SPATFIS - Supplementary Material

S. Samanta, M. Pratama, S. Sundaram,

I. Introduction

The supplementary material for the SPATFIS article is provided here. This document will contain the followings,

- Pseudo code for SPATFIS algorithm
- Added experiment (Sun spot problem) for the purpose of statistical analysis
- Actual vs predicted plots for the conducted experiments
- Rank based statistical analysis
- Prequential Test-Then-Train protocol for datastream modeling
- Variability Analysis
- Final fuzzy rule base after training completion

II. Materials

A. SPATFIS Pseudo code

Pseudo Code of SPATFIS allows future reader to grasp the concept clearly as this provides a snap shot of the learning process. **Input:** The training dataset $[\mathbf{u}(k), y(k)]$ to be learned one by one in accordance with the principle of sequential learning. **Output:** Optimal number of fuzzy rules(R), their membership parameters (μ_{ri}, σ_r) and weights (**w***) are to be obtained/tuned.

Start with zero rule in the rule base, Memory neuron parameters are initialized with random values in [0,1];
Set all the memory outputs to 0 wherever necessary (during initialization and rule growing) ;
Set the first rule (center and spread) from first sample ;
while *samples are learned (for $k = 1, 2, \cdots, k$)* **do**
  if $\frac{1}{R}\Sigma_{r=1}^{R}\left(\frac{1}{\hat{P}}\Sigma_{i=1}^{\hat{P}}\phi_{ri}(k)\right) < 0.110$ **then**
    Add a New Rule and Initialize the corresponding Gaussian parameters and consequent weights (Eqn. 13 and 16);
  else
    Update all the weights (Eqn. 22 or 25);
    Update only the winning rule parameters i.e. centers and spreads (Eqn. 35-36);
    Update all the memory outputs (Eqn. 6 and 11);
  end
  Prune redundant rule if any.
end
Perform Testing ;

Algorithm 1: SPATFIS Learning Mechanism

Please note the equations here refer to the equations in main paper.

B. Sunspot Problem

In the main paper 4 experiments are demonstrated (due to space constraints) but to ensure a statistical significance at least 5 experiments are required hence the popular sunspot prediction problem is employed and the results are provided here.

Black blotches on the surface of the sun are called sunspots. They were first found in the 1700s after discovery of the telescope. Sunspots are the reason behind numerous solar exercises (i.e. change in solar magnetism and so on) yet the most possible causalities behind this marvel are yet unknown. For this reason it has been utilized as a mainstream benchmark time series problem for decades [1] by researchers. The monthly American sunspot data has 778 samples for the range of 1944 December to 2009 October.[1] The first 699 samples were fed one by one to learn from while the remaining 79 samples are employed for testing. The target was to forecast one month ahead (single step ahead) sunspot number using past two month's values. Different performance matrices are shown in table I.

From table I it is apparent that, for the aforementioned problems SPATFIS is able to attain better accuracy while taking much shorter (or similar) training time and using a smaller fuzzy rule base.

S. Samanta is a PhD student with ERI@N, IGS, Nanyang Technological University, Singapore. e-mail: sama0021@e.ntu.edu.sg
M. Pratama is an Assistant Prof. with SCSE, Nanyang Technological University, e-mail: mpratama@ntu.edu.sg
S. Suresh is a Prof. with Department of Aerospace Engineering, IISC Bengaluru, India, e-mail: sureshsundaram@iisc.ac.in

[1] https://www.ngdc.noaa.gov/stp/space-weather/solar-data/solar-indices/sunspot-numbers/american/lists

Table I: Performance comparison on benchmark problems

Problem	Network Model	#Rules	Test RMSE	Time(s)	Rank
Sun-spot	eTS [2]	23	0.065	3.5	**5**
	SimpleTS [3]	20	0.085	3.2	**4**
	SAFIS [4]	21	0.110	4.4	**7**
	McFIS [5]	12	0.100	4.2	**6**
	PANFIS [6]	50	0.090	1.8	**3**
	GENFIS [7]	06	0.070	0.9	**1**
	SPATFIS	**07**	**0.050**	**1.1**	**2**

C. *Actual vs Predicted plots*

In this section we have provided the actual vs estimated plots for test datasets for each of the problems explained in the main manuscript to demonstrate superior prediction capability of SPATFIS. In MG-85 and BJ problem, SPATFIS deals with chaotic time series with severe non linearity and sharp dynamics change whereas the real world datasets are not only dynamic and non linear in nature they are also imprecise, noisy and contain uncertainty.

- **MG-85 Chaotic Time Series Prediction:**

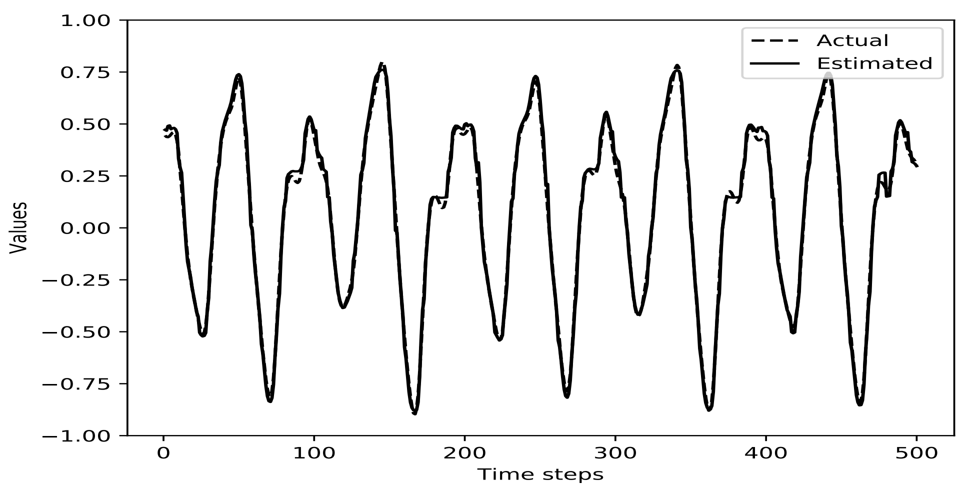

Figure 1: mg-85 prediction using SPATFIS

- **Box Jenkins CO_2 Emission Prediction:**

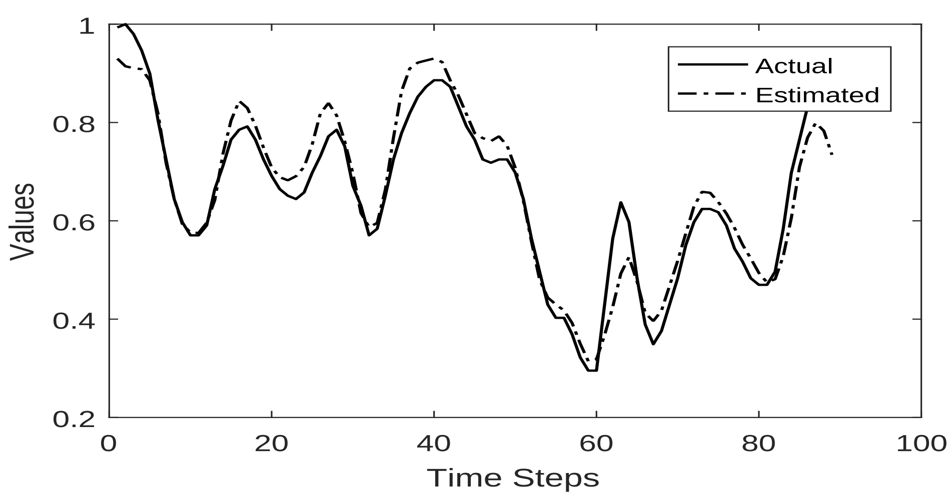

Figure 2: BJ prediction using SPATFIS

- **Wind Speed Prediction:**

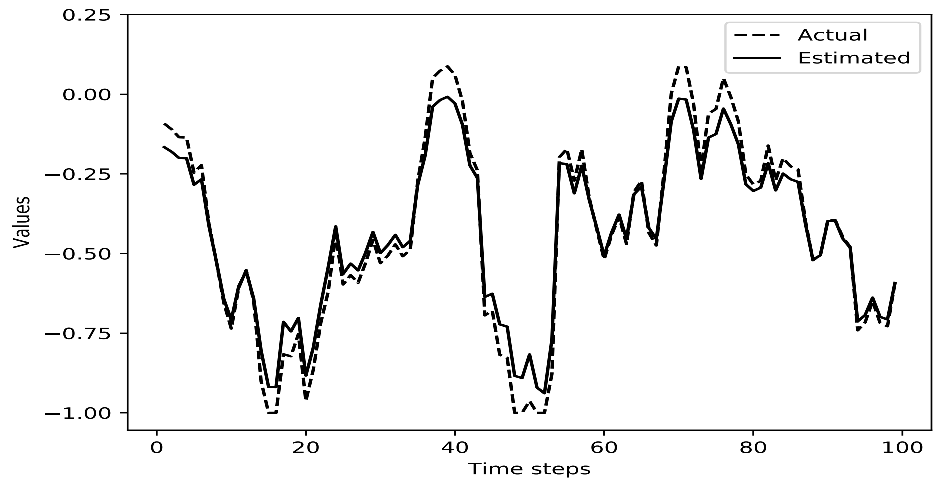

Figure 3: Wind speed prediction using SPATFIS

- **Stock Price Prediction:**

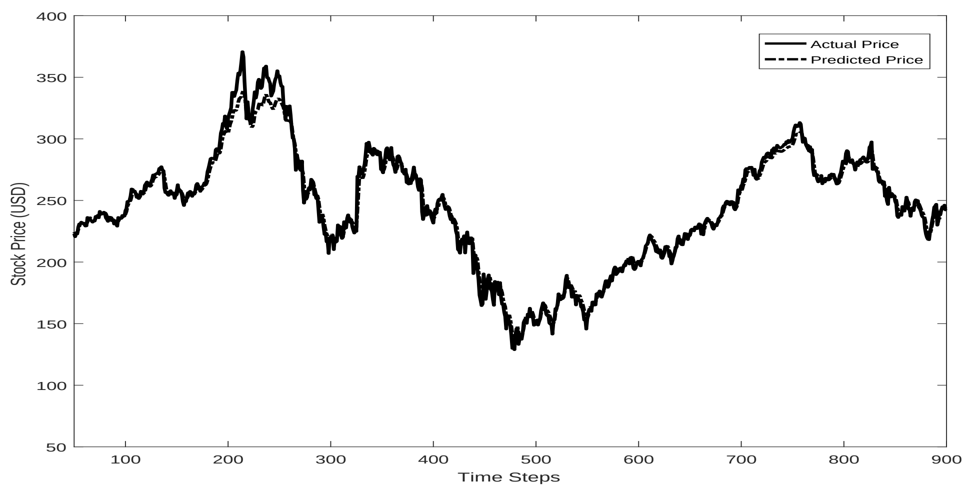

Figure 4: Stock price prediction using SPATFIS

D. Rank based statistical Analysis

Apart from SPATFIS six other state-of-the-art neural fuzzy methods are rebuilt (i.e. eTS [2], SimpleTS [3], SAFIS [4], McFIS [5], PANFIS [6] and GENEFIS [7]) for the sake of fair comparison in MATLAB 2016. The first rank analysis is provided here with detailed procedure for the execution speed of SPATFIS. Following the same pattern rank analysis is performed on the other metrices.

1) ***Rank Analysis on Execution Time****:* In the following table the training time based rank assignment is provided (along with the average rank and average rank difference with SPATFIS).

The non-parametric Friedman's test [8] compares the average rank of algorithms, $\bar{r}_i = \frac{1}{p}\sum_{j=1}^{p} r_{ij}$ under the null hypothesis that all algorithms are equivalent with same average rank. The Friedman statistic,

$$\chi_F^2 = \frac{12p}{q(q+1)}\left[\sum_{i=1}^{q} \bar{r}_i^2 - \frac{q(q+1)^2}{4}\right] \tag{1}$$

Table II: Performance comparison on benchmark problems

Problems/Methods	MG	BJ	Sunspot	Wind	Stock	Rank Average	Average Rank Differance
eTS	2	5	5	4	3	3.8	2.6
SimpleTS	3	6	4	3	7	4.6	3.4
SAFIS	4	7	7	5	5	5.6	4.4
McFIS	6	2	6	6	4	4.8	3.6
PANFIS	5	3	3	7	6	4.8	3.6
GENEFIS	7	4	1	2	2	3.2	2
SPATFIS	1	1	2	1	1	1.2	0

follows the χ_F^2 distribution with $q-1$ degrees of freedom, where $p=5$ is the number of problems and $q=7$ is the number of competing algorithms in the test. For smaller number of problem sets, a better statistic can be derived which is less conservative,

$$F_F = \frac{(p-1)\chi_F^2}{p(q-1)-\chi_F^2} \tag{2}$$

which follows F distribution with $(q-1)=6$ and $(p-1)(q-1)=5*6=30$ degrees of freedom. For this study the F statistics value can be found from Eqn. 1 and 2 to be 3.329 whereas the critical F value at 95% confidence interval is obtained from the F table, $F_{6,30}=2.42$. As the experimental F value is greater than the critical value the null hypothesis can be rejected, hence the algorithms are significantly different in terms of their execution time.

Once the null hypothesis is rejected pairwise pos-hoc Bonferroni-Dunn test is conducted to demonstrate SPATFIS's superior performance over the other methods. The test statistics for comparing i^{th} and j^{th} algorithms is,

$$z_{ij} = \frac{\bar{r}_i - \bar{r}_j}{\sqrt{q(q+1)/6p}} \tag{3}$$

From table II one can notice that the average rank difference with SPATFIS is minimum for GENEFIS, hence the first pairwise test is executed for them and the z value is found to be 1.464. From the z table, one can obtain the corresponding probability to be 0.927 hence SPATFIS is better than GENEFIS at 90% confidence interval. The same test is performed for all the other pairs i.e. for eTS (2nd best ranking after GENEFIS) and SPATFIS, the z value is found to be 1.90 and the corresponding probability is 0.971. Hence it can be concluded that performance of SPATFIS is better than all the other algorithms at a 95% confidence interval.

*2) **Rank Analysis on Predictive Accuracy**:* In a similar manner the ranks can be assigned for the algorithms on the basis of their predictive accuracy. Friedman's test statistic F value is found to be 6.938 whereas the F-value obtained from F-Table $F_{6,30}$ = 2.42 at 95% confidence interval hence null hypothesis can be rejected and it can be safely concluded that the participating algorithms are significantly different.

Pairwise pos-hoc Bonferroni-Dunn test is executed next and it is found that SPATFIS is better than all the other algorithms (except for GENEFIS) at 95% confidence interval. However it performed better than GENEFIS but at a slightly lower 85% confidence interval (with z value 1.1).

*3) **Rank Analysis on Rule base size**:* The last batch of testing is performed on basis of rule base size. Ranks are assigned accordingly and Friedman's test statistic (F value) is calculated to be 7.764 and the F-value from F-Table($F_{6,30}$) = 2.42 at 95% confidence interval so here also the null hypothesis is rejected and it is concluded that the competing eNFS are significantly different.

After rejecting the null hypothesis pairwise pos-hoc Bonferroni-Dunn test is performed in a similar manner like before and similar result is obtained. It is found that SPATFIS is better than all the other algorithms (apart for GENEFIS) at 95% confidence interval and better than GENEFIS but at a little lower 85% confidence interval (with z value 1.07).

E. Prequential Test-Then-Train validation scheme for datastream modeling

In real world situations the incoming data streams are often endless. To model such a datastream the regular cross-validation method is not suitable because it requires multiple runs, it is slower, it is highly memory and computationally expensive also it loses the inherent temporal ordering of the datastream. On the other hand in a prequential learning with First-Test-Then-Train protocol each of the data chunks are first tested and then the same is trained without disrupting the temporal ordering. The main goal of CV is to train the model with as many datasets as possible and test with unseen data. In prequential scenario as the testing takes place before training, there's no dearth of unseen data. Hence modelling and validating a datstream with prequential learning protocol is much more suitable compared to a CV or sequential learning.

To conduct the prequential experiments with available real world time series data we divide the dataset into equal sized batches to mimic the stream of data chunks. The chunk size is kept around 5% of the total sample size. According to the principle of prequential learning, these chunks are first tested and then the same chunk is used to train the model in one shot, one by one.

Prequential learning predominantly provides a pessimistic estimate of prediction accuracy than a sequential one, as testing is always performed before training. The initial higher error drags the overall accuracy down by some extent.

Upon in depth investigation with different batch sizes, it was found that a batch size of 5% of the total sample size, produces best result in terms of prequential accuracy. It was also observed that **SPATFIS prequential test RMSE** (averaged over all chunks) is close to its sequential accuracy. This concludes that proposed SPATFIS model is well suited even for real world stream scenarios. For wind and stock problems SPATFIS attained preqential test RMSE of 0.16 and 0.03 respectively. Table III provides a comparison of the SPATFIS sequential vs prequetial test accuracy for the real world problems.

Table III: Sequential vs Prequential testing compariso

Problem	Sequential test RMSE	Prequential test RMSE
Wind Speed Prediction	0.143	**0.160**
Stock Price Prediction	0.021	**0.030**

F. Variability Analysis:

In all of the experiments conducted in this article the weights are updated with one shot PBL method to make SPATFIS fast. As the weight update method is kept one shot and not gradient based (incremental PBL) the variability of the results in terms of accuracy was found to be very low. This observation suggests that SPATFIS is robust to random initialization. This can also be attributed to the updating of the winning rule parameters (centers and spreads) throughout the training. Because of the low variability as shown below we have took the liberty of producing the best results and doing the rank based statistical analysis on them in the main manuscript. In the following table we provide the variability of accuracy in terms of their means and standard deviations over 20 independent trials.

Table IV: Variability Analysis on test Accuracy

Problem	Mean Accuracy	Standard Deviation	SD as percentage of Mean accuracy
Mackey Glass Problem	0.115	0.004	**3.4%**
Box Jenkins Prediction	0.047	0.003	**6.3%**
Wind Speed Problem	0.151	0.011	**7.1%**
Stock Price Prediction	0.025	0.001	**4.2%**

G. Plot of Fuzzy Rule Base

In this subsection we plot the final fuzzy rule base to demonstrate how does that look. We again take the example of the Mackey Glass problem, In the main manuscript we have shown the rule evolution process where we have seen that the final number of fuzzy rule gets stable at 6 once the training is done. We then plot these Gaussian fuzzy rules. In the figure below one can observe that across the feature axis we have the centers and across the y axis the corresponding membership values for each of these rules can be found. We demonstrate each rules with different color as below,

III. Conclusion

Pseudo Code of SPATFIS allows future reader to grasp the concept clearly and code easily. Extra experiment is provided to ensure significant testing. Actual vs Predicted plots for the real world problems demonstrates the superior predictive capability of SPATFIS. Finally the rak based statistical analysis proves that SPATFIS is singnificantly better than the competing state-of-the-art evolving neuro-fuzzy systems, in terms of both execution time, accuracy and size. The suitability of SPATFIS in terms of prequential learning scenario is also demonstrated.

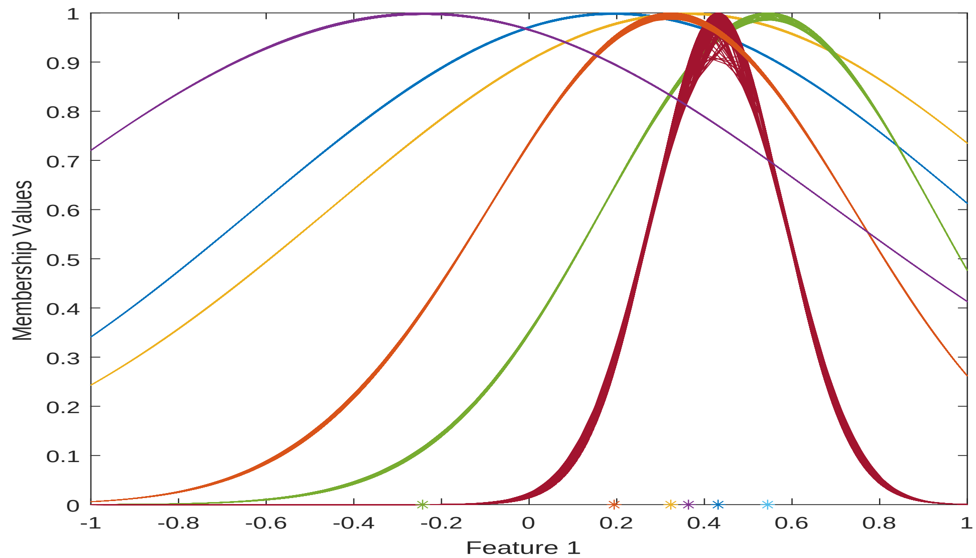

Figure 5: Fuzzy rule base after SPATFIS rule evolution for MG-85 problem

References

[1] G. E. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[2] P. P. Angelov and D. P. Filev, "An approach to online identification of takagi-sugeno fuzzy models," *IEEE Trans. on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 1, pp. 484–498, Feb 2004.

[3] P. Angelov and D. Filev, "Simpl_ets: a simplified method for learning evolving takagi-sugeno fuzzy models," in *Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on*. IEEE, 2005, pp. 1068–1073.

[4] H.-J. Rong, N. Sundararajan, G.-B. Huang, and P. Saratchandran, "Sequential adaptive fuzzy inference system (safis) for nonlinear system identification and prediction," *Fuzzy sets and systems*, vol. 157, no. 9, pp. 1260–1275, 2006.

[5] K. Subramanian and S. Suresh, "A meta-cognitive sequential learning algorithm for neuro-fuzzy inference system," *Applied soft computing*, vol. 12, no. 11, pp. 3603–3614, 2012.

[6] M. Pratama, S. G. Anavatti, P. P. Angelov, and E. Lughofer, "Panfis: A novel incremental learning machine," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 25, no. 1, pp. 55–68, 2014.

[7] M. Pratama, S. G. Anavatti, and E. Lughofer, "Genefis: Toward an effective localist network." *IEEE Trans. Fuzzy Systems*, vol. 22, no. 3, pp. 547–562, 2014.

[8] J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *Journal of Machine learning research*, vol. 7, no. Jan, pp. 1–30, 2006.