Comparison of Time Series Approaches applied to Greenhouse Gas Analysis: ANFIS, RNN, and LSTM

Simone A. Ludwig
North Dakota State University
Fargo, ND, USA
simone.ludwig@ndsu.edu

***Abstract*—Forecasting is the process of predicting the future using past and current data. Uncertainty in real-world data makes this process challenging. The two major forecasting techniques usually applied are causal forecasting and time series forecasting. In causal forecasting the independent variables are used to predict the dependent variable. Time series forecasting on the other hand is a technique used to predict the future values based on historical observations of the same variable and patterns that exist in the data. This paper analyzes time series data of greenhouse gas concentrations at different grid cells in California. The forecasting methods used are Adaptive Neuro-Fuzzy Inference System (ANFIS), Recurrent Neural Network (RNN) and Long Short-term Memory (LSTM) NN. The experimental results reveal that LSTM and ANFIS perform equally well with ANFIS having the shortest execution time.**

I. Introduction

Forecasting is the process of predicting the future using past and current data. The two major forecasting techniques are causal forecasting and time series forecasting. In causal forecasting the independent variables are used to predict the dependent variable. Time series forecasting is a technique used to predict the future values based on historical observations of the same variable and patterns that exist in the data [1].

A time series consists of data points obtained over time with equally spaced intervals. These intervals can be based for example on daily, monthly, quarterly, or yearly data [2]. The main difference between linear regression and time series is that time series data is time dependent whereas a linear regression model assumes that the data is independent of time. This implies that each observation is independent of each other.

Time series data often contain seasonality and trend based on the frequency of the data [3]. There are two ways of analyzing time series. One is referred to as fundamental analysis, and the other is called technical analysis. Fundamental analysis determines the future values based on the underlying factors that affect the data's outcome and its future predictions. Technical analysis on the other hand determines the future values based on the historical values and its behavior over time [4].

The accuracy of the forecasting model is important for the followings reasons. First, forecasts are used to inform both short-term and long-term decision making. Second, forecasts help to deal with the uncertainty in the data. In many industries, most of the financial operations are based on the accuracy of the forecast such that operative decisions in purchasing, marketing and advertising, etc. are possible to make. For instance, less accurate forecasts may lead a company to make wrong decisions and this might lead to a loss in revenue. Thus, the research to develop a good forecasting model and to improve the effectiveness of existing forecasting models has been an active research area [5].

Greenhouse gas (GHG) emissions are said to be difficult to measure directly and thus two different categories of methods to estimate the emission rates were proposed namely the 'Bottom-up' and the 'Top-down' methods. The 'Bottom-up' methods join data on economic activity, fuel consumption, emission factors, and other disparate sources to store them as GHG emissions inventories [6]. The 'top-down' methods estimate the emissions by combining measurements of GHG concentrations in the atmosphere. These measurements are taken from many different stations with information about the atmospheric transport of the gases from their point of origin to the point of the measurement location [7], [8]. Both methods are thought to be important players in verifying GHG emissions policies for the state and on national and international levels [9], [10], [11].

The aim of this paper is to use the GHG emission time series data and apply forecasting methods (ANFIS, RNN, LSTM) that are compared with each other. The remainder of this paper is outlined as follows: Section II describes related work in the area of forecasting, and Section III describes the three approaches applied. The results of the experiments are listed and discussed in Section IV which is followed by the conclusion given in Section V.

II. Related Work

There are several linear statistical and econometric models available to perform time series analysis such as autoregressive (AR) methods, pure moving average (MA) methods, exponential smoothing, and combined AR and MA (ARMA) techniques. The main disadvantage of these linear methods is that they only capture the linear correlation and do not consider nonlinear patterns that might exist in the data [12]. To overcome the limitations of linear models some nonlinear

978-1-5386-1728-1/19/$31.00 ©2019 IEEE

forecasting models such as the bilinear model, the threshold autoregressive (TAR) model, and the autoregressive conditional heteroscedastic (ARCH) models have been introduced. However, the applicability of these models to general forecasting problems is limited [12].

One of the nonlinear models is artificial neural networks (ANN). In was in 1964 when Hu stated his idea to use ANNs for forecasting weather without any learning algorithm [13]. Furthermore, in 1988 Werbos used a learning algorithm and reported that ANNs are better than regression methods and the Box-Jenkin model for prediction problems [13]. The main advantage of ANNs is the nonlinear model that is generated, which is good in capturing nonlinear patterns that exist in the data. In [12], Zhang applied both the ARIMA (linear) model and the ANN (nonlinear) model to forecast time series data in order to improve the forecasting accuracy. The experimental results on a real data set showed that the combined (hybrid) model achieved good accuracy compared to either of the models.

In [14] the performances of different neural networks such as feedforward and recurrent neural networks were compared as well as different training algorithms were used to predict the exchange rates.Before applying the NN model, the authors applied preprocessing techniques to remove the correlation between the data and to normalize the data. Based on the experimental results, recurrent neural networks performed better than feed forward neural network.

III. Comparison Approaches

This section describes the three different methods that are applied to the GHG emission time series data. These are Adaptive Neuro-Fuzzy Inference System (ANFIS), Recurrent Neural Network (RNN) and Long Short-term Memory (LSTM) NN.

A. Adaptive Neuro-Fuzzy Inference System - ANFIS

Adaptive Neuro-Fuzzy Inference System (ANFIS) [15] has been applied to different problem spaces, e.g., control, modeling and parameter estimation [16]. ANFIS comprises of two parts, a neural network (NN) and a fuzzy inference system (FIS) thus taking advantage of both methods.

Essentially, ANFIS combines the learning capability of NNs with the capability of a FIS to model uncertainty in data. Thus, ANFIS creates a NN model of the uncertain problem space using fuzzy logic. An ANFIS model is easily trained without having to rely on precise expert knowledge. The advantage of ANFIS is that it uses both numerical and linguistic knowledge. Furthermore, the NN portion of the model allows to classify data and identify patterns. In comparison to NN, the ANFIS model is more transparent. Thus, ANFIS's advantages include the ability of adaptation, nonlinearity, and rapid learning [17].

More specifically, ANFIS is a hybrid model where the nodes in the different layers of a feed-forward NN use fuzzy parameters, which is the same as a FIS with distributed parameters. ANFIS splits the representation of prior knowledge into subsets in order to reduce the search space, and subsequently uses the backpropagation algorithm to adjust the fuzzy parameters. The resulting system is an adaptive NN functionally equivalent to a first-order Takagi-Sugeno [18] FIS where the input-output relationship is linear. More information on ANFIS can be found in [19].

B. Recurrent Neural Network - RNN

A Recurrent Neural Network (RNN) [20] is a special case of neural network. The aim of an RNN is to predict the next step in a sequence of observations with respect to the previous steps observed in the sequence. Basically, RNN makes use of the sequential observations and learns from the earlier stages in order to forecast/predict future trends. During the earlier stages data need to be remembered when guessing the next steps. In RNN, the hidden layers act as internal storage for storing the information that was gathered during the earlier stages of the processing of the sequential data.

The reason RNNs are called "recurrent" is because they perform the same task for every element of the sequence while utilizing information that was captured earlier in order to predict future unseen sequential data. One of the challenges with RNN is that these networks remember only a few earlier steps within the data sequence and thus are not suitable to remembering longer sequences. To overcome this shortcoming another type of recurrent network namely LSTM was introduced. More information on RNN can be found in [21].

C. Long Short-term Memory - LSTM

Long Short-term Memory (LSTM) NN is a special kind of RNN with the additional ability to memorize the sequence of data. The memorization of the earlier trend and not only a short sequence of the data is available with the help of gates as well as with a memory line that is incorporated in a typical LSTM. In particular, each LSTM contains a set of cells or modules were the data streams are captured and stored. The cells resemble a transport line that connects one module with another module conveying data from the past and gathering them for the present. The different gates in each cell allow data to be disposed, filtered, or added to the following cells. Thus, the gates enable the cells to optionally let data pass through or get disposed of.

The three types of gates involved in each LSTM cell are:

- Forget Gate: outputs a number between 0 and 1, where 1 allows the complete flow through, whereas 0 implies to completely block the stream.
- Memory Gate: chooses which data needs to be stored in the cell.
- Output Gate: the output value is based on the cell state along with the filtered and newly added data.

More information on LSTM can be found in [22].

IV. Experiments

In this section, the description of the data set used is given followed by the parameters of the simulation experiments, and the results that were obtained.

A. Data Description

The data set used for this research investigation is the time series data of greenhouse gas (GHG) concentrations at 2,921 grid cells in California [23]. The data set was created using simulations of the Weather Research and Forecast model with Chemistry (WRF-Chem). There is one data file per grid cell, which contains 16 time series of GHG concentration, and each grid cell covers an area of 12 km by 12 km. The data was recorded over the period May 10 - July 31, 2010 and the data points in the time series are spaced 6 hours apart (4 samples per day).

For the experiments, data of 4 different locations is used. The four GHG data sets were preprocessed using differencing in order to make the data stationary. Furthermore, normalization to transform the data between zero and one was done. To give an idea about the data distribution, data file labeled site1340 has an average value of 2.74 with minimum and maximum values of 1.00E-04 and 7.87E+01, respectively.

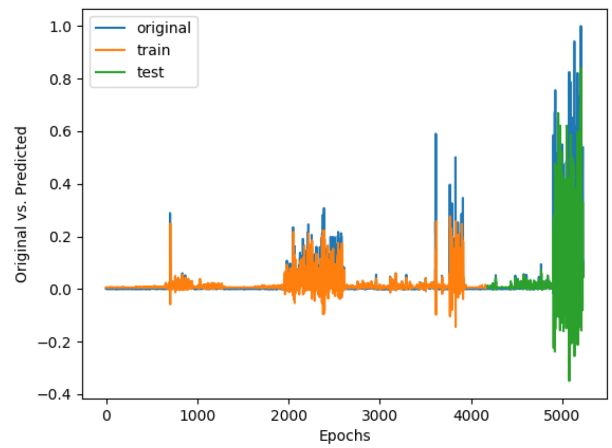


Fig. 1. Time series data

B. Results of Experiments

Two set of experiments were done. The first set of experiments fine-tuned the ANFIS model and the second conducted experiments using all three evaluation methods. Tensorflow and Keras were used to implement the three different models.

1) Experiment 1: Different parameters of the ANFIS model were experimented with and the results of the best model is reported below. The different parameters that were experimented with were number of regressors, number of rules, learning rate, number of epochs, loss function, and learning algorithm.

The parameters of the best performing ANFIS model were the following:

- Number of regressors = 4
- Number of rules = 12
- Learning rate = 0.002
- Number of epochs = 200
- Loss function = Mean squared error
- Learning algorithm = Adam

Figure 1 shows the time series data in terms of the original data, and the train data portion and test data portion. The training loss and the validation loss of one run is given in Figure 2 and Figure 3, respectively.

The generated membership functions according to the 12 different rules are shown for completeness in Figure 4.

2) Experiment 2: The RNN and LSTM models were kept similar with identical parameters. The difference between RNN and LSTM is that the corresponding nodes are used in the hidden layer (RNN OR LSTM cell). The identical parameters include:

- Input layer = 4
- Optimization algorithm = Adam
- Loss function = Mean squared error
- Maximum number of iterations = 20

As for the ANFIS model, the same parameters given as for Experiment 1 are used. The evaluation measures used are

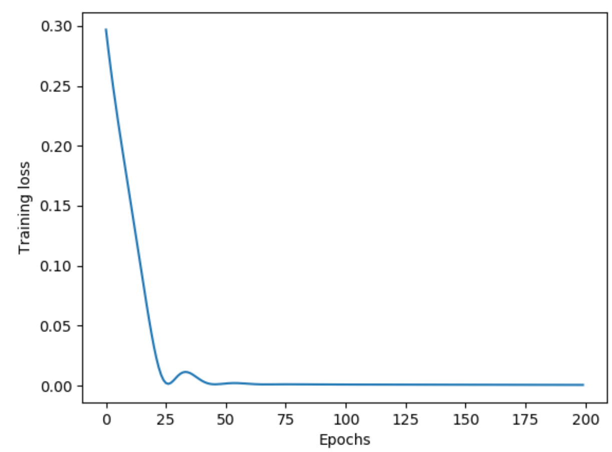


Fig. 2. Training loss - ANFIS

Root Mean Squared Error (RMSE) and Mean Absolute Error (MAE).

Table I shows the result of the comparison of RNN, LSTM, and ANFIS on the train and test portion. Results are reported in terms of RMSE and MAE. As can be seen ANFIS scores best for Location 2. For Location 4, LSTM and ANFIS score equally well and for location 6 LSTM outperforms the other two methods. As for Location 8, LSTM again slightly outperforms ANFIS. Overall considering all locations, ANFIS outperforms LSTM and RNN in terms of RMSE, and LSTM outperforms ANFIS and RNN in terms of MAE.

Table II shows the execution time in seconds for RNN, LSTM and ANFIS. As can be seen the ANFIS method is by far the fastest method with 6.87 seconds. RNN takes 73.74 seconds to perform, and LSTM is the computationally most expensive method with a running time of 150.39 seconds.

Figure 5 show the train and test results of the time series analysis in comparison to the original time series for different

TABLE I
RMSE AND MAE FOR RNN, LSTM, AND ANFIS ON TRAIN AND TEST PORTION OF TIME SERIES

Location	Train/Test	RNN		LSTM		ANFIS	
		RMSE	MAE	RMSE	MAE	RMSE	MAE
2	Train	0.034434	**0.010440**	0.033947	0.012294	**0.030264**	0.010726
	Test	0.114815	0.048943	0.111844	0.048530	**0.103310**	**0.045879**
4	Train	0.026004	0.011753	0.025513	0.009933	**0.023573**	**0.008442**
	Test	0.124423	0.054158	**0.120220**	**0.050290**	0.124444	0.051655
6	Train	0.024437	0.014198	**0.023343**	**0.007892**	0.023545	0.008325
	Test	0.101140	0.048277	**0.100480**	**0.043279**	0.103294	0.045501
8	Train	0.033131	0.012284	0.033050	**0.011106**	**0.031690**	0.011954
	Test	**0.123935**	0.053905	0.124932	**0.053156**	0.124221	0.054795

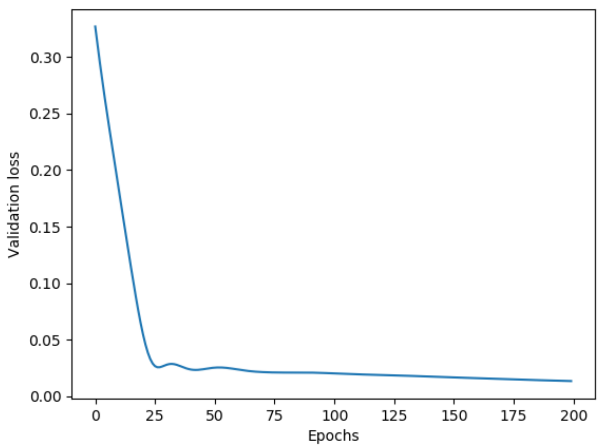


Fig. 3. Validation loss - ANFIS

TABLE II
EXECUTION TIME OF RNN, LSTM, AND ANFIS

Method	Time (seconds)
RNN	73.740552
LSTM	150.393666
ANFIS	6.878668

locations of the GHC sensors.

V. CONCLUSION

This paper investigated different time series forecasting techniques such as ANFIS, RNN and LSTM. The different forecasting techniques are used to predict the future values based on historical observations analyzing time series data of greenhouse gas (GHG) concentrations at different grid cells in California.

Two different set of experiments were conducted. The first one experimented with the ANFIS model in order to find the best configuration for the GHG data. The second set of experiments ran all three models (ANFIS, RNN and LSTM) and evaluated the outcomes based on RMSE and MAE. Overall it can be concluded that LSTM and ANFIS perform equally well with regards to both error measures with RNN not scoring very well. Comparing the execution times revealed that the ANFIS model quite dramatically outperformed the RNN and LSTM models. Thus, we can summarize that the ANFIS model is the technique to use when both accuracy and execution time are concerned.

REFERENCES

[1] N. Pant. A Guide For Time Series Prediction Using Recurrent Neural Networks (LSTMs). Stats and Bots, Stats and Bots, 2017.
[2] K. Chakraborty et al. Forecasting the behavior of multivariate time series using neural networks. Neural networks 5.6 (1992): 961-970.
[3] A. Jain. Complete guide to create a Time Series Forecast (with Codes in Python). Analytics Community, Analytics Discussions, Big Data Discussion, 2016.
[4] S. Y. Xu, and C. U. Berkely. Stock price forecasting using information from Yahoo finance and Google trend. UC Brekley, 2014.
[5] J. Hiray. Time-series methods of forecasting, All about Business and management. All about Business and management, A blog devoted to business management and processes, 2008.
[6] EDGAR: Emission Database for Global Atmospheric Research (EDGAR), release version 4.1, European Commission, Joint Research Centre (JRC)/Netherlands Environmental Assessment Agency (PBL), available at: http://edgar.jrc.ec.europa.eu (last access: 20 December 2014), 2009.
[7] R. F. Weiss, R. G. Prinn. Quantifying greenhouse-gas emissions from atmospheric measurements: a critical reality check for climate legislation, Philos. T. Roy. Soc. A, 369, 1925-1942, 2011.
[8] E. Nisbet, R. Weiss. Top-down versus bottom-up, Science, 328, 1241-1243, 2010.
[9] P. Ciais, A. J. Dolman, R. Dargaville, L. Barrie, A. Bombelli, J. Butler, P. Canadell, T. Moriyama. GEO Carbon Strategy, Tech. rep., Geo Secretariat Geneva, FAO, available at: http://www.earthobservations.org/documents/sbas/cl/201006_geo_carbon_strategy_report.pdf (last access: 20 December 2014), 2010.
[10] K. K. Jonietz, P. E. Dimotakis, D. A. Rotman, B. C. Walker. A Greenhouse-Gas Information System: Monitoring and Validating Emissions Reporting and Mitigation, Tech. Rep. LA-UR-11-5512, Report for the US Department of Energy, Office of Science, prepared by Los Alamos National Laboratory, Los Alamos, New Mexico, USA, doi:10.2172/1033495, 2011.
[11] M. L. Fischer, S. Jeong. Inverse Modeling to Verify California's Greenhouse Gas Emission Inventory, Tech. rep. ver. 1.3, Report for California Air Resources Board and the California Environmental Protection Agency, available at: http://www.arb.ca.gov/research/rsc/10-18-13/item5dfr09-348.pdf (last access: 20 December 2014), 2013.
[12] G. P. Zhang. Time series forecasting using a hybrid ARIMA and neural network model. Neurocomputing 50:159-175, 2003.
[13] M. P. Naeini, H. Taremian, and H. B. Hashemi. Stock market value prediction using neural networks. Computer Information Systems and Industrial Management Applications (CISIM), 2010 International Conference on. IEEE, 2010.
[14] B. Oancea, and S. C. Ciucu. Time series forecasting using neural networks. arXiv preprint arXiv:1401.1333, 2014.
[15] J.-S.R. Jang, Fuzzy Modeling Using Generalized Neural Networks and Kalman Filter Algorithm. In Proceedings of the AAAI'91 Proceedings of the Ninth National Conference on Artificial Intelligence, Anaheim, CA, USA, 14-19 July 1991.
[16] S. Amid, T. M. Gundoshmian. Prediction of output energies for broiler production using linear regression, ANN (MLP, RBF), and ANFIS models. Environ. Prog. Sustain. Energy 2017, 36, 577-585.

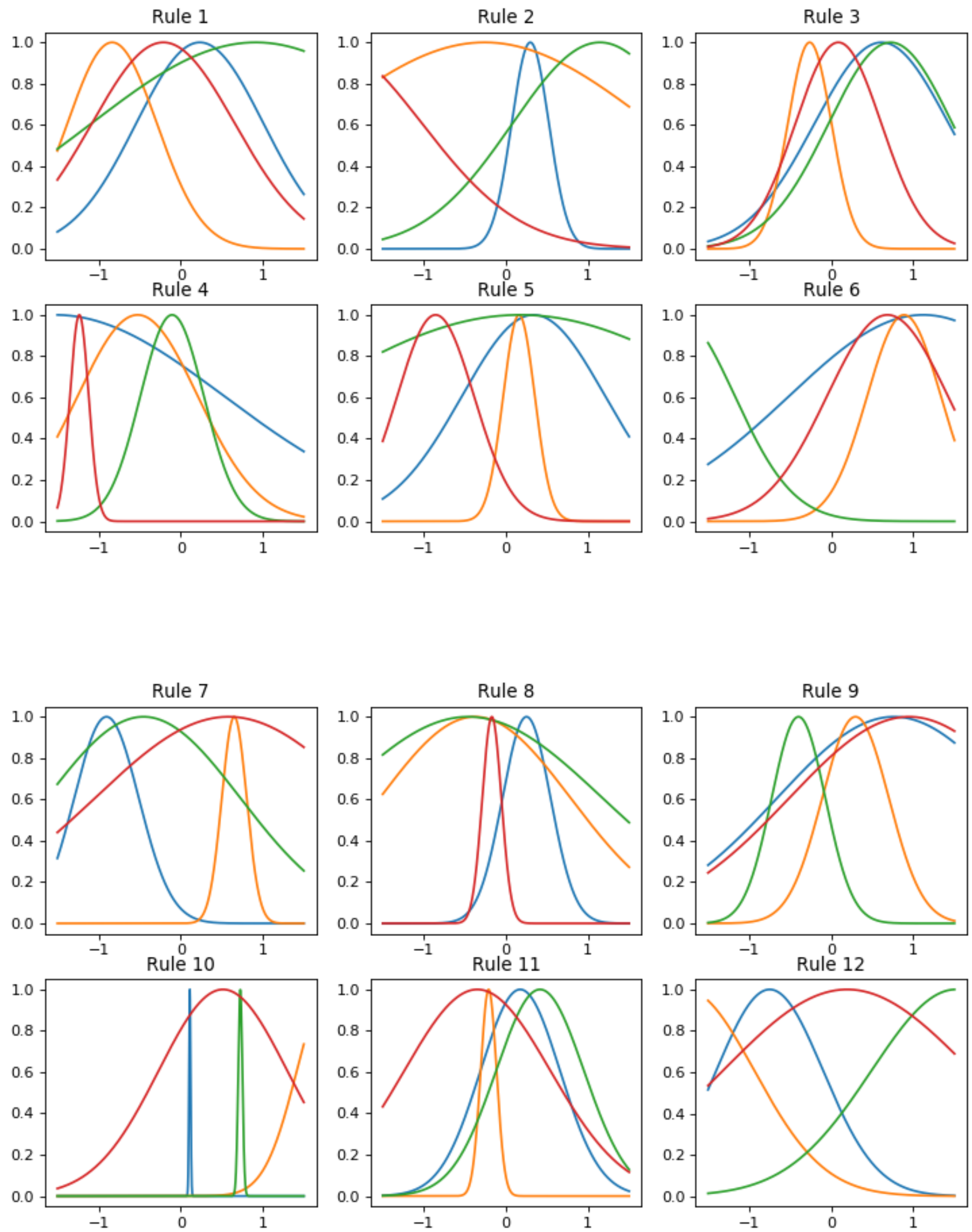


Fig. 4. Membership functions of 12 rules - ANFIS

[17] P. Srisaeng, G. S. Baxter, G. Wild. An adaptive neuro-fuzzy inference system for forecasting Australia's domestic low cost carrier passenger demand. Aviation 2015, 19, 150-163.

[18] M. Sugeno. Industrial applications of fuzzy control, Elsevier Science Pub. Co., 1985.

[19] J.-S.R. Jang. ANFIS: adaptive network-based fuzzy inference systems. IEEE Trans Sys Man Cybern 23:665685, 1993.

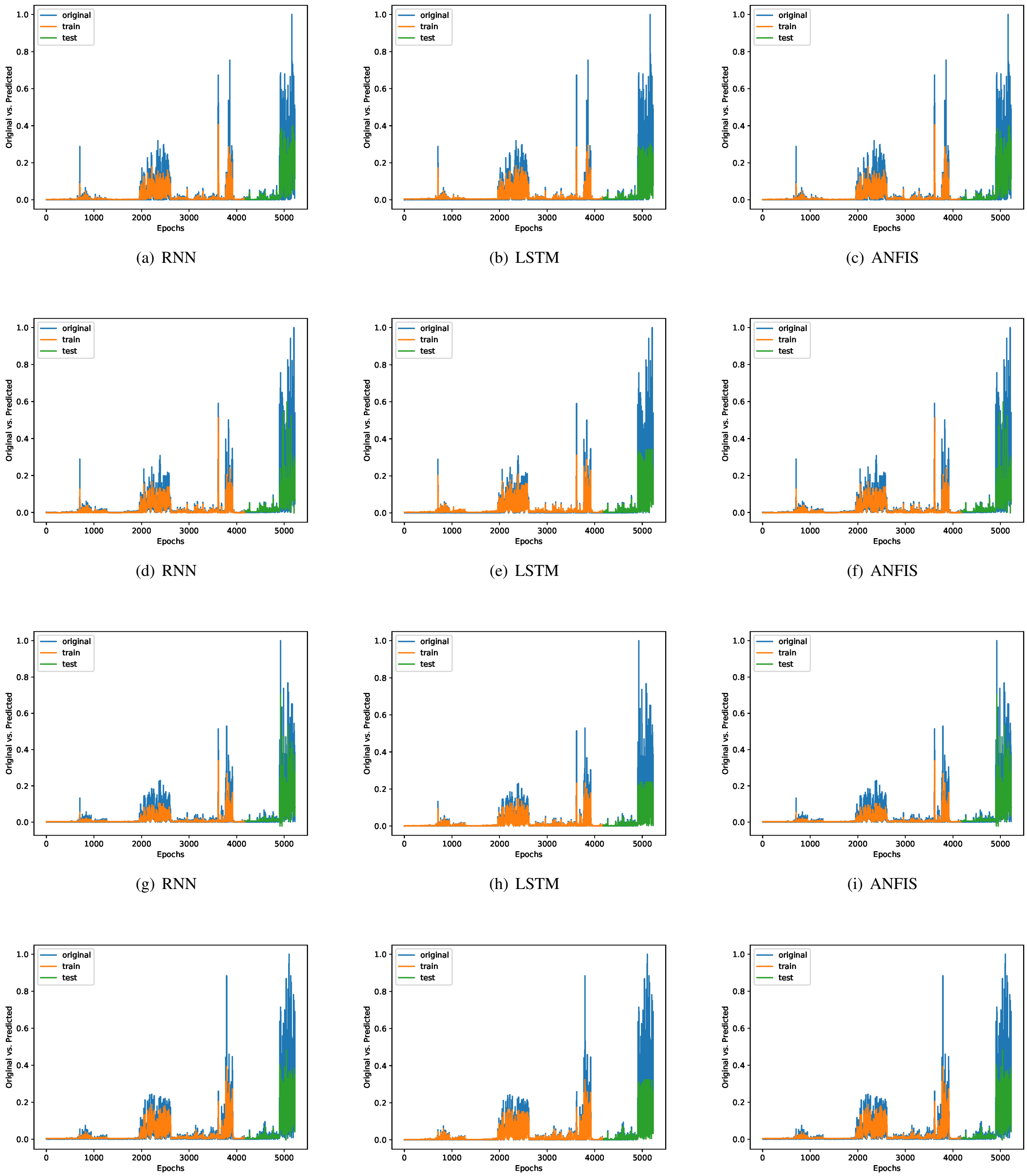
[20] M. I. Jordan. 1990. Attractor dynamics and parallelism in a connectionist

[21] J. Schmidhuber. Habilitation thesis: System modeling and optimization, 1993.

sequential machine. In Artificial neural networks, Joachim Diederich (Ed.). IEEE Press, Piscataway, NJ, USA 112-127.

[22] S. Hochreiter, J. Schmidhuber. Long Short-Term Memory. Neural Computation 9(8):1735-1780, 1997.

[23] D. D. Lucas, C. Yver Kwok, P. Cameron-Smith, H. Graven, D. Bergmann, T. P. Guilderson, R. Weiss, R. Keeling. Designing optimal greenhouse gas observing networks that consider performance and cost, Geoscientific Instrumentation Methods and Data Systems, 2015.



(a) RNN (b) LSTM (c) ANFIS

(d) RNN (e) LSTM (f) ANFIS

(g) RNN (h) LSTM (i) ANFIS

(j) RNN (k) LSTM (l) ANFIS

Fig. 5. Time series data showing original, train and test outputs from RNN, LSTM, and ANFIS; each row shows one location starting with Location 2 (first row) until Location 8 (last row)