Developing a Data-Driven Unsupervised Pattern Recognition Approach for Sensor Signal Anomaly Detection

Ensieh Iranmehr [1], Ricardo Ferreira [2], Tim Bohnert [2], and Paulo Freitas [2]

[1]International Iberian Nanotechnology Laboratory (INL)
[2]Affiliation not available

October 30, 2023

Abstract

Coming up with a system for early detection of machine damages and failures is one of the important challenges in the industrial maintenance procedure to avoid additional costs and downtimes. To approach this goal, this paper uses the signal gathered by a sensing system which employed a spintropic sensor to measure the magnetic field around the machine which somehow shows the machine's behaviour. Using this signal and focusing on analysing and processing the signal, this paper develops a data-driven method to recognize signal patterns and subsequently detects anomalies. A challenging task that we succeeded to overcome in this paper is recognizing relevant signal patterns without having any prior knowledge. An algorithm designed for this task is therefore completely unsupervised which makes it consistent and suitable to apply it for the signals gathered for other types of machines. Using both frequency and time domain information, the proposed algorithm, which utilizes signal processing and machine learning techniques, is able to efficiently identify relevant signal patterns. Clustering results on the real data gathered by the aforementioned sensor have shown the high accuracy of 99.38% in recognizing patterns. Furthermore, an anomaly score measure is used and according to its distribution, anomalies are detected appropriately.

Developing a Data-Driven Unsupervised Pattern Recognition Approach for Sensor Signal Anomaly Detection

Ensieh Iranmehr, Ricardo Ferreira, *Member, IEEE*, Tim Böhnert, and Paulo P. Freitas, *Senior Member, IEEE*

***Abstract*—Coming up with a system for early detection of machine damages and failures is one of the important challenges in the industrial maintenance procedure to avoid additional costs and downtimes. To approach this goal, this paper uses the signal gathered by a sensing system which employed a spintropic sensor to measure the magnetic field around the machine which somehow shows the machine's behaviour. Using this signal and focusing on analysing and processing the signal, this paper develops a data-driven method to recognize signal patterns and subsequently detects anomalies. A challenging task that we succeeded to overcome in this paper is recognizing relevant signal patterns without having any prior knowledge. An algorithm designed for this task is therefore completely unsupervised which makes it consistent and suitable to apply it for the signals gathered for other types of machines. Using both frequency and time domain information, the proposed algorithm, which utilizes signal processing and machine learning techniques, is able to efficiently identify relevant signal patterns. Clustering results on the real data gathered by the aforementioned sensor have shown the high accuracy of $99.38\,\%$ in recognizing patterns. Furthermore, an anomaly score measure is used and according to its distribution, anomalies are detected appropriately.**

Index Terms* - *Pattern Recognition. Clustering. Anomaly Detection, Sensor Signal, Hybrid Time-Frequency Domain Method. Data-Driven.

I. Introduction

Anomaly detection refers to the tasks of identifying the rare events or observations that are deviated from its normal frequent behaviour. Today, anomaly detection is broadly used in many research areas such as health monitoring [1], [2], [3], [4], [5], [6] for example heart disease diagnosis [1] and neuromuscular disorders diagnosis [5], environment monitoring such as sewer pipeline fault identification [7] and solar farms anomalies detection [8], and machine condition monitoring [9], [10] for example machinery fault diagnosis [11], [12], [13], [14], [9]. Depending on the anomaly detection problem, it is required to design algorithms which are able to identify anomalies in different types of data such as image [15], [2], [16], [17], video [7], sound signal [9] speech signal [18], sensor signal [19], [5], text [20], spatio-temporal data [4], streaming data [21] and time-series [22], [23]. Hence, it seems that no general solution works for all of the anomaly detection problems.

There are several anomaly detection techniques which are model-based [24], [25]. It means prior physical knowledge for modelling specific types of anomalies are required in model-based methods. Machine learning techniques lead us to be able to identify anomalies with a limited requirement of prior physical knowledge. These are called data-driven methods [8], [6], [26] which are principally rely on the information gathered by data acquisition system.

Ensieh Iranmehr, Ricardo Ferreira, Tim Böhnert, and Paulo P. Freitas are with International Iberian Nanotechnology Laboratory (INL) 4715-330 Braga, Portugal (e-mail: ensieh.iranmehr@inl.int, ricardo.ferreira@inl.int, tim.bohnert@inl.int, Paulo.Freitas@inl.int)

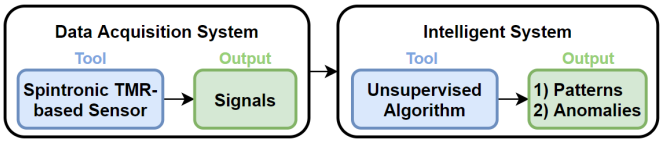

Fig. 1: Proof of concept demonstration for anomaly detection. Non-invasive tools provide the data (signal) representing the machine's behaviour. The signal is then fed to an intelligent system to extract the relevant patterns and anomalies.

Data-driven methods can be categorized into three main categories: 1) supervised [14], [1], [8], [27], 2) semi-supervised [28], and 3) unsupervised [6], [26], [3], [23] anomaly detection. Since the labelled data, which is required for supervised anomaly detection, is often not available or in other words, collecting sufficient anomolous samples is infeasible in most of the cases, many researchers have tried to detect anomalies without labelled data. However, they were forced to take advantage of supervised learning along with unsupervised for other steps of their algorithms such as pattern recognition to achieve better diagnosis [29], [30].

For anomaly detection, there are several machine learning approaches categorized into simple statistical techniques [31] such as Z-Score, interquartile range and histogram or distribution plot, density based methods such as local outlier factor (LOF) [32], connectivity-based outlier factor (COF) [33], distance-based methods [34] and isolation forest (iForest) [35].

Detecting faults in machines is a hot topic nowadays due to this fact that anomalies might cause a significant degradation in machine's performance leads to wasting huge amount of money and time. Although several valuable works have been published in the field of detecting specific faults in some machines [11], [12], [13], [14], [9], it still seems a long way ahead to reach an intelligent system which is able to early detect anomalies automatically without having any knowledge about the machines. In this paper, anomaly does not mean bad behaviour, but rather rare/new/unexpected behaviour, which if not detected may lead to bad behaviour of the machine. Hence, this paper tries to get closer to this aim by designing a scheme for machine early anomaly detection using machine learning and signal processing techniques.

Fig. 1 demonstrates the proof of concept for non-invasive machine early anomaly detection. A spintronic sensor based on tunnel magnetoresistance (TMR) [36], [37] is placed nearby a machine which measures a changing magnetic field around that machine. This sensor was previously used in some other proof of concepts [38], [39], [40] such as real-time interaction

system for eye movement gesture control [38] which successfully classified eye movements. In this work, the TMR sensor's output signal which somehow shows the machine's behaviour is then fed into an intelligent system in order to detect signal patterns and subsequently anomalies. The latter forms the basis of present study in which an algorithm is proposed that can recognize patterns and anomalies efficiently.

In this paper, it is assumed that we do not have any prior physical knowledge about the machine. Hence, using machine learning techniques, we want to get as much information as it is possible from the TMR sensor's output signal by developing an unsupervised data-driven approach. Using this data, we want to detect abnormal behaviour of the machine before damage, failure or breakdown. Hence, the important task that we are going to do in this work is to recognise signal patterns in the input data unsupervisedly. Each signal pattern represents a state of the machine. Thus, by recognizing the patterns, we are able to understand the machine's behaviour. Deviation from average of signal patterns, informs that something unusual has happened which leads us to predict the failure or damage before it happens which is a very essential and challenging step in maintenance.

There are many supervised and unsupervised pattern recognition techniques including different types of neural networks and machine learning techniques. Depending on the data type and the problem to be solved, different algorithms have been developed so far. Among them, feed-forward [41], [42], recurrent [43] and reservoir neural networks [44], [45], [46] should be mentioned as powerful tools to handle labelled data. If we encounter a problem that data labels are not available, we can benefit of some other techniques such as dimension reduction [47], clustering [48], [49] to recognize patterns. The latter along with some signal processing techniques was used in our proposed algorithm since a completely unsupervised solution for the problem is our desire. Using patterns extracted from the input data, this paper used one of the simplest statistical technique (Z-Score) to compute anomaly score. Besides, another method was used to find rare pattern sequences which were then considered as anomalies.

Concisely, the main contribution of this paper is: 1) Developing a data-driven completely unsupervised algorithm for recognizing patterns; 2) Using hybrid time-frequency domain information in the proposed pattern recognition algorithm; 3) Utilizing some statistical techniques to find anomalies; 4) Conducting a proof-of-concept validation on real data gathered by the aforementioned sensor.

The paper is organized as follows. Section II focuses on the intelligent system designed for patterns recognition and anomaly detection. In this section, we explain in detail the whole algorithm analysing hybrid time-frequency domain information using signal processing and machine learning techniques and also explains the algorithm used for finding anomalous using some statistical techniques. Section III includes the results obtained from every step to achieve the goal. Moreover, performance of the proposed algorithm relying on several evaluation parameters is studied. We also discuss advantages of the present study and express some future works. Finally, we conclude the paper in section IV.

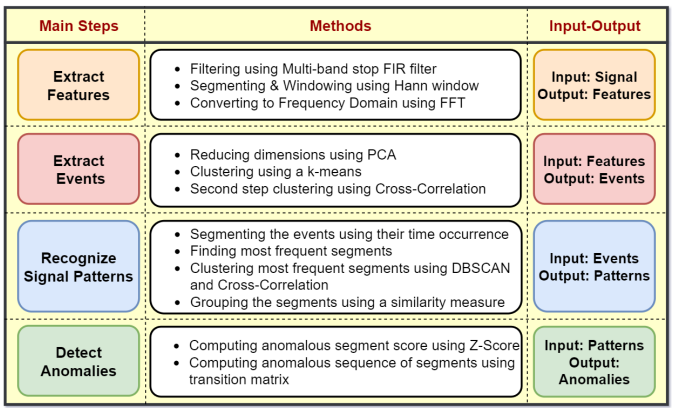

Fig. 2: Overview of the scheme (main steps, methods and input/output of each step) designed to recognize patterns and anomalies.

II. Proposed Scheme for Pattern Recognition and Anomaly Detection

This section explains the scheme proposed for pattern recognition and anomaly detection. An overview of the whole algorithm designed for this work was demonstrated in Fig. 2. In this figure, the main steps of the proposed algorithm, the input/output of each step and the methods used for every step was briefly shown. As it can be seen in Fig. 2, the proposed scheme consists of 4 main steps: 1) feature extraction; 2) event extraction; 3) signal pattern recognition; and 4) anomaly detection. By combining different methods and concepts, we were able to provide a completely unsupervised data-driven system for signal pattern recognition. Then, using the recognized signal patterns and by using some simple statistical techniques, anomalies were detected. In the following subsections, we discuss about all of the steps in detail.

A. Extract Features from Signal

Here, the methods used in this paper to extract features from the signal gathered by the TMR sensor is explained. This main step consists of three steps. First, by designing a filter to remove poweline harmonics, the signal is filtered and it is used for time-domain analysis. The signal is segmented into the same-sized windows. Each signal segment is then windowed with a smoothing window function. Afterwards, to extract the frequency domain features, each signal segment is transformed into the frequency domain. Frequency components related to powerline harmonics are removed and the remaining frequency components are considered as features. Thus, we take advantage of both ways (time and frequency) to look at the signal.

1) Filtering the signal: To filter the powerline harmonics, a finite inpulse response (FIR) multi-band stop filter was designed, shown in Fig. 3. As it can be seen from this figure, only frequencies in some frequency bands related to the powerline harmonics are attenuated and the other frequencies are passed. The order of filter was considered $N_I = 400$ and to approach an optimal filter design, we used the equiripple method. If $x[n]$ is the input signal, $h[n]$ is the impulse response of the designed filter, the filtered signal $y[n]$ is computed by convolving $x[n]$ with $h[n]$ (Eq. 1). In this equation, b_i denotes

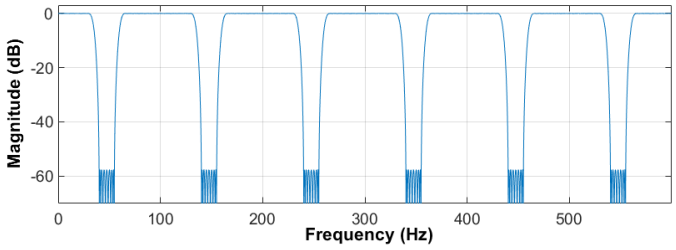

Fig. 3: FIR multi band stop filter designed for removing the 50 Hz power line noise and its harmonics from the signal

the value of the impulse response at the i^{th} instant of the designed filter.

$$\begin{aligned} y[n] &= x[n] * h[n] \\ h[n] &= \sum_{i=0}^{N_I} b_i.\delta[n-i] \end{aligned} \tag{1}$$

2) Segmenting the signal: The length of signal to be transformed into the frequency domain is so large that we have to subdivide it into smaller signal segments. Otherwise, the desired frequency resolution can not be achieved. Depending on the sampling rate frequency of the signal (F_s) and the fact that to reconstruct a signal to its frequency domain, new length is the next power of 2 from the signal window, the length of each window was set (S_w). Otherwise, the signal will be padded with trailing zeros. Lets assume that the number of signal windows is N_w. Also, to avoid the edges of each signal window from being lost, the signal windows should overlap with each other for which an overlap of O_w was considered.

3) Transforming into frequency domain: To deconstruct a time domain representation of a signal into its frequency domain, Fast Fourier Transform (FFT) is used. Due to the significant limitation of FFT based on the fact that FFT interprets two endpoints of the signal connected to each other, we improved the signal clarity using windowing before applying FFT. Hence, we applied Hanning, a smoothing window function, since it is generally more satisfactory compared to other window functions. Smoothing window reduces the amplitudes of boundaries of the signal window toward zero, smoothly and gradually. If $x[n]$ is a signal window, the windowing result $y[n]$ is computed using Eq. 2 where $w[n]$ is the Hanning window. Afterwards, y is deconstructed to frequency components using Eq. 3. Finally, removing the frequency bands of powerline harmonics from the magnitude spectrum leads to extraction of features. In the following, the windowed signal segments were denoted by $\{w_1, w_2, ..., w_{N_w}\}$ and the frequency components of magnitude spectrum before and after removing powerline harmonic frequencies were shown by $\{f_1^i - f_{N_f}^i | 1 \leq i \leq N_w\}$ and $\{f_1^i - f_N^i | 1 \leq i \leq N_w\}$, respectively, where $N < N_{fft}$.

$$\begin{aligned} y[n] &= x[n] \times w[n] \\ w[n] &= 0.5 - 0.5 \times \cos(\tfrac{2\pi n}{N_{fft}}),\ 0 \leq n \leq N_{fft} \end{aligned} \tag{2}$$

$$Y[k] = \sum_{n=0}^{N_{fft}-1} e^{-i\frac{2\pi kn}{N_{fft}}} y[n] \tag{3}$$

B. Extracting Events

Here, how to extract events from the frequency domain features and the time domain information is explained. A dimension reduction technique is applied on the frequency domain features to transform the high dimensional features to low dimensional components. The extracted components are then fed into a k-means method in order to be clustered. Using some techniques based on cross-correlation (CC) method, some of the event clusters are selected and then divided into two different groups in order to distinguish between the signals having similar frequency domain features but different time domain information.

1) Reducing dimension of features: To transform high-dimensional features into a low-dimensional features so that the latter retains most of the meaningful information, we can take advantage of dimension reduction techniques. We used principle component analysis (PCA) [47] which is an important linear technique that performs mapping of data to a lower dimensions by maximizing the data variance in low dimensional representation using singular value decomposition (SVD). The number of principle components was determined in such a way that low dimensional representation contains a specific percent of total variability (V_{PC}), denoted by $\{PC_1^i - PC_n^i | 1 \leq i \leq N_w\}$.

2) Clustering into event clusters: So far, each signal segment has been represented by its extracted principle components. Here, how all of the signal segments cluster into different event clusters using their PCs is explained. In other words, we want to put the signal segments, whose behaviours are much alike, in a same group. Hence, the similarity between their corresponding PCs should be measured with an appropriate measure. Algorithm 1 presents the algorithm proposed to cluster the signal segments. The proposed algorithm consists of three parts: 1) first level clustering; 2) find relevant event clusters; and 3) second level clustering.

In the first level clustering, all the signal segments cluster into different groups. Depending on the initial centroids or seeds, the clustering result will be different. Hence, we run the k-means method in several iteration (N_{itr}), compute within-cluster sums of point-to-centroid distance for each iteration ($Dist_{within}$), find the iteration with minimum distance (I), and finally, run the 1-iteration k-means using the centroids computed in I^{th} iteration. For this purpose, we used the well-known k-means which is a centroid-based clustering method [49] for the first level clustering. Afterwards, the most frequent events is considered as background.

So far, the clustering has been done based on frequency domain information. What if in some signal segments, the frequency components of magnitude spectrum are similar while in the time domain are inverted? To overcome this challenge, it is first required to find the relevant clusters and then run the second level clustering algorithm on these clusters. To find relevant event clusters, the transition probability between event clusters is calculated. The relevant clusters are the ones that have not happened too much and the variance of its transition probabilities are not minimum or maximum.

In the second level clustering, each of the signal segments belonging to the relevant cluster are compared with its first signal segment using cross-correlation after normalization. Eq. 4 expresses the cross-correlation between two signals where $\overline{S_1}$ denotes the conjugate of S_1. If two signals are very

Algorithm 1 Clustering the PCs into Event Clusters

Assumption
Number of Clustering Iterations $= N_{itr}$
Number of Event Clusters $= k$
Input/Output
Input: Principle Components (PCs) of signal segments
Input: Signal segments (S_e)
Output: Event Clusters (C_E)

Clustering Level 1
for every iteration $i = 1$ to N_{itr} **do**
Cluster PCs using k-means
Find within-cluster distances ($Dist_{within}(i)$)
end for
$I = argmin_i(Dist_{within})$
Run 1-iteration k-means using I^{th} centroids
Remove the most frequent events (background)
$k \Leftarrow k - 1$

Find Relevant Clusters
for every Event Cluster $i = 1$ to k **do**
for every Event Cluster $j = 1$ to k **do**
$t^e_{ij} = prob(S^{m+1}_e \in C^i_E | S^m_e \in C^j_E)$
end for
$\sigma^2_i = Var(T = \{t^e_{ij}\}^k_{j=1})$
end for
$I1 = argmax_i(\sigma^2)$
$I2 = argmin_i(\sigma^2)$
$RC = \{1, ..., k\} \backslash (I1 \cup I2)$

Clustering Level 2
for $i = 1$ to Number of Relevant Clusters ($|RC|$) **do**
$k \Leftarrow k + 1$
for $j = 1$ to Number of Events in C^i_E **do**
Normalize $S^i_e(j)$
$CC^j = S^i_e(j) \otimes S^i_e(1)$
Find Minima, Maxima and their Prominence (P)
if $P_{min} > \alpha \times P_{max}$ **then**
$C^k_E = C^k_E \cup event_j$
end if
end for
end for

similar, the most prominent maximum (P_{max}) is significantly larger than the most prominent minimum (P_{min}), whereas the opposite is true if the signals are inverted. As a result, if the most prominent extremum is maximum, the signal segment belongs to a same cluster. Otherwise, it belongs to a new created cluster.

$$(S_1 \otimes S_2)[n] = \sum_{m=-\infty}^{+\infty} \overline{S_1[m]} S_2[m+n] \tag{4}$$

C. Recognizing Signal Patterns

The algorithm designed for recognizing signal patterns consists of three steps, which are explained here. First, the consecutive events are subdivided into different segments depends on their time of occurrences. Afterwards, the segments that were repeated the most are found and clustered into appropriate groups using a density-based clustering algorithm applied for

Algorithm 2 Clustering the frequent sequence segments into Pattern Clusters

Assumption
Number of frequent sequence segments $= N_f$
Input/Output
Input: Signals of most frequent sequence segments (S_f)
Output: Pattern Clusters (C_P)

Clustering Level 3
$M_{cc}, Max_{cc} \Leftarrow 0$
for every frequent sequence segment $i = 1$ to N_f **do**
for every frequent sequence segment $j = 1$ to N_f **do**
Normalize $S_f(i)$ and $S_f(j)$
$CC^j = S_f(i) \otimes S_f(j)$
$Max_{cc}(i, j) = max(CC^j)$
Find Minima, Maxima and their Prominence (P)
if $P_{max} > \alpha \times P_{min}$ **then**
$M_{cc}(i, j) = 1$
end if
end for
$Mean_{cc}(i) = \frac{1}{N_f} \Sigma_j (Max_{cc}(i, j))$
end for
$\mu_{cc} = \frac{1}{N_f} \Sigma_i (Mean_{cc}(i))$
for every frequent sequence segment $i = 1$ to N_f **do**
if $Mean_{cc}(i) < \beta \times \mu_{cc}$ **then**
$M_{cc}(i) = 0$
end if
end for
Cluster M_{cc} using DBSCAN (C_P)

the cross-correlation results. By proposing a similarity measure, each segment is compared with the segments repeated a lot in order to find the appropriate group to which this segment belongs. After grouping all the segments, a dataset for each pattern cluster containing the corresponding segments' signal patterns are provided.

1) Segmenting the consecutive events: So far, we could identify the events that are presented by their corresponding event clusters (C_E) and their time of occurrences. Although in some part of the signal, no event has been identified because those part detected as background, on the contrary, some of these events occurred consecutively. Here, the consecutive events whose time of occurrence is near (less than t) are considered as a sequence. As a result, the events are transformed into the fewer number of sequences, called sequence segments (Seg).

2) Clustering the frequent sequence segments into pattern clusters: Of all the sequence segments, some have happened a lot. However, their corresponding signals might be very similar. Hence, we first find the frequent segments and then cluster them in order to put the similar frequent sequence segments into a same group. Algorithm 2 expresses the way to cluster these frequent sequence segments (third level clustering). In this algorithm, we first extract the features appropriate for third level clustering, which is based on the results of cross-correlation between corresponding signals (S_f) of each pair of frequent segments. By taking advantage of the same method used in section II-B2, a matrix (M_{cc}) with the size

of $(N_f \times N_f)$ is created. It is assumed that each row of M_{cc} represents an observation with N_f features which are then fed into a density-based spatial clustering of applications with noise (DBSCAN) [50]. Due to its advantage not having to specify the number of clusters, we chose DBSCAN method for the third level clustering. DBSCAN clusters the observations based on two parameters $epsilon$ and $minpts$ which indicate neighbourhood search radius and a minimum number of neighbours, respectively. If some of the frequent sequence segments represents noisy signals, then DBSCAN clusters them in separate groups. To group these sequence segments in one cluster, before applying DBSCAN, the maximum value of cross-correlation is computed (Max_{cc}) and for each frequent segment, its average ($Mean_{cc}$) is computed. If it is less than a coefficient (β) multiplied by the total average (μ_{cc}), they are considered as the first pattern cluster.

3) Grouping all the sequence segments using a similarity measure: So far, we figured out how many pattern clusters exists in the dataset. Here, we want to allocate each sequence segment or its corresponding signal (signal pattern) to an appropriate pattern cluster. For simplicity, instead of sequence segments/signal patterns, we will say samples. Algorithm 3 expresses the way to measure the similarity of all samples to pattern clusters and finally, consider it as a member of the most similar pattern cluster set. The similarity measure is based on both the frequency and time domain information, i.e. sequence segments and their corresponding signal patterns. As it can be seen from the algorithm, distance ($Dist$) between every sequence segment and the frequent ones are computed, and the index with minimum distance ($I1$) determines its label if minimum distance is short and the cardinality of $I1$ equals to 1 (cardinality of sets were shown by $|.|$). Otherwise, we take advantage of time domain analysis using the same method mentioned in section II-B2 to measure similarity between signal patterns and to find the index with maximum similarity ($I2$). The intersection between $I1$ and $I2$ determines label of the sample if it is not empty. Otherwise, the maximum index of Max_{cc} determines the label. As a result, all samples are recognized and now, we are able to make dataset for each pattern cluster consisting of all the samples belonging to it. It should be noted that the function used to compute ($Dist$) between two sequence segments returns the minimum number of elements which must be omitted from each of them or both, so that the remaining elements of sequence segment are the same.

Algorithm 3 Grouping all the sequence segments into Pattern Clusters

Assumption
Number of all segments (Seg) or samples $= N_s$
Number of frequent segments (Seg_f) $= N_f$
Number of Pattern Clusters (C_P) $= N_c$
Input/Output
Input: Segments (Seg) & their corresponding signals (S)
Output: Predicted label of i^{th} segment (L^i)

for every segment $i = 1$ to N_s **do**
  for every frequent segment $j = 1$ to N_f **do**
    Find distance ($Dist$) between $Seg(i)$ and $Seg_f(j)$
  end for
  $I1 = argmin_j(Dist)$
  if $|I1| = 1$ & $min(Dist)$ is short **then**
    $L^i = \{k | \, 1 \leq k \leq N_c, \; I1 \in C_P^k \}$
  else
    for every frequent segment $j = 1$ to N_f **do**
      Normalize $S(i)$ and $S_f(j)$
      $CC^j = S(i) \otimes S_f(j)$
      Find Minima, Maxima and their Prominence (P)
      $V_{cc} \Leftarrow 0$
      if $P_{max} > \alpha \times P_{min}$ **then**
        $V_{cc}(j) = 1$
      end if
    end for
    $I2 = argmax_j(V_{cc})$
    if $I1 \cap I2 \neq \emptyset$ **then**
      $I = I1 \cap I2$
      if $|I| = 1$ **then**
        $L^i = \{k | \, 1 \leq k \leq N_c, \; I \in C_P^k \}$
      else
        Count elements of I belonging to C_Ps (N_{EI})
        $L^i = argmax(N_{EI})$
      end if
    else
      for every frequent segment $j = 1$ to N_f **do**
        $Max_{cc}(j) = max(CC^j)$
      end for
      $I = argmax_j(Max_{cc})$
      $L^i = \{k | \, 1 \leq k \leq N_c, \; I \in C_P^k \}$
    end if
  end if
end for

D. Detecting Anomalies

Up to now, we explained all the details about recognizing signal patterns. Here, the proposed method to find anomalies relying on the results of previous steps is described. We are supposed to detect two types of anomalies in this work: one is the samples deviated from the norm of the pattern cluster to which it belongs, and two is the sequence of patterns which happen rarely. For detecting the first type of anomaly, we compute the anomaly score and by using a threshold, anomalies are found. Whereas for the second type of anomaly, we compute the transition probability between each pair of pattern clusters. In this way, the most frequent sequences of patterns and also the sequences which happens rarely can be found using the transition probability. Algorithm 4 expresses how to identify anomalies.

1) Detecting anomalous samples: Sometimes, the samples are categorized in a wrong cluster or differ significantly from the corresponding cluster norm. If the latter happens, it may be due to the poor condition of the machine or in other words, it may indicates a malfunction of the machine that can lead to the machine's breakdown. Hence, detecting anomalous samples is a very important task to prevent failures. To detect anomalous signal patterns or sample, a simple statistical method based on Z-Score is used. Eq. 5 expresses how to compute Z-

Algorithm 4 Detecting Anomalies

Assumption
Number of Pattern Clusters $(CL) = N_c$
Total number of samples (s) = N_s
Input/Output
Input: Signals of all samples (S^i) belonging to C_P^i
Input: Predicted label of m^{th} samples (L^m)
Output: Anomalies (A)

Anomalous Pattern
for every cluster $i = 1$ to N_c **do**
  Zero-padding the signals belonging to i^{th} cluster
  $\mu^i = S^i(1)$
  for every sample in i^{th} cluster $j = 1$ to N_{si} **do**
    Find lag between $S^i(j)$ and μ^i
    Shift μ^i by lag
    $\mu^i = \mu^i + S^i(j)$
  end for
  $\mu^i = \frac{\mu^i}{N_{si}}$
end for
for every cluster $i = 1$ to N_c **do**
  for every sample in i^{th} cluster $j = 1$ to $|C_P^i|$ **do**
    $D(j) = (S^i(j) - \mu^i)^2$
  end for
  $\sigma^i = \sqrt{\frac{\Sigma_j(D(j))}{N_{si}}}$
  for every sample i^{th} cluster $j = 1$ to N_{si} **do**
    $Score = \frac{\sqrt{D(j)}}{\sigma^i}$
  end for
end for

Anomalous Sequence of Pattern
for every cluster $i = 1$ to N_c **do**
  $P^i = \frac{|C_P^i|}{N_s}$
  for every cluster $j = 1$ to N_c **do**
    $t_{ij} = P^i \times prob(s_{m+1} \in C_P^i | s_m \in C_P^j)$
  end for
end for
$T = \{t_{ij}\}_{i,j=1}^{N_c}$
$Rare_{PS} = \{(i,j) | t_{ij} < \theta\}$
for every sample $m = 1$ to N **do**
  if $Score > \nu$ or $(L^m, L^{m+1}) \subseteq Rare_{PS}$ **then**
    $A = A \cup S^i$
  end if
end for

Score where μ, σ and x represent mean, standard deviation and a sample, respectively. The scoring method used here is similar but slightly different. The difference is that the score is calculated in such a way that is always positive.

$$Z_{Score} = \frac{x - \mu}{\sigma} \tag{5}$$

In the method used here, the average of all samples belonging to each pattern cluster is computed. Afterwards, the deviation of each signal pattern from the average indicates the degree of anomaly. As it can be seen from Algorithm 4, to compute the average of all signal patterns, all the signal patterns must be changed to a same size using zero-padding which simply increases the signal's length by adding zeros to the start and end of it. Then, by using cross-correlation, the delay (lag) between signals is computed. By shifting one of the signal as much as the computed lag and then summing all the same-sized shifted signals, the average of i^{th} pattern cluster samples (μ^i) and the standard deviation of i^{th} pattern cluster are computed. If the $Score$ is less than a threshold (ν), the sample deviation from the mean is small and it will be considered as normal or non-anomalous sample. Otherwise, the deviation from mean is large and we can consider it as anomalous sample.

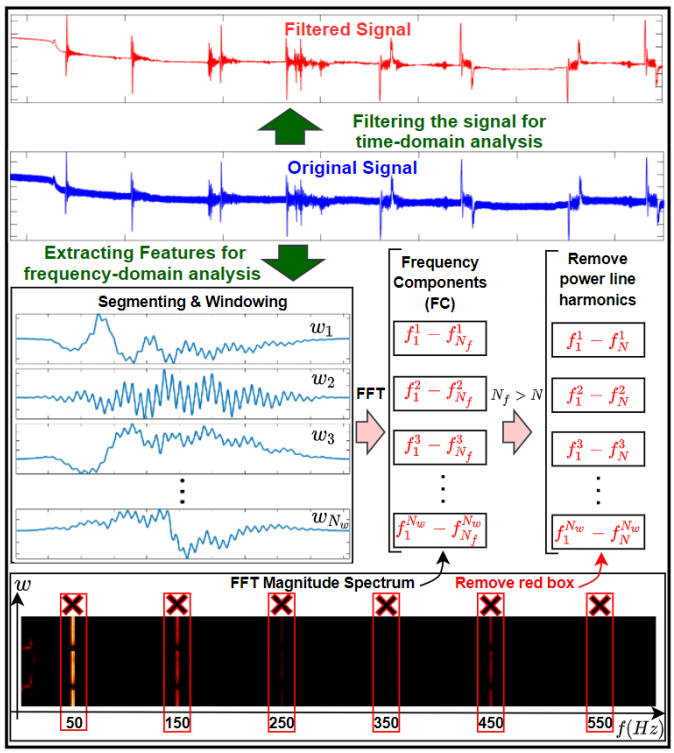

Fig. 4: Steps to extract information from the signal. A small part of original signal was shown here filtered using the designed FIR filter to be suitable for time domain analysis. On the other hand for frequency domain analysis, the original signal were segmented to the same-sized windows each of which was then windowed by a Hanning window. Afterwards, it was converted into the frequency domain using FFT. The magnitude spectrum of FFT represents the frequency components (FCs). The FCs relating to the power line harmonics were removed and the remaining FCs were considered as features.

2) Detecting anomalous pattern sequences: It is worth mentioning that the disposition of signal patterns in relation to each other is important because each signal pattern indicates a specific state or mode of the machine. Hence, the order of signal patterns recognised in the data can be referred to basic information of the machine's behaviour. Here, we are going to distinguish common signal patterns from rare signal patterns, which could indicate anomalous sequences. For this purpose, state transition probability between every pair of pattern clusters occurring consecutively is computed. The state transition probability between i^{th} and j^{th} pattern clusters is computed by $prob(s_{m+1} \in C_P^i | s_m \in C_P^j)$, that indicates the conditional probability of the $(m+1)^{th}$ sample belonging to C_P^i given the $(m)^{th}$ sample belonging to C_P^j. By computing the probability of each pattern cluster occurring and

multiplying with its corresponding state transition probability, transition probability matrix ($T = \{t_{ij}\}_{i,j=1}^{N_c}$) is computed. The matrix T indicates the occurrence probability of different pairs of pattern clusters. In this algorithm, $|C_P^i|$ is cardinality of the set of i^{th} pattern cluster. If the transition probability between pairs is less than a threshold (θ), i.e. these transitions are rarely happened and they may be anomalous sequences. Anomalous sequences may be the order of more than two signal patterns. In this case, we can find the transition probability between more than 2 pattern clusters and find the ones less than threshold as rare pattern sequences.

III. Results and Discussion

In this section, performance of the system designed to recognize patterns and detect anomalies is presented. The simulation results of each main step are shown separately and also the signal patterns which are detected as anomalies are shown. To fully investigate performance of the proposed system, we labelled the patterns manually and then, using some evaluation parameters, the actual results was compared with the desired ones. Finally, the advantages of the proposed approach and some future works are discussed.

A. Results of the algorithm's main steps

Here, the results of proposed algorithm's main steps are depicted. Fig. 4 shows the results of every single step to extract frequency features for frequency domain analysis as well as filtered signal for time domain analysis. As it is seen, for frequency domain analysis, the original signal was first segmented and windowed to the N_w same-sized windows which the amplitude of their boundaries were gradually decreased. Each window was transformed into frequency components (FCs) using FFT and the FCs related to power line harmonics were then removed. The remaining components indicates the frequency domain features. As a result, we have N_w window segments each of which represented by N features.

Fig. 5 shows the results of first and second level clustering of the principle components (PCs) to extract events. As it is seen, the features of each window segment was transformed into the lower dimensions. Each window segment was therefore represented by its corresponding n dimensional principle components. The window segments were then clustered using the k-means method that its result was shown in the scatter plots. The window segments belonging to a specific cluster were shown by an individual sign and colour. In the first level clustering, the window segments were first clustered into different groups. The optimal value of k is automatically determined by an internal evaluation of the clustering for a range of different values of k. Here, $k = 5$ was determined by the algorithm with one cluster representing the background, which leaves 4 event clusters. However, during the conversion into the frequency domain, some time domain specific information

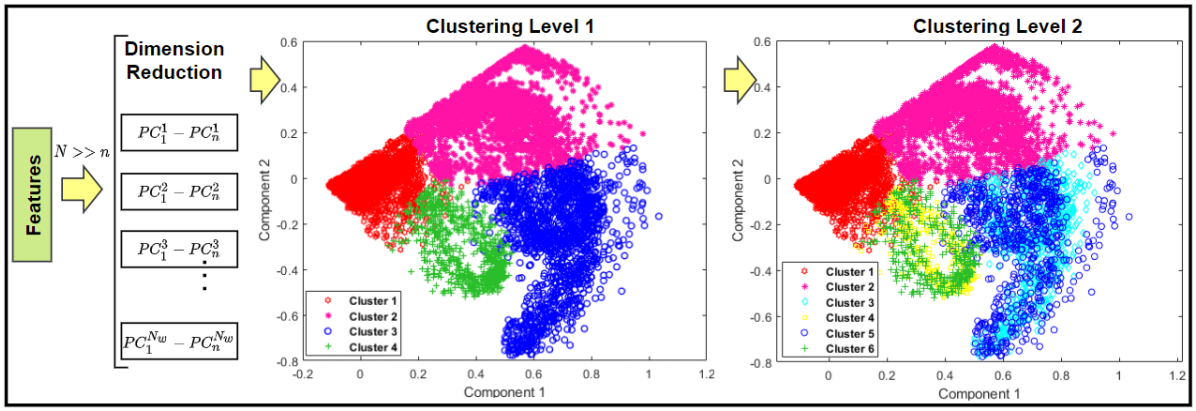

Fig. 5: Displaying the results of how to extract different events from the features (FC). First, PCA reduced the dimension of each window's features and transformed it to a smaller number of components (PC). The PCs extracted from all windows was then clustered into k event clusters (Clustering Level 1). Then, using a proposed method based on cross-correlation, some of the clusters were chosen and each of which was then split into 2 different clusters (Clustering Level 2), leading to more than k event clusters.

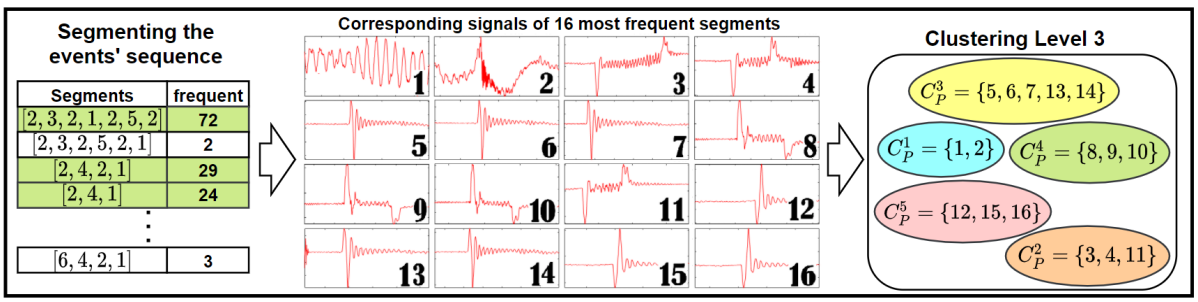

Fig. 6: Depiction of the steps to recognize signal patterns. The consecutive events were segmented based on their time of occurrences. Then, the k-most frequent segments ($k = 16$ in this case) were selected. By running cross-correlation between each pair of frequent segments' corresponding signals and then by taking advantage of DBSCAN method, the similar items were grouped automatically into clusters.

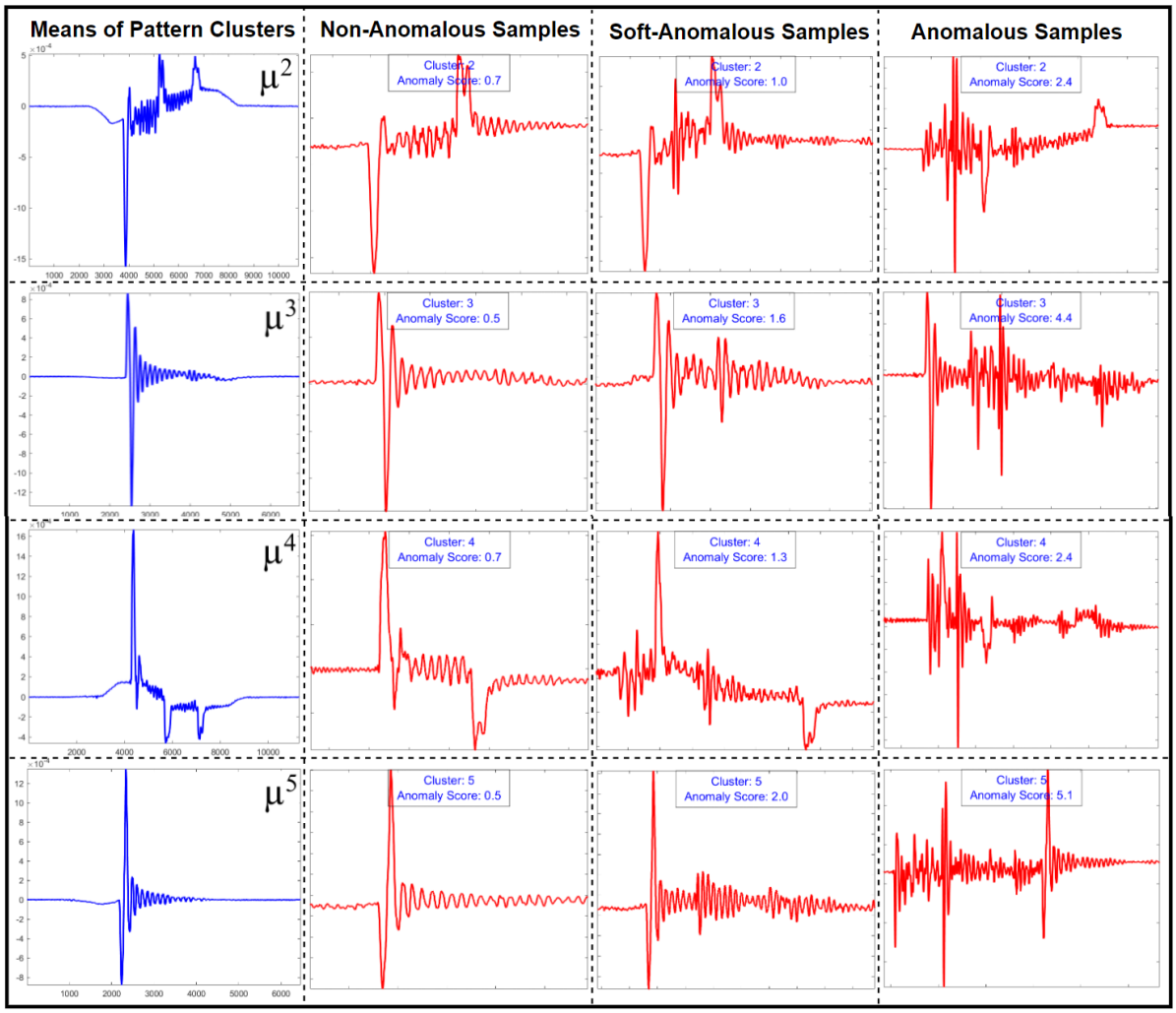

Fig. 7: Depiction of the pattern clusters' seeds or averages and displaying some non-anomalous, soft-anomalous and anomalous samples. The average of all signal patterns belonging to every pattern cluster were computed, shown in first column. Then, by computing anomaly score based on Z-Score, a non-anomalous, soft-anomalous and anomalous sample belonging to each pattern cluster were shown in second, third and forth columns, respectively.

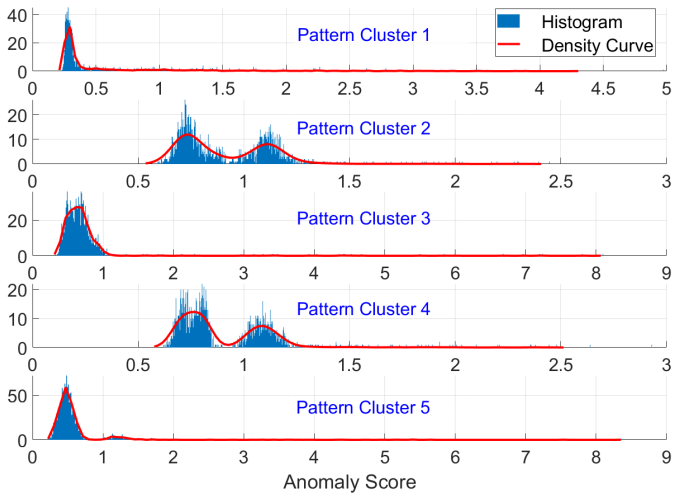

Fig. 8: Anomaly distribution of every pattern cluster. The histogram of anomaly score considering all of the signal patterns belonging to each pattern cluster was displayed here. Also, a density function fitted to each histogram was drawn to show the anomaly score distribution.

got lost. For example, this dataset shows a characteristic peak that is either positive or negative in the time domain, but indistinguishable in the frequency domain. Thus, using the proposed cross-correlation-based method in second level clustering, these different behaviours become distinguishable. It is worth mentioning that this method is not specific for this dataset and it could be applied to a general dataset. Because if all of the samples belonging to a cluster are similar in time domain, all of them will be remained in the same cluster.

In this dataset, the relevant clusters were this occurred are $Cluster\,3$ and $Cluster\,4$ and they were split into two clusters using the cross-correlation based method resulted in the total 6 different clusters (C_E), as shown in Fig. 5. As a result, except the window segments related to the background, others were considered as different types of events.

Considering the time of events occurrences, they were segmented into sequence segments which were then used to find the frequent ones. Fig. 6 shows the corresponding signals of $N_f = 16$ most frequent sequence segments which were then clustered using DBSCAN method. As it can be seen, they

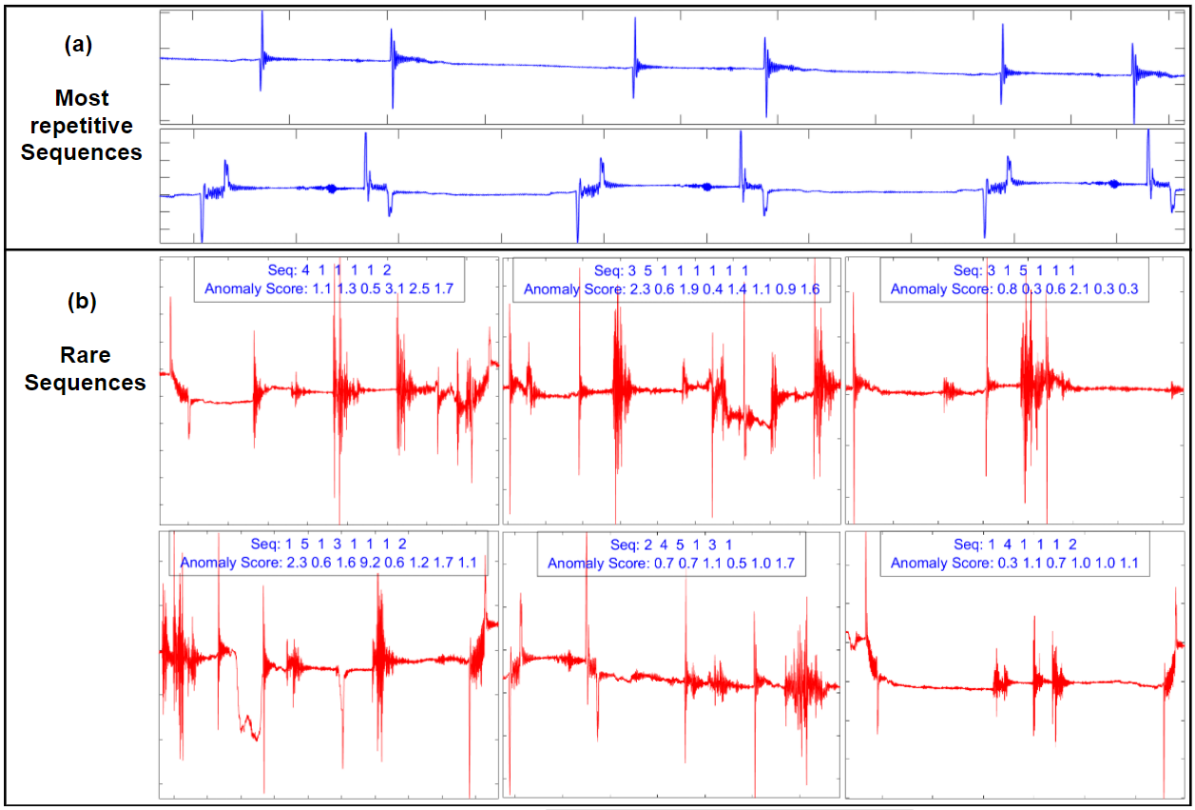

Fig. 9: Displaying several examples of normal and anomalous sequences of signal patterns. a) Two most repetitive sequences detected as normal sequences, b) some rare sequences detected as sequence anomalies.

have been clustered into 5 different pattern clusters (C_P). By measuring similarity between each sequence segment and the frequent ones, it was grouped into one of the 5 pattern clusters, which led to the recognition of all signal patterns.

To find anomaly, the average of all signal patterns existing in each pattern cluster was calculated and the deviation of each signal pattern from the mean indicates degree of anomaly. The mean or seed (μ) of second to fifth pattern clusters have been shown in the first column of Fig. 7. For each pattern cluster, a non-anomalous or normal, a soft-anomalous and an anomalous sample have been shown. Higher score indicates higher probability of an anomalous signal pattern. From all the samples for non-anomalous signal patterns, it is obvious that the anomaly score less than a value leads to non-anomalous samples. Whereas, the signal patterns with anomaly score more than this value can be considered as soft-anomalous and anomalous samples depending on the anomaly score distribution of each pattern cluster. The anomaly distribution of each pattern cluster has shown in Fig. 8. As it can be seen, the total standard deviation of first, third and fifth pattern clusters are much smaller than second and forth ones, leading to narrower distributions. Moreover, it is seen that the anomaly score range in the second and fourth pattern clusters are less than the rest. According to these distributions and the score range for each pattern clusters, we can set the parameters to find soft-anomalies and anomalies. If density curve and its most prominence local maximum are denoted by D and D^{max}, respectively, the thresholds to find soft-anomalies and anomalies can be computed using Eq. 6 and Eq. 7.

$$\begin{aligned} \nu_L &= Score(\tfrac{D^{max}}{\gamma_1}) \\ \nu_H &= \tfrac{max(Score)-\nu_L}{\gamma_2} + \nu_L \end{aligned} \tag{6}$$

$$\begin{aligned} \nu_L \leq Score \leq \nu_H, &\mapsto Soft\ Anomalies \\ Score > \nu_H, &\mapsto Anomalies \end{aligned} \tag{7}$$

It is also clearly seen in Fig. 8 that the density curve of cluster 2 and 4 have two peaks unlike the other clusters which have one peak. This happened due to the two different width of signal patterns belonging to each of these clusters. It should be noted that because of this, the mean of these pattern clusters (Fig. 7) are a little different from their corresponding non-anomalous samples. In this case, two set of thresholds should be set, each of which should be applied for the corresponding samples. For this purpose, two representative samples, each of them represents a peak, are compared with all the samples belonging to the cluster and the samples are grouped to a more similar one (G_1 or G_2) using a cross-correlation based method. If two most prominence local maxima and the most prominence local minimum are denoted by D_1^{max}, D_2^{max} and D^{min}, respectively, soft-anomalies and anomalies samples can be computed using Eq. 8 and Eq. 9. Using the parameters specified in Table II, the threshold values were computed and mentioned in Table I and based on these values, the soft-anomalous and anomalous samples were found.

Also, very rare sequences of signal patterns are considered as sequence anomalies. As mentioned earlier, they were found

TABLE I: Anomaly score thresholds for different pattern clusters automatically extracted based on the distributions

Pattern Clusters	**1**	**2**	**3**	**4**	**5**
ν_{L1}	0.42	0.93	1.1	0.91	0.8
ν_{H1}	1.19	1.06	2.49	1	2.31
ν_{L2}	-	1.3	-	1.26	-
ν_{H2}	-	1.53	-	1.51	-

using the probability transition matrix between different pattern clusters. The sequence anomalies were extracted and their corresponding signals can be shown as anomalous sequences. Some of the anomalous sequences have been shown in Fig. 9.b. As seen in this figure, each anomalous sequence may consists of different number of signal patterns. The sequence of corresponding pattern clusters and their anomaly scores have been displayed above each sequence. To see the difference between these rare sequences and the normal sequences, two most repetitive sequences of signal patterns have been shown in Fig. 9.a.

$$\begin{gathered}\nu_{L1} = Score(D^{min}) - \frac{|Score(D^{min}) - \nu_{L1}|}{\lambda_1} \\ \nu_{H1} = \nu_{L1} + \frac{|Score(D^{min}) - \nu_{L1}|}{\lambda_2} \\ \nu_{L2} = Score(\frac{D^{max}}{\gamma_1}) \\ \nu_{H2} = \frac{max(Score) - \nu_L}{\gamma_2} + \nu_{L2}\end{gathered} \tag{8}$$

$$\begin{gathered}((sample \in G_1 \,\&\, \nu_{L1} \leq Score \leq \nu_{H1})\, or \\ (sample \in G_2 \,\&\, \nu_{L2} \leq Score \leq \nu_{H2})) \mapsto Soft\, Anomalies \\ ((sample \in G_1 \,\&\, Score > \nu_{H1})\, or \\ (sample \in G_2 \,\&\, Score > \nu_{H2})) \mapsto Anomalies\end{gathered} \tag{9}$$

B. Evaluation Process

All of the parameters necessary for the proposed algorithm have been specified for this real data and shown in Table II. Also, the sampling rate and the duration of dataset has been shown. Since the ultimate aim of this work is to figure out the machine's behaviour quite unsupervisedly, a small part of dataset (about 10 %), that is unlabelled, was used in the training stage in order to extract effective and helpful information, which were then used in the test stage. The information extracted from training stage and used in test stage is the coefficients of PCA, the centroids of first level clustering, the clustering results of third level and the mean of pattern clusters. Using this information, we could diagnose anomaly.

To evaluate the scheme designed for signal pattern recognition, we used different evaluation parameters including Accuracy (Acc), Precision ($Prec$), Sensitivity (Sen), Specificity ($Spec$), F1Score ($F1S$) and Purity (Pu) which are expressed by the equations in Eq.10. In these equations, the number of true positive, false positive, true negative and false negative are denoted by TP, FP, TN and FN. Also, N_d is the total number of signal patterns, N_c is the number of clusters, C_P^i is i^{th} pattern cluster and $Target^j$ is the desired target of j^{th} pattern.

The performance of the proposed clustering model using the above evaluation parameters have been shown in Table III. All of these parameters were computed for training and test sets.

TABLE II: Dataset Information and Parameters Specification

Dataset Information					
F_s	1200 Hz		Duration	22 Hours	
Parameters Specification					
S_w	0.85 Sec	O_w	75 %	N_{fft}	1024
V_{pc}	95 %	k	5	N_{itr}	20
$epsilon$	1	$minpts$	2	t	S_w
α	1.5	N_f	26	β	0.8
θ	0.5 %	γ_1	15	γ_2	5
λ_1	10	λ_2	3		

TABLE III: The clustering results of the training and test sets using the proposed algorithm

	Acc(%)	**Prec(%)**	**Sens(%)**	**Spec(%)**	**Pu(%)**	**F1S(%)**
Training	99.37	99.15	99.08	99.85	99.37	99.11
Test	99.38	99.24	99.27	99.85	99.38	99.21

It is clearly seen from this table that the proposed algorithm for recognizing signal patterns has excellent performance in both training and test sets. The clustering accuracy of 99.38 % achieved on test set confirms that the model was trained correctly without knowing anything about the labels.

$$\begin{gathered}Acc = \frac{TP+TN}{TP+TN+FP+FN} \\ Prec = \frac{TP}{TP+FP} \\ Sens = \frac{TP}{TP+FN} \\ Spec = \frac{TN}{TN+FP} \\ F1S = \frac{2}{\frac{1}{Sens}+\frac{1}{Prec}} \\ Pu = \frac{1}{N_d}\sum_{i=1}^{N_c} max_j(C_P^i \cap Target^j)\end{gathered} \tag{10}$$

Moreover, confusion matrix was computed for training and test sets individually and the results have been shown in Table IV. The target clusters was compared with the predicted output clusters. The diagonal cells indicate the number of total observations which were correctly recognized. Whereas, the off-diagonal cells indicate the number of observations which were incorrectly clustered. In these both confusion matrices, the far right columns show the precisions of every clusters, while the lowermost rows show the sensitivity of every clusters. The overall accuracy was shown in bottom right cell.

Furthermore, using the proposed method finding the soft-anomalies and anomalies of signal patterns, the number of samples in every cluster detected as soft-anomalies and anomalies were brought in Table V. Using the total number of samples belonging to each pattern cluster, the percentage of soft-anomalies and anomalies were computed and shown. It can be seen from this table that 3.88 % and 5.41 % of all the signal patterns were detected as anomalies and soft-anomalies, respectively.

TABLE IV: Confusion Matrix for training and test sets

Training Set							
		Target Cluster					
		1	***2***	***3***	***4***	***5***	%
Output Cluster	***1***	80	1	0	1	0	97.56
	2	0	199	0	0	0	100
	3	0	0	155	0	0	100
	4	1	0	0	198	0	99.50
	5	2	0	0	0	152	98.70
	%	96.39	99.50	100	99.50	100	99.37

Test Set							
		Target Cluster					
		1	***2***	***3***	***4***	***5***	%
Output Cluster	***1***	776	10	1	2	2	98.10
	2	0	1457	0	0	0	100
	3	11	2	1136	0	0	98.87
	4	1	0	0	1471	0	99.93
	5	7	1	0	0	1144	99.31
	%	97.61	99.12	99.91	99.86	99.83	99.38

TABLE V: The number and percentage of soft-anomalous and anomalous samples in entire dataset.

Pattern Clusters	**1**	**2**	**3**	**4**	**5**	**Total**
Total (Num)	873	1656	1304	1671	1306	6810
Soft Anomalies (Num)	160	81	10	20	98	369
Soft Anomalies (%)	18.33	4.89	0.77	1.19	7.5	5.41
Anomalies (Num)	128	39	23	53	21	264
Anomalies (%)	14.66	2.36	1.76	3.17	1.61	3.88

C. Discussion and Future Works

Applying the proposed algorithm on the real data gathered by a sensor, led us to conducting the proof of concept validation. Also, the high accuracy, precision, sensitivity and other evaluation parameters demonstrate high efficiency of the proposed algorithm in pattern recognition. Considering both time and frequency domain information in different steps of the proposed algorithm enabled us to achieve these promising results. The time domain information was used to distinguish upward and downward shape of the patterns, which significantly increased accuracy and evaluation of other parameters. The anomalies could be detected efficiently because of its dependence to the results of pattern recognition. Also, it should be noted that the proposed algorithm requires reasonable resources, which can be easily run in real-time.

An important advantage of the proposed algorithm relates to its learning which is totally unsupervised and will be useful in cases where no prior knowledge is available. This property made the algorithm suitable for analysing other signal data and subsequently recognizing signal patterns. One of the future work we are going to do is gathering more data from the aforementioned sensor to figure out how well the proposed algorithm works for other signal data. It is worth mentioning that the algorithm was designed in such a way to have the fewest parameters and least need to initialize them. However, to increase its generalization, we are going to use some optimization algorithm to set the parameters automatically.

Other advantage of the proposed algorithm relates to the size of the training set compared to the test set. We used almost 2 hours of data for training stage and the rest (about 20 hours) for the test stage. Actually, we could figure out the machine's states and their transitions just by using a small amount of data. Using this information extracted from the training stage, we are able to analyse all the data that will be collected for the same machine. Moreover, as it can be seen from the accuracy of the results, the training of the algorithm on the training set was representative for the test set. In addition the unsupervised learning, enables the algorithm to train on easily available data without prior knowledge of the system and achieve high reliability.

In the future, we plan to associate the detected anomalies with the specific behaviours of this machine and enable machine operators to teach an additional algorithm layer, which anomalies are relevant for machine maintenance and failure.

IV. Conclusion

In this paper, a data-driven intelligent system has been developed for the non-invasive machine early anomaly detection to avoid machines' breakdown. The approach proposed for this system is totally unsupervised, depending only on the input data. The algorithm consists of different levels so that the input of each level of data analysis is the results of previous level. The cascaded structure of the proposed algorithm which was designed by taking advantage of hybrid time-frequency information, created a strong and robust clustering method that led to a high accuracy in pattern recognition. The recognized patterns were then used for anomaly detection relying on the deviation from the norms of patterns and their sequences. The proof of concept was evaluated using a real data gathered by the spintropic TMR-based sensor and the results shows high performance of the proposed algorithm. One advantage of this work relates to the extraction of useful information from a very small amount of data, which allows the algorithm to be performed on a huge amount of unseen data successfully. As a result, this paper provided an interesting approach for recognizing signal patterns and thereupon identifying anomalies.

Acknowledgement

INL acknowledges funding from the NORTE-06-3559-FSE-000102 program.

References

[1] F. C. Bauer, D. R. Muir, and G. Indiveri, "Real-time ultra-low power ECG anomaly detection using an event-driven neuromorphic processor," *IEEE TRANSACTIONS ON BIOMEDICAL CIRCUITS AND SYSTEMS*, vol. 13, no. 6, pp. 1575–1582, 2019.

[2] T. Schlegl, P. Seeböck, S. M. Waldstein, G. Langs, and U. Schmidt-Erfurth, "f-AnoGAN: Fast unsupervised anomaly detection with generative adversarial networks," *Medical Image Analysis*, pp. 30–44.

[3] Z. Liang, S. Oba, and S. Ishii, "An unsupervised EEG decoding system for human emotion recognition," *Neural Networks*, pp. 257–268.

[4] Y. Karadayi, M. N. Aydin, and A. S. Öğrenci, "Unsupervised anomaly detection in multivariate spatio-temporal data using deep learning: Early detection of covid-19 outbreak in Italy," *IEEE Access*, vol. 8, pp. 164 155–164 177, 2020.

[5] C. I. Christodoulou and C. S. Pattichis, "Unsupervided pattern recognition for the classification of EMG signals," *IEEE Transactions on Biomedical Engineering*, vol. 46, no. 2, pp. 169–178, 1999.

[6] L. Meneghetti, M. Terzi, S. Del Favero, G. A. Susto, and C. Cobelli, "Data-Driven Anomaly Recognition for Unsupervised Model-Free Fault Detection in Artificial Pancreas," *IEEE Transactions on Control Systems Technology*, vol. 28, no. 1, pp. 33–47, 2020.

[7] X. Fang, W. Guo, Q. Li, J. Zhu, Z. Chen, J. Yu, B. Zhou, and H. Yang, "Sewer Pipeline Fault Identification Using Anomaly Detection Algorithms on Video Sequences," *IEEE Access*, vol. 8, pp. 39 574–39 586, 2020.

[8] Y. Zhao, Q. Liu, D. Li, D. Kang, Q. Lv, and L. Shang, "Hierarchical anomaly detection and multimodal classification in large-scale photovoltaic systems," *IEEE Transactions on Sustainable Energy*, vol. 10, no. 3, pp. 1351–1361, 2019.

[9] Y. Koizumi, Y. Kawaguchi, K. Imoto, T. Nakamura, Y. Nikaido, R. Tanabe, H. Purohit, K. Suefusa, T. Endo, M. Yasuda, and N. Harada, "Description and discussion on dcase2020 challenge task2: Unsupervised anomalous sound detection for machine condition monitoring," *arXiv*, 2020.

[10] C. Li, L. Guo, H. Gao, and Y. Li, "Similarity-Measured Isolation Forest: An Anomaly Detection Method for Machine Monitoring Data," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, 2021.

[11] F. Jia, Y. Lei, J. Lin, X. Zhou, and N. Lu, "Deep neural networks: A promising tool for fault characteristic mining and intelligent diagnosis of rotating machinery with massive data," *Mechanical Systems and Signal Processing*, vol. 72-73, pp. 303–315, 2016.

[12] N. G. Lo, J. M. Flaus, and O. Adrot, "Review of Machine Learning Approaches in Fault Diagnosis applied to IoT Systems," *2019 International Conference on Control, Automation and Diagnosis, ICCAD 2019 - Proceedings*, 2019.

[13] S. Nasiri, M. R. Khosravani, and K. Weinberg, "Fracture mechanics and mechanical fault detection by artificial intelligence methods: A review," *Engineering Failure Analysis*, vol. 81, no. January, pp. 270–293, 2017.

[14] M. Barakat, M. El Badaoui, and F. Guillet, "Hard competitive growing neural network for the diagnosis of small bearing faults," *Mechanical Systems and Signal Processing*, no. 1-2, pp. 276–292.

[15] R. Touati, M. Mignotte, and M. Dahmane, "Anomaly Feature Learning for Unsupervised Change Detection in Heterogeneous Images: A Deep Sparse Residual Model," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 13, pp. 588–600, 2020.

[16] R. Zhao, B. Du, L. Zhang, and L. Zhang, "Beyond background feature extraction: An anomaly detection algorithm inspired by slowly varying signal analysis," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 54, no. 3, pp. 1757–1774, 2016.

[17] N. Merrill and A. Eskandarian, "Modified Autoencoder Training and Scoring for Robust Unsupervised Anomaly Detection in Deep Learning," *IEEE Access*, vol. 8, pp. 101 824–101 833, 2020.

[18] J. Deng, X. Xu, Z. Zhang, S. Fruhholz, and B. Schuller, "Universum Autoencoder-Based Domain Adaptation for Speech Emotion Recognition," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 500–504, 2017.

[19] M. Kwak and S. B. Kim, "Unsupervised Abnormal Sensor Signal Detection with Channelwise Reconstruction Errors," *IEEE Access*, vol. 9, pp. 39 995–40 007, 2021.

[20] D. Guthrie, L. Guthrie, B. Allison, and Y. Wilks, "Unsupervised anomaly detection," *IJCAI International Joint Conference on Artificial Intelligence*, pp. 1624–1628, 2007.

[21] S. Ahmad, A. Lavin, S. Purdy, and Z. Agha, "Unsupervised real-time anomaly detection for streaming data," *Neurocomputing*, vol. 262, pp. 134–147, 2017.

[22] W. Wu, L. He, W. Lin, Y. Su, Y. Cui, C. Maple, and S. A. Jarvis, "Developing an Unsupervised Real-time Anomaly Detection Scheme for Time Series with Multi-seasonality," *IEEE Transactions on Knowledge and Data Engineering*, vol. 4347, no. c, pp. 1–1, 2020.

[23] Y. Guo, T. Ji, Q. Wang, L. Yu, G. Min, and P. Li, "Unsupervised Anomaly Detection in IoT Systems for Smart Cities," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2231–2242, 2020.

[24] L. Chen, S. Li, and X. Wang, "Quickest fault detection in photovoltaic systems," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 1835–1847, 2018.

[25] R. Platon, J. Martel, N. Woodruff, and T. Y. Chau, "Online fault detection in pv systems," *IEEE Transactions on Sustainable Energy*, vol. 6, no. 4, pp. 1200–1207, 2015.

[26] L. Meneghetti, M. Terzi, G. A. Susto, S. Del Favero, and C. Cobelli, "Fault Detection in Artificial Pancreas: A Model-Free approach," *Proceedings of the IEEE Conference on Decision and Control*, vol. 2018-December, no. Cdc, pp. 303–308, 2019.

[27] X. Chen, B. Li, R. Proietti, Z. Zhu, and S. J. Yoo, "Self-taught anomaly detection with hybrid unsupervised/supervised machine learning in optical networks," *Journal of Lightwave Technology*, vol. 37, no. 7, pp. 1742–1749, 2019.

[28] Y. Zhao, R. Ball, J. Mosesian, J. F. De Palma, and B. Lehman, "Graph-based semi-supervised learning for fault detection and classification in solar photovoltaic arrays," *IEEE Transactions on Power Electronics*, vol. 30, no. 5, pp. 2848–2858, 2015.

[29] T. Tanprasert, C. Saiprasert, and S. Thajchayapong, "Combining Unsupervised Anomaly Detection and Neural Networks for Driver Identification," *Journal of Advanced Transportation*, vol. 2017, 2017.

[30] T. Ergen, "Unsupervised Anomaly Detection with Spark - MapR," no. 8, pp. 1–15.

[31] R. E. Shiffler, "Maximum z scores and outliers," *The American Statistician*, vol. 42, no. 1, pp. 79–80, 1988.

[32] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," vol. 29, no. 2, 2000.

[33] J. Tang, Z. Chen, A. W.-c. Fu, and D. W. Cheung, "Enhancing effectiveness of outlier detections for low density patterns," in *Advances in Knowledge Discovery and Data Mining*, M.-S. Chen, P. S. Yu, and B. Liu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2002, pp. 535–548.

[34] L. Tran, M. Y. Mun, and C. Shahabi, "Real-time distance-based outlier detection in data streams," *Proceedings of the VLDB Endowment*, vol. 14, no. 2, pp. 141–153, 2020.

[35] F. T. Liu, K. M. Ting, and Z. H. Zhou, "Isolation forest," *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 413–422, 2008.

[36] E. Paz, S. Serrano-Guisan, R. Ferreira, and P. P. Freitas, "Room temperature direct detection of low frequency magnetic fields in the 100 pT/Hz0.5 range using large arrays of magnetic tunnel junctions," *Journal of Applied Physics*, vol. 115, no. 17, pp. 2012–2015, 2014.

[37] E. Paz, R. Ferreira, and P. P. Freitas, "Linearization of Magnetic Sensors with a Weakly Pinned Free-Layer MTJ Stack Using a Three-Step Annealing Process," *IEEE Transactions on Magnetics*, vol. 52, no. 7, 2016.

[38] A. Tanwear, X. Liang, Y. Liu, A. Vuckovic, R. Ghannam, T. Bohnert, E. Paz, P. P. Freitas, R. Ferreira, and H. Heidari, "Spintronic Sensors Based on Magnetic Tunnel Junctions for Wireless Eye Movement Gesture Control," *IEEE Transactions on Biomedical Circuits and Systems*, vol. 14, no. 6, pp. 1299–1310, 2020.

[39] A. Tanwear, H. Heidari, E. Paz, T. Böhnert, and R. Ferreira, "Eyelid gesture control using wearable tunnelling magnetoresistance sensors," in *2020 27th IEEE International Conference on Electronics, Circuits and Systems (ICECS)*, 2020, pp. 1–4.

[40] S. Zuo, K. Nazarpour, T. Böhnert, E. Paz, P. Freitas, R. Ferreira, and H. Heidari, "Integrated pico-tesla resolution magnetoresistive sensors for miniaturised magnetomyography," in *2020 42nd Annual International Conference of the IEEE Engineering in Medicine Biology Society (EMBC)*, 2020, pp. 3415–3419.

[41] C. Nebauer, "Evaluation of convolutional neural networks for visual recognition," *IEEE Transactions on Neural Networks*, vol. 9, no. 4, pp. 685–696, 1998.

[42] E. Iranmehr, B. Vosughi Vahdat, and M. M. Faraji, "Fpga implementation of character recognition using spiking neural network," in *7th iranian conference on electrical and electronic engineering (ICEEE2015)*, 2015.

[43] Y. Chherawala, P. P. Roy, and M. Cheriet, "Feature set evaluation for offline handwriting recognition systems: Application to the recurrent neural network model," *IEEE Transactions on Cybernetics*, vol. 46, no. 12, pp. 2825–2836, 2016.

[44] B. Schrauwen, D. Verstraeten, and J. Van Campenhout, "An overview of reservoir computing: theory, applications and implementations," *Proceedings of the 15th European Symposium on Artificial Neural Networks*, no. April, pp. 471–82, 2007.

[45] E. Iranmehr, S. B. Shouraki, M. M. Faraji, N. Bagheri, and B. Linares-Barranco, "Bio-inspired evolutionary model of spiking neural networks in ionic liquid space," *Frontiers in Neuroscience*, vol. 13, p. 1085, 2019.

[46] E. Iranmehr, S. Baghri Shouraki, and M. M. Faraji, "Ils-based reservoir computing for handwritten digits recognition," in *2020 8th Iranian Joint Congress on Fuzzy and intelligent Systems (CFIS)*, 2020, pp. 1–6.

[47] I. Jolliffe, "Springer series in statistics," in *Principal Component Analysis*. New York: Springer, New York, NY, 2002, pp. XXX, 488.

[48] V. Estivill-Castro, "Why so many clustering algorithms – a position paper," *ACM SIGKDD Explorations Newsletter*, vol. 4, 06 2002.

[49] H.-H. Bock, *Clustering Methods: A History of k-Means Algorithms*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2007, pp. 161–172.

[50] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "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*, ser. KDD'96. AAAI Press, 1996, p. 226–231.

Ensieh Iranmehr is a Research Engineer of the Technology Engineering Group at International Iberian Nanotechnology Laboratory (INL). Her focus is on using machine learning techniques to overcome the challenges of real problems. She received her PhD in Digital Systems from the electrical engineering department of Sharif University of Technology in 2020. Her works revolve around artificial neural network, machine learning, neuromorphic engineering and digital systems. For her PhD project, she has proposed a new neuromorphic structure of spiking neural network inspired by biological studies called ILS-based Reservoir Network. She also has MSc in Digital Electronic Engineering from the electrical engineering department of Amirkabir University of Technology. In her MSc project, she designed a system for tracking vehicle using image processing, and artificial neural networks. Ensieh involved in several academic projects considering artificial intelligence techniques in different applications and implement some of these algorithms on FPGA and GPU for accelerating the process and made them suitable for real-time tasks. She currently has several peer-reviewed publications.

Ricardo Ferreira received the Ph.D. in physics engineering from Instituto Superior Técnico (IST) in 2008 upon his work on ion beam deposited magnetic tunnel junctions targeting hard disk drive read heads, nonvolatile memories, and magnetic field sensor applications. He is a Researcher with the International Iberian Nanotechnology Laboratory (INL), where he is the Principal Investigator of the Spintronics Research group and the Coordinator of the ICT research area. INESC Microsistemas e Nanotecnologias was the host institution during his Ph.D. and also the lab where he conducted his research as a Post-Doctoral Fellow until joining INL. At INL, the work has been focused on the production of MTJs using MgO barriers and the development of two high-yield and high-uniformity fabrication processes on 200 mm wafers. The current research goals include production of pT magnetic field sensors, the production of devices that explore the spin transfer effect (STT nano-oscillators, MRAM cells), the incorporation of perpendicular magnetization materials on MTJs, and the integration of MTJs with MEMS and CMOS devices. He is the co-author of 80+ peer-reviewed publications.

Tim Böhnert finished his study of physics at the University of Hamburg, he was hired as a scientific assistant by the cluster of excellence “Nanospintronics”. He received his PhD degree from the University of Hamburg (Germany). He joined INL within the Spintronics Research Group to work for SpinCal project (EXL04). In 2018, he took a position as Staff Researcher at INL with focus on MTJ based magnetic field sensors. He succeeded at magnetic imaging of microstructures using an array of 65000 integrated MTJs on a 4x4mm CMOS chip and fabricated an embedded magnetic field sensor demonstrator, which was successfully applied in multiple industry production facilities for smart sensing and predicative maintenance. He contributes in FET Open SpinAge project and he develops weighted spin torque nano-oscillators for neuromorphic computing systems. He currently has 38 peer-reviewed articles.

Paulo P. Freitas was born in Lisbon, Portugal, in 1958. He received the B.S. and M.S. degrees in physics from the University of Porto, Porto, Portugal, in 1981, and the Ph.D. degree in physics from Carnegie Mellon University, Pittsburgh, PA, USA, in 1986.,After a postdoctoral appointment at the IBM T.J. Watson Research Center, Yorktown Heights, NY, USA, in 1988, he joined INESC, Porto, where he created the Solid State Technology Group. He is a Full Professor of physics with the Instituto Superior Tecnico, University of Lisbon, Lisbon. Since 2008, he has been a Deputy Director General with the International Iberian Nanotechnology Laboratory, Braga, Portugal. He has authored 450 research articles, advised 20 Ph.D. students, and participated in a startup using INESC MN biochip technology and holds several patents. His current research interests include spintronics and applications in sensing, memory, and biomedical applications.,Dr. Freitas was a recipient of the Gulbenkian Foundation Nanotechnology Award in 2004, the Portuguese Foundation for Science and Technology Excellence Award in 2006, and the Scientific Award of the Technical University of Lisbon in 2008. He was one of the team members awarded as second Finalist for the EU Descartes Prize for Research in 2004. Within the IEEE Magnetics Society, he has been a senior member since 2016. He was elected for the IEEE Mag Soc Procom in 2007 and a Distinguished Lecturer in 2008. Since 2016, he has been one effective member of the Portuguese Academy of Sciences.