PREDICTION FROM 7 YEARS TIME-SERIES RAINFALL DATA

COMPUTATIONAL INTELLIGENCE AND ITS APPLICATIONS
November 25, 2019

Dániel Nánási
UiT The Arctic University of Norway
Department of Computer Science
Email: dna005@post.uit.no

I. INTRODUCTION

In the twentieth century computer industry opened a way in every scientific field which had any high amount of data what unable to process by human mind. First computer huge data processing usage was for summary census data of countries, later science started to use it to support researches. After, computer scientists started to compute the answers for problems in the same way as our brain does, so they started to create models with neural networks, which are computational models based on the structure of biological neural networks. For the cognitive thinking some of these models are using fuzzy sets. Fuzzy logic is based on the observation that people make decisions based on non-numerical information. Fuzzy sets are representing a non exact group or sets of information. For example we cannot program the computer to detect cold or warm temperature exact for people. Because it is subjective, but we are able to make measurements of peoples' subjective opinions, for example most of the people would say it is cold if there is 15 °C in a room, but there would be bigger difference in a 20 °C room. Both neural networks and fuzzy systems are dynamic processing systems that estimate input-output functions without any mathematical model and using training data.

Nowadays we are able to have or rent from the biggest IT companies resource for computing with huge amount or data or/and do a high number of learning iterations. From stock market prices to the spread of an epidemic, and from the recording of an audio signal to sleep monitoring, it is common for real world data to be registered taking into account some notion of time. When collected together, the measurements compose what is known as a Time-Series. These kinds of problems are addressed in the literature by a range of different approaches (for a recent review of the main techniques applied to perform tasks such as Classification, Segmentation, Anomaly Detection and Prediction). [1] The way of learning or training the system is with a deep learning architecture. In a deep learning architecture there are an input layer an output layer and at least one hidden layer. There are connections where the nodes has an activation function which is defined by the designer. The nodes has a bias and the connections have weight which learn in every iteration by a defined method.

This paper talks about a time-series prediction problem, where there is a data sequence with time stamps and we would like to know what will happen in the future with the value what the measured data refers to. We do not know any details, just the data flow and the time difference between the measurements what is constant. The input of our system is one or more data vectors $x = [x_1, x_2, ...x_n]$ for training and we except data after n data with accuracy. The ideal model is where we can predict for the longest time with an accuracy rate above a threshold. Our point is to find the deep learning model which one could give the prediction with the best accuracy for our specific data set.

II. ABOUT THE DATA

This section describes and shows visualisations from the data what this paper discusses. The data is a rainfall value gathered by sensors. The data is gathered on three different stations, call them A, B and C. The data is gathered on all stations from 2010 January 1st midnight to 2016 December 31st midnight. There is 5

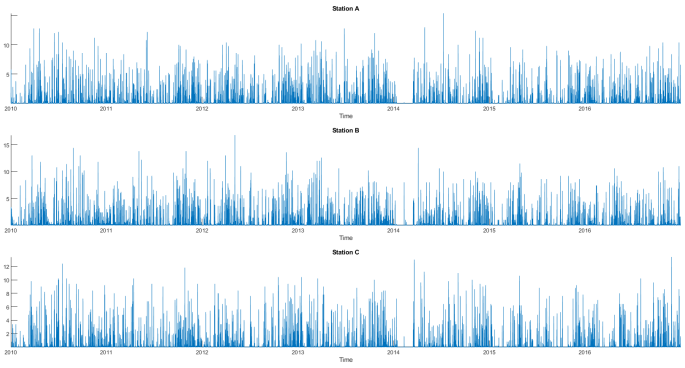

Fig. 1. Rainfall data for 7 years for 3 different stations in Singapore

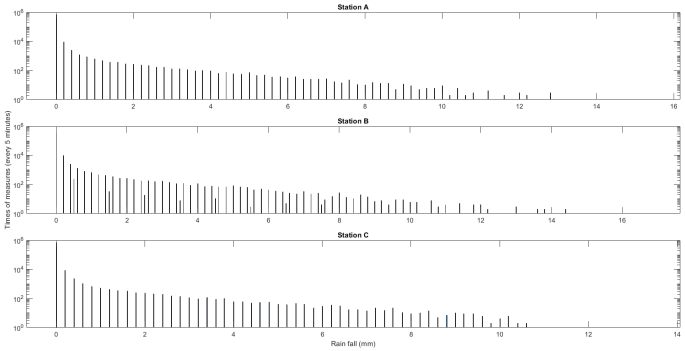

Fig. 2. Histograms of the different stations rainfall data

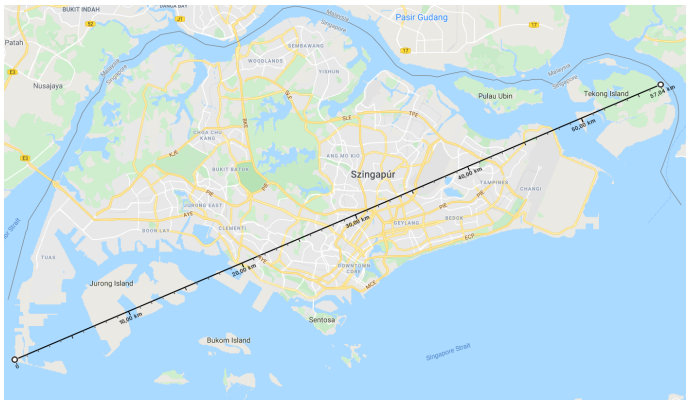

Fig. 3. Map of Singapore with a 60km long black line [2]

minutes difference between the measurements, so there are 6 values for each hour each different stations. So these are time-series data for 7 years on discrete time and continuous values, we do not know anything about the sensors. The raw data with stations A, B and C is visualised on figure 1. This figure shows the rainfall amount in millimeter for 7 years of the three different stations. As we can see these 3 station data are very similar to each other by the randomization and the peak values, but if we compare the parts, we can see these are absolutely different data sets.

There is an another way also to compare the data of the three stations. By the histograms of the data could show us which value of rainfall (mm) have been represented how many times in the measurements. These histograms are shown on figure 2. As we can see, the rainfall detection of each values are very similar, but there are clearly differences. For example Station A and B had more heavy rain (¿10mm) than Station C. There is no information from the positions of the stations or the distance of each stations to each other. We know only the stations are in Singapore and we can assume the distances of two stations cannot be more than 60km by the map shown on figure 3.
After checking the time-series data, we should think about that what periodicity could this data have. From a rainfall data we can assume some periodicity duo to the yearly repeating seasons. So the spectrum of one station is created with Discrete Fourier Transform. By definition

$$X_i = \sum_{n=0}^{N-1} x_n e^{\frac{-j2\pi}{N} in}$$

where N is the length of the data, i is the indexing of the new vector and j is the unit imaginary value. The generated spectrum is shown on figure 4, where the amplitude is highlighted red less frequency than 1/(1 day). As we can see the spectrum is constantly decreasing, which means there is less periodicity on higher frequency, but there are more component in lower frequency especially lower than 1/(1 day) frequencies. So there are more similarity in the data if we check what happens in different times where the time is more than a day.

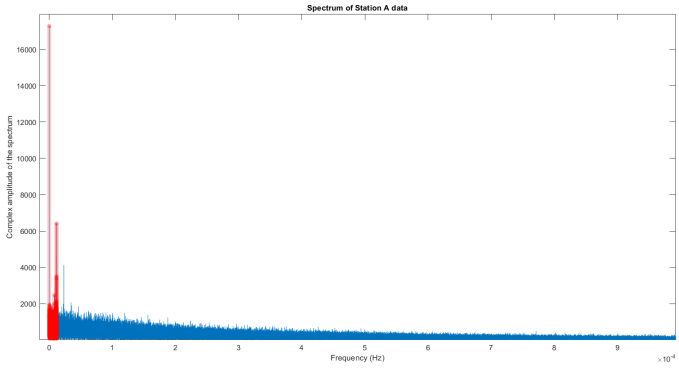

Fig. 4. Spectrum of station A

The excepted values of the stations data are the following:

$$E(A) = 0.0235,\ E(B) = 0.0260,\ E(C) = 0.0201$$

The standard deviations of the stations data are the following:

$$\sigma_A = 0.2722,\ \sigma_B = 0.2968,\ \sigma_C = 0.2468$$

III. Model selection

Before the starting the implementation of a prediction method, there should be gathered the methods for this task and the excepted results should be discussed. Prediction models that estimate the risk of an event of interest are most commonly evaluated with respect to their capacity to discriminate between events and nonevents. [3] There are two ways to a prediction learning method ends up with lower accuracy than it could be. These two ways are under-fitting and over-fitting.

a) Under-fitting: A model that is too simple to capture the underlying model is likely to have high bias and low variance. Training and test error are poorer than desired, adding more training data will not help.

b) Over-fitting: Overly complex models typically have low bias and high variance. Training error is too optimistic, test error is too pessimistic, adding more training data could help.

Under- and over-fitting are common problems in both regression and classification. For example, a straight line under-fits a third-order polynomial underlying a model with normally distributed noise. It is important to evaluate a model on data that were not used to train it or select it. Finding a model with the appropriate complexity for a data set requires finding a balance between bias and variance. It is important to evaluate a model on data that were not used to train it or select it. [4]

IV. Regression analysis

Regression analysis is a form of predictive modelling technique which investigates the relationship between a dependent (target) and independent variable (predictor). This technique is used for forecasting, time series modelling and finding the causal effect relationship between the variables.

The data has been trained with different regression learner models. The independent variables were the data of Station A and Station B and the dependent, predicted data was Station C rainfall data. In a regression training problem we can chose which part of the data we would like to used for training and which (the complement) to forecast. The first trivial option is holdout validation, when the first part of the data (for example 80%) is used for train and the last part is predicted. But why would we predict one big block of the data, especially for a rainfall data, which is chaotic and hard to predict from older samples.

So in this experience the regression models used with cross-validation 14. This means the time interval (7 years) is divided to 14 parts (half years) and the variables are iterating between prediction and training. So the data is divided into 14 random subsets and a total of 14 models are fit, and 14 validation statistics are obtained.

The succes of a model prediction for a specific data is comparable by the Root Mean Square Error (RMSE), which is the standard deviation of the residuals (prediction errors).

$$RMSE = \sqrt{(f - o)^2}$$

where f is the forecasts, expected values or unknown results and o is the observed values, known results.

A. Linear regression

Linear Regression establishes a relationship between dependent variable (Y) and one or more independent variables (X) using a best fit linear line, this is the regression line. [5] The following equation describe the linear regression,

$$Y = a + b * X + e$$

where a is intercept, b is slope of the line and e is error term.

B. Tree technique

Decision trees can be described also as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data, where the data looks like this:

$$(\mathbf{x}, Y) = (x_1, x_2, x_3, ..., x_k, Y)$$

The dependent variable, Y, is the target variable that we are trying to understand, classify or generalize. The vector **x** is composed of the features, $x_1, x_2, x_3 x_1, x_2, x_3$ etc., that are used for that task. A decision tree is a flow-chart-like structure, where each internal (non-leaf) node

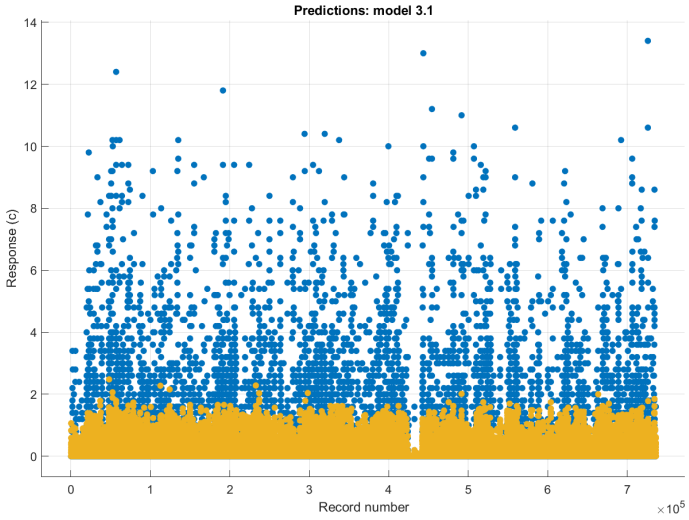

Fig. 5. Response plot of ensemble boosted tree model

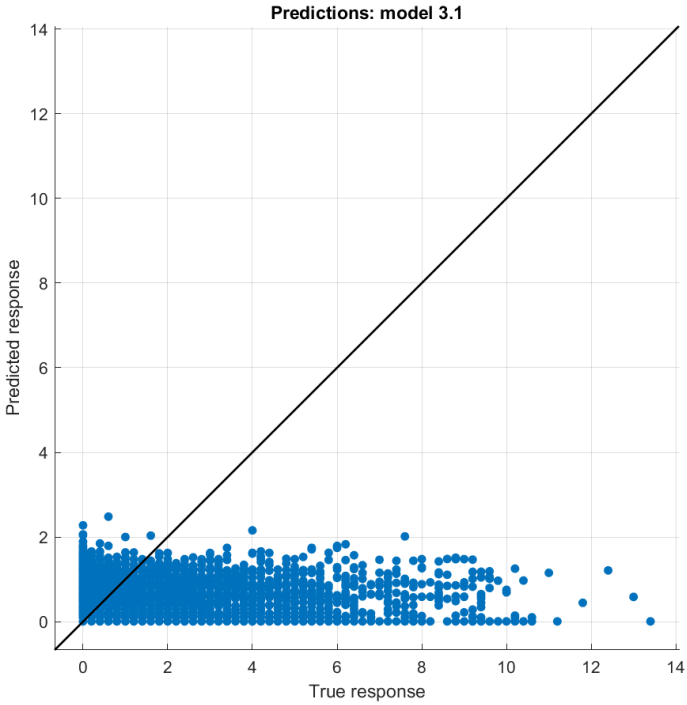

Fig. 6. Regression model plot of ensemble boosted tree model

denotes a test on an attribute, each branch represents the outcome of a test, and each leaf holds a class label. The topmost node in a tree is the root node. [6] There are many different models for Tree techniques regression.

C. Experience

The 7 years rainfall data with the three different stations were tested with many different models. There was not radically big errors with the cross-validation 14 comparing to lower number or holdout validation technique. The lowest $RMSE$ for all models was with ensemble boosted tree, but none of the regression learning models gave a good result. The best result is $RMSE = 0.23121$ and the plot of the predicted data is shown on figure 5. The orange dots are the predicted data and the blue plots are the real data. The difference is very big on the visualisation. As we can see the regression learner does not fit well on these kind of data. However there was heavier rains on Station A and B at the same time the learner gave less heavy rain for Station C many times. As we can see on figure 6, which visualise the model and the difference between the real values and predicted values, we can see as heavier the rain was, the model predicted values with higher error.

V. ANFIS

A. About the model

A fuzzy system based on the logical rules of premises and conclusions cannot be analyzed with traditional probability theories. A system is used in this section to predict rainfall data is Adaptive Neuro-Fuzzy Inference System (ANFIS). The purpose of this model for a prediction problem is to build a system with fuzzy sets and rules which can find out the future data from a data set. Fuzzy sets are helping to define the different weather situations. People defines fuzzy sets for themselves, for example sunny, cloudy, rainy, heavy rainy. We can assume it is not raining (0 mm rainfall) if somebody says it is sunny. But people do not measure exact value of the fallen rain, so if someone says cloudy or rainy it could be more than 0 mm excepted value for the fallen rain. The computer based Neuro-Fuzzy Interferenc System is like this, but not with 5-6 words but more ten or more hundreds of fuzzy sets. For instance, a fuzzy inference system-with two rules
Rule1: if x is A_1 and y is B_1 then $f_1 = p_1x + q_1y + r_1$
Rule2: if x is A_2 and y is B_2 then $f_2 = p_2x + q_2y + r_2$
looks like this. [7]

The ANFIS system is bailed up with neural network layers, where the input has two data set x and y. For this case x is the 7 years long time data and y is a rainfall data of a station. The architecture of the layers is shown on figure 7. The first layer is the input layer and contains adaptive nodes with functions. Each node in this layer corresponds to a linguistic rule, and the output is the value of the membership function. In layer 2, every node multiplies the incoming signals with output given

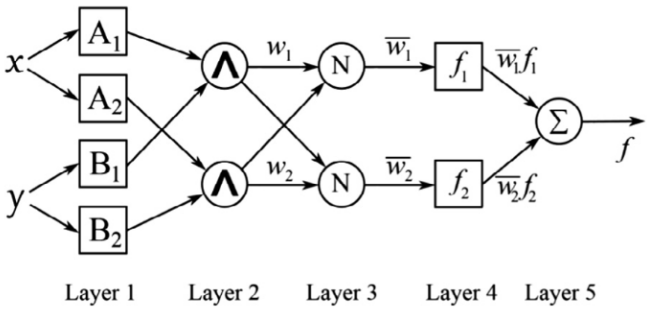

Fig. 7. Architecture of ANFIS [7]

by

$$w_i = \mu A_i(x) * \mu B_i(y)$$

like the rules described. In layer 3, nodes calculate the ratio of the firing strength of their rules to the sum of all the firing strengths like this

$$\bar{w_i} = \frac{w_i}{w_1 + w_2}$$

For the layer 4, the nodes do the following function for the input data:

$$\bar{w_i} f_i = \bar{w_i}(p_i x + q_i y + r_i)$$

In layer 5 the single nodes compute the overall output as the sum of all incoming signals, as follows

$$O = \sum_i \bar{w_i} f_i = \frac{\sum_i w_i f_i}{\sum_i w_i}$$

B. Experiment

For the experiment of the ANFIS model with the rainfall data MATLAB ANFIS modul was used. [8] So as it has mentioned in the model describing the ANFIS is a robust learner architecture. If the data is changing very fast the nodes must be increased for an acceptable accuracy. For a data set like this rain data (figure 1) this model needs more hundreds of fuzzy rules and nodes to predict accuracy more than the linear regression. The training time could be more hours with this scenario on a PC.

What is the configurable details for this architecture?

1) the number of epochs,
2) the number of nodes,
3) the number of fuzzy rules,
4) the membership function type,
5) and the data for train and validate.

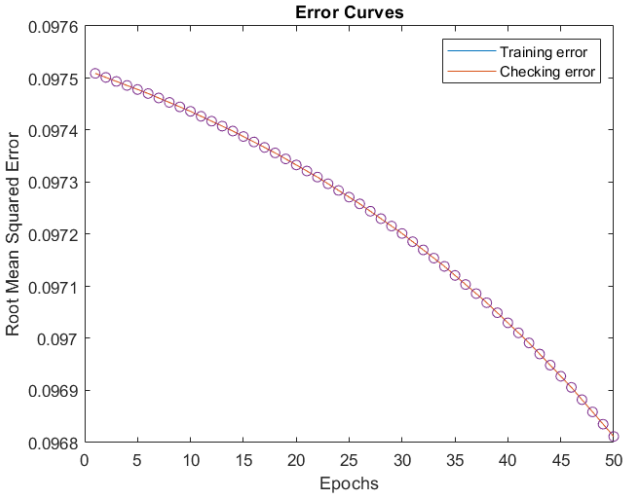

Fig. 8. RMSE decreasing in function of epochs

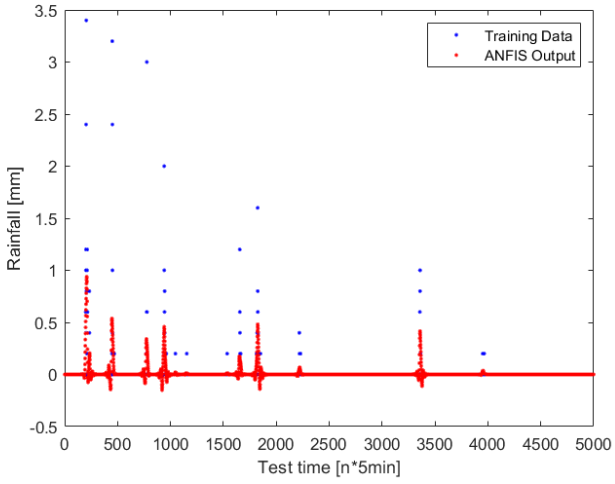

Fig. 9. Response plot for the Station B data with ANFIS architecture

So for this kind of sensitive data set the nodes should be more than 50 or 80. For the epochs number it does not worth it give a very big number of epochs, the RMSE is decreasing after every epoch, but the computation time could be very high and over-fitting can happen also. On figure 8 it is shown how the RMSE decrease after every epoch. The membership function is optional also, but in the experiments trapezoid, bell, or gauss membership functions did not have considerable differences.

RMSE could be lower than 0.1 with high number of nodes, fuzzy rules. But this number is still not too low. There is a response for the predicted data shown on figure 9. The predicted curves converge the original rain data, but there are many local cases where the distance

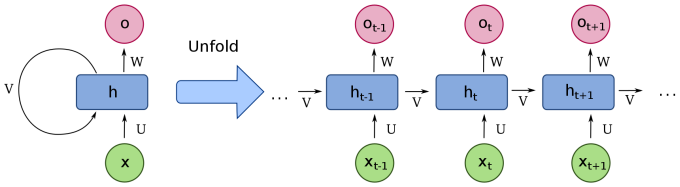

Fig. 10. Recurrent Neural Networks [10]

between the predicted and original data is very big.

According to [9], it is very hard to predict this kind of sensitive data because of its dynamic, nonlinear and complex behavior.

VI. LSTM

A. Conception

As the humans have short and long time memories scientists came to the way deep learning models could have too. The way of the human thinking is a very big mystery for mankind, we do not even know where a thinking or a perception start. But we know there are short term and long term memories and both of them has some kind of forget functions and transient times for forgets. Each type of memories forget a memory if brain does not remember it frequently. The time transients gives the time for a memory cell to forget if it is not remembered by the brain. This is a very simple and surfaced description of human brain and memory and this paper is not about biology, but the LSTM method which is described in this section based by this thoughts.

In the figure 10 shown a chunk of neural network, where h has some input $x(t)$ and outputs a value is $o(t)$. A loop v allows information to be passed from one step of the network to the next. These loops make recurrent neural networks (RNN) seem kind of mysterious. However, if you think a bit more, it turns out that they are not all that different than a normal neural network. A recurrent neural network can be thought of as multiple copies of the same network, each passing a message to a successor. Consider what happens if we unroll the loop. This chain-like nature reveals that recurrent neural networks are intimately related to sequences and lists. They're the natural architecture of neural network to use for such data. [10]

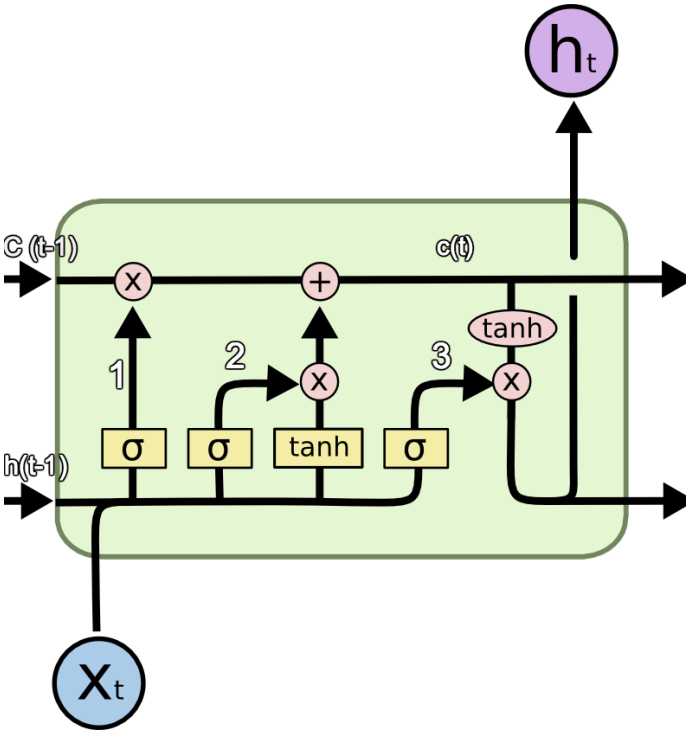

Fig. 11. LSTM learning block [10]

B. What is inside in an LSTM cell

The main idea behind Long Short Term Memory networks (LSTMs) is to combine RNN and forget features of long and short term memory segments. LSTM is a type of RNN, where the learning cell looks like the same on figure 11.

The block has two inputs from the previous block (C_{t-1}, h_{t-1}), two output for the next block (C_t, h_t). The system input is X_t and output data is h_t. The last cell output data concatenates the actual input data for the computing ($[h_{t-1}x_t]$). The numbered function of connections do the main sense of LSTM.

1) The first connection represent the forget function. The forget output is multiplied with the system information before computing the new sequence (C_{t-1}). The output of the first connection is

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$$

where W_f is the weight and b_f is the bias of the sigmoid layer. The sigmoid layer is a neural network with the fire function sigmoid which is

$$\phi(x) = \frac{1}{1 + e^{-x}}$$

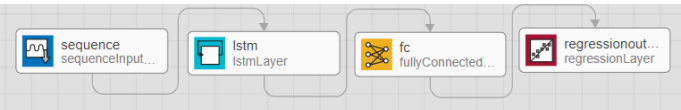

Fig. 12. LSTM network architecture in Matlab

2) Second connection has also a sigmoid layer and a tangent hyperbolic layer too and the output of both layer is multiplied with each other, like this:

$$f_C = \sigma(W_i[h_{t-1}, x_t]+b_i)*tanh(W_C[h_{t-1}, x_t]+b_C)$$

This is the learning process of the LSTM. While the first connection represents the forget function which is multiplied with the previous system data (C_{t-1}), the second connection is for learning and after the forget happened f_C, the learning factor adds to the system value. So hidden system value generated by the following equation:

$$C_t = C_{t-1} \cdot f_t + f_C$$

3) Output data is created by the input and previous data computed with a sigmoid layer and multiplied with the system data after a tangent hyperbolic function. The output looks like this:

$$h_t = tanh(C_t) \cdot \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$$

where W_o are the weights and b_o is the bias of the output neural network with sigmoid fire function. [11]

C. LSTM network architecture

LSTM for prediction needs to make it available for a sequence data, like the rainfall data and it needs a regression layer too. So if we would like to build a simple LSTM model in matlab the layers should be the following

1) sequence input layer
2) LSTM layer
3) fully connected layer
4) regression output layer

as it shown on figure 12. [12]

D. Preparation of the data

The first step with the data is normalization, where the whole data set is divided by the standard deviation of it and subtract the mean of the data set, like this

$$X_i := \frac{X_i - \mu}{\sigma}$$

to get a standardized data. After the process, we have to do the inverse with the output.

After partitioning the data to train and validation data, we have to make different delays in the data for train. For a rain data like this, we do not want to know just about how the next 5 minutes data, but more days or months data. So in the training process, it have to learn more different delays from each delayed data set. For some experiment I have chosen the Fibonacci numbers (under 1000) to make delays in the data and learn to predict to this offsets.

E. Experiment with LSTM

For the experiment of LSTM model one station average day data was used. So the data looked like as the original data, but there were a sample for every day of the 7 years period and the values are the averages of the days.

The training options for the experiment were:

1) 250 of hidden unit in the deep learning architecture.
2) The network was trained with station a, b, c day average data and tested with the last of 1/3 station C data. The data which was tested was not in the training data.
3) And finally the length of the training was 500 epoch.

The results for the prediction is shown on figure 13. On the first diagram the output data is shown with the original data for every day average rainfall. On the second diagram we can see the error on every day and last picture shows the error distribution. The regressions of the LSTM prediction also shown on figure 14. We can see the regression of the trained data and regression of the predicted data. As we can see, it is a further step to ANFIS. The data is more like the original data, than ANFIS prediction, but the tracing is worse, there is a delay and the has a lot of error.

VII. BI-LSTM

A. Architecture

Bi-directional long short term memory, BI-LSTM is an extension of LSTM, based on the same cell architecture. Like LSTM, BI-LSTM is also based on RNN. Bidirectional recurrent neural networks are really just putting two independent RNNs together. This structure allows the networks to have both backward and forward information about the sequence at every time step. The architecture of bidirectional RNN shown

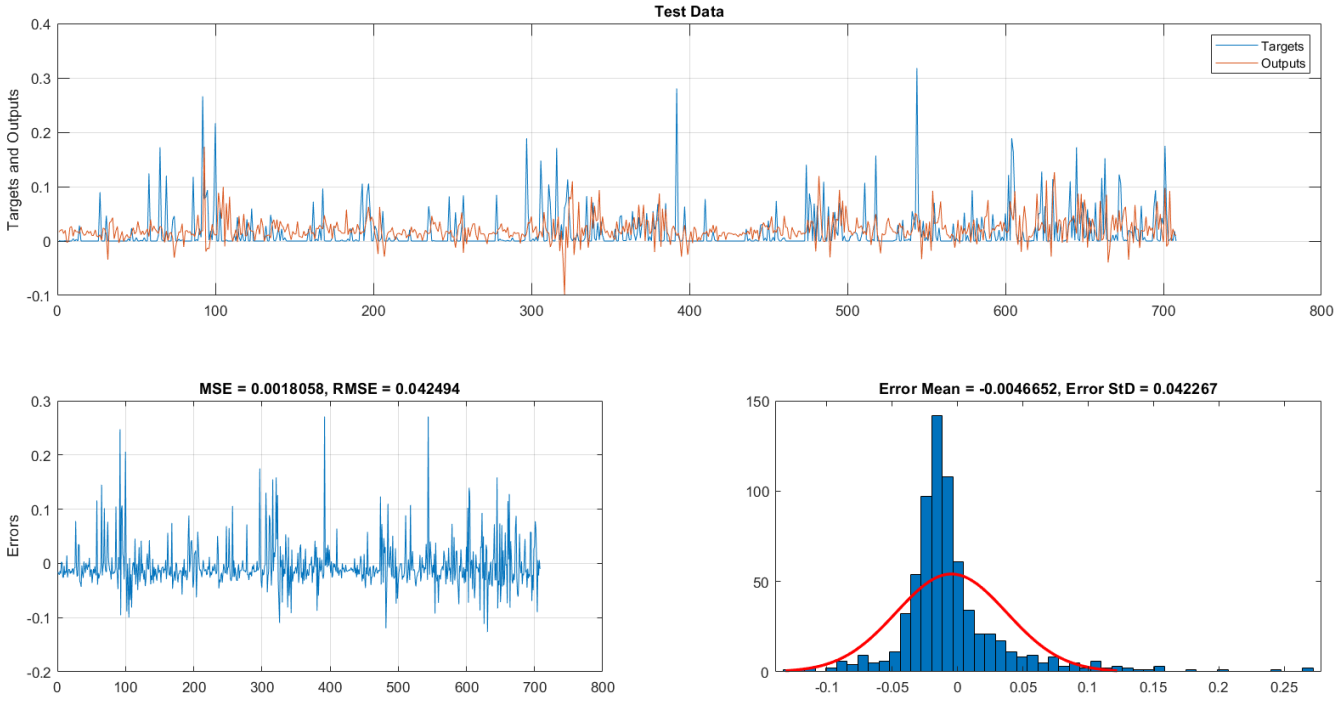

Fig. 13. Error of the LSTM prediction

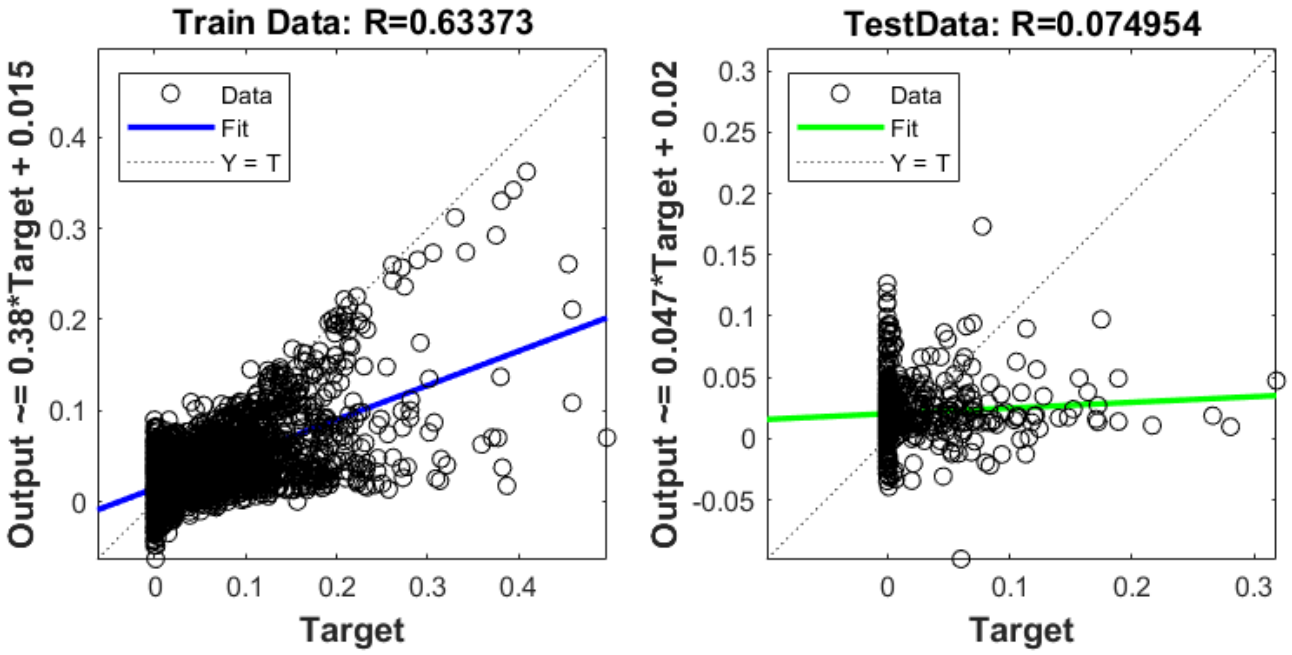

Fig. 14. Regression of the LSTM prediction

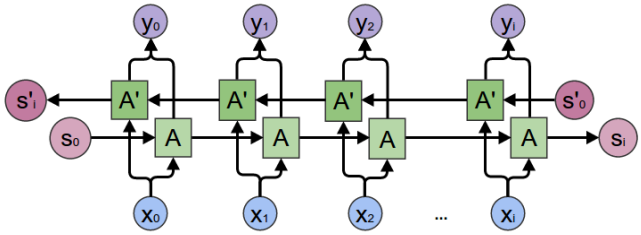

Fig. 15. Bidirectional recurrent neural network architecture

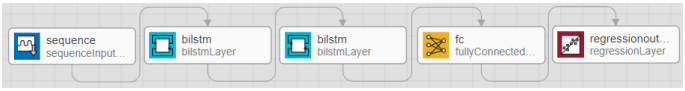

Fig. 16. BI-LSTM simple architecture in a Matlab model

on figure 15.

Using bidirectional will run the inputs in two ways, one from past to future and one from future to past and what differs this approach from unidirectional is that in the LSTM that runs backward you preserve information from the future and using the two hidden states combined we are able in any point in time to preserve information from both past and future. [10]

If we would like to model it in Matlab, the layers we have to use for a simple BI-LSTM architecture are the followings:

1) sequence input layer
2) Bi-LSTM layer
3) Bi-LSTM layer
4) fully connected layer
5) regression output layer

It is shown on figure 16, in Matlab's deep learning toolbox. It involves duplicating the first recurrent layer in the network so that there are now two layers side-by-side, then providing the input sequence as-is as input to the first layer and providing a reversed copy of the input sequence to the second.

B. Experiment with the BI-LSTM model

In this experiment all of the station data was trained in a sequence. And it was tested on the last 25% of Station C data. So it means we are trying to predict 1.75 years of rainfall data for one station by 7 + 7 + 5.25 years of rainfall data. The training options for the experiment were:

1) 60 for both bi-lstm layers' hidden unit in the deep learning architecture.
2) The length of the training was 50 epoch.

Unfortunately Matlab does not support parallel CPU or parallel GPU computation for LSTM or BI-LSTM layers, so the computation time was 14 hour for these low numbers of hidden layers and epochs too.
The results of the testing data is shown on figure 17, where the first diagram shows us 1.75 year of prediction with 5 minutes time steps. The error in the same scale is shown on the second diagram and the error distribution on the third. The regression of the prediction for the test data is shown on figure 18. Where we can see it has a similar error shape like regression, heavier rain was not detected correctly, but this method gave the best response from the methods.

If we check the full 1.75 years of predicted data, we can record that it has less error in the beginning than at the last part of the time period. If we check the data closer on figure 19. It is a zoom from the second month of the predicted period. The shape of the data looks like the original data, it has a bit delay and a bit error between the low and medium rain. But there is no error if we would like to check if it is raining or not, error comes up only in the amount of rainfall. If we check figure 20, we can say the same in the end of the data, around 1.15 years of the predicted data.

C. Summary

To sum it up, BI-LSTM model could give us a really similar data of 1.75 years of rainfall from 7 years of data. The measurement was not on the highest training set, with more hidden units (for example 200) the prediction would have much less error. But the computation time is a very big factor a big amount of data set like this, it can compute for more days with more hidden units and epochs. We can say that we can get better accuracy in a shorter prediction time if we check the error time diagram on figure 17, so the good news is if we would like to predict for 1 month forward from more years of data, BI-LSTM can give a very satisfying response for prediction.

VIII. Discussion

The basic concept is that we forecast the time series of interest y assuming that it has a linear relationship with other time series x. Data showed in this paper, we wish to forecast rainfall in every 5 minutes y using rainfall values for timestamps x as a predictor. The forecast variable y is dependent or explained

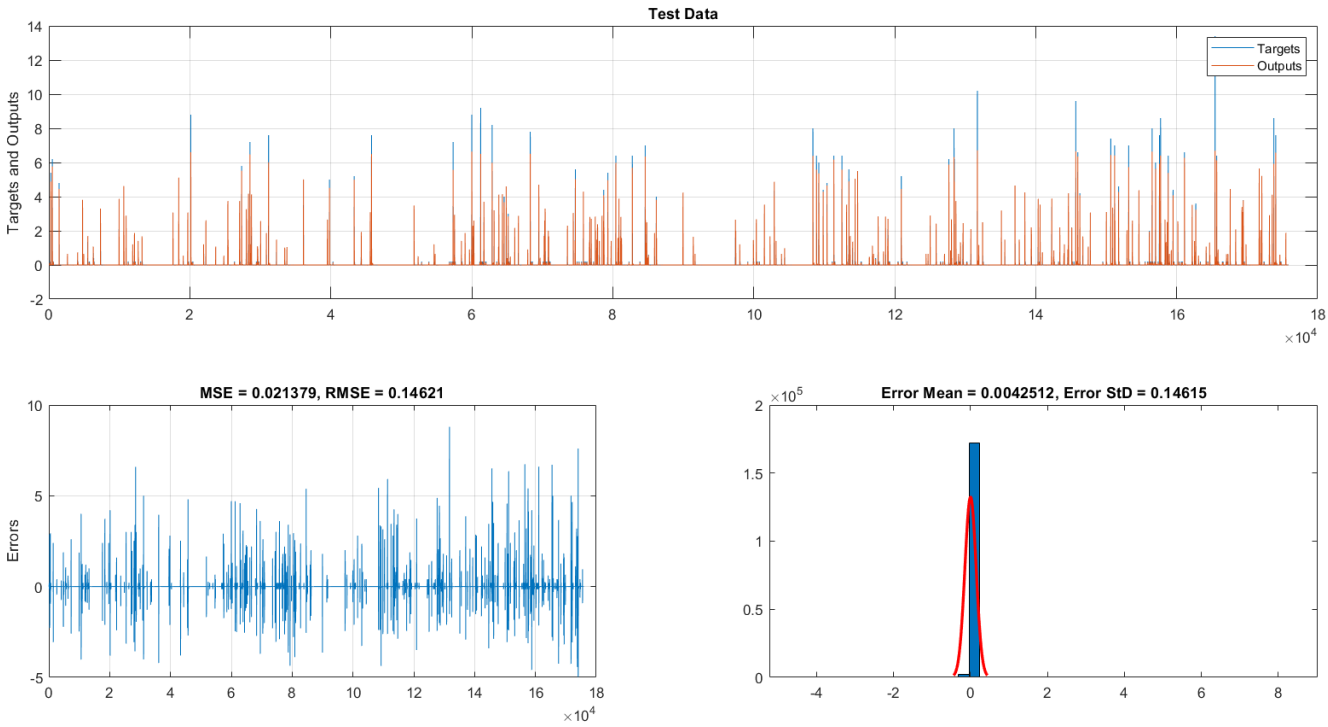

Fig. 17. Error of the BI-LSTM prediction

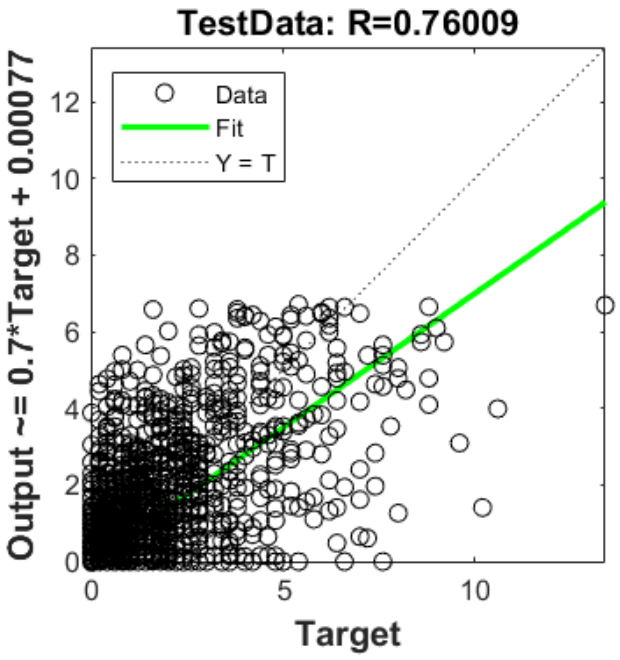

Fig. 18. Regression of the BI-LSTM prediction

variable. The predictor variables x are independent or explanatory variables. [13] In this paper we have discussed some ways of a time-series data prediction, regression, ANFIS, LSTM and BI-LSTM. The great advantage of regression models is that they can be used to capture important relationships between the forecast variable of interest and the predictor variables. A major challenge however, is that in order to generate ex-ante forecasts, the model requires future values of each predictor. If scenario based forecasting is of interest then these models are extremely useful.

As we could see on figure 5, this 7 years long sensitive and chaotic data is not working well with regression. Regression is good only when we would like to get a fast prediction, because it predicts well the 0mm rain, but have bigger errors to predict if is it low amount or high amount of rainfall, according to figure 6.

Next, ANFIS is a bit different from other models, duo to fuzzy sets and learning system. Both artificial neural network and fuzzy logic are used in ANFIS architecture. ANFIS consists of if–then rules and couples of input–output. Also for ANFIS training, learning algorithms of neural network are used. [14]

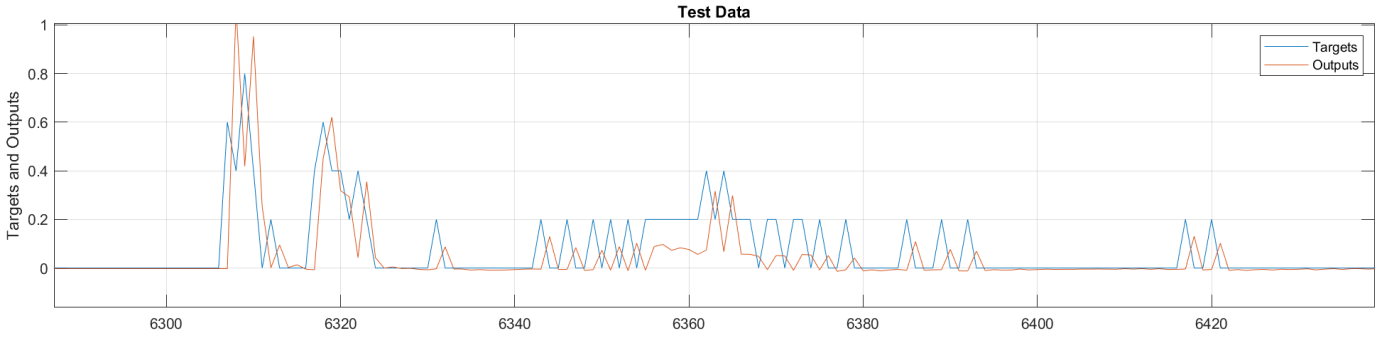

Fig. 19. Zoom in the beginning of BI-LSTM predicted data

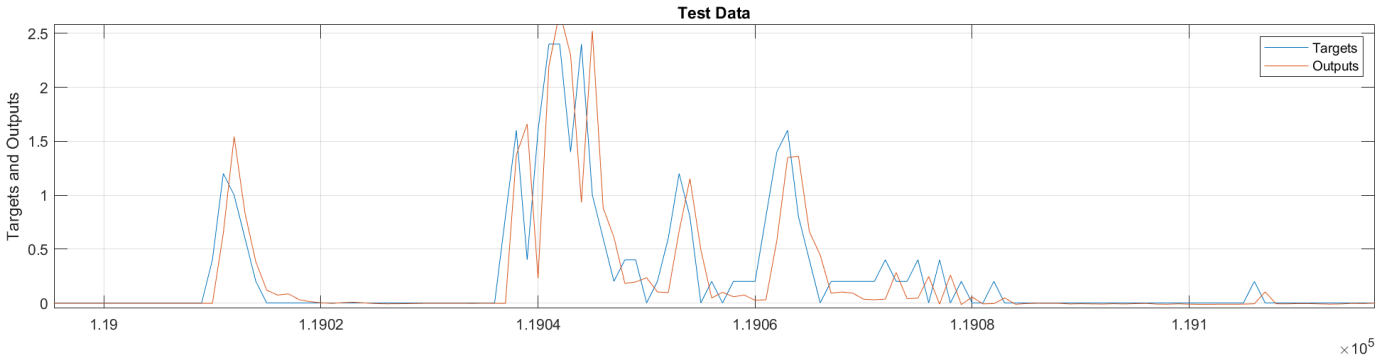

Fig. 20. Zoom in the end of BI-LSTM predicted data

We have seen in the experiment the ANFIS a response which is converting to the original data, but it is not like the original data in shapes and this makes the accuracy extreme low.

For this problem LSTM and BI-LSTM models made the best measurements. To forecast the values of future time steps of a sequence, we could train a sequence-to-sequence regression LSTM network, where the responses are the training sequences with values shifted by one time step. That is, at each time step of the input sequence, the LSTM network learns to predict the value of the next time step. With the BI-LSTM extension the model gave the best RMSE, this can be shown on figure 17. Not just the best error rate, but it had the most similar shape to the original data.

To sum it up, it is very hard to predict a dynamic and chaotic time-series data like rainfall, which has nonlinear and complex behavior. First we have to choose a group of models to try out with a data set and if the accuracy is good enough we can train our system with bigger data sets and testing the system parameters, if manipulating them gives back better accuracy or not. In the experiment of this paper, we found the best for this data set the BI-LSTM architecture. There could be many ways to increase the accuracy of the final prediction, but always there is a limit for deep learning, the computation time.

REFERENCES

[1] J. Gamboa, "Deep learning for time-series analysis," *University of Kaiserslautern, Germany*, Jan. 7, 2017.

[2] (). Map of singapore, Google Maps, [Online]. Available: https://www.google.com/maps/place/Singapore/ (visited on 11/18/2019).

[3] A. J. V. Ben Van Calster, "Calibration of risk prediction models: Impact on decision-analytic performance," *Department of Development and Regeneration, Herestraat 49 Box 7003, 3000 Leuven, Belgium*, Aug. 25, 2014.

[4] J. von Neumann, "Model selection and overfitting," *Nature America*, Sep. 2016.

[5] (Aug. 14, 2015). 7 regression types and techniques, Analytics Vidhya, [Online]. Available: https://www.analyticsvidhya.com/blog/2015/08/comprehensive-guide-regression/ (visited on 11/18/2019).

[6] J. R. QUINLAN, "Induction of decision trees," *Kluwer Academic Publishers, Boston*, Aug. 1, 1985.

[7] M. A. Ahmad Bagheri Hamed Mohammadi Peyhani, "Financial forecasting using anfis networks with quantum-behavedparticle swarm optimization," *University of Guilan, Iran*, 2014.

[8] (2019). Matlab anfis, MathWorks, [Online]. Available: https://www.mathworks.com/help/fuzzy/anfis.html (visited on 11/21/2019).

[9] N. B. Berna Seref, "Neural network and neuro fuzzy model for forecasting equity market data," *Dumlupinar University, Turkey*, 2014.

[10] (Aug. 27, 2015). Understanding lstm networks, Colah's blog, [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/ (visited on 11/23/2019).

[11] J. S. Felix A Gers Nicol N Schraudolph, "Learning precise timing with lstm recurrent networks," *Journal of Machine Learning Research 3*, 2002.

[12] (2019). Long short-term memory networks, MathWorks, [Online]. Available: https://www.mathworks.com/help/deeplearning/ug/long-short-term-memory-networks.html (visited on 11/24/2019).

[13] (2016). Forecasting with regression, Otexts, [Online]. Available: https://otexts.com/fpp2/forecasting-regression.html (visited on 11/24/2019).

[14] X.-b. Jin, "Anfis model for time series prediction," *Beijing Technology and Business University*, Aug. 2013.