v. 3, n. 33, 2023

Journal of
Engineering
Research

DEVELOPING A DATA SCIENCE PLATFORM FOR PIPELINE APPLICATIONS TENNESSEE EASTMAN

Erick Gonzalez-Arce
Doctorate in Engineering Science with mention in Process Engineering, Department of Chemical Engineering, ``Universidad de Santiago de Chile``

Abstract: A data science platform in a smart chemical plant enables simultaneity of multiple applications. However, in each application there is a wide variety of algorithms that can be implemented. Therefore, the objective of this work is to evaluate the performance of the algorithms to be implemented in a data science platform for applications related to fault classification, fault detection and virtual sensor in the Tennessee Eastman (TE) process. To achieve this, a synthetic data set obtained by simulating the TE process was built, where each algorithm was trained, optimized and evaluated according to the type of data science task associated with each application. The results show that the artificial neural network only achieved the best performance in fault classification. In the virtual sensor, gradient boosting (GB) and k-nearest neighbor (k-NN) achieved the best performance. Meanwhile, in fault detection, most of the evaluated algorithms achieved a fault detection rate around 88%. In conclusion, algorithms based on artificial neural networks did not achieve the best performance in all implemented applications, being surpassed by other non-linear algorithms.
Keywords*:* predicción, clasificación, detección, industria 4.0, smart chemical plant.

INTRODUCTION

Currently, new data sources (e.g. social networks, streaming platforms, e-commerce, etc.) and new measurement devices (wearables, internet of things devices, etc.) are generating and storing a large amount and diversity of data every second. These data are fundamental in various sectors, because the analysis of this data contributes to competitiveness, productive growth and innovation (Manyika et al., 2011, as cited in Qin, 2014). However, the analysis of these new data requires the use of new tools and large computational resources (Rajaraman, 2016). In this sense, data science emerges as a new discipline that allows knowledge and information to be discovered from these massive data (Beck et al., 2016; L. Chiang et al., 2017; National Academies of Sciences Engineering and Medicine, 2018; Qin, 2014).

Similarly, the implementation of industry 4.0 (I4.0) and smart *manufacturing* (SM) allows the generation and storage of a large amount of data from different locations of the process plant (Dorneanu et al., 2022). Therefore, data science algorithms exploit the information contained in this process data for decision making.

The involvement of data science algorithms in decision making makes the chemical process industry "smart"; and a smart chemical plant is characterized by being faster, more flexible and more efficient to produce high-quality services at low cost (Lin et al., 2017). For example, Kim (2017) reported that a smart chemical plant improved 0.5 to 2 times compared to the existing plant. However, data science tools must be considered within the design plan of new smart chemical processes (L. Chiang et al., 2022)

In the architecture of a smart chemical plant, the data science platform is designed to enable efficient data analysis, exponential growth of connected devices (scalability), and concurrency of multiple data science applications (Voigt et al, 2021). This platform includes a series of technologies that allow the collection, storage, management, processing and modeling of data (Kabugo et al., 2020; Lee et al., 2017). But, in general, the data science platform must have the following three fundamental elements for its implementation: process data, algorithms and infrastructure.

Figure 1 shows each of these elements present in a data science platform. Within these three elements, data science algorithms play a fundamental role, since they are responsible for extracting knowledge from

process data, and then using that knowledge for decision making. Data science algorithms come from the field of machine and statistical learning (Beck et al., 2016); and these can be divided according to the type of learning into supervised and unsupervised. Supervised learning algorithms seek to predict an output variable from a set of input variables, and are generally associated with the tasks of regression (prediction of a numerical value) or classification (prediction of classes). Whereas, unsupervised algorithms work with unlabeled data to learn or explore the hidden structure of the data, therefore, these algorithms are used for data clustering, anomaly detection, and dimensional reduction.

The problem with data science algorithms is that there is a wide variety of algorithms that can be used for each application. Furthermore, the No Free Lunch theorem states that, if there is no assumption about the data obtained from the process, there is no reason to prefer a specific algorithm (Géron, 2020, p. 63). So a good strategy is to evaluate all the algorithms and select the best performing one for each application. The data analytics platform proposed by Kabugo et al. (2020) evaluated and compared some regression algorithms in two applications related to a virtual sensor in a *waste-to-energy* plant. However, the evaluation of data science algorithms in the context of the multiple applications of a data science platform in a smart chemical plant is a still underexplored area.

One of the applications of data science in the smart chemical industry is fault detection. Fault detection consists of determining whether the system is in normal operating conditions or not (Sun et al., 2020). Then, the detection algorithms are trained with data under normal operating conditions, thus finding a function that differentiates the new observations between normal operation or failure (Quiñones-Grueiro et al., 2020, p. 5). Some algorithms traditionally used in fault detection are principal component analysis (PCA), independent component analysis (ICA) and partial least squares (PLS) (Ge et al., 2013; Park et al., 2020 ; Qin, 2012; Quiñones-Grueiro et al., 2020, p. 70 - 74; Yin et al., 2014). However, these techniques have been adapted to the nonlinear and complex nature of chemical processes, thus creating new nonlinear fault detection algorithms, such as Kernel PCA (Fazai et al., 2016; Samuel & Cao, 2016 ; Y. Zhang, 2009) or automatic autoencoder type neural network (Loy-Benitez et al., 2020; Lv et al., 2016; Neubürger et al., 2021; Qiu & Dai, 2019; Yan et al., 2016; C. Zhang et al., 2021; Z. Zhang et al., 2018).

Failure classification is another application of data science in the smart chemical industry. Failure classification occurs after the failure has been detected, and consists of identifying the variable or part of the process that is failing (Quiñones-Grueiro et al., 2020, p. 5). In general, fault classification can be considered as a pattern classification problem, and some algorithms used in fault classification highlight the Bayesian classifier (Quiñones-Grueiro et al., 2021), the random forest (RF) (Chai & Zhao, 2020; Liu & Ge, 2018; Quiñones-Grueiro et al., 2021), the support vector machine (SVM) (Jing & Hou, 2015; Quiñones-Grueiro et al., 2021; F. Wu et al., 2021; al., 2014) and artificial neural networks (ANN) (Ayubi Rad & Yazdanpanah, 2015; Heo & Lee, 2018; Quiñones-Grueiro et al., 2021). Pero, Heo and Lee (2018) highlight that neural networks Deep learning methods have shown better performance in fault classification than other data-driven methods. Then, deep learning architectures have been used in the classification of failures in industrial processes, highlighting the convolutional neural network (CNN) (H. Wu & Zhao, 2018), the deep belief network (DBN) (Z. Zhang & Zhao, 2017) and LSTM-type

recurrent neural networks (Lei et al., 2019; Lomov et al., 2021; Omar et al., 2020; Zhao et al., 2018).

The virtual sensor is another commonly used application within smart chemical processes. Several authors (Kadlec et al., 2009; Souza et al., 2016) define the virtual sensor as an inferential model that predicts a variable that is difficult or expensive to measure (e.g. quality of a product or KPI) through other variables continuously. measurements in the industrial process (e.g. temperature, pressure, flow, etc.). If the variable to be predicted corresponds to a numerical value, the virtual sensor is related to the regression task. Among the simplest regression algorithms to use in the virtual sensor, multivariate linear regression, principal component regression (PCR) (Ge, 2018), partial least squares regression (PLS), and Gaussian process regression stand out. (GPR) (Wang et al., 2016) and the support vector machine for regression (SVR) (Meng et al., 2019; Zhongda et al., 2016). However, if the phenomenon modeled by the virtual sensor has a non-linear nature, the most appropriate would be to use non-linear regression algorithms, such as artificial neural networks (Wang et al., 2016), extreme learning machines (ELM) (He et al., 2015, 2016), k-nearest neighbors (k-NN) regression and deep belief network (Shang et al., 2014). Meanwhile, if the dynamics of the process affects the regression models, it is necessary to insert the past values into the output variable and the input variables. Some regression models with dynamic considerations are the NARMAX model (Acuña et al., 2014) or dynamic PLS (Shang et al., 2015). But, in the field of deep learning there are also architectures that can model the temporal dynamic behaviors of sequential data, these architectures are recurrent neural networks (RNN) and their extensions (LSTM and GRU) (Kwon et al., 2021; Yuan et al. al, 2020).

In summary, there are a wide variety of algorithms used in fault classification, fault detection, and virtual sensing, but there are still certain algorithms that have little or no use cases in these applications. So, the objective of this work is to evaluate the performance of supervised and unsupervised learning algorithms, to be implemented in a data science platform for applications related to fault classification, fault detection and virtual sensor in the Tennessee process. Eastman (TE). To this end, the hypothesis stated states that algorithms based on artificial neural networks will have better performance in all applications implemented in the TE process. The results obtained by this work contribute to the knowledge of the performance of different algorithms in the multiple applications (multiclass fault classification, fault detection and virtual sensor) that can be implemented in a data science platform in a smart chemical plant.

Therefore, the article is organized into the following sections: the methodology section presents the data acquisition, preprocessing and modeling steps carried out in the three applications; In the results and discussions section, the performance of the different data science algorithms in each application is evaluated; and in the conclusions section the proposed hypothesis is verified and the most appropriate algorithms to implement in each application are proposed.

METHODOLOGY

TENNESSEE EASTMAN PROCESS

The data science platform for applications in chemical processes was implemented in the Tennessee Eastman (TE) process. The TE process was described by Downs and Vogel (1993); and is composed of five process units: reactor, condenser, liquid-vapor separator, *stripper* column and compressor. Figure 2

presents the flow chart of the TE process.

The TE process is characterized by its high complexity, because it has the following characteristics:

(a) High dimensionality: in total the TE process has 12 manipulated variables (XMV) and 41 measured variables (XMEAS, 22 continuous variables and 19 sampled type current compositions variables). But, in the last revision of the TE process, 32 new measured variables were added (eight continuous variables and 24 composition variables) (Bathelt et al., 2015).

(b) Multimode process: The TE process has six modes of operation at three different mass ratios of G and H in the product stream.

(c) Multiple failures: Table 1 details the 20 disturbances that affect different variables of the TE process.

TENNESSEE EASTMAN PROCESS DATABASE CONSTRUCTION

To implement fault detection, fault classification, and virtual sensor applications in the TE process, process data is required. These data were obtained through a dynamic simulation of the TE process. For this, Simulink models were used with the decentralized control strategy proposed by Ricker (1996). These models are stored in the repository *Tennessee Eastman Challenge Archive* (Ricker, 2015).

The Simulink models were implemented with a code programmed in MATLAB of the TE process developed by Braun and Rivera (1999). In addition, the MATLAB code was adapted to add the 32 new measured variables (XMEAS) from the article by Bathelt et al. (2015). Consequently, the MATLAB code of the TE process with the decentralized control strategy allows the extraction of a total of 85 variables (12 manipulated variables and 73 measured variables).

The dynamic simulation of the TE process consisted of a total of 40 runs, one in normal operation and 19 with disturbance activated (IDV6 disturbance is excluded from Table 1) for operation modes 1 and 3. Each simulation run had a duration of five days (120 h), with a process data storage rate of 0.01 h (36 s). In the case of simulation runs with perturbation enabled, the perturbation was introduced after the first 8 h of simulation.

TENNESSEE EASTMAN PROCESS DATABASE PREPROCESSING

Table 2 shows that the TE process database is composed of 480,040 rows and 88 columns. The variables (columns) contain numerical and categorical data. The columns with numerical variables are related to the measured (XMEAS) and manipulated (XMV) variables of the TE process. While, the categorical variables identify the mode of operation (1 or 3) and type of disturbance that affects the TE process. Furthermore, Table 2 shows that the TE process database does not contain missing values, so treatment of missing values was not required in the preprocessing stage.

Data science applications were implemented for each mode of operation of the TE process. Then, the original database was divided by each mode of operation. Each data set was then manipulated according to each data science application.

In the case of failure classification, the manipulation of the database consisted of eliminating the columns related to the measured variables of composition of the TE process (XMEAS23 to XMEAS41 and from XMEAS50 to XMEAS73). This way, the resulting data set is made up of the manipulated variables (XMV) and continuous measures (XMEAS1 to XMEAS22 and from XMEAS42 to XMEAS49) with the respective column

that indicates the type of disturbance present in the sample. The dimensions of the data set used in fault classification have 240,020 rows and 44 columns for each mode of operation.

In the case of the virtual sensor and fault detection, the manipulation of the data set began with the synchronization of the composition variables of the currents with the continuous measured variables. This synchronization allowed the generation rate of product G in the reactor to be calculated. Once calculated, the non-synchronized samples were removed and columns related to the measured composition variables were eliminated. The result of this manipulation allowed us to obtain a data set that has 4,780 samples and 45 columns for each mode of operation.

In each database of the TE process, the samples whose sampling time was within the fifth day of operation (between 96 h to 120 h) were selected. This set will be used to evaluate the generalization capacity of results on data unknown to the models. While, in the rest of the data, the samples were selected between 8 h and 96 h of operation, and then in this set the division was carried out in the training and test set using the following division reasons: (a) in the case For the classification of failures, the division ratio was set at 75% training and 25% testing; (b) in the case of the virtual sensor, the division ratio was set at 80% training and 20% testing.

Finally, in each of the data sets (training, test and fifth day of operation) the detection and treatment of inconsistent data (negative values in certain measurements) was carried out. In addition, the Local *Outlier* Factor (LOF) was used to detect and remove outliers. The preprocessing stage culminated with feature selection in the test set by measuring the feature importance obtained by the *Extra Tree* ensemble algorithm. Table 3 indicates the selected input variables and target variable in the fault classification and virtual sensor applications. However, it must be noted that the fault detection considered all the manipulated variables and continuous measures indicated in Table 2, because an unsupervised algorithm approach is used for fault detection.

IMPLEMENTATION OF A DATA SCIENCE PLATFORM FOR APPLICATIONS IN THE TENNESSEE EASTMAN PROCESS

The algorithms were programmed in Python using the *Scikit-Learn* (machine learning algorithms) and *Tensorflow* (artificial neural networks and autoencoders) libraries. The execution of the algorithms was carried out in the *Google Colab* cloud computing environment.

Table 4 shows the supervised algorithms used in the fault classification (multiclass classification) and virtual sensor (regression) applications. Fault classification in the TE process is related to the multiclass classification problem with slightly balanced classes. Therefore, binary classification algorithms, such as logistic regression (RL) and support vector machine with radial basis kernel function (SVM-rbf), used the binary decomposition technique to overcome the multiple problem. classes. Meanwhile, the regression algorithms used in the virtual sensor did not require any adjustment, because only the numerical value of the G generation rate in the reactor is predicted based on the input variables described in Table 3.

The algorithms used in fault detection were Principal Component Analysis (PCA), Independent Component Analysis (ICA) and Kernel PCA (KPCA). In addition, fault detection techniques based on artificial neural networks such as simple autoencoder (AE), stacked autoencoder (SAE) and variational or probabilistic autoencoder (VAE) were used. Table 5 specifies the procedure used

by the algorithms to detect failures in the TE process, which uses the squared prediction error statistic.(*squared prediction error*, SPE) for fault detection.

The hyperparameters of the algorithms were optimized by the grid search method (GridSeachCV) in order to obtain the best possible result by the algorithm in each application. Meanwhile, the artificial neural networks and autoencoders were optimized through trial and error. Hyperparameter optimization seeks to obtain the best possible result from the algorithms in fault classification, fault detection and virtual sensor.

Finally, cross validation techniques with five iterations (*5-Fold Cross Validation*) were used in order to train the algorithm with different partitions of the training set. Then, the trained algorithms were evaluated on the test set, where the different metrics were calculated that allow measuring and comparing the performance of the different algorithms.

EVALUATION OF ALGORITHMS IN THE DEVELOPMENT OF A DATA SCIENCE PLATFORM FOR APPLICATIONS IN THE TENNESSEE EASTMAN PROCESS

To evaluate the performance of the algorithms in fault classification, the indicators of precision (PREC), sensitivity or detection rate (FDR), missed detection rate (MDR) and false alarm rate (FAR) were used for each fault of the TE process. Equations (1) to (4) allow each of these metrics to be calculated according to the values obtained in the confusion matrix represented in Table 6. Also, the general performance of the algorithms in classifying process failures was measured. TE through the accuracy metric (ACC, see equation (5)), weighted average precision and weighted average sensitivity.

$$Precision = Prec = \frac{No.\ of\ failure\ samples\ i\ correctly\ labeled}{Total\ number\ of\ sample\ labeled\ as\ failure\ i} = \frac{VP}{VP + FP} \quad (1)$$

$$\text{Sensitivity or "Detection rate"} = \text{FDR} = \frac{\text{No. of failure samples i correctly labeled}}{\text{"Total number of failure samples" i}} = \frac{VP}{P} \quad (2)$$

$$\text{Missed detection rate} = \text{MDR} = \frac{\text{No. of failure samples i incorrectly labeled}}{\text{Total number of failure samples i}} = \frac{FN}{P} \quad (3)$$

$$False\ alarm\ rate = \text{FAR} = \frac{\text{Total number of samples wrongly labeled as failure i}}{\text{Total number of sample other than failure } i} = \frac{FP}{N} \quad (4)$$

$$\text{Accuracy} = \text{Acc} = \frac{\text{Sample number correctly labeled}}{\text{Total number of samples}} = \frac{VP + VN}{P + N} \quad (5)$$

To evaluate the fault detection, the fault detection rate (equation (2)) and the missed detection rate (equation (3)) were used. In this, true positives (TP) were considered as those samples that were correctly detected as a failure, while, if the sample was incorrectly detected as normal operation, it was counted as a false positive value (FN). Furthermore, it must be noted that, in the case of normal operation detection, the definition of true positives and false negatives are based on the number of samples under normal operation.

Finally, the virtual sensor application used the root mean square error metrics(*mean square error*, MSE) and mean absolute error (*mean absolute error*, MAE) to evaluate regression algorithms. Equations (6) and (7) allow these two indicators to be calculated.

$$MSE = \frac{1}{m}\sum_{i=1}^{m}(y_i - \hat{y}_i)^2 \quad (6)$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}|y_i - \hat{y}_i| \quad (7)$$

Where $\hat{y}_i$ represents the predicted value obtained by the algorithm in the virtual sensor, y_i represents the current value of the output and m is the sample number of the testing or evaluation set.

The metrics used to evaluate the classification of faults and virtual sensor in the TE process were extracted from the module *metrics* from the library *scikit-learn*. While, in the case of fault detection, a function was programmed that allows calculating the fault detection rate according to the value obtained from the SPE_{lim}.

RESULTS AND DISCUSSIONS

EVALUATION OF ALGORITHMS FOR THE CLASSIFICATION OF MULTICLASS FAULTS IN THE TENNESSEE EASTMAN PROCESS

To evaluate the general performance of the algorithms in classifying faults in the TE process, the indicators of accuracy, average precision and average sensitivity were used.

Figure 3 shows that artificial neural network 1 (ANN 1) achieved the highest accuracy in fault classification in operation modes one and three. However, the *random forest* (RF) and support vector machine with radial basis function kernel (SVM-rbf) algorithms achieved similar accuracy to the neural network in the respective modes one and three of operation.

In the case of the average precision, Figure 4 shows that the ANN neural network 1 and the RF algorithm achieved the best average precision in mode one of operation. Meanwhile, the SVM-rbf algorithm achieved a higher average precision in mode three, surpassing the performance achieved by artificial neural networks in this mode of operation.

The average sensitivity or detection rate also reached a similar trend to the other two indicators. Therefore, Figure 5 shows that ANN algorithm 1 achieved the best performance on this evaluation metric in mode one of operation. However, the SVM-rbf algorithm matched the sensitivity achieved by the ANN algorithm 1 in mode three, consequently, both algorithms achieved the best performance in fault classification in mode three by the sensitivity metric.

Therefore, these results demonstrate that artificial neural networks did not achieve the best performance in fault classification. The good performance achieved by the RF and SVM-rbf algorithms is explained because it also has the ability to establish nonlinear decision boundaries in pattern classification. Therefore, if the nature of the TE process data has high nonlinearity (Quiñones-Grueiro et al., 2021), the nonlinear algorithms (ANN 1, ANN 2, RF and SVM-rbf) are capable of adequately dividing the multiple faults of the TE process, thus achieving the best results in fault classification.

Other nonlinear algorithms, such as k-nearest neighbors (k-NN), decision tree (DT) and *gradient boosting* (GB), also achieved acceptable performance, given that in their three metrics they achieved a range of values between a 84% to 88%. On the contrary, classification algorithms that establish linear decision boundaries, such as logistic regression (RL) and linear discriminant analysis (LDA), achieved poor performance in classifying TE process failures, obtaining an accuracy, average precision and average sensitivity between 56% to 60%.

Figure 6 and Figure 7 allow evaluating the performance achieved by each algorithm in each disturbance of the TE process in both operating modes. The results show that the nonlinear classification algorithms managed to correctly classify most of the disturbances, because they obtained a high precision and detection rate (over 84%), with a low missed detection rate and false alarm rate. On the contrary, the linear algorithms only managed to correctly classify faults IDV1 to IDV7 (step type disturbance), while the rest of the disturbances exhibited a greater tendency to misclassify certain samples, affecting the evaluation indicators.

Figure 6 and Figure 7 also show that all algorithms presented difficulties in classifying IDV15 and IDV16 failures. Several articles have also reported this problem in the classification of IDV15 and IDV16 perturbations using different algorithms (Li et al., 2020; H. Wu & Zhao, 2018; Yin

et al., 2012; Z. Zhang & Zhao, 2017). This classification problem is explained because these perturbations are small and have little influence on the process (L. H. Chiang et al., 2001, p. 137; Ge & Song, 2013, p. 19; F. Zhang & Ge, 2015; Y Zhang, 2009). Furthermore, Isermann (2006) points out that a failure must generate a large permanent deviation in the controlled and manipulated variables to be detected. Therefore, disturbances IDV15 and IDV16 were correctly controlled by the TE process control strategy, resulting in no large deviation from normal operation occurring.

Figure 8 shows the effect of removing perturbations IDV15 and IDV16 from the data set. The results demonstrate that all algorithms improved classification performance by removing these faults, increasing the three indicators between 4.8% to 16% in both modes of operation. However, Figure 8 shows that the non-linear classification algorithms presented better performance than the linear ones, reaching an accuracy greater than 95%. Therefore, nonlinear classification algorithms are able to cope with the complexity and nonlinearity of the TE process.

EVALUATION OF ALGORITHMS FOR FAULT DETECTION IN THE TENNESSEE EASTMAN PROCESS

Table 7 **Table** 6. highlights that most of the algorithms achieved good performance in detecting faults in the TE process, because the average detection rate was over 85%. Therefore, it was not possible to demonstrate the superiority of fault detection using algorithms based on artificial neural networks (AE, SAE and VAE). However, the stacked autoencoder (SAE) neural network and Kernel PCA (KPCA) achieved slightly better performance in fault detection in modes one and three of the TE process.

Figure 9 and Figure 10 break down the detection rate achieved by each algorithm in each disturbance in modes one and three of operation. The results show that there were easy-to-detect perturbations for all algorithms in both modes of operation; These disturbances were IDV1, IDV2, IDV3, IDV4, IDV5, IDV7, IDV8, IDV9, IDV10, IDV11, IDV12, IDV14 and IDV19. While, faults IDV15 and IDV16 were the most difficult to detect for all algorithms, experiencing the same problem as in fault classification.

Meanwhile, the group of perturbations composed of the lDV18 in mode one and the IDV13, IDV17 and IDV20 in mode three, it was obtained that the principal component analysis (PCA) and independent component analysis (ICA) achieved the highest rate. detection compared to the other algorithms. But, the PCA and ICA algorithms achieved low detection of the normal operation of the TE process. In contrast, the autoencoders (AE, SAE, and VAE) achieved better detection of normal operation. Indeed, the variational autoencoder (VAE) achieved a normal operation detection rate of 100% in both operation modes.

The results obtained by the PCA algorithm do not agree with the results achieved in other works, where the PCA algorithm obtained a low failure detection rate in the TE process (Yin et al., 2012). So, the good performance achieved by PCA can be explained by the nature of the data set used in this application. Therefore, it must be noted that the failure detection was carried out with a synchronized subsample of the TE process, and it is possible that, in this set with a smaller amount of sample, the variables have a lower linearity with a greater Gaussian behavior. Consequently, the characteristics of the synchronized sample were more similar to the PCA, and this allowed good fault detection with this algorithm.

The size of the subsample used in fault detection could also affect the fault detection achieved by the autoencoders. The reason is

because autoencoders, like any artificial neural network, perform best when a large amount of data is available. When a neural network is trained with few samples, it tends to overfit to the training data. To avoid overfitting, the autoencoders were implemented with the early training detection technique and with autoencoder configurations with a small number of neurons and hidden layers. But, the small number of neurons and hidden layers could limit the ability to differentiate normal operation from a failure state. Therefore, it is possible that the autoencoders have suffered from this problem and that explains the lower detection rate achieved in certain perturbations.

EVALUATION OF ALGORITHMS FOR THE VIRTUAL SENSOR IN THE TENNESSEE EASTMAN PROCESS

The results demonstrate that the neural network (ANN) was not the most effective in the application of the virtual sensor in the TE process. Table 8 shows that the *Gradient Boosting* (GB) algorithm was the one that obtained the lowest mean square error (MSE) in both operating modes, therefore, it obtains the best performance in this metric. Meanwhile, Table 9 shows that the k nearest neighbors (k-NN) algorithm obtained the best performance in the virtual sensor when the algorithms are evaluated by the mean absolute error (MAE) metric.

Thus, the poor results achieved by the artificial neural network on the virtual sensor contrast with the performance achieved in fault classification. In the virtual sensor, the neural network was also configured with a small number of neurons and hidden layers, added to the small number of samples used in training, the neural network could not adequately model the phenomenon of the generation rate of G in the reactor. In the case of fault classification, the two neural network models were trained with a large number of samples, which allowed the use of neural network configurations with a greater number of parameters, which better captured the fault classification of the TE process. So, this finding demonstrates the effect of data set size on the performance of artificial neural network in different applications. Therefore, in this TE process data set with few samples, the k-NN and GB algorithms presented a greater ability to model the G generation rate in the reactor.

EVALUATION OF THE GENERALIZABILITY OF RESULTS ON THE FIFTH DAY OF OPERATION OF THE TENNESSEE EASTMAN PROCESS

The fifth day of operation of the TE process was used to evaluate the ability of the algorithms in generalizing results in the three applications.

In the case of fault classification, Figure 11 shows that the algorithms experienced a slight reduction in all three classification evaluation metrics. In the case of mode one of operation, the reduction in predictive ability was on average 2.57% in accuracy, 2.44% in precision, and 2.71% in sensitivity. While, mode three of operation experienced a reduction of 2.49% in accuracy, 2.94% in precision and 2.47% in sensitivity. Despite this, most of the classification algorithms managed to maintain a good generalization capacity of results against the new unknown data that corresponds to the fifth day of operation.

In the case of fault detection, Figure 12 shows that the detection algorithms on average improved the fault detection rate compared to the results achieved in the test set. Furthermore, it is observed that the PCA and ICA algorithms not only presented the greatest improvement in this indicator, but also achieved the best performance in detecting faults in this data set corresponding

to the fifth day of operation.

However, when breaking down this result for each failure, Figure 13 shows that the PCA and ICA algorithms experienced an excessive increase in the detection of IDV15 and IDV16, with a decrease in the detection of normal operation. Similarly, the KPCA, AE, and SAE algorithms also experienced a decrease in detecting normal operation. Consequently, the implementation of these algorithms would bring about the problem of excessive false positives, since many data obtained under normal conditions would be incorrectly detected as a failure.

This result can be explained by a possible overfitting of the detection algorithms. Therefore, these algorithms characterized normal operation to certain specific patterns. So, when they were evaluated on unknown data from normal operation, they were most likely detected as failures. Similarly, the overfitting detection algorithms better detected the IDV15 and IDV16 perturbations, because these perturbations that caused a slight deviation from normal operation are easier to detect when normal operation was characterized with specific patterns.

Finally, Figure 14 shows that most of the regression algorithms lost their ability to generalize results when comparing the metrics achieved on the test set and fifth day of operation.

Despite this, the algorithm that showed the least deterioration was multivariable linear regression (LR). Furthermore, linear regression achieved the best performance on this data set corresponding to the fifth day of operation. However, the results achieved by the LR algorithm on the test set demonstrate that this algorithm could not model the phenomenon of the G generation rate in the reactor. So, these results would indicate that the LR algorithm is underfit to the problem; and underadjusted models do not capture all the relevant information about the phenomenon, but they present a better generalization of results when faced with unknown data.

On the other hand, the loss in the generalization of results observed by the non-linear regression algorithms (ANN, SVM-rbf, DT, RF and GB) presents a possible overfitting of the models. However, it must be noted that the performance achieved on the test set was also obtained on data not seen by the model. Therefore, the loss of generalization of results is due to the fact that the set of the fifth trading day contained new patterns that the algorithms could not generalize.

So, to overcome the problem observed in fault detection and virtual sensor, methods can be implemented that allow increasing the training data of the models. So, one way is to include some samples obtained from the fifth day of operation or apply a method that increases the data by sample repetition (bootstrapping method) (Emmert-Streib & Dehmer, 2019; Raschka, 2018).

CONCLUSIONS

In this work, different learning algorithms for fault classification, fault detection and virtual sensor were evaluated in the context of the development of a data science platform in the Tennessee Eastman process. The hypothesis raised in this work was rejected, because artificial neural networks only achieved the best performance in fault classification. In virtual sensor and fault detection applications there were other nonlinear algorithms that equaled or exceeded the performance achieved by algorithms based on artificial neural networks.

In general, several nonlinear algorithms simpler than the neural network managed to adequately model the phenomenon of each application, achieving the best value in the evaluation metrics of each application. In that sense, the support vector machine with radial

basis function kernel (SVM-rbf) is a good option to implement in fault classification in mode three of operation. Whereas, the k-nearest neighbors (k-NN) algorithm would be a good option to implement in the virtual sensor in both modes of operation. Meanwhile, model assembly algorithms, such as algorithms *Random Forest* (RF) and *Gradient Boosting* (GB), They are a good option to use in fault classification and virtual sensor.

In fault detection it was difficult to determine the most appropriate algorithm to implement, because all algorithms obtained similar performance. However, it is worth highlighting the effect of the size of the data set used on the results of the fault detection application. The synchronized subsample of the TE process has a greater Gaussian distribution and lower linearity, which explains the anomalous performance achieved by the principal component analysis (PCA). Furthermore, the small sample size could have affected the performance achieved by the auto-scramblers.

The effect of the size of the data set is also evident when comparing the generalizability of results on the fifth day of operation. In that sense, the generalization of the result was favored when the training set has a large amount of sample, for example, the classification of failures on the fifth day of operation experienced a slight deterioration. Meanwhile, the small amount of sample could affect the generalization of results, because possibly the fifth day of operation brings with it new samples that the algorithms cannot generalize.

Finally, the evaluation of the virtual sensor in the context of few training samples can be further explored. To this end, it is proposed to continue the evaluation of the virtual sensor using the data repetition method technique (bootstrapping method) in order to increase the training data.

GRATITUDE

This work was funded by ANID-PFCHA/ Doctorado Nacional/2018- 21180498.

REFERENCES

Acuña, G., Curilem, M., & Cubillos, F. (2014). Desarrollo de un sensor virtual basado en modelo NARMAX y máquina de vectores de soporte para molienda semiautógena. *RIAI - Revista Iberoamericana de Automatica e Informatica Industrial*, *11*(1), 109-116. https://doi.org/10.1016/j.riai.2013.09.008

Ayubi Rad, M. A., & Yazdanpanah, M. J. (2015). Designing supervised local neural network classifiers based on EM clustering for fault diagnosis of Tennessee Eastman process. *Chemometrics and Intelligent Laboratory Systems*, *146*, 149-157. https://doi.org/10.1016/j.chemolab.2015.05.013

Bathelt, A., Ricker, N. L., & Jelali, M. (2015). Revision of the Tennessee Eastman Process Model. *IFAC-PapersOnLine*, *48*(8), 309-314. https://doi.org/10.1016/j.ifacol.2015.08.199

Beck, D. A. C., Carothers, J. M., Subramanian, V. R., & Pfaendtner, J. (2016). Data science: Accelerating innovation and discovery in chemical engineering. *AIChE Journal*, *62*(5), 1402-1416. https://doi.org/10.1002/aic.15192

Braun, M., & Rivera, D. E. (1999). *Tennessee Eastman Process Control Test Problem Re-Written in MATLAB 5.2* [MATLAB]. Control Systems Engineering Laboratory at Arizona State University.

Chai, Z., & Zhao, C. (2020). Enhanced random forest with concurrent analysis of static and dynamic nodes for industrial fault classification. *IEEE Transactions on Industrial Informatics*, *16*(1), 54-66. https://doi.org/10.1109/TII.2019.2915559

Chiang, L., Braun, B., Wang, Z., & Castillo, I. (2022). Towards artificial intelligence at scale in the chemical industry. *AIChE Journal*, *68*(6). https://doi.org/10.1002/aic.17644

Chiang, L. H., Russell, E. L., & Braatz, R. D. (2001). *Fault Detection and Diagnosis in Industrial Systems*. Springer London.

Chiang, L., Lu, B., & Castillo, I. (2017). Big Data Analytics in Chemical Engineering. *Annual Review of Chemical and Biomolecular Engineering*, *8*(1), 63-85. https://doi.org/10.1146/annurev-chembioeng-060816-101555

Dorneanu, B., Zhang, S., Ruan, H., Heshmat, M., Chen, R., Vassiliadis, V. S., & Arellano-Garcia, H. (2022). Big data and machine learning: A roadmap towards smart plants. *Frontiers of Engineering Management*. https://doi.org/10.1007/s42524-022-0218-0

Downs, J. J., & Vogel, E. F. (1993). A plant-wide industrial process control problem. *Computers & Chemical Engineering*, *17*(3), 245-255. https://doi.org/10.1016/0098-1354(93)80018-I

Emmert-Streib, F., & Dehmer, M. (2019). Evaluation of Regression Models: Model Assessment, Model Selection and Generalization Error. *Machine Learning and Knowledge Extraction*, *1*(1), 521-551. https://doi.org/10.3390/make1010032

Fazai, R., Taouali, O., Harkat, M. F., & Bouguila, N. (2016). A new fault detection method for nonlinear process monitoring. *The International Journal of Advanced Manufacturing Technology*, *87*(9-12), 3425-3436. https://doi.org/10.1007/s00170-016-8745-7

Ge, Z. (2018). Distributed predictive modeling framework for prediction and diagnosis of key performance index in plant-wide processes. *Journal of Process Control*, *65*, 107-117. https://doi.org/10.1016/j.jprocont.2017.08.010

Ge, Z., & Song, Z. (2013). *Multivariate Statistical Process Control*. Springer London.

Ge, Z., Song, Z., & Gao, F. (2013). Review of Recent Research on Data-Based Process Monitoring. *Industrial & Engineering Chemistry Research*, *52*(10), 3543-3562. https://doi.org/10.1021/ie302069q

He, Y. L., Geng, Z. Q., & Zhu, Q. X. (2015). Data driven soft sensor development for complex chemical processes using extreme learning machine. *Chemical Engineering Research and Design*, *102*, 1-11. https://doi.org/10.1016/j.cherd.2015.06.009

He, Y. L., Geng, Z. Q., & Zhu, Q. X. (2016). Soft sensor development for the key variables of complex chemical processes using a novel robust bagging nonlinear model integrating improved extreme learning machine with partial least square. *Chemometrics and Intelligent Laboratory Systems*, *151*, 78-88. https://doi.org/10.1016/j.chemolab.2015.12.010

Heo, S., & Lee, J. H. (2018). Fault detection and classification using artificial neural networks. *IFAC-PapersOnLine*, *51*(18), 470-475.

Isermann, R. (2006). *Fault-diagnosis systems: An introduction from fault detection to fault tolerance*. Springer.

Jing, C., & Hou, J. (2015). SVM and PCA based fault classification approaches for complicated industrial process. *Neurocomputing*, *167*, 636-642. https://doi.org/10.1016/j.neucom.2015.03.082

Kabugo, J. C., Jämsä-Jounela, S.-L., Schiemann, R., & Binder, C. (2020). Industry 4.0 based process data analytics platform: A waste-to-energy plant case study. *International Journal of Electrical Power & Energy Systems*, *115*, 105508. https://doi.org/10.1016/j.ijepes.2019.105508

Kadlec, P., Gabrys, B., & Strandt, S. (2009). Data-driven Soft Sensors in the process industry. *Computers and Chemical Engineering*, *33*(4), 795-814. https://doi.org/10.1016/j.compchemeng.2008.12.012

Kim, J.-Y. (2017). *Smart chemical plant architecture development based on a systems engineering*. 1-5. https://doi.org/10.1109/SysEng.2017.8088315

Kwon, H., Oh, K. C., Choi, Y., Chung, Y. G., & Kim, J. (2021). Development and application of machine learning-based prediction model for distillation column. *International Journal of Intelligent Systems*, *36*(5), 1970-1997. https://doi.org/10.1002/int.22368

Lee, J. Y., Yoon, J. S., & Kim, B.-H. (2017). A big data analytics platform for smart factories in small and medium-sized manufacturing enterprises: An empirical case study of a die casting factory. *International Journal of Precision Engineering and Manufacturing*, *18*(10), 1353-1361. https://doi.org/10.1007/s12541-017-0161-x

Lei, J., Liu, C., & Jiang, D. (2019). Fault diagnosis of wind turbine based on Long Short-term memory networks. *Renewable Energy*, *133*, 422-432. https://doi.org/10.1016/j.renene.2018.10.031

Li, S., Luo, J., & Hu, Y. (2020). Semi-supervised process fault classification based on convolutional ladder network with local and global feature fusion. *Computers and Chemical Engineering*, *140*, 106843. https://doi.org/10.1016/j.compchemeng.2020.106843

Lin, Y. C., Hung, M. H., Huang, H. C., Chen, C. C., Yang, H. C., Hsieh, Y. S., & Cheng, F. T. (2017). Development of Advanced Manufacturing Cloud of Things (AMCoT)-A Smart Manufacturing Platform. *IEEE Robotics and Automation Letters*, *2*(3), 1809-1816. https://doi.org/10.1109/LRA.2017.2706859

Liu, Y., & Ge, Z. (2018). Weighted random forests for fault classification in industrial processes with hierarchical clustering model selection. *Journal of Process Control*, *64*, 62-70. https://doi.org/10.1016/j.jprocont.2018.02.005

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*, 100216. https://doi.org/10.1016/j.jii.2021.100216

Loy-Benitez, J., Li, Q., Nam, K., & Yoo, C. (2020). Sustainable subway indoor air quality monitoring and fault-tolerant ventilation control using a sparse autoencoder-driven sensor self-validation. *Sustainable Cities and Society*, *52*, 101847. https://doi.org/10.1016/j.scs.2019.101847

Lv, F., Wen, C., Bao, Z., & Liu, M. (2016). Fault diagnosis based on deep learning. *Proceedings of the American Control Conference*, *2016-July*(2), 6851-6856. https://doi.org/10.1109/ACC.2016.7526751

Meng, Y., Lan, Q., Qin, J., Yu, S., Pang, H., & Zheng, K. (2019). Data-driven soft sensor modeling based on twin support vector regression for cane sugar crystallization. *Journal of Food Engineering*, *241*(June 2017), 159-165. https://doi.org/10.1016/j.jfoodeng.2018.07.035

National Academies of Sciences Engineering and Medicine. (2018). *Data Science: Opportunities to Transform Chemical Sciences and Engineering: Proceedings of a Workshop in Brief* (L. Casola & E. Mantus, Eds.; Vol. 60, pp. 285-286). National Academies Press. https://doi.org/10.17226/25191

Neubürger, F., Saeid, Y., & Kopinski, T. (2021). *Variational-Autoencoder Architectures for Anomaly Detection in Industrial Processes*.

Omar, A. M. S., Osman, M. K., Ibrahim, M. N., Hussain, Z., & Abidin, A. F. (2020). Fault classification on transmission line using LSTM network. *Indonesian Journal of Electrical Engineering and Computer Science*, *20*(1), 231-238. https://doi.org/10.11591/ijeecs.v20.i1.pp231-238

Park, Y.-J., Fan, S.-K. S., & Hsu, C.-Y. (2020). A Review on Fault Detection and Process Diagnostics in Industrial Processes. *Processes*, *8*(9), Article 9. https://doi.org/10.3390/pr8091123

Qin, S. J. (2012). Survey on data-driven industrial process monitoring and diagnosis. *Annual Reviews in Control*, *36*(2), 220-234. https://doi.org/10.1016/j.arcontrol.2012.09.004

Qin, S. J. (2014). Process data analytics in the era of big data. *AIChE Journal*, *60*(9), 3092-3100. https://doi.org/10.1002/aic.14523

Qiu, Y., & Dai, Y. (2019). A Stacked Auto-Encoder Based Fault Diagnosis Model for Chemical Process. En *Computer Aided Chemical Engineering* (Vol. 46, pp. 1303-1308). Elsevier. https://doi.org/10.1016/B978-0-12-818634-3.50218-6

Quiñones-Grueiro, M., Llanes-Santiago, O., & Neto, A. J. S. (2020). *Monitoring Multimode Continuous Processes: A Data-Driven Approach* (Vol. 309). Springer Nature.

Quiñones-Grueiro, M., Llanes-Santiago, O., & Silva Neto, A. J. (2021). Fault Classification with Data-Driven Methods. En M. Quiñones-Grueiro, O. Llanes-Santiago, & A. J. Silva Neto, *Monitoring Multimode Continuous Processes* (Vol. 309, pp. 99-122). Springer International Publishing. https://doi.org/10.1007/978-3-030-54738-7_5

Rajaraman, V. (2016). Big data analytics. *Resonance*, *21*(8), 695-716. https://doi.org/10.1007/s12045-016-0376-7

Raschka, S. (2018). *Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning*.

Ricker, N. L. (1996). Decentralized control of the Tennessee Eastman Challenge Process. *Journal of Process Control*, *6*(4), 205-221. https://doi.org/10.1016/0959-1524(96)00031-5

Ricker, N. L. (2015). *Tennessee Eastman Challenge Archive*. https://depts.washington.edu/control/LARRY/TE/download.html#Updated_TE_Code

Samuel, R. T., & Cao, Y. (2016). Nonlinear process fault detection and identification using kernel PCA and kernel density estimation. *Systems Science & Control Engineering*, *4*(1), 165-174. https://doi.org/10.1080/21642583.2016.1198940

Shang, C., Huang, X., Suykens, J. A. K., & Huang, D. (2015). Enhancing dynamic soft sensors based on DPLS: A temporal smoothness regularization approach. *Journal of Process Control*, *28*, 17-26. https://doi.org/10.1016/j.jprocont.2015.02.006

Shang, C., Yang, F., Huang, D., & Lyu, W. (2014). Data-driven soft sensor development based on deep learning technique. *Journal of Process Control*, *24*(3), 223-233. https://doi.org/10.1016/j.jprocont.2014.01.012

Souza, F. A. A., Araújo, R., & Mendes, J. (2016). Review of soft sensor methods for regression applications. *Chemometrics and Intelligent Laboratory Systems*, *152*, 69-79. https://doi.org/10.1016/j.chemolab.2015.12.011

Sun, W., Paiva, A. R. C., Xu, P., Sundaram, A., & Braatz, R. D. (2020). Fault detection and identification using Bayesian recurrent neural networks. *Computers and Chemical Engineering*, *141*, 106991. https://doi.org/10.1016/j.compchemeng.2020.106991

Voigt, T., Migenda, N., Schone, M., Pelkmann, D., Fricke, M., Schenck, W., & Kohlhase, M. (2021). *Advanced Data Analytics Platform for Manufacturing Companies*. *2021-September*, 01-08. https://doi.org/10.1109/ETFA45728.2021.9613499

Wang, L., Jin, H., Chen, X., Dai, J., Yang, K., & Zhang, D. (2016). Soft Sensor Development Based on the Hierarchical Ensemble of Gaussian Process Regression Models for Nonlinear and Non-Gaussian Chemical Processes. *Industrial and Engineering Chemistry Research*, *55*(28), 7704-7719. https://doi.org/10.1021/acs.iecr.6b00240

Wu, F., Yin, S., & Karimi, H. R. (2014). Fault detection and diagnosis in process data using support vector machines. *Journal of Applied Mathematics*, *2014*. https://doi.org/10.1155/2014/732104

Wu, H., & Zhao, J. (2018). Deep convolutional neural network model based chemical process fault diagnosis. *Computers and Chemical Engineering*, *115*, 185-197. https://doi.org/10.1016/j.compchemeng.2018.04.009

Yan, W., Guo, P., gong, L., & Li, Z. (2016). Nonlinear and robust statistical process monitoring based on variant autoencoders. *Chemometrics and Intelligent Laboratory Systems*, *158*, 31-40. https://doi.org/10.1016/j.chemolab.2016.08.007

Yin, S., Ding, S. X., Haghani, A., Hao, H., & Zhang, P. (2012). A comparison study of basic data-driven fault diagnosis and process monitoring methods on the benchmark Tennessee Eastman process. *Journal of Process Control*, *22*(9), 1567-1581. https://doi.org/10.1016/j.jprocont.2012.06.009

Yin, S., Ding, S. X., Xie, X., & Luo, H. (2014). A review on basic data-driven approaches for industrial process monitoring. *IEEE Transactions on Industrial Electronics*, *61*(11), 6414-6428. https://doi.org/10.1109/TIE.2014.2301773

Yuan, X., Li, L., & Wang, Y. (2020). Nonlinear Dynamic Soft Sensor Modeling with Supervised Long Short-Term Memory Network. *IEEE Transactions on Industrial Informatics*, *16*(5), 3168-3176. https://doi.org/10.1109/TII.2019.2902129

Zhang, C., Yu, J., & Ye, L. (2021). Sparsity and manifold regularized convolutional auto-encoders-based feature learning for fault detection of multivariate processes. *Control Engineering Practice*, *111*, 104811. https://doi.org/10.1016/j.conengprac.2021.104811

Zhang, F., & Ge, Z. (2015). Decision fusion systems for fault detection and identification in industrial processes. *Journal of Process Control*, *31*, 45-54. https://doi.org/10.1016/j.jprocont.2015.04.004

Zhang, Y. (2009). Enhanced statistical analysis of nonlinear processes using KPCA, KICA and SVM. *Chemical Engineering Science*, *64*(5), 801-811. https://doi.org/10.1016/j.ces.2008.10.012

Zhang, Z., Jiang, T., Li, S., & Yang, Y. (2018). Automated feature learning for nonlinear process monitoring – An approach using stacked denoising autoencoder and k-nearest neighbor rule. *Journal of Process Control*, *64*, 49-61. https://doi.org/10.1016/j.jprocont.2018.02.004

Zhang, Z., & Zhao, J. (2017). A deep belief network based fault diagnosis model for complex chemical processes. *Computers and Chemical Engineering*, *107*, 395-407. https://doi.org/10.1016/j.compchemeng.2017.02.041

Zhao, H., Sun, S., & Jin, B. (2018). Sequential Fault Diagnosis Based on LSTM Neural Network. *IEEE Access*, *6*, 12929-12939. https://doi.org/10.1109/ACCESS.2018.2794765

Zhongda, T., Shujiang, li, Yanhong, W., & Xiangdong, W. (2016). A multi-model fusion soft sensor modelling method and its application in rotary kiln calcination zone temperature prediction. *Transactions of the Institute of Measurement and Control*, *38*(1), 110-124. https://doi.org/10.1177/0142331215573099

FIGURES

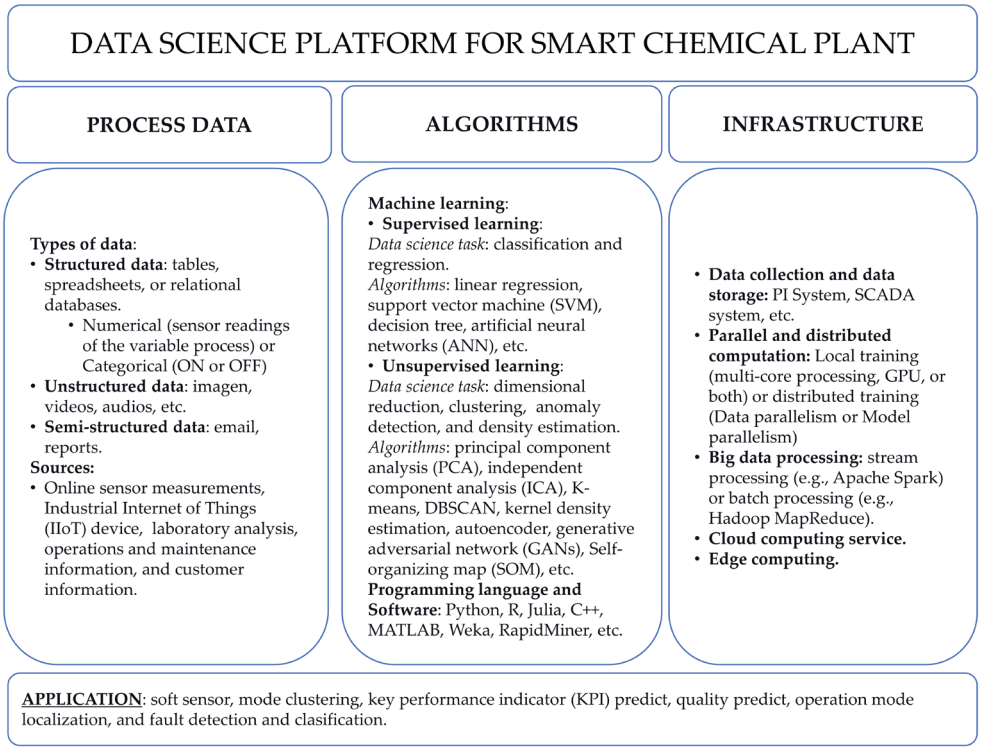


Figure 1 Fundamental elements of a data science platform for the smart chemical industry.

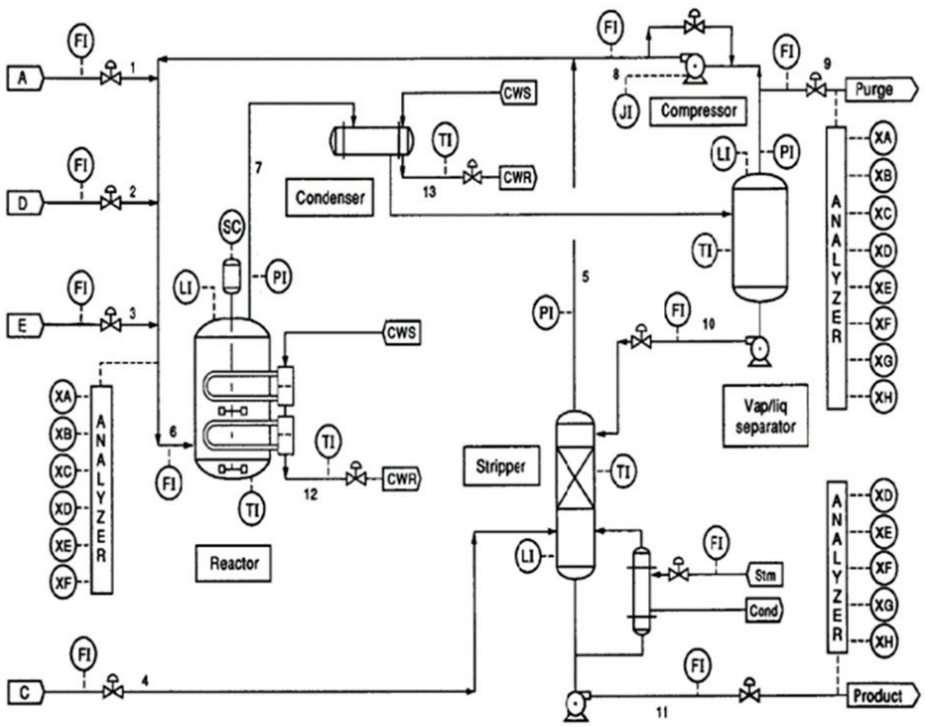


Figure 2 Tennessee Eastman Process Flow Chart.

Note. Extracted from "The plant-wide industrial process control problem", by J. Downs and E. Vogel, 1993, *Computers and Chemical Engineering*, 17(3), p. 246. Copyright © 1993 by Published by Elsevier Ltd.

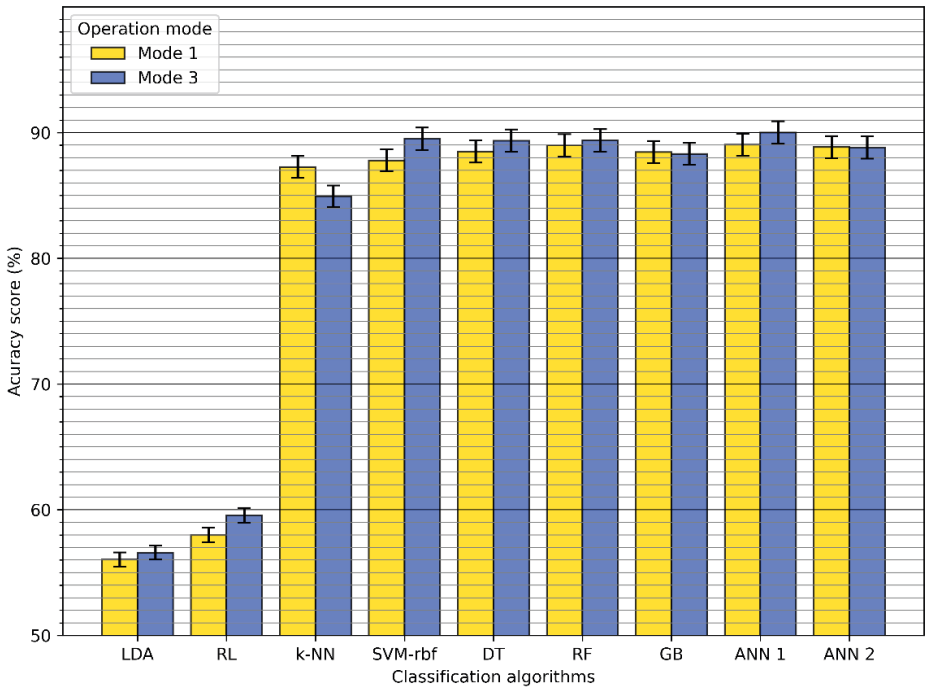


Figure 3. Accuracy obtained by the algorithms in the classification of failures in the Tennessee Eastman process.

Note. Results obtained on the test set. The error bar is the 95% confidence interval.

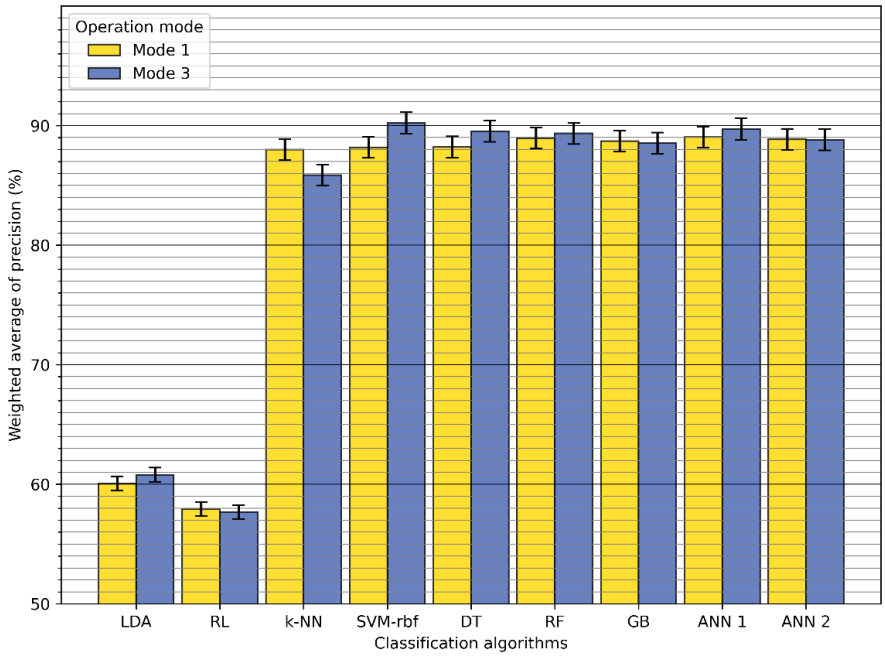


Figure 4. Average precision obtained by the algorithms in the classification of failures in the Tennessee Eastman process.

Note. Results obtained on the test set. The error bar is the 95% confidence interval.

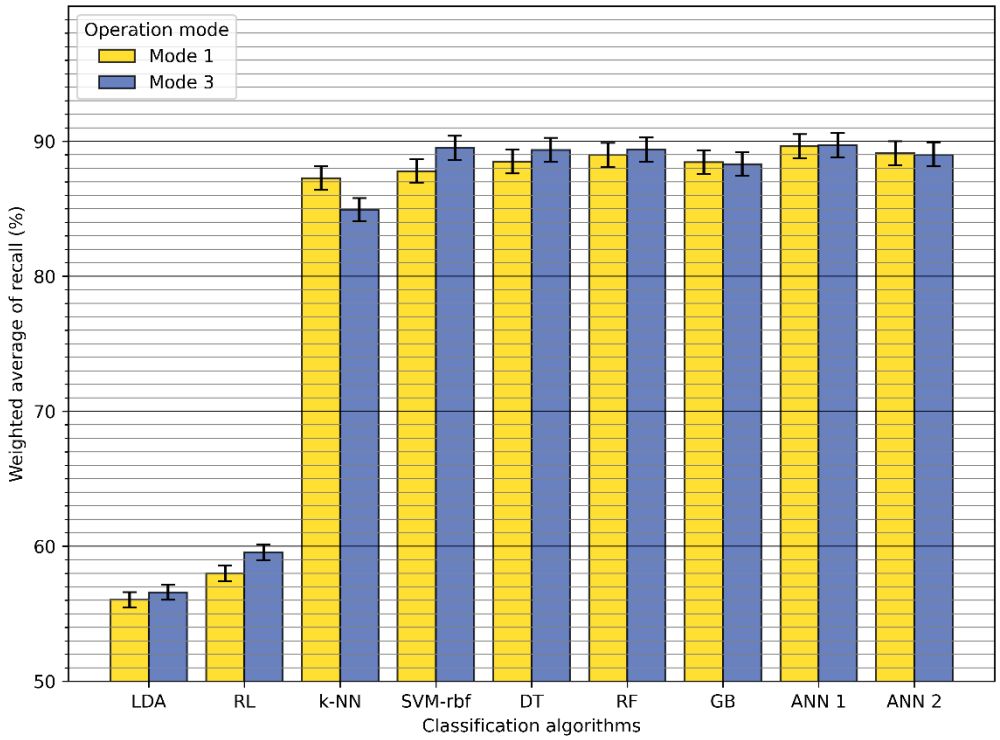
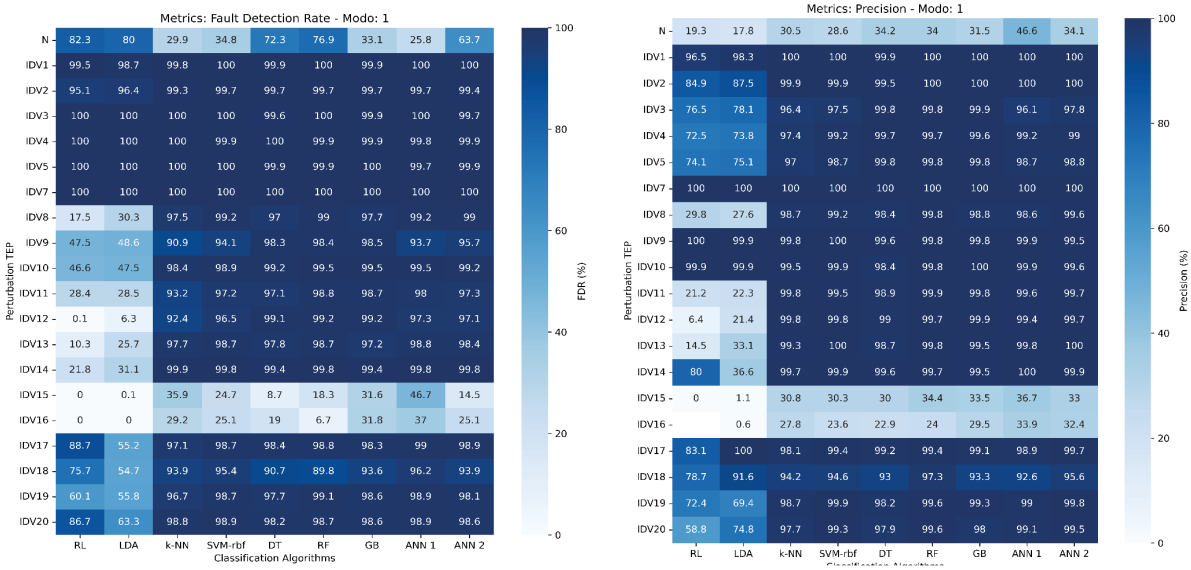
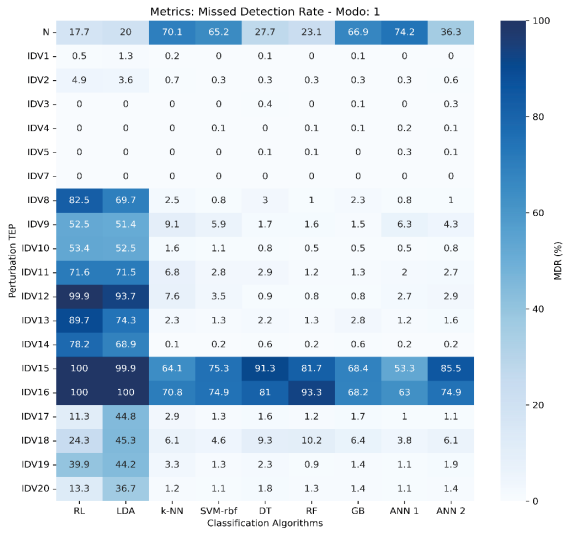


Figure 5 Average sensitivity obtained by the algorithms in the classification of failures in the Tennessee Eastman process.

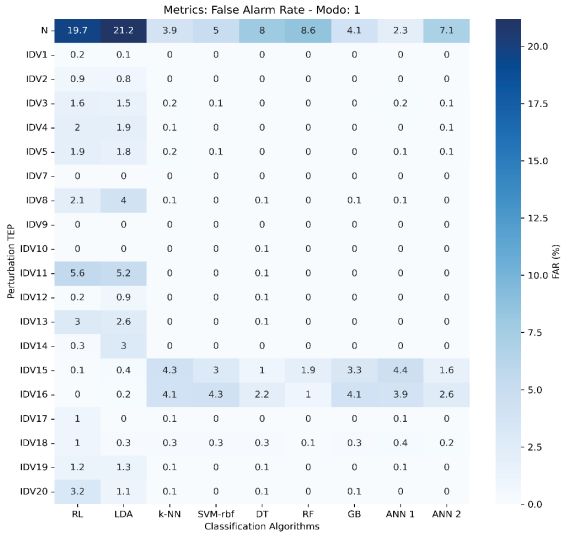


(a) Detection rate in mode one

(b) Mode one precision

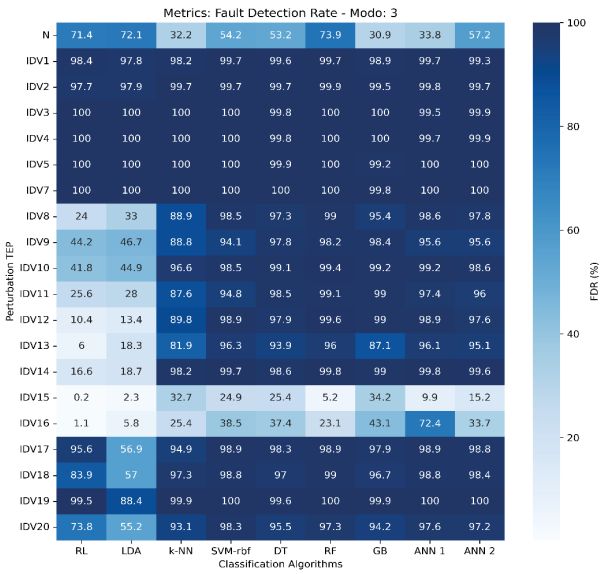


(c) Missed detection rate in mode one

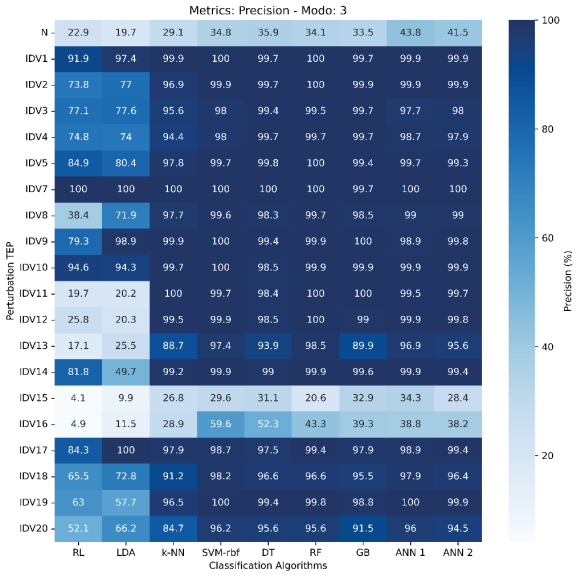


(d) False alarm rate in mode one

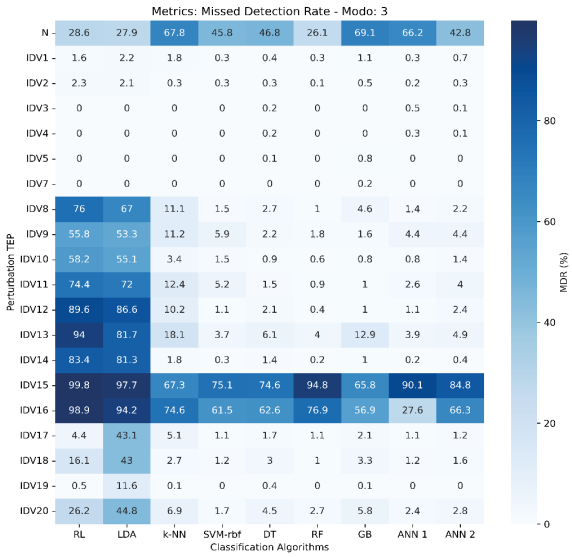
Figure 6 Evaluation of the performance achieved by each algorithm in each disturbance of the Tennessee Eastman process in mode one of operation.



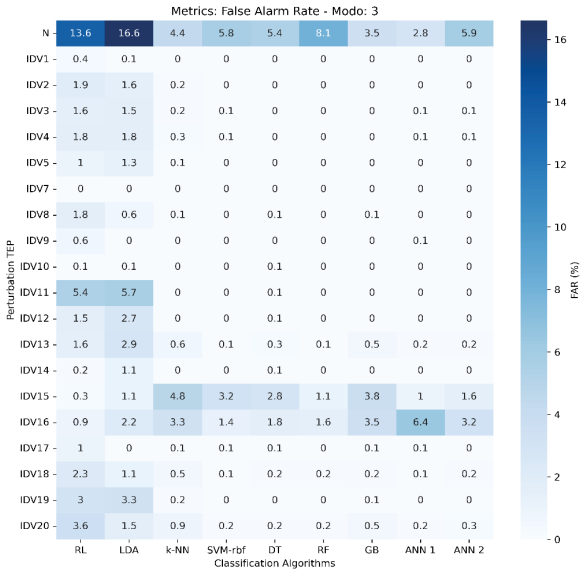
(a) Detection rate in mode three



(b) Mode three precision

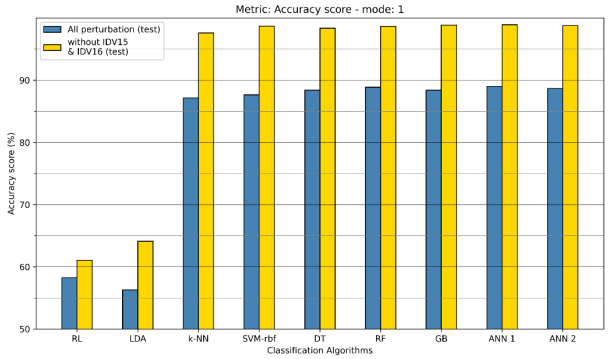


(c) Lost detection rate in mode three

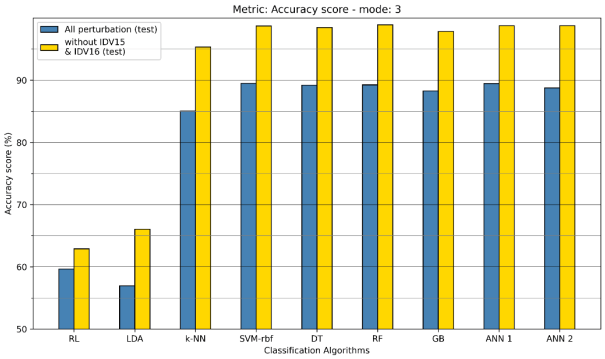


(d) False alarm rate in mode three

Figure 7 Evaluation of the performance achieved by each algorithm in each perturbation of the Tennessee Eastman process in mode one of operation.



(a) Accuracy in mode one of operation



(b) Accuracy in mode three of operation

Figure 8 Comparison of the accuracy obtained in the fault classification of the Tennessee Eastman process without disturbances IDV15 and IDV16.

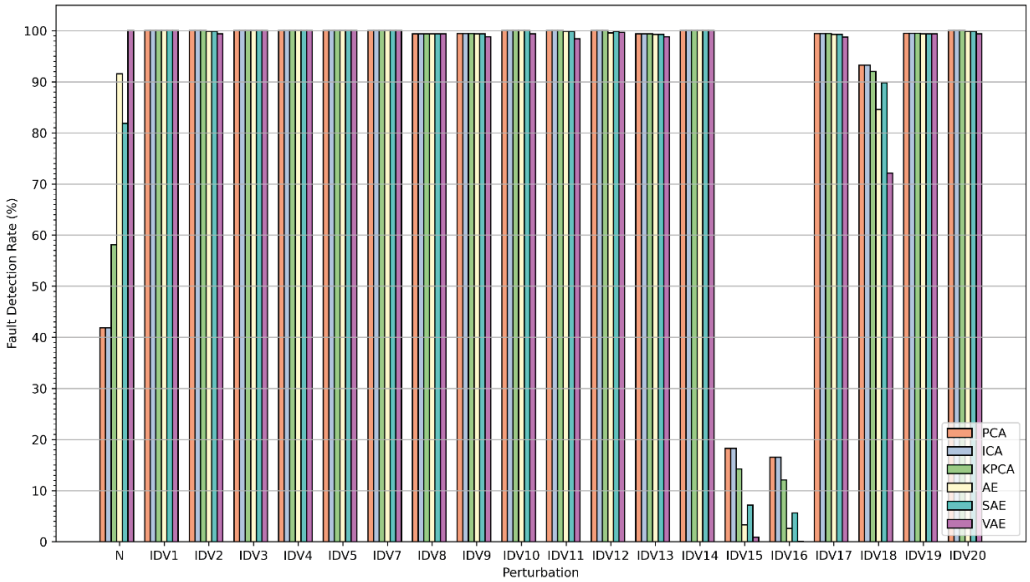


Figure 9 Failure detection rate obtained by different algorithms in mode one of operation of the Tennessee Eastman process.

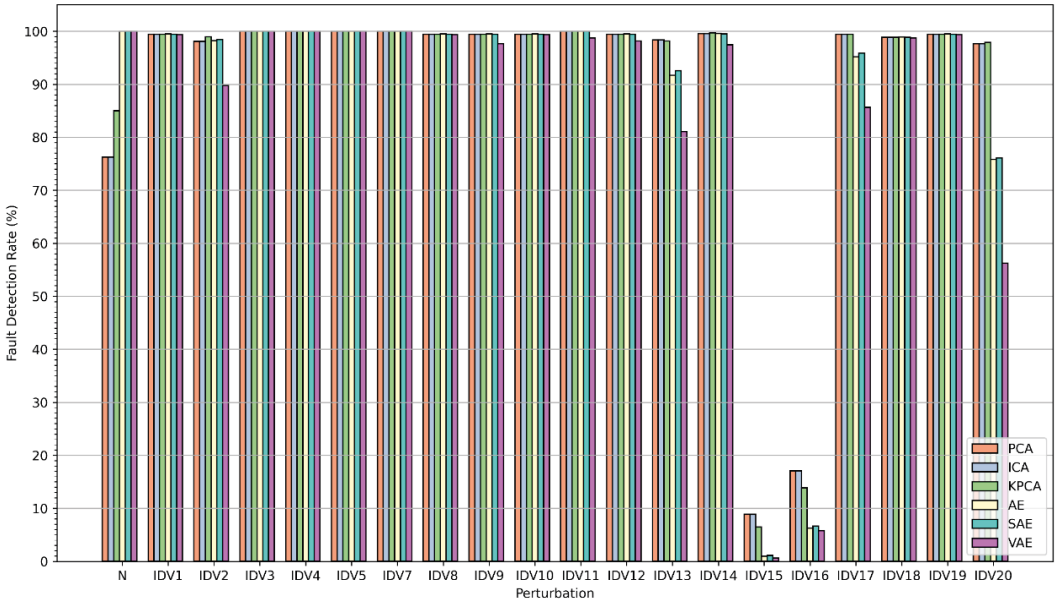


Figure 10 Failure detection rate obtained by different algorithms in mode three of operation of the Tennessee Eastman process.

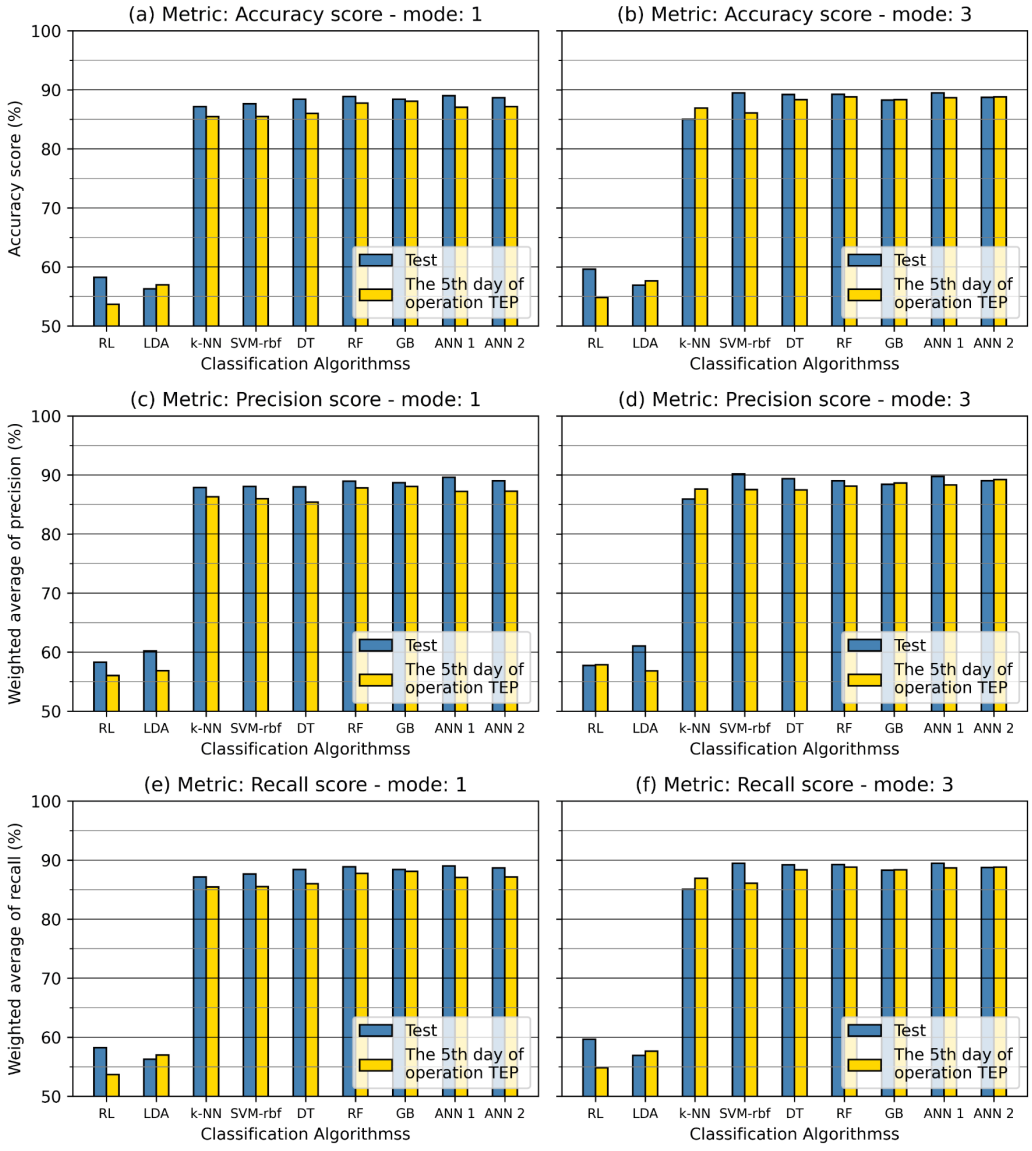


Figure 11 Comparison of the generalization of results experienced by the algorithms in the classification of failures in the Tennessee Eastman process.

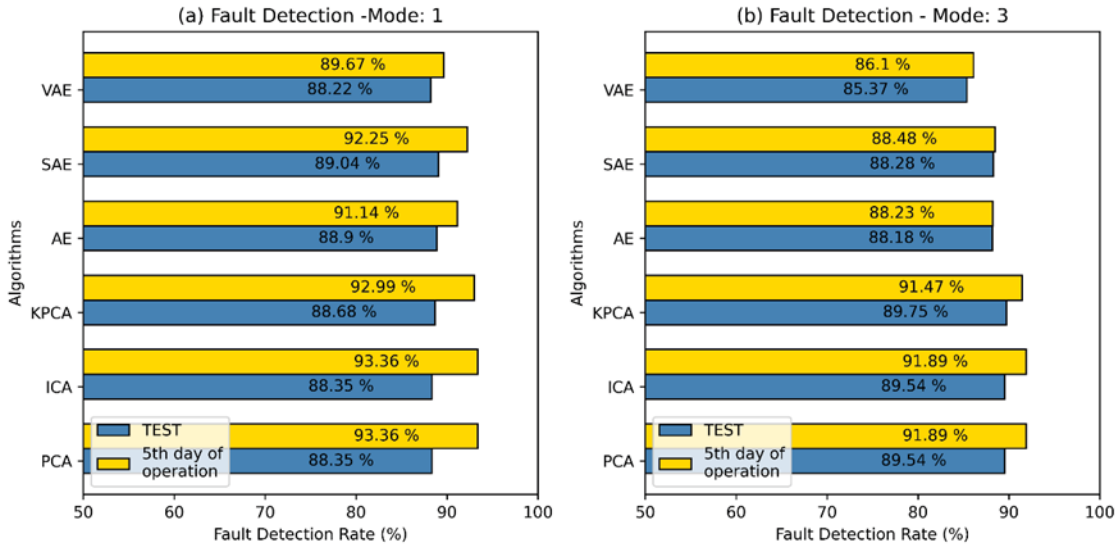
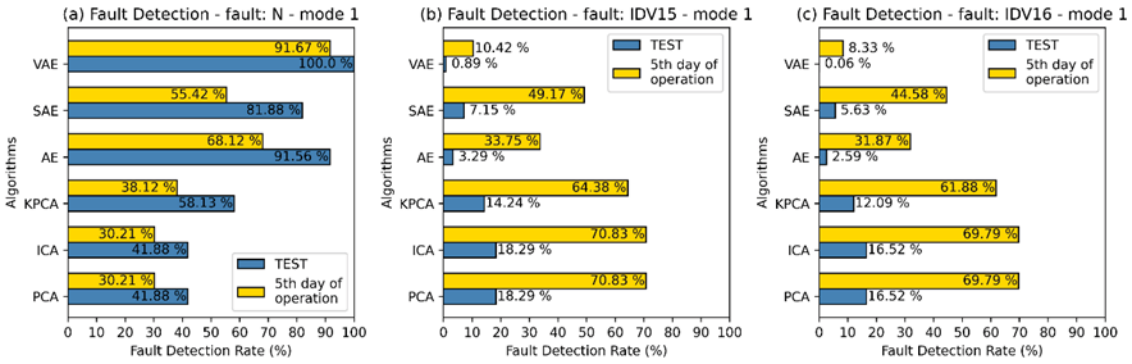
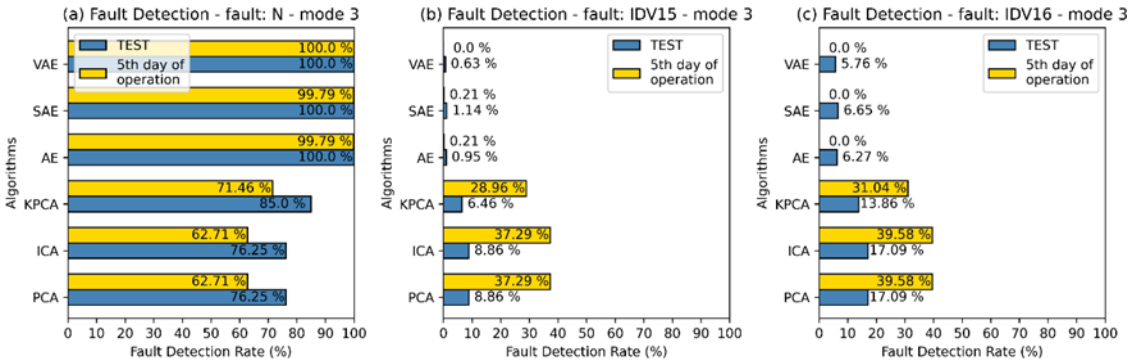


Figure 12 *Comparison in the generalization of results obtained by the fault detection algorithms on the fifth day of operation of the Tennessee Eastman process.*



(a) Mode one of operation of the TE process



(b) Mode three of operation of the TE process

Figure 13 Evaluation of the generalization of the results of the detection algorithms in normal operation and IDV15 and IDV16 disturbances in mode one and three of operation.

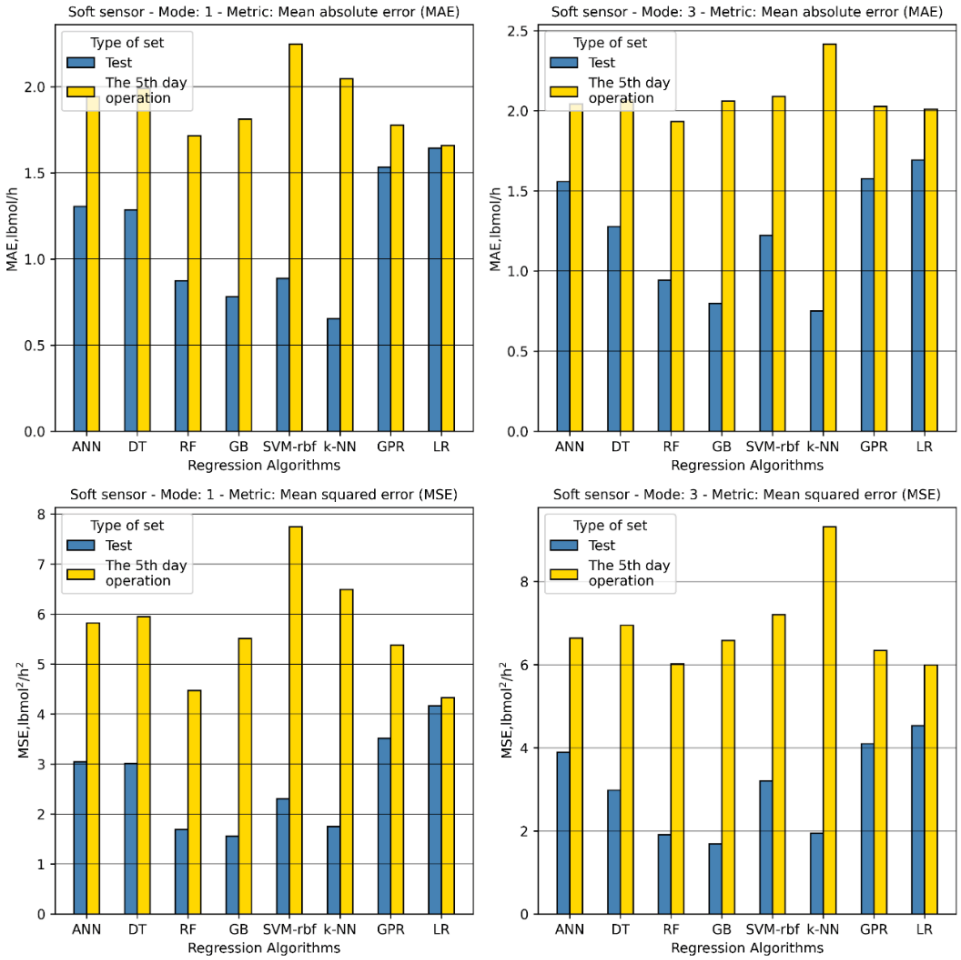


Figure 14 Comparison of the generalization of results experienced by the regression algorithms in the virtual sensor in the Tennessee Eastman process.

TABLES

ID	Process variable	Type
IDV1	A/B ratio, constant B composition (current 4)	Step
IDV2	Composition B, constant A/B ratio (current 4)	Step
IDV3	DE supply temperature (stream 2)	Step
IDV4	Reactor cooling water inlet temperature	Step
IDV5	Condenser cooling water inlet temperature	Step
IDV6	Power A loss (current 1)	Step
IDV7	C header pressure loss - availability reduction (stream 4)	Step
IDV8	Composition of feed A, B and C (current 4)	Random variation
IDV9	Feed temperature D (stream 2)	Random variation
IDV10	Feed temperature C (current 4)	Random variation
IDV11	Reactor cooling water inlet temperature	Random variation
IDV12	Condenser cooling water inlet temperature	Random variation
IDV13	Reaction kinetics	slow drift
IDV14	reactor cooling water valve	*Sticking*

IDV15	Condenser cooling water valve	*Sticking*
IDV16	Unknown	Unknown
IDV17	Unknown	Unknown
IDV18	Unknown	Unknown
IDV19	Unknown	Unknown
IDV20	Unknown	Unknown

Table 1 Disturbances of the Tennessee Eastman process.

Note. Extracted from "A plant-wide industrial process control problem", by J. Downs and E. Vogel, 1993, *Computers and Chemical Engineering*, 17(3), p. 250. Copyright © 1993 by Published by Elsevier Ltd.

Column number	Column name	Queue number	Counting non-null values	Variable type
0 – 21[a]	XMEAS1 – XMEAS22	0 a 480,039	480,040	Numeric
22 – 40[b]	XMEAS23 – XMEAS41	0 a 480,039	480,040	Numeric
41 – 48[a]	XMEAS42 – XMEAS49	0 a 480,039	480,040	Numeric
49 – 72[b]	XMEAS50-XMEAS73	0 a 480,039	480,040	Numeric
73 – 84[c]	XMV1 - XMV12	0 a 480,039	480,040	Numeric
85	Mode	0 a 480,039	480,040	Categorical
86	Perturbation	0 a 480,039	480,040	Categorical
87	Time	0 a 480,039	480,040	Numeric

Table 2 Tennessee Eastman Process Database Summary

Note. [a]: Continuous measured variables.

[b]: Sampled type measured variables (composition of TE process streams).

[c]: Manipulated variables of the TE process.

	Failure classification	
	Input variables	Objective variable
Mode 1	XMEAS1, XMEAS10, XMEAS11, XMEAS18, XMEAS21, XMEAS22, XMEAS43, XMEAS45, XMEAS46, XMEAS47, XMEAS48, XMV3, XMV4, XMV6, XMV7, XMV8, XMV10	Process Disturbance TE (and = {N, IDV1, IDV2, IDV3 …, IDV17, IDV18, IDV19, IDV20})
Mode 3	XMEAS1, XMEAS10, XMEAS11, XMEAS18, XMEAS20, XMEAS21, XMEAS22, XMEAS43, XMEAS45, XMEAS46, XMEAS47, XMEAS48, XMEAS49, XMV3, XMV4, XMV6, XMV10, XMV11	
	virtual sensor	
	Input variables	Objective variable
Mode 1	XMEAS17, XMEAS10, XMV6, XMEAS44, XMEAS2, XMEAS42, XMEAS19, XMEAS3, XMEAS14, XMEAS12, XMEAS8	Generation rate of product G in the reactor
Mode 3	XMEAS17, XMEAS44, XMEAS2, XMEAS42	
	Fault Detection	
	Input variables	Objective variable
Mode 1	XMEAS1, XMEAS2, …, XMEAS20, XMEAS21, XMEAS22, XMEAS42, XMEAS43, XMEAS44, XMEAS47, XMEAS48, XMEAS49, XMV1, XMV2, XMV3, …, XMV9, XMV10, XMV11, XMV12	
Mode 3		

Table 3 Feature Selection in Data Science Applications in the Tennessee Eastman Process.

Algorithm	Failure classification (Multi-class classification)	Virtual sensor (Regression)
Linear Discriminant Analysis (LDA)	X	
Logistic regression (RL)	X	
Multivariable Linear Regression (LR)		X
Gaussian Process Regression (GPR)		X
K nearest neighbors (KNN)	X	X
Support Vector Machines with Radial Basis Kernel Function (SVM)	X	X
Decision tree (DT)	X	
Random Forest (RF)	X	X
Gradient Boosting (GB)	X	X
Artificial Neural Networks (ANN)	X	X

Table 4 Data Science Algorithms Used in Fault Classification and Virtual Sensor Applications in Tennessee Eastman Process.

Algorithm training

1. Select the instances under normal operation from the training set.
2. Normalize normal operating data by its mean and variance.
3. Train unsupervised algorithms with normal operation data.
4. Project normal operation training data into low-dimensional space.
5. Reconstruct training data from low-dimensional space to original space(X).
6. Calculate the residual between the original observation in normal operation (X) and the reconstruction ($\hat{X}$) using the equation (7.

$$E = X - \hat{X} \quad (8)$$

7. Calculate the SPE of the training instance under normal operation using the equation (9.

$$SPE_i = E_i \cdot E_i^T \quad (9)$$

8. 8. Calculate the limit SPE of normal operation using equation (7, assuming that this statistic follows a Chi-square type distribution ($\chi^2_{h,\alpha}$) in the case of PCA and ICA.

$$SPE_{Lim} = g\chi^2_{h,\alpha} \quad (10)$$

Where, α is the level of significance (in this work 1% was used) $g = v/(2m)$, $h = 2m^2/v$, m is the mean of the SPE values calculated at point 5 and v is the variance of the SPE values calculated at point 5.
In the case of the KPCA, AE, SAE and VAE it was assumed that the SPE statistic does not have a Chi-square distribution. Therefore, kernel density estimation (KDE) was used to determine SPE limit.

Fault Detection

9. Normalize the new observation.
10. Project the new observation (X_New) to low-dimensional space using unsupervised algorithms.
11. Reconstruct the projection of the new observation from the low-dimensional space to the original space(X_{New}).
12. Calculate the residual between the new observation(X_{New}) and reconstruction of the projection (X_{New}) through the equation (8.
13. Calculate the SPE of the new observation ($SPE_{i, New}$) through the equation (9.
14. Fault detection in new observation.

- Si $SPE_{i, New}$ SPE_{Lim}, then the new observation is detected as a fault.
- Si $SPE_{i, New}$ SPE_{Lim}, then the new observation is detected as a normal operation.

Table 5. Failure detection procedure by unsupervised algorithms.

		Predicted class		
		YEAH	NO	Total
Class true	YEAH	True positive (TP)	False negative (FN)	Positive (P)
	NO	False positive (FP)	True negative (TV)	Negative (N)
	Total	Predicted Positives (“P")	Predicted Negatives (“N”)	P + N

Table 6. Confusion matrix for binary fault classification.

Mode 1						
	PCA	ICA	KPCA	AE	SAE	VAE
FDR	88.3501 %	88.3501 %	88.6784 %	88.9041 %	**89.0431 %**	88.2184 %
MDR	11.6499%	11.6499%	11.3216%	11.0959%	**10.9569%**	11.7816%
Mode 3						
	PCA	ICA	KPCA	AE	SAE	VAE
FDR	89.5372 %	89.5372 %	**89.7468 %**	88.1836 %	88.2817 %	85.3703 %
MDR	10.4628 %	10.4628 %	**10.2532 %**	11.8164 %	11.7183 %	14.6297 %

Table 7 Average detection rate and missed detection rate obtained by data science algorithms in fault detection in Tennessee Eastman process.

Note. The results in this table are calculated based on the test set.

Metrics	MSE [lb-mol²/min²]	
Algorithm	Mode 1 ***	Mode 3 ***
ANN	3.0470±0.0559	3.8958±0.1744
DT	3.0131±0.0981	2.9830±0.0921
RF	1.6934±0.0279	1.9074±0.0320
GB	**1.5596±0.0685**	**1.6875±0.0517**
SVM-rbf	2.3054±0.0469	3.2051±0.0372
k-NN	1.7502±0.0254	1.9429±0.0437
GPR	3.5153±0.0640	4.0991±0.0936
LR	4.1642±0.0066	4.5351±0.0041

Table 8 Mean Square Error (MSE) obtained by each regression algorithm in the Tennessee Eastman virtual process sensor.

Note. The results in this table are calculated based on the test set. $\bar{x}$ = average, s = Deviation standard, n = number of iterations of the cross validation method.

***: confidence intervals calculated at 95% confidence.

Metrics	MAE [lb-mol/min]	
Algorithm	Mode 1[a]	Mode 3[a]
ANN	1.3055±0.0188	1.5583±0.0387
DT	1.2850±0.0186	1.2779±0.0202
RF	0.8741±0.0056	0.9430±0.0069
GB	0.7801±0.0141	0.7975±0.0130
SVM-rbf	0.8880±0.0093	1.2226±0.0059
k-NN	**0.6539±0.0060**	**0.7511±0.0097**
GPR	1.5334±0.0322	1.5764±0.0301
LR	1.6434±0.0011	1.6923±0.0007

Table 9 Mean absolute error (MAE) obtained by each regression algorithm in the Tennessee Eastman virtual process sensor.

Note. The results in this table are calculated based on the test set. $\bar{x}$ = average, s = Standard deviation, n = number of iterations of the cross validation method. ***: confidence intervals calculated at 95% confidence.