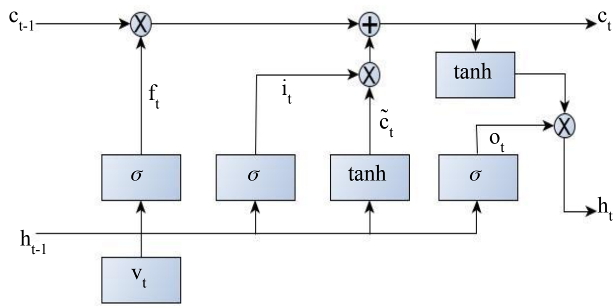

Fig. 4. LSTM Architecture

The equations 5 to 10 explains the work in detail, where v_t is the recurrent layer input, h_t is recurrent unit output, and W is the weight matrices.

The fig. 4 shows the architecture for an LSTM cell. Further, algorithm 2 explains the cell state calculations that selects the next cell state and predicts the output value.

Algorithm 2: Calculation of cell state values.

Output: Determination of cell state c_t and value h_t, the next state should have from previous values.
Input: Previous c_t and h_t values along with current input V
c_t = [0, 0, ..., 0];
h_t = [0, 0, ..., 0];
while v_t *in* V **do**
  $combine$ = h_t + v_t;
  f_t = forget_layer($combine$);
  $\tilde{c}_t$ = candidate_layer($combine$);
  i_t = input_layer($combine$);
  c_t = c_t * f_t + $\tilde{c}_t$ * i_t;
  o_t = output_layer($combine$);
  h_t = o_t + $\tanh(c_t)$;
  if $h_t > t$ **then**
    select that hidden state as the next state;
  else
    Save h_t and h_t values for next iteration;
    continue to loop;
  end
end

Cell state is the most important part of LSTM architecture, denoted as c_t, where t represents the timestamp. Three different gates, namely, the forget-gate, the input-gate, and the output-gate evaluates the cell state c_t and output h_t. The forget-gate, f_t, decides which value from previous data to forget or remember. The input-gate, i_t selects the input signal that updates the values of current cell state. The output-gate, o_t allows the cell state to determine whether it has effect on other neurons or not. It generates the output considering the dependencies through activation function at gates. This hence, suits for modeling of sequential data with long-term dependencies. Moreover, it prevents the vanishing gradient problem of RNN.

The values v_t and h_t concatenated as $combine$ and fed into the $forget\_layer$ which in turn removes any unnecessary data. A $candidate\_layer$ is created using $combine$. The $candidate\_layer$ holds possible values for combining with the cell state. Moreover, $combine$ is further supplied to the $input\_layer$. This layer selects the data from the $candidate\_layer$ that should be added to the next cell state. After computing the $forget\_layer$, $candidate\_layer$, and the $input\_layer$, calculations using these newly generated values and the previous cell evaluates the next cell value.

G. Prediction

This part presents the final result produced after evaluation at the Fuzzy and LSTM units respectively. If k values are presented to the LSTM unit, it generates the output for the $(k+1)^{th}$ unit. Hence, for P test cases, the system generates output for the next $(P-1-Q)$ unit, where Q is the values required for next prediction in the LSTM unit.

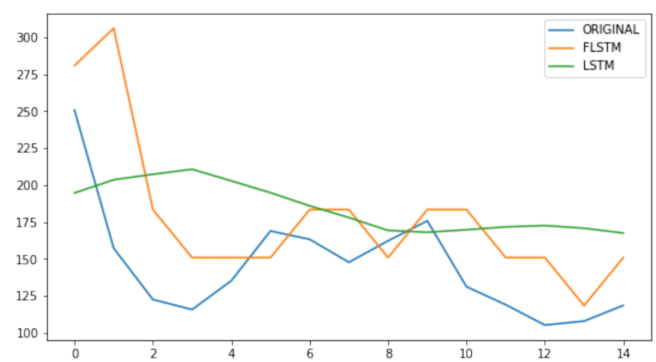

Fig. 5. FLSTM and LSTM comparison.

The fig. 5 shows the output graph for FLSTM system and its comparison with expected values. This also shows the graph generated from the predictions of LSTM system only. This shows that the proposed FLSTM model outperforms the traditional LSTM model.

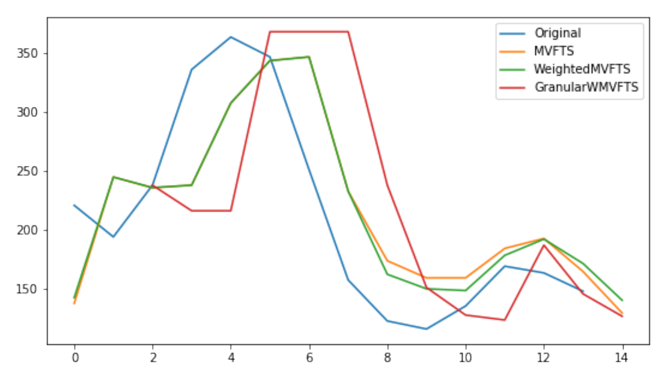

Fig. 6. Multi-variate Results

Further, the graph 6 represents result of FLSTM for Multi-variate methods. This uses multiple parameters based on granularity. The graph uses Cartesian product of Weekly and Monthly data as multiple parameters. The results validate the fact that HOFTS method of order=2 for the FLSTM model gives the best result than any other Uni-variate or Multi-variate method as well as the standard LSTM model. Hence, the system forms the basis for FLSTM (HOFTS of order=2 + LSTM).

IV. Experimental Setup

This section illustrates a working example of the proposed model through processing of various stages wherever possible. Illustration takes a small numeric data of $(id, value)$ pair represented as $S = \{(1,6),(2,7),(3,2),(4,4)\}$ and tries to predict values for $id = 5$. The fig. 7 explains the working in detail.

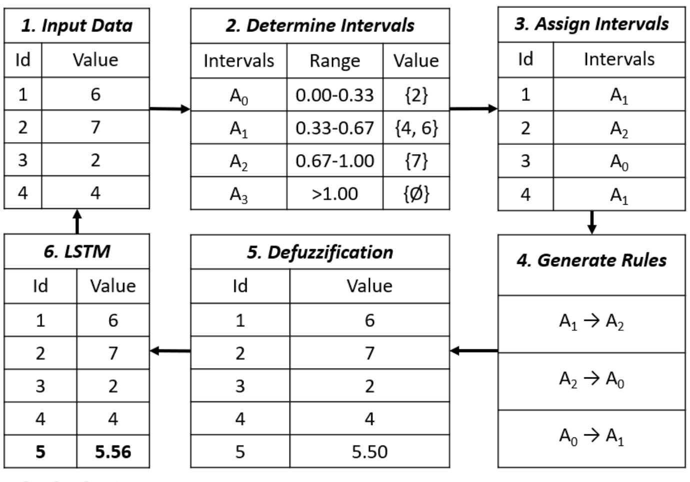

Fig. 7. Example Illustration

The example input S value ranges from 0 to 1 after normalization which is the first step after input to the FZ as explained in 1. The Universe of discourse U is hence $(0, 1)$. Equi-spaced interval A_0, A_1, A_2 and A_3 creates bucket. Further, values of S gets place in appropriate bucket and assigns id to corresponding intervals. Sequential relationship for time-series analysis generates from the association rules. The rules $\{A_1 \longrightarrow A_2, A_2 \longrightarrow A_0, A_0 \longrightarrow A_1\}$ depends on the occurrence of intervals in the time-series. For example, $id = 3$ corresponds to interval A_0 as determinant. The consequent A_1 of $id = 4$ follows as in the time-series bucket A_1 follows A_0. For a new prediction $id = 5$, the system looks for previous $id = 4$ and its bucket A_1. Further, depending on the generated rule it selects appropriate consequent. So, using the rule $A_1 \longrightarrow A_2$, A_2 is selected as the consequent. The average of corresponding $value = 7$ and the value of previous $id = 4$ act as correction for next $id = 5$. The addition of this correction term and value of $id = 4$ gives output as $(4 + \frac{7-4}{2}) = 5.50$ for $id = 5$.

This predicted value get transferred to the LSTM unit. The unit fine tunes with the graph as explained in Section II and results to more accurate prediction of 5.56. The output completes one cycle and adds to the data set. Further, the value corresponding to $id = 1$ is removed from the data set. That is, the window is shifted one step forward to compute the next value. This continues to generates next n values.

V. Experimental Results

This section performs a comparative study between the FLSTM and the standard LSTM models for different agriculture commodities. The comparison is listed in table II where the $Size$ represents the number of training data set. The $Product$ represents different commodities associated to everyday agricultural products. The other two columns give the $RMSE$ of $FLSTM$ and the $LSTM$ model respectively with the original test values. This reinstate the performance of the proposed FLSTM model.

TABLE II
Comparison between FLSTM and LSTM for agricultural commodities.

Product	Size	FLSTM	LSTM
Onion	304	47.70	63.10
Potato	314	46.65	61.12
Tomato	314	41.20	61.10
Brinjal	314	40.81	60.66
Cabbage	314	40.66	60.66
Garlic	314	39.12	58.50
Carrot	324	37.18	58.14
Radish	324	37.22	57.61
Ginger	330	37.14	56.64
Pumpkin	330	37.00	50.58

VI. Conclusion

With the help of both FZ and NN, the proposed architecture performs better then the LSTM models in limited data scenario subject to high degree of correlations. The HOFTS in itself capable of producing good results in limited data cases if the requirement of accuracy is low and fast processing is required. Further, this method with different parameters such as weather can be used to produce more realistic predictions. This can help farmers as well as business to plan their marketing strategies. However, generating accurate rules is the limitation of this work. Probability based fuzzy rules can be explored further for producing more accurate rules. Also, the work is limited to data with more correlations. Reinforcement learning techniques can be tried next to improve the performance and generalize the approach.

References

[1] Data Source. *https://data.gov.in/catalog/annual-wholesale-price-index-agriculture-produce* Wholesale price index of agriculture commodities with 2011-12 as base year. Last accessed on 01-04-2020.

[2] Banerjee, Debanjan and Ghosal, Arijit and Mukherjee, Imon. *Prediction of Gold Price Movement Using Geopolitical Risk as a Factor. Emerging Technologies in Data Mining and Information.* Security(pp.879-886) Springer, Singapore, 2019.

[3] L. Madaan, A. Sharma, P. Khandelwal, S. Goel, P. Singla, A. Seth. *Price forecasting & anomaly detection for agricultural commodities in India.* Proceedings of the 2nd ACM SIGCAS Conference on Computing and Sustainable Societies, 2019.

[4] J. Silva, M. G. Angulo, J. R. Borré, L. P. L. Ayarza, O. B. P. Lezama, Z. C. Martínez, J. N. Beltran. *An Early Warning Method for Agricultural Products Price Spike Based on Artificial Neural Networks Prediction.* Computational Science – ICCS 2019, Lecture Notes in Computer Science, vol 11537, Springer, 2019.

[5] F. Nhita, D. Saepudin, A. Paramita, S. Marliani and U. N. Wisesty. *Price Prediction for Agricultural Commodities in Bandung Regency. School of Computing.* Telkom University, Bandung 40257, Indonesia, Springer, 2019.

[6] Shahriar Yousefi, Ilona Weinreich, Dominik Reinarz. *Wavelet-based prediction of oil prices.* Chaos, Solitons and Fractals 25 (2005) 265–275, Springer, 2019.

[7] Janardhanan, Deepak, and Enda Barrett. *CPU workload forecasting of machines in data centers using LSTM recurrent neural networks and ARIMA models.* 12th International Conference for Internet Technology and Secured Transactions (ICITST). IEEE, 2017.

[8] Lofti A. Zadeh. *Fuzzy Logic.* University of California, Berkeley, IEEE, 1988.

[9] Q. Song and B. S. Chissom. *Fuzzy time series and its models.* Fuzzy Sets Syst., vol. 54, no. 3, pp. 269–277, 1993.

[10] Pabuçcu, H. *Time series forecasting with neural network and fuzzy logic.* Recent Studies in Economics (2017): 195-204, 2018.

[11] Silva, P. C. L. et al. *pyFTS: Fuzzy Time Series for Python.* Belo Horizonte, 2018.

[12] C. H. Cheng, R. J. Chang, and C. A. Yeh. *Entropy-based and trapezoidal fuzzification-based fuzzy time series approach for forecasting IT project cost.* Technol. Forecast. Social Change, vol. 73, no. 5, pp. 524–542, Jun. 2006.

[13] H. K. Yu. *Weighted fuzzy time series models for TAIEX forecasting.* Phys. A Stat. Mech. its Appl., vol. 349, no. 3, pp. 609–624, 2005.

[14] H. J. Sadaei, R. Enayatifar, A. H. Abdullah, and A. Gani. *Short-term load forecasting using a hybrid model with a refined exponentially weighted fuzzy time series and an improved harmony search.* Int. J. Electr. Power Energy Syst., vol. 62, no. from 2005, pp. 118–129, 2014.

[15] R. Efendi, Z. Ismail, and M. M. Deris. *Improved weight Fuzzy Time Series as used in the exchange rates forecasting of US Dollar to Ringgit Malaysia.* Int. J. Comput. Intell. Appl., vol. 12, no. 1, p. 1350005, 2013.

[16] C.-H. Cheng, Y.-S. Chen, and Y.-L. Wu. *Forecasting innovation diffusion of products using trend-weighted fuzzy time-series model.* Expert Syst. Appl., vol. 36, no. 2, pp. 1826–1832, 2009.

[17] K. H. Huarng. *Effective lengths of intervals to improve forecasting in fuzzy time series.* Fuzzy Sets Syst., vol. 123, no. 3, pp. 387–394, Nov. 2001.

[18] S. M. Chen. *Forecasting enrollments based on fuzzy time series.* Fuzzy Sets Syst., vol. 81, no. 3, pp. 311–319, 1996.

[19] Vargas, Manuel R and De Lima, Beatriz SLP and Evsukoff, Alexandre G. *Deep learning for stock market prediction from financial news articles.* IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2017.