Efficient Anomaly Detection in a Laser-Surface Heat-Treatment Process via Laser-Spot Tracking

David Atienza, Concha Bielza, *Member, IEEE*, Javier Diaz-Rozo,
and Pedro Larrañaga, *Member, IEEE*

***Abstract*—This article proposes a novel algorithm to detect anomalies during a laser-surface heat-treatment process recorded using a high-speed thermal camera. Our approach detects anomalies by tracking the movement of the laser spot along the surface of a production artifact. No previous knowledge of the anomalies is assumed. The model attempts to learn the behavior of a normal process so that anomalous video frames can be detected when the test data differ significantly from the learned model. The model is trained using real process data provided by a company operating in the automotive sector. First, laser-spot movements are obtained by computing a sequence of their positions. Second, the expected movement of the laser spot is accurately determined from the nonanomalous data by using a model based on training multiple kernel density estimation models. Finally, an anomaly score is introduced to classify a workpiece as normal or anomalous using the trained model. Furthermore, our methodology is computationally efficient when compared to other techniques. Additionally, our objective is to perform in-process classification, that is, to perform the classification within a short period after the laser-surface heat-treatment process ends.**

***Index Terms*—Anomaly detection, in-process classification, kernel density estimation (KDE), laser reliability, machine learning, tracking.**

I. Introduction

RECENTLY, a full range of industries has made efforts to produce better products more reliably using modern technologies. In the near future, they are expected to undergo major transformations in this regard to improve productivity. This transformation has been termed “Industry 4.0” [1] and is closely associated with the industrial Internet of things, both of which encourage the fusion of advanced sensors using machine-learning techniques. One of the most useful tasks performed by machine learning in industry is anomaly detection [2]–[10]. Anomaly detection [11] is the task of identifying anomalous patterns in data. Usually, anomalous patterns are related to a fault in the system under study. In various industries, anomalies should ideally be detected instantaneously to allow for immediate corrective actions.

In this article, we detect anomalies with a novel laser-surface heat-treatment process recorded using a high-speed thermal camera at 1000 frames per second (fps). The laser heat-treatment process improves the mechanical surface properties of cylindrical steel workpieces (see Section II for further details). In the premise of this article, we have limited knowledge about the possible anomalies that may occur. In fact, only a single anomaly has been recorded out of thousands of completed processes. Thus, our model describes the normal behavior of a system. Then, it computes an anomaly score for each workpiece. To take advantage of the detected anomalies, the classification should be completed in-process. That is, a workpiece must be classified within a short period (5 s) after the end of the process. For this reason, we need a model that provides a fast classification. This is a challenging objective because of the numerous video frames required to view a rapidly changing dynamic process.

We approach this problem using a novel methodology that employs multiple kernel density estimation (KDE) [12] models, to characterize the movement of the system’s laser along the workpiece surface. Our methodology (see Section IV) is intended to consume data of well-defined sequential behaviors with few spatial variations. Furthermore, we propose a preprocessing step to obtain a sequence of positions from a video (see Section IV-A). We then review some of the studies most related to this research (see Section III).

We train our model using real data provided by a company in the automotive sector. The application of anomaly detection procedures has a significant impact on the quality and cost-effectiveness of production in this industry. If a fault is not detected, it can lead to a loss of reputation of a company, high repair costs, and sometimes, human losses. Methods of anomaly detection based on the destruction of a sample of the entire production have two problems: the cost of the pieces destroyed and the nonexhaustiveness of the analysis. Our approach (see Section V) offers a promising capability to detect subtle differences in system behavior to enable the analysis of each working piece. Moreover, our system is compared with other algorithms to validate its effectiveness.

Manuscript received October 22, 2018; revised September 10, 2019, December 20, 2019, April 23, 2020, and June 2, 2020; accepted September 5, 2020. Date of publication September 18, 2020; date of current version February 16, 2021. Recommended by Technical Editor F. Janabi-Sharifi and Senior Member I.-M. Chen. This work was supported in part by the Spanish Ministry of Education, Culture and Sport under Grant FPU16/00921, and in part by the Spanish Ministry of Science and Innovation through the PID2019-109247GB-I00 and RTC2019-006871-7 Projects. *(Corresponding author: David Atienza.)*

David Atienza, Concha Bielza, and Pedro Larrañaga are with the Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid, 28660 Boadilla del Monte, Spain (e-mail: datienza@fi.upm.es; mcbielza@fi.upm.es; pedro.larranaga@fi.upm.es).

Javier Diaz-Rozo is with the Aingura IIoT, 00000 Elgoibar, Spain (e-mail: jdiaz@ainguraiiot.com).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMECH.2020.3024613

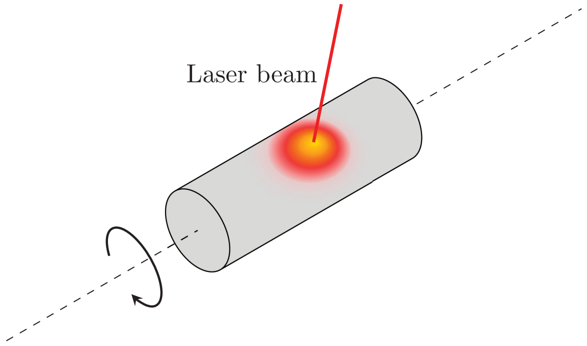


Fig. 1. Laser-surface heat-treatment process.

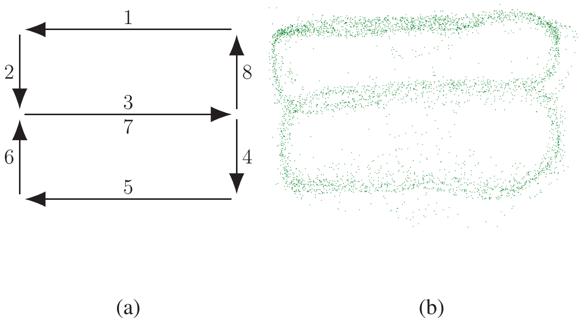


Fig. 2. Laser spot pattern. (a) Expected. (b) Computed laser spot positions.

II. Laser-Surface Heat-Treatment Process

In this section, we describe the laser-surface heat-treatment (laser-heating) process, which is often applied to cylindrical workpieces made of steel to improve their surface mechanical properties. The laser-heating process uses a laser beam that heats the surface of a cylindrical workpiece as it rotates around its axis (see Fig. 1). The laser power source is 8 kW. Thus, it is only safe to operate in a controlled environment. While applying the laser-heating process, the laser beam heats a region (i.e., the laser spot), which is always much smaller than the workpiece surface. Furthermore, high-power (> 6 kW) laser sources have laser spots with Gaussian or top-hat energy distributions. For these reasons, a static laser spot would not provide the energy distribution required to treat the entire surface of the workpiece. Thus, to provide the correct energy deposition and distribution over the surface, the process moves the laser spot following a known trajectory, as represented in Fig. 2(a). The trajectory pattern is executed at a frequency of 100 Hz to guarantee the desired energy magnitudes. However, the expected pattern is modified whenever the laser beam is required, to avoid an obstacle (see Fig. 3). To control the pattern, the laser process uses a scanner system with galvanometric mirrors to accurately position the laser spot over the surface. The scanner is a closed-loop control system that sends a specific pattern to the galvanometric mirrors and receives feedback from the encoders where the laser-spot positioning is adjusted. The need to use a moving laser spot is justified because the average workpiece weight is around 12 kg. Thus, it would not be possible to move any mechanical/physical part with the necessary frequency. The laser spot has no such restrictions. All details of the laser heating process can be found in [13].

Anomalies in the laser-heating process can originate from different components: the scanner system, the workpiece positioning system, and the laser source. This article focuses on anomalies generated by the scanner system because it is the most critical element of the machine. If it fails, it can create problems at the product and process level, and can stop the entire machine and production line. To define the scanner as the most critical element, the machine manufacturer carried out a failure model and effects analysis that returned the highest risk level. However, we emphasize that because it is a real production system, failures are rare, as will be shown in Section V-A when the real data are described. The scanner control system can produce anomalies in which positioning data are lost because of communication failures between the electronics and the mirrors, as well as incorrect movement/pattern creation. Additionally, anomalies can be generated through mechanical degradation in the scanner (e.g., changes in the rigidity of the physical linkages between galvanometers and mirrors or galvanometers seizures). All these disturbances strongly affect the scanner behavior and its positioning precision.

Given the novelty of the laser-heating process, prior to this article, there were no automatic procedures for detecting anomalies in this particular laser-treatment process. However, after reviewing the anomaly detection procedures for other laser-treatment processes (see Section III-B), we found that the most common technique is temperature reading using pyrometers and thermal cameras. Thus, the laser-heating process is recorded with a high-speed thermal camera (Tachyon μCore) at a resolution of 32×32 pixels and a frame rate of 1000 fps. Each pixel can take a value ranging from 0 to 1023, proportional to the surface temperature of the workpiece. Fig. 4 shows a sample frame from one of these videos. A schematic indicating the physical arrangement of the laser beam (red line) and the thermal camera is shown in Fig. 5.

III. Related Work

Before introducing the proposed algorithm, we review KDE and other related work of anomaly detection in the laser-treatment processes.

A. Kernel Density Estimation

KDE is a nonparametric method used to estimate a probability distribution from data. Unlike parametric models, KDE makes no assumptions about the underlying distribution.

The kernel density estimator, $\hat{p}$, of a variable X is usually expressed as

$$p(X = x) = \frac{1}{Nh} \sum_{i=1}^{N} K\left(\frac{x - x_i}{h}\right) = \frac{1}{N} \sum_{i=1}^{N} K_h\left(x - x_i\right) \tag{1}$$

where N is the number of instances in the data and x_i is the ith data point. K is a kernel function whose integral over the entire space is equal to 1. The smoothness of the estimation is controlled by parameter h, which is called the bandwidth. Small values of h produce wiggly density estimations, whereas large

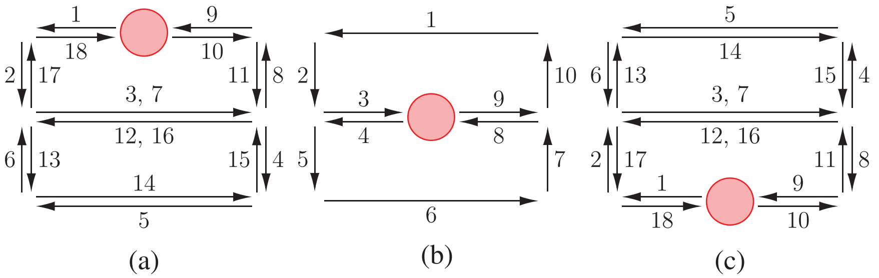


Fig. 3. Modified patterns when the laser beam avoids an obstacle at different positions. (a) Top. (b) Middle. (c) Bottom.

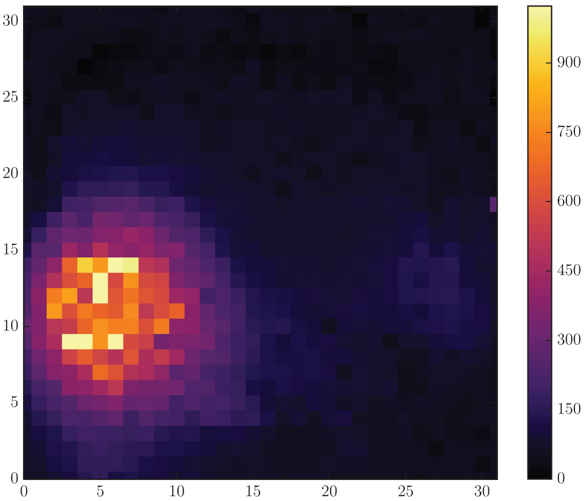


Fig. 4. Video frame recorded during the laser-surface heat-treatment process.

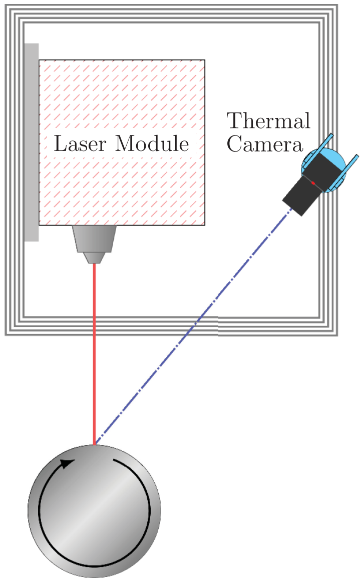


Fig. 5. Diagram of the physical arrangement of the laser-heating process components.

values produce smoother ones. The selection of the bandwidth parameter is an important topic of research in KDE [14]. We note that KDE is equivalent to place a distribution with the form of K_h on each data instance, x_i, and then obtaining the mean density. $K_h(x) = \frac{1}{h}K(\frac{x}{h})$ is a useful notation for showing that KDE is the mean of the contribution of each data instance.

KDE can be extended to the multivariate case if we use a multivariate kernel. However, in the multivariate case, the parameter h is replaced by a positive definite symmetric matrix $\mathbf{H}$, called the bandwidth matrix. In this article, we use a multivariate Gaussian kernel that is defined as

$$K_{\mathbf{H}}(\mathbf{x}) = \frac{1}{(2\pi)^{d/2}|\mathbf{H}|^{-1/2}} \exp\left(-\frac{1}{2}\mathbf{x}^T\mathbf{H}^{-1}\mathbf{x}\right) \quad (2)$$

where $|\mathbf{H}|$ is the determinant of the matrix $\mathbf{H}$. We refer the reader to several studies describing KDE for further details [15]–[17].

B. Laser-Treatment Anomaly Detection

Our proposal is based on the tracking of the laser movement. The most closely related studies apply to anomaly detection during a laser-welding process using a laser-tracking technique. A model based on the Sage–Husa adaptive Kalman filters using an embedded Elman neural network was proposed in [18] to minimize the tracking error of the weld-seam position. The laser movement traced a straight line, and therefore, the tracking was 1-D, as opposed to the complex pattern addressed in this article. Another study [19] aimed to find anomalies by analyzing the trajectories of the sputters produced during the laser-welding process. These sputters were visualized in a video as hot objects moving across the field of view. A Kalman filter was used for leveraging a state transition matrix that assumed a constant accelerated motion.

Moreover, there are other approaches unrelated to tracking for video analysis, such as hidden Markov models [20], where the likelihood ratio of a model trained with nonanomalous sequences to a model extended with anomalous sequences is used as the classification criterion. Because the number of pixels can be quite large, dimensionality reduction techniques, such as principal component analysis, have been proposed [21], [22] as a prior step to classification. In [22], the same camera as that used in this research was employed, but at a higher frame rate, 10 000 fps, for the classification of defects in a laser-welding process.

Our proposal combines the tracking of laser movements while training a statistical model that computes the likelihood of the test data to detect anomalies. As opposed to our research, none of the previous related studies had any classification time requirement.

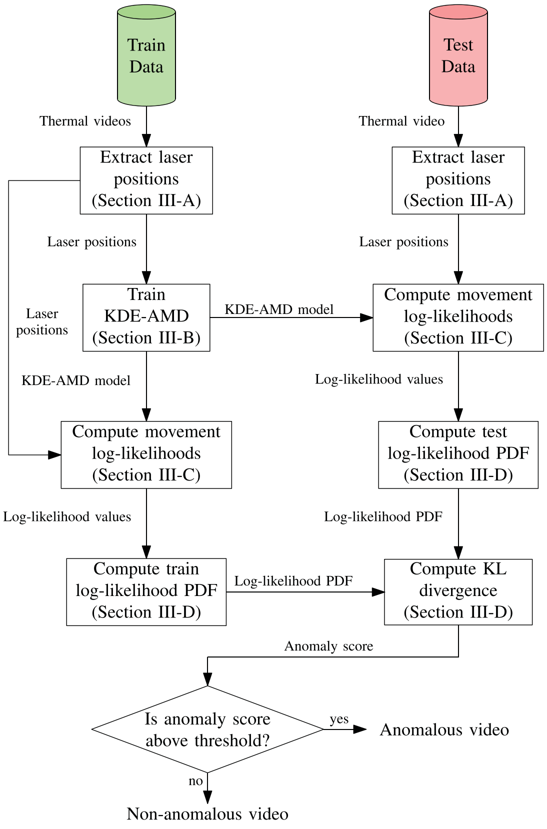

Train Data
Test Data
Thermal videos
Thermal video
Extract laser positions (Section III-A)
Extract laser positions (Section III-A)
Laser positions
Laser positions
Laser positions
Train KDE-AMD (Section III-B)
KDE-AMD model
Compute movement log-likelihoods (Section III-C)
KDE-AMD model
Log-likelihood values
Compute movement log-likelihoods (Section III-C)
Compute test log-likelihood PDF (Section III-D)
Log-likelihood values
Log-likelihood PDF
Compute train log-likelihood PDF (Section III-D)
Log-likelihood PDF
Compute KL divergence (Section III-D)
Anomaly score
Is anomaly score above threshold?
yes
Anomalous video
no
Non-anomalous video

Fig. 6. KDE–AMD preprocessing, training, and classification flowchart.

IV. KDE-Anomaly Movement Detection (AMD)

We detect the anomalies in the laser heating process by identifying unusual laser-spot movements. For this reason and because the algorithm relies on a combination of KDE models, we call our approach "KDE–AMD." If the laser-spot movement deviates significantly from the expected pattern shape or the expected laser speed, the workpiece can be considered anomalous. Fig. 6 presents the flowchart of the proposed methodology.

A model of the process evolution is learned using the training data, which capture the normal behavior of the process. In our case, the model considers the spatio-temporal characteristics of the laser-heating process by learning the expected movements of the laser in different spatial regions and temporal moments. Thus, KDE–AMD can be considered an ensemble of KDE models taking advantage of the temporal/spatial locality of the laser movements to train a faster and more accurate model. This is because only the most similar (in terms of time and position) training laser movements are considered to classify each movement. The divergence between the outcomes of the model learned using the training and test data is used as the discriminant to detect an anomaly. The details of each procedure are presented in the remainder of this section.

A. Extraction of the Laser-Spot Positions

Our preprocessing step entails the extraction of the laser-spot positions from the recorded videos. In Section II, we mentioned that the laser spot follows a known pattern. The closer the surface region is to the laser beam, the larger is the expected temperature increase. Thus, we can detect the laser-spot positions by computing the regions having higher temperatures.

Let $\mathbf{O} = (\mathbf{O}_1, \ldots, \mathbf{O}_N)$ denote an original video from the laser-heating process, where $\mathbf{O}_i$ represents the ith frame.

First, we create a video showing the positive temperature variation between each pair of two consecutive frames. This video is called the difference video

$$\mathbf{Z} = (\mathbf{Z}_1, \ldots, \mathbf{Z}_{N-1})$$

$$\text{such that } z^i_{j,k} = \begin{cases} o^{i+1}_{j,k} - o^i_{j,k}, & \text{for } o^{i+1}_{j,k} - o^i_{j,k} > 0 \\ 0, & \text{for } o^{i+1}_{j,k} - o^i_{j,k} \leq 0 \end{cases} \quad (3)$$

$o^i_{j,k}$ is the pixel intensity value in the jth row and kth column of $\mathbf{O}_i$, and $z^i_{j,k}$ denotes the difference in pixel intensity value in the jth row and kth column between frames $\mathbf{O}_{i+1}$ and $\mathbf{O}_i$. Thus, the frames $\mathbf{Z}_i$ produce the difference videos. Note that the pixels having negative variation are set as zero in the difference video, which contains one less frame than does the original video. Using $\mathbf{Z}_i$, the laser-spot position for that frame can be computed as a weighted mean, as follows:

$$\mathrm{s}^{\mathrm{row}}_i = \frac{1}{\sum_{z^i_{j,k} \in \mathbf{Z}_i} z^i_{j,k}} \sum_{z^i_{j,k} \in \mathbf{Z}_i} j z^i_{j,k}$$

$$\mathrm{s}^{\mathrm{col}}_i = \frac{1}{\sum_{z^i_{j,k} \in \mathbf{Z}_i} z^i_{j,k}} \sum_{z^i_{j,k} \in \mathbf{Z}_i} k z^i_{j,k}. \quad (4)$$

$\mathrm{s}^{\mathrm{row}}_i$ and $\mathrm{s}^{\mathrm{col}}_i$ denote the row and column, respectively, of the laser-spot position in frame i. With this computation, the calculated laser-spot position will be near the pixels having higher temperature values. A similar approach was used in [18]. Because $\mathrm{s}^{\mathrm{row}}_i$ and $\mathrm{s}^{\mathrm{col}}_i$ are computed with a weighted mean, they are continuous values. This implies that the center of the laser spot does not necessarily lie in a pixel coordinate; it can be found between pixels.

B. Training

KDE–AMD must be trained using data representing the normal processes. For numerous anomaly detection applications, only a few instances of anomalous data are often available. Therefore, our methodology builds a model of the normal system behavior instead of an anomalously driven classifier. The anomalous processes are expected to show a different behavior than the trained model. In our use case, the laser-spot positions described in Section IV-A are used to train the KDE–AMD model.

Let $\mathbf{S} = (\mathbf{s}_1, \ldots, \mathbf{s}_{N-1})$ denote the calculated laser-spot positions, where $\mathbf{s}_i \in [0, 31] \times [0, 31]$. $\mathbf{s}_i$ is the laser-spot position for frame i, whose value comprises the computed row and column, $(\mathrm{s}^{\mathrm{row}}_i, \mathrm{s}^{\mathrm{col}}_i)$, in accordance with (4).

We define a movement $\mathbf{M} = (\mathbf{s}_O, \mathbf{s}_D)$ as the transition between two consecutive frames. Thus, $\mathbf{s}_O$ is the original position of the movement and $\mathbf{s}_D$ is the destination position, such that $D = O + 1$. A video has a set of $N - 2$ movements, because there are $N - 1$ laser-spot positions, and each of the two consecutive laser spot positions are grouped in a single movement. The KDE–AMD model can be trained using the movements of multiple videos. Therefore, from the training data, we have a set

$$\mathbf{E} = \{\mathbf{M}_1, \ldots, \mathbf{M}_L\} \tag{5}$$

containing L different movements that are obtained from different videos. Because all videos may have different lengths, L may not be equal to $(N-2)\#\text{videos}$ (where $\#\text{videos}$ denotes the number of videos). Note that, in the set $\mathbf{E}$, the destination position of the movement, $\mathbf{M}_i$, can differ from the origin position of the movement, $\mathbf{M}_{i+1}$, because $\mathbf{M}_i$ and $\mathbf{M}_{i+1}$ can be obtained from different videos.

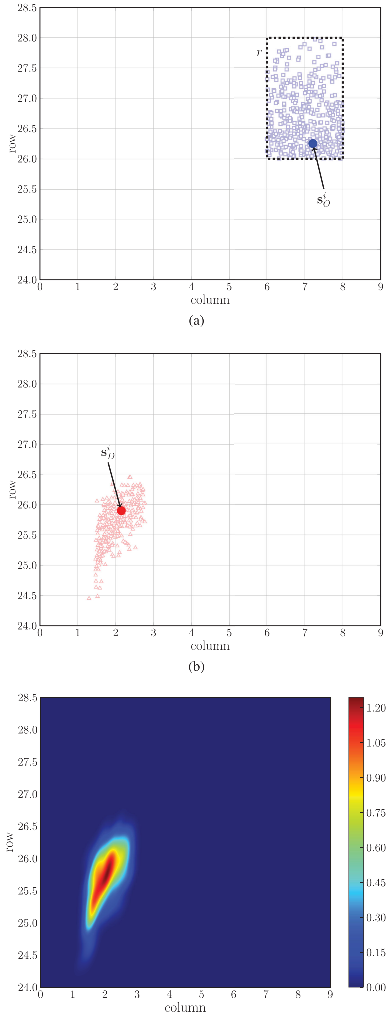
The KDE–AMD algorithm assumes that the destination positions of two movements should be close if their origin positions are close. We model the expected destination position of a movement using bivariate KDE models to model the two axes of the laser-spot position. Multiple KDE models are generated to consider the spatio-temporal characteristics of each movement. The KDE models are generated at the time of training to expedite the classification phase. First, we distribute the set of movements, $\mathbf{E}$, into W nonoverlapping temporal windows. The number of windows, W, is determined from the number of obstacles that we need to avoid (see Fig. 3), i.e., $W = 1 + 2\#\text{obstacles}$. For example, if there is a single obstacle, then there would be three different temporal frames: preobstacle, obstacle, and postobstacle. The start/end frame for each temporal moment is known in advance, because the laser-heating process is controlled by a computer-aided design/computer-aided manufacturing (CAD/CAM) system that ascertains the position of the obstacles. The aim of the temporal division is to learn different KDE models when the expected pattern is different (see Fig. 3), because both the recorded thermal footprint and computed laser positions will be quite different. The number of obstacles can be different for each type of workpiece. Details about the number of obstacles for our use case are given in Section V. This type of temporal division can be avoided if the laser-heating process were to not exhibit different patterns at different temporal moments.

We then divide each temporal window into R rows and C columns to consider the spatial information. The result of these divisions is a 3-D matrix of $W \times R \times C$ regions. R and C are parameters of the algorithm that determine the number of regions, r, in the 3-D matrix. Uniform partitioning is used to attach the same importance to all spatial regions. As mentioned, the laser-spot positions are continuous values, and can be found between pixels. Therefore, any integer value can be used for R and C, because we can create subpixel size partitions. For example, if we select $R = C = 64$, the spatial regions would be of size 0.5 × 0.5 pixels.

For each region, $r = (t, j, k)$ in the tth temporal window, and the jth row and kth column of the 3-D matrix, we find a set of movements whose origin position is inside region r

$$\mathbf{E}_r = \{\mathbf{M}_i = (\mathbf{s}_O^i, \mathbf{s}_D^i) | \mathbf{s}_O^i \in r\}, \ i = 1, \ldots, L \tag{6}$$

where $\mathbf{s}_O^i$ denotes the origin position of the ith movement, $\mathbf{M}_i$. Note that in the set $\mathbf{E}_r$, not all movements are necessarily contiguous. That is, the destination position of $\mathbf{M}_i$ is not the original position of $\mathbf{M}_{i+1}$. This is because the set $\mathbf{E}_r$ only contains movements having an origin position in region r. Usually, the laser-spot position changes its region between consecutive frames. Thus, contiguity is lost. Additionally, the movements can originate from different videos. Thus, there is no contiguity for movements from different videos. A bivariate KDE model is assigned to region r, trained with the set of destination positions of the movements in $\mathbf{E}_r$ as



(a)

(b)

(c)

Fig. 7. Example application of the methodology. (a) Origin positions in region r. (b) Their corresponding destinations. (c) PDF $\hat{p}$ in (8) assigned to region r.

$$\mathbf{D}_r = \{\mathbf{s}_D^i | \mathbf{M}_i = (\mathbf{s}_O^i, \mathbf{s}_D^i) \in \mathbf{E}_r\}, \ i = 1, \ldots, L. \tag{7}$$

When $\mathbf{D}_r = \emptyset$, r will not have an assigned KDE model, because there is no information on the possible destination positions of the laser spot. In fact, we can try to generate more informative KDE models by ignoring the KDE models trained with fewer than λ movements. If we use a larger λ, we expect more regions without an assigned KDE model.

As an example of the application of the methodology, we show the training process for a given region r, in Fig. 7. All origin positions inside region r are represented by blue squares in Fig. 7(a), and their respective movement destination positions are depicted as red triangles in Fig. 7(b). The red triangles are the training set for the KDE model assigned to region r. As illustrated in Fig. 7(c), the probability density function (PDF) of

Algorithm 1: KDE–AMD Training Procedure.

```
Input: Set of movements $\mathbf{E}$ (see (5)), number of windows
  $W$, number of rows $R$, number of columns $C$, minimum
  number of positions $\lambda$.
Output: Trained KDE–AMD model.
 1:  KDE–AMD $\leftarrow$ Create a 3D matrix of $W \times R \times C$
     regions.
     // Computes $\mathbf{D}_r$ (see (7)), for all regions
 2:  for each movement $(\mathbf{s}_O, \mathbf{s}_D)$ in E do
 3:    Compute region $r = (t, j, k)$ using $\mathbf{s}_O$ where:
 4:        $t \leftarrow$ Temporal index of the frame $O$ in the video.
 5:        $j \leftarrow$ Row index computed using $\mathbf{s}_O^{\text{row}}$.
 6:        $k \leftarrow$ Column index computed using $\mathbf{s}_O^{\text{col}}$.
 7:    $\mathbf{D}_r \leftarrow \mathbf{D}_r \cup \mathbf{s}_D$.
 8:  endfor
     // Train the bivariate KDE models
 9:  for each region $r = (t, j, k)$ in KDE–AMD do
10:    if $|\mathbf{D}_r| \geq \lambda$then
11:        Train a KDE model with the positions included
           in $\mathbf{D}_r$ and save it in KDE–AMD$[t, j, k]$.
12:    else
13:        Discard the positions saved in $\mathbf{D}_r$.
14:    end if
15:  end for
16:  return KDE–AMD
```

the KDE model shows higher densities in a small region where the destination positions are located. Therefore, the KDE model for region r can be considered an estimation of the destination position PDF for the movements with an origin position inside region r. For a test movement, $\mathbf{M}_t = (\mathbf{s}_O^t, \mathbf{s}_D^t)$, where $\mathbf{s}_O^t \in r$, the PDF for the destination position, $\mathbf{s}_D^t$, is

$$p(\mathbf{s}_D^t) = \frac{1}{|\mathbf{D}_r| 2\pi |\mathbf{H}|^{1/2}} \sum_{\mathbf{t} \in \mathbf{D}_r} \exp\left(-\frac{1}{2}(\mathbf{s}_D^t - \mathbf{t})^T \mathbf{H}^{-1} (\mathbf{s}_D^t - \mathbf{t})\right) \tag{8}$$

where $\mathbf{t}$ denotes the training positions in set $\mathbf{D}_r$, $\mathbf{H}$ is the bandwidth matrix obtained using Scott's rule [16], and $|\mathbf{D}_r|$ is the cardinality of the set $\mathbf{D}_r$. The training procedure of KDE–AMD is summarized in Algorithm 1. In lines 3–6, the needed temporal and spatial indices are computed to detect the region r for each movement. To compute them, we need to know the limits of the temporal moments and spatial regions. As indicated, the temporal window limits are known in advance, and the spatial limits are known when R and C are fixed. When the loop in lines 2–8 is completed, each $\mathbf{D}_r$ is known. In lines 9–15, we check whether there are enough positions in each region to train sufficiently accurate KDE models. If there are enough positions, a KDE model is trained and assigned to the 3-D matrix (line 11). Otherwise, the $\mathbf{D}_r$ positions are discarded (line 13).

C. Movement Likelihood

To classify a test video, the likelihood of each test movement is first computed at the time of classification. The likelihood of each movement can be obtained by evaluating the destination position in the correct KDE model of the trained 3-D matrix.

TABLE I
Parameters of Our Methodology

Parameter	Description	Value
λ	Minimum number of positions in each KDE model	50
R	Number of rows in the 3D matrix	35
C	Number of columns in the 3D matrix	35

First, we must extract the laser-spot positions of the test video, as described in Section IV-A. For each movement, $\mathbf{M}_i$, the appropriate KDE model in the 3-D matrix is selected using position $\mathbf{s}_O^i$. Two types of information are needed to access the correct KDE model: temporal (to select the correct temporal window) and spatial (row and column). Then, the likelihood of the movement is obtained by evaluating $\mathbf{s}_D^i$ in the KDE model with (8), as (8) defines the PDF of the destination position. A Laplace smoothing is applied to the evaluated result to deal with the 0 likelihood values. Thus, the smoothed likelihood value is $\frac{\hat{p}(\mathbf{s}_D^i)+\epsilon}{1+\epsilon}$, with ϵ being a small value (10^{-6} in our experiments). As noted in Section IV-B, some regions will not have any assigned KDE model. In that case, the likelihood of the movement is equal to $\frac{\epsilon}{1+\epsilon}$.

Fig. 7 also shows an example of the computation of the likelihood of a test movement. Because $\mathbf{s}_O^i$ [in Fig. 7(a)] is inside region r, we use the KDE model depicted in Fig. 7(c), trained with the positions in Fig. 7(b). In the example, the result of evaluating position $\mathbf{s}_D^i$ [in Fig. 7(b)] in the KDE model is a likelihood, $\hat{p}$, of 1.2 for the movement.

We note that each test movement is evaluated using only the destination positions in $\mathbf{D}_r$. Therefore, KDE–AMD can be fast, because it does not use all training movements. Additionally, the classification procedure can be easily parallelized, because the evaluation of different KDE models can be computed in parallel without further dependencies on each movement. Additionally, we do not need the complete video to begin evaluating the KDE models. For instance, the evaluation of the first movement only requires the first two frames. The next movement can be classified with one more frame. Thus, some movements can be classified before the laser-heating process is completed. These characteristics can reduce the classification time even more, so that the algorithm can achieve in-process classification.

Table I summarizes all parameters defined in the methodology and their values when applied to real data in the experiments (see Section V).

D. Anomaly Score

We compute an anomaly score using all of the obtained movement probabilities, as discussed in the previous section, to classify a given test workpiece. An anomaly score provides an assessment of an abnormality for a given workpiece. When the anomaly score is above a predefined threshold, the workpiece is classified as anomalous. Usually, the threshold is found by studying the receiver operating characteristic (ROC) curves [23], selecting the threshold that provides a convenient tradeoff between the true- and false-positive rate. This decision can be motivated by the cost of a misclassified anomalous/nonanomalous workpiece. The likelihood of movement $\mathbf{M}_i$ is denoted as p_i. For each likelihood value, its negative log-likelihood, $l_i = -\log p_i$, is calculated.

The set of l_i values characterizes a laser-heating video and is used to compute an anomaly score, which is based on the dissimilarity between the training and test log-likelihood values. A convenient method for comparing the training and test videos is to estimate the PDF of the log-likelihood values for the training and test data. Analyzing the PDFs allows a comparison of the log-likelihood information, even when there are more available training log-likelihood values than test ones. The PDFs of the log-likelihood values can be estimated using a univariate KDE model, $\hat{p}(l)$, over a set of log-likelihood values.

After the PDFs are computed, the anomaly score is calculated using the Kullback–Leibler divergence [24] between the training and test PDFs. This divergence is lower-bounded by zero. Thus, a nonanomalous workpiece is expected to have a near-zero value. The Kullback–Leibler divergence is neither upper-bounded nor normalized. Thus, the anomalous workpieces can have arbitrarily large anomaly scores.

The methodology is described in terms of our use case (i.e., a laser-heating process) for easiness. However, the methodology can be adapted to other problems by creating multiple regions over each variable of interest. In our use case, the variables of interest are the axes of the laser-spot position. For other use cases, the dimensionality of the generated matrix can be customized, depending on the number of variables and if the data exhibit different behavior at several temporal moments.

V. Experiments

The KDE–AMD algorithm is compared with other methods, including deep neural networks [25], Kalman filters [26], D-Markov machines [3], [27], and a global KDE algorithm, to demonstrate its effectiveness. Real (see Section V-A) and simulated (see Section V-B) data are used to test the algorithms.

The global KDE algorithm trains a single bivariate KDE model with a complete set of positions of the set of normal videos. Therefore, there is no spatial division in the complete set of positions as in KDE–AMD. Another single bivariate KDE model is generated for the test positions. Then, the Kullback–Leibler divergence was computed between both KDE models. Similar to the presented approach, temporal windows can be created for the global KDE model. In this case, we computed W different Kullback–Leibler divergences. The classification criterion was based on the weighted (by the length of each temporal window) average of the Kullback–Leibler divergences.

The D-Markov machine builds a Markov chain of order D, where the possible states are represented by different symbols. First, a symbolization (partition) process is applied to the data, where a symbol is assigned to each partition of the input space. Next, a transition matrix is computed such that the probability of the next symbol depends only on the previous D symbols. The classification criterion is based on a state probability vector calculated from the transition matrix. In [3], further details are provided on the use of D-Markov machines.

The Kalman-filter algorithm is based on the research reported in [19] and is described in Section III. In this article, we cannot obtain good estimates of acceleration because the process has no jerk (i.e., da/dt) control, allowing high magnitudes of acceleration change only limited by the physics of the mechanical design. As a result, the sampling rate of only 10 frames per cycle is too low to reliably reconstruct a signal with high jerk. Thus, the state vector of our Kalman filter comprises only the laser-spot position and speed.

The deep neural network is trained to predict the next laser-spot position from the current one. The network comprises seven hidden layers with 8, 128, 256, 512, 256, 128, and 8 neurons, with rectified linear-unit activation functions. Thus, the network has 331 058 parameters. This is the best structure found after trying different numbers of hidden layers and neurons per layer. The deep neural network is trained using Keras [28]. The anomaly score is computed as the sum of the Euclidian distance between the predicted and actual laser-spot positions. For this reason, smaller anomaly scores are expected for nonanomalous videos.

For each algorithm, the tests are conducted multiple times to select the best values for their parameters. In the KDE–AMD and D-Markov algorithms, different numbers of spatial divisions (R rows $\times$ C columns) are tested: 16×16, 20×20, 25×25, 30×30, 35×35, and 40×40. The best results (reported here) are obtained using 35×35 divisions for KDE–AMD. For the D-Markov machines, various symbolization techniques are used: equal width, equal frequency, and a variant of equal frequency in which the areas within the limits of the space without any position are not considered for the purpose of symbolization. Slightly better results are obtained for a 40×40 equal-width division, where D is set to 1 because a higher D is highly computationally expensive.

The D-Markov machine and KDE–AMD algorithms are tested using leave-one-out cross-validation. The Kalman filter does not need training data to build a model. The global KDE cannot be evaluated using the leave-one-out cross-validation method because of the associated computational burden. Therefore, we train the global KDE model with four videos and test it with the remaining 4219. The deep neural network is validated using two-fold cross validation to reduce the high computational cost of the training phase.

The code for all algorithms can be found in the repository https://github.com/davenza/KDE–AMD. Also, the data are available in [29] to reproduce all results in the experiments.

A. Real Data

To test the presented methodology, 4223 videos of real laser-surface heat-treatment processes are available, which are recorded on four different workstations and contain nine different workpiece types. Each workpiece type has a different size between 12 000 and 14 500 frames and several obstacles between 0 and 2. Altogether, there are 36 batches of videos, because there is a batch type for each workstation and workpiece type. In Table II, we describe the mean length in frames, its standard deviation, and the number of obstacles (denoted as Obs.) for each batch

TABLE II
BATCH-TYPE LENGTH (NUMBER OF FRAMES) AND NUMBER OF OBSTACLES

Type	Length (frames)	Obs.
1	14541.50 ± 22.41	0
2	14495.28 ± 37.47	2
3	14157.77 ± 33.04	0
4	14529.95 ± 22.05	2
5	14097.07 ± 36.20	0
6	12310.78 ± 28.60	1
7	12316.06 ± 34.07	1
8	12296.55 ± 19.19	1
9	12071.47 ± 21.71	1
10	14633.50 ± 92.56	0
11	14624.44 ± 61.77	2
12	14274.75 ± 67.28	0
13	14644.38 ± 70.09	2
14	14265.64 ± 65.26	0
15	12458.49 ± 65.27	1
16	12457.25 ± 55.25	1
17	12454.47 ± 43.15	1
18	12230.65 ± 55.81	1
19	14492.94 ± 43.95	0
20	14473.76 ± 36.67	2
21	14137.31 ± 33.90	0
22	14483.90 ± 37.51	2
23	14135.84 ± 39.85	0
24	12333.90 ± 24.51	1
25	12338.62 ± 22.92	1
26	12334.88 ± 26.64	1
27	12115.34 ± 24.29	1
28	14665.61 ± 18.99	0
29	14652.72 ± 19.91	2
30	14314.01 ± 25.06	0
31	14659.39 ± 19.67	2
32	14309.83 ± 22.49	0
33	12510.37 ± 16.12	1
34	12516.75 ± 18.32	1
35	12520.57 ± 22.16	1
36	12296.82 ± 20.28	1

TABLE III
ANOMALY SCORE VALUES IN THE ANOMALOUS AND NONANOMALOUS VIDEOS IN THE REAL DATA

Anomaly score		Anomalous video	Non-anomalous videos
KDE-AMD	Min.	1.775×10^{-1}	5.552×10^{-5}
	Mean.	1.775×10^{-1}	8.676×10^{-4}
	Max.	1.775×10^{-1}	2.845×10^{-2}
	Std.	0	1.186×10^{-3}
Global KDE	Min.	1.102	2.960×10^{-2}
	Mean.	1.102	3.727×10^{-2}
	Max.	1.102	9.478×10^{-2}
	Std.	0	4.175×10^{-3}
D-Markov machine	Min.	1.832	5.810×10^{-2}
	Mean.	1.832	1.036
	Max.	1.832	2.064
	Std.	0	9.280×10^{-1}
Kalman filter	Min.	7.915×10^{-2}	3.718×10^{-3}
	Mean.	7.915×10^{-2}	5.271×10^{-3}
	Max.	7.915×10^{-2}	7.392×10^{-3}
	Std.	0	8.319×10^{-4}
Deep neural network	Min.	1558.703	133.241
	Mean.	1558.703	358.351
	Max.	1558.703	1014.479
	Std.	0	162.292

type. For this reason, each video is classified using only videos of the same type, applying leave-one-out cross-validation. Out of the 4223 videos, only a single video is classified as anomalous by human experts. Each final workpiece on this production line was inspected by nondestructive testing procedures. In addition, a sample of them (1 in 1000) was inspected by destructive testing procedures. These testing procedures confirmed that only a single video was anomalous. In the anomalous video, the laser moves with a slightly different speed than in the remaining videos, changing the heat distribution and final result of the workpiece. However, the pattern shape in the anomalous video is the same as that in Fig. 2(a). Table III lists the minimum, mean, maximum, and standard deviation values for the nonanomalous anomaly scores. For the anomalous video, only one anomaly score is reported. The anomaly scores are computed differently for each algorithm, and therefore, not comparable with each other. However, we analyze the behavior of the anomaly score for each algorithm when classifying anomalous and nonanomalous workpieces. Anomaly scores should be higher for anomalous workpieces than those for nonanomalous workpieces. Note that the D-Markov does not detect the anomaly, because the anomaly score for the anomalous video is lower than the maximum anomaly score for the nonanomalous videos. However, the KDE–AMD, global KDE, Kalman filter, and deep neural network detect the real anomaly with an anomaly score approximately 6.24, 11.63, 10.71, and 1.54 times higher than the maximum anomaly score value in the nonanomalous videos, respectively. Moreover, the standard deviations are quite small, showing that the anomaly scores are quite stable and are very close to zero in the nonanomalous videos. Note that the judgment of human experts is used only for evaluation purposes. Using human experts to inspect each workpiece would be infeasible, given the numerous videos generated. In fact, the human experts found an anomaly video driven by the anomaly scores provided by the KDE–AMD algorithm. In a real-world scenario, the system can work completely autonomously, because a threshold value between 2.845×10^{-2} and 1.775×10^{-1} will correctly classify all videos. Thus, we consider the KDE–AMD performance on par with the global KDE and Kalman-filter algorithms, even when the anomaly score ratio of the anomalous to nonanomalous videos (6.24) is slightly smaller than that of the other algorithms. Furthermore, the KDE–AMD anomaly score ratio is higher than the deep neural network anomaly score ratio. Further experimentation (computing the anomaly score values of more anomalous videos) is necessary to fine-tune the threshold value and provide a more accurate comparison of the classifiers.

B. Simulated Data

Owing to the high reliability of the laser-heating process, there is only a single anomalous video. Consequently, the evaluation can be strengthened using simulated anomalous data by modifying the normal datasets to generate them. The real-data scenario in Section V-A showed that KDE–AMD can detect unusual movement patterns. However, more subtle deviations from the baseline can occur with a gradual degradation of the laser positioning system. We must recall that the working principle of a galvanometer is the oscillation at high frequency around its required position with a Gaussian shape. Therefore, Gaussian noise is introduced at each laser-spot position in both dimensions independently to model Gaussian-shaped abnormal oscillations. Such anomaly is simulated because a gradual degradation of the laser positioning system is expected to modify the laser-spot positions very slightly. This is important because the laser positioning system requires constant maintenance, and appropriate corrective actions can be taken only when the anomaly

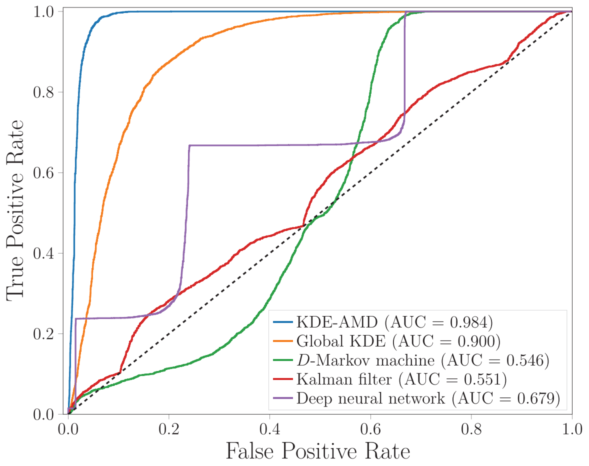


Fig. 8. Comparison results of the algorithms with $\sigma^2 = 0.02$.

score deviation from the baseline is sufficiently large (predictive maintenance).

A total of 4222 datasets are available for testing, because an anomalous dataset can be created for each normal dataset. In this analysis, we omit the known real anomaly to generate subtle variations relative to the normal data. A Gaussian noise of mean 0 and variance 0.02 is added to each laser position dimension in the normal datasets to produce anomalous datasets. The methods are validated using ROC curves and the area under the curve (AUC) [30]. Fig. 8 shows the results obtained from the four algorithms. The AUC is rounded to three decimals in the results reported here. The ROC curves show the true- and false-positive rates of the algorithm for different threshold values for the anomaly score. The best threshold should be selected while accounting for the importance of reducing the false-positive rate or increasing the true-positive rate. This decision can be made by measuring the cost of errors (false positive versus false negative). The D-Markov and Kalman filter have an AUC close to 0.5. Their performance is almost equivalent to that of a random classifier (a classifier that selects a class randomly). The deep neural network algorithm performs worse than the KDE–AMD algorithm but clearly better than a random classifier. However, the global KDE and KDE–AMD algorithms have a reasonably good performance with an AUC greater than 0.9. In particular, KDE–AMD is extremely close to the perfect classification (achieved when AUC $=$ 1). This simulation demonstrates the good classification capabilities of the KDE–AMD algorithm for the different anomaly types.

C. Classification Times

As noted, we require an in-process classification. Given the different characteristics of each batch type (see Table II), the classification times can be different for each. Nevertheless, the analysis of the classification times for the same batch type can give us some insights about those for each algorithm. Table IV lists the average classification time taken for a single workpiece for batch type 1. The deep neural network and D-Markov machine are the fastest models and perform in-process classification. Note that for the D-Markov machine having different symbolization techniques and number of divisions, the classification times are rather different. For example, with a symbolization of equal frequency and 35 × 35 divisions, the average classification time is equal to 3.014 s. The Kalman filter can also conduct the in-process classification.

TABLE IV
CLASSIFICATION TIMES FOR DIFFERENT ALGORITHMS

Algorithm	Time (s)
KDE-AMD	3.072
Global KDE	4.137
D-Markov machine	0.147
Kalman filter	0.634
Deep neural network	0.142

As mentioned, the global KDE model is only trained with a complete set of positions of the four videos. The classification times would have been impractical if we had used the leave-one-out cross-validation. Therefore, it cannot be implemented in a real system trained with 4222 normal videos.

The KDE–AMD algorithm is slightly slower than the Kalman filter and D-Markov machines. However, the classification times are sufficiently good to achieve in-process classification. The classification time for the KDE–AMD algorithm is measured without implementing the work parallelization, so that the system can be made faster without changing the algorithm. Furthermore, approximately 0.5 s are required to compute the Kullback–Leibler divergence, regardless of the selected parameter values for the KDE–AMD algorithm. A faster evaluation of the Kullback–Leibler divergence can be obtained by computing it with less precision.

VI. Parameter Sensitivity Analysis

The proposed KDE–AMD algorithm requires three parameters to train the model, as outlined in Table I. In this section, we show how the selection of these parameters affects the classification performance and classification times of the proposed algorithm.

We always set parameters R and C as equal, because the size of both axes is the same in the input data. This is a simplification of the parameter search space because it is clearly possible for nonequal R and C values to have better results. As noted, we test parameters R and C using the configurations 16 × 16, 20 × 20, 25 × 25, 30 × 30, 35 × 35, and 40 × 40, fixing the parameter λ to 5. We choose $\lambda = 5$ because with a lower value, the algorithm would estimate densities with less than five instances, which is very less data. The AUC scores for all configurations are shown in Table V, where the differences in the classification performance are marginal. This result demonstrates that the KDE–AMD algorithm is quite robust to the parameter values selected for R and C. The classification times are also affected by the selection of parameters R and C. It is remarkable that classification times are reduced when the number of regions increases. This is explained by the fact that when there are

TABLE V
RESULTS FOR DIFFERENT R AND C WITH $\lambda = 5$

$R \times C$	AUC score	Time (s)	$R \times C$	AUC score	Time (s)
16×16	0.979	22.416	30×30	0.975	4.328
20×20	0.977	13.048	35×35	0.979	3.098
25×25	0.969	9.307	40×40	0.974	2.771

TABLE VI
RESULTS FOR DIFFERENT λ WITH $R = C = 35$

λ	AUC score	Time (s)	λ	AUC score	Time (s)
5	0.979	3.098	30	0.982	3.090
10	0.980	3.098	40	0.975	3.069
15	0.978	3.070	50	0.984	3.072
20	0.981	3.110	200	0.983	3.026

more regions, the KDE model of each region contains fewer laser-spot positions. Therefore, KDE takes less time to evaluate each region. Additionally, because there are fewer laser-spot positions per region, it is more likely that a KDE model would not be assigned to some regions, because the number of laser-spot positions is below λ. This also reduces the classification time because the nonassigned regions are easily evaluated by assigning the constant value $\frac{\epsilon}{1+\epsilon}$.

For the λ parameter, values 5, 10, 15, 20, 30, 40, 50, and 200 are tested by fixing R and C to 35, because this is the best configuration in Table V. The results are shown in Table VI. Again, the differences in the classification performance are marginal. Thus, the KDE–AMD algorithm is also quite robust with respect to the λ parameter. Note that increasing the λ parameter can improve the classification performance, because the KDE models of these regions having few instances are probably produced by noise in the data. An increase in the λ parameter implies that noise-generated KDE models are not included in the KDE–AMD model. Thus, the classification performance increases. We recommend testing different values of λ if there is noise in the data to find the best compromise between reduction of noise and discarding a lot of data. Also, increasing the λ value should decrease the classification time because there are fewer regions with KDE models assigned.

VII. CONCLUSION

Anomaly detection is a key step for ensuring the production of high-quality products in industry. This article presents a novel methodology (i.e., KDE–AMD) for an efficient detection of anomalies in a laser surface-heating process. This process applies energy to the surface of a workpiece in a controlled manner following a known pattern containing numerous changes in direction (eight changes per cycle). It is applied at a high frequency (100 Hz and 10 fps). Pattern analysis using a high-speed thermal camera with a frame rate of 1000 fps is complex, because the classification times must be short, even when processing numerous frames per video. The first step of the anomaly detection methodology is to detect the laser-spot positions in each frame. Then, a matrix of KDE models is built to accurately represent the changes in the position between two consecutive frames. This matrix is an ensemble of KDE models taking advantage of the temporal/spatial locality of each laser-spot movement. Using this matrix of KDE models, we compute the likelihood for each movement. An anomaly score can then be computed using the Kullback–Leibler divergence between the distribution of the training and test movement probabilities. KDE–AMD is fast enough to achieve in-process classification, and the classification results are at least as good as the other state-of-the-art algorithms.

In the future, we will improve the performance and classification times of the methodology. An alternative variant of the algorithm can entail using nearest neighbors. To this end, for a given movement $\mathbf{M} = (\mathbf{s}_O, \mathbf{s}_D)$, we would search for other movements whose origin positions are the nearest neighbors of $\mathbf{s}_O$, instead of distributing all movements in different regions during training. This variant can be computationally expensive. Thus, a fast approximation to the nearest neighbors, such as k-d trees [31], is required. The parallelization and streaming-like optimizations can improve the speed of our approach. If we obtain more anomalous videos in the future, a fine-grained threshold can be computed to detect an anomaly. For example, the application of the extreme value theory [32], [33] would be an interesting future research direction, because it would use a small sample of real anomalies to obtain another sound anomaly score.

REFERENCES

[1] R. Drath and A. Horch, "Industrie 4.0: Hit or hype? [industry forum]," *IEEE Ind. Electron. Mag.*, vol. 8, no. 2, pp. 56–58, Jun. 2014.

[2] D. A. Clifton, L. A. Clifton, P. R. Bannister, and L. Tarassenko, "Automated novelty detection in industrial systems," in *Proc. Adv. Comput. Intell. Ind. Syst.*, 2008, pp. 269–296.

[3] C. Rao, A. Ray, S. Sarkar, and M. Yasar, "Review and comparative evaluation of symbolic dynamic filtering for detection of anomaly patterns," *Signal Image Video. Process.*, vol. 3, no. 2, pp. 101–114, 2009.

[4] S. Yin and X. Zhu, "Intelligent particle filter and its application to fault detection of nonlinear system," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3852–3861, Jun. 2015.

[5] B. Zhang, C. Sconyers, C. Byington, R. Patrick, M. E. Orchard, and G. Vachtsevanos, "A probabilistic fault detection approach: Application to bearing fault detection," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 2011–2018, May 2011.

[6] D. Tobon-Mejia, K. Medjaher, and N. Zerhouni, "CNC machine tool's wear diagnostic and prognostic by using dynamic Bayesian networks," *Mech. Syst. Signal. Process.*, vol. 28, pp. 167–182, 2012.

[7] J. Mori and J. Yu, "Dynamic Bayesian network based networked process monitoring for fault propagation identification and root cause diagnosis of complex dynamic processes," *IFAC Proc. Volumes*, vol. 46, no. 32, pp. 678–683, 2013.

[8] S. Gupta and A. Ray, "Real-time fatigue life estimation in mechanical structures," *Meas. Sci. Technol.*, vol. 18, no. 7, pp. 1947–1957, 2007.

[9] S. Sarkar, S. Sarkar, N. Virani, A. Ray, and M. Yasar, "Sensor fusion for fault detection and classification in distributed physical processes," *Front. Robot. AI.*, vol. 1, no. 16, pp. 1–9, 2014.

[10] P. Freeman, R. Pandita, N. Srivastava, and G. J. Balas, "Model-based and data-driven fault detection performance for a small UAV," *IEEE/ASME Trans. Mechatron.*, vol. 18, no. 4, pp. 1300–1309, Aug. 2013.

[11] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009.

[12] E. Parzen, "On estimation of a probability density function and mode," *Ann. Math. Statist.*, vol. 33, no. 3, pp. 1065–1076, 1962.

[13] A. Gabilondo, J. Domínguez, C. Soriano, and J. Ocaña, "Method and system for laser hardening of a surface of a workpiece," *U.S. Pat.* US20150211083A1, Jul. 30, 2015.

[14] N. B. Heidenreich, A. Schindler, and S. Sperlich, "Bandwidth selection for kernel density estimation: A review of fully automatic selectors," *AStA Adv. Statist. Anal.*, vol. 97, no. 4, pp. 403–433, 2013.

[15] B. W. Silverman, *Density Estimation Statist. Data Anal.*, 1st ed., (Monographs on Statistics and Applied Probability). London, U.K.: Chapman and Hall, 1986.

[16] D. W. Scott, *Multivariate Density Estimation: Theory, Practice, and Visualization*, 2nd ed., (Probability and Statistics). Hoboken, NJ, USA: Wiley, 2015.

[17] M. Wand and M. Jones, *Kernel Smoothing*, 1st ed., (Monographs on Statistics and Applied Probability). London, U.K.: Chapman and Hall, 1994.

[18] X. Gao, D. You, and S. Katayama, "Seam tracking monitoring based on adaptive Kalman filter embedded Elman neural network during high-power fiber laser welding," *IEEE Trans. Ind. Electron.*, vol. 59, no. 11, pp. 4315–4325, Nov. 2012.

[19] M. Jäger, S. Humbert, and F. A. Hamprecht, "Sputter tracking for the automatic monitoring of industrial laser-welding processes," *IEEE Trans. Ind. Electron.*, vol. 55, no. 5, pp. 2177–2184, May 2008.

[20] M. Jäger, C. Knoll, and F. A. Hamprecht, "Weakly supervised learning of a classifier for unusual event detection," *IEEE Trans. Image Process.*, vol. 17, no. 9, pp. 1700–1708, Sep. 2008.

[21] M. Jäger and F. A. Hamprecht, "Principal component imagery for the quality monitoring of dynamic laser welding processes," *IEEE Trans. Ind. Electron.*, vol. 56, no. 4, pp. 1307–1313, Apr. 2009.

[22] R. Linares, "Laser beam welding quality monitoring system based in high-speed (10 kHz) uncooled MWIR imaging sensors," *Proc. SPIE*, vol. 9485, pp. 289–295, 2015.

[23] T. Fawcett, "An introduction to ROC analysis," *Pattern. Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[24] S. Kullback and R. A. Leibler, "On information and sufficiency," *Ann. Math. Statist.*, vol. 22, no. 1, pp. 79–86, 1951.

[25] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016.

[26] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Trans. ASME–J. Basic Eng.*, vol. 82, no. 1, pp. 35–45, 1960.

[27] A. Khatkhate, A. Ray, E. Keller, S. Gupta, and S. C. Chin, "Symbolic time-series analysis for anomaly detection in mechanical systems," *IEEE/ASME Trans. Mechatron.*, vol. 11, no. 4, pp. 439–447, Aug. 2006.

[28] F. Chollet*et al.*, "Keras," 2015.[Online]. Available: https://keras.io

[29] D. Atienza, C. Bielza, J. Diaz-Rozo, and P. Larrañaga, "Anomaly detection with laser heat treatment thermal videos," 2019. [Online]. Available: http://dx.doi.org/10.21227/7zbf-se57

[30] A. P. Bradley, "The use of the area under the ROC curve in the evaluation of machine learning algorithms," *Pattern. Recognit.*, vol. 30, no. 7, pp. 1145–1159, 1997.

[31] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975.

[32] A. A. Balkema and L. de Haan, "Residual life time at great age," *Ann. Probab.*, vol. 2, no. 5, pp. 792–804, Oct. 1974.

[33] J. Pickands, "Statistical inference using extreme order statistics," *Ann. Statist.*, vol. 3, no. 1, pp. 119–131, Jan. 1975.

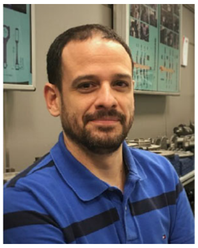
David Atienza received the B.S. degree in computer science from the University of Burgos, Burgos, Spain, in 2014, and the M.S. degree in artificial intelligence, in 2016, from the Technical University of Madrid, Madrid, Spain, where he is currently working toward the Ph.D. degree in artificial intelligence.

His research interests include anomaly detection, density estimation, and probabilistic graphical models, as well as their applications to the Industry 4.0 paradigm.

Concha Bielza (Member, IEEE) received the M.S. degree in mathematics from Universidad Complutense de Madrid, Madrid, Spain, in 1989, and the Ph.D. degree in computer science from Universidad Politécnica de Madrid, Madrid, Spain, in 1996 (extraordinary doctorate award).

Since 2010, he has been a Full Professor of Statistics and Operations Research with the Departamento de Inteligencia Artificial, Universidad Politécnica de Madrid. She has authored or coauthored more than 120 papers in impact factor journals and has supervised 17 Ph.D. theses. Her research interests include in the areas of probabilistic graphical models, decision analysis, metaheuristics for optimization, data mining, classification models, and real applications, such as biomedicine, bioinformatics, neuroscience, industry, and sport analytics.

Dr. Bielza was awarded the 2014 UPM Research Prize and the 2020 Research Award for Significant Contribution in the Field of Machine Learning (Amity University, India).



Javier Diaz-Rozo received the M.Eng. degree in mechanical engineering from the University of Los Andes, Bogotá, Colombia, in 2001, the M.Sc. degree in advanced manufacturing technology and systems management from the University of Manchester, Manchester, U.K., in 2003, and the Ph.D. degree in artificial intelligence from Universidad Politécnica de Madrid, Madrid, Spain, in 2019.

He is currently the Chief Technology Officer with Aingura IIoT, Elgoibar, Spain. He has accumulated nearly 15 years of industrial experience researching in different positions related with research and development: Research and Development Project Manager with Ikergune, the Etxe-Tar Group Research and Development Unit, responsible for the advanced manufacturing research area, Senior Consultant in a research and development consulting firm from 2010 to 2014, the Research and Development Director in a business group mainly dedicated to the wind energy sector from 2008 to 2010, and the Director for the Advanced Manufacturing Area with the ASCAMM Technology Centre, Barcelona, Spain, from 2006 to 2008.

Pedro Larrañaga (Member, IEEE) received the M.Sc. degree in mathematics (statistics) from the University of Valladolid, Valladolid, Spain, in 1981, and the Ph.D. degree in computer science from the University of the Basque Country, Leioa, Spain (excellence award), in 1995.

He is currently a Full Professor of computer science and artificial intelligence with the Technical University of Madrid (UPM), Madrid, Spain, since 2007. Before moving to UPM, his academic career was developed with the University of the Basque Country (UPV-EHU), Leioa, Spain, at several faculty ranks: Assistant Professor (1985–1998), Associate Professor (1998–2004), and Full Professor (2004–2007). He earned the habilitation qualification for Full Professor in 2003. He has authored or coauthored more than 200 papers in impact factor journals and has supervised 30 Ph.D. theses. His research interests include the areas of probabilistic graphical models, metaheuristics for optimization, data mining, classification models, and real applications, such as biomedicine, bioinformatics, neuroscience, Industry 4.0, and sports.

Dr. Larrañaga has been a Fellow of the European Association for Artificial Intelligence since 2012 and that of the Academia Europaea since 2018. He has been awarded the 2013 Spanish National Prize in Computer Science and the prize of the Spanish Association for Artificial Intelligence in 2018. In 2020, he will receive the Amity Research Award in Machine Learning, New Delhi.