Flavio Manenti, Gintaras V. Reklaitis (Eds.), Proceedings of the 34th European Symposium on Computer Aided Process Engineering / 15th International Symposium on Process Systems Engineering (ESCAPE34/PSE24), June 2-6, 2024, Florence, Italy
 http://dx.doi.org/10.1016/B978-0-443-28824-1.50328-8

Deep Neural Networks-Based Fault Diagnosis Model For Process Systems

Mohammad Shahab[a]*, Zoltan Nagy[a], and Gintaras Reklaitis[a]

[a] Davidson School of Chemical Engineering, Purdue University, West Lafayette, IN-47907, USA

** Corresponding Author: moshahab@purdue.edu.*

Abstract

The diagnosis of faults is crucial to ensure process safety and increased product quality. Faults can emerge with time across different unit operations due to changes in the process that the process controllers are unable to handle appropriately. This undesirable divergence in the variables of the system is found to adversely affect the product quality in process industries. In this work, a deep neural network (DNN) model driven by feature engineering on the process dataset using genetic programming is developed to classify faults in a process system. Feature extraction and construction using process data is carried out before the transformed features are used for fault diagnosis in a DNN. The DNN model performs fault diagnostics on the process data that contains the normal operating conditions and the abnormal operating conditions which arise due to variations in the characteristic quality of the system. The genetic programming driven DNN methodology is illustrated on a benchmark chemical process, where its effectiveness is evaluated by classifying faults in the Tennessee Eastman Process (TEP) and is compared against existing methodologies.

Keywords: fault diagnosis, deep neural networks, genetic programming, feature engineering, pharmaceutical process

1. Introduction

Fault diagnosis refers to the identification of abnormality present in a system which implies determining the type, location, magnitude and time of the fault (Isermann R., 1995). Therefore, a fault diagnosis methodology is essential for the elimination of faults. Several fault diagnosis methods have been proposed in the past and the approaches can be broadly categorized into three types namely quantitative model methods, qualitative model methods, and data driven methods (Venkatasubramanian et al., 2003b). The first two model methods are integrated with a priori domain knowledge which is derived from a fundamental understanding of the process dynamics. On the other hand, data driven methods require large amounts of process data. However, with the increase in complexity of modern processes, creating a mathematical model that accurately represents the dynamic behavior of the system becomes increasingly difficult. Therefore, data-driven methods, which depend on process data are getting increased attention (Venkatasubramanian et al., 2003a).

The most crucial step in data driven methods is the feature extraction process which allows the transformation and conversion of the process data as priori knowledge to the fault diagnosis system. The transformed data can be used by the data driven model for fault diagnosis of the system by describing itself as a classification problem. Deep neutral networks (DNN) are a prominent data driven tool and has received significant interest in

machine learning community as a preferred method for monitoring of process systems. However, instead of using raw process data as an input to DNNs, the efficiency of this data driven method can be increased multifold by filtering out redundant and irrelevant information present in the process data, while reducing computational cost during the classification of faults. This can be carried out by using a suitable feature engineering step which essentially retains useful process variables and also constructs new features from the data.

In this work, we propose a genetic programming (GP)-based methodology for construction/extraction of features from raw process data for classification of faults. The effectiveness of the approach is tested on a benchmark chemical process for fault diagnosis, the Tennessee Eastman process (TEP) as an illustrative example and the performance of the approach is compared with other data driven methods.

2. Proposed methodology

Genetic programming (GP) is an evolutionary computation (EC) technique and a popular feature construction method. GP automatically and adaptably constructs high level features from low level ones using variable length tree-based representation. GP usually generates programs for feature construction using operators (like +, −, and ×) and given initial features (Vouk et al., 2023). The use of GP allows to gain insight into what features are important for construction and why the features that are constructed work well by interpreting the tree-based solutions of GP. These benefits have led to the development of many GP techniques that have shown promise in feature construction across a range of applications and therefore has been deployed in the current study. Figure 1 shows an example of a GP program which represents a mathematical expression (x1 – (x2 × x3)) + (x4 / x5), where the intermediate nodes (i.e., +, -, /, etc.) belong to the function set and leaf nodes such as arguments x1, x2, etc. belongs to the terminal set which can be associated with the process dataset in the current study.

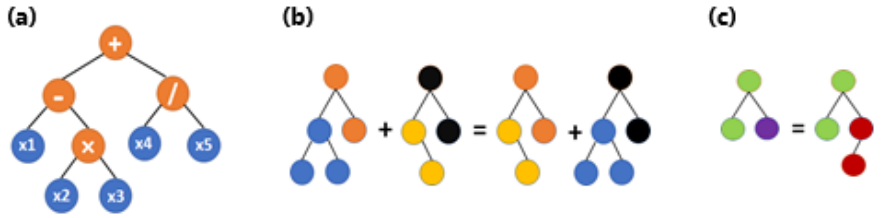

Figure 1: (a) An example of GP individual; (b) Crossover operation; (c) Mutation operation

The GP based feature extractor starts with randomly initializing a population of tree programs in the search space, each evaluated by a fitness function. The evolutionary process generates a new population through Elitism, Crossover, and Mutation operations at each generation. The Elitism operation copies the best individuals from the current generation to the next, while the Selection operation selects individuals with better fitness values for Crossover and Mutation operations (Figure 1). The process is terminated when a termination criterion is met, and the best GP derived tree is returned. For the fitness calculation of individual GP trees, DNNs are employed to evaluate the fault classification/diagnosis performance by the extracted and constructed features. By

utilizing DNN, the GP technique may automatically extract and create significant features while avoiding redundant or unnecessary features.

The evaluation process involves standardization of the extracted/constructed features from the process data by the GP individuals. The standardization ensures classification performance without singular values or feature bias. In addition, the five-fold cross-validation method is used to improve the generalization ability of the features. The DNN is fed with the standardized features and is evaluated five times, using one-fold as the test set and the remaining four folds for training. The average test classification results of the five-folds are used as the fitness value of the individual GP tree. Figure 2 shows the schematic of the GP-DNN model for fault diagnosis.

1.2 Fault diagnosis as a classification problem

The fault classification problem can be formulated as a multiclass classification problem where the extracted features by GP can be used as inputs to the DNNs. The performance of the GP-DNN model can be assessed using accuracy and confusion matrix. The diagonal elements of the confusion matrix can be directly used to calculate the accuracy which serves as the fitness value for the GP. The final GP-DNN model is tested on an unseen test data set to calculate the model performance.

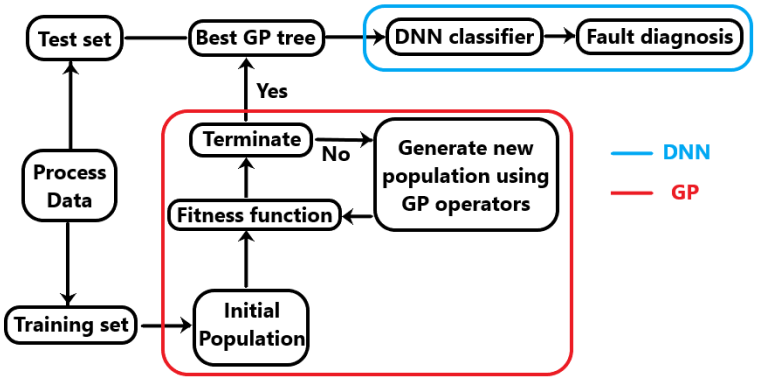

Figure 2: The main principle of GP-DNN model for fault diagnosis

3. Illustrative Example – Tennessee Eastman Process

In this section, we illustrate the effectiveness of the GP-DNN model using the Tennessee Eastman Process (TEP). The widely used TEP first introduced by Downs and Vogel, 1993 is a realistic chemical plant simulation tool that serves as a standard for process control and monitoring research (Lagare et al., 2023). The process consists of five major unit operations, the reactor, separator, condenser, stripper, and compressor and produces two products G and H, a by-product F from four reactants A, C, D, and E along with an inert compound B. The process data has 41 process variables and 11 manipulated variables (52 measurement variables) and a total of 21 different fault types. The process data which is available in Chiang et al. (2001), provides 21 training datasets that logs process measurements for 24 hours. Furthermore, for a 48-hour operating period, 21 test datasets were generated, with the faults appearing after 8 simulation hours.

Table 1: Genetic programming parameters

Function Set	Terminal Set	Pop. Size/ generations	Initial Pop. /Max tree depth	Selection method/size	Crossover/ mutation rate/Elitism
+, -, ×, protected division	Process variables	100/30	Ramp half and half/8	Tournament/ 7	0.8/0.2/1

The GP-DNN model uses the TEP data to extract features using GP and identify the best GP tree which describes the selected features (process variables) in the leaf nodes and the constructed feature(s) in the root node (top node). The selected features and the constructed features are fed as inputs to the DNN after standardization of the combined features. The DNN uses the combined features to classify the process sample into one of the 21 different faults. Therefore, the approach not only allows us to select relevant features but also construct useful new features from the process data. Before using DNN for evaluating the fitness of GP individuals, the hyperparameters of the neural networks such as learning rate, hidden layers, neurons, optimizer, and activation function can be fine-tuned to increase the performance of the GP-DNN model. Therefore, a single GP-DNN model can be used to classify faults for the TEP.

Table 2: Deep neural network parameters

Hidden layers	Learning rate	Neurons	Optimizer	Activation function	Dropout
6	0.124	128	Adagrad	elu	0.06

4. Results and Discussion

Our proposed GP-DNN selects certain features (process variables) from the process dataset using GP and it is noteworthy that in the current work, we have constructed only one feature from the selected features by GP for demonstration purposes. However, even the addition of a single constructed feature was found to be very useful for fault diagnosis. The feature extraction using GP is implemented using DEAP library in python and its parameters like the function set, terminal set, population size, generations, and other parameters are mentioned in Table 1. The DNNs are optimized using Bayesian optimization to identify the optimal learning rate, neurons, hidden layers, and also the optimizer and the activation function using the TEP dataset using *bayes_opt* library in python and is shown in Table 2.

We demonstrate the effectiveness of the proposed approach for fault diagnosis in TEP. In order to carry out a fair comparison between different model performances, we have considered studies like Jing et al., 2014 and Eslamloueyan R., 2011, where a principal component analysis (PCA) model and a multilayer perceptron (MLP) model are used for classification of all 21 faults using a single model only, respectively. The results are reported in Table 3 after averaging the multiclass classification accuracy for 10 simulation runs on the unseen TEP test data set. The accuracy of our GP-DNN model is based on 25 extracted features (averaged across 10 simulations) from the process data along with one constructed feature. The following observations can be made: First, our proposed approach results in a higher average accuracy across all 21 faults using a single model which is 7% and 40% more than the PCA and MLP method, respectively; Secondly, the average fault diagnosis accuracy of incipient faults such as 3, 9, 15 is also relatively higher compared to other techniques which are very difficult to diagnose due to absence of any

observable changes in means, variances or peak time. It is to be noted that both these findings occur with lower number of features along with a constructed feature that assisted in increasing the performance by identifying what kind of feature operations in the GP tree would result in a better discrimination of faults. The confusion matrix, which measures performance of classification problems, for one of the simulations using the GP-DNN model is shown in Figure 3.

Table 3: Fault classification accuracy of our proposed approach against single Principal component analysis (PCA) model (Jing et al., 2014), single multilayer perceptron (MLP) model (Eslamloueyan R. 2011).

Fault type	PCA	MLP	GP-DNN
1	88	81	90
2	89	82	89
3	21	0	49
4	81	79	83
5	87	73	85
6	89	84	91
7	88	80	89
8	83	48	86
9	22	0	45
10	76	12	80
11	70	18	81
12	87	25	87
13	69	15	86
14	88	29	86
15	26	0	46
16	73	15	77
17	75	52	86
18	73	75	84
19	85	14	71
20	79	48	73
21	85	0	80
Average (%)	73	40	80

5. Conclusions

In this study, we proposed a GP-DNN model for fault diagnosis in process systems. It was shown that the GP-DNN model is capable of generating useful features and reducing redundant information in process data used to execute fault classification. The tree-based GP approach allows interpretability of which process variables are useful for classifying faults. The optimized DNN boosts performance when appropriate features are plugged in giving a balance of both model interpretation and high learning speed. The GP-DNN model is also compared with existing technologies for fault diagnosis and is found to outperform most methods when all 21 faults are considered. Therefore, feature engineering can be a key step in transforming process data into useful a priori domain knowledge which is one of the major challenges in developing data driven methods for fault detection and diagnosis.

Although, the results indicating that a single GP-DNN model can be used to diagnose all process faults seem to be promising, numerous points need to be addressed in future work to improve the effectiveness. First, the DNN needs to be tuned proportional to which

features in process data are deemed useful and optimizing the DNN using the entire dataset might not be efficient. Second, instead of constructing single feature from the selected features by GP, multiple constructed features can be concatenated for improved performance, in which case, the selected features may be not be needed at all. Finally, the DNNs could be replaced with recurrent neural network (RNN) models that are much more relevant to sequential data which occurs in process industries.

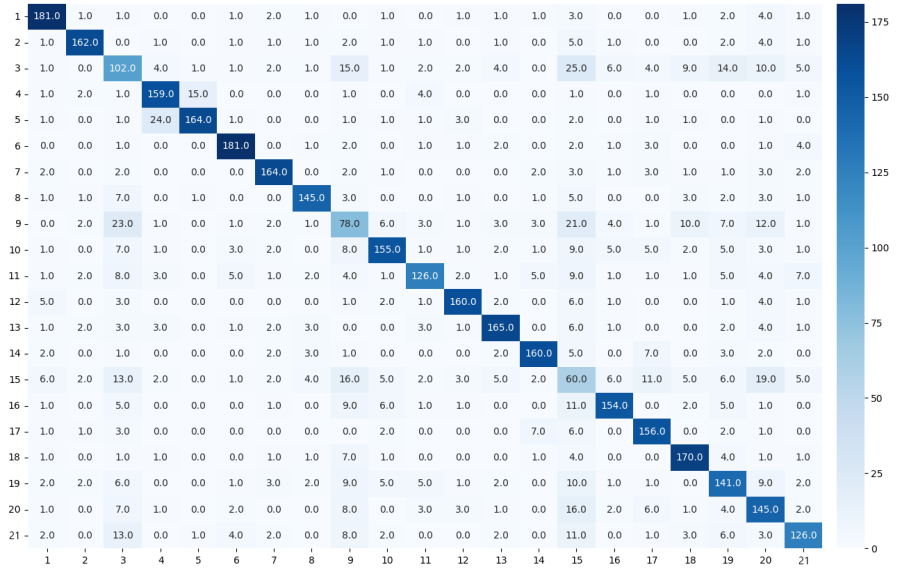

Figure 3: Confusion matrix for test data using the GP-DNN model (the axes denote the fault classes from 1-21)

References

Downs, J.J., Vogel, E.F. (1993). A plant-wide industrial process control problem. *Comput. Chem. Eng.*, 17(3), 245–255.

Eslamloueyan R. (2011) Designing a hierarchical neural network based on fuzzy clustering for fault diagnosis of the Tennessee–Eastman process. *Appl. Soft Comput.*, 11(1), 1407-15.

Isermann, R. (1995). Model base fault detection and diagnosis methods. In Proceedings of 1995-*ACC'95 IEEE*, Vol. 3, 1605-1609.

Jing C, Gao X, Zhu X, Lang S. (2014) Fault classification on Tennessee Eastman process: PCA and SVM. In *ICMC-IEEE*. pp. 2194-2197.

Lagare, R.B., Gonzalez, M., Nagy, Z.K. and Reklaitis, G.V. (2023). Modular Development of Condition Monitoring Systems for the Tennessee Eastman Process. In *Comput. Aided Chem. Eng.* Elsevier. Vol. 52, 1579-1584.

Venkatasubramanian, V., Rengaswamy, R., Yin, K., and Kavuri, S. (2003b). A review of process fault detection and diagnosis: Part I: Quantitative model-based methods. *Comput. Chem. Eng.*, 27(3), 293–311.

Venkatasubramanian, V., Rengaswamy, R., Kavuri, S., and Yin, K. (2003a). A review of process fault detection and diagnosis: Part III: Process history based methods. *Comput. Chem. Eng.*, 27(3), 327–346.

Vouk, B., Guid, M. and Robnik-Šikonja, M., 2023. Feature construction using explanations of individual predictions. *Eng. Appl. Artif. Intell.*, *120*, p.105823.