COBEM 2023
27TH INTERNATIONAL CONGRESS OF MECHANICAL ENGENEERING

ABCM
Brazilian Society of Mechanical Sciences and Engineering
27th ABCM International Congress of Mechanical Engineering
December 04-08, 2023, Florianópolis-SC, Brazil

COB-2023-1929
ONE-CLASS SUPPORT VECTOR MACHINES FOR REAL-TIME AND UNSUPERVISED CONDITION MONITORING OF ROTARY MACHINES

Alexandre Henrique Pereira Tavares
Aldemir Ap Cavalini Jr
Universidade Federal de Uberlândia, Uberlândia, MG, Brazil
alexandrehptavares@gmail.com
aacjunior@ufu.br

Antonio C. Veiga Paschoarelli
Universidade Federal de Uberlândia, Uberlândia, MG, Brazil
acpveiga@ufu.br

Amirhassan Abbasi
Villanova University, Villanova, PA, USA
C. Nataraj
Villanova University, Villanova, PA, USA
c.nataraj@villanova.edu
aabbasi@villanova.edu

Abstract. *Constant monitoring of industrial rotating machines is of great importance to ensure their health, reliability, and safety. In addition, costs with unplanned downtime due to preventive maintenance are reduced, as maintenance will only be performed if necessary. Numerous systems perform constant, real-time health monitoring of rotating machines. However, these systems require domain experts to define their parameters based on the type of rotating machine being monitored. Alternatively, some systems use machine learning algorithms with supervised or unsupervised learning techniques, to allow the algorithm to learn by itself the behavior of the rotating machine, reducing the need for an expert to set the system parameters to work ideally with each machine. Supervised learning algorithms require labeled data of normal and abnormal behavior, which is rarely available in a real-world system as it requires an enormous amount of time from experienced personnel to label enough data to train learning algorithms. The current study develops an algorithm that uses a one-class support vector machine algorithm to perform unsupervised conditioning monitoring. The condition monitoring problem was approached in a way that is similar to detecting anomalies, except that the algorithm focuses on identifying variations in the condition of the machine rather than detecting any condition that deviates from the normal. The inputs for the algorithm are the signals collected from proximity sensors attached to the rotor shaft processed by a discrete wavelet transformation. The algorithm was validated with different signals, including synthetically generated ones, and signals collected from sensors attached to real rotating machines. During the validation, the performance of the algorithm was measured using the F1 score, which showed a value of close to 0.9 for most of the Tests. This high value demonstrates the high capability of the algorithm in detecting variations in the machine conditions.*

Keywords: *one-class support vector machine, machine learning, anomaly detection, rotary machine, condition monitoring*

1. INTRODUCTION

The continuous monitoring of industrial rotating machines holds significant importance in ensuring their well-functioning, dependability, and overall safety. Furthermore, by minimizing unplanned downtime caused by preventive maintenance, costs are effectively reduced, as maintenance interventions are solely conducted when necessary (Tandon and Parey, 2006). There exist numerous systems that actively engage in real-time monitoring of rotating machines' health. Nonetheless, most of these systems necessitate the involvement of domain experts to define specific parameters that align with the monitored machine's characteristics (Aswin *et al.*, 2020; Qin *et al.*, 2012; Liu *et al.*, 2018).

As an alternative approach, certain systems employ machine learning algorithms with supervised or unsupervised learning techniques (Avci *et al.*, 2022). By allowing the algorithm to autonomously identify the current behavior of the rotating machine, the reliance on an expert to set the system parameters for each machine is significantly reduced.

There are different techniques used in the literature to identify anomalies, among them we can mention clustering (Loureiro *et al.*, 2004; Knorr *et al.*, 2000; Stetco *et al.*, 2019); recurrent neural networks (RNN) (Malhotra *et al.*, 2016; Yin *et al.*, 2020), autoencoders neural networks (Guo *et al.*, 2018; Amarbayasgalan *et al.*, 2018; Zong *et al.*, 2018). In addition to the techniques mentioned previously, there are other classes of machine learning methodologies for detecting anomalies, such as algorithms based on decision trees (Primartha and Tama, 2020); algorithms based on separation boundaries (Erfani *et al.*, 2016); and algorithm that use convolutional filters (Janssens *et al.*, 2016; Yin *et al.*, 2020) .

The present study aims to develop an algorithm utilizing the one-class support vector machine (OCSVM) approach to execute unsupervised condition monitoring. In this context, the condition monitoring task is addressed in a manner analogous to anomaly detection, albeit with a focus on identifying variations in the machine's state rather than only detecting any deviation from the normalcy. The algorithm's inputs are the signals collected from proximity or accelerometer sensors affixed to the rotor shaft. Consequently, the algorithm is capable of autonomously acquiring knowledge belonging to the normality patterns exhibited by the rotating machine. By doing so, it can effectively extract the most informative characteristics from the sensor signals without the need for an expert to define any parameters. Consequently, the algorithm can proficiently discern any deviations from the normal condition.

The algorithm was validated with different signals, including synthetically generated ones, and signals collected from sensors attached to real rotating machines. During the validation, the performance of the algorithm was measured using the F1 score, which showed a value of close to 0.9 for most of the tests. This high value demonstrates the high effectiveness of the algorithm in detecting variations in the machine conditions.

Additionally, when executing the algorithm on a laptop equipped with an 8th generation i5 processor, the processing time required for each second of the signal was around 5 to 6 milliseconds. This efficiency leads to a reasonable hypothesis that future real-time testing using an embedded board would be feasible.

1.1 Support Vector Machine (SVM)

The Support Vector Machine (SVM) algorithm was initially developed as a binary classification algorithm (Cortes and Vapnik, 1995) capable of distinguishing between two distinct classes. Unlike dense artificial neural networks, which are often considered black-box models due to their high complexity and non-linearity, the SVM algorithm operates based on a statistical and geometric interpretation of data in an N-dimensional feature space (Bishop and Nasrabadi, 2006). As a result, the SVM algorithm can be categorized as a white-box or gray-box model, depending on the level of non-linearity introduced through hyperparameters such as the activation function.

Due to its inherent simplicity compared to dense artificial neural networks, the SVM algorithm requires a lower-dimensional input data space (Bishop and Nasrabadi, 2006). While an artificial neural network can effectively map systems with thousands of features given a sufficient number of samples, less complex algorithms like SVM, OCSVM, and non-neural-network clustering algorithms necessitate dimensionality reduction techniques such as PCA or other feature extraction methods (Bishop and Nasrabadi, 2006; Hyndman *et al.*, 2015).

The main objective of the SVM algorithm is to find the optimal straight line, referred to as the boundary, that separates the data points of one class from another. However, unlike other linear classifiers, the SVM algorithm ensures that the distance between the extreme points of each class and the boundary is maximized. This notion of maximizing the margin, or "street," between the extreme points and the boundary enhances the robustness of SVM algorithms when tested with unseen data (Bishop and Nasrabadi, 2006). Figure 1 shows the location of the separation boundary after training the SVM algorithm with two input features.

Many studies have utilized SVM algorithms for anomaly detection in rotating machines (Wu *et al.*, 2012; Yang *et al.*, 2007; Zhang and Zhou, 2013). However, as previously mentioned, these studies relied on labeled data indicating the presence or absence of failures for training the models. To address the challenge of requiring labeled data for anomaly detection in rotating machines, some authors have employed the One-Class Support Vector Machine (OCSVM) algorithm (Vos *et al.*, 2022; McKinnon *et al.*, 2020; Tutivén *et al.*, 2022), eliminating the need for labeled data in such cases.

1.2 One-Class Support Vector Machine (OCSVM)

The One-Class Support Vector Machine (OCSVM) algorithm, as the name suggests, is a modified version of the original SVM algorithm designed to be trained with data from a single class (Cristianini and Shawe-Taylor, 2000). Consequently, the prevalent method for utilizing the OCSVM algorithm in anomaly detection is to adjust the model (i.e., perform training) solely using normal data or a dataset where the majority belongs to the normal class.

Unlike the SVM algorithm, the One-Class Support Vector Machine (OCSVM) algorithm operates as an unsupervised method, eliminating the need for data labeling into different classes. However, it does require the training data to be predominantly or partially representative of the normal class (Cristianini and Shawe-Taylor, 2000). This holds true for most unsupervised anomaly identification algorithms since, even in unsupervised training scenarios, the model must learn the patterns associated with normal data in order to effectively distinguish normal data from anomalous data.

The OCSVM algorithm can be used for both novelty detection and anomaly/outlier detection (Zhang *et al.*, 2007). In the case of novelty detection, all the training data must exclusively belong to the same class, meaning they must represent normal data entirely. For anomaly detection, however, it is possible to specify the percentage of data considered normal, acknowledging the challenge of labeling real-world data as solely belonging to the normal class in unsupervised problems.

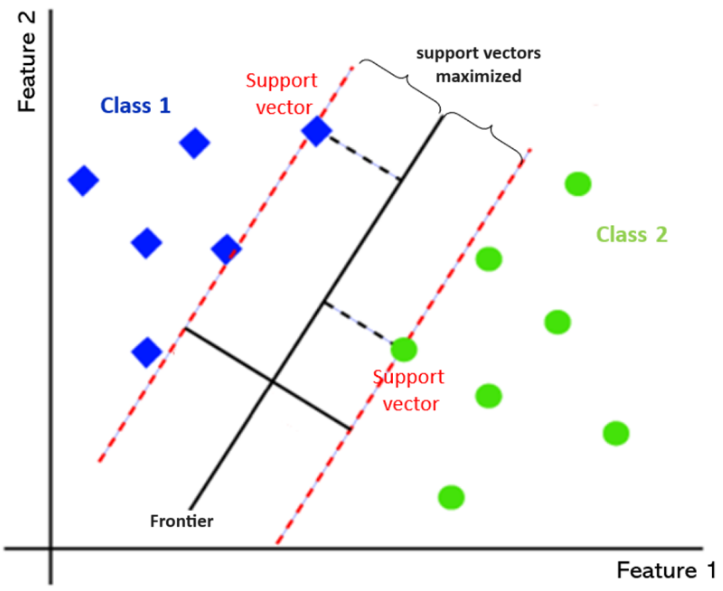

Figure 1. Location of the separation boundary after training the SVM algorithm with two input features.

Unlike the SVM algorithm, which establishes a separation line between two classes, the OCSVM defines a boundary (margin) that encompasses the entire set of normal data presented during training (Bounsiar and Madden, 2014). Following the adjustment process, any data located outside this boundary are classified as outliers (anomalous data), while data positioned within the boundary are classified as inliers (normal data).

Indeed, the OCSVM algorithm offers the capability to define two types of separation boundaries during its training process. One of these boundaries, known as the "soft margin," is determined by the internal parameter of the algorithm. This parameter controls the percentage of data that will be encompassed within the soft margin. The second separation boundary is established along the extreme data points of the training dataset.

This distinction between the Hard margin and the Soft margin is crucial as it enhances the algorithm's robustness in handling noise and outliers during the training phase (Zhang *et al.*, 2007). By incorporating both types of boundaries, the OCSVM algorithm becomes more adept at accommodating variations and abnormalities present in the data.

2. METHODOLOGY

In this section, we will provide a detailed description of the developed algorithm and the benchmark tests conducted to assess its performance. The primary objective of the work is to accurately detect significant variations occurring during the operation of a rotating machine. Hence, the tests were specifically designed to validate whether the algorithm could identify these variations at the precise moment they occur within each signal.

2.1 Test description

The algorithm's responses to 5 benchmark tests were thoroughly analyzed. A concise description of each test is provided.

Test 1 involved a numerically generated signal with 4 proximity sensors, sample frequency of 1000 Hz, and duration of 3600 seconds, specifically designed to exhibit unbalance and misalignment variations at some specific times.

Test 2 entailed a signal with 4 proximity sensors, sample frequency of 19638 Hz, and duration of 98 seconds, showcasing variations in speed. This signal was collected from a magnetic bearing bench located in the LMEst laboratory at the Federal University of Uberlândia (UFU).

Test 3 comprised a signal obtained from 4 accelerometer sensors, sample frequency of 16384 Hz, and duration of 92 seconds, featuring variations in speed and impacts. This signal was collected from an industrial exhaust fan located in the LAV laboratory at UFU.

Test 4 encompassed a signal collected from the same machine utilized in Test 2. However, the acquisition card employed for data collection was changed and the sample frequency has been increased to 10000 Hz.

Finally, Test 5 involved a signal obtained from a single accelerometer sensor, sample frequency of 4096 Hz, and duration of 984 seconds. The signals were downloaded from the internet (Rathore and Harsha, 2022). This signal pertained to a wear test of a rolling bearing, where it started with the bearing in a healthy state and concluded with the complete failure of the bearing.

2.2 Input Features

The input features from the OCSVM model are described below. In this article, the term "signal" will refer to a one-second window with no overlap of the sensors' amplitude values in the time domain. The time of one second was chosen, as it was a design requirement to have a response time every one second.

Numerous statistical features were tested in terms of their impact on the F1 score metric. These features included RMS, Skewness, Kurtosis, Peak Value, Entropy, Crest Factor, Shape Factor, and Estimated Rotation Speed. Following the tests, we noted that only the RMS feature, in combination with wavelet decomposition, yielded the highest score. To decompose the signal in the time domain, the discrete wavelet transform was employed using 5 levels. Subsequently, the root mean square (RMS) values of the resulting time signals from the wavelet decomposition were utilized as input features for the OCSVM algorithm. In this case, the dimensionality of the input data is 6, as it includes the signal from the low-pass component of the last level, as well as the signals from the high-pass components of the five levels.

To ensure consistency and facilitate effective analysis, the features were normalized within the range of [0,1]. Additionally, each feature was individually normalized with respect to the training data, ensuring that the feature values were scaled appropriately for the algorithm's processing.

2.3 Training

The division of the data into training and test sets followed a specific approach: the initial 'n' samples, where 'n' varied depending on the duration of the normal operational condition for each signal, were allocated to the training set. These initial 'n' samples were indicative of the machine's normal behavior. Subsequently, all remaining samples were assigned to the test set, and it was with these samples that performance scores were computed.

During the training phase of the OCSVM algorithm, the position of the separation hyper sphere was adjusted. The purpose of this adjustment was to effectively delineate the boundary between normal and anomalous data points in the feature space. The trained OCSVM model would then utilize this separation hyper sphere to classify unseen data points as either normal or anomalous based on their proximity to the boundary.

2.4 Threshold definition

Following the adjustment of the separation hyper sphere, the subsequent phase involves collecting a new dataset for determining the threshold. It is crucial to gather a dataset distinct from the training set to ensure greater reliability in identifying anomalies for out-of-sample data (data that differs from those encountered during training).

In this new dataset, the input features are extracted, and the OSCVM model generates an output, which represents the distance from the center of the hypersphere. The next step involves calculating the percentage difference between the output of each sample (signal) and a designated baseline sample, which is typically the last sample used during training.

By obtaining a vector of percentage differences between the outputs and the baseline sample, the threshold is computed using Equation (1). The threshold serves as a criterion for determining whether a particular sample is anomalous based on the percentage difference in relation to the baseline sample.

$$Threshold = \mu(x) + 5\sigma(x) \tag{1}$$

where $\mu(x)$ represents the mean of the vector of percentage differences between the outputs, and $\sigma(x)$ represents the standard deviation of the same vector.

It is important to note that this formula assumes that the input data follow a normal distribution. The threshold is set to separate 99.9% of the normal data, and any new data with an error greater than this threshold will be classified as anomalous.

The use of percentage difference in the calculation is significant because it captures the relative change in prediction compared to the baseline sample. This approach is more focused on identifying variations in machine behavior rather than solely detecting anomalous signals.

2.5 Operation

Once the threshold is determined, the operation of the algorithm proceeds as follows:

- Extract features from each new signal.
- Calculate the output of the model for the given signal.
- Compute the percentage difference between the current output and the output of the baseline sample.
- If the percentage difference value is below the threshold, classify the current signal as normal.

- If the percentage difference value exceeds the threshold, classify the current signal as anomalous and update the baseline sample to the last analyzed sample.

By following these steps, the algorithm effectively evaluates each new signal, comparing its output to the baseline sample and making a classification decision based on the calculated percentage difference. This approach enables the identification of anomalous signals that exhibit significant deviations from the baseline, while classifying normal signals that remain within the expected range of variation.

3. RESULTS AND DISCUSSION

This section will present the results of the 5 tests described above in section 2.1. Figures 1 to 5 show the signals for all sensors throughout the experiment time (blue line). The signals are in the time domain. For Tests 1, 2, and 4, in which proximity sensors were used, the value on the y-axis represents the displacement of the axis close to the sensor due to machine vibrations and the scale is in μV. For Tests 3 and 5, in which accelerometer sensors were used, the value on the y-axis represents the acceleration of the axis close to the sensor and the scale is in g.

The row below the signals (blue line) shows the algorithm answers (red line) and the targets (green line). The targets indicate the moments when some variation was occurring, and the answers indicate the moments when the algorithm identified the variation.

Based on the targets and obtained answers, we calculated and presented the metrics recall, precision, F1 score, and confusion matrix in tables 1 to 5. Recall represents the percentage of all positive instances correctly predicted as positive, indicating the proportion of identified variations. Precision indicates whether the algorithm predicts more positives than it should; higher values indicate better algorithm performance. The F1 score, which combines recall and precision, provides a more nuanced measure of performance. The closer the score is to 100, the better the results. Finally, the processing time of the algorithm to return the answer for each second of signal is shown. The algorithm was run on a Windows laptop with an eighth generation i5 processor.

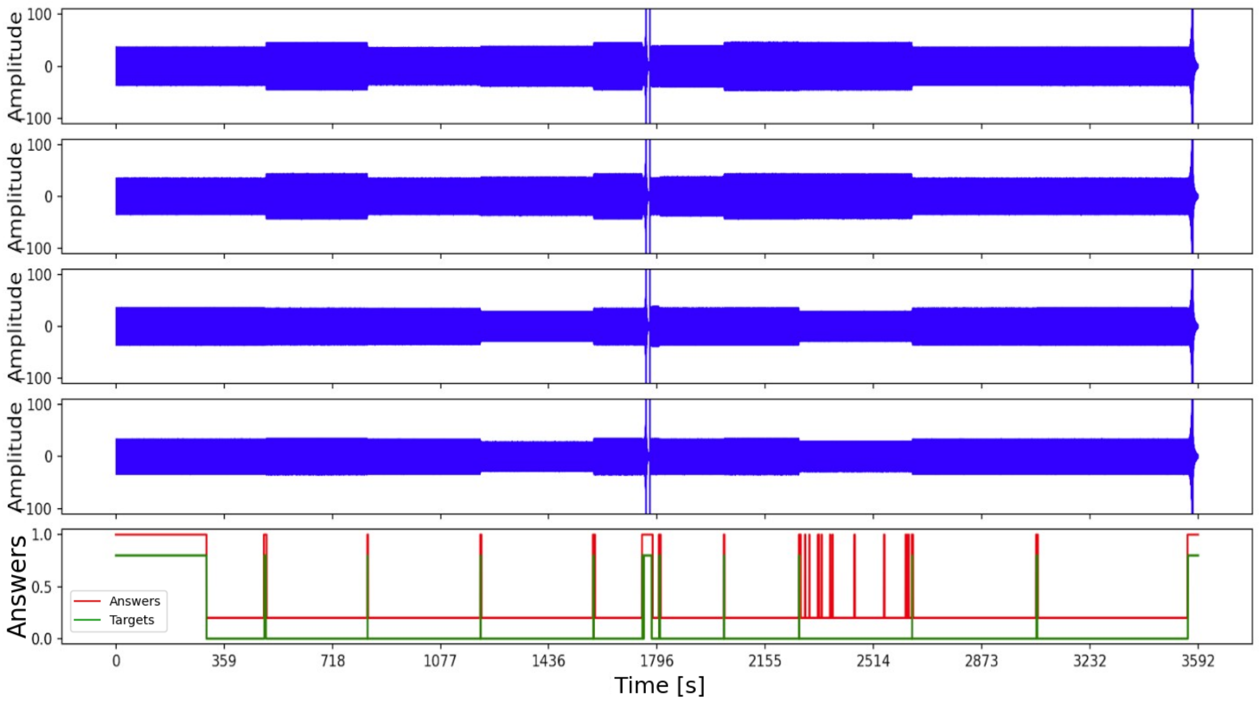

Figure 2. Algorithm response for Test 1.

Table 1. Algorithm performance metrics for Test 1.

Tn	Fp	Fn	Tp	Recall	Precision	F1 score	Time (s/signal)
3162	40	0	390	100	90.7	95.12	0.0041

Analyzing Figure 2, it can be seen that all regions of interest were identified (which can be proven by the recall of 100). Some regions close to the second 2514 were mistakenly identified as anomalous, probably due to the greater presence of noise in this region.

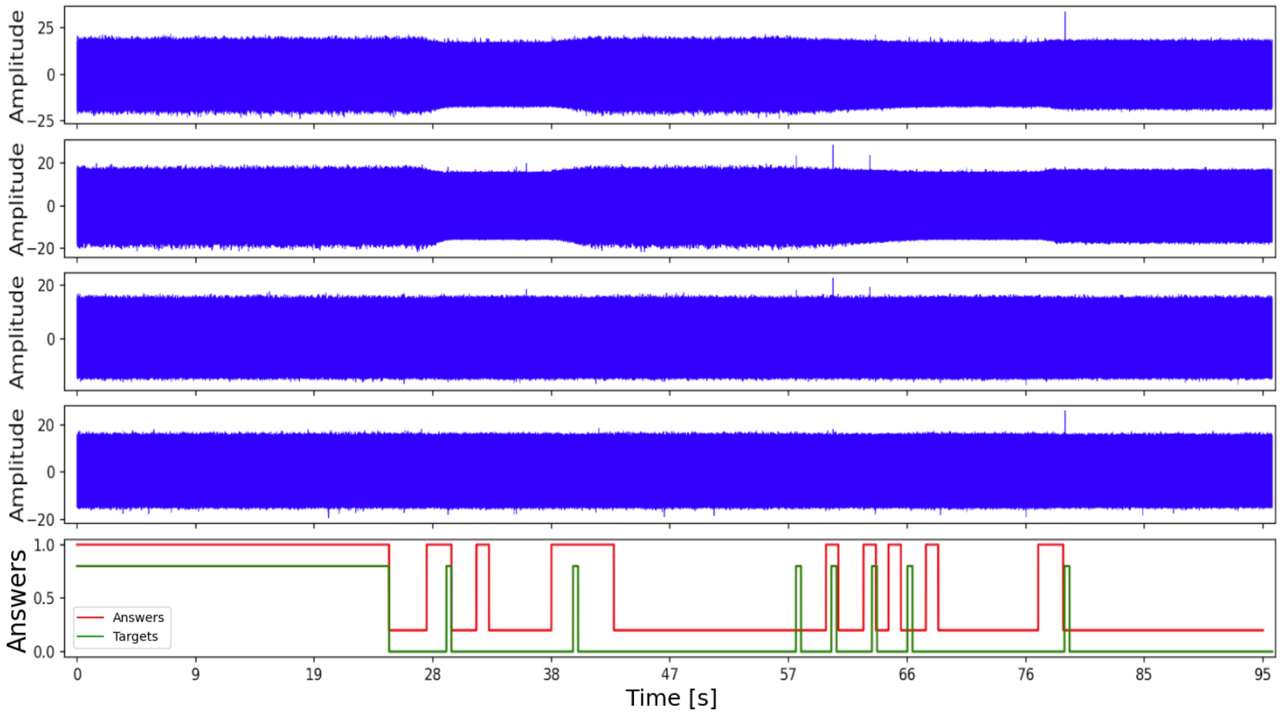

Figure 3. Algorithm response for Test 2.

Table 2. Algorithm performance metrics for Test 2.

Tn	Fp	Fn	Tp	Recall	Precision	F1 score	Time (s/signal)
52	9	4	30	88.24	76.92	82.19	0.0061

Figure 3 depicts four distinct moments of variation: the first occurring close to 28 seconds, the second close to 38 seconds, the third close to 66 seconds, and the fourth close to 76 seconds. However, it should be noted that the exact moment the algorithm identifies the variation does not always align with when the target identifies it. This discrepancy arises due to the nature of slow variations, such as the time interval between 57 seconds and 66 seconds, where the target only selected specific points rather than encompassing the entire moment. Even though the algorithm's response for this particular interval did not align precisely with the target, certain areas were still recognized as variations. This highlights the challenge of establishing a precise target for accurate validation. A similar observation holds true for the period after 76 seconds, where the algorithm identified the variation with a slight delay compared to the target.

Table 3. Algorithm performance metrics for Test 3.

Tn	Fp	Fn	Tp	Recall	Precision	F1 score	Time (s/signal)
32	6	1	50	98.04	89.29	93.46	0.0064

Figure 4 visually demonstrates the accurate identification of nearly all variations, as evidenced by the high recall score of 98.04. However, the precision score slightly decreased due to the algorithm's inclusion of additional seconds before and after the target variations.

Table 4. Algorithm performance metrics for Test 4.

Tn	Fp	Fn	Tp	Recall	Precision	F1 score	Time (s/signal)
114	11	0	49	100	81.67	89.91	0.0045

In the case of Test number 4, all variations were correctly identified, leading to a perfect recall score of 100. Only a small number of areas were erroneously selected, indicating a high level of accuracy overall.

Figure 6 exhibits results similar to those of test 2. The target specifically chose certain areas, and there were instances where the algorithm's responses were not perfectly synchronized. However, it is important to note that the areas identified by the algorithm were not necessarily incorrect despite the discrepancy in timing.

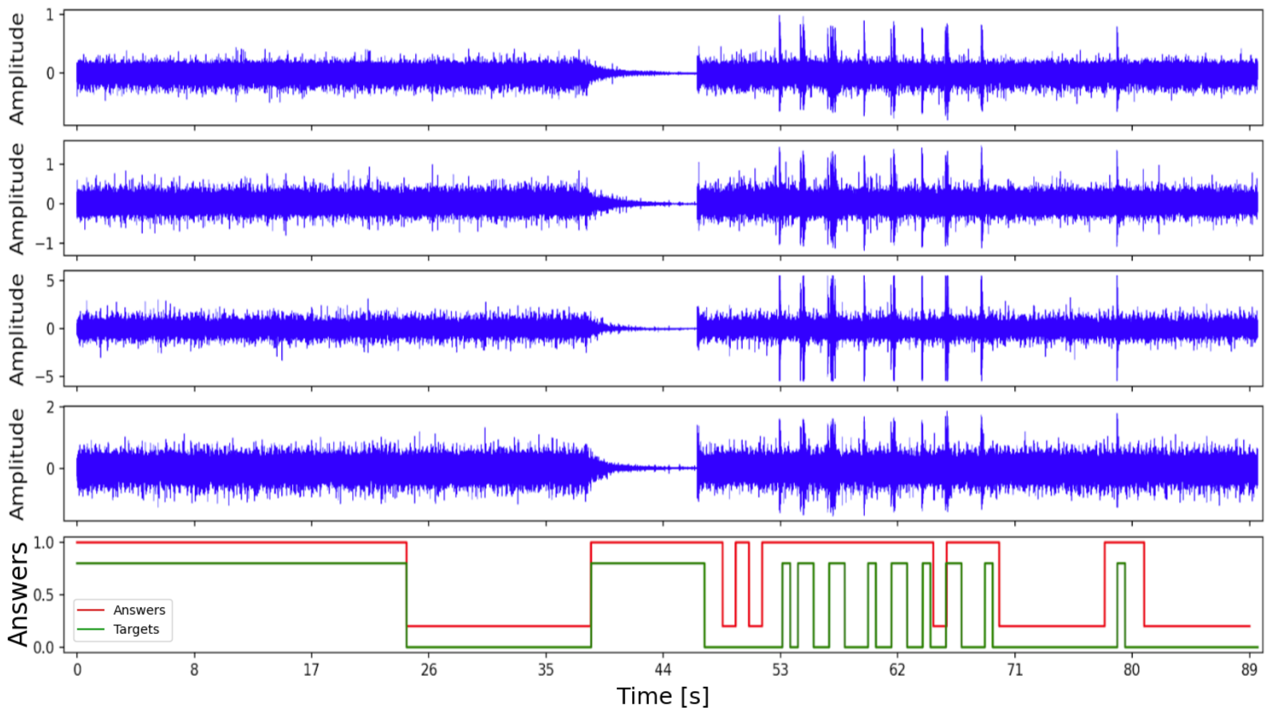

Figure 4. Algorithm response for Test 3.

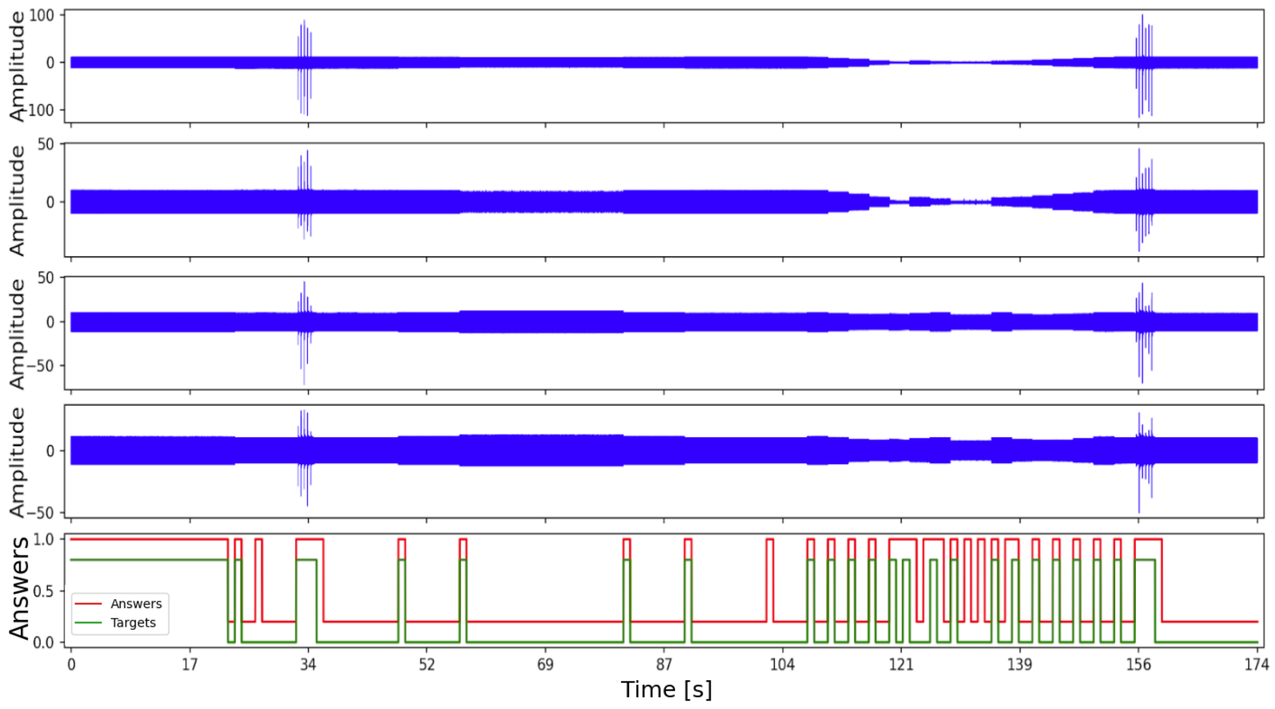

Figure 5. Algorithm response for Test 4.

Table 5. Algorithm performance metrics for Test 5.

Tn	Fp	Fn	Tp	Recall	Precision	F1 score	Time (s/signal)
567	4	165	248	60.05	98.41	74.59	0.0017

4. CONCLUSION

The proposed algorithm for real-time variation detection in rotating machines successfully identified a majority of the variations in the conducted tests, achieving an F1 score in the range of to 90% for Tests 1 to 4 and approximately 75%

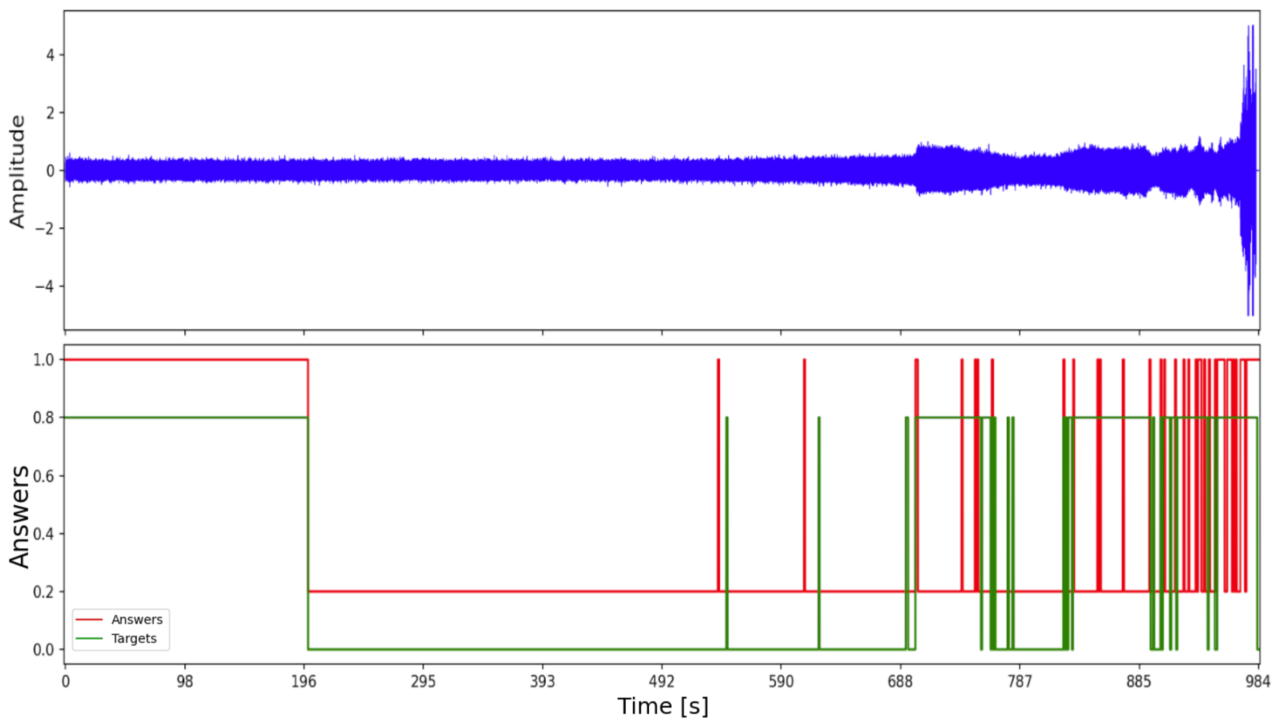

Figure 6. Algorithm response for Test 5.

for Test 5, probably because of the difficulties in creating a reliable target for this type of signal and for this specific application. The algorithm achieved a processing time of approximately 4 to 6 milliseconds per second of the signal when executed on a laptop with an 8th generation i5 processor. This efficient performance suggests the possibility of conducting real-time testing using an embedded board in future experiments.

5. REFERENCES

Amarbayasgalan, T., Jargalsaikhan, B. and Ryu, K.H., 2018. "Unsupervised novelty detection using deep autoencoders with density based clustering". *Applied Sciences*, Vol. 8, No. 9, p. 1468.

Aswin, F., Dwisaputra, I. and Afriansyah, R., 2020. "Online vibration monitoring system for rotating machinery based on 3-axis mems accelerometer". In *Journal of Physics: Conference Series*. IOP Publishing, Vol. 1450, p. 012109.

Avci, O., Abdeljaber, O., Kiranyaz, S., Sassi, S., Ibrahim, A. and Gabbouj, M., 2022. "One-dimensional convolutional neural networks for real-time damage detection of rotating machinery". In *Rotating Machinery, Optical Methods & Scanning LDV Methods, Volume 6: Proceedings of the 39th IMAC, A Conference and Exposition on Structural Dynamics 2021*. Springer, pp. 73–83.

Bishop, C.M. and Nasrabadi, N.M., 2006. *Pattern recognition and machine learning*, Vol. 4. Springer.

Bounsiar, A. and Madden, M.G., 2014. "One-class support vector machines revisited". In *2014 International Conference on Information Science & Applications (ICISA)*. IEEE, pp. 1–4.

Cortes, C. and Vapnik, V., 1995. "Support-vector networks". *Machine learning*, Vol. 20, No. 3, pp. 273–297.

Cristianini, N. and Shawe-Taylor, J., 2000. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge university press.

Erfani, S.M., Rajasegarar, S., Karunasekera, S. and Leckie, C., 2016. "High-dimensional and large-scale anomaly detection using a linear one-class svm with deep learning". *Pattern Recognition*, Vol. 58, pp. 121–134.

Guo, Y., Liao, W., Wang, Q., Yu, L., Ji, T. and Li, P., 2018. "Multidimensional time series anomaly detection: A gru-based gaussian mixture variational autoencoder approach". In *Asian Conference on Machine Learning*. PMLR, pp. 97–112.

Hyndman, R.J., Wang, E. and Laptev, N., 2015. "Large-scale unusual time series detection". In *2015 IEEE international conference on data mining workshop (ICDMW)*. IEEE, pp. 1616–1619.

Janssens, O., Slavkovikj, V., Vervisch, B., Stockman, K., Loccufier, M., Verstockt, S., Van de Walle, R. and Van Hoecke, S., 2016. "Convolutional neural network based fault detection for rotating machinery". *Journal of Sound and Vibration*, Vol. 377, pp. 331–345.

Knorr, E.M., Ng, R.T. and Tucakov, V., 2000. "Distance-based outliers: algorithms and applications". *The VLDB Journal*, Vol. 8, No. 3, pp. 237–253.

Liu, R., Yang, B., Zio, E. and Chen, X., 2018. "Artificial intelligence for fault diagnosis of rotating machinery: A review".

Mechanical Systems and Signal Processing, Vol. 108, pp. 33–47.

Loureiro, A., Torgo, L. and Soares, C., 2004. "Outlier detection using clustering methods: a data cleaning application". In *Proceedings of KDNet Symposium on Knowledge-based Systems for the Public Sector. Bonn, Germany*.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P. and Shroff, G., 2016. "Lstm-based encoder-decoder for multi-sensor anomaly detection". *arXiv preprint arXiv:1607.00148*.

McKinnon, C., Carroll, J., McDonald, A., Koukoura, S., Infield, D. and Soraghan, C., 2020. "Comparison of new anomaly detection technique for wind turbine condition monitoring using gearbox scada data". *Energies*, Vol. 13, No. 19, p. 5152.

Primartha, R. and Tama, B.A., 2020. "Anomaly detection using random forest: A performance revisited". In *2020 on data and software engineering (ICoDSE)*. IEEE, pp. 1–6.

Qin, Q., Jiang, Z.N., Feng, K. and He, W., 2012. "A novel scheme for fault detection of reciprocating compressor valves based on basis pursuit, wave matching and support vector machine". *Measurement*, Vol. 45, No. 5, pp. 897–908.

Rathore, M.S. and Harsha, S., 2022. "Rolling bearing prognostic analysis for domain adaptation under different operating conditions". *Engineering Failure Analysis*, Vol. 139, p. 106414.

Stetco, A., Dinmohammadi, F., Zhao, X., Robu, V., Flynn, D., Barnes, M., Keane, J. and Nenadic, G., 2019. "Machine learning methods for wind turbine condition monitoring: A review". *Renewable energy*, Vol. 133, pp. 620–635.

Tandon, N. and Parey, A., 2006. "Condition monitoring of rotary machines". *Condition monitoring and control for intelligent manufacturing*, pp. 109–136.

Tutivén, C., Vidal, Y., Insuasty, A., Campoverde-Vilela, L. and Achicanoy, W., 2022. "Early fault diagnosis strategy for wt main bearings based on scada data and one-class svm". *Energies*, Vol. 15, No. 12, p. 4381.

Vos, K., Peng, Z., Jenkins, C., Shahriar, M.R., Borghesani, P. and Wang, W., 2022. "Vibration-based anomaly detection using lstm/svm approaches". *Mechanical Systems and Signal Processing*, Vol. 169, p. 108752.

Wu, S.D., Wu, P.H., Wu, C.W., Ding, J.J. and Wang, C.C., 2012. "Bearing fault diagnosis based on multiscale permutation entropy and support vector machine". *Entropy*, Vol. 14, No. 8, pp. 1343–1356.

Yang, Y., Yu, D. and Cheng, J., 2007. "A fault diagnosis approach for roller bearing based on imf envelope spectrum and svm". *Measurement*, Vol. 40, No. 9-10, pp. 943–950.

Yin, C., Zhang, S., Wang, J. and Xiong, N.N., 2020. "Anomaly detection based on convolutional recurrent autoencoder for iot time series". *IEEE Transactions on Systems, Man, and Cybernetics: Systems*.

Zhang, R., Zhang, S., Muthuraman, S. and Jiang, J., 2007. "One class support vector machine for anomaly detection in the communication network performance data". In *Proceedings of the 5th conference on Applied electromagnetics, wireless and optical communications*. Citeseer, pp. 31–37.

Zhang, X. and Zhou, J., 2013. "Multi-fault diagnosis for rolling element bearings based on ensemble empirical mode decomposition and optimized support vector machines". *Mechanical Systems and Signal Processing*, Vol. 41, No. 1-2, pp. 127–140.

Zong, B., Song, Q., Min, M.R., Cheng, W., Lumezanu, C., Cho, D. and Chen, H., 2018. "Deep autoencoding gaussian mixture model for unsupervised anomaly detection". In *International Conference on Learning Representations*.

6. RESPONSIBILITY NOTICE

The authors are solely responsible for the printed material included in this paper.

7. Acknowledgement

Alexandre Tavares received support from CAPES and Petrobras gas and oil Inc Reseach Grants.

C. Nataraj and A. Abassi received partial support from Office of Naval Research Grants, N-00014-19-1-2070 and N-000014-22-1-2480 with Captain Lynn Petersen as the program manager, This support is gratefully acknowledged.