Predicting Chaotic Time Series with Fuzzy If-Then Rules

Jyh-Shing Roger Jang and Chuen-Tsai Sun
Department of Electrical Engineering and Computer Science
University of California, Berkeley, CA 94720

Abstract— **This paper presents the continued work of a previously proposed ANFIS (Adaptive-Network-based Fuzzy Inference System) architecture [2, 1, 3], with emphasis on the applications to time series prediction. We explain how to model the Mackey-Glass chaotic time series with 16 fuzzy if-then rules. The performance we obtained outperforms various standard statistical approaches and artificial neural network modeling reported in the literature. Other potential applications of ANFIS are also suggested.**

I. INTRODUCTION

A fuzzy inference system with adaptive capability is drawing more and more attention since it can not only incorporate human expertise in the form of fuzzy if-then rules but also fine-tune the membership functions according to a desired input-output data set. One architecture of the adaptive fuzzy inference systems proposed by the author [2, 1] employs the back-propagation-type gradient descent [16, 12] and the least squares estmate to achieve the capability of learning by examples. A detailed treatment of the ANFIS (Adaptive-Network-based Fuzzy Inference System) architecture can be found in [3].

This paper reports further simulation results where a 16-rule ANFIS is used to predict the future values of the Mackey-Glass chaotic time series [6]. We choose this time series for the simulation simply because it is a benchmark problem that have been cited quite often in the literature, which allows us to compare the results obtained from other approaches such as linear regression and neural networks.

In the next section, the basics of ANFIS is introduced. Section 3 presents the simulation settings and simulation results, along with comparisons of generalization tests with other methods. Section 4 gives a concluding remarks and suggests other applications of ANFIS.

II. BASICS OF ANFIS

This section introduces the basic architecture and the hybrid learning rule of ANFIS. For a detailed coverage, see [1, 3].

*Research supported in part by NASA Grant NCC-2-275; LLNL Grant No. ISCR 89-12; MICRO State Program Award No. 90-191; and OMRON Corporation.

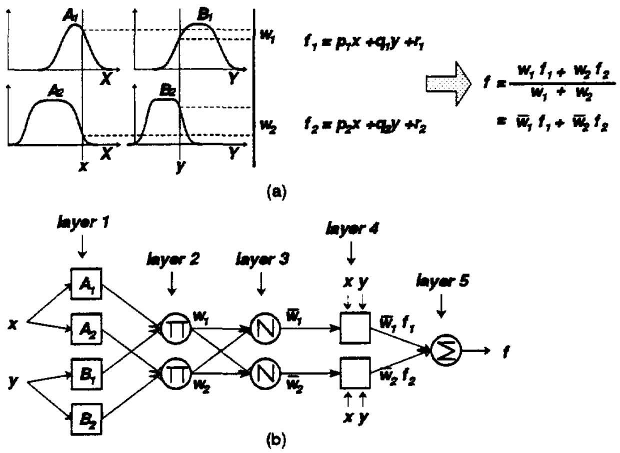


Figure 1: *(a) fuzzy reasoning; (b) equivalent ANFIS.*

Suppose that the fuzzy inference system contains two rules of Takagi and Sugeno's type [14]:

Rule 1: If x is A_1 and y is B_1, then $f_1 = p_1x + q_1y + r_1$,
Rule 2: If x is A_2 and y is B_2, then $f_2 = p_2x + q_2y + r_2$.

Figure 1(a) and (b) illustrate the fuzzy reasoning mechanism and the corresponding ANFIS architecture, respectively. Node functions in the same layer of the ANFIS are of the same function family, as described below. (Note that O_i^j denotes the output of i-th node in layer j.)

Layer 1 Each node in this layer corresponds to a linguistic label and the node output is equal to the membership value of this linguistic label. The parameters of a node can change the shape of the membership function used to characterize the linguistic label. For instance, the node function of i-th node is

$$O_i^1 = \mu_{A_i}(x) = \frac{1}{1 + [(\frac{x - c_i}{a_i})^2]^{b_i}}, \qquad (1)$$

where x is the input to node i; A_i is the linguistic label (*small*, *large*, etc.) associated with this node; $\{a_i, b_i, c_i\}$

0-7803-0614-7/93/$03.00 ©1993 IEEE

Authorized licensed use limited to: National Taiwan University. Downloaded on June 29,2025 at 14:18:31 UTC from IEEE Xplore. Restrictions apply.

is the parameter set. Parameters in this layer are referred to as *premise parameters*.

Layer 2 Each node in this layer calculates the firing strength of each rule:

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y),\ i = 1, 2. \tag{2}$$

Layer 3 The i-th node of this layer calculates the ratio of the i-th rule's firing strength to the sum of all rules' firing strengths:

$$O_i^3 = \overline{w}_i = \frac{w_i}{w_1 + w_2},\ i = 1, 2. \tag{3}$$

Layer 4 Node i in this layer has the following node function

$$O_i^4 = \overline{w}_i f_i = \overline{w}_i(p_i x + q_i y + r_i), \tag{4}$$

where $\overline{w}_i$ is the output of layer 3, and $\{p_i, q_i, r_i\}$ is the parameter set. Parameters in this layer will be referred to as *consequent parameters*.

Layer 5 The single node in this layer computes the overall output as the summation of all incoming signals:

$$O_1^5 = overall\ output = \sum_i \overline{w}_i f_i = \frac{\sum_i w_i f_i}{\sum_i w_i} \tag{5}$$

Thus we have constructed an adaptive network (Figure 1(b)) which is functionally equivalent to a fuzzy inference system (Figure 1(a)), therefore it is called ANFIS which stands for adaptive-network-based fuzzy inference systems.

The basic learning rule of ANFIS is the back-propagation-type gradient descent [16] which calculates the error rates (defined as the derivative of the squared error with respect to each node's output) recursively from the output backward to the input nodes. This learning rule is exactly the same as the back-propagation learning rule used in the artificial neural networks [12].

From the ANFIS architecture in Figure 1, it is observed that given the values of premise parameters, the overall output f can be expressed as a linear combinations of the consequent parameters:

$$\begin{aligned} f &= \overline{w}_1 f_1 + \overline{w}_2 f_2 \\ &= (\overline{w}_1 x)p_1 + (\overline{w}_1 y)q_1 + (\overline{w}_1)r_1 \\ &\quad +(\overline{w}_2 x)p_2 + (\overline{w}_2 y)q_2 + (\overline{w}_2)r_2. \end{aligned} \tag{6}$$

As a result, we have developed a hybrid learning algorithm [1, 3] which combines the gradient descent and the least squares estiamte. More specifically, in the forward pass of the hybrid learning algorithm, functional signals (node outputs) go forward till layer 4 and the consequent parameters are identified by the least squares estimate. In the backward pass, the error rates propagate backward and the premise parameters are updated by gradient descent. Note that the least squares estimate has a well-defined sequential version which can account for the time varying characteristics of the given data pairs, therefore the hybrid learning rule can be easily adapted to a on-line paradigm [3].

III. Predicting a Chaotic Time Series

The time series used in our simulation is generated by the chaotic Mackey-Glass differential delay equation [6] defined below:

$$\frac{dx(t)}{dt} = \frac{0.2x(t-\tau)}{1 + x^{10}(t-\tau)} - 0.1x(t). \tag{7}$$

The prediction of future values of this time series is a benchmark problem which has been considered by a number of connectionist researchers (Lapedes and Farber [5], Moody [8, 7], Jones et al. [4], Crower [11] and Sanger [13]). The goal of the task is to use known values of the time series up to the point $x = t$ to predict the value at some point in the future $x = t + P$. The standard method for this type of prediction is to create a mapping from D points of the time series spaced $\triangle$ apart, that is, $(x(t-(D-1)\triangle), ..., x(t-\triangle), x(t))$, to a predicted future value $x(t+P)$. To allow comparison with earlier work (Lapedes and Farber [5], Moody [8, 7], Crower [11]), the values $D = 4$ and $\triangle = P = 6$ were used. All other simulation settings in this example were purposedly arranged to be as close as possible to those reported in [11].

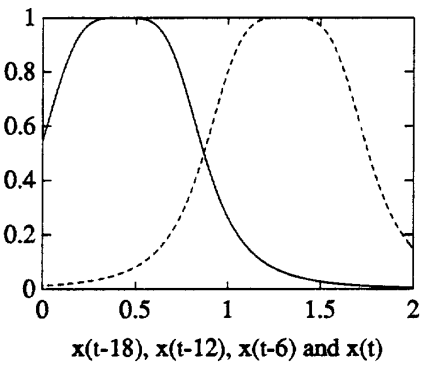
To obtain the time series value at each integer point, we applied the fourth-order Runge-Kutta method to find the numerical solution to equation (7). The time step used in the method is 0.1, initial condition $x(0) = 1.2$, $\tau = 17$, and $x(t)$ is thus derived for $0 \le t \le 2000$. (We assume $x(t) = 0$ for $t < 0$ in the integration.) From the Mackey-Glass time series $x(t)$, we extracted 1000 input-output data pairs of the following format:

$$[x(t-24), x(t-18), x(t-12), x(t-6); x(t)], \tag{8}$$

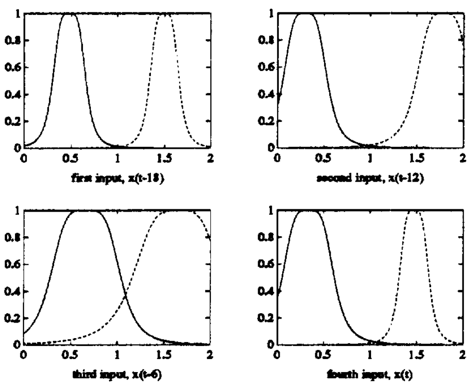
where $t = 124$ to 1123. The first 500 pairs (training data set) was used for training the ANFIS while the remaining 500 pairs (checking data set) were used for validating the identified model. The number of membership functions assigned to each input of the ANFIS is arbitrarily set to 2, so the rule number is 16. Figure 2 (a) is the initial membership functions for each input variable. Note that the ANFIS used here contains a total of 104 fitting parameters, of which 24 are premise parameters and 80 are consequent parameters

After 499.5 epochs, we had $RMSE_{trn} = 0.0016$ and $RMSE_{chk} = 0.0015$, which are much better when compared with other approaches explained below. The desired and predicted values for both training data and checking data are essentially the same in Figure 3(a); their differences (Figure 3(b)) can only be seen on a finer scale. Figure 2 (b) is the

Authorized licensed use limited to: National Taiwan University. Downloaded on June 29,2025 at 14:18:31 UTC from IEEE Xplore. Restrictions apply.



(a) Initial MF's for all four input variables.



(b) Final MF's for each input variable.

Figure 2: *Membership functions, (a) before learning; (b) after learning.*

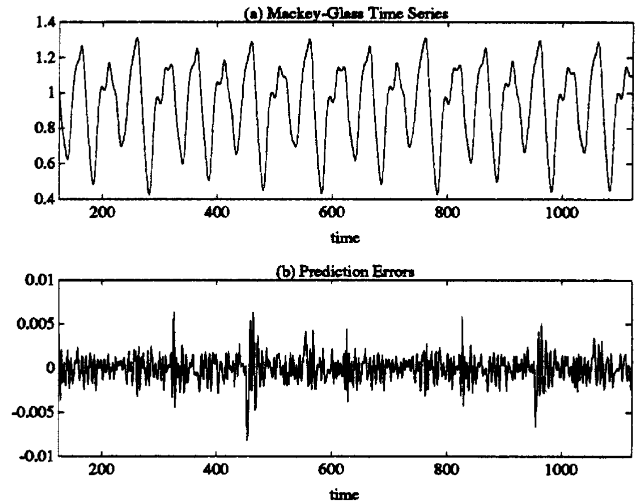


Figure 3: Example 3, (a) Mackey-Glass time series from $t = 124$ to 1123 and one-step ahead prediction (which is indistinguishable from the time series here); (b) prediction error.

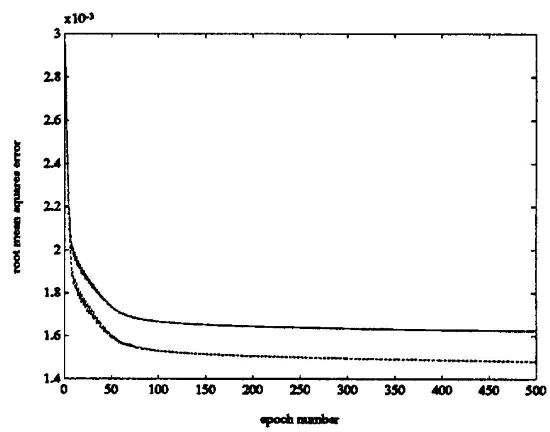


Figure 4: *Training and checking RMSE curves for ANFIS modeling.*

final membership functions; Figure 4 shows the RMSE curves which indicate most of the learning was done in the first 100 epochs. It is interesting to note the unusual phenomenon that $RMSE_{trn} < RMSE_{chk}$ during the learning process. Considering both the RMSE is vary small, we conclude that this phenomenon is due to the following two facts: (1) the ANFIS has captured the essential components of the underlying dynamics; (2) the training data contains the effects of the initial conditions (remember that we set $x(t) = 0$ for $t \leq 0$ in the integration) which might not be easily accounted for by the essential components identified by the ANFIS.

As a comparison, we performed the same prediction by using the auto-regressive (AR) model with the same number of parameters:

Authorized licensed use limited to: National Taiwan University. Downloaded on June 29,2025 at 14:18:31 UTC from IEEE Xplore. Restrictions apply.

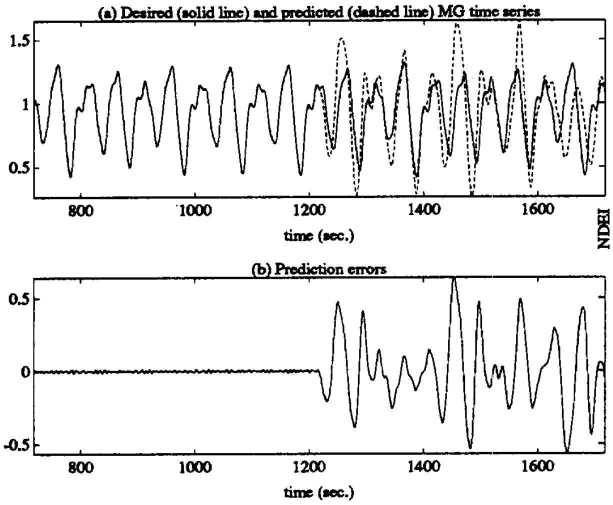


Figure 5: (a) Mackey-Glass time series (solid line) from t = 718 to 1717 and one-step ahead prediction (dashed line) by AR model with parameter = 104; (b) prediction errors.

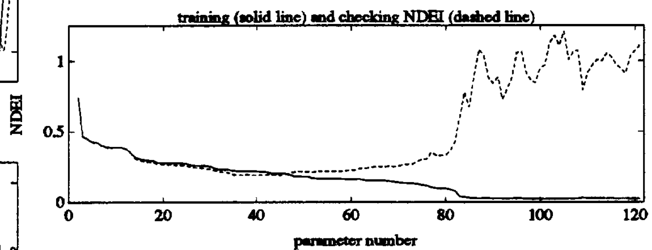


Figure 6: *Training and checking errors of AR models with different parameter numbers.*

$$x(t) = a_0 + a_1 x(t-6) + a_2 x(t-2*6) + \ldots + a_{103} x(t-103*6), \quad (9)$$

where there are 104 fitting parameters a_k, $k = 0$ to 103. From $t = 718$ to 1717, we extracted 1000 data pairs, of which the first 500 were used to identify a_k and the remaining were used for checking. The results obtained through the standard least squares estimate are $RMSE_{trn} = 0.005$ and $RMSE_{chk} = 0.078$ which is much worse than those of ANFIS. Figure 5 shows the predicted values and the prediction errors; it is obvious the over-fitting of the training data cause large errors in the checking data, which in term is caused by the over-parameterization of equation (9). To search for the best AR model in terms of generalization capability, we tried out AR models with parameter number being varied from 2 to 104; Figure 6 shows the results where the AR model with the best generalization capability is obtained when the parameter number is 45. Based on this best AR model, we repeat the generalization test and Figure 7 shows the results where there is no over-fitting at the price of larger training error. It goes without saying that the nonlinear ANFIS outperforms the linear AR model. However, it should be noted that the identification of the AR model took less than one second, while the ANFIS simulation took about 2.5 hours on a HP Apollo 700 Series workstation. (We did not pay special attention on the optimization of the C programs, though.)

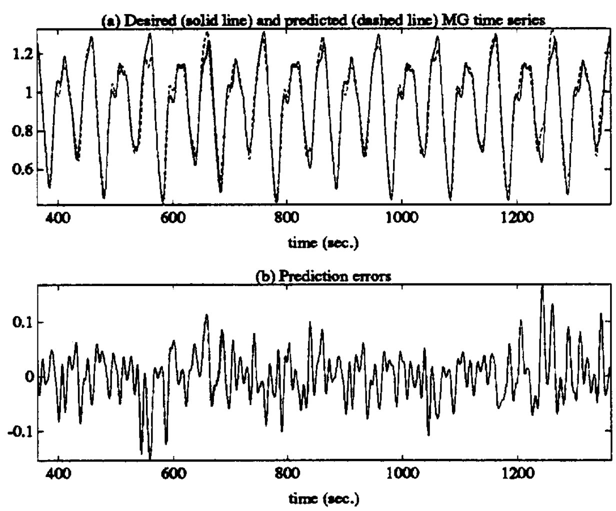


Figure 7: Example 3, (a) Mackey-Glass time series (solid line) from t = 364 to 1363 and one-step ahead prediction (dashed line) by the best AR model (parameter number = 45); (b) prediction errors.

Table 1 lists other methods' generalization capabilities which are measured by using each method to predict 500 points im-

Authorized licensed use limited to: National Taiwan University. Downloaded on June 29,2025 at 14:18:31 UTC from IEEE Xplore. Restrictions apply.

mediately following the training set. Here the *non-dimensional error index* (NDEI) is defined as the root mean square error divided by the standard deviation of the target series [5, 11, 15]. The outstanding generalization capability of ANFIS, we believe, comes from the following:

1. The ANFIS can achieve a highly nonlinear mapping [1] and therefore it is superior to linear methods in reproducing nonlinear time series.

2. The ANFIS used here has 104 adjustable parameters, much less than those of the cascade-correlation NN (693, the median size) and back-prop NN (about 540).

3. The initial parameter settings of ANFIS are intuitively reasonable, which leads to fast learning that captures the underlying dynamics.

Table 1: *Generalization result comparisons for $P = 6$. (The last four rows are taken from [11].)*

Method	Training Cases	Error Index (NDEI)
ANFIS	500	0.007
AR Model	500	0.19
Cascade-Correlation NN	500	0.06
Back-Prop NN	500	0.02
6th-order Polynomial	500	0.04
Linear Predictive Method	2000	0.55

Table 2 lists the results of the more challenging generalization test when $P = 84$ (the first six rows) and $P = 85$ (the last four rows). The results of the first six rows were obtained by iterating the prediction of $P = 6$ till $P = 84$. ANFIS still outperforms these statistic and connectionist methods unless a substantially large amount of training data are used instead. Figure 8 illustrates the generalization test for $P = 84$, where the first 500 points are the desired outputs of the training set while the last 500 are the predicted outputs for $P = 84$.

IV. Concluding Remarks

We have successfully applied the ANFIS to the prediction of the future values of a chaotic time series; the obtained results outperforms the AR (auto-regressive) models and other connectionist approaches. We think that the strengths of ANFIS primarily come from its effective hybrid learning rule and remarkable approximation power. Furthermore, the ability to incorporate human knowledge (which is not explored in our simulation) could make ANFIS a more advantageous candidate over others.

The application shown here is only the tip of a iceberg. Virtually the ANFIS can replace almost any feedforward neural

Table 2: *Generalization result comparisons for $P = 84$ (the first six rows) and 85 (the last four rows). Results for the first six methods are generated by iterating the solution at $P = 6$. Results for localized receptive fields (LRF) are multi-resolution hierarchies (MRH) are for networks trained for $P = 85$. (The last eight rows are taken from [11].)*

Method	Training Cases	Error Index (NDEI)
ANFIS	500	0.036
AR Model	500	0.39
Cascade-Correlation NN	500	0.32
Back-Prop NN	500	0.05
6th-order Polynomial	500	0.84
Linear Predictive Method	2000	0.60
LRF	500	0.10-0.25
LRF	10000	0.025-0.05
MRH	500	0.05
MRH	10000	0.02

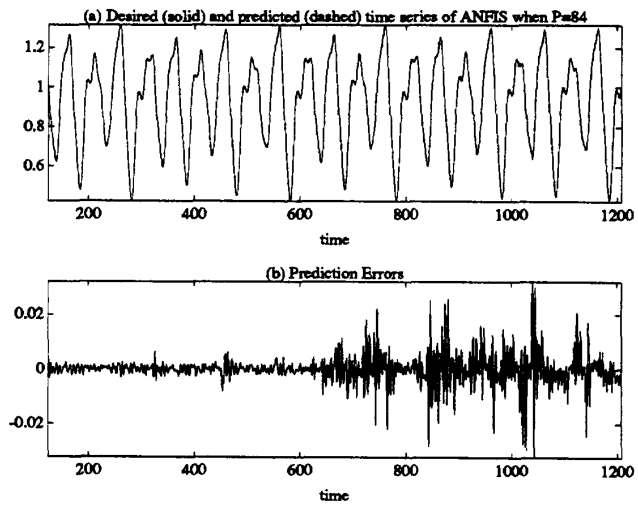


Figure 8: *Generalization test of ANFIS for $P = 84$.*

Authorized licensed use limited to: National Taiwan University. Downloaded on June 29,2025 at 14:18:31 UTC from IEEE Xplore. Restrictions apply.

networks and any regression models in almost all kind of applications. To name a few, ANFIS can replace neural networks in self-learning control [10], adaptive control [9], and adaptive filtering application [17].

References

[1] J.-S. Roger Jang. Fuzzy modeling using generalized neural networks and Kalman filter algorithm. In *Proc. of the Ninth National Conference on Artificial Intelligence (AAAI-91)*, pages 762–767, July 1991.

[2] J.-S. Roger Jang. Rule extraction using generalized neural networks. In *Proc. of the 4th IFSA World Congress*, pages 82–86 (in the Volume for Artificial Intelligence), July 1991.

[3] J.-S. Roger Jang. ANFIS: Adaptive-network-based fuzzy inference systems. *IEEE Trans. on Systems, Man, and Cybernetics*, 23(03), May 1992. (Forthcoming).

[4] R. D. Jones, Y. C. Lee, C. W. Barnes, G. W. Flake, K. Lee, and P. S. Lewis. Function approximation and time series prediction with neural networks. In *Proc. of IEEE International Joint Conference on Neural Networks*, pages I–649–665, 1990.

[5] A. S. Lapedes and R. Farber. Nonlinear signal processing using neural networks: prediction and system modeling. Technical Report LA-UR-87-2662, Los Alamos National Laboratory, Los Alamos, New Mexico 87545, 1987.

[6] M. C. Mackey and L. Glass. Oscillation and chaos in physiological control systems. *Science*, 197:287–289, July 1977.

[7] J. Moody. Fast learning in multi-resolution hierarchies. In D. S. Touretzky, editor, *Advances in Neural Information Processing Systems I*, chapter 1, pages 29–39. Morgan Kaufmann, San Mateo, CA, 1989.

[8] J. Moody and C. Darken. Fast learning in networks of locally-tuned processing units. *Neural Computation*, 1:281–294, 1989.

[9] K. S. Narendra and K. Parthsarathy. Identification and control of dynamical systems using neural networks. *IEEE Trans. on Neural Networks*, 1(1):4–27, 1990.

[10] D. H. Nguyen and B. Widrow. Neural networks for self-learning control systems. *IEEE Control Systems Magazine*, pages 18–23, April 1990.

[11] III R. S. Crowder _ Predicting the Mackey-Glass time-series with cascade-correlation learning. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proc. of the 1990 Connectionist Models Summer School*, pages 117–123, Carnegic Mellon University, 1990.

[12] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning internal representations by error propagation. In D. E. Rumelhart and James L. McClelland, editors, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, volum 1*, chapter 8, pages 318–362. The MIT Press, 1986.

[13] T. D. Sanger. A tree-structured adaptive network for function approximate in high-dimensional spaces. *IEEE Trans. on Neural Networks*, 2(2):285–293, March 1991.

[14] T. Takagi and M. Sugeno. Derivation of fuzzy control rules from human operator's control actions. *Proc. of the IFAC Symp. on Fuzzy Information, Knowledge Representation and Decision Analysis*, pages 55–60, July 1983.

[15] A. A. Weigend, D. E. Rumelhart, and B. A. Huberman. Back-propagation, weight-elimination and time series prediction. In D. Touretzky, G. Hinton, and T. Sejnowski, editors, *Proc. of the 1990 Connectionist Models Summer School*, pages 105–116, Carnegic Mellon University, 1990.

[16] P. Werbos. *Beyond regression: New tools for prediction and analysis in the behavioral sciences*. PhD thesis, Harvard University, 1974.

[17] B. Widrow and R. Winter. Neural nets for adaptive filtering and adaptive pattern recognition. *IEEE Computer*, pages 25–39, March 1988.

Authorized licensed use limited to: National Taiwan University. Downloaded on June 29,2025 at 14:18:31 UTC from IEEE Xplore. Restrictions apply.