Application of unsupervised machine learning techniques to evaluate control performance in PID control loops ⋆

Tainara Marques * **Gilberto Reynoso-Meza** *,** **Jesús Carrillo-Ahumada** ***

* *Industrial and Systems Engineering Graduate Program (PPGEPS), Pontifícia Universidade Católica do Paraná (PUCPR), Rua Imaculada Conceição 1155, Curitiba, Brazil. (e-mail: g.reynosomeza@pucpr.br).*
** *Control Systems Optimization Laboratory (LOSC), Pontifícia Universidade Católica do Paraná (PUCPR).*
*** *Ingeniería en Alimentos. Universidad del Papaloapan. Circuito Central 200, Col. Parque Industrial, 68301 Tuxtepec, Oaxaca, Mex (e-mail: jcarrillo@unpa.edu.mx)*

Abstract: Seeking to ensure safe and reliable operation in industrial processes, Control Performance Assessment (CPA) has been employed in control systems to ensure they operate effectively and efficiently. Traditionally, CPA has been approached using control performance indicators. However, the integration of data science and unsupervised machine learning has emerged as a promising alternative for classification tasks related to CPA. This study is a continuation of previous research, exploring the use of unsupervised machine learning for CPA classification. We utilized a database describing 30 control performance indicators in a PID control loop. The results reveal that the effectiveness of unsupervised learning technique was not as significant as the approach in the previous study. This suggests the need to investigate alternative approaches or adjust model parameters to enhance its performance in CPA classification.

Keywords: Control performance assessment; unsupervised machine learning; PID control.

1. INTRODUCTION

Control Performance Assessment (CPA) in the industry is a crucial challenge involving the evaluation of the effectiveness of control systems. The need to measure and enhance control performance in industrial settings has been a persistent issue due to its direct impact on process quality and operational efficiency (Jelali, 2006).

Traditionally, CPA has been addressed through solutions that might face significant challenges. Conventional methodologies often focus on specific metrics to assess control performance but might find difficulties in dealing with the complexity and variability of modern industrial systems. In response to these challenges, data science and machine learning have emerged as promising alternatives. Their ability to process large datasets and extract complex patterns offers new insights for addressing CPA more efficiently and effectively (Grelewicz et al., 2023a,b).

For example, the goal of classification tasks within the supervised learning context is to minimize misclassifications. Nevertheless, this implies that a training dataset, with several instances, has been labeled. This is not always the case, given that the activity of labeling sometimes could be time consuming. A different approach is to use unsupervised learning techniques, for anomalies or outliers detection and verify their capability to identify discriminate between classes. This could be specially interesting when looking for faulty events.

In this context, we propose an unsupervised learning approach to address the CPA problem. We present an application of this approach on a dataset, describing 30 control performance indicators (features) in a PID control loop, for identifying bad tuning setting. The remainder of this paper is as follows: in Section 2, a brief theoretical background is presented; in Section 3, methods, tools, and experimental framework for this paper are described; in Section 4, results are discussed; and in Section 5, some conclusions and future work are given.

2. THEORETICAL BACKGROUND

Next, theoretical background on CPA and machine learning are discussed.

2.1 Control performance assessment

Control Performance Assessment is a crucial aspect in evaluating the effectiveness of control systems, ensuring their optimal functioning within industrial processes (Jelali, 2012; Domański et al., 2020). It involves the quantitative analysis of how well a control system meets the desired performance criteria. Mathematically, CPA is often

⋆ This study was financed in part by the *Conselho Nacional de Desenvolvimento Científico e Tecnológico* (CNPq) via the projects PQ-2/310195/2022-5 & Universal/408164/2021-2.

expressed through performance indices and metrics, such as the Integral of Time-weighted Absolute Error (ITAE), Integral of Squared Error (ISE), or other relevant measures such as the Harris index or minimum variance. These metrics provide a quantitative basis for assessing the system's ability to maintain stability, reduce oscillations, and respond accurately to setpoint changes or disturbances. Several works propose improvements or variations of such indexes (Horch and Isaksson, 1999; Khosroshahi et al., 2022). In any case, the formalism of CPA aims to capture the dynamic behavior of the controlled system, offering insights into its overall efficiency and robustness.

2.2 Machine learning, unsupervised learning, and binary classification

Machine learning refers to computer algorithms that enhance their capabilities autonomously through exposure to data. This learning process can be supervised, unsupervised, or via reinforcement (Mitchell et al., 1997). In unsupervised learning, instances ($\boldsymbol{I}$) with features ($\boldsymbol{X}$) are used to train a learner without relying on explicit target information ($\boldsymbol{T}$). The main goal is to discover patterns, structures, or relationships within the data that can provide insights or useful representations for further analysis or decision-making without the need for predefined labels or target predictions.

In binary classification, the task is to predict whether incoming data belongs to class 0 (C_0) or class 1 (C_1). For anomaly detection, these classes signify the presence or absence of an anomaly. Class 0 denotes a situation without anomalies, while class 1 indicates an anomalous situation. The learner undergoes a training phase using a dataset to adjust its parameters $\boldsymbol{\beta}$ through optimization, utilizing an evaluation criteria or cost function. That is, given a loss function statement $J(\boldsymbol{\beta})$ to minimize misclassifications for a given hypothesis $h_\beta(\boldsymbol{x})$ (learner structure) based on the tuneable $\boldsymbol{\beta}$ parameters for a data vector $\boldsymbol{x}$.

Several unsupervised machine learning structures have been proposed over the years. To name few basic approaches: K-means (MacQueen et al., 1967), Density-Based Spatial Clustering of Applications with Noise (DBSCAN): (Ester et al., 1996), Principal Component Analysis (PCA) (F.R.S., 1901), Autoencoders (Rumelhart et al., 1986), Apriori Algorithm (Agrawal et al., 1993), t-Distributed Stochastic Neighbor Embedding (t-SNE) (van der Maaten and Hinton, 2008), local outlier factor (LOF) (Breunig et al., 2000), Isolation Forest (Liu et al., 2008). Each one of them provides a particular structure that must be trained to adjust its tuneable parameters $\boldsymbol{\beta}$. Several optimization tools and techniques are used for such a training stage; among them, multi-objective optimization has brought interesting results. Some of them are explained below.

2.3 Related Works and Contextual Framework

On the one hand, it is not new to deal with the CPA problem from the machine learning perspective. For example, in Grelewicz et al. (2023a,b) a novel machine learning derived CPA classification system is proposed and implemented for a wide class of PID-based control industrial loops of reduced order. In Marques et al. (2024), a multi-objective optimization training approach was proposed for the same database. Nevertheless, no record of an unsupervised learning approach has been reported.

3. EXPERIMENTAL FRAMEWORK, TOOLS AND METHODS

The research question posed in this paper is *how effective could be unsupervised learning approaches for anomaly/outlier detection to clearly identify bad tuning settings in PI control loops?* For such a purpose, we will compare the performance of K-Means, isolation forest, and Local Outlier Factor (LOF) to identify bad tuning settings. Such tools and the dataset are explained below.

3.1 Data set description

The data set proposed by Grelewicz et al. (2023a) will be used. It consists of a data set of 30 control performance indicators computed from the rejection response when a disturbance step change is applied in a PID control loop. A large and representative set of different second-order plus deadtime processes was defined to build the database. For each process, a PID tuning was derived via an optimization problem, and several controllers were proposed in the proximities of them via random perturbation of the PID parameters. The labeling process was performed in the following way: if the disturbance rejection response was similar to the reference one, the instance is labeled as *OK*; otherwise, as *NOK*. This similarity was calculated in terms of gain margin, phase margin, and normalized error. The training dataset consists of 60,000 instances, while the test database has 10,000 instances, both of them with equally balanced classes and normalized values.

3.2 K-Means Clustering

K-Means (MacQueen et al., 1967) is a popular unsupervised machine learning technique used for partitioning a dataset into K distinct, non-overlapping clusters. The primary objective of K-Means is to minimize the within-cluster variance, defined as the sum of the squared Euclidean distances between each data point and the centroid of its assigned cluster.

Given a set of instances $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$ where each $\mathbf{x}_i$ is a d-dimensional vector, K-Means clustering aims to partition the data into K clusters $\{C_1, C_2, \ldots, C_K\}$. The algorithm proceeds iteratively through the following steps:

(1) **Initialization**: Randomly select K initial centroids $\{\mu_1, \mu_2, \ldots, \mu_K\}$.
(2) **Assignment Step**: Assign each instance $\mathbf{x}_i$ to the nearest centroid based on the Euclidean distance: $C_j = \{\mathbf{x}_i : \|\mathbf{x}_i - \mu_j\|^2 \leq \|\mathbf{x}_i - \mu_k\|^2, \forall k \in \{1, 2, \ldots, K\}\}$
(3) **Update Step**: Recalculate the centroids as the mean of all data points assigned to each cluster:
$$\mu_j = \frac{1}{|C_j|} \sum_{\mathbf{x}_i \in C_j} \mathbf{x}_i$$
(4) **Convergence Check**: Repeat the assignment and update steps until the centroids no longer change significantly or a predetermined number of iterations is reached.

The final objective function that K-Means minimizes is:

$$J = \sum_{j=1}^{K} \sum_{\mathbf{x}_i \in C_j} \|\mathbf{x}_i - \mu_j\|^2$$

where J is the sum of squared distances between instances and their corresponding centroids.

3.3 Isolation Forest

Isolation Forest (Liu et al., 2008) is an unsupervised machine learning technique used for anomaly detection. Unlike other methods that profile normal instances, Isolation Forest explicitly isolates anomalies. It builds an ensemble of random trees, where anomalies are more likely to be isolated early due to their distinct characteristics.

Given a dataset of instances $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, Isolation Forest works through the following steps:

(1) **Random Subsampling**: Randomly select a subsample of the dataset.
(2) **Tree Construction**: For each subsample, construct an isolation tree (iTree) by recursively partitioning the data:
  - Select a random feature f from the feature space.
  - Select a random split value $\mathbf{s}$ between the minimum and maximum values of f.
  - Partition the data based on $\mathbf{s}$ and repeat until each instance is isolated or a predefined depth limit is reached.

The path length $h(\mathbf{x})$ of an instance $\mathbf{x}$ in the tree is the number of edges traversed from the root to the terminating node. The average path length $\mathbb{E}[h(\mathbf{x})]$ across a forest of such trees provides a measure of normality:

$$c(n) = 2H(n-1) - \frac{2(n-1)}{n}$$

where $H(i)$ is the harmonic number and n is the number of samples. For an instance $\mathbf{x}$, the anomaly score is given by:

$$s(\mathbf{x}, n) = 2^{-\frac{h(\mathbf{x})}{c(n)}}$$

where $h(\mathbf{x})$ is the path length of $\mathbf{x}$. Instances with shorter average path lengths are more likely to be anomalies.

3.4 Local Outlier Factor (LOF)

Local Outlier Factor (Breunig et al., 2000) of LOF, is an unsupervised machine learning algorithm used for anomaly detection. LOF identifies anomalies by comparing the local density of a data point with that of its neighbors. Points that have a significantly lower density than their neighbors are considered anomalies.

Given a set of instances $\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$, LOF works through the following steps:

(1) **Compute k-Distance and k-Distance Neighborhood**:
  - For each data point $\mathbf{x}_i$, compute the k-distance, which is the distance to its k-th nearest neighbor.
  - Define the k-distance neighborhood $N_k(\mathbf{x}_i)$ as the set of points whose distance to $\mathbf{x}_i$ is less than or equal to the k-distance.
(2) **Reachability Distance**:
  - For each point $\mathbf{x}_i$ and a point $\mathbf{x}_j$ in its k-distance neighborhood, compute the reachability distance:
    $\text{reach-dist}_k(\mathbf{x}_i, \mathbf{x}_j) = \max\{\text{k-distance}(\mathbf{x}_j), \|\mathbf{x}_i - \mathbf{x}_j\|\}$
(3) **Local Reachability Density (LRD)**:
  - Compute the local reachability density of $\mathbf{x}_i$ as the inverse of the average reachability distance based on the k-distance neighborhood:

$$\text{LRD}_k(\mathbf{x}_i) = \left(\frac{\sum_{\mathbf{x}_j \in N_k(\mathbf{x}_i)} \text{reach-dist}_k(\mathbf{x}_i, \mathbf{x}_j)}{|N_k(\mathbf{x}_i)|} \right)^{-1}$$

(4) **Local Outlier Factor**:
  - Compute the LOF score for each point $\mathbf{x}_i$ as the average ratio of the local reachability density of $\mathbf{x}_i$ to that of its neighbors:

$$\text{LOF}_k(\mathbf{x}_i) = \frac{\sum_{\mathbf{x}_j \in N_k(\mathbf{x}_i)} \frac{\text{LRD}_k(\mathbf{x}_j)}{\text{LRD}_k(\mathbf{x}_i)}}{|N_k(\mathbf{x}_i)|}$$

A point $\mathbf{x}_i$ is considered an outlier if $\text{LOF}_k(\mathbf{x}_i)$ is significantly greater than 1.

3.5 t-Distributed Stochastic Neighbor Embedding (t-SNE)

t-Distributed Stochastic Neighbor Embedding or t-SNE (van der Maaten and Hinton, 2008) is an unsupervised machine learning technique primarily used for dimensionality reduction and visualization of high-dimensional data. t-SNE maps multi-dimensional data to two or three dimensions, making it easier to visualize complex relationships. It works through the following steps:

(1) **Compute Pairwise Similarities in High Dimensions**:
  - Compute the pairwise similarities between data points using a Gaussian distribution. For data points $\mathbf{x}_i$ and $\mathbf{x}_j$, the similarity p_{ij} is given by:

$$p_{ij} = \frac{\exp\left(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma_i^2}\right)}{\sum_{k \neq l} \exp\left(-\frac{\|\mathbf{x}_k - \mathbf{x}_l\|^2}{2\sigma_k^2}\right)}$$

    where σ_i is the variance of the Gaussian centered at $\mathbf{x}_i$.
(2) **Compute Pairwise Similarities in Low Dimensions**:
  - Map the high-dimensional data to a lower-dimensional space (usually 2D or 3D) with points $\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$.
  - Compute the pairwise similarities in the lower-dimensional space using a Student's t-distribution with one degree of freedom (which is equivalent to the Cauchy distribution). For points $\mathbf{y}_i$ and $\mathbf{y}_j$, the similarity q_{ij} is given by:

$$q_{ij} = \frac{\left(1 + \|\mathbf{y}_i - \mathbf{y}_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|\mathbf{y}_k - \mathbf{y}_l\|^2\right)^{-1}}$$

(3) **Minimize the Kullback-Leibler Divergence**:
  - Minimize the Kullback-Leibler (KL) divergence between the high-dimensional similarities p_{ij} and the low-dimensional similarities q_{ij} using gradient descent. The KL divergence is defined as:

ISSN: 2525-8311

DOI: 10.20906/CBA2024/4728

$$KL(P\|Q) = \sum_{i \neq j} p_{ij} \log \frac{p_{ij}}{q_{ij}}$$

The optimization process aims to ensure that similar data points in the high-dimensional space remain close together in the low-dimensional space, while dissimilar points are mapped far apart.

t-SNE is widely used for visualizing clusters, patterns, and relationships in high-dimensional datasets, making it a powerful tool for exploratory data analysis.

4. RESULTS AND DISCUSSION

To performed the experiment, Google Colaboratory was utilized, a cloud-based platform provided by Google that offers free access to GPUs and TPUs for running Python code in Jupyter notebooks Google (n.d.). To improve the visualization of outliers in the database and the predictions of each technique, the t-SNE technique was applied to reduce the dimensionality of the data to a two-dimensional space, using a perplexity of 30 and 300 iterations. A new dataset was created for the training phase to reduce the percentage of outliers in the database, comprising 90% of good data and 10% of outliers. The result is presented in Figure 1. For the Isolation Forest and LOF techniques, a contamination of 0.1 was considered, representing 10% of outliers in the data. To choose the number of clusters in the K-Means algorithm, the silhouette method was applied. After evaluating Figure 2, it was concluded that the most appropriate number of clusters is 3. This was determined based on the interpretation of the silhouette method, which aims to find well-defined and separated clusters, indicated by a high mean silhouette value. In Figures 3, 4 and 5, the performance from de training phase for Isolation Forest, K-Means and LOF techniques are shown, respectively. The blue points represent the normal data, while the red points represent the outliers detected by each of the techniques.

To evaluate the effectiveness of each model in detecting outliers, was used the confusion matrix, which highlights the number of true positives (TP), false positives (FP), true negatives (TN) and false negatives (FN). In Figure 9, you can see the result, where the blue circle indicates Isolation Forest, the green star represents K-Means and the red square represents Local Outlier Factor. Isolation Forest has demonstrated outstanding performance in identifying TN, indicating its effectiveness in accurately detecting normal data, as well as a low FP rate, which is beneficial for avoiding false alarms. K-Means, in turn, performed better at identifying TP, but performed worse at TN and FP, suggesting it may have difficulties correctly distinguishing between normal data and outliers despite a reduced FN rate. Local outlier factor showed the worst performance in identifying TP and FN, indicating a lower ability to effectively detect outliers and a tendency to not identify important anomalies in the data.

After training models, they are applied to test data to identify outliers and evaluate their performance on this task. The visualization of the methods can be seen in Figures 6, 7 and 8, while the effectiveness of the applied models is evaluated by the confusion matrix represented by Figure 10. In the test, the Isolation Forest technique stood out for presenting a high rate of true negatives (TN), which highlights its effectiveness in accurately identifying normal data. Furthermore, it demonstrated a relatively low false positive (FP) rate, indicating a low occurrence of normal data misclassified as outliers. On the other hand, KMeans also exhibited a high rate of true negatives, similar to Isolation Forest, indicating its ability to distinguish normal data. However, it was observed that KMeans had the lowest true positive (TP) rate among the three techniques, which suggests a lower ability to identify outliers. Despite this, KMeans demonstrated a low rate of false positives, demonstrating its accuracy in identifying normal data. LOF, on the other hand, presented the highest rate of false negatives among the techniques, indicating a lower effectiveness in correctly identifying outliers. Its true negative and false positive rates were intermediate compared to the other techniques. These observations highlight the different characteristics and performances of each technique in detecting outliers. Table 1 presents the percentages of successes and errors for each evaluation metric, revealing a remarkable consistency between the rates of true negatives and false positives between the training and testing phases, indicating a good generalization capacity of the Isolation Forest and KMeans models, while the LOF model exhibits a greater discrepancy, especially regarding true positive and false negative rates.

Table 1. Training and Testing Results of Techniques

Technique		Isolation Forest	KMeans	LOF
Training	TP	93.82%	98.41%	91.22%
	TN	44.37%	16.47%	20.97%
	FN	55.63%	83.53%	79.03%
	FP	6.18%	1.59%	8.78%
Testing	TP	93.70%	98.34%	95.24%
	TN	43.46%	15.92%	15.24%
	FN	56.54%	84.08%	84.76%
	FP	6.30%	1.66%	4.76%

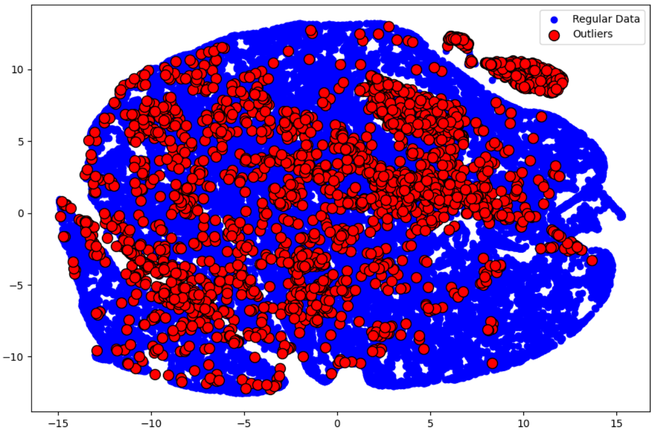

Figure 1. Composition of New Dataset.

5. CONCLUSIONS

In this article, three unsupervised machine learning techniques were presented to address the CPA classification problem, namely Isolation Forest, KMeans and LOF (Local Outlier Factor), which were evaluated for detecting anomalies in PI control loops. The results show significant

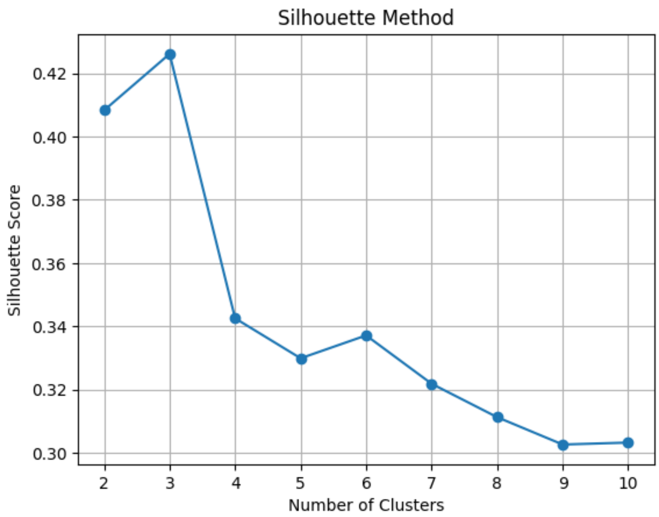

Figure 2. Choosing the Number of Clusters using the Silhouette Method.

variations in performance metrics, with each technique showing different levels of effectiveness in identifying bad tuning configurations. One question about the effectiveness of unsupervised learning approaches for identifying bad tuning setting in PI control loops was posed and answered by the experimental results, demonstrating that when using a forest isolation technique a considerable result was obtained. As expected, the results demonstrated that unsupervised learning techniques performed worse compared to multi-objective optimization training in supervised machine learning proposed in Marques et al. (2024), due to the absence of targets and the essential labeling process in supervised learning; however, exploring hybrid approaches that combine both methods may offer an ideal balance between accuracy and pattern detection capability, suggesting a promising direction for future research.

ACKNOWLEDGMENTS

We want to acknowledge the kindness and support of Patryk Grelewicz, Jacek Czeczot, et al. for allowing us to use their data set presented in Grelewicz et al. (2023a,b).

REFERENCES

Agrawal, R., Imieliński, T., and Swami, A. (1993). Mining association rules between sets of items in large databases. In *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, SIGMOD '93, 207–216. Association for Computing Machinery, New York, NY, USA. doi:10.1145/170035.170072. URL `https://doi.org/10.1145/170035.170072`.

Breunig, M.M., Kriegel, H.P., Ng, R.T., and Sander, J. (2000). Lof: identifying density-based local outliers. In *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, SIGMOD '00, 93–104. Association for Computing Machinery, New York, NY, USA. doi:10.1145/342009.335388. URL `https://doi.org/10.1145/342009.335388`.

Domański, P.D. et al. (2020). *Control Performance Assessment: Theoretical Analyses and Industrial Practice*, volume 245. Springer.

Ester, M., Kriegel, H.P., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, 226–231. AAAI Press.

F.R.S., K.P. (1901). Liii. on lines and planes of closest fit to systems of points in space. *Philosophical Magazine Series 1*, 2, 559–572. URL `https://api.semanticscholar.org/CorpusID:125037489`.

Google (n.d.). Google colaboratory. `https://colab.research.google.com/`.

Grelewicz, P., Khuat, T.T., Czeczot, J., Nowak, P., Klopot, T., and Gabrys, B. (2023a). Application of machine learning to performance assessment for a class of pid-based control systems. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Grelewicz, P., Nowak, P., Khuat, T.T., Czeczot, J., Klopot, T., and Gabrys, B. (2023b). Practical implementation of computationally-efficient machine learning-based control performance assessment system for a class of closed loop systems. *Applied Soft Computing*, 146, 110690.

Horch, A. and Isaksson, A.J. (1999). A modified index for control performance assessment. *Journal of Process Control*, 9(6), 475–483.

Jelali, M. (2006). An overview of control performance assessment technology and industrial applications. *Control engineering practice*, 14(5), 441–466.

Jelali, M. (2012). Control performance management in industrial automation: assessment, diagnosis and improvement of control loop performance.

Khosroshahi, M., Poshtan, J., and Alipouri, Y. (2022). Practical control performance assessment method using hurst exponents and crossover phenomena. *Computers & Chemical Engineering*, 161, 107774.

Liu, F.T., Ting, K.M., and Zhou, Z.H. (2008). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, 413–422. doi:10.1109/ICDM.2008.17.

MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, 281–297. Oakland, CA, USA.

Marques, T., Carrillo-Ahumada, and Jesús Reynoso-Meza, G. (2024). Multi-objective machine learning for control performance assessment in PID control loops. In *Proceedings og the IFAC PID Advances*.

Mitchell, T.M. et al. (1997). *Machine learning*. McGraw-hill New York.

Rumelhart, D.E., Hinton, G.E., and Williams, R.J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533–536.

van der Maaten, L. and Hinton, G. (2008). Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(86), 2579–2605. URL `http://jmlr.org/papers/v9/vandermaaten08a.html`.

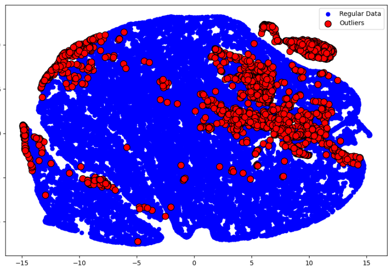

Figure 3. Visualization of Isolation Forest Performance (Train)

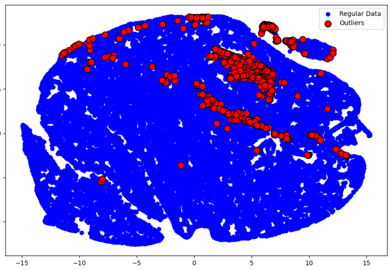

Figure 4. Visualization of K-Means Performance (Train)

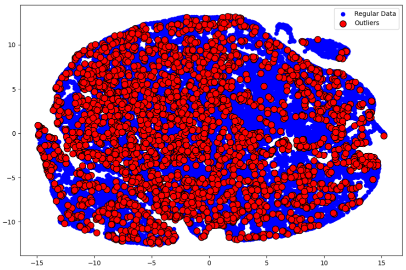

Figure 5. Visualization of LOF Performance (Train)

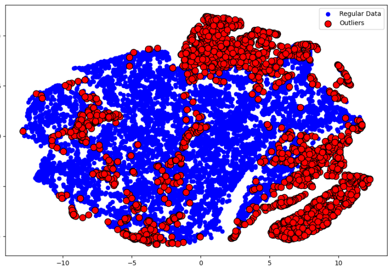

Figure 6. Visualization of Isolation Forest Performance (Test)

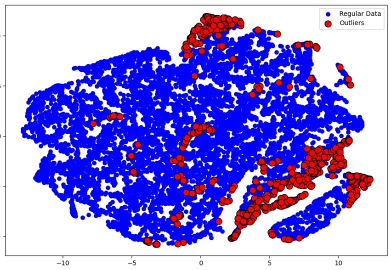

Figure 7. Visualization of K-Means Performance (Test).

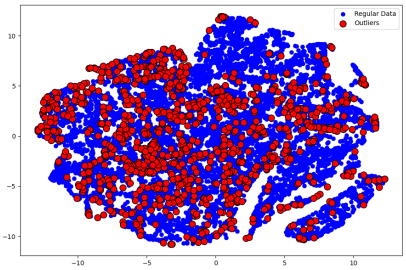

Figure 8. Visualization of LOF Performance (Test)

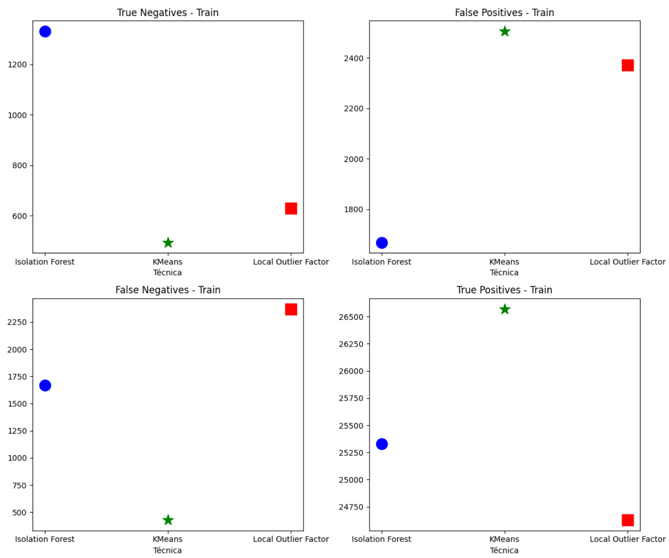

Figure 9. Confusion Matrix - Train

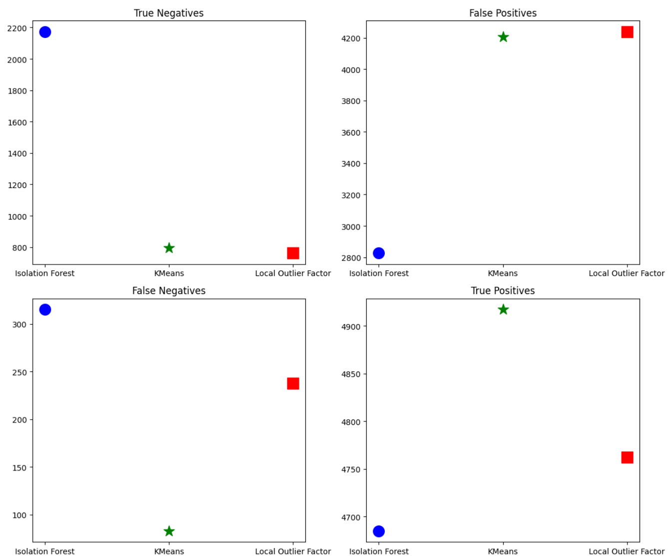

Figure 10. Confusion Matrix - Test