Semi-Supervised Learning Algorithm for Running-in Analysis on Compressors

Jonatan Leandro MACHADO[1], Gabriel THALER [2], Rodolfo César Costa FLESCH[1*],
João Paulo Zomer MACHADO[2]

[1] Universidade Federal de Santa Catarina, Departamento de Automação e Sistemas,
Florianópolis, Santa Catarina, Brazil
jonatan.machado@labmetro.ufsc.br, rodolfo.flesch@ufsc.br

[2] Universidade Federal de Santa Catarina, Programa de Pós-Graduação em Engenharia de Automação e Sistemas,
Florianópolis, Santa Catarina, Brazil
gabriel.thaler@labmetro.ufsc.br, joao.zomer@labmetro.ufsc.br

* Corresponding Author

ABSTRACT

This work presents a semi-supervised machine learning algorithm for identifying the end of the running-in process in reciprocating hermetic compressors. This identification is important for guaranteeing that the compressor reaches its tribological steady state before it is tested. Testing a not run-in compressor can result in erroneous measurements, particularly in energy performance tests. The most widely used procedure to avoid such problems is to operate the compressor under specific conditions for an empirically determined number of hours, but this method does not guarantee the end of the running-in process. The objective of this work is to present a fully automatic method for classification of the compressor state into running-in or tribological steady state. A test rig was built for imposing particular operating conditions for the compressor under test and for acquiring experimental data, such as pressures and temperatures at specific points of the refrigeration circuit, as well as the electric current. Four compressors which have never been turned on before were tested using the proposed rig to acquire data of the running-in phenomenon. The same compressors were tested twice more after operating for several hours to acquire data of compressors which are known to be in a tribological steady state. Processing based on a delayed space sliding window was used in the test time series to subsequence the root mean squared values of the electric current dataset. In addition, a semi-supervised machine learning method named self-training was used, considering k-nearest neighbors (KNN), random forest, and support vector machine methods as classifiers. The only two pieces of information assumed for the proposed method are that at the beginning of the first test of each compressor unit it was not run in yet, and that every other test the compressor was in its tribological steady state. The best classifier for identifying the end of the running-in process was obtained using the KNN method with less than 60 neighbors and a small number of features (4 or less). The results are consistent with comparative analyses and the running-in literature.

1. INTRODUCTION

The running-in process is associated with the onset of sliding contact between two surfaces of a mechanism. It is characterized by transient changes in wear rate, temperature, coefficient of friction, surface roughness, and other parameters that define the tribological state of the material (Blau, 2009).

In reciprocating hermetic compressors, it is assumed that running-in occupies only a small portion of the device lifespan, being a transitional period followed by the tribological equilibrium of the system, known as a steady state, where the device will have its performance values constant for most of its lifespan. To assess these parameters, it is necessary, by international standards such as ANSI/ASHRAE 23 ASHRAE (2005) and ISO 917 ISO (1989), for the device to have completed its running-in process and be in a steady state. However, due to the hermetic characteristic of the tested compressors, direct detection of the running-in period in these compressors is unfeasible and costly. The upper and lower shell are welded, which would result in the destruction of the part and a risk of lubricating oil contamination. Even in cases where this does not occur, it would still require disassembly and reassembly of the device, thus generating a new running-in process.

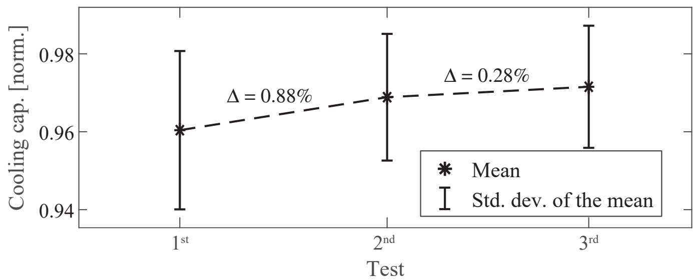

Figure 1: Mean and standard deviation of the normalized cooling capacity in consecutive performance tests

Currently, methods in the field of reciprocating hermetic compressors for the analysis and detection of running-in are not effective, with the most common method being the operation of the device for a empirically determined number of hours (Martins et al., 2011). Given the lack of objective indicators for the end of the running-in process in compressor units, the preparation period may be either excessively long or insufficient, impacting performance test results.

Figure 1 represents the results of 3 consecutive tests of 33 compressors of the same model, after being subjected to the empirical running-in method. It evaluates the mean value and standard deviation of the refrigeration capacity of these compressors, normalized to a range [0;1]. As can be observed, there is a greater variation in the measurements equivalent to the first test, an effect that may indicate the effects of running-in process.

This analysis revealed that the mean cooling capacity in the first test cycle was lower than in subsequent cycles, indicating potential inadequacy in the running-in preparation period for some compressor units. This reinforces the need for an automated method to detect the end of the running-in period, which could optimize preparation time and ensure accurate measurements. Nevertheless, analyzing sliding surfaces directly, as commonly done to assess the running-in condition in simpler systems, proves unfeasible in compressors owing to their intricate nature.

To address these issues, indirect analysis methods have been developed in the literature for simple mechanisms. The work by Argatov & Jin (2023) developed a time-delay neural network model for the analysis of the variation of the wear rate during the running-in wear process under constant relative sliding velocity. The work by Martin (1992) employs the kurtosis of the system vibration to analyze friction, demonstrating that the distribution of the vibration signal approaches normal distribution as the running-in period approaches its end. Particularly for compressors, the only work found in the literature was the one by Thaler et al. (2022), which compared the use of different performance metrics when training random forest (RF) models to detect the running-in. Upon application to experimental data, the results from these models indicated that the Matthews correlation coefficient and the F-score were the most suitable metrics for the specific case study. This approach, however, does not guarantee the reliability of the results. Another aspect regarding the use of unsupervised machine learning techniques in this case is the impossibility of applying such techniques for an online detection of the running-in phenomenon, as they rely on data from the compressor unit after the running-in procedure. However, direct analysis of sliding surfaces, typically used to estimate the running-in state in simpler mechanisms, is impractical in compressors, as shown in the work of Khonsari et al. (2020). Most experimental running-in analyses investigate friction, lubrication, topography, and wear behaviors, generally directed toward tribological pairs, so this type of study rarely translate into analyses of complex machinery (Blau, 1991).

The objective of this work is to propose a method to identify the end of the running-in state in hermetic reciprocating compressors for refrigeration. The proposed method is based on a semi-supervised machine learning method known as self-training, along with its various classifiers and parameters. In addition, empirical knowledge of the running-in process is used to pre-categorize moments within the process.

The remainder of the work is organized as follows. Section 2 describes the experimental development of the work, including the description of the test rig used and the data obtained from it, as well as explaining the proposed machine learning method. Section 3 presents the results generated by the proposed method used. Finally, Section 4 provides the conclusions of this work.

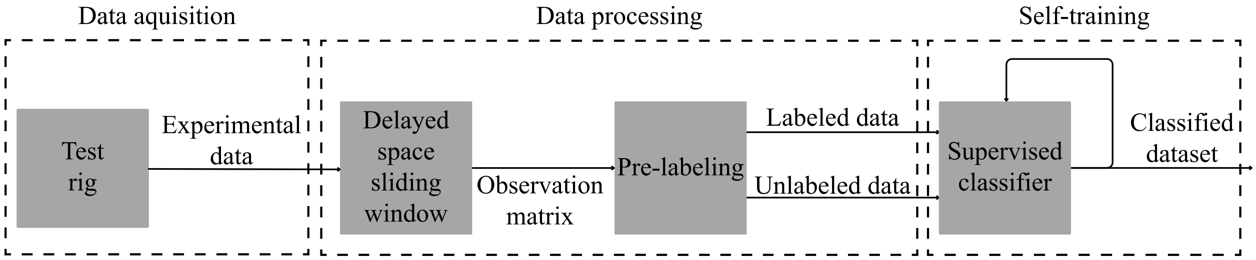

Figure 2: Overview of the processing pipeline of the proposed method

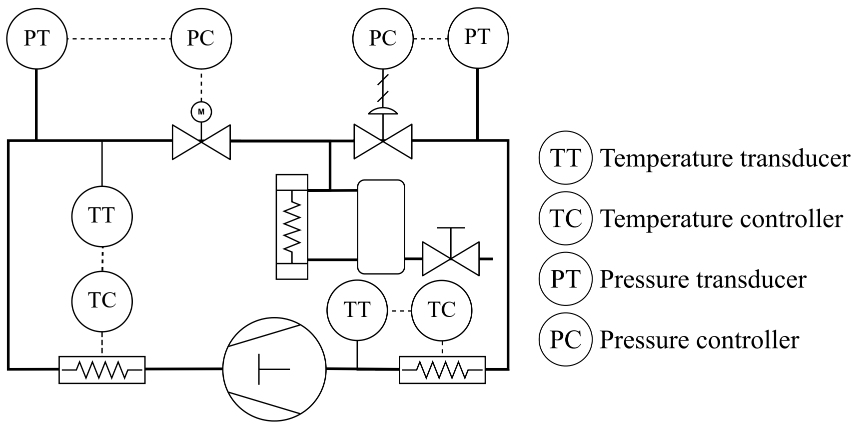

Figure 3: Diagram of the test rig

2. PROPOSED METHOD

This section presents the procedures used to obtain the relevant data for analysis of the running-in phenomenon, as well as those used for subsequent data processing. In addition, the proposed method based on a semi-supervised machine learning model is described. As can be seen in Figure 2, this section is divided into three parts, Section 2.1 addresses data acquisition in the test rig used, while Section 2.2 discusses how the data were processed for the model using a delayed space sliding window and how they were pre-labeled based on empirical notions of the running-in process. Finally, Section 2.3 discusses the machine learning model used, how its classifiers operate, and how it uses both labeled and unlabeled data.

2.1 Data Acquisition

This work makes use of an experimental test rig for compressor testing. The simplified piping and instrumentation diagram of this test rig is presented in Figure 3.

The test rig was designed to operate in a hot cycle, without phase change of the refrigerant fluid used. The suction pressure of the compressor is controlled using a proportional-integral controller, which acts on the opening of the valve regulated by an electropneumatic converter, and the discharge pressure is controlled using a cascade controller, with a fast inner loop to control the position of the valve and an outer loop for pressure control. In this control system, the inner loop is controlled using a proportional controller, and the outer loop using a proportional-integral controller. Temperature control is manually performed using resistors under pulse width modulation, and current measurement is carried out using a Hall effect transducer.

Running-in tests lasting 20 hours were conducted with data being measured every minute, a duration established based on experimental data as sufficient to ensure the running-in of the model. Sensors were implemented in this test rig to acquire data on the motor electric current, potentially associated with the running-in process, as well as to monitor and control the test rig. From these sensors, statistical parameters were extracted. This work focuses on the results of extracting information from the electric current of the compressor. After the running-in tests, two tests were conducted on the same units already run-in, following the same procedure, with the aim of acquiring data from the tribological steady state.

2.2 Data Processing

The metric used for the analysis in this study was the root mean square (RMS) value of the electric motor inside the compressor. For the construction of observation vectors, a sliding window of N moments within a delay space of D samples was used. The vector os samples used as input to the proposed method is given by:

$$X(k) = \left[x(k), \quad x(k-D), \quad x(k-2D), \quad \cdots, \quad x(k-(N-1)D)\right], \tag{1}$$

where $x(k)$ is the RMS current at the instant of time k, and $X(k)$ is the processed observation vector. Due to the inrush current spike and the natural transient period in compressor tests (Coral et al., 2019), only data acquired after the first hour of operation are considered for analysis, and $k = 1$ is defined as the 1 h threshold. Lastly, the early moments of the first conducted test were categorized as "running-in", while the remainder of the first test, labeled as "uncertain", will serve as the domain for the self-training model described in Subsection 3.3. The results from the subsequent tests with the same compressor were classified as "tribological steady state".

2.3 Self-training

Traditional supervised machine learning relies on using labeled examples to train a model that can then classify new examples. On the other hand, unsupervised learning focuses on discovering patterns or underlying structures in unlabeled datasets (Hastie et al., 2009). Semi-supervised machine learning is an area that bridges supervised and unsupervised learning. In this context, part of the dataset is labeled, while another part remains unlabeled (van Engelen & Hoos, 2020). This approach becomes useful in real-life scenarios since the availability of unlabeled datasets is typically abundant, given that the cost of labeling large volumes of data can be significant (Zhou et al., 2007).

The self-training method is a semi-supervised machine learning method based on a teacher-student framework. In this framework, initially, the teacher generates pseudo-labels for an unlabeled dataset X_{ω}. These pseudo-labels are then used to train a second model, known as the student, which attempts to predict the labels of these unlabeled data based on the pseudo-labels generated by the teacher model (Zuo et al., 2022).

This algorithm is iterative, with each iteration selecting a portion of the labeled set S and training a supervised classifier on that subset. Then, this classifier is used to generate pseudo-labels on a portion of the unlabeled data X_{ϖ}, removing these data from the unlabeled set X_{ω} and adding them to the labeled set S. In a new iteration, a new supervised classifier is trained using the data from $S \cup X_{\varpi}$, treating the pseudo-labels as additional labeled examples. To do this, the classifier h minimizes an empirical regularized loss L over S and X_{ϖ}, expressed in Equation 2. The iterations end when the unlabeled set X_{ω} becomes empty (Amini et al., 2023).

$$L = \frac{1}{m} \sum_{((x,y)\in S)} \ell(h(x), y) + \frac{\gamma}{|X_{\varpi}|} \sum_{(x,\tilde{y})\in X_{\varpi}} \ell(h(x), \tilde{y}) + \lambda \left\|h\right\|^2, \tag{2}$$

where $\ell : Y \times Y \rightarrow [0, 1]$ represents an instantaneous loss, γ is a hyperparameter controlling the impact of pseudo-labeled data on the learning process, and λ is the regularization hyperparameter.

The self-training method can be seen as a gradual expansion process of a labeled dataset by adding unlabeled data. As more data are added, the model accuracy is expected to improve. However, it is important to highlight that the quality of pseudo-labels generated by the teacher model is crucial for the success of the method. If these pseudo-labels are of low quality, they can impair the model accuracy and result in poor performance (Li et al., 2023).

Although it is not possible to use comparative performance metrics on unlabeled data, they can be applied in the supervised portion of the data. Since the dataset is unbalanced, with a large number of "steady state" data over "running-in" data, the Matthews correlation coefficient (MCC) was used as performance index. The MCC is a contingency matrix method for calculating the Pearson product-moment correlation coefficient between actual and predicted values Powers (2020). The MCC is calculated as follows:

$$\mathrm{MCC} = \frac{T_{\mathrm{P}} \times T_{\mathrm{N}} - F_{\mathrm{P}} \times F_{\mathrm{N}}}{\sqrt{(T_{\mathrm{P}} + F_{\mathrm{P}}) \times (T_{\mathrm{P}} + F_{\mathrm{N}}) \times (T_{\mathrm{N}} + F_{\mathrm{P}}) \times (T_{\mathrm{N}} + F_{\mathrm{N}})}},$$

where T_{P} stands for true positives, T_{N} for true negatives, F_{P} for false positives, and F_{N} for false negatives.

The MCC is a binary classification metric that assigns a high score only when the binary predictor can correctly predict the majority of positive and negative data elements. It scales from -1 to $+1$, with extreme values of -1 representing cases of perfect misclassification and $+1$ representing perfect classification (Chicco, 2017).

The MCC, besides being widely used and recognized as a robust metric for evaluating binary classification, as demonstrated in studies like Chicco et al. (2021), works such as Thaler et al. (2022) demonstrate the advantages of using MCC over other binary classification metrics for classifying the running-in process.

With the aim of evaluating the chosen models, an acceptance rate was created. It was applied to all models with their variations of N and D, where each variation that met the created criteria would receive a value of 1, and those that did not would receive 0. The final acceptance rate of each model was calculated by dividing the accepted models by the total number of models. The criteria used were as follows: firstly, a MCC value above 0.7, and secondly, the presence of both "running-in" and "steady state" classes in the classification of unsupervised data, since by design the running-in test should contain both tribological states.

3. RESULTS

In this work, four units of compressors from a household model using R134a as refrigerant fluid with a cooling capacity of 56 W in ASHRAE low back pressure condition were used. Using the ASHRAE LBP conditions (-23.3 °C for evaporating temperature and 54.4 °C for condensing temperature) for testing, their data were processed using the method described in Section 2.2 and varying the parameters N and D for the chosen classifiers. The following values were used for $N: 1, 2, 4, 8, 16$, and 32, and for $D: 1, 5, 10, 15, 20, 25$, and 30, but limiting the total window, given by $(N-1)D$, to 180 min to avoid missing too much of the running-in process.

The supervised machine learning models used for the self-training classifier were as follows: K-nearest neighbors (KNN) with K neighbors equal to 3, 6, 30, 60, and 300, support vector machine (SVM) with four different kernels: linear kernel, quadratic kernel, cubic kernel, and radial basis function (RBF) kernel, and RF with the number of trees equal to 10, 100, and 1000. For evaluating the performance of the supervised classifier models in the self-training model, the acceptance rate was used. Figure 4 shows the acceptance rate of the models. In general, it is possible to identify that the increase in the number of feature N has a negative effect on the model acceptance, with the value of $N = 1$ being the one that had its MCC closest to +1 for the classifier. The values of KNN with the number of neighbors equal to 6 were omitted due to their similarity to the results with the number of neighbors equal to 3, and the same was done with KNN with 60 neighbors, given its similarity to 30 neighbors. The change in the value of D did not appear to have any direct effect on the acceptance.

The supervised SVM classifier yielded the most unsatisfactory results, failing to generate a classification model that suited the system. It exhibited null results in terms of acceptance ratio, regardless of the values of N and D used. An example of this inadequate classification can be seen in Figure 5, which illustrates the SVM classification of running-in tests, for $N = 4$ and $D = 10$, with a moving average filter of 10 samples over the classification results, such that the values 0 and 1 on the Y-axis represent "running-in" and "steady state", respectively. This behavior indicates that most SVM models failed to produce reasonable results, except for the RBF kernel, which managed to classify units 1 and 2 of the compressors but lacked the ability to classify the others. Is also noticeable that all the lines, except for the RBF Kernel are overlapping in the value 1.0 for the duration of the test.

The RF model showed favorable results, with its classification being consistent regardless of the chosen parameters, particularly for the number of trees equal to $n_T = 100$ and 1000. An example of classification is illustrated in Figure 6, using the same parameters and filter as the models in Figure 5. However, when the number of trees was set to $n_T = 10$, some discrepancies were observed, with more noise at the beginning of the process. It is worth noting that this model was unable to correctly classify compressor Unit 4, showing significant variations in results throughout the experiment.

Finally, the KNN model also produced satisfactory results, with higher acceptance ratio values achieved with lower neighbor values, such as $K = 3$. However, as in the example shown in Figure 7, produced with the same filter and processing parameters as Figures 5 and 6, the models had similar classifications for units 1 and 3, but discrepancies were observed for units 2 and 4. Additionally, it exhibited a similar issue to the RF classification model, with inconsistent responses during the analysis, albeit better than the RF model. It is also possible to identify that the results for $K = 300$ were not as good as the other ones, since it almost immediately classified all tests as in the steady-state, and that

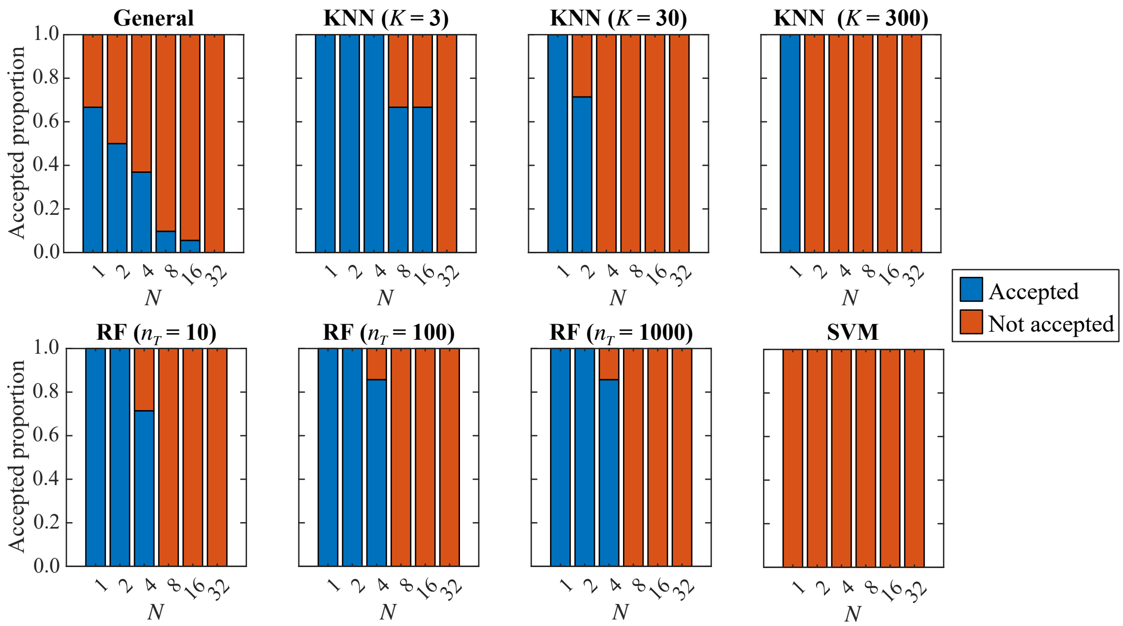

Figure 4: Acceptance rate values as a function of the value of N

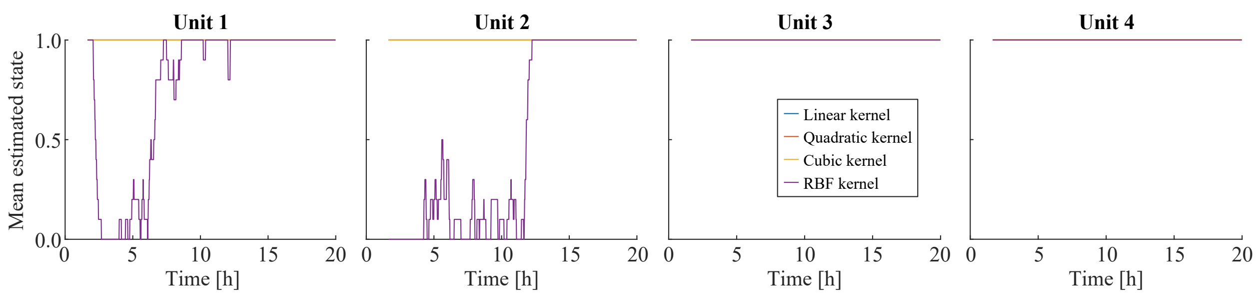

Figure 5: Results of the running-in state probability using the SVM model, using D = 10 and N = 4

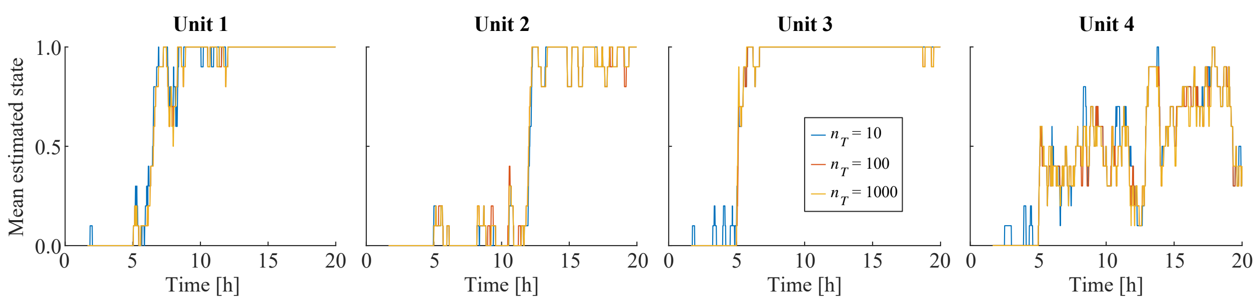

Figure 6: Results of the running-in state probability using the RF model, using D = 10 and N = 4

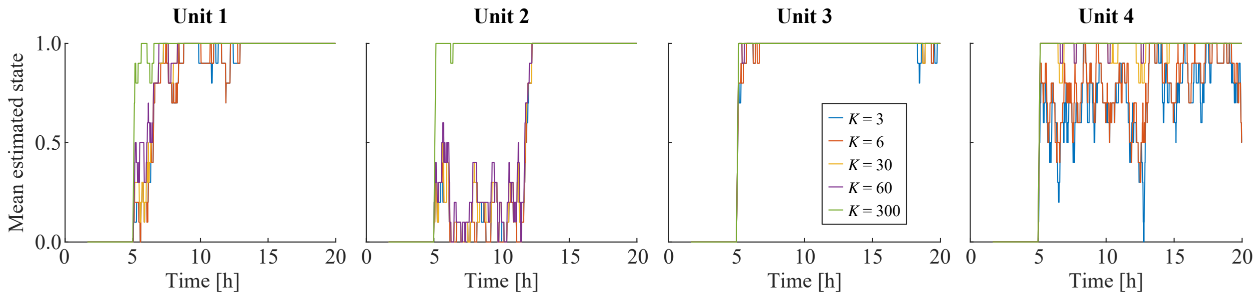

Figure 7: Results of the running-in state probability using the KNN model, using D = 10 and N = 4

models with lower K values (3 and 6) were noisier than those with intermediate values (30 and 60), varying more between classes for the running-in test of Unit 4.

Overall, evaluation of the predicted classes of all the accepted classifiers in running-in tests showed that, while KNN and RF models with low N values passed the MCC score criteria, they also led to early detections of tribological steady-state. These results are most likely a bias due to the natural transient period of compressors, since performance test data suggest that the running-in should take longer than suggested by these models. As such, the best results considered were those with N values between 4 and 16, which at least for some units detected a somewhat longer running-in period (at least 6 hours).

4. CONCLUSIONS

This research introduces a semi-supervised machine learning algorithm designed to identify the end of the running-in process in reciprocating hermetic compressors. Traditionally, determining the end of the running-in period relies on empirical methods, which are neither reliable nor efficient since each compressor unit can have its own running-in dynamics. This study then aims to automate this process of classification by using the electric current value to classify compressor operation into “running-in” or “tribological steady state”.

Four units of compressors underwent testing, with data collected throughout the running-in period and subsequent operation. The data were processed to obtain the RMS value and a sliding window N with a space delay of D samples was used to characterize the dynamic behavior. Processed data were used as input for three semi-supervised machine learning approaches, namely SVM, KNN, and RF, to distinguish between the two tribological states of the compressor. KNN models with N values between 4 and 16 and a medium number of neighbors K (30 or 60) that passed the performance criteria of an MCC score greater than or equal to 0.7 provided the best results, fitting the expected behavior for running-in. Other notable results are that no SVM models achieved satisfactory results in the classification task, and the same happened for models with a higher number of features (N = 32). The value of D does not directly influence the classification results, but can be tuned in order to achieve acceptable scores.

These findings align with existing literature on running-in phenomena, offering a semi-supervised solution for compressor testing, contributing to advancing compressor evaluation protocols. In addition, it is a novel application for semi-supervised classification methods. Future work in the field may involve switching the supervised classifiers of the self-training model, as well as analyzing different semi-supervised methods in the context of hermetic reciprocating compressors.

NOMENCLATURE

D	delay between samples in sliding window
F_{P}	false positive
F_{N}	false negative
h	classifier function of the self-training model
K	number of neighbors in the KNN algorithm
L	loss of the self-training model
m	number of observations in the labeled dataset
N	number of samples in sliding window
n_T	number of trees in the RF algorithm
S	labeled dataset
T_{P}	true positive
T_{N}	true negative
X_{ω}	unlabeled dataset
X_{ϖ}	portion of the unlabeled data
$X(k)$	sliding window vector of RMS current measurements in delayed space
$x(k)$	RMS current at instant k
y	label data
$\tilde{y}$	pseudo-label data of the self-training model
ℓ	instantaneous loss of the self-training function
γ	hyperparameter of the self-training function
λ	regularization hyperparameter of the self-training function

REFERENCES

Amini, M.-R., Feofanov, V., Pauletto, L., Hadjadj, L., Devijver, E., & Maximov, Y. (2023). *Self-training: A survey.* arXiv. doi: 10.48550/arXiv.2202.12040

Argatov, I., & Jin, X. (2023). Time-delay neural network modeling of the running-in wear process. *Tribology International*, *178*. doi: 10.1016/j.triboint.2022.108021

ASHRAE. (2005). *ANSI/ASHRAE 23: Methods of testing for rating positive displacement refrigerant compressors and condensing units* (Standard). Atlanta: ASHRAE.

Blau, P. J. (1991). Running-in: Art or engineering? *Journal of Materials Engineering*, *13*(1), 47–53. doi: 10.1007/BF02834123

Blau, P. J. (2009). *Friction science and technology* (2nd ed.). Boca Raton, USA: CRC Press.

Chicco, D. (2017). Ten quick tips for machine learning in computational biology. *BioData Mining*, *10*, 1-17. doi: 10.1186/S13040-017-0155-3

Chicco, D., Starovoitov, V., & Jurman, G. (2021). The benefits of the matthews correlation coefficient (MCC) over the diagnostic odds ratio (DOR) in binary classification assessment. *IEEE Access*, *9*, 47112-47124. doi: 10.1109/ACCESS.2021.3068614

Coral, R., Flesch, C. A., & Flesch, R. C. C. (2019). Measurement of refrigeration capacity of compressors with metrological reliability using artificial neural networks. *IEEE Transactions on Industrial Electronics*, *66*(12), 9928–9936. doi: 10.1109/TIE.2019.2898613

Hastie, T., Tibshirani, R., Friedman, J., Hastie, T., Tibshirani, R., & Friedman, J. (2009). *The elements of statistical learning: Data mining, inference, and prediction* (2nd ed.). New York, NY: Springer.

ISO. (1989). *ISO 917:1989 – Testing of refrigerant compressors* (Standard). Geneva: International Organization for Standardization.

Khonsari, M., Ghatrehsamani, S., & Akbarzadeh, S. (2020). On the running-in nature of metallic tribo-components: A review. *Wear*, *474–475*, 203871. doi: 10.1016/j.wear.2021.203871

Li, S., Liu, J., Shen, W., Sun, J., & Tan, C. (2023). Robust teacher: Self-correcting pseudo-label-guided semi-supervised learning for object detection. *Computer Vision and Image Understanding*, *235*, 103788. doi: 10.1016/j.cviu.2023.103788

Martin, H. (1992). Vibration analysis. In *ASM Handbook, volume 18: Friction, lubrication, and wear technology.* Materials Park, EUA: ASM International.

Martins, E. F., Flesch, C. A., Flesch, R. C., & Borges, M. R. (2011). Conception of a test bench to generate known and controlled conditions of refrigerant mass flow. *ISA Transactions*, *50*(3), 513–520. doi: 10.1016/j.isatra.2011.01.013

Powers, D. M. W. (2020). *Evaluation: from precision, recall and f-measure to ROC, informedness, markedness and correlation.* arXiv. doi: 10.48550/arXiv.2010.16061

Thaler, G., Menegali, J. V. B., de S. Nascimento, A. S. B., Pacheco, A. L. S., Flesch, R. C. C., & Machado, J. P. Z. (2022). Efeito da seleção de métricas de desempenho em modelos de aprendizado supervisionado para detecção de amaciamento em compressores [Effect of performance metrics selection in supervised learning models for detecting compressor running-in]. In *Procedings of the XXIV congresso brasileiro de automática* (pp. 3635–3642). Fortaleza, Brazil. doi: 10.20906/CBA2022/3670

van Engelen, J. E., & Hoos, H. H. (2020). A survey on semi-supervised learning. *Machine Learning*, *109*, 373–440. doi: 10.1007/S10994-019-05855-6

Zhou, Z.-H., Zhan, D.-C., & Yang, Q. (2007). Semi-supervised learning with very few labeled training examples. In *Proceedings of the twenty-second AAAI Conference on Artificial Intelligence* (pp. 675–680). Vancouver, Canada: AAAI.

Zuo, S., Yu, Y., Liang, C., Jiang, H., Er, S., Zhang, C., … Zha, H. (2022). Self-training with differentiable teacher. In *Findings of the association for computational linguistics: NAACL 2022* (p. 933-949). Seattle, USA: Association for Computational Linguistics (ACL). doi: 10.18653/v1/2022.findings-naacl.70

ACKNOWLEDGMENT

This work was supported in part by the Brazilian National Agency of Petroleum, Natural Gas and Biofuels (ANP) under the Human Resource Training Program (PRH), in part by the Brazilian National Council for Scientific and Technological Development (CNPq) under Grant 315546/2021-2, in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) – Finance Code 001, and in part by Nidec Global Appliance.