Appendix A
n-Fold Cross Validation

This technique is intended to avoid the possible bias introduced by relying on any one particular division into test and train components by partitioning the original data in several different ways and computing the average of the performances over the different partitions. When the available data is divided into n parts, this approach is called *n-fold cross-validation*. Because of the n identification and verification steps, this method is computationally expensive. An extreme variant of this is to split the N training data points into a training set of size $N-1$ and a test set of size 1 and average the squared error on the left-out pattern over the N possible ways of obtaining such partition.

The leave-one-out validation (LOO) can be calculated analytically for linear-in-parameter models [95]. The fuzzy model is linear in its consequent parameters. Hence, the LOO criteria and its derivatives can be easily used for these models:

$$\hat{\sigma}^2_{LOO} = \frac{\mathbf{y}^T \mathbf{P} \left(\text{diag}(\mathbf{P})\right)^{-2} \mathbf{P}\mathbf{y}}{N}, \tag{A.1}$$

where in the case of global identification, $\mathbf{P}$ denotes the projection matrix

$$\begin{aligned} \mathbf{y} - \widehat{\mathbf{y}} &= \mathbf{y} - \mathbf{Q}\theta && \text{(A.2)} \\ &= \mathbf{y} - \mathbf{Q}\left(\mathbf{Q}^T\mathbf{Q}\right)^{-1}\mathbf{Q}^T\mathbf{y} && \text{(A.3)} \\ &= \left(\mathbf{I}_N - \mathbf{Q}\left(\mathbf{Q}^T\mathbf{Q}\right)^{-1}\mathbf{Q}^T\right)\mathbf{y} && \text{(A.4)} \\ &= \mathbf{P}\mathbf{y}, && \text{(A.5)} \end{aligned}$$

where the $\mathbf{Q}$ matrix contains the N regressors, $\mathbf{y}$ denotes the estimated outputs of the model, and $\mathbf{P}\widehat{\mathbf{y}}$ denotes the vector of the modeling error. The matrix $diag(\mathbf{P})$ is the same size and has the same diagonal as $\mathbf{P}$, but it is zero off-diagonal, and $\mathbf{I}_N$ represents an identity matrix.

T. Kenesei and J. Abonyi, *Interpretability of Computational Intelligence-Based Regression Models*, SpringerBriefs in Computer Science,
DOI 10.1007/978-3-319-21942-4

Appendix B
Orthogonal Least Squares

An often applied solution is to prune the identified model trained with the classical cost function. In the following, model reduction techniques of this type will be considered. In general, it can be stated that linear model reduction methods are preferred to nonlinear ones because they are exhaustively studied and effectively applied for several types of problems. For that purpose the model should be linear in parameters. A possible method family is that of orthogonal techniques. These methods can roughly be divided into two groups: the rank-revealing ones, such as the SVD-QR algorithm, and those that evaluate the individual contribution of the rule or local models, such as the orthogonal least squares approach (OLS). This latter technique requires more computations, but for system identification purposes it is preferable as it gives a better approximation result. In this book OLS is applied for rule ranking and model reduction purposes. OLS works as follows (for a throughout discussion see [1]). Consider a general linear-in-parameters model;

$$\mathbf{y} = \mathbf{Z}\theta + \mathbf{e}, \tag{B.1}$$

where $\mathbf{y} = [y_1, \ldots, y_N]^T$ is the measured output, $\mathbf{Z} = [\mathbf{z_1}, \ldots, \mathbf{z_n}]^T$ is the regressor matrix ($\mathbf{z_i} = [z_{i1}, \ldots, z_{iN}]^T$, $i = 1, \ldots, h$ are the regressors), $\theta = [\theta_1, \ldots, \theta_h]$ is the parameter vector and $\mathbf{e} = [e_1, \ldots, e_N]^T$ is the prediction error. OLS transforms the columns of the regressor matrix $\mathbf{Z}$ into a set of orthogonal basis vectors in order to inspect the individual contribution of each regressor. If they were not orthogonal, they could not have been inspected individually. An orthogonalization method should be used to perform the orthogonal decomposition $\mathbf{Z} = \mathbf{VR}$ (often the simple Gram-Schmidt method is used), where $\mathbf{V}$ is an orthogonal matrix such that $\mathbf{V}^T\mathbf{V} = \mathbf{I}$ and $\mathbf{R}$. Substituting $\mathbf{Z} = \mathbf{VR}$ into Eq. B.1, we get $\mathbf{y} = \mathbf{VR}\theta + \mathbf{e} = \mathbf{Vg} + \mathbf{e}$, where $\mathbf{g} = \mathbf{R}\theta$. Since the columns $\mathbf{v_i}$ of $\mathbf{V}$ are orthogonal, the sum of squares of y_k can be written as

T. Kenesei and J. Abonyi, *Interpretability of Computational Intelligence-Based Regression Models*, SpringerBriefs in Computer Science,
DOI 10.1007/978-3-319-21942-4

$$\mathbf{y}^T\mathbf{y} = \sum_{i=1}^{h} g_i^2 \mathbf{v}_i^T \mathbf{v}_i + \mathbf{e}^T\mathbf{e}. \tag{B.2}$$

The part of the output variance $\mathbf{y}^T\mathbf{y}/N$ explained by regressors is $\sum_{i=1}^{h} g_i^2 \mathbf{v}_i^T \mathbf{v}_i/N$ and an error reduction ratio due to an individual regressor i can be defined as

$$err_i = \frac{g_i^2 \mathbf{v}_i^T \mathbf{v}_i}{\mathbf{y}^T\mathbf{y}},\ i = 1, \ldots, h. \tag{B.3}$$

This ratio offers a simple means for ordering the regressors. As [1] shows, "there are only two restrictions to the application of this subset selection technique. First, the model has to be linear in parameters. Second, the set of regressors from which the significant ones will be chosen must be precomputed." This latter restriction is an important one because it means that all regressors are fixed during this procedure. This requirement is not met by normalized RBF networks and Takagi-Sugeno fuzzy models; therefore the original version of OLS cannot be applied. It is because the normalization denominator changes with the number of selected rules; thus the fuzzy basis functions (here: regressors) change. To overcome this problem, the value of the denominator can be fixed, but in this case interpretability issues are discarded completely. However, OLS can be very useful for various purposes; modified versions of OLS can also be applied to determine the centers of radial basis functions, or to generate Takagi-Sugeno-Kang fuzzy models.

Appendix C
Model of the pH Process

The modeling and control of pH (the concentration of hydrogen ions) in a continuously stirred tank reactor (CSTR) is a well-known control problem that presents difficulties due to the nonlinearity of the process dynamics. The CSTR is shown schematically in Fig. C.1.

A dynamic model of the pH in a tank can be obtained by considering the material balances on $[Na^+]$ and the total acetate $[HAC + AC^-]$ and assuming that the acid-base equilibrium and electroneutrality relationships hold [96].

Total acetate balance:

$$F_{HAC}[HAC]_{in} - (F_{HAC} + F_{NaOH})[HAC + AC^-] = V\frac{d[HAC + AC^-]}{dt}.$$

Sodium ion balance:

$$F_{NaOH}[NaOH]_{in} - (F_{HAC} + F_{NaOH})[Na^+] = V\frac{d[Na^+]}{dt}.$$

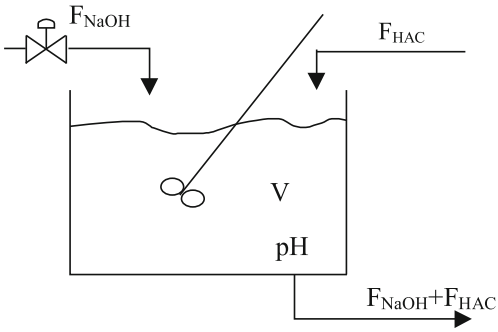

Fig. C.1 Scheme of the pH setup

T. Kenesei and J. Abonyi, *Interpretability of Computational Intelligence-Based Regression Models*, SpringerBriefs in Computer Science,
DOI 10.1007/978-3-319-21942-4

Table C.1 Parameters used in the simulations

Parameter	Description	Nominal value
V	Volume of the tank	1000 [l]
F_{HAC}	Flow rate of acetic acid	81 [l/min]
F_{NaOH}	Flow rate of NaOH	515 [l/min]
$[NaOH]_{in}$	Inlet conc. of NaOH	0.05 [mol/l]
$[HAC]_{in}$	Inlet conc. of acetic acid	0.32 [mol/l]
$[Na^+]$	Initial conc. of sodium in the CSTR	0.0432 [mol/l]
$[HAC + AC^-]$	Initial conc. of acetate in the CSTR	0.0432 [mol/l]
K_a	Acid equilibrium constant	1.75310^{-5}
K_w	Water equilibrium constant	10^{-14}

HAC equilibrium:

$$\frac{[AC^-][H^+]}{[HAC]} = K_a.$$

Water equilibrium:

$$[H^+][OH^-] = K_w.$$

Electroneutrality:

$$[Na^+] + [H^+] = [OH^-] + [AC^-].$$

The pH can be calculated from the previous equations as

$$[H^+]^3 + [H^+]^2(K_a + [Na^+]) + [H^+]([Na^+]K_a - [HAC + AC^-]K_a - K_w) - K_w K_a = 0,$$

$$pH = -\log[H^+].$$

The parameters used in our simulations are taken from [97] and are given in Table C.1.

Appendix D
Model of Electrical Water Heater

The schematic diagram of the water heater is shown in Fig. D.1.

The water comes from the water pipeline into the heater through a control valve and a pair of metal pipes containing a cartridge heater. The control task is to control the T_{out} outlet temperature by adjusting the u heating signal of the cartridge heater.

The temperature measurement is realized using Pt100 thermometers. The system has four analog inputs (inlet temperature T_{in}, outlet temperature T_{out}, valve position, and the flow rate F), and two digital (open and close in the valve, or CVO and

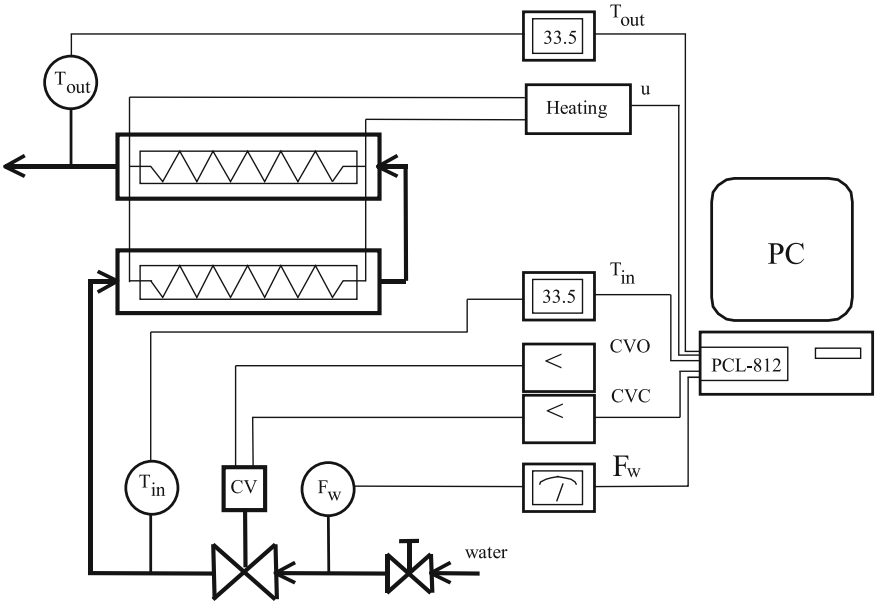

Fig. D.1 The scheme of the physical system

T. Kenesei and J. Abonyi, *Interpretability of Computational Intelligence-Based Regression Models*, SpringerBriefs in Computer Science,
DOI 10.1007/978-3-319-21942-4

CVC) and one analog output (heating control signal, u). The heaters are linked in parallel and have a performance of 1 kW. The process is connected to a PC computer through ADVANTECH LabCard PCLD-780 and PCL-812 data acquisition boards. The GENIE 3.02 data acquisition and control software was used to filter and convert the input signals (0–5 V). The control algorithm runs in MATLAB 4.2. The sampling time of the control system is seconds [98].

For the purpose of physical modeling, the system was decomposed into four interacting elements: the cartridge heater (subscript h), the streaming water (subscript w), the pipe wall (subscript p) and the environment (subscript e). The following three heat balances in the form of partial differential equations can be established:

$$
\begin{aligned}
V_h \rho_h C_{p_h} \frac{\partial T_h}{\partial t}(t,z) &= Q(u) - \alpha_1 A_1 (T_h - T_w), \\
V_w \rho_w C_{p_w} \frac{\partial T_w}{\partial t}(t,z) + (F \rho C_p)_w \frac{\partial T_w}{\partial z}(t,z) &= \alpha_1 A_1 (T_h - T_w) - \alpha_2 A_2 (T_w - T_p), \\
V_p \rho_p C_{p_p} \frac{\partial T_p}{\partial t}(t,z) &= \alpha_2 A_2 (T_w - T_p) - \alpha_e A_e (T_p - T_e),
\end{aligned}
$$

with $z \in [0, L]$ where L denotes the length of the pipe. The description and the nominal values of the parameters are given in Table D.1.

Table D.1 Parameters used in the simulation model of the heating system

Parameter	Description	Nominal value
L	Length of the pipe	2×48010^{-3} m
ρ_h	Density of the cartridge	3650 kg/m^3
C_{ph}	Heat capacity of the cartridge	1047 J/kg K
A_h	Surface of the cartridge	2.41×10^{-2} m^2
V_h	Volume of the cartridge	4.82×10^{-5} m^3
α_1	$h - w$ heat transfer coefficient	316.3 W m^{-2} K^{-1}
ρ_w	Density of the water	1000 kg/m^3
C_{pw}	Heat capacity of the water	4186 J/kg K
T_{in}	Inlet water temperature	11.8 °C
V_w	Volume of the water	1.16×10^{-4} m^3
α_2	$w - p$ heat transfer coefficient	1196.1 W m^{-2} K^{-1}
ρ_p	Density of the wall	7850 kg/m^3
C_{pp}	Heat capacity of the wall	502 J/kg K
t_e	Temperature of the environment	21.6 °C
A_p	Inner surface of the wall	4.46×10^{-2} m^2
V_p	Volume of the wall	7.37×10^{-5} m^3
A_e	Outer surface of the wall	5.36×10^{-2} m^2
α_e	$p - e$ heat transfer coefficient	1015.9 W m^{-2} K^{-1}

The performance of the cartridge heater is given by

$$Q(u) = Q_M \left[u - \frac{\sin(2\pi u)}{2\pi} \right], \tag{D.1}$$

where Q_M is the maximal power, and u is the heating signal (voltage). The partial differential equations are approximated by eight compartments of equal volume. As Eq. D.1 shows, the heating performance is a static nonlinear function of the heating signal (control input).

References

1. Nelles, O. (2001). *Nonlinear system identification*. New York: Springer.
2. Mamdani, E. H., Teraqno, T., Asai, K., & Sugeno, M. (1992). Fuzzy-systems theory and its applications. *Nature*, *359*, 788–788.
3. Mendel, J. M. (1995). Fuzzy logic systems for engineering: A tutorial. *Proceedings of the IEEE*, *83*, 345–377.
4. Bhat, N. V., & McAvoy, T. J. (1990). Use of neural nets for dynamic modelling and control of chemical process systems. *Computers and Chemical Engineering*, *4*(5), 573–585.
5. Bhat, N. V., Minderman, P. A., McAvoy, T. J., & Wang, N. S. (1990, April 24–30). Modelling chemical process systems via neural computation. *IEEE Control Systems Magazine*.
6. Hernandez, E., & Arkun, Y. (1993). Control of nonlinear systems using polynomial ARMA models. *AICHE Journal*, *39*(3), 446–460.
7. Ydstie, B. E. (1990). Forecasting and control using adaptive connectionist networks. *Computers and Chemical Engineering Journals*, *15*(4/5), 583–599.
8. Sjöberg, J., Zhang, Q., Ljung, L., Benveniste, A., Deylon, B., Glorennec, P.-Y., et al. (1995). Nonlinear black-box modeling in system identification: A unified overview. *Automatica*, *31*, 1691–1724.
9. Abonyi, J., & Feil, B. (2005). Computational intelligence in data mining. *Informatica*, *29*, 3–12.
10. Piatetsky-Shapiro, G. (1991). *Knowledge discovery in databases*. Cambridge, MA: AAAI Press.
11. Zadeh, L. A. (1973). Outline of a new approach to the analysis of complex systems and decision processes. *IEEE Transactions and Systems, Man, and Cybernetics*, *3*, 28–44.
12. Sugeno, M., & Kang, G. T. (1988). Structure identification of fuzzy model. *Fuzzy Sets and Systems*, *28*, 15–33.
13. Sugeno, M., & Tanaka, K. (1991). Successive identification of a fuzzy model and its application to prediction of a complex system. *Fuzzy Sets and Systems*, *42*, 315–334.
14. Sugeno, M., & Yasukawa, T. (1993). A fuzzy-logic-based approach to qualitative modeling. *IEEE Transactions on Fuzzy Systems*, *1*(1), 7–31.
15. Jang, J. -S. R. (1996). Input selection for ANFIS learning. In *Proceedings of the IEEE International Conference on Fuzzy Systems* (Vol. 2, pp. 1493–1499). New York, USA.
16. Zadeh, L. A. (1994). Fuzzy logic, neural networks, and soft computing. *Communications of the ACM*, *36*(3), 77–84.
17. Kecman, V. (2001). *Learning and Soft Computing*.

T. Kenesei and J. Abonyi, *Interpretability of Computational Intelligence-Based Regression Models*, SpringerBriefs in Computer Science,
DOI 10.1007/978-3-319-21942-4

18. Niu, S., & Pucar, P. (1995). Hinging hyperplanes for non-linear identification. http://www.control.isy.liu.se.
19. Ernst, S. (1998). Hinging hyperplane trees for approximation and identification. In *Proceedings of the 37th IEEE Conference on Decision and Control* (Vol. 2, pp. 1266–1271).
20. Breiman, L. (1993). Hinging hyperplanes for regression, classification and function approximation. *IEEE Transactions on Information Theory*, *39*, 311–325.
21. Nunez, H., Angulo, C., Catala, A. (2002). Rule extraction from support vector machines. In *European Symposium on Artificial Neural Networks Proceedings* (pp. 107–112).
22. Chen, Y. X., & Wang, J. Z. (2003). Support vector learning for fuzzy rule-based classification systems. *IEEE Transactions on Fuzzy Systems*, *11*, 716–728.
23. Chen, Y. X., & Wang, J. Z. (2003). Kernel machines and additive fuzzy systems: Classification and function approximation. In *Proceedings of IEEE International Conference on Fuzzy Systems* (pp. 789–795).
24. Cortes, C., & Vapnik, V. (1995). *Support-vector networks*. USA: AT&T Research Labs.
25. Jacek, Q., & Leski, M. (2006). On support vector regression machines with linguistic interpretation of the kernel matrix. *Fuzzy Sets and Systems*, *157*, 1092–1113.
26. Cun, Y. L., Denke, J., & Solla, S. (1990). Optimal brain damage. *Advances in Neural Information Processing Systems*, *2*, 598–605.
27. Duch, W. (2003). Coloring black boxes: Visualization of neural network decisions. *The International Joint Conference on Neural Networks Portland*, *1*, 1735–1740.
28. Duch, W. (2004). Visualization of hidden node activity in neural networks: I. visualization methods. *Lecture Notes in Artificial Intelligence*, *3070*, 38–43.
29. Duch, W. (2004). Visualization of hidden node activity in neural networks: II. application to RBF networks. *Lecture Notes in Artificial Intelligence*, *3070*, 44–49.
30. Henrique, H. M., Lima, E. L., & Seborg, D. E. (2000). Model structure determination in neural network models. *Chemical Engineering Science*, *55*, 5457–5469.
31. Benitez, J. M., Castro, J. L., & Requena, I. (1995). Are artifical neural networks black boxes? *IEEE Transactions on Neural Networks*, *8*(5), 1156–1164.
32. Abonyi, J. (2003). *Fuzzy model identification for control*. Boston: Birkhäuser.
33. Ramirez, D. R., Camacho, E. F., & Arahal, M. R. (2004). Implementation of Min-Max MPC using hinging hyperplanes to a heat exchanger. *Control Engineering Practice*, *12*, 1197–1205.
34. Pucar, P., & Sjoberg, J. (1995). Parameterization and conditioning of hinging hyperplane models. In *Technical Report LiTH-ISY-R-1809* (Vol. 40, pp. 37–50), Department of Electrical Engineering, Linkoping University.
35. Wen, C., Wang, S., Jin, X., & Ma, X. (2007). Identification of dynamic systems using piecewise-affine basis function models. *Automatica*, *43*, 1824–1831.
36. Roll, J., Bemporad, A., & Ljung, L. (2004). Identification of piecewise affine systems via mixed-integer programming. *Automatica*, *40*, 37–50.
37. Hathaway, R., & Bezdek, J. (1993). Switching regression models and fuzzy clustering. *IEEE Transactions on fuzzy systems*, *3*, 195–204.
38. Abonyi, J., & Feil, B. (2007). *Cluster analysis for data mining and system identification*. Basel: Birkhäuser.
39. Sala, A., Guerrab, T. M., & Babuska, R. (2005). Perspectives of fuzzy systems and control. *Fuzzy Sets and Systems*, *156*, 432–444.
40. Hellendoorn, H., & Precup, R. (2011). A survey on industrial applications of fuzzy control. *Computers in Industry*, *63*, 213–226.
41. Feng, G. (2011). A survey on analysis and design of model-based fuzzy control systems. *IEEE Transactions on Fuzzy Systems*, *14*(5), 676–697.
42. Babuška, R. (1998). *Fuzzy Modeling for Control*. Boston: Kluwer Academic Publishers.
43. Abonyi, J., Babuska, R., Setnes, M., Verbruggen, H. B., & Szeifert, F. (1999). Constrained parameter estimation in fuzzy modeling. In *Proceedings FUZZ-IEEE'99* (pp. 951–956).
44. Abonyi, J., Babuska, R., Szeifert, F., & Nagy, L. (2000). Identification and control of nonlinear systems using fuzzy Hammerstein models. *Industrial and Engineering Chemistry Research*, *39*(11), 4302–4314.

45. Kubat, M. (1998). Decision trees can initialize radial-basis-function networks. *IEEE Transactions on Neural Networks*, *9*, 813–821.
46. Wen, C., & Ma, X. (2008). A max-piecewise-linear neural network for function approximation. *Neurocomputing*, *71*, 843–852.
47. Quinlan, J. R. (1986). Induction of decision trees. *Machine Learning*, *1*, 81–106.
48. Pucar, P., & Sjoberg, J. (1998). On the hinge finding algorithm for hinging hyperplanes. *IEEE Transaction on Information Theory*, *44*, 1310–13109.
49. Bemporad, A., Garulli, A., Paoletti, S., & Vicino, A. (2005). A bounded-error approach to piecewise affine system identification. *IEEE Transactions on Automatic Control*, *50*, 1567–1280.
50. Friedman, J. H. (1991). Multivariate adaptive regression splines. *The Annals of Statistics*, *19*, 1–141.
51. Cai, Q., Hao, Z., & Yang, X. (2012). Gaussian kernel-based fuzzy inference systems for high dimensional regression. *Neurocomputing*, *77*, 197–204.
52. Weierstrass, K. (1872). Über continuirliche Functionen eines reellen Arguments die für keinen Werth des letzeren einen bestimmten Differentialquotienten besitzen. Königliche Akademie der Wissenschaften.
53. Weierstrass, K. (1885). *Über die analytische Darstellberkeit sogenannter willkürlicher Functionen einer reellen veränderlichen*. Berlin: Sitzungsberichte der Königlich Preussischen Akademie der Wissenschaften zu.
54. Hilbert, D. (1900). Mathematische Probleme. *2. International Congress of Mathematicians*, Paris, France.
55. Arnold, V. I. (1957). On functions of three variables. *Doklady Akademii Nauk USSR*, *114*, 679–681.
56. Kolmogorov, A. N. (1957). On the representation of continuous functions of many variables by superpositions of continuous functions of one variable and addition. *Doklady Akademii Nauk USSR*, *114*, 953–956.
57. Sprecher, D. A. (1965). On the structure of continuous functions of several variables. *Transactions of the American Mathematical Society*, *115*, 340–355.
58. Lorentz, G. G. (1966). *Approximation of functions*.
59. De Figueiredo, J. P. (1980). Implications and applications of Kolmogorov's superposition theorem. *IEEE Transactions on Automated Control*, *25*(6), 1127–1231.
60. Hornik, K., Stinchcombe, M., & White, H. (1989). Multilayer feedforward networks are universal approximators. *Neural Networks*, *2*, 359–366.
61. Blum, E. K., & Li, L. K. (1991). Approximation theory and feedforward networks. *Neural Networks*, *4*(4), 511–515.
62. Kurková, V. (1992). Kolmogorov's theorem and multilayer neural networks. *Neural Networks*, *5*, 501–506.
63. Zadeh, L. A. (1965). Fuzzy sets. *Information and Control*, *8*, 338–353.
64. Wang, L. X. (1992). Fuzzy systems are universal approximators. In *IEEE International Conference on Fuzzy Systems* (pp. 1163–1169). San Diego, USA.
65. Kosko, B. (1994). Fuzzy systems are universal approximators. *IEEE Transactions on Computers*, *43*(11), 1329–1333.
66. Castro, J. L. (1995). Fuzzy logic controllers are universal approximators. *IEEE Transactions on SMC*, *25*, 629–635.
67. Moser, B. (1999). Sugeno controllers with a bounded number of rules are nowhere dense. *Fuzzy Sets and Systems*, *104*(2), 269–277.
68. Tikk, D. (1999). On nowhere denseness of certain fuzzy controllers containing prerestricted number of rules. *Tatra Mountains Mathematical Publications*, *16*, 369–377.
69. Klemet, E. P., Kóczy, L. T., & Moser, B. (1999). Are fuzzy systems universal approximators? *International Journal of General Systems*, *28*(2), 259–282.
70. Stone, M. H. (1937). Applications of the theory of boolean rings to general topology. *Transactions of the American Mathematical Society*, *41*(3), 375–481.

71. Stone, M. H. (1948). The generalized Weierstrass approximation theorem. *Mathematics Magazine*, *21*(4), 167–184.
72. Cotter, N. E. (1990). The Stone-Weierstrass theorem and its application to neural networks. *IEEE Transaction on Neural Networks*, *1*(4), 290–295.
73. Wang, L. X., & Mendel, J. M. (1992). Fuzzy basis functions, universal approximators and orthogonal least squares learning. *IEEE Transactions on Neural Networks*, *3*, 807–814.
74. Setiono, R., Leow, W. K., & Thong, J. Y. L. (2000). Opening the neural network black box: An algorithm for extracting rules from function approximating artificial neural networks. In *Proceedings of the 21st International Conference on Information Systems* (pp. 176–186). Queensland.
75. de Castro, L. N., Iyoda, E. M., Zuben, F. J. V., & Gudwin, R. (1998). Feedforward neural network initialization: An evolutionary approach. In *Proceedings of the 5th Brazilian Symposium on Neural Networks* (pp. 43–49).
76. Chumieja, M., & Markowska-Kaczmar, U. (2003). Opening neural network black box by evolutionary approach. In *Design and Application of Hybrid Intelligent Systems* (pp. 147–156).
77. Bath, N. V., & McAvoy, T. J. (1992). Determining model structure for neural models by network stripping. *Computers and Chemical Engineering*, *16*(4), 271–281.
78. Wolff, G., Hassibi, B., & Stork, D. (1992). Optimal brain surgeon and general network pruning. *Technical report*.
79. Chiang, J. H., & Hao, P. Y. (2004). Support vector learning mechanism for fuzzy rule-based modeling: A new approach. *IEEE Transactions on Fuzzy Systems*, *12*, 1–12.
80. Yang, X., Cai, Q., & Haom, Z. (2011). Gaussian kernel-based fuzzy inference system for high dimensional regression. *Neurocomputing*.
81. Castro, L., Flores-Hidalgo, L. D., Mantas, C. J., & Puche, J. M. (2007). Extraction of fuzzy rules from support vector machines. *Fuzzy Sets and Systems*, *158*(18), 2057–2077.
82. Hong, D. H., & Hwang, C. (2003). Support vector fuzzy regression machines. *V*, *138*(2), 271–281.
83. Gunn, S. R. (1998). Support vector machines for classification and regression. *Technical report*.
84. Juang, C.-F., & Hsieh, C.-D. (2009). TS-fuzzy system-based support vector regression. *Fuzzy Sets and Systems*, *160*(17), 2486–2504.
85. Celikyilmaz, A., & Türksen, I. B. (2007). Fuzzy functions with support vector machines. *Information Sciences*, *177*(23), 5163–5177.
86. Wang, H.-F., & Tsaur, R.-C. (2000). Insight of a fuzzy regression model. *Fuzzy Sets and Systems*, *112*(3), 355–369.
87. Üstün, B., Buydens, L. M. C., & Melssen, W. J. (2007). Visualisation and interpretation of support vector regression models. *Analytica Chimica Acta*, *595*(1–2), 299–309.
88. Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Üller, K.-R. M., Ratsch, G., & Smola., A. (1999). Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, *10*(5).
89. Setnes, M., Babuska, R., Kaymak, U., & van Nauta Lemke, H. R. (1998). Similarity measures in fuzzy rule base simplification. *IEEE Transactions on SMC-B*, *28*, 376–386.
90. Yen, J., & Wang, L. (1999). Simplifying fuzzy rule-based models using orthogonal transformation methods. *IEEE Transactions on SMC-B*, *29*, 13–24.
91. Setnes, M., & Hellendoorn, H. (2000). Orthogonal transforms for ordering and reduction of fuzzy rules. In *FUZZ-IEEEE* (p. 700.705). San Antonio, Texas.
92. Smola, A. M., & Schölkopf, B. (2003). A tutorial on support vector regression. *NeuroCOLT Tehnical Report*.
93. Jin, Y. (2000). Fuzzy modeling of high-dimensional systems. *IEEE Transactions on Fuzzy Systems*, *8*, 212–221.
94. Setnes, M., & Babuška, R. (2001). Rule base reduction: Some comments on the use of orthogonal transforms.
95. Orr, M. J. L. (1996, April). Introduction to radial basis function networks. In *Research Report*. Centre of Cognitive Science, University of Edinburgh.

96. McAvoy, T. J., Hsu, E., & Lowenthal, S. (1972). Dynamics of pH in controlled stirred tank reactor. *Industrial Engineering Chemical Process Design and Development*, *11*(1), 68–70.
97. Bhat, N. V., & McAvoy, T. J. (1992). Determining model structure for neural models by network stripping. *Computers and Chemical Engineering*, *16*, 271–281.
98. Abonyi, J., Bodizs, A., Nagy, L., & Szeifert, F. (1999) Hybrid fuzzy convolution model based predictor corrector controller. In M. Mohamadian (Ed.), *Computational intelligence for modelling control and automation* (pp. 265–270). IOS Press, ISBN 9-051-99474-5.

Index

T. Kenesei and J. Abonyi, *Interpretability of Computational Intelligence-Based Regression Models*, SpringerBriefs in Computer Science,
DOI 10.1007/978-3-319-21942-4