

# Integrated genetic algorithm and back-propagation neural network approach for fault detection and prediction in the Tennessee Eastman process

Qi Ji, Azura Che Soh<sup>#</sup>, Ribhan Zafira Abdul Rahman,  
Raja Kamil Raja Ahmad, Siti Anom Ahmad

Control System and Signal Processing Research Group (CSSP), Department of Electrical and Electronic Engineering,  
Faculty of Engineering, Universiti Putra Malaysia, 43400, Serdang, Selangor, MALAYSIA.  
<sup>#</sup>Corresponding author. azuracs@upm.edu.my; Tel: +603-97696322; Fax: +603-89466327.

**ABSTRACT** The current era emphasizes the application of intelligent algorithms for automating industrial processes. Among these, fault detection and prediction take precedence. This research introduces fault detection method that combine the genetic algorithm with back-propagation neural networks (GA-BPNN). The integration of these two methods, GA-BPNN, enhances their effectiveness. GA-BPNN effectively addresses the challenges of poor convergence in traditional genetic algorithms (GAs) and the difficulty of accurately defining parameters in back-propagation neural networks (BPNNs). In this approach, BPNN serves as the foundational framework, while GA dynamically optimizes various parameters within the BPNN. The proposed method GA-BPNN exhibits excellent parameter self-regulation ability and can adapt to various training conditions. This optimization process enhances precision and speed, making GA-BPNN a powerful and efficient solution. The Tennessee Eastman (TE) chemical process is employed as the simulated domain to validate the efficacy and superiority of the GA-BPNN approach in process control. The simulation results indicate that the GA-BPNN method outperforms the traditional BPNN. Additionally, the proposed method demonstrates excellent self-regulation ability, automatically optimizing parameters, and ensuring outstanding adaptability and learning ability in various situations.

**KEYWORDS:** Back-propagation neural network; Fault detection; Genetic algorithm; Optimization; Prediction

Received 20 October 2023 Revised 10 December 2023 Accepted 14 December 2023 In press 14 December 2023 Online 31 December 2023

© Transactions on Science and Technology

Original Article

## INTRODUCTION

In chemical industrial processes, a fault is defined as any abnormal deviation from the normal operating conditions. Faults are a concern because even small faults in a complex industrial system can initiate a series of events that result in loss of efficiency and reliability. As a result, there is a need for techniques to improve the process's reliability and up-time. Effective fault detection and identification is important for monitoring components for making appropriate maintenance decisions. First, fault detection determines whether a fault has occurred in the system also characterized as anomaly detection in other applications. Then fault identification determines which observation variables are most relevant to diagnosing the fault detected, thereby helping operators to focus on specific subsystems. Systems that can accurately and promptly detect and identify faults can more effectively inform operators and engineers and significantly reduce the effort and time to recover the system.

The Tennessee Eastman (TE) process is a chemical simulative model based on the actual chemical production process (Downs & Vogel, 1993). The data generated by the TE process exhibits time-varying, strong coupling, and nonlinear characteristics. While manual detection and judgment at the current time have reached a relatively mature state, the application of advanced theories and equipment has improved the controllability of the TE process. However, several challenges persist,

including the extensive human effort required for detection and the potential for delays in identifying failures when detection instruments malfunction. Consequently, there is a growing need for a dependable automatic fault detection and prediction system.

A fault detection system based on Artificial Neural Networks (ANN) has demonstrated successful applications in numerous research studies (Adeli & Mazinan, 2020; Heo & Lee, 2019; Lomov *et al.*, 2021; Xie and Bai, 2015). An analysis of the recent performance of Artificial Neural Networks (ANN), Convolutional Neural Networks (CNN), Genetic Algorithms (GA), and Fuzzy Logic in the field of fault recognition reveals that these algorithms can be individually employed effectively only when there are few fault modes, a limited number of parameters, and a relatively simple system structure (Garud *et al.*, 2021). Each of these algorithms has its own strengths and limitations. The CNN has demonstrated effective application in the fault identification of wind turbines (Xiang *et al.*, 2021). This method can learn from recorded fault situations and achieve anomaly detection through actual values and reserved residuals, providing high reliability. However, due to substantial environmental differences in wind turbine locations, this method lacks broad adaptability. CNN's parameters need adjustment, indicating a lack of inherent adaptability. In the collaborative design problem of asynchronous fault detection filters, genetic algorithms (GA) can achieve fault detection and identification (Zhang *et al.*, 2023). Nevertheless, this scenario involves fewer parameters, a relatively simple system structure, and lower computational complexity, making GA suitable. However, when dealing with complex industrial systems, GA may become slow in fault identification and prediction, or even infeasible. Therefore, combining these algorithms can result in better overall results. GA-BPNN has successfully realized real-time fault detection in the field of liquid rocket engines (Yu & Wang, 2021). This method exhibits good real-time and accuracy attributes, validated in a rocket launch, demonstrating sensitivity and robust characteristics.

In this study, the integration of GA with BPNN has been selected for the TE process. Combining Genetic Algorithm (GA) with Backpropagation Neural Network (BPNN) to create GA-BPNN offers several advantages for more efficient problem-solving. Traditional GA exhibits adaptability but often suffers from poor convergence, which can lead to issues in complex environments, including convergence difficulties and local optimization errors. On the other hand, traditional BPNN, while adaptable, is sensitive to parameter errors due to its numerous parameters. GA-BPNN capitalizes on the strengths of both methods, with BPNN as its core. It utilizes genetic algorithms to optimize BPNN parameters such as learning rate, momentum, and hidden layers. This approach results in a more precise and targeted optimization process, effectively addressing the convergence and parameter definition challenges typically encountered in traditional GA and BPNN.

## OVERVIEW TENNESSEE EASTMAN CHEMICAL PROCESS

The TE process platform serves as an open and challenging chemical model test bench, comprising multiple operating units including a continuous stirred reactor, a condenser, a gas-liquid separation tower, a stripper, a reboiler, and a centrifugal compressor. The process schematic is depicted in Figure 1. Three gas reactants, namely A, D, and E, directly enter the reactor. Feed C, along with a certain amount of feed A, is introduced into the process via the condenser.

The TE process encompasses a total of 11 action variables and 41 process variables. For the selection of process monitoring variables, several commonly used parameters closely linked to the process's operation are chosen. Among these, 11 operating variables—designated as XMV (1) through XMV (11)—serve as inputs. These variables include the flow rates of material D (flow rate

2), material E (flow rate 3), material A (flow rate 1), and materials A and C (flow 4), as well as the compressor circulation valve, air discharge valve, separator liquid flow, desorption tower steam valve, reactor cold water flow, and condenser cold water flow. Components F, G, and H are utilized as output variables.

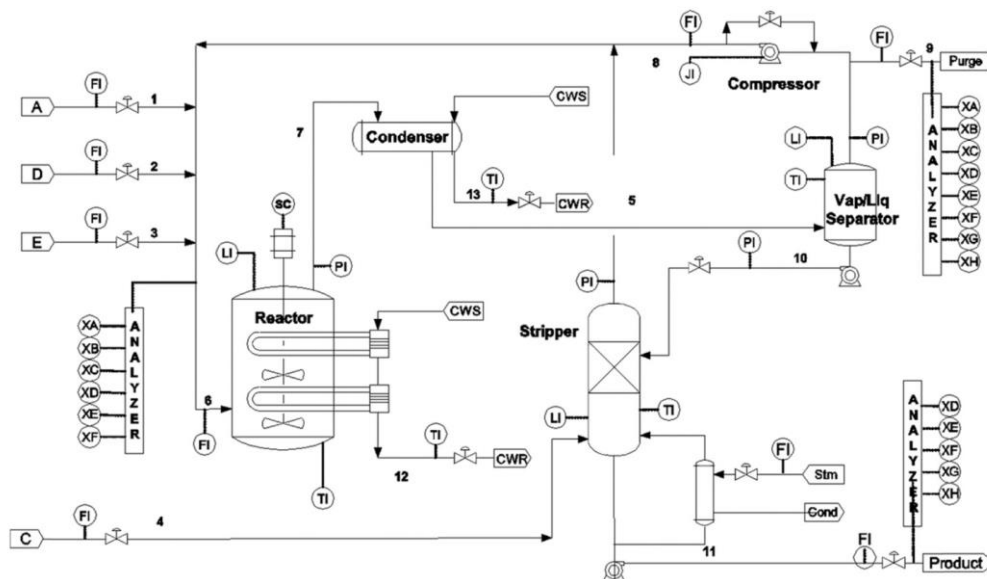


Figure 1. Tennessee Eastman (TE) process (Zhang et al., 2020)

## METHODOLOGY

### Datasets

The Tennessee Eastman (TE) process simulates real chemical processes and serves as a benchmark for testing fault diagnosis and process control. The TE simulation platform provides a total of 22 sets of data, each consisting of both a training set and a test set. Each set of TE datasets is derived from actual data collected every 3 minutes. The training set comprises a total of 500 data points, collected over 25 hours, while the test set consists of 960 data points, collected over 48 hours. Within the 01-21 fault dataset, the first 160 data points represent normal conditions, with faults occurring from the 161st data point, corresponding to the 8th hour. Each dataset includes 52 observational variables, comprising 11 control variable data, 22 reaction process data, and 19 product component measurement data. Details on the datasets are provided on the IEEE dataport (Cheng, 2019)

### Genetic Algorithm with Back-propagation Neural Networks (GA-BPNN) Method

The proposed method, GA-BPNN, represents a more efficient intelligent algorithm that combines Genetic Algorithms (GA) with Backpropagation Neural Networks (BPNN). In the GA-BPNN algorithm, the learning rate  $\eta$  is expressed as the momentum  $\Delta\omega(n)$  during the gradient descent process, as shown in the following equation.

$$\Delta\omega(n) = -\eta \frac{\partial E(w)}{\partial w(n)} + \alpha \Delta\omega(n - 1), \quad n = 1, 2 \dots \quad (1)$$

$$\Delta\omega_{min} = \Delta\omega(n) - \Delta\omega(n - 1) \quad (2)$$

where  $\Delta\omega(n)$  is the  $n$ th momentum term,  $\Delta\omega(n - 1)$  is the momentum term of the  $n$ th order,  $\eta$  represents the learning rate, and  $\alpha$  represents the additional momentum factor.

In the GA-BP neural network, the learning rate  $\eta$  determines the step size. A larger  $\eta$  results in faster optimization but can lead to issues such as repeated oscillations and failure to converge. Conversely, a smaller  $\eta$  results in a smaller step size, slower optimization, and reduced risk of oscillations and convergence problems. However, it may increase the likelihood of getting stuck in local optima or encountering gradient disappearance.  $\Delta\omega(n)$  represents the change after the  $n$ th update,  $\Delta\omega(n - 1)$  represents the change after the  $n-1$  update, and  $\alpha$  denotes the additional change factor, also known as the momentum factor. Consequently, in the BPNN, the optimal change amplitude during the learning process is determined by both the learning rate  $\eta$  and the additional momentum factor  $\alpha$ . The presence of  $\alpha$  allows for the control of the change within a certain range, preventing adverse occurrences like gradient disappearance or gradient explosion. The minimum gradient change  $\Delta\omega_{min}$  is defined as the difference between two gradient changes, and this parameter ensures continuous optimization, preventing it from halting at a local optimal solution.

The number of optimization epochs is another parameter worthy of optimization. Too many epochs can result in excessively long optimization times for the GA-BPNN, while too few may cause the optimal solution to be missed. Similarly, the choice of the number of populations in GA also has a significant impact on the optimization outcome, with too few or too many populations affecting the optimization effectiveness. The regression parameter  $R^2$  is one of the criteria for evaluating GA-BPNN optimization. During the training of GA-BPNN,  $R^2$  reflects the difference between the estimated data and the real data, as shown in Equation (3).

$$R^2 = 1 - \frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{\sum_{i=1}^n (y_i - \bar{y}_i)^2} \tag{3}$$

where  $y_i$  is actual value,  $\bar{y}_i$  is average of the actual values and  $\hat{y}_i$  is predicted value.

In this research, the GA-BPNN optimization algorithm selects genes such as the learning rate  $\eta$ , the momentum factor  $\alpha$ , the number of optimizations  $N_{ep}$ , and the minimum gradient change  $\Delta\omega_{min}$ . The objective function is defined as the regression coefficient  $R^2$ . The GA-BPNN analysis matrix is then established as given in Equation (4).

$$\begin{bmatrix} \eta_1 & \alpha_1 & N_{ep1} & \Delta\omega_1 & R^2_1 \\ \eta_2 & \alpha_2 & N_{ep2} & \Delta\omega_2 & R^2_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \eta_n & \alpha_n & N_{epn} & \Delta\omega_n & R^2_n \end{bmatrix} \tag{4}$$

In GA-BPNN,  $R^2$  is utilized as the objective function for the genetic algorithm. Because that the range of  $R^2$  is [0-1], continuous optimization of the gene matrix is pursued based on the numerical values of  $R^2$ . Throughout the optimization process, the genes from each row of the gene matrix are initially individually introduced into the neural network for training. The neural network is then trained, and the  $R^2$  of the output parameters is calculated. Based on the magnitudes of  $R^2$ , the genes within the gene matrix are rearranged. Subsequently, using the genetic algorithm, the gene matrix undergoes evolution to obtain the second-generation gene matrix. The genes of this new generation are reintroduced into the neural network to further seek the optimal  $R^2$  and simultaneously identify the optimal parameter settings for the neural network.

Notably, genes do not serve as inputs to the neural network. Instead, the inputs to the neural network comprise various operational parameters of industrial systems' equipment. The output consists of system parameters that necessitate fault detection. The genetic algorithm employs the training results of the neural network as its objective function. Consequently, these two algorithms operate concurrently. The configuration of the neural network's layers, encompassing the number of layers, is also encoded in the gene matrix. Essentially, determining the optimal number of layers in the neural network is one of the primary optimization objectives. The GA-BPNN is constructed by utilizing the first four columns of this matrix as genes, with the fifth column serving as the objective function. Subsequently, the GA-BPNN is employed to efficiently simulate and train the TE model. A comparison is then made between the results obtained from GA-BPNN training under TE fault conditions and the results under normal TE conditions. If a significant disparity is observed, fault determination is successfully accomplished. The GA-BPNN design process as shown in Figure 2.

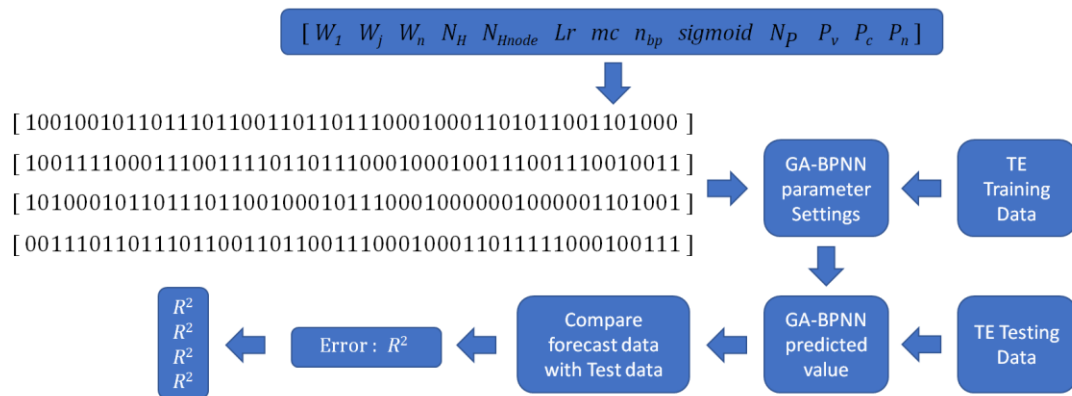


Figure 2. The block diagram of GA-BPNN design process

### RESULT AND DISCUSSION

In the first stage, the simulation has been conducted to compare the traditional BPNN method with the proposed GA-BPNN method. The TE dataset of the 13<sup>th</sup> data group from the XMV (9) extractor, which measures water flow rate pressure, is used as a test case for both methods. As shown in Figure 3(a), the traditional BPNN can capture the general trend of XMV (9). However, although it broadly reflects the trend, it cannot provide accurate estimates, resulting in a significant margin of error. In contrast, GA-BPNN as shown in Figure 2(b) not only accurately captures the overall trend but also exhibits minimal error, showcasing excellent regression performance with an  $R^2$  value close to 1. Therefore, GA-BPNN is a reliable method for fault detection, producing highly dependable results.

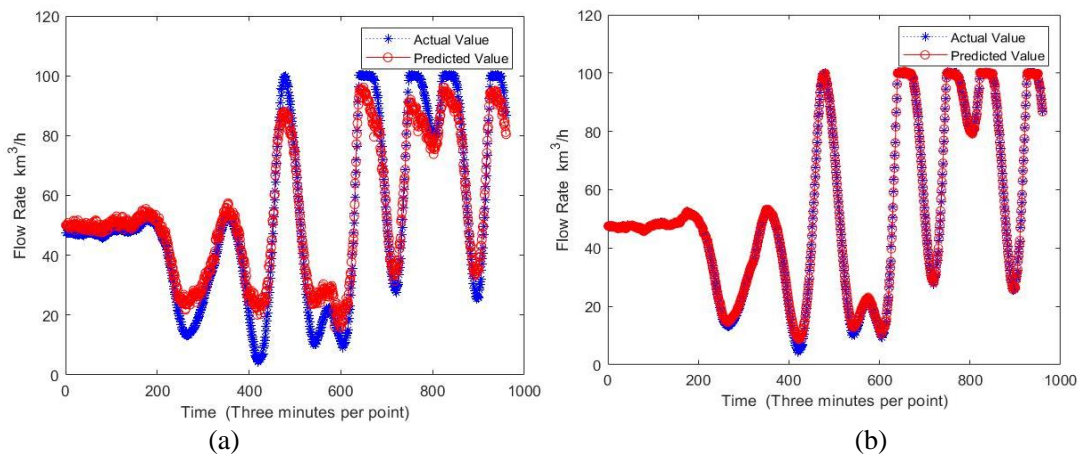


Figure 3. The comparison simulation results between (a) traditional BPNN and (b) GA-BPNN methods.



In the next stage, simulations were conducted to test the proposed method using a single fault data or abnormal condition. GA-BPNN inherits the excellent adaptability of BPNN and includes automatic parameter correction, providing the algorithm with exceptional learning capabilities that can adapt to various environments. For this specific case, we used the data material C feed pressure XMV (5) for testing. Disrupting the feed volume of material C (Flow 4) has an impact on the internal pressure of the Stripper and also disrupts the loop through the Flow 5 pair. In the initial stage of our experiments, we set the learning rate  $\eta$  to 0.01, the momentum factor  $\alpha$  to 0.95, performed 1000 optimizations ( $N_{ep}$ ), and considered a minimum gradient change amount of  $1 \times 10^{-6}$  ( $\Delta\omega_{min}$ ). We began with an initial gene chain, where the first 99 gene strands were randomly created within a  $\pm 50\%$  range. These 100 gene strands were assembled into a matrix and trained and optimized using GA-BPNN. After two generations of optimization and learning, GA-BPNN achieved an impressive regression parameter  $R^2$  of 0.9997 as shown in Figure 4. Subsequently, different settings for the initial gene were explored several times. However, the excellent self-correction ability of GA-BPNN, even with different initial gene chains, the method eventually converged to nearly the same optimal gene, resulting in a consistently excellent regression effect.  $G_{int n}$  means the initial gene chain for  $n$ , while  $G_{Final n}$  represents the final gene chain for  $n$ , as shown in Table 1.

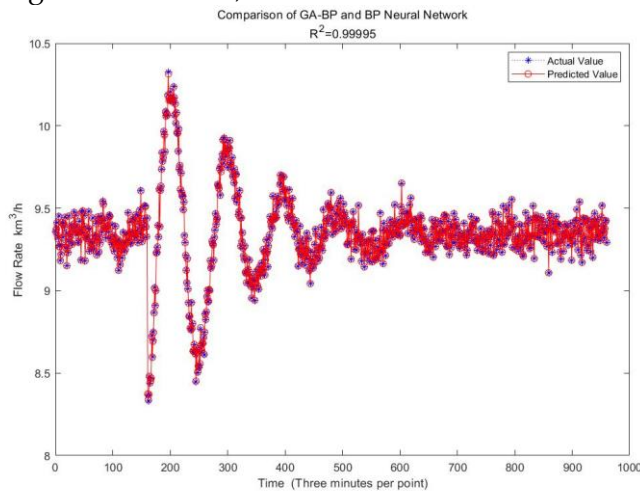


Figure 4. Learning effects of GA-BPNN on testing fault data.

Table 1. Final  $R^2$  in different initial gene strands

Gene	$\eta$	$\alpha$	$N_{ep}$	$\Delta\omega_{min}$	$R^2$
$G_{int 1}$	0.01	0.95	1000	$1 \times 10^{-6}$	0.5426
$G_{int 2}$	0.1	0.95	500	$1 \times 10^{-5}$	0.3651
$G_{int 3}$	0.04	1.0	500	$1 \times 10^{-5}$	0.3481
$G_{int 4}$	0.01	1.2	750	$1 \times 10^{-4}$	0.4897
$G_{int 5}$	0.1	0.09	500	$1 \times 10^{-4}$	0.5423
$G_{int 6}$	0.9	0.09	500	$1 \times 10^{-3}$	0.3304
$G_{Final 1}$	0.04	6.25	122	0	0.9998
$G_{Final 2}$	0.04	6.20	235	0	0.9997
$G_{Final 3}$	0.04	6.28	264	0	0.9998
$G_{Final 4}$	0.04	6.31	358	0	0.9999
$G_{Final 5}$	0.04	6.24	475	0	0.9994
$G_{Final 6}$	0.04	6.19	254	0	0.9997

### CONCLUSION

In conclusion, traditional BPNN lack the ability to optimize their own parameters, leading to inappropriate parameter selection and hindering their learning capabilities. In contrast, our

proposed method, GA-BPNN, demonstrates excellent parameter self-regulation abilities and can adapt effectively to various training conditions. This positions GA-BPNN as a superior alternative to traditional BP neural networks. Furthermore, GA-BPNN exhibits a highly reliable fault identification capability and can be applied to complex chemical processes, as evidenced by simulation results obtained from the Tennessee Eastman process data. This method boasts strong adaptability and reliability, enabling it to achieve timely and accurate fault detection. Based on the successful simulation results derived from the Tennessee Eastman process, our proposed method holds significant potential for application in other intricate engineering processes, particularly within the oil and gas industry and reactor power plants, where the implementation of a robust fault detection and identification system is essential for safety concerns.

## ACKNOWLEDGEMENTS

Special gratitude is extended to the IEEE Dataport platform for providing the database necessary for researchers to test the algorithm in this field and UPM for providing access to the MATLAB software and laboratory, which played a crucial role in the successful completion of this research project.

## REFERENCES

- [1] Downs, J.J. & Vogel, E. F. 1993. A plant-wide industrial process control problem. *Computers and Chemical Engineering*, 17(3), 245–255.
- [2] Adeli, M. & Mazinan, A. 2020. High efficiency fault-detection and fault-tolerant control approach in Tennessee Eastman process via fuzzy-based neural network representation. *Complex & Intelligent Systems*, 6 (1), 199–212.
- [3] Heo, S. & Lee, J. H. 2019. Statistical process monitoring of the Tennessee Eastman process using parallel autoassociative neural networks and a large dataset. *Processes*, 7(411), 1-21.
- [4] Lomov, I., Lyubimov, M., Makarov, I. & Zhukov, L.E. 2021. Fault detection in Tennessee Eastman process with temporal deep learning models. *Journal of Industrial Information Integration*, 23(1), 100216.
- [5] Xie, D. & Bai, L. 2015. A Hierarchical Deep Neural Network for Fault Diagnosis on Tennessee-Eastman Process. *Proceedings of the 14th International Conference on Machine Learning and Applications (ICMLA)*. 9-11 December, 2015, Miami, Florida, USA. pp 745–748.
- [6] Garud, K. S., Jayaraj, S. & Lee, M. 2021. A review on modeling of solar photovoltaic systems using artificial neural networks, fuzzy logic, genetic algorithm and hybrid models. *International Journal of Energy Research*, 45(1),1-30.
- [7] Xiang, L., Wang, P., Yang, X., Hu, A. & Su, H. 2021. Fault detection of wind turbine based on SCADA data analysis using CNN and LSTM with attention mechanism. *Measurement*, 175(4),1-10.
- [8] Zhang, X., Wang, H., Song, J., He, S. & Sun, C. 2023. Co-design of adaptive event generator and asynchronous fault detection filter for markov jump systems via genetic algorithm. *IEEE Transactions on Cybernetics*, 53(8), 5059-5068.
- [9] Yu, H. & Wang, T. 2021. A method for real-time fault detection of liquid rocket engine based on adaptive genetic algorithm optimizing back propagation neural network. *Sensors*, 15(21),1-15
- [10] Zhang, C., Guo, Q. & Li, Y. 2020. Fault detection in the Tennessee Eastman benchmark process using principal component difference based on K-nearest neighbors, *IEEE Access*, 8,49999–50009.
- [11] Chen, X. 2019. *Tennessee Eastman simulation dataset*. (<https://ieee-dataport.org/documents/tennessee-eastman-simulation-dataset>). Last accessed on 1 July 2022.