Proceedings of the Korean Nuclear Society Spring Meeting
Kwangju, Korea, May 2002

Sensor Monitoring Using a Fuzzy Neural Network with an Automatic Input Selection and Rule Generation Algorithm

Man Gyun Na, Young Rok Sim, and Kyung Ho Park

Department of Nuclear Engineering, Chosun University

375 Seosuk-dong, Dong-gu, Kwangju 501-759, Korea

Abstract

The performance of fuzzy neural networks applied to sensor signal estimation strongly depends on the selection of input signals. In estimating sensor signals for sensor failure detection, there are usually a large number of input signals related to output signal estimation. As the number of input variables increases, the required training time of a fuzzy neural network increases exponentially and also there is much possibility which it has wrong information. Thus, it is needed to reduce the number of inputs to a fuzzy neural network and moreover, to select the optimum number of mutually independent inputs that are able to clearly define the input-output mapping. In this work, to automatically select important input signals, an automatic input selection routine that combines the correlation analysis and genetic algorithm is got in a fuzzy neural network which estimates a specific relevant signal. Also, since the number of fuzzy inference rules depends on that of selected inputs, the number of its rules is decided automatically according to the number of inputs. Whether the sensors fail or not is determined by applying the sequential probability ratio test to the residuals between the actually measured signals and the signals estimated by the fuzzy neural network. The proposed sensor monitoring method was verified by using various sensor signals acquired in Yonggwang unit 3&4 pressurized water reactors.

Index Terms **–** fuzzy neural network, genetic algorithm, input selection, sensor failure detection, sequential probability ratio test

1. Introduction

Many artificial intelligence techniques including neural networks and fuzzy inference methods have recently been proposed to detect sensor failures in a nuclear engineering field. This work was started in the late 80's by Upadhyaya [1,2], was also conducted by Singer et. al. [3], Hines and Uhrig, et. al. [4,5], and Fantoni, et. al [6]. These techniques have been known to have the good capability for data-driven plant model identification, especially when expert diagnostic knowledge and the prior relation of fault symptom model are not clear. It is experienced that the performance of a neural network strongly depends on which input data are used for its output estimation. Non-salient input data to an artificial neural network can have even negative impacts on the performance. In recent years, the general problem of selecting a proper input set for fuzzy neural networks has been generating a great deal of interest.

If a large number of input signals are used in the fuzzy neural network, it would require a large amount of time to train a fuzzy neural network since the number of the training parameters such as connection weights for neural networks and parameters for fuzzy inference system would be extremely large. Thus, it is essential to reduce the number of inputs to a neural network and to select the optimum number of mutually independent inputs that are able to clearly define the input-output mapping and to select the optimum number of fuzzy inference rules according to the selected inputs. Also, by eliminating unimportant sensors and sensor parameters, the cost and time of collecting the data can be reduced. Many input selection methods have been developed including principal component analysis (PCA), genetic algorithm (GA), and others [7-13]. Although the PCA input selection method can reduce the number of inputs to the fuzzy neural network, the PCA method has a disadvantage of not reducing the number of the input signals actually used, which increases a possibility that we use unreliable and faulty sensor signals. Also, genetic algorithms have a disadvantage that it requires too much computational time. Therefore, in this work, to reduce computing time a correlation analysis and the GA are combined to select important input signals.

An important problem in sensor failure decision is to decide whether a sensor fails or not after only one abnormal observation. At every new sample of a sensor signal, a new mean and a new variance may be computed and then, these quantities may be used to check if the sensor fails or not. However, this procedure requires too many samples to obtain a meaningful mean and a meaningful variance and also, during the acquisition of the samples, a significant degradation of the process monitored may occur. Therefore, in this work the sequential probability ratio test (SPRT) [14] was used. The method can decide a failure using the degree of degradation and the continuous behavior of the sensor, without having to calculate a new mean and a new variance at each sample. The signal estimated by the fuzzy neural network is compared with the measured signal, and then the sensor failure is determined by the SPRT using the residual which is the difference between the estimated signal and the measured signal.

A proposed failure detection algorithm will be applied to monitor the steam generator water level, the hot-leg temperature, the ex-core neutron flux sensors in Yonggwang unit 3&4 pressurized water reactors and compared with other algorithms.

2. Signal Estimation Using a Fuzzy Neural Network

2.1 Fuzzy Inference System

A system that consists of a fuzzy inference system and its neuronal training system is usually called an adaptive network-based fuzzy inference system (ANFIS) [15]. In a fuzzy inference system, the i -th rule can be described using the first-order Sugeno-Takagi type [16] as follows:

$$\text{If } x_1 \text{ is } A_{i1} \text{ } AND \cdots AND \text{ } x_m \text{ is } A_{im}, \text{ then } \hat{y}_i \text{ is } f_i(x_1, \cdots, x_m), \tag{1}$$

where

$x_1, \cdots, x_m$ = input variables to the fuzzy neural network (m = number of input variables),

$A_{i1}, \cdots, A_{im}$ = antecedent membership function of each input variable for the i -th rule (i = 1, 2, ..., n),

$\hat{y}_i$ = output of the i -th rule,

$$f_i(x_1,\cdots,x_m) = \sum_{j=1}^{m} q_{ij} x_j + r_i, \tag{2}$$

q_{ij} = weighting value of the j -th input onto the i -th rule output,

r_i = bias of the i -th output,

n = number of rules.

In this work, the sigmoid membership function is used for the maximum and minimum center values in each input variable and the Gaussian membership function is used for other center values. The output of an arbitrary i -th rule, f_i, consists of the first-order polynomial of inputs as given in Eq. (2). The output of a fuzzy inference system with n rules is obtained by weighting the real values of consequent part for all rules with the corresponding membership grade and indicates the estimated value of the relevant sensor signal. The estimated signal is obtained as follows:

$$\hat{y} = \sum_{i=1}^{n} \overline{w}_i f_i = \mathbf{w}^T \mathbf{q}, \tag{3}$$

where

$$\overline{w}_i = \frac{w_i}{\sum_{i=1}^{n} w_i},$$

$$w_i = \prod_{j=1}^{m} A_{ij}(x_j),$$

$$\mathbf{q} = [q_{11} \cdots q_{n1} \cdots \cdots q_{1m} \cdots q_{nm} \; r_1 \cdots r_n]^T,$$

$$\mathbf{w} = [\overline{w}_1 x_1 \cdots \overline{w}_n x_1 \cdots \cdots \overline{w}_1 x_m \cdots \overline{w}_n x_m \; \overline{w}_1 \cdots \overline{w}_n]^T.$$

2.2 Training of the Fuzzy Inference System

The back-propagation algorithm is a general method for recursively solving for parameter optimization of a fuzzy inference system. Since this conventional optimization algorithm is susceptible to getting stuck at local optima, the genetic algorithm is used in this work. However, the genetic algorithm requires much computational time if there are many parameters to be optimized. Therefore, the least-squares method that is a one-pass optimization method is combined for a part of the parameters. The genetic algorithm is used to optimize the antecedent parameters (center position and sharpness of membership functions), and the least-squares algorithm is used to solve the consequent parameters q_{ij} and r_i (the polynomial coefficients of the consequent part).

In optimization problems using genetic algorithms, the term *chromosome* refers to a candidate solution that minimizes a cost function, generally encoded as a bit string. Each chromosome can be thought of as a point in the search space of candidate solutions. Genetic algorithm is an optimization technique that imitates the evolutionary process of a living organism. An initial population of chromosomes is iteratively altered by mechanisms inspired by natural evolution such as selection, crossover and mutation. Thus genetic algorithms process populations of chromosomes, successively replacing one such population with another. Genetic algorithms start from many points simultaneously climbing many peaks in parallel, and hence the probability of finding a false peak is reduced compared to the methods that move from one point to another. Accordingly, genetic algorithms are less susceptible to being stuck at local optima than conventional search methods [10,17,18]. The genetic algorithm is to minimize the overall sum of squared prediction errors, the maximum absolute prediction error and the number of used sensor signals. A more detailed explanation on the genetic algorithm will be given in the next section. The genetic algorithm is applied to the membership functions optimization as well as the input signals selection and the rule number optimization.

If we fix some parameters of the fuzzy inference system by the genetic algorithm, the resulting fuzzy inference system is equivalent to a series of expansions of some basis functions. This basis function expansion is linear in its adjustable parameters. Therefore, we can use the least-squares method to determine the remaining parameters. If a total of N input-output training data are given, from Eq. (3) the consequent parameters are chosen to minimize the following cost function:

$$J = \frac{1}{2}\sum_{i=1}^{N}(y-\hat{y})^2 = \frac{1}{2}(\mathbf{y}-\hat{\mathbf{y}})^2 = \frac{1}{2}(\mathbf{y}-\mathbf{Wq})^2, \tag{4}$$

where

$$\mathbf{y} = \left[y^1 \; y^2 \; \cdots \; y^N\right]^T,$$

$$\mathbf{q} = [q_{11} \cdots q_{n1} \cdots \cdots q_{1m} \cdots q_{nm} \; r_1 \cdots r_n]^T,$$

$$\mathbf{W} = \left[\mathbf{w}^1 \; \mathbf{w}^2 \; \cdots \; \mathbf{w}^N\right]^T,$$

$$\mathbf{w}^k = [\overline{w}_1 x_1 \cdots \overline{w}_n x_1 \cdots \cdots \overline{w}_1 x_m \cdots \overline{w}_n x_m \ \overline{w}_1 \cdots \overline{w}_n]^T, \quad k = 1, 2, \cdots, N .$$

$\mathbf{y}$ is the output data vector, $\mathbf{q}$ is the parameter vector, and the matrix $\mathbf{W}$ includes the input data. An equation for minimizing the cost function is as follows:

$$\mathbf{y} = \mathbf{W}\mathbf{q} . \tag{5}$$

The fuzzy neural network output is represented by the $N \times (m+1)n$ -dimensional matrix $\mathbf{W}$ and the $(m+1)n$ -dimensional parameter vector $\mathbf{q}$. In order to solve the parameter vector $\mathbf{q}$ in Eq. (5), the matrix $\mathbf{W}$ should be invertible but is not usually a square matrix. Therefore, we solve the vector by using the pseudo-inverse as follows:

$$\mathbf{q} = \left(\mathbf{W}^T \mathbf{W}\right)^{-1} \mathbf{W}^T \mathbf{y} . \tag{6}$$

2. Input Selection and Rule Generation Methods

The number of input variables has to be reduced for several reasons. But this seems to be paradoxical at first since a dimension reduction decreases the information content. A reduction of the number of variables can lead to an improved performance due to at least three reasons. First, irrelevant inputs will result in a model which is not robust. Thus, it becomes important to use only high information descriptors. Secondly, since studies have shown that the prediction results can get worse if colinearity is present among the variables, it is necessary to remove highly correlated variables. Finally, when making a model containing many input variables, a large number of observations are required to span the complete input space. The number of required observations grows exponentially with the number of input variables, which makes a dimension reduction necessary to get a good model. The number of fuzzy inference rules depends on that of selected inputs. That is, many rules are not needed for a few input signals. Therefore, it is required to select the optimum number of rules for selected inputs. In this work, in order to select proper input signals and the optimum number of rules, a modified genetic algorithm will be developed.

The genetic algorithms require a fitness function that assigns a score to each chromosome in the current population. The fitness of a chromosome (individual) depends on how well that chromosome solves the problem at hand. In this work, a fitness function that evaluates the extent to which each individual is suitable for the given objectives such as small maximum error together with small total squared error and the small number of input variables, is suggested as follows:

$$F = \exp\left(-\mu_1 E_1 - \mu_2 E_2 - \mu_3 E_3\right), \tag{7}$$

where μ_1, μ_2 and μ_3 are the weighting coefficients, and E_1, E_2 and E_3 are the overall sum of squared prediction errors, the maximum absolute prediction error and the number of input variables, respectively, defined as

$$E_1 = \sum_{k=1}^{N} \left(y(k) - \hat{y}(k)\right)^2 , \tag{8}$$

$$E_2 = \max_k \left\{ \left| y(k) - \hat{y}(k) \right| \right\} , \tag{9}$$

$$E_3 = N_{input} . \tag{10}$$

$y(k)$ and $\hat{y}(k)$ denote the measured signal and the estimated signal, respectively. Therefore, the genetic algorithm minimizes the overall sum of squared prediction errors, the maximum absolute prediction error and the number of used sensor signals.

A chromosome is encoded as a bit string which consists of two parts of bits where one is related to the input selection and another is related to the rules number. The input selection part is composed of the same bit number as the number of input variables, and one '1' in this part represents that the corresponding input is selected and zero '0' represents that the corresponding input is not selected.

On the other hand, genetic algorithms have a disadvantage that it requires too much computational time Therefore the genetic algorithm is modified to reduce computing time. Note that the correlation coefficient matrix of the original data set is equal to the covariance matrix of the data after the data have been standardized. This correlation matrix indicates how closely variables (signals) depend one another. The high specific (i, j) component of the correlation matrix means that the two variables are closely related to each other. These values between the input variables and the output variable are used to initialize the input selection part of the chromosomes of the genetic algorithm. This is described in Fig. 1. The correlation (dotted line) between the output (circle) bit and the selected input (triangle) bit has to be as large as possible and the correlations (solid lines) between the selected input (triangle) bit and the possible inputs (cross) bits have to be as small as possible. To run a conventional genetic algorithm, each bit of the chromosomes is usually randomly assigned one or zero which represent that the corresponding input (bit) is selected or not, respectively. However, in this work, there is a high probability that the corresponding (triangle) bit is assigned one in case that a correlation between a specific input (triangle) and an output is high and correlations between the specific input (triangle) and the possible inputs (cross) are low. On the contrary, there is a high probability that the corresponding (triangle) bit is assigned zero in the case that a correlation between the specific input and the output is low and correlations between the specific input (triangle) and the possible inputs (cross) are low. This helps reduce the computational time by reducing a probability of selecting the inputs that are not almost related to the output and also, are much related to other inputs.

This algorithm for input selection and rule generation is described in Fig. 2. First, the input selection bits of the initial chromosomes are generated by using the correlation coefficient matrix, which reduces the computational burden of the genetic algorithm that requires much computational time and its rules number bits are allocated randomly at first. Also, every input selection generation a part of chromosomes with low fitness are replaced by the correlation analysis and a general genetic algorithm succeeds.

4. Failure Decision

In sensor failure decision, at every new sample of a signal, a new mean and a new variance of the signal may be required to check if the sensor is degraded or not. However, this procedure requires too many samples to obtain a meaningful mean and a meaningful variance. During the acquisition of the samples, a significant degradation of the monitored process may occur. So Sequential Probability Ratio Test (SPRT) is used to detect a sensor failure using the degree of failure and the continuous behavior of the sensor, without having to calculate a new mean and a new variance at each sample. The SPRT is a statistical model developed by Wald in 1945 [14].

The objective of sensor monitoring is to detect the failure as soon as possible with a very small probability of making a wrong decision. In the application of sensor failure detection, the SPRT uses the residual (difference between the sensor measurement and the sensor estimate). Normally the residual signals are randomly distributed, so they are nearly uncorrelated and have a Gaussian (normal) distribution $P_i(\varepsilon_k, m_i, \sigma_i)$, where ε_k is the residual signal at time k, and m_i and σ_i are the mean and the standard deviation under hypothesis i, respectively. The sensor failure can be stated in terms of a change in the mean m or a change in the variance σ^2. If a set of samples x_i, $i = 1, 2, \cdots, n$, is collected with a density function P describing each sample in the set, an overall likelihood ratio is given by

$$\gamma_n = \frac{P_1(\varepsilon_1 \mid H_1) \cdot P_1(\varepsilon_2 \mid H_1) \cdot P_1(\varepsilon_3 \mid H_1) \cdots P_1(\varepsilon_n \mid H_1)}{P_0(\varepsilon_1 \mid H_0) \cdot P_0(\varepsilon_2 \mid H_0) \cdot P_0(\varepsilon_3 \mid H_0) \cdots P_0(\varepsilon_n \mid H_0)}, \tag{11}$$

where H_1 represents a hypothesis that the sensor is degraded and H_0 represents a hypothesis that the sensor is normal. By taking the logarithm of the above equation and replacing the probability density functions in terms of residuals, means and variances, the log likelihood ratio (LLR, λ_n) can be written as the following recurrent form:

$$\lambda_n = \lambda_{n-1} + \ln\left(\frac{\sigma_0}{\sigma_1}\right) + \frac{(\varepsilon_n - m_0)^2}{2\sigma_0{}^2} - \frac{(\varepsilon_n - m_1)^2}{2\sigma_1{}^2}. \tag{12}$$

This is the form we use for deriving the sensor drift detection algorithm. For a normal sensor, the log likelihood ratio would decrease and eventually reach a specified bound A, a smaller value than zero. When the ratio reaches this bound, the decision is made that the sensor is normal, and then the ratio is initialized by setting it equal to zero. For a degraded sensor, the ratio would increase and eventually reach a specified bound B, a larger value than zero. When the ratio is equal to B, the decision is made that the sensor is degraded. The decision boundaries A and B are chosen by a false alarm probability α and a missed alarm probability β; $A = \ln\left(\frac{\beta}{1-\alpha}\right)$ and $B = \ln\left(\frac{1-\beta}{\alpha}\right)$.

5. Applications to Real Nuclear Plant Sensor Signals

The proposed sensor-monitoring algorithm that combines the above-mentioned methods is described in Fig. 3. The input-output data consist of a total of 11 different signals acquired from the startup data of the Yonggwang nuclear power plant unit 3&4. These data were standardized before they are presented to the fuzzy neural network. The acquired signals are S/G(steam generator) pressure (SP), S/G wide-range water level (WL), S/G narrow-range water level (NL), hot-leg temperature (HT), cold-leg temperature (CT), pressurizer pressure (PP), pressurizer temperature (PT), pressurizer water level (PL), feedwater temperature (FT), S/G temperature (ST), and excore neutron flux (NF). The proposed algorithm was applied to the steam generator water level, the hot-leg temperature, and the excore neutron flux sensors. Each signal consists of a total of 1400 discrete time points where the sampling period is 3 min. The fuzzy neural network was trained using one fifth of the given data in the training stage and was validated using the remaining data in the verification stage.

It is very important to accurately estimate the signals to determine a sensor failure. Table 1 shows the signal estimation and failure detection results of all the application cases and also, in this table the proposed method is compared with the other two methods for input selection; conventional genetic method and PCA method.

In the summary, from Table 1, it is shown that although PCA method uses the largest number of input signals, PCA method is the worst of the three methods. But the PCA method is the fastest. It is determined that the conventional genetic and proposed methods show similar performance as the input selection methods of fuzzy neural networks with application to sensor signal monitoring. However, the training time for the genetic method is about two times slower than that for the proposed method even though calculation time depends on the test cases and the relationship between input signals and output signal. From above simulations, it is shown that a fuzzy neural network with the proposed input selection and rule generation method actually estimates the relevant sensor signal using other sensor signals and SPRT failure detection algorithm detects the gradual degradation of sensors.

6. Conclusions

In this work, a fuzzy neural network with an automatic input selection and rule generation algorithm was developed for sensor monitoring. The real reduction of number of input signals is accomplished by the genetic algorithm which requires the substantial computational burden. Thus, the computational burden is reduced by using the correlation coefficient matrix which provides information on the relationship between input signals and an output signal. Also, since the number of fuzzy inference rules depends on that of selected inputs, the number of its rules is decided automatically according to the number of inputs.

The developed sensor monitoring algorithm was applied to the steam generator water level, hot-leg temperature and excore neutron flux sensors. The neuro-fuzzy inference system actually estimates the relevant output signal using other input signals. The SPRT decides fast whether a sensor is degraded or not by using the residuals between the measured signal and the estimated signal. The fuzzy neural network with the proposed input selection and rule generation algorithm is superior to the fuzzy neural network with the other two input selection algorithms (PCA method, genetic method).

References

1. B. R. Upadhyaya, et. al., Development and Testing of an Integrated Signal Validation System for Nuclear Power Plants, DOE/NE/37959-35, Sept. 1989.

2. A. Erbay and B. R. Upadhyaya, "A personal computer-based on-line signal validation system for nuclear power plants," *Nucl. Technology*, vol. 119, no. 1, pp. 63-75, July 1997.
3. R. M. Singer, K. C. Gross, J. P. Herzog, R. W. King, and S. W. Wegerich, "Model-based nuclear plant monitoring and fault detection: theoretical foundations," Proc. 9th Intl. Conf. on Intelligence Systems Applications to Power Systems, Seoul, Korea, 1997.
4. J. W. Hines, D. J. Wrest, and R. E. Uhrig, "Signal validation using an adaptive neural fuzzy inference system," *Nucl. Technology*, vol. 119, no. 2, pp. 181-193, 1997.
5. J. W. Heins, Andrei V. Gribok, Ibrahim Attieh, and Robert E. Uhrig, "Regularization method for inferential sensing in nuclear power plants," in Da Ruan (ed) *Fuzzy Systems and Soft Computing in Nuclear Engineering*, Springer-Verlag, Berlin Heidelberg New York, pp.285-314, 1999.
6. P. Fantoni, S. Figedy, and A. Racz, "A neuro-fuzzy model applied full range signal validation of PWR nuclear power plant data," FLINS-98, Antwerpen, Belgium, 1998.
7. X. Z. Wang and R. F. Li, "Combining conceptual clustering and principal component analysis for state space based process montoring," *Ind. Eng. Chem. Res*., vol. 38, no. 11, pp. 4345-4358, 1999.
8. J. Chen and J. Liu, "Mixture principal component analysis models for process monitoring," *Ind. Eng. Chem. Res*., vol. 38, no. 4, pp. 1478-1488, 1999.
9. Tomas Eklov, Per Martensson, and Ingemar Lundstorm, "Selection of variables for interpreting multivariate gas sensor data," *Analytica Chimica Acta*, vol. 381, pp. 221-232, 1999.
10. B. K. Lavine, A. Moores, and L. K. Helfend, "A genetic algorithm for pattern recognition analysis of pyrolysis gas chromatographic data," *J. Anal. Appl. Pyrolysis*, vol. 50, pp. 47-62, 1999.
11. K. W. Bauer Jr., S. G. Alsing, and K. A. Greene, "Feature screening using signal-to-noise ratios," *Neurocomputing*, vol. 31, pp. 29-44, 2000.
12. P. van de Laar and T. Heskes, "Input selection based on an ensemble," *Neurocomputing*, vol. 34, pp. 227-238, 2000.
13. N. R. Pal, "Soft computing for feature analysis," *Fuzzy Sets and Systems*, vol. 103, pp. 201-221, 1999.
14. A. Wald, *Sequential Analysis*, John Wiley & Sons, New York, 1947.
15. J.-S. Roger Jang, "ANFIS: adaptive-network-based fuzzy inference systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 23, no. 3, pp 665-685, May 1993.
16. T. Takagi and M. Sugeno, "Fuzzy identification of systems and its applications to modeling and control," *IEEE Trans. System, Man, Cybern.*, vol. 1, pp. 116-132, 1985.
17. D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison Wesley, Reading, Massachusetts, 1989.
18. M. Mitchell, *An Introduction to Genetic Algorithms*, The MIT Press, Cambridge, Massachusetts, 1996.

Table 1. Final results for four application cases [after 30 generations training for input selection (genetic and proposed methods only) and after 50 generations training for the fuzzy neural network].

Sensors		S/G water level			Hot-leg temperature			Excore neutron flux		
Methods		PCA	Genetic	Proposed	PCA	Genetic	Proposed	PCA	Genetic	Proposed
Training data	maximum error	0.09329	0.06318	0.04921	0.11693	0.05663	0.05376	0.13863	0.09034	0.09061
	sum of all squared errors	0.46998	0.13267	0.14105	0.55318	0.14979	0.16091	0.87693	0.33749	0.51287
	std. dev.	0.04097	0.02177	0.02244	0.04445	0.02313	0.02397	0.05596	0.03472	0.04280
	error (2σ)	0.08194	0.04354	0.04489	0.08890	0.04626	0.04795	0.11193	0.06944	0.08560
	fitness[1)]	0.59045	0.73874	0.76972	0.54547	0.75212	0.77307	0.49481	0.64098	0.62931
Verification data	maximum error	0.11415	0.06371	0.06006	0.20939	0.05540	0.05415	0.18291	0.09702	0.09220
	sum of all squared errors	1.92980	0.51955	0.56311	2.35390	0.59794	0.63719	3.64940	1.35500	2.03930
	std. dev.	0.04153	0.02155	0.02243	0.04586	0.02312	0.02386	0.05711	0.03480	0.04269
	error (2σ)	0.08305	0.04310	0.04487	0.09173	0.04623	0.04773	0.11422	0.06960	0.08538
Number of FNN inputs		6	5	5	6	5	4	6	7	7
Number of rules		-	4	5	-	4	4	-	4	4
Used signals		all signals	WL,HT PT,ST NF	WL,HT PL,FT NF	all signals	NL,PT PL,FT NF	SP,WL FT,NF	all signals	SP,NL HT,CT PP,PL FT	SP,NL HT,CT PP,PL FT
Failure detection time[2)]		52	62	126	141	126	90	114	108	107

1) The fitness value was calculated in combination with the genetic optimization of the fuzzy neural network parameters.
2) Sampling time steps after the beginning of purposely gradual degradation

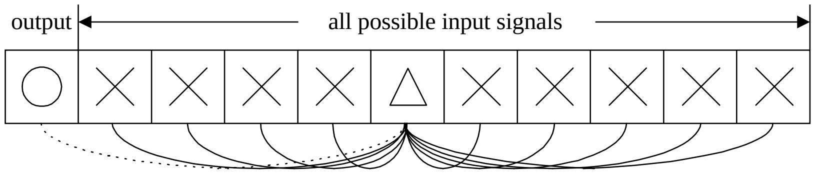

Fig. 1. A figure describing a process of input selection (an input selection part of chromosomes).

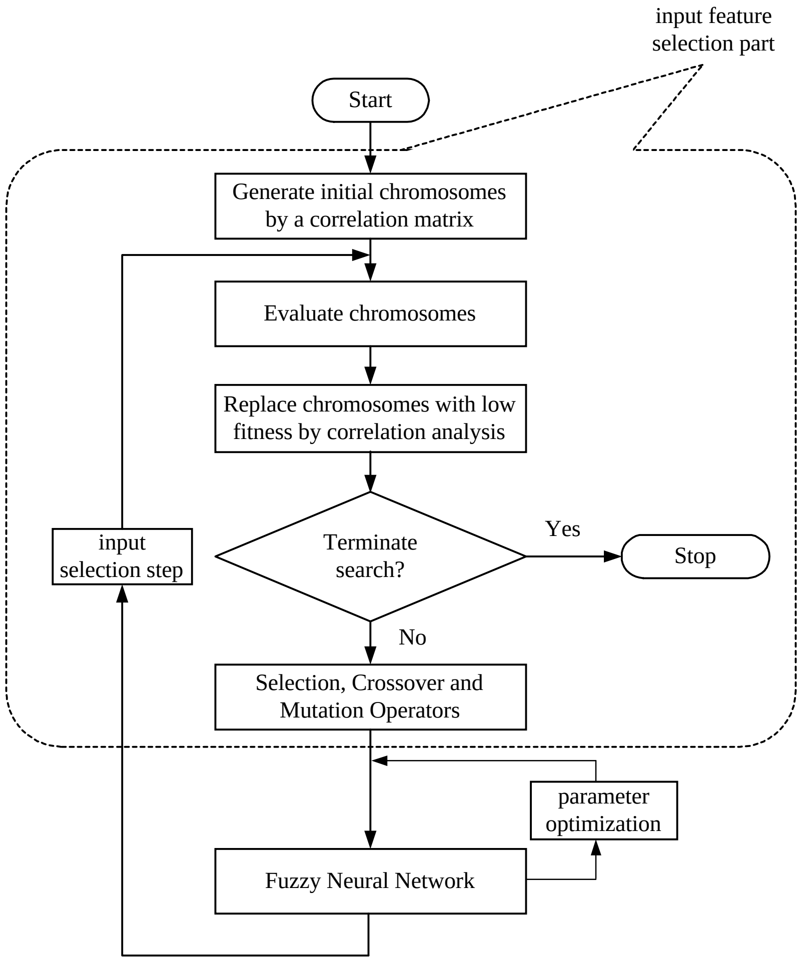

Fig. 2. Schematic diagram of the input selection algorithm.

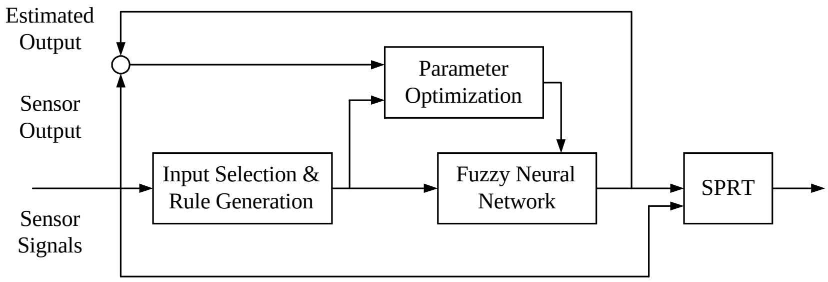

Figure. 3. Schematic diagram of the proposed sensor-monitoring algorithm.