Journal of Robotics and Control (JRC)
Volume 5, Issue 4, 2024
ISSN: 2715-5072, DOI: 10.18196/jrc.v5i4.22115

Design Intelligent Control Based on Fuzzy Neural Network and GA Algorithm for Prediction and Identification

Van-Truong Nguyen [1], Duc-Hung Pham [2*], Hoang-Nam Nguyen [3]
[1] Faculty of Mechatronics, SMAE, Hanoi University of Industry, Hanoi 10000, Vietnam
[2] Faculty of Electrical and Electronic Engineering, Hung Yen University of Technology and Education, Hung Yen, Vietnam
[3] Department of Applied Science and Technology, University of Science and Technology of Hanoi, Vietnam
Email: [1] nguyenvantruong@haui.edu.vn, [2] duchung.pham@utehy.edu.vn, [3] namnh.bi12-295@st.usth.edu.vn
*Corresponding Author

Abstract—**One of the central aspects in system identification and prediction is dealing with nonlinearity and uncertainties. This need involves the design of a novel method for achieving high efficiency and effectiveness, which is crucial for several applications. In this paper, a new intelligent control based on a hybrid fuzzy neural network (FNN) combined with a genetic algorithm (GA) is proposed for the prediction and identification of nonlinear systems. Two adaptations are considered in the proposed method: the backpropagation (BP) algorithm and the genetic algorithm method to correct various parameters in the neural network. Through adjustment, the proposed method not only achieves error convergence efficiently and quickly but also ensures continuous error reduction while avoiding the limitation of the regional optimal solution. Mackey-Glass differential delay and fuzzy neural system are utilized for system prediction and identification, respectively. Finally, the performance of the proposed method is justified through an application on a nonlinear system. Based on the findings, this paper proposed a hybrid strategy combining BP-GA and FNN where the outcome is greatly influenced by the balance of accuracy and computational efficiency.**

Keywords—Fuzzy System; Fuzzy Neural Network; Genetic Algorithm; Prediction; Identification.

I. INTRODUCTION

The advancement of control systems demands increasingly sophisticated solutions capable of addressing challenges like complex dynamics, uncertainties, and dynamic environments [1]. Prediction and identification of nonlinear systems have always caused great concerns in the control field [2] since the unpredictable nature of these systems makes it difficult to foresee their future behavior or identify their underlying dynamics. Moreover, the presence of uncertainties such as noise and unmodeled dynamics also introduce inaccuracies [3]-[4]. Identification and prediction of nonlinear systems are crucial in various applications due to their complexity and unpredictability: weather forecasting [5], finance [6], medical [7], and energy [8]. Therefore, issues accompanying the prediction and identification of nonlinear systems have gained great attention from academics.

In addition, several research on system identification have been published in recent years, across several applications, from power, and image processing, to energy and healthcare [9]-[15]. Recently, a deep neural network approach was proposed for robot tool dynamics identification for bilateral teleoperation [16]. Mao et al. introduced a Modified PSO algorithm on Recurrent Fuzzy Neural Network for system identification [17]. Kaheman developed a robust algorithm for parallel implicit sparse identification of nonlinear dynamics [18]. Nevertheless, the proposed methods are not optimal and may not perform at their peaks when applied to nonlinear systems due to the imprecision of system uncertainties and external disturbances, identification and prediction of nonlinear systems always pose a challenge to researchers. To counter these problems, Pham et al. introduced the wavelet interval type-2 Takagi -Sugeno-Kang hybrid controller for time-series prediction and chaotic synchronization [19].

The development of intelligent control systems [20]- [23] in prediction and identification using fuzzy neural network (FNN) [24]-[28] has seen a significant surge across various fields [29]-[45]. This is the hybridization of Takagi-Sugeno-Kang fuzzy systems (TSKFS) [46]-[53] and neural network (NN) [54]-[58], largely driven by the inspiration drawn from the human brain. This combination of the two artificial intelligence techniques aims to resolve the complexity of real-world systems. The FNN expands the ability of the conventional neural network and fuzzy logic system to take into account uncertainties. In essence, FNN leverages the power of NN, combined with various algorithms, to achieve desired outputs and create intelligent control systems capable of adapting and making optimal decisions in complex circumstances.

This research proposes a novel technique based on the strengths of FNN, gradient estimation, and optimization algorithms to create a brain-inspired learning process within a neural network. The approach utilizes a fuzzy neural network to predict future outcomes based on historical data of the system. Mackey-Glass [59]-[65] differential-delay equation will be applied to calculate a nonlinear system for employing FNN. This method aims to accelerate learning for intelligent control systems by employing both the back-propagation (BP) algorithm [66]-[75] for adjusting network parameters and genetic algorithm (GA) [76]-[80] for optimizing overall structure. The study emphasizes the significance of balancing accuracy and computational efficiency. In evaluations, the FNN considers performance indexes, and the presence of GA and BP algorithms helps to improve computation durations for various circumstances. The main contribution of the work is presented below:

1. A fuzzy neural network will be proposed to improve response to member function input uncertainty.
2. GA and BP are combined for enhanced learning rate and improved efficiency.
3. Using Mackey-Glass differential delay for the prediction of the nonlinear system.
4. An intelligent control is designed for the identification and prediction of nonlinear systems.

The content of this paper consists of the following parts: Section I introduces the problem to be solved, Section II presents the proposed fuzzy neural network and the adjustment of the controller parameters through the genetic algorithm, Section III illustrates the simulation and results, including the Mackey-Glass differential-delay for prediction and fuzzy neural system for identification of a nonlinear system. Finally, the conclusion of the paper is presented with the demonstration and combination of both BP and GA.

II. The Proposed Methodology

A. Fuzzy Neural Network

The human brain is composed of approximately 1011 nerve cells, which are interconnected to form a very complex neural network. When the human senses are stimulated by the outside world, signals are transmitted to the brain through nerve cells, and the brain will issue commands to the relevant receptors (effector) to respond (for example: let go immediately when the skin of your hand touches a hot object). This process often requires repeated training before you can make appropriate judgments and remember them in your brain cells. If the brain is damaged (such as a stroke patient), it will need to be rehabilitated to learn again. The operation of neural networks stems from this. Different algorithms are used to train neural networks so that the output of the neural network can achieve the results we require.

FNN effectively addresses challenges in system prediction and identification, particularly for complex and nonlinear systems. By combining fuzzy logic and neural networks, FNN leverages fuzzy sets and rules to handle imprecise information, while neural networks capture complex patterns and adapt to nonlinear dynamics. FNN's flexibility comes from modifiable fuzzy rules and learning algorithms that adjust parameters for improved accuracy. This synergy allows FNN to accurately and robustly model complex, nonlinear systems, making them ideal for practical applications where traditional methods may be inadequate.

In the control field, if you want to accurately analyze the relationship between input and output, the system must be modeled mathematically. However, actual systems are often complex and nonlinear, so how to simplify and linearize the system mathematically has become an important topic in control science. One advantage of neural networks is that they do not need to understand the mathematical model of the system. By directly replacing the system model with a neural network, the relationship between input and output can still be obtained. The membership function used in the neural network in this article is represented by the Gaussian function, as shown in Fig. 1, which is $exp\left[-\frac{(X-m)^2}{(v)^2}\right]$, where X is the input, m is the center of the Gaussian function, and v is the width of the Gaussian function. The neural network used in this paper is illustrated in Fig. 2. The FNN comprises four layers: input, Gaussian function, multiplication, and output. The input layer receives signals, with each node representing an input variable and passing these inputs unmodified to the next layer. The Gaussian function layer applies membership functions to the inputs, mapping them to values between 0 and 1, thereby fuzzifying the data to handle uncertainty. The multiplication layer combines these membership values to form the firing strength of each rule, with each node representing a rule in the fuzzy rule base. The final output layer aggregates the multiplication layer's results and converts them back into a crisp value through defuzzification, providing the network’s final prediction or classification.

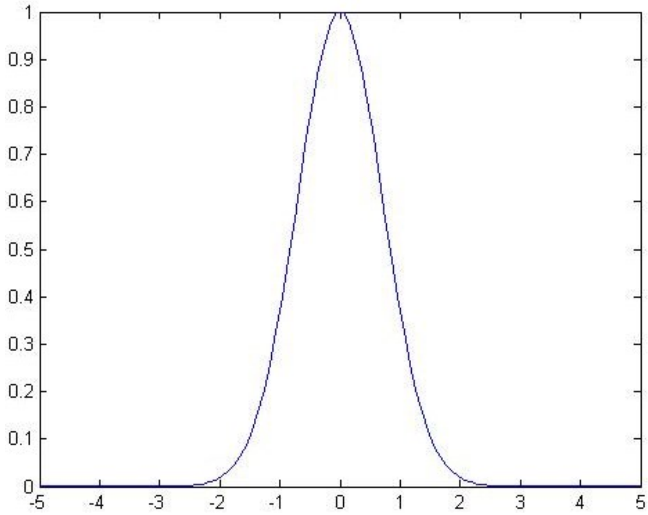

Fig. 1. Gaussian function (m = 1, v = 1)

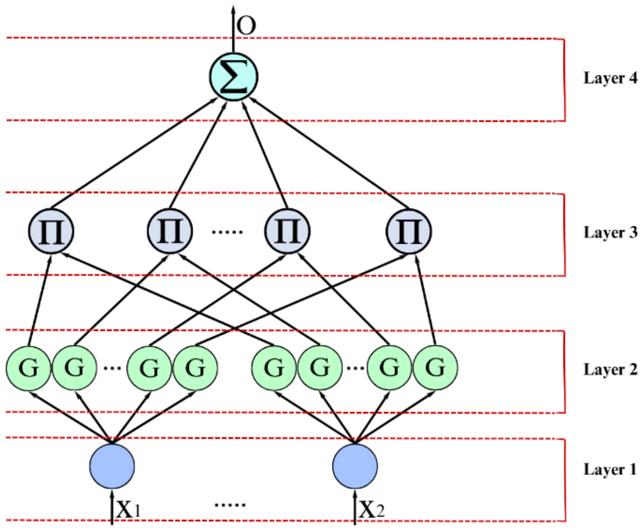

Fig. 2. The proposed neural network

Layer 1 is the input layer:

$$O_i^{(1)}(k) = X_i \tag{1}$$

Layer 2 is the Gaussian function layer:

$$O_{ij}^{(2)}(k) = exp\left[\frac{(X_i - m_{ij})^2}{(v_{ij})^2}\right] \quad (2)$$

Layer 3 is the multiplication layer:

$$O_j^{(3)}(k) = \prod_i^R O_{ij}^{(2)}(k) \quad (3)$$

Layer 4 is the output layer:

$$y_p = O^{(4)}(k) = \sum_{j=1}^R w_j O_j^{(3)}(k) \quad (4)$$

B. Adjustment Rules

The backward transfer algorithm, also known as the steepest slope method, ensures that the object being adjusted converges effectively. This algorithm operates in two main phases. The first phase involves introducing inputs from the input layer and propagating them through the hidden layers to the output layer, where the network output value is calculated with all weight values fixed. In the second phase, the backward transmission phase, the error is computed as the difference between the expected output and the actual network output. This error is then propagated back through the network to adjust the weight values. The learning rate is a crucial parameter that determines the step size during optimization, balancing convergence speed and stability. The following is the derivation of the backpropagation used in this paper:

The objective function used is determined by equation (5)

$$E(k) = \frac{1}{2}\left(\hat{y}(k) - y(k)\right)^2 = \frac{1}{2}\sum_j\left(\hat{y}(k) - O^4(k)\right)^2 \quad (5)$$

where $y(k)$ and $\hat{y}(k)$ are the actual output and expected output values of the network.

According to the guidance algorithm, adjusting the adjustment values of each weight or parameter can obtain equation (6)

$$W(k+1) = W(k) + \Delta W(k) = W(k) + \eta\left(-\frac{\partial E(k)}{\partial W}\right) \quad (6)$$

In $W = [m, v, \omega]$, then you can get

$$\frac{\partial E(k)}{\partial W} = \left[\frac{\partial E(k)}{\partial m_{ij}}, \frac{\partial E(k)}{\partial v_{ij}}, \frac{\partial E(k)}{\partial \omega_j}\right] \quad (7)$$

where $\frac{\partial E(k)}{\partial m_{ij}}, \frac{\partial E(k)}{\partial v_{ij}}, \frac{\partial E(k)}{\partial \omega_j}$ can be derived from formulas (8), (9) and (10), respectively.

$$\frac{\partial E(k)}{\partial m_{ij}} = e(k) \cdot \omega_j \cdot \underset{i}{\Pi} O_{ij}^{(2)} \cdot \frac{2(u_{ij}{}^{(2)}(k) - m_{ij})}{(v_{ij})^2} \quad (8)$$

$$\frac{\partial E(k)}{\partial v_{ij}} = e(k) \cdot \omega_j \cdot \underset{i}{\Pi} O_{ij}^{(2)} \cdot \frac{2(u_{ij}{}^{(2)}(k) - m_{ij})^2}{(v_{ij})^3} \quad (9)$$

$$\frac{\partial E(k)}{\partial \omega_j} = e(k) \cdot O_j^{(3)} \quad (10)$$

where $e(k)$ is the error calculated by comparing current output to the expected output.

C. Genetic Algorithm (GA)

In Darwin's theory of evolution, the concept of "natural selection and survival of the fittest" is mentioned, which became the fundamentals of the GA's principle. In nature, individuals with favorable traits are more likely to survive and reproduce, passing on their genes to future generations with stronger viability and adaptability. A population of candidate solutions (individuals) undergoes selection. Those solutions that perform better on a specific task (survival) are more likely to be chosen for reproduction. Each individual is called a chromosome (Chromosome), its gene value is generated randomly, and the set of chromosomes in each generation is called a population (Population). By competing with each other, the one that is more suitable for the environment has a higher fitness value. Chromosomes with higher fitness values can copy more offspring, and then select pairs among them to mate (Crossover) to produce the next generation, in order to produce the next generation with higher fitness. Furthermore, in order to avoid missing some useful information, mutation is added to prevent the incident although the rate is usually minor. In control system parameter tuning, GA efficiently searches the parameter space, balancing exploration and exploitation, and effectively handling complex, nonlinear, and optimization problems.

The fitness function is an indicator used to evaluate the adaptability of each individual in the population. When applied to genetic algorithms, the higher the fitness function value, the greater the individual's adaptability and competitiveness, which means the stronger, the greater the chance of survival and the greater the possibility of passing on one's genes to the next generation. In the process of evolution, the characteristics of the surviving individuals will change according to the selected fitness function. Therefore, the ecological environment of evolution can be controlled by setting different fitness functions. In order to apply genetic algorithms to practical problems, we must appropriately express the optimization problem and goals as fitness functions. For example, if we require the minimum value of function F, the fitness function can be defined as: $Fitness = 1/F(x)$ or $Fitness = e^{-F(x)}$, where x is the solution corresponding to the chromosome in the search space.

Replication plays the role of natural selection in genetic algorithms. It determines the probability of an individual being replicated based on the fitness value of each individual. The higher the fitness value, the greater the chance that individuals will be copied into the next generation, and the elimination of the unsuitable will increase the fitness value of the next generation population. The simplest and most widely used copying process is Roulette Wheel Selection. Its procedure is as follows:

1. First calculate the fitness function value f_i of each individual in the population, and the sum of all fitness function values f_{sum}.
2. Then randomly $[0, f_{min}]$ select a random number in this interval f_{rand}, and subtract the fitness function value of the individuals in the population in sequence:

$$f_{rand} = f_{rand} - f_i \quad (11)$$

3. Until minus f_{rand} is less than f_k or equal to zero, the k^{th} individual will be copied to the mating slot, and the selection is repeated until the number of individuals in the mating slot is the same as the number of individuals in the population.

Since an excessively large learning rate will cause the parameter adjustment of the second half of the algebra to oscillate, and a smaller learning rate will make the learning efficiency too slow, a dynamic learning rate adjustment method is added to make the learning effect more effective. A dynamic learning rate is an adaptive approach that adjusts the learning rate during the training process. Instead of using a fixed learning rate, it changes over time to improve training efficiency and performance, enhancing the model's ability to reach a global minimum and improve overall performance. This adaptability allows the learning rate to be higher at the beginning for faster convergence and lower later to fine-tune the model and avoid overshooting the optimal solution. The dynamic learning rate used in this paper is as follows:

$$\xi_k = \xi_{max} - \frac{\xi_{max} - \xi_{min}}{Total_{epoch}} \cdot epoch(k) \tag{12}$$

where ξ is the dynamic learning rate, ξ_{max} is the maximum learning rate, and ξ_{min} is the minimum learning rate. According to equation (12), the learning rate will decrease with the learning algebra. At the same time, the parameter adjustment of the second half of the algebra will not be in an oscillating state, nor will it cause the learning efficiency to be too slow.

III. SIMULATION AND RESULTS

A. Mackey-Glass Differential-Delay Prediction

In this paper, a neural network is used to predict a nonlinear system represented by equation (13) and illustrated in Fig. 3.

$$\frac{dx(t)}{dt} = \frac{ax(t-\tau)}{1+x^{10}(t-\tau)} - bx(t);\quad a = 0.2, b = 0.1, \tau = 17 \tag{13}$$

In the proposed model, $ref(t), ref(t-6), ref(t-12)$ and $ref(t-18)$ are used as inputs. After the neural network operation, $ref(t+6)$ information will be the output. The entire system architecture is shown in Fig. 4; at the beginning, the parameters in the neural network are random values, so each learning will produce different results; in the data part, a total of 1,000 pieces of data are generated. The first 500 transactions are used to train a neural network, and the last 500 transactions are used to verify the trained network.

All data is normalized, i.e. the size of the training data is scaled in equal proportions so that the maximum value in the training data is 1, which does not exceed the performance of the network. After the training is completed, the network output is also scaled proportionally to match the original data. Later, the maximum value of the network output may exceed 1 due to the modification of the network. However, the regularization of this data does not affect the quality of the network training, so the original design is maintained.

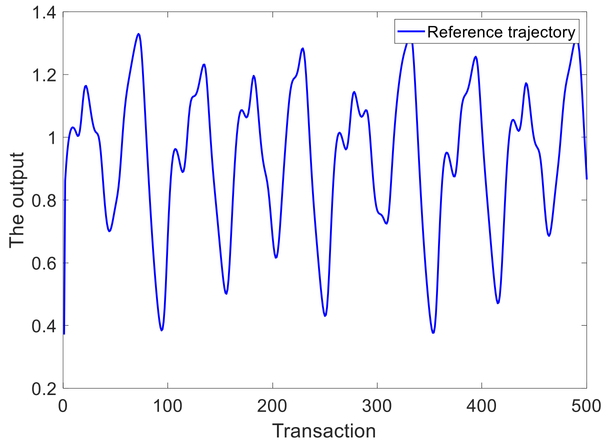

Fig. 3. Output of the nonlinear system

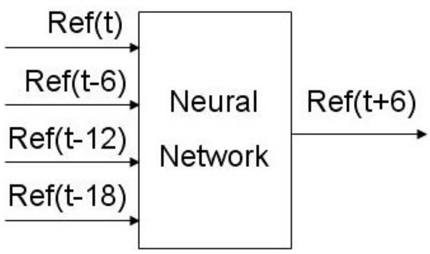

Fig. 4. Neural network system architecture

The initial layer of the network comprises four nodes, which serve as the entrances for input data. As the data progresses through the network, it encounters the second hidden layer, which is composed of 1200 nodes corresponding to each input. In alignment with the network architecture, the third layer also consists of 1200 nodes. Operating as a multiple-input single-output (MISO) network, the output layer culminates in a single node dedicated to producing the final output. The network undergoes 2500 generations of training, with a dynamic learning rate that ranges from a maximum value of 0.01 to a minimum value of 0.0000000001. Fig. 5 shows the use of backpropagation to adapt the neural network. Furthermore, Fig. 6 showcases the average root error of each generation, with the final generation exhibiting an average error of 6.0235e-003 for each data point.

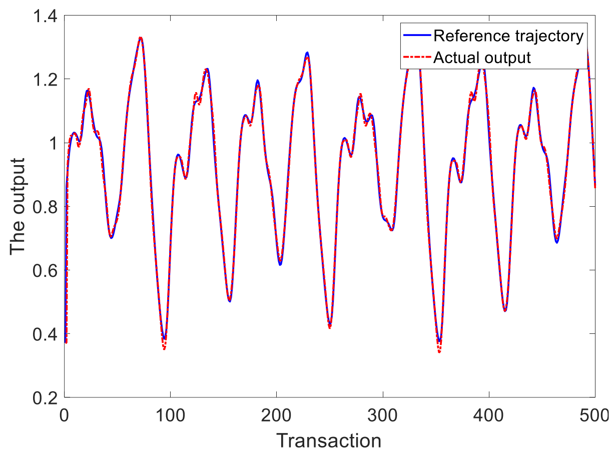

Fig. 5. Output diagram of neural network adjusted using back pass

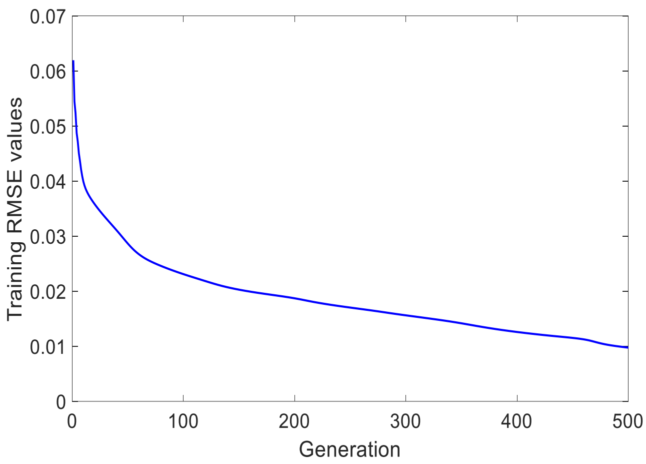

Fig. 6. Error mean value (BP) of each generation

The neural network architecture consists of multiple layers, with each layer containing a specific number of nodes. The initial layer consists of four nodes, corresponding to the four inputs of the network. Moving on to the second hidden layer, there are 50 nodes, each aligning with one of the inputs. This structural pattern extends to the subsequent layer, also containing 50 nodes. Transitioning to the outpt layer, the network's configuration operates under a MISO setup, indicating the presence of a single node for output processing. The genetic algorithm implemented within this network involves a total of 5000 generations, harnessing 500 distinct sets of chromosomes within the evolutionary process. Critical parameters have been set meticulously, with the mating rate established at 0.8, and the mutation rate at 0.2. In terms of the dynamic learning rate, the range spans from a maximum value of 0.001 to a minimum of 0.000001. The genetic algorithm's copying mechanism adopts a roulette wheel approach to accomplish the replication process efficiently. However, after executing the program, it was found that sampling was completely based on probability, resulting in a notable impact on the outcome of the program. When it is small, it cannot show the actual effect, that is, the better chromosomes cannot be guaranteed to continue to be passed on. Analysis of the comprehensive error map across generations revealed a pattern of oscillation akin to taking three steps forward followed by two steps back. Therefore, to effectively implement the roulette wheel concept, a substantial number of generations must be stipulated. It may take tens of thousands of generations before the overall effect will come out. Since this is really inefficient, for the copying part, the best chromosomes in each generation are copied (in this paper, the chromosomes with the smallest total error are chosen) to the next generation, which means that the good chromosomes are passed on and the best chromosomes are used. Following deduplication and random alterations to the entire group of contemporary chromosomes, the subsequent program execution indeed demonstrated an enhancement in efficiency compared to the initial implementation. Fig. 7 shows the modified genetic algorithm's output diagram to adjust the neural network. Meanwhile, Fig. 8 illustrates the root mean value of the average error of each generation, where the average error of each data in the last generation is 6.12195e-003.

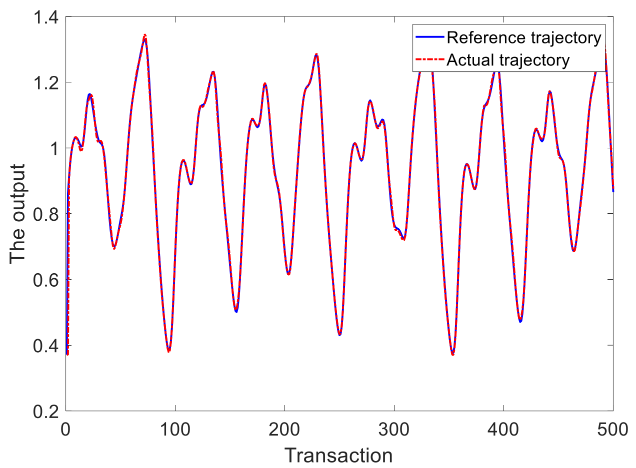

Fig. 7. Output diagram of the modified genetic algorithm to adjust the neural network

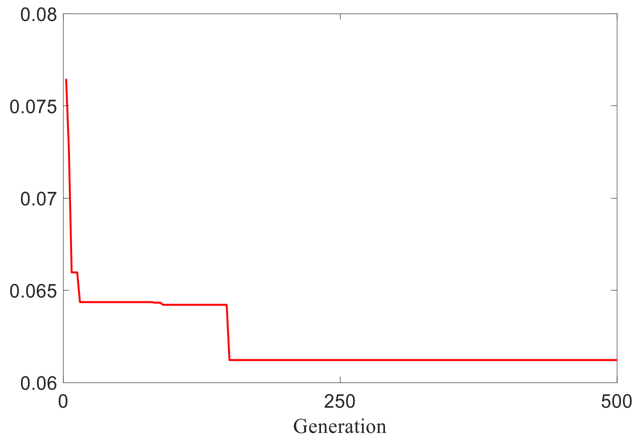

Fig. 8. Root mean error (GA) of each generation

Fig. 5 and Fig. 7 show the output diagrams for system identification using BP and GA, respectively, demonstrating their efficiency. Although the overall pattern appears to be identical, there are notable differences between the two. Despite the GA's capacity to find the optimal solution throughout the full domain, it happens to be outperformed by BP in terms of precision.

B. System Identification Using Fuzzy Neural System

From Fig. 9, the output of the nonlinear system is defined as:

$$y_p(k+1) = f(y_p(k), y_p(k-1), y_p(k-2), u(k), u(k-1)) \tag{14}$$

where Eq. (15) is the calculation function, y_p is the output signal.

$$f(x_1, x_2, x_3, x_4, x_5) = \frac{x_1 \cdot x_2 \cdot x_3 \cdot x_5 \cdot (x_3 - 1) + x_4}{1 + x_2^2 + x_3^2} \tag{15}$$

The training input $u(k)$ can be calculated as:

$$u(k) = 0.3 \cdot sin(\frac{\pi \cdot k}{25}) + 0.1 \cdot sin(\frac{\pi \cdot k}{32}) + 0.6 \cdot sin(\frac{\pi \cdot k}{10}) \tag{16}$$

where the testing input $u(k)$ is given by (17).

$$u(k) = \begin{cases} sin(\frac{\pi \cdot k}{25}) & 0 < k < 250 \\ 1.0 & 250 \le k < 500 \\ -1.0 & 500 \le k < 750 \\ 0.3 \cdot sin(\frac{\pi \cdot k}{25}) + 0.1 \cdot sin(\frac{\pi \cdot k}{32}) + 0.6 \cdot sin(\frac{\pi \cdot k}{10}) & 750 \le k < 1000 \end{cases} \tag{17}$$

In simulation, the learning rate chosen for BP are $\eta^{\omega} = 0.1$, $\eta^{m} = 0.1$, $\eta^{\sigma} = 0.1$, $\eta^{\theta} = 0.1$, and the total generation number is 100. Initial parameters $[m, v, \omega]$ for the neural network are chosen randomly between [-1,1]. Finally, the structure used for the FNN is 2-10-5-1, along with rule number of 5, and the parameter number equals to 35. Fig. 10 shows the training result with 1000 samples between the red solid line (the reference output) and the blue dashed line (the system identification using FNN).

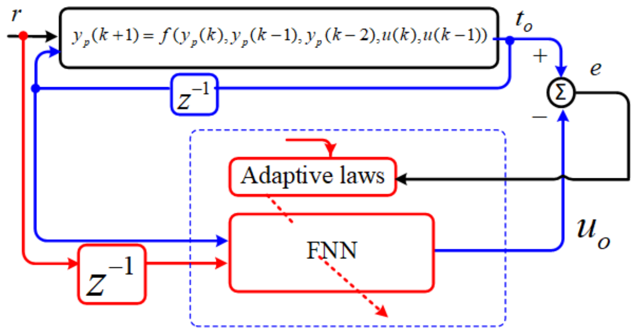

Fig. 9. The block diagram of the FNN for nonlinear systems identification

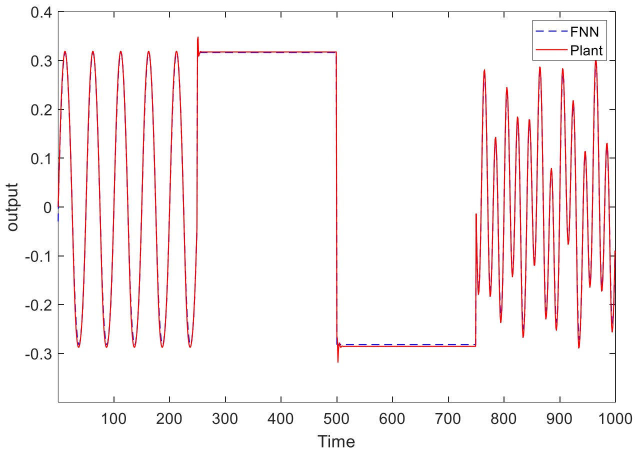

Fig. 10. Training result of nonlinear system identification

The data presented in Table I illustrates the efficiency of the proposed method by considering the relationship between the number of rules, Root Mean Square Error (RMSE), and computation time. As the number of rules increases from 2 to 20, witnessing a consistent decrease in RMSE, indicating an enhancement in accuracy. More specifically, the RMSE decreases from 0.0324 with 2 rules to 0.02347 with 20 rules, demonstrating that the method's precision improves with an increasing number of rules. However, this improvement in accuracy is accompanied by a substantial rise in computation time, which escalates from 0.393489 seconds for 2 rules to 2.532497 seconds for 20 rules. This pattern suggests a trade-off between achieving lower RMSE and higher computation time.

A more detailed analysis reveals that the initial increment in the number of rules from 2 to 5 results in a minor improvement in RMSE (from 0.0324 to 0.03213) alongside a moderate increase in computation time (from 0.393489 to 0.731853 seconds). More significant improvements in RMSE are observed when the number of rules is increased from 5 to 15, with the RMSE decreasing from 0.03213 to 0.02623 and computation time rising from 0.731853 to 1.941199 seconds. However, the improvement in RMSE begins to exhibit diminishing returns as the number of rules increases from 15 to 20, with RMSE only slightly decreasing from 0.02623 to 0.02347, while the computation time still increases significantly from 1.941199 to 2.532497 seconds.

These findings highlight the necessity of balancing accuracy and computational efficiency. The FNN takes into consideration both RMSE and computation time in performance evaluations, along with the presence of GA and BP algorithms to optimize computation times for varied scenarios.

TABLE I. EFFICIENCY OF THE PROPOSED METHOD

Rule number	RMSE	Times
2	0.0324	0.393489
5	0.03213	0.731853
10	0.03021	1.333072
15	0.02623	1.941199
20	0.02347	2.532497

IV. CONCLUSION

This paper presented an intelligent control technique based on fuzzy neural networks, with the back-propagation algorithm and the genetic algorithm used for adjustment. The results demonstrate the effectiveness of these algorithms in achieving error convergence efficiently and rapidly. From each step of the two algorithms, analysis of the average error value per generation reveals that the back-propagation algorithm consistently reduces the RMSE at 6.0235e-003 for each data point, indicating superior efficiency. Although the RMSE graph of the genetic algorithm is also decreasing at most, there exist several generations of errors where it is maintained at a fixed value of 6.12195e-003. The incident can be improved by the adjustment of the proportion between Crossover and Mutation. By increasing the Mutation's probability, it is more likely to achieve the desired output, with the drawback of increased computation time. However, from the situation, despite the genetic algorithm's capability to search for the best solution in the entire domain, it appears to be surpassed by the backpropagation algorithm in terms of efficiency. The genetic algorithm is inspired by natural selection and generational evolution, leading to the idea of combining its global optimization capabilities with the convergence efficiency of the back-propagation algorithm. This hybrid approach aims to ensure continuous error reduction while avoiding the local minima that can trap the back-propagation algorithm. The GA phase explores the global solution space, while the BP phase fine-tunes solutions for local optimality. Adaptive parameter tuning, parallel processing, and regularization techniques further enhance the method's performance and robustness. Overall, by integrating the back-propagation algorithm into the genetic algorithm's mating process and using mutation to escape local optima, the proposed GA-BP hybrid maintains high efficiency and effectiveness. As a result, the proposed approach can be implemented in a variety of applications in the prediction and identification of nonlinear systems for overcoming certain challenges. Finally, more advanced algorithms such as differential evolution (DE) or balancing composite motion optimization (BCMO) can be implemented for improved efficiency and optimization.

REFERENCES

[1] V.-T. Nguyen, “Non-negative adaptive mechanism-based sliding mode control for parallel manipulators with uncertainties,” *Computers, Materials & Continua*, vol. 74, no. 2, pp. 2771–2787, 2023, doi: 10.32604/cmc.2023.033460.

[2] V. -T. Nguyen, S. -F. Su, A. -T. Nguyen, and V. -T. Nguyen, "Adaptive Nonsingular Fast Terminal Sliding Mode Tracking Control for Parallel Manipulators with Uncertainties," *International Conference on System Science and Engineering (ICSSE)*, pp. 522-525, 2019, doi: 10.1109/ICSSE.2019.8823137.

[3] H. Liang, G. Liu, H. Zhang, and T. Huang, "Neural-Network-Based Event-Triggered Adaptive Control of Nonaffine Nonlinear Multiagent Systems With Dynamic Uncertainties," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 5, pp. 2239-2250, May 2021, doi: 10.1109/TNNLS.2020.3003950.

[4] D.-T. Viet, “Design the encoder inspection system based on the voltage wave form using finite impulse response (FIR) filter,” *International Journal of Modern Physics B*, vol. 34, p. 2040146, Aug. 2020, doi: 10.1142/s0217979220401463.

[5] A. Chattopadhyay, P. Hassanzadeh, and S. Pasha, “Predicting clustered weather patterns: A test case for applications of convolutional neural networks to spatio-temporal climate data,” *Scientific Reports*, vol. 10, no. 1, Jan. 2020, doi: 10.1038/s41598-020-57897-9.

[6] X. Li, J. Wang, and C. Yang, “Risk prediction in financial management of listed companies based on optimized BP Neural Network under Digital Economy,” *Neural Computing and Applications*, vol. 35, no. 3, pp. 2045–2058, May 2022, doi: 10.1007/s00521-022-07377-0.

[7] S. Kumari, D. Kumar, and M. Mittal, “An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier,” *International Journal of Cognitive Computing in Engineering*, vol. 2, pp. 40–46, Jun. 2021, doi: 10.1016/j.ijcce.2021.01.001.

[8] D. Yu, Y. Wang, H. Liu, K. Jermsittiparsert, and N. Razmjooy, “System identification of PEM fuel cells using an improved Elman Neural Network and a new hybrid optimization algorithm,” *Energy Reports*, vol. 5, pp. 1365–1374, Nov. 2019, doi: 10.1016/j.egyr.2019.09.039.

[9] S. Hou, J. Fei, C. Chen, and Y. Chu, "Finite-Time Adaptive Fuzzy-Neural-Network Control of Active Power Filter," in *IEEE Transactions on Power Electronics*, vol. 34, no. 10, pp. 10298-10313, Oct. 2019, doi: 10.1109/TPEL.2019.2893618.

[10] T. T. Huynh *et al*., “4-D Memristive Chaotic Systems-Based Audio Secure Communication Using Dual-Function-Link Fuzzy Brain Emotional Controller,” *Int. J. Fuzzy Syst*., vol. 24, no. 6, pp. 2946–2968, 2022, doi: 10.1007/s40815-022-01312-0.

[11] Q. D. Nguyen, V. N. Giap, D. -H. Pham, and S. -C. Huang, "Fast Speed Convergent Stability of T-S Fuzzy Sliding-Mode Control and Disturbance Observer for a Secure Communication of Chaos-Based System," in *IEEE Access*, vol. 10, pp. 95781-95790, 2022, doi: 10.1109/ACCESS.2022.3205027.

[12] C. -M. Lin, D. -H. Pham, and T. -T. Huynh, "Synchronization of Chaotic System Using a Brain-Imitated Neural Network Controller and Its Applications for Secure Communications," in *IEEE Access*, vol. 9, pp. 75923-75944, 2021, doi: 10.1109/ACCESS.2021.3080696.

[13] E. M. Rentería-Vargas *et al*., "Neural Network-Based Identification of a PSA Process for Production and Purification of Bioethanol," in *IEEE Access*, vol. 10, pp. 27771-27782, 2022, doi: 10.1109/ACCESS.2022.3155449.

[14] D. H. Pham, T. T. Huynh, and C. M. Lin, “Fault-Tolerant Control for Robotic Systems Using a Wavelet Type-2 Fuzzy Brain Emotional Learning Controller and a TOPSIS-Based Self-organizing Algorithm,” *Int. J. Fuzzy Syst*., vol. 25, pp. 1727–1741, 2023, doi: 10.1007/s40815-023-01516-y.

[15] T. -T. Huynh, D. -H. Pham, and N. T. Son, "A Brain Emotional Fuzzy CMAC for Chaos Synchronization and Communication," *2022 6th International Conference on Green Technology and Sustainable Development (GTSD)*, pp. 131-134, 2022, doi: 10.1109/GTSD54989.2022.9989169.

[16] H. Su, W. Qi, C. Yang, J. Sandoval, G. Ferrigno, and E. D. Momi, "Deep Neural Network Approach in Robot Tool Dynamics Identification for Bilateral Teleoperation," in *IEEE Robotics and Automation Letters*, vol. 5, no. 2, pp. 2943-2949, April 2020, doi: 10.1109/LRA.2020.2974445.

[17] W. L. Mao, C. W. Hung, and H. Y. Huang, “Modified PSO algorithm on Recurrent Fuzzy Neural Network for system identification,” *Intelligent Automation and Soft Computing*, vol. 25, no. 2, 2019, doi: 10.31209/2019.100000094.

[18] K. Kaheman, J. N. Kutz, and S. L. Brunton, “Sindy-pi: A robust algorithm for parallel implicit sparse identification of nonlinear dynamics,” *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Sciences*, vol. 476, no. 2242, Oct. 2020, doi: 10.1098/rspa.2020.0279.

[19] D. -H. Pham, C. -M. Lin, V. N. Giap, T. -T. Huynh, and H. -Y. Cho, "Wavelet Interval Type-2 Takagi-Kang-Sugeno Hybrid Controller for Time-Series Prediction and Chaotic Synchronization," in *IEEE Access*, vol. 10, pp. 104313-104327, 2022, doi: 10.1109/ACCESS.2022.3210260.

[20] H. Mo and G. Farid, “Nonlinear and adaptive intelligent control techniques for QUADROTOR UAV – A survey,” *Asian Journal of Control*, vol. 21, no. 2, pp. 989–1008, Apr. 2018, doi: 10.1002/asjc.1758.

[21] P. K. Muthusamy, M. Garratt, H. Pota, and R. Muthusamy, "Real-Time Adaptive Intelligent Control System for Quadcopter Unmanned Aerial Vehicles With Payload Uncertainties," in *IEEE Transactions on Industrial Electronics*, vol. 69, no. 2, pp. 1641-1653, Feb. 2022, doi: 10.1109/TIE.2021.3055170.

[22] M. Mao *et al*., “Classification and summarization of solar photovoltaic MPPT techniques: A review based on traditional and Intelligent Control Strategies,” *Energy Reports*, vol. 6, pp. 1312–1327, Nov. 2020, doi: 10.1016/j.egyr.2020.05.013.

[23] A. Rodríguez-Molina, E. Mezura-Montes, M. G. Villarreal-Cervantes, and M. Aldape-Pérez, “Multi-objective meta-heuristic optimization in intelligent control: A survey on the controller tuning problem,” *Applied Soft Computing*, vol. 93, p. 106342, Aug. 2020, doi: 10.1016/j.asoc.2020.106342.

[24] J. Wang, Q. Chang, T. Gao, K. Zhang, and N. R. Pal, “Sensitivity analysis of Takagi–Sugeno fuzzy neural network,” *Information Sciences*, vol. 582, pp. 725–749, Jan. 2022, doi: 10.1016/j.ins.2021.10.037.

[25] K. Bhalla, D. Koundal, B. Sharma, Y.-C. Hu, and A. Zaguia, “A fuzzy convolutional neural network for enhancing multi-focus image fusion,” *Journal of Visual Communication and Image Representation*, vol. 84, p. 103485, Apr. 2022, doi: 10.1016/j.jvcir.2022.103485.

[26] V. -T. Nguyen, C. -Y. Lin, S. -F. Su, N. -Q. Nguyen, and Q. -V. Tran, "Adaptive PD networks tracking control with full-state constraints for redundant parallel manipulators," 2017 Joint 17th World Congress of *International Fuzzy Systems Association and 9th International Conference on Soft Computing and Intelligent Systems (IFSA-SCIS)*, pp. 1-5, 2017, doi: 10.1109/IFSA-SCIS.2017.8023289.

[27] V. -T. Nguyen, X. T. Vu, and H. B. Giap, "Adaptive Neural Network Hierarchical Sliding-Mode Control for Pendubot Based Genetic Algorithm Optimization," *Intelligent Systems and Networks. Lecture Notes in Networks and Systems*, vol. 471, 2022, doi: 10.1007/978-981-19-3394-3_66.

[28] Z. Wang and J. Fei, "Fractional-Order Terminal Sliding-Mode Control Using Self-Evolving Recurrent Chebyshev Fuzzy Neural Network for MEMS Gyroscope," in *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 7, pp. 2747-2758, July 2022, doi: 10.1109/TFUZZ.2021.3094717.

[29] V. -T. Nguyen, A. T. Nguyen, V. T. Nguyen, and H. A. Bui, "A Real-Time Human Tracking System Using Convolutional Neural Network and Particle Filter," *Intelligent Systems and Networks: Selected Articles from ICISN 2021*, pp. 411-417, 2021.

[30] K. Zheng, Q. Zhang, Y. Hu, and B. Wu, “Design of fuzzy system-fuzzy neural network-backstepping control for Complex Robot System,” *Information Sciences*, vol. 546, pp. 1230–1255, Feb. 2021, doi:10.1016/j.ins.2020.08.110.

[31] J. Fei, Z. Wang, X. Liang, Z. Feng and Y. Xue, "Fractional Sliding-Mode Control for Microgyroscope Based on Multilayer Recurrent Fuzzy Neural Network," in *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 6, pp. 1712-1721, June 2022, doi: 10.1109/TFUZZ.2021.3064704.

[32] H. Hewamalage, C. Bergmeir, and K. Bandara, “Recurrent neural networks for time series forecasting: Current status and future directions,” *International Journal of Forecasting*, vol. 37, no. 1, pp. 388–427, Jan. 2021, doi: 10.1016/j.ijforecast.2020.06.008.

[33] J. Wang and T. Kumbasar, "Parameter optimization of interval Type-2 fuzzy neural networks based on PSO and BBBC methods," in *IEEE/CAA Journal of Automatica Sinica*, vol. 6, no. 1, pp. 247-257, 2019, doi: 10.1109/JAS.2019.1911348.

[34] R.-T. Wu and M. R. Jahanshahi, “Deep Convolutional Neural Network for Structural Dynamic Response Estimation and system identification,” *Journal of Engineering Mechanics*, vol. 145, no. 1, Jan. 2019, doi: 10.1061/(asce)em.1943-7889.0001556.

[35] P. Kiran, B. D. Parameshachari, J. Yashwanth, and K. N. Bharath, “Offline signature recognition using image processing techniques and back propagation neuron network system,” *SN Computer Science*, vol. 2, no. 3, Apr. 2021, doi: 10.1007/s42979-021-00591-y.

[36] H. Nasiri and M. M. Ebadzadeh, “MFRFNN: Multi-functional recurrent fuzzy neural network for chaotic time series prediction,” *Neurocomputing*, vol. 507, pp. 292–310, Oct. 2022, doi: 10.1016/j.neucom.2022.08.032.

[37] S. Nobukawa and H. Nishimura, “Synchronization of Chaos in Neural Systems,” *Frontiers in Applied Mathematics and Statistics*, vol. 6, Jun. 2020, doi: 10.3389/fams.2020.00019.

[38] X. Liu, D. Zhang, J. Zhang, T. Zhang, and H. Zhu, “A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm,” *Cluster Computing*, vol. 24, no. 3, pp. 1901–1915, Jan. 2021, doi: 10.1007/s10586-021-03235-1.

[39] A. A. Kuz’menko, “Forced sliding mode control for chaotic systems synchronization,” *Nonlinear Dynamics*, vol. 109, no. 3, pp. 1763–1775, May 2022, doi: 10.1007/s11071-022-07552-x.

[40] M. Sangiorgio and F. Dercole, “Robustness of LSTM neural networks for multi-step forecasting of Chaotic Time Series,” *Chaos, Solitons & Fractals*, vol. 139, p. 110045, Oct. 2020, doi: 10.1016/j.chaos.2020.110045.

[41] R. Jafari, M. A. Contreras, W. Yu, and A. Gegov, “Applications of fuzzy logic, artificial neural network and neuro-fuzzy in Industrial Engineering,” *Industrial and Robotic Systems*, pp. 9–14, 2020, doi: 10.1007/978-3-030-45402-9_2.

[42] X. Xu, D. Cao, Y. Zhou, and J. Gao, “Application of neural network algorithm in fault diagnosis of mechanical intelligence,” *Mechanical Systems and Signal Processing*, vol. 141, p. 106625, Jul. 2020, doi: 10.1016/j.ymssp.2020.106625.

[43] T. R. Khalifa, A. M. El-Nagar, M. A. El-Brawany, E. A. El-Araby, and M. El-Bardini, “A novel Hammerstein model for nonlinear networked systems based on an interval type-2 fuzzy takagi–sugeno–kang system,” *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 2, pp. 275–285, Feb. 2021, doi: 10.1109/tfuzz.2020.3007460.

[44] X. Gu, C. Zhang, and T. Ni, "Feature Selection and Rule Generation Integrated Learning for Takagi-Sugeno-Kang Fuzzy System and its Application in Medical Data Classification," in *IEEE Access*, vol. 7, pp. 169029-169037, 2019, doi: 10.1109/ACCESS.2019.2954707.

[45] M. Chen, H. Wang, and X. Liu, "Adaptive Fuzzy Practical Fixed-Time Tracking Control of Nonlinear Systems," in *IEEE Transactions on Fuzzy Systems*, vol. 29, no. 3, pp. 664-673, March 2021, doi: 10.1109/TFUZZ.2019.2959972.

[46] D. -H. Pham, C. -M. Lin, V. N. Giap, V. -P. Vu, and H. -Y. Cho, "Design of Missile Guidance Law Using Takagi-Sugeno-Kang (TSK) Elliptic Type-2 Fuzzy Brain Imitated Neural Networks," in *IEEE Access*, vol. 11, pp. 53687-53702, 2023, doi: 10.1109/ACCESS.2023.3277537.

[47] O. Shaheen, A. M. El-Nagar, M. El-Bardini, and N. M. El-Rabaie, “Stable adaptive probabilistic takagi–sugeno–kang fuzzy controller for dynamic systems with uncertainties,” *ISA Transactions*, vol. 98, pp. 271–283, Mar. 2020, doi: 10.1016/j.isatra.2019.08.035.

[48] Y. Zhang *et al*., “Takagi-Sugeno-Kang Fuzzy System Fusion: A survey at hierarchical, wide and stacked levels,” *Information Fusion*, vol. 101, p. 101977, Jan. 2024, doi: 10.1016/j.inffus.2023.101977.

[49] J. Li and Q. Liu, “Forecasting of short-term photovoltaic power generation using combined interval type-2 takagi-sugeno-kang fuzzy systems,” *International Journal of Electrical Power & Energy Systems*, vol. 140, p. 108002, Sep. 2022, doi: 10.1016/j.ijepes.2022.108002.

[50] S. Gu, Y. Chou, J. Zhou, Z. Jiang, and M. Lu, “Takagi–sugeno–kang fuzzy clustering by direct fuzzy inference on Fuzzy Rules,” *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 8, no. 2, pp. 1264–1279, Apr. 2024, doi: 10.1109/tetci.2023.3336537.

[51] Y. Chen, J. Yang, and C. Li, “Design of Takagi Sugeno Kang type interval type-2 fuzzy logic systems optimized with hybrid algorithms,” *International Journal of Fuzzy Systems*, vol. 25, no. 2, pp. 868–879, Oct. 2022, doi: 10.1007/s40815-022-01410-z.

[52] Q. Lou, Z. Deng, Z. Xiao, K. -S. Choi, and S. Wang, "Multilabel Takagi-Sugeno-Kang Fuzzy System," in *IEEE Transactions on Fuzzy Systems*, vol. 30, no. 9, pp. 3410-3425, Sept. 2022, doi: 10.1109/TFUZZ.2021.3115967.

[53] Z. Hu *et al*., "Uncertainty Modeling for Multicenter Autism Spectrum Disorder Classification Using Takagi–Sugeno–Kang Fuzzy Systems," in *IEEE Transactions on Cognitive and Developmental Systems*, vol. 14, no. 2, pp. 730-739, June 2022, doi: 10.1109/TCDS.2021.3073368.

[54] V. T. Nguyen and H. A. Bui, "A real-time defect detection in printed circuit boards applying deep learning," *EUREKA: Physics and Engineering*, vol. 2, pp. 143-153, doi: 10.21303/2461-4262.2022.002127.

[55] J. Li, X. Xiang, and S. Yang, “Robust adaptive neural network control for dynamic positioning of marine vessels with prescribed performance under model uncertainties and input saturation,” *Neurocomputing*, vol. 484, pp. 1–12, May 2022, doi: 10.1016/j.neucom.2021.03.136.

[56] Y. Liu and Q. Zhu, "Event-Triggered Adaptive Neural Network Control for Stochastic Nonlinear Systems With State Constraints and Time-Varying Delays," in *IEEE Transactions on Neural Networks and Learning Systems*, vol. 34, no. 4, pp. 1932-1944, April 2023, doi: 10.1109/TNNLS.2021.3105681.

[57] V. Nguyen, S. Su, N. Wang, and W. Sun, “Adaptive finite-time neural network control for redundant parallel manipulators,” *Asian Journal of Control*, vol. 22, no. 6, pp. 2534–2542, May 2019, doi: 10.1002/asjc.2120.

[58] C. -M. Lin, D. -H. Pham, and T. -T. Huynh, "Synchronization of Chaotic System Using a Brain-Imitated Neural Network Controller and Its Applications for Secure Communications," in *IEEE Access*, vol. 9, pp. 75923-75944, 2021, doi: 10.1109/ACCESS.2021.3080696.

[59] M. Akhmet, K. Başkan, and C. Yeşil, “Revealing chaos synchronization below the threshold in coupled Mackey–Glass Systems,” *Mathematics*, vol. 11, no. 14, p. 3197, Jul. 2023, doi: 10.3390/math11143197.

[60] H. Carreon-Ortiz, F. Valdez, P. Melin, and O. Castillo, “Architecture optimization of a non-linear autoregressive neural networks for Mackey-glass time series prediction using discrete mycorrhiza optimization algorithm,” *Micromachines*, vol. 14, no. 1, p. 149, Jan. 2023, doi: 10.3390/mi14010149.

[61] J. P. Tarigo, C. Stari, C. Cabeza, and A. C. Marti, “Characterizing multistability regions in the parameter space of the mackey–glass delayed system,” *The European Physical Journal Special Topics*, vol. 231, no. 3, pp. 273–281, Dec. 2021, doi: 10.1140/epjs/s11734-021-00353-0.

[62] H. Carreon-Ortiz, F. Valdez, P. Melin, and O. Castillo, “Architecture optimization of a non-linear autoregressive neural networks for Mackey-glass time series prediction using discrete mycorrhiza optimization algorithm,” *Micromachines*, vol. 14, no. 1, p. 149, Jan. 2023, doi: 10.3390/mi14010149.

[63] M. Akhmet, K. Başkan, and C. Yeşil, “Revealing chaos synchronization below the threshold in coupled Mackey–Glass Systems,” *Mathematics*, vol. 11, no. 14, p. 3197, Jul. 2023, doi: 10.3390/math11143197.

[64] H. Carreon-Ortiz, F. Valdez, P. Melin, and O. Castillo, “Architecture optimization of a non-linear autoregressive neural networks for Mackey-glass time series prediction using discrete mycorrhiza

optimization algorithm," *Micromachines*, vol. 14, no. 1, p. 149, Jan. 2023, doi: 10.3390/mi14010149.

[65] M. Furkan Gül, S. Arslan, B. Mühendisliği, and Y. Mühendisliği, "Mackey-Glass Time Series Prediction with Immune Plasma Programming," *2023 31st Signal Processing and Communications Applications Conference (SIU)*, pp. 1-4, 2023, doi: 10.1109/SIU59756.2023.10223926.

[66] C. Lee, S. S. Sarwar, P. Panda, G. Srinivasan, and K. Roy, "Enabling spike-based backpropagation for training deep neural network architectures," *Frontiers in Neuroscience*, vol. 14, Feb. 2020, doi: 10.3389/fnins.2020.00119.

[67] T. P. Lillicrap, A. Santoro, L. Marris, C. J. Akerman, and G. Hinton, "Backpropagation and the brain," *Nature Reviews Neuroscience*, vol. 21, no. 6, pp. 335–346, Apr. 2020, doi: 10.1038/s41583-020-0277-3.

[68] L. G. Wright *et al*., "Deep physical neural networks trained with backpropagation," *Nature*, vol. 601, no. 7894, pp. 549–555, Jan. 2022, doi: 10.1038/s41586-021-04223-6.

[69] C. Chen and G. X. Gu, "Generative deep neural networks for inverse materials design using backpropagation and active learning," *Advanced Science*, vol. 7, no. 5, Jan. 2020, doi: 10.1002/advs.201902607.

[70] P. R. Vlachas *et al*., "Backpropagation algorithms and Reservoir Computing in recurrent neural networks for the forecasting of complex spatiotemporal dynamics," *Neural Networks*, vol. 126, pp. 191–217, Jun. 2020, doi: 10.1016/j.neunet.2020.02.016.

[71] T. C. Wunderlich and C. Pehle, "Event-based backpropagation can compute exact gradients for spiking neural networks," *Scientific Reports*, vol. 11, no. 1, Jun. 2021, doi: 10.1038/s41598-021-91786-z.

[72] J. Zhao, H. Nguyen, T. Nguyen-Thoi, P. G. Asteris, and J. Zhou, "Improved Levenberg–Marquardt backpropagation neural network by particle swarm and whale optimization algorithms to predict the deflection of RC beams," *Engineering with Computers*, vol. 38, no. S5, pp. 3847–3869, Jan. 2021, doi: 10.1007/s00366-020-01267-6.

[73] J. Jiang *et al*., "Analysis and prediction of printable bridge length in fused deposition modelling based on back Propagation Neural Network," *Virtual and Physical Prototyping*, vol. 14, no. 3, pp. 253–266, Feb. 2019, doi: 10.1080/17452759.2019.1576010.

[74] G. Wang, O. I. Awad, S. Liu, S. Shuai, and Z. Wang, "NOx emissions prediction based on Mutual Information and back propagation neural network using correlation quantitative analysis," *Energy*, vol. 198, p. 117286, May 2020, doi: 10.1016/j.energy.2020.117286.

[75] W. Sun and C. Huang, "A carbon price prediction model based on secondary decomposition algorithm and optimized back propagation neural network," *Journal of Cleaner Production*, vol. 243, p. 118671, Jan. 2020, doi: 10.1016/j.jclepro.2019.118671.

[76] P. C. Jennings, S. Lysgaard, J. S. Hummelshøj, T. Vegge, and T. Bligaard, "Genetic algorithms for Computational Materials Discovery Accelerated by machine learning," *npj Computational Materials*, vol. 5, no. 1, Apr. 2019, doi: 10.1038/s41524-019-0181-4.

[77] K. M. Hamdia, X. Zhuang, and T. Rabczuk, "An efficient optimization approach for designing machine learning models based on genetic algorithm," *Neural Computing and Applications*, vol. 33, no. 6, pp. 1923–1933, Jun. 2020, doi: 10.1007/s00521-020-05035-x.

[78] G. L. Guo *et al*., "Decoupled Sliding Mode Control of Underactuated Nonlinear Systems Using a Fuzzy Brain Emotional Cerebellar Model Control System," *Int. J. Fuzzy Syst*., vol. 25, pp. 15–28, 2023, doi: 10.1007/s40815-022-01378-w.

[79] Y. Wang, Z. Zhu, A. Sha, and W. Hao, "Low cycle fatigue life prediction of titanium alloy using genetic algorithm-optimized BP Artificial Neural Network," *International Journal of Fatigue*, vol. 172, p. 107609, Jul. 2023, doi: 10.1016/j.ijfatigue.2023.107609.

[80] H. B. Nguyen *et al*., "Fuzzy Hybrid Neural Network Control for Uncertainty Nonlinear Systems Based on Enhancement Search Algorithm," *Int. J. Fuzzy Syst*., vol. 24, pp. 3384–3402, 2022.