From fault detection to one-class severity discrimination of 3D printers with one-class support vector machine

Chuan Li [a,1], Diego Cabrera [a,b,*,1], Fernando Sancho [c], Mariela Cerrada [b],
René-Vinicio Sánchez [b], Edgar Estupinan [d]

[a] National Research Base of Intelligent Manufacturing Service, Chongqing Technology and Business University, Chongqing 400067, China
[b] GIDTEC, Universidad Politécnica Salesiana, Ecuador
[c] Department of Computer Science and Artificial Intelligence, Universidad de Sevilla, Spain
[d] Department of Mechanical Engineering, University of Tarapaca, Arica, Chile

Keywords:
Fault detection
Severity discrimination
One-class support vector machine
3D printer
Bidirectional generative adversarial network

ABSTRACT

The lack of faulty condition data reduces the feasibility of supervised learning for fault detection or fault severity discrimination in new manufacturing technologies. To deal with this issue, one-class learning arises for building binary discriminative models using only healthy condition data. However, these models have not been extrapolated to severity discrimination. This paper proposes to extend OCSVM, which is typically used for fault detection, to 3D printer fault severity discrimination. First, a set of features is extracted from a set of normal signals. An optimized OCSVM model is obtained by tuning the kernel and model hyperparameters. The resulting models are evaluated for fault detection and fault severity discrimination using a proposed performance evaluation approach. Experimental comparisons for belt-based faults in 3D printers show that the distance to the hyperplane has the information to discriminate the severity level, and its use is feasible. The proposed hyperparameter optimization technique improves the OCSVM for fault detection and severity discrimination compared to some other methods.

1. Introduction

Fault detection and severity assessment are two fundamental tasks of system health management. In this context, fault (anomaly) detection is the determination of whether a system is in normal working condition or some component contains a fault. Anomaly detection has been extensively studied in such fields as abnormal activity recognition [1], communication networks [2], malicious file detection [3], fault detection in power transmission lines and distribution systems [4], and industrial condition monitoring [5]. In the last field, although model-based approaches have been proposed [6], the current trend is to use data-driven techniques for reasons of flexibility and accuracy [7].

In data-driven approaches, a detection model is built using data collected from the monitored system. Informative features are extracted from raw signals. A classifier is then created using supervised, semi-supervised [8], unsupervised, or one-class learning [9]. Although supervised learning is superior to one-class learning in diverse applications [10,11], it only applies when labeled data in both normal and abnormal conditions can be obtained from the system, and it is usually expensive. Semi-supervised and unsupervised learning typically require fewer labeled examples [12], but still need data in every system condition. Evolving approaches [13] address the issue of data availability in the training stage by recognizing new patterns in the testing stage and including them in the base of knowledge. These approaches require retraining of classification models after identifying a new cluster, which could be computationally expensive. One-class learning only uses data in normal condition to build the classifier, showing clear advantages in the absence of abnormal data and requiring less computation.

The popular OCSVM [14] is a one-class learning model that has shown versatility and success in combination with feature extraction techniques for different applications [15,16]. It aims to build a decision hyperplane in a projected space of the input features. The hyperplane has samples in normal condition on one side, and abnormal ones on the other. The hyperplane is optimized using only data in normal condition.

☆ Acknowledgments: The work was sponsored in part by GIDTEC Research Group of Universidad Politécnica Salesiana, the National Natural Science Foundation of China (51775112), the MoST Science and Technology Partnership Program (KY201802006), the Key Project of the Chongqing Natural Science Foundation (cstc2019jcyj-zdxmX0013), and the CTBU Open Project (KFJJ2019059)

* Corresponding author at: GIDTEC, Universidad Politécnica Salesiana, Ecuador.
E-mail address: dcabrera@ups.edu.ec (D. Cabrera).

[1] The two authors contributed equally to this paper.

Nomenclature

Symbols

$\|x\|_1$	L1 norm of x
$\|x\|_2$	L2 norm of x
α, β	Lagrange multipliers
$\cdot$	Inner product
γ	Kernel hyperparameter
$\mathbb{R}^d$	Space of input examples
$\mathbb{R}^n$	Feature space
$\mathbb{R}^T$	Signal space of length T
∇	Gradient operator
ν	OCSVM hyperparameter
$\oslash$	Element-wise division
ρ	Offset of hyperplane
$\top$	Transpose operator
ξ_i	Slack variable of ith training example
w	Vector normal to hyperplane

Abbreviations

1DCNN	One-dimensional convolutional neural network
1DDNN	One-dimensional deconvolutional neural network
ase	Average severity error
BiGAN	Bidirectional generative adversarial network
FNR	False-negative rate, rate of normal condition examples classified as faulty condition examples
FPR	False-positive rate, rate of faulty condition examples classified as normal condition examples
HVAC	Heating, ventilation, and air-conditioning system
OCSVM	One-class support vector machine
RBF	Radial basis function

OCSVM has the advantage that few hyperparameters (typically two for OCSVM with an RBF kernel) are required from the user. In industrial condition monitoring, grid searching combined with cross-validation is most often used for hyperparameter optimization. It has been successfully applied to supervised learning [17,18], but cannot be used with one-class learning, considering the unavailability of faulty data. For example, grid searching has been used to optimize OCSVM hyperparameters in bearing fault detection [19,20] and HVAC anomaly detection [21], but such applications require a validation set with abnormal data, which violates the restriction of one-class learning.

OCSVM hyperparameter configuration is most often accomplished by setting them to specific values or by rule of thumb. Such approaches have been applied to fault detection of complex industrial processes [22,23], bearing fault detection [24], simulated dynamical system anomaly detection [25], and HVAC system fault detection [26,27]. However, they suffer from lack of evidence that the hyperparameters are optimally configured, hence better performance could be possible. Considering the lack of anomalous data in OCSVM applications, heuristics-based hyperparameter optimization methods have been proposed. Specifically, [28] proposed the searching of γ by optimizing the ratio between the variance and a metric of the kernel matrix. [29] presented a method for optimizing ν and γ by a trade-off between the number of support vectors and the maximization of the objective function. [30] introduced an heuristic based on a metric over the Gaussian space for optimizing γ. The previous works have shown a high performance in synthetic and applications different to severity discrimination, and their evaluation was performed only for RBF kernels. Therefore, other heuristics or the generalization of the above methods to other kernels are open study field.

Fault severity discrimination is a more advanced task that requires the classification of a severity condition for a detected fault. It has been studied from a supervised learning perspective in applications of a wind turbine gearbox [31], helical gearbox [32,33], and bearings [34]. As before, data availability for all severity levels is assumed, but is infeasible in practice.

From the above, we identify three issues. (i) Information about the fault severity that one-class models can contain has not been exploited. Specifically, information from the distance from a sample to the hyperplane in OCSVM could be used as a severity level discriminator for a set of features also learned using a one-class training set. (ii) OCSVM hyperparameter optimization has not been addressed beyond fault detection. Supervised approaches based on grid search are useless without an evaluation set (faulty data) because an evaluation metric for the one-class classifier without this set has not yet been defined. (iii) Unsupervised optimization approaches have been restricted to the RBF kernel in applications different to fault severity discrimination.

This work introduces hyperparameter optimization approaches for γ and ν of OCSVM-based models in the context of 3D printer fault detection and fault severity discrimination. Our main contributions are summarized as follows: (i) a previously reported unsupervised optimization method for the RBF kernel hyperparameter is formally generalized and applied to other kernels. (ii) an unsupervised method is introduced to optimize the OCSVM hyperparameter; and (iii) a methodology to evaluate OCSVMs created with different kernels and optimization approaches in the context of fault severity discrimination is provided.

The rest of the paper is organized as follows. Section 2 introduces the formulation of OCSVM. Section 3 presents the proposed OCSVM hyperparameter optimization, and the methodology is detailed in Section 4. In Section 5, the 3D printer test bed and experimental setup for comparisons are introduced. Results are shown in Section 6. We relate some conclusions in Section 7.

2. Preliminaries

An OCSVM description as the core of a fault detection model is presented, with details on its optimization, the kernel trick, and some popular kernels.

2.1. OCSVM

One-class support vector machines were first proposed [14] to address the one-class learning task using a similar framework to support vector machines [35] in binary classification. The fundamental idea is to build a hyperplane that optimally separates the available one-class dataset from the origin in a high-dimensional feature space. In this context, the optimum refers to the hyperplane that maximizes its margin (distance) to the origin, while keeping most of the one-class examples. In this way, a new example can be classified as positive (belonging to the available class) if it is over the hyperplane (far from the origin), and negative otherwise.

Formally, let $x \in \mathbb{R}^d$ be an example to classify, and let $\phi : \mathbb{R}^d \to \mathbb{R}^n$ be a nonlinear mapping function from the input to the feature space. A decision function to classify x is defined by

$$f(x) = \operatorname{sgn}(w \cdot \phi(x) - \rho), \tag{1}$$

where $\operatorname{sgn}(z)$ is the sign function, which returns 1 when $z \geq 0$, and -1 otherwise, and w and ρ are the parameters of the hyperplane in $\mathbb{R}^n$, where $w \cdot y - \rho = 0$.

Let $\{x_i\}_{i=1}^m$ be a one-class training set. Then the optimal hyperplane parameters are found by solving the following optimization problem:

$$\begin{aligned} \min_{w,\rho,\xi} \quad & \frac{1}{2}\|w\|^2 + \frac{1}{\nu m}\sum_{i=1}^{m} \xi_i - \rho \\ \text{s.t.} \quad & w \cdot \phi(x_i) \geq \rho - \xi_i, \quad i = 1, \dots, m \\ & \xi_i \geq 0, \end{aligned} \tag{2}$$

where the slack variables $\xi = [\xi_1, \dots, \xi_m]$ allow for the presence of anomalous examples in the training set, and ν limits the fraction of training examples classified as anomalous.

The method of Lagrange multipliers can be used to solve (2), enabling the transformation of the problem to the primal problem:

$$\begin{aligned} \min_{w,\rho,\xi} \quad & \max_{\alpha,\beta} L(w, \rho, \xi, \alpha, \beta) \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \dots, m \\ & \beta_i \geq 0, \quad i = 1, \dots, m, \end{aligned} \tag{3}$$

where $\alpha = \{\alpha_1, \dots, \alpha_m\}$ and $\beta = \{\beta_1, \dots, \beta_m\}$ are Lagrange multipliers, and $L(\cdot)$ is the Lagrangian, defined as

$$L = \frac{1}{2}\|w\|^2 + \frac{1}{\nu m}\sum_{i=1}^{m} \xi_i - \rho - \sum_{i=1}^{m} \alpha_i [w \cdot \phi(x_i) + \xi_i - \rho] - \sum_{i=1}^{m} \beta_i \xi_i. \tag{4}$$

Instead of solving (3), it is computationally convenient to reformulate the problem in its dual version,

$$\begin{aligned} \max_{\alpha,\beta} \quad & \min_{w,\rho,\xi} L(w, \rho, \xi, \alpha, \beta) \\ \text{s.t.} \quad & \alpha_i \geq 0, \quad i = 1, \dots, m \\ & \beta_i \geq 0, \quad i = 1, \dots, m, \end{aligned} \tag{5}$$

which has the same solution if the Karush–Kuhn–Tucker (KKT) conditions are met for the Lagrangian [36], which is the case with (4).

Application of the KKT conditions, $\nabla_{w,\xi,\rho} L = 0$, to (5) results in the following equations:

$$\nabla_w L = w - \sum_{i=1}^{m} \alpha_i \phi(\xi_i) = 0 \implies w = \sum_{i=1}^{m} \alpha_i \phi(\xi_i) \tag{6}$$

$$\nabla_\xi L = \frac{1}{\nu m} - \alpha - \beta = 0 \implies \beta = \frac{1}{\nu m} - \alpha \tag{7}$$

$$\nabla_\rho L = -1 + \sum_{i=1}^{m} \alpha_i = 0 \implies \sum_{i=1}^{m} \alpha_i = 1. \tag{8}$$

Combining the constraints in (5) with (7) results in the restriction for α:

$$0 \leq \alpha_i \leq \frac{1}{\nu m}. \tag{9}$$

Let L^* be the $\min L$ from the substitution of (6), (7), and (8) in (4),

$$L^* = -\frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j \phi(x_i)^\top \phi(x_j), \tag{10}$$

and combine that with (8) and (9), so the dual optimization problem becomes

$$\begin{aligned} \max_{\alpha} \quad & L^*(\alpha) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \frac{1}{\nu m}, \quad i = 1, \dots, m \\ & \sum_{i=1}^{m} \alpha_i = 1, \end{aligned} \tag{11}$$

which is simpler than the primal optimization problem.

2.2. Kernel trick and decision function

Let $\phi(x_i) \cdot \phi(x_j) = \phi(x_i)^\top \phi(x_j)$ be the inner product in the feature space of the projected x_i and x_j. The kernel trick defines a kernel function $k(x_i, x_j) = \phi(x_i) \cdot \phi(x_j)$ to avoid an explicit projection to the feature space. Considering $k(\cdot, \cdot)$ in (11), the dual version is

$$\begin{aligned} \min_{\alpha} \quad & \frac{1}{2}\sum_{i=1}^{m}\sum_{j=1}^{m} \alpha_i \alpha_j k(x_i, x_j) \\ \text{s.t.} \quad & 0 \leq \alpha_i \leq \frac{1}{\nu m}, \quad i = 1, \dots, m \\ & \sum_{i=1}^{m} \alpha_i = 1, \end{aligned} \tag{12}$$

where $\phi(\cdot)$ is not required, and an algorithm [14] exists to solve the problem.

The decision function in (1) can be reformulated to be independent from w and $\phi(\cdot)$ by replacing (6) and applying the kernel trick to obtain

$$f(x) = \operatorname{sgn}\Big(\sum_{i=1}^{m} \alpha_i k(x_i, x) - \rho\Big), \tag{13}$$

where the elements x_i with $\alpha_i > 0$ are called support vectors. It is easy to check that the hyperplane passes through the support vectors, or equivalently,

$$\rho = \sum_{\alpha_i \neq 0} \alpha_i k(x_i, x_s), \ \forall x_s. \tag{14}$$

2.3. Kernels

A function can be used as a kernel only if it is positive-definite. Among the most used functions that fulfill this condition we can find:

$$\text{linear:} \quad k(x_i, x_j) = x_i \cdot x_j \tag{15}$$

$$\text{cosine:} \quad k(x_i, x_j) = \frac{x_i \cdot x_j}{\|x_i\| \, \|x_j\|} \tag{16}$$

$$\text{RBF:} \quad k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2} \tag{17}$$

$$\text{Laplacian:} \quad k(x_i, x_j) = e^{-\gamma \|x_i - x_j\|_1} \tag{18}$$

$$\text{additive } \chi^2: \quad k(x_i, x_j) = -\sum (x_i - x_j)^2 \oslash (x_i + x_j) \tag{19}$$

$$\chi^2: \quad k(x_i, x_j) = e^{-\gamma \sum (x_i - x_j)^2 \oslash (x_i + x_j)}, \tag{20}$$

where γ is the kernel hyperparameter. The use of additive χ^2 and χ^2 kernels is subject to $x_i \geq 0$ and $x_j \geq 0$.

Hyperparameters ν and γ should be determined based on the dataset properties before building the OCSVM model, although we usually rely on rule of thumb.

3. Hyperparameter optimization

In the optimization of γ and ν, we extend the approach of DFN (distances from training samples to their farthest neighbors and distances to their nearest neighbors) [37], which was initially proposed only for RBF kernels, for γ optimization in other kernels. A histogram-based approach is introduced to optimize ν.

3.1. γ optimization

RBF kernel

The DFN method [37] considers the optimal γ value that maximizes the difference between the farthest element average distance and the nearest element average distance of the dataset projected by ϕ, i.e., the objective function is defined as

$$g(\gamma) = \frac{1}{m}\sum_{i=1}^{m}\max_{j}\|\phi(x_i)-\phi(x_j)\|^2 - \frac{1}{m}\sum_{i=1}^{m}\min_{j\neq i}\|\phi(x_i)-\phi(x_j)\|^2. \tag{21}$$

Thus $g(\gamma)$ for the RBF kernel is:

$$g_{RBF}(\gamma) = \frac{2}{m}\sum_{i=1}^{m} e^{-\gamma \min_{j\neq i}\|x_i-x_j\|^2} - e^{-\gamma \max_j \|x_i-x_j\|^2}. \tag{22}$$

Expansion for Laplacian and χ^2 kernels

In (21), $\|\phi(x_i)-\phi(x_j)\|^2 = k(x_i,x_i)+k(x_j,x_j)-2k(x_i,x_j)$ from the expansion of the L2 norm and the kernel trick. Also, for Laplacian and χ^2 kernels, we have $k(x_i,x_i)=k(x_j,x_j)=1$, hence $\|\phi(x_i)-\phi(x_j)\|^2 = 2-2k(x_i,x_j)$. Replacing the previous term in (21) and solving gives

$$g(\gamma) = \frac{2}{m}\sum_{i=1}^{m}\max_{j\neq i} k(x_i,x_j) - \frac{2}{m}\sum_{i=1}^{m}\min_{j} k(x_i,x_j). \tag{23}$$

Finally, the $g(\gamma)$ function for each γ-dependent kernel is obtained by replacing (18) and (20) for Laplacian and χ^2 kernels, respectively, i.e.,

$$g(\gamma) = \frac{2}{m}\sum_{i=1}^{m} e^{-\gamma Near(x_i)} - \frac{2}{m}\sum_{i=1}^{m} e^{-\gamma Far(x_i)}, \tag{24}$$

where:

$$Near(x_i) = \min_{j\neq i}\|x_i-x_j\|_1, \text{ for Laplacian} \tag{25}$$
$$Far(x_i) = \max_{j}\|x_i-x_j\|_1, \text{ for Laplacian}$$

$$Near(x_i) = \min_{j\neq i}\sum (x_i-x_j)^2 \oslash (x_i+x_j), \text{ for } \chi^2 \tag{26}$$
$$Far(x_i) = \max_{j}\sum (x_i-x_j)^2 \oslash (x_i+x_j), \text{ for } \chi^2.$$

3.2. ν optimization

By definition, the hyperparameter ν represents the percentage of training elements considered out of the class, and it is commonly set to a small value, assuming a few outliers in the one-class set. The positive–negative distance approach [38] estimates this hyperparameter by iteratively incrementing it and choosing the value that maximizes the average in- to out-class distance. While this approach makes no assumptions about the data, it fails to estimate ν when in- and out-class are not separable.

Without loss of generality, we propose an approach to estimate ν without assuming there are few out-class samples in the dataset. Consider that for a specific ν, the percentage of support vector candidates of the OCSVM model is $1-\nu$, where both correct and incorrect in-class examples are present. Because out-class examples tend to be grouped, the incorrect in-class examples are closest to the resulting hyperplane, and consequently, several become support vectors. Thus, although the correct out-class examples are successfully classified, they appear in the feature space through sampling the less probable values for the same in-class data distribution, i.e., they are in the tail of the in-class data distribution.

However, proper configuration of ν minimizes the number of out-class examples as support vectors. Then the correct out-class examples have a different distribution and are not in the distribution tail. Based on this observation, the optimization of ν is summarized in Alg. 1.

```
Data: Dataset ($D$), kernel ($k$), $inc$, $\gamma$ (if applicable)
Result: Optimal $\nu$
$\nu = inc$;
while $\nu < 1$ do
    Create an OCSVM model for $D$ with $\nu$, $\gamma$, and $k$;
    $H_{in}$ = Histogram of in-class hyperplane distances;
    $H_{out}$ = Histogram of out-class hyperplane distances;
    if $H_{out}$ is not tail of $H_{in}$ then
        break;
    else
        $\nu = \nu + inc$;
    end
end
Return: $\nu$
```

Algorithm 1: ν estimation.

In Alg. 1, *inc* is the increment of ν for each iteration (usually, $inc = 0.05$), and γ is considered only if the kernel requires it. The "is not tail of" comparison verifies the continuity between the H_{in} and H_{out} histograms. For simplicity, this verification is performed by comparing the last bin of H_{out}, H_{out}^{last}, with the first one of H_{in}, H_{in}^{first}. The rule that H_{out} is not the tail of H_{in} if $H_{out}^{last} > H_{in}^{first}$ works for us, but more complex heuristics can also be evaluated.

4. Methodology

OCSVM-based models are learned using only datasets from normal condition, and this is one of the most often used frameworks for fault detection. In this case, the binary decision function introduced in (13) allows us to determine whether a new example belongs to normal condition (+1) or faulty condition (−1). However, the information given by the position of the example in the feature space could also be exploited as a fault severity discriminator. It is useful to study the relationship between the perpendicular distance from an example to the hyperplane under different fault severity levels, and how different configurations of the OCSVM hyperparameters (optimized vs. not optimized) and kernels determine the model performance beyond fault detection.

We introduce a methodology to compare OCSVM models in the context of fault detection and severity discrimination (see Fig. 1). The methodology can be summarized in the following steps.

1. **Signal acquisition:** We suppose that we can sample representatives (time-series) from a space $\mathcal{S}$ of signals produced by a machine. These signals can be sampled from both known and unknown severity conditions.
2. **BiGAN-based modeling:** An unsupervised BiGAN-based feature extractor model [39] is built using only signals in normal condition in the training stage. In the testing stage, this model is used to extract features of signals in other severity conditions.
3. **Min–max normalization:** The normalization parameters are computed from normal signals and applied to normalize training and testing datasets.

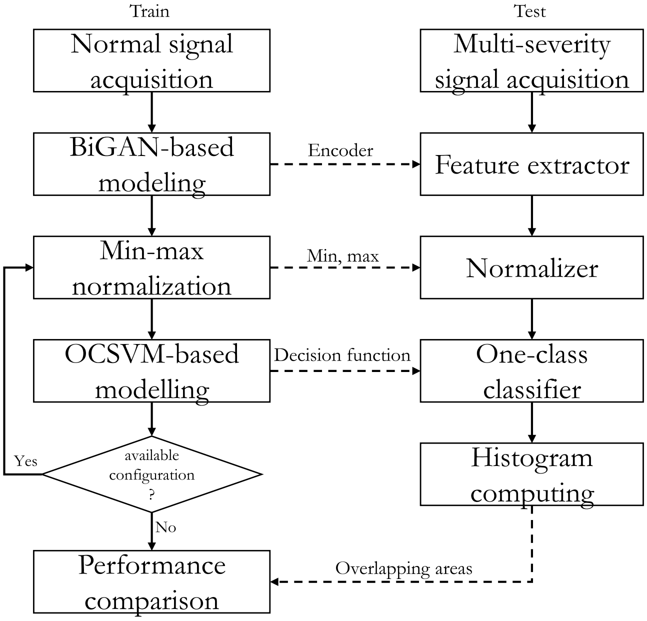


Fig. 1. Methodology for OCSVM-based fault severity discrimination and model performance evaluation.

4. **OCSVM-based modeling:** For each hyperparameter configuration and kernel, an OCSVM model is built from normal signals. The resulting model is used to determine the distance of each new signal to the hyperplane in the feature space.
5. **Histograms and performance comparison:** The unavailability of samples from different severity levels in the training dataset does not allow to evaluate the model with the classical classification metrics. Therefore, a novel evaluation approach is proposed, i.e., by using samples of each severity condition from step 1, the overlapping area in the histogram of hyperplane distances is computed for each model. Each histogram estimates the probability distribution of the distance given a severity condition. Then the overlapping area represents the probability of confusing a pair of severity conditions. Therefore, the best models for fault severity discrimination have the smallest overlapping areas.

4.1. Signal acquisition

Let $C = \{c_i\}_{i=1}^{K}$ be labels for the set of K severity levels. For every $c \in C$, let $s_c \in \mathbb{R}^T$ be one time-series representative of the machinery dynamics under condition c, with T sufficiently large. The goal is to build two sets, D_{train} and D_{test}, to respectively train and test the models:

- $D_{\text{train}} = \{s_{c_1,i}\}_{i=1}^{m}$ is a set of m normal condition (c_1) signals, which are easily obtained from most types of machinery. This set will be used for BiGAN-based model building (step 2), and a transformed version will be used for the remaining training stages (steps 3 and 4).
- $D_{\text{test}} = \{s_{c_i,j}\}_{i=1,j=1}^{K,M}$ is a set of M signals for every severity level, including c_1. It will be used to test only the models created from D_{train}. Note that this set, with data from different severity conditions, is unknown in the training phase, and will be used only to evaluate the models.

4.2. BiGAN-based modeling

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be a feature space where the different time-series of $\mathcal{S}$ can be represented. We aim to build an encoder function, $\mathrm{Enc} : \mathcal{S} \to \mathcal{X}$, that maps each element in $\mathcal{S}$ into a feature vector $x \in \mathcal{X}$.

One commonly used approach is to fix Enc by computing statistical condition indicators from different domains of s_c such as time, frequency, or time–frequency. Among the most common statistical indicators are RMS, standard deviation, kurtosis, and Shannon entropy [40]. The resulting vectors in each domain are concatenated to obtain a unique feature vector. The requirement of significant knowledge about the most informative statistical condition indicators regarding the severity level in the machinery is the main drawback of the previous approach, which makes it inapplicable to less-studied machinery such as 3D printers.

Another approach is supervised deep learning of the Enc function. A signal set containing examples from every severity level is used to estimate an Enc function that can separate instances within each severity level. Although this is the optimal approach, it cannot be used because of the signal acquisition constraints in machinery.

With this in mind, we propose to use a BiGAN to build Enc using only D_{train} [39]. This approach combines two 1DCNNs called Encoder (Enc) and Discriminator (Dis), and a 1DDNN called Generator (Gen). Generator is optimized to compete with Discriminator and try to fool it, i.e., Gen generates a synthetic signal that Dis may incorrectly recognize as real.

Formally,

$$\mathrm{Dis}(s, x) = \begin{cases} 1, & \text{if } s \in D_{\text{train}} \text{ and } x = \mathrm{Enc}(s) \\ 0, & \text{otherwise.} \end{cases} \tag{27}$$

Practically speaking, the previous models are parameterized by the θ_{Enc}, θ_{Gen}, and θ_{Dis} weight sets, which respectively define Enc, Gen, and Dis. The optimal parameter sets are subject to the solution of the following minimax optimization by gradient-based training algorithm:

$$\hat{\theta}_{\text{Dis}}, \hat{\theta}_{\text{Gen}}, \hat{\theta}_{\text{Enc}} = \arg\min_{\theta_{\text{Gen}},\theta_{\text{Enc}}} \arg\max_{\theta_{\text{Dis}}} L(\theta_{\text{Dis}}, \theta_{\text{Gen}}, \theta_{\text{Enc}}), \tag{28}$$

using the next loss function,

$$L = \sum_{B} \log \mathrm{Dis}(s, \mathrm{Enc}(s))_{|s\sim D_{\text{train}}} + \sum_{B} \log\left(1 - \mathrm{Dis}(G(x), x)\right)_{|x\sim P_{\mathcal{X}}}, \tag{29}$$

where $B \ll m$ is the mini-batch size, and we take B samples from D_{train} in the first summation, and B samples from $P_{\mathcal{X}}$ (a random noise distribution in $\mathcal{X}$) in the second summation.

Hence the optimal models are obtained. However, only Enc is necessary to realize the initial aim of this step.

Finally, we obtain the datasets

$$\tilde{D}_{\text{train}} = \mathrm{Enc}(D_{\text{train}}) \tag{30}$$

$$\tilde{D}_{\text{test}} = \mathrm{Enc}(D_{\text{test}}) \tag{31}$$

for subsequent steps. As we have provided an order in the original datasets, we define $\tilde{D}_{\text{train}}^{i,j} = (\mathrm{Enc}(s_i))_j$ as the jth feature of the ith element.

4.3. Min–max normalization

The sets $\tilde{D}_{\text{train}}$ and $\tilde{D}_{\text{test}}$ ((30) and (31), respectively) are subject to unknown distributions based on the severity level of the original raw signals. Therefore, the scale range of extracted features is also unknown.

The scale of features plays a key role in the performance of various machine learning models, such as SVM [41]. Its application avoids the idea that features with a larger scale range are necessarily more important. Another advantage is to avoid numerical instability due to high values in the calculus.

With this in mind, we normalize $\tilde{D}_{\text{train}}$ and $\tilde{D}_{\text{test}}$ by a min–max approach, which conserves the data probability distribution and only rescales the features, specifically by linearly rescaling to the range [0, 1]. Let $\tilde{x}^j = \tilde{D}^{:,j}_{\text{train}}$ be the jth feature of the examples in the training dataset. The normalization parameters are computed as

$$\tilde{x}^j_{\text{min}} = \min(\tilde{x}^j) \tag{32}$$

$$\tilde{x}^j_{\text{max}} = \max(\tilde{x}^j), \tag{33}$$

where the *min* and *max* functions respectively compute the minimum and maximum of the input vector. These parameters are used to rescale the features in $\tilde{D}_{\text{train}}$ and $\tilde{D}_{\text{test}}$ according to the following equations:

$$\bar{D}^{:,j}_{\text{train}} = \frac{\tilde{D}^{:,j}_{\text{train}} - \tilde{x}^j_{\text{min}}}{\tilde{x}^j_{\text{max}} - \tilde{x}^j_{\text{min}}} \tag{34}$$

$$\bar{D}^{:,j}_{\text{test}} = \frac{\tilde{D}^{:,j}_{\text{test}} - \tilde{x}^j_{\text{min}}}{\tilde{x}^j_{\text{max}} - \tilde{x}^j_{\text{min}}}, \tag{35}$$

where $\bar{D}_{\text{train}}$ and $\bar{D}_{test}$ are the normalized datasets for respectively training and testing the model.

As the normalization parameters are obtained only for $\tilde{D}_{\text{train}}$, its range is guaranteed. However, the test dataset could have some features outside [0, 1] that affect its use in OCSVM with additive χ^2 and χ^2 kernels, as previously stated. Then, for configurations with these kernels, elements from each dataset are squared after normalization, i.e., $\bar{D}_{\text{train}}$ and $\bar{D}_{\text{test}}$ are replaced by $\bar{D}^2_{\text{train}}$ and $\bar{D}^2_{\text{test}}$, respectively.

4.4. OCSVM-based modeling

An OCSVM model is built for each evaluated kernel and hyperparameter configuration. We must decide whether they should be optimized or configured with default values. For γ-dependent kernels, we have four options (depending on whether γ and ν take default values or are optimized), while only default values of ν and optimized ν are available for γ-independent models.

When required, γ optimization is performed by the DFN method, which was extended to other kernels in Section 3.1. Then ν can be optimized by Alg. 1. Finally, OCSVM is obtained by solving the optimization problem (12) [14]. The procedure used to build each OCSVM model is summarized in Alg. 2.

```
Data: $\bar{D}_{\text{train}}$, kernel, optimize $\gamma$?, optimize $\nu$?
Result: OCSVM model
if does kernel require $\gamma$? then
    if optimize $\gamma$? then
        Optimize $\gamma$ (Sec: 3.1);
    else
        Assign default value to $\gamma$;
    end
end
if optimize $\nu$? then
    Optimize $\nu$ (Alg: 1);
else
    Assign default value to $\nu$;
end
Build OCSVM model;
Return: OCSVM model;
```

Algorithm 2: OCSVM model building procedure.

After obtaining α and ρ by respectively solving (12) and (14), we can apply OCSVM to new examples, and fault detection can be performed using (13).

For fault severity discrimination, we propose that the perpendicular distances from a set of examples to the OCSVM hyperplane can discriminate between their fault severity levels. This distance is defined as

$$\mathrm{dh}(x) = \sum_{\alpha_i \neq 0} \alpha_i k(x_i, x) - \rho, \tag{36}$$

where x is the position of the example for which is desired to calculate the distance from the hyperplane. Contrary to search severity patterns in a limited set of features obtained by BiGAN-based modeling, we characterize patterns with a single metric (L_2 norm) in a new, possibly infinite feature space, such as those obtained using *RBF*, Laplacian, or χ^2 kernels.

4.5. Histograms and performance evaluation

To evaluate the fault severity discrimination performance of different model configurations, for each example in $\bar{D}_{\text{test}}$, we obtain the distances to the hyperplane, i.e., if $x_{c_i,j}$ is the jth feature vector under the ith severity level in $\bar{D}_{\text{test}}$, then the multiset distance is built from

$$(DH)_{c_i,j} = \mathrm{dh}(x_{c_i,j}),\ \forall x_{c_i,j} \in \bar{D}_{\text{test}}. \tag{37}$$

Let $h \in \mathbb{R}$ be a severity-dependent random variable defining the distance to the hyperplane (positive over the hyperplane, negative under it). Let $p(h|c = c_i),\ 1 \leq i \leq K$, be the hyperplane distance distribution family conditioned in fault severity. We propose the severity pairwise error *spe* as the mixed area between the distributions of h under the c_i and c_j severity levels, i.e.,

$$spe_{i,j} = \int_{-\infty}^{\infty} \min(p(h|c_i), p(h|c_j))\, dh,\ i \neq j. \tag{38}$$

The $p(h|c)$ distributions are analytically unknown, but an estimated severity pairwise error, $\hat{spe}$, can be proposed as follows. With a stratified set of examples per severity level, the multiset *DH* can be represented by a matrix of dimension $K \times \frac{M}{K}$, where rows and columns respectively represent severity levels and evaluated test examples. Furthermore, since each example x in the dataset is a transformation applied to a randomly sampled time series s, the ith row of DH is the result of sampling the $p(h|c = c_i)$ distribution.

Consequently, the estimated severity pairwise error can be defined by

$$\hat{spe}_{i,j} = \begin{cases} \sum_{bin=1}^{Bins} \min(Hist_{c_i}(h_{bin}), Hist_{c_j}(h_{bin})), & \text{if } i \neq j \\ 0, & \text{otherwise }, \end{cases} \tag{39}$$

where $Hist_{c_i}(h_{bin})$ is the *bin*-th value of the computed histogram of the ith row of DH, and the estimated severity pairwise error is a $K \times K$ triangular matrix whose (i, j) element shows the common area of two histograms representing a pair of fault severity levels.

In the same way, the estimated error per severity level (sle_i), representing the performance of the model at each severity level, and average severity error (*ase*), as an overall performance metric, are respectively computed as

$$sle_i = \frac{1}{K-1} \sum_{j \neq i} spe_{i,j} \tag{40}$$

$$ase = \frac{1}{K} \sum_{i=1}^{K} sle_i. \tag{41}$$

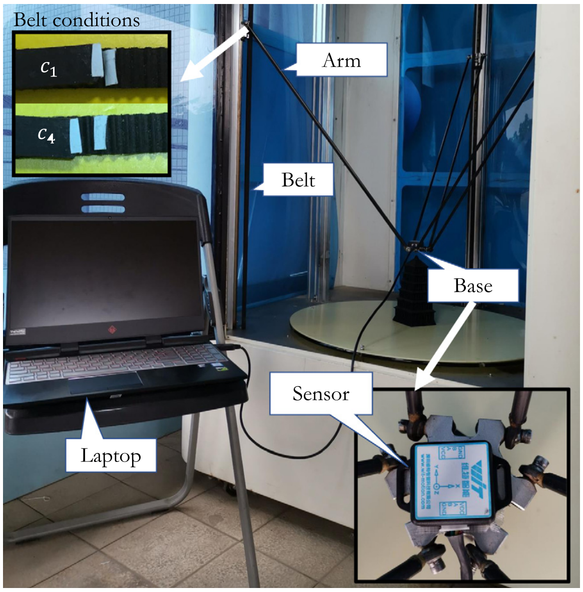


Fig. 2. 3D printer experimental setup.

5. Experiments

The testbed for fault severity discrimination in 3D printers and the experiments to compare OCSVM models are detailed.

5.1. Testbed

Data acquisition was performed on a 3D printer test platform developed for this work (Fig. 2). It was composed of an SLD-BL600-6 3D printer with a belt-driven mechanism in a delta kinematic configuration, with three stepper motors moving the belts to control its three degrees of freedom, and a joint bearing in the terminal points of each arm to obtain free rotational movement.

Magnetic field signals were selected as the information source for the severity discrimination task. They were acquired by a WIT MEMS BWT901 sensor with 14− bit resolution, sensitivity of 0.6 uT, and sampling frequency of 100 samples/s.

The selected fault case was synchronous belt degradation. This type of fault appears due to an extended working period. It consists of the belt stretching due to looseness in the rigidity of the belt material. Thus additional vibrations begin to occur when the machine experiences high-velocity changes in areas with sharp corners. This fault reduces the quality of the final product. To identify the severity level of this fault is relevant to estimate the degradation of product quality and determine whether it is acceptable. To simulate degradation, the effective tension of one synchronous belt was decreased by 0 mm (optimal tension), 1 mm, 2 mm, and 3 mm, corresponding to severity levels c_1, c_2, c_3, and c_4, respectively.

For these severity levels, three magnetic field signals were acquired for 324 s (32 400 samples at 100 samples/s). Each represented 20 circular patterns of 75 mm radius traveled by the printer head. Fig. 3 shows an extract of the signals for the four severity levels where distinguishable patterns cannot be observed for classification of different conditions.

5.2. Experimental setup

The number of signals originally captured rendered us unable to tune the OCSVM parameters due to the curse of dimensionality. To address this issue, the signals were divided into sub-signals of 1620 samples, each containing one circular pattern. Therefore, with a sliding window of 10 samples, three groups of 3078 sub-signals were obtained.

A three-fold cross-validation strategy was configured with these groups to minimize the risk of bias in the results. The comparison method was repeatedly applied with the training and testing sets D_{train} and D_{test} respectively containing two and one of the previous groups. As stated before, only signals in normal condition were added to D_{train}.

Let T and l be the input signal length and referred layer, respectively, in the 1DCNNs [42]. Then the number of layers (NL) and number of output channels in the lth layer (K_l) were set according to the equations

$$NL = \lfloor \log_2 T \rfloor \tag{42}$$

$$K_l = 2^l,\ \forall\ l = 1, \ldots, NL - 1, \tag{43}$$

resulting in 10 layer networks. As required for adversarial training, one output was set for Dis. In the same way, 100 outputs were assigned to the Enc model, i.e., $d = 100$.

The 1DDNN architecture (Gen model) also has 10 layers to represent the inverse function on Enc. For Gen, the number of output channels of each l-layer (KG_l) was

$$KG_l = 2^{NL-l},\ \forall\ l = 1, \ldots, NL. \tag{44}$$

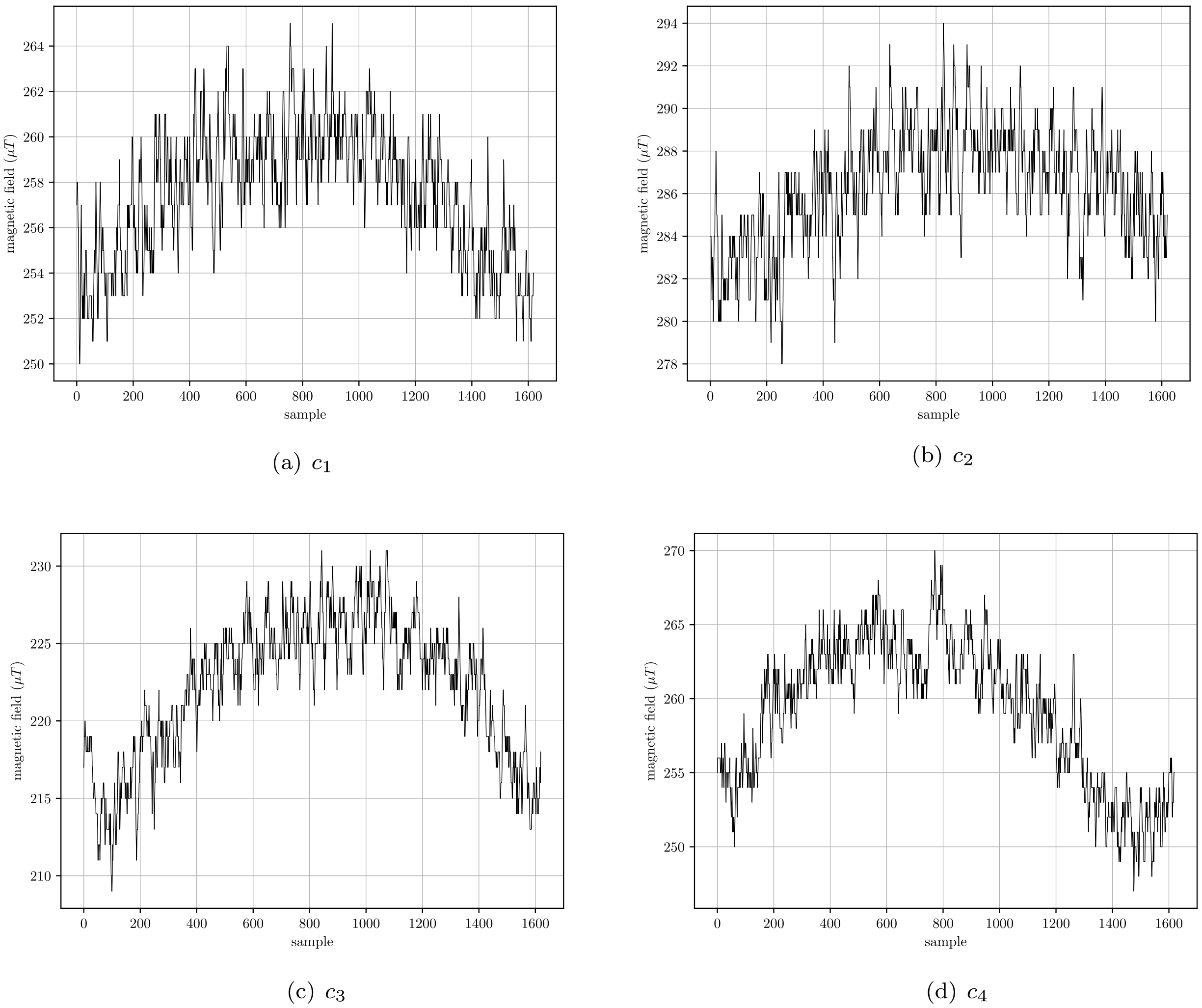
The experiments were to compare OCSVM models with optimized and default hyperparameters, as well as various kernels. For the experiments with default values, $\gamma = 1/d = 0.01$, and $\nu = 0.1$, considering a fixed 10% of anomaly samples in the training set [43]. The γ optimization was performed by the DFN method extended in this work. The positive–negative distance approach and our proposal were compared regarding the ν optimization.

6. Results and analysis

To analyze the characteristics of the input data for OCSVM models, Fig. 4 presents 2D projections of the BiGAN-based feature set with t-SNE and PCA techniques. It shows similar data distributions in both projections: two adjacent groups for c_2, two adjacent groups for c_3, and one group for c_1 adjacent to one c_3 group, although a difference is noticeable between the closeness of c_4 with c_1 shown in PCA compared with t-SNE. The closeness of $c3$ and c_4 to c_1 highlights the complexity of the detection task with OCSVM models. In addition, multiple groups within the same severity condition, such as from c_2 and c_3, represent a multivariable multimodal distribution, which greatly increases the difficulty of severity discrimination by assigning different mean distances to the same severity condition. In Fig. 4(b), it is appreciated that BiGAN-based feature extraction does not guarantee a separating distance between the groups according to the fault severity degree. Thus, the consequence is a non-monotonic dh function regarding to the severity degree, as is shown in Figs. 5 and 6.

The comparisons between kernels, and of methods to optimize γ and ν, are shown in Tables 1, 2, 3, and 4, where the column "optimize ν" shows the method used to optimize ν, and can take three values:

1. distances: optimization using positive–negative distance method;
2. histograms: optimization using the proposal shown in Section 3.2;
3. none: a default value of $\nu = 0.1$.



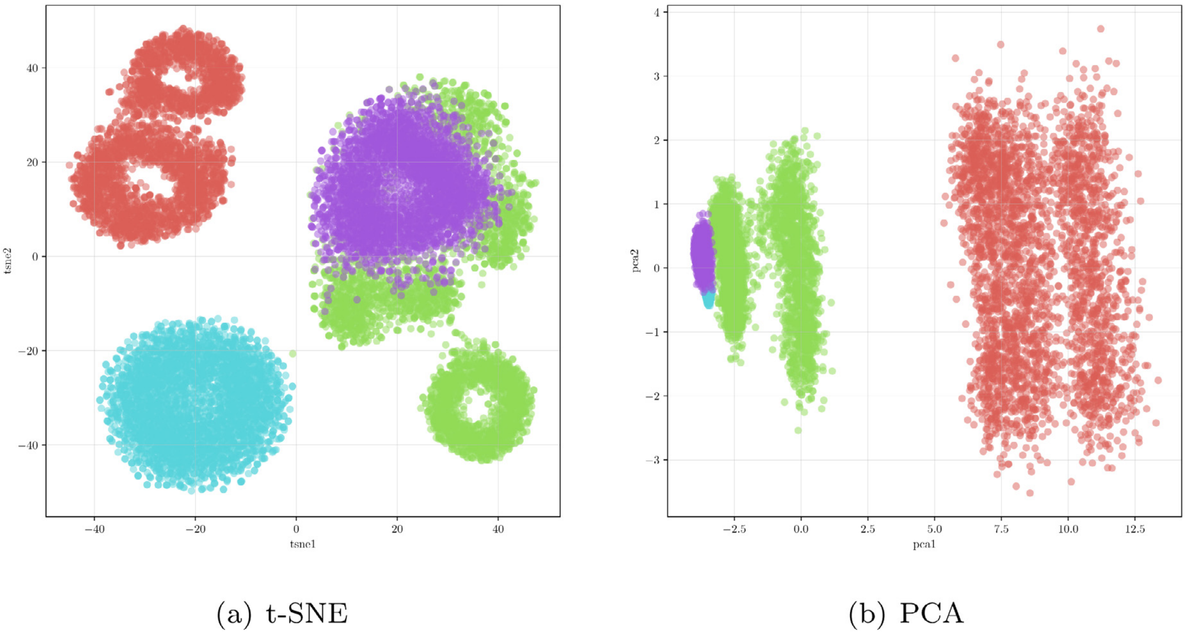
(a) c_1

(b) c_2

(c) c_3

(d) c_4

Fig. 3. Magnetic signals of healthy condition and fault in belts.



(a) t-SNE

(b) PCA

Fig. 4. 2D projections of 100-dimensional 3D printer dataset. Examples in c_1, c_2, c_3, and c_4 severity conditions are drawn respectively in violet, red, green, and blue. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 1
Performance of the models for fault detection and fault severity discrimination using nonparametric kernels (independent of γ).

Kernel	Optimize ν	ν	FNR (%)	FPR (%)	*ase* (%)
Additive χ^2	Distances	0.95	87.20	0.00	0.56
Additive χ^2	Histograms	0.15	0.36	6.39	0.75
Additive χ^2	None	0.10	0.06	21.96	0.83
Cosine	Distances	0.95	87.98	0.00	0.50
Cosine	Histograms	0.20	0.71	2.86	0.69
Cosine	None	0.10	0.13	27.30	0.86
Linear	Distances	0.95	94.61	0.24	28.10
Linear	Histograms	0.40	27.31	35.12	29.71
Linear	None	0.10	3.41	42.65	32.35

The column "optimize γ" indicates whether γ is optimized by the extended DFN method shown in Section 3.1. The FNR and FPR columns list the false negative (type II error) and false positive (type I error) rates, respectively, as evaluation metrics in the fault detection task performed by the OCSVM model. FPR is the ratio between the number of faulty samples classified as normal ones and the total number of faulty samples, and FNR is the ratio between the number of normal samples classified as faulty ones and the total number of normal samples. Finally, the proposed average severity error, *ase*, is presented as a performance metric of the OCSVM model in the severity discrimination task.

6.1. Nonparametric kernels

Table 1 presents the fault detection (FNR and FPR) and fault severity discrimination (*ase*) performance for the nonparametric kernels. FNR ranks additive χ^2, cosine, and linear kernel with a default value of ν in the first, second, and third place, respectively. This ranking corresponds to the lower value for ν. Using FPR results in the same ranking order but using the positive–negative distance optimization method obtains the higher ν value. However, the counterpart (FPR or FNR) was substantially increased in both cases without reaching an equilibrium.

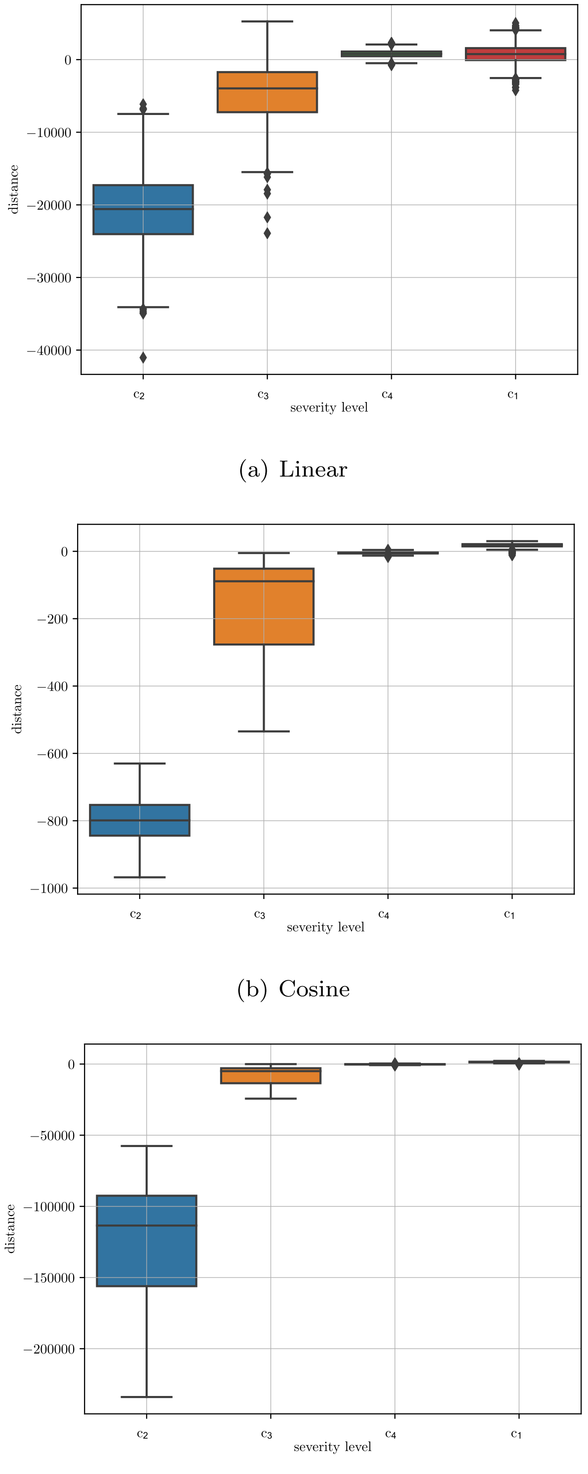
From these results, we confirm that these models are useless for the detection task. In addition, the models obtained with the histogram-based optimized ν improve both the FPR and FNR. This effect is noticeable for cosine and additive χ^2 kernels, with better performance for the first.

Fig. 5 compares the distance distributions for different severity conditions using nonparametric kernels with the proposed histogram-based method. The overlapping of c_1 with distances lower than 0 produces an increment of FNR, and c_2-c_4 distances bigger than 0 produce an increment of FPR.

6.2. Parametric kernels

Models using the RBF kernel are compared in Table 2. As with nonparametric kernels, FNR is reduced with the lower fixed ν, and FPR is reduced using the positive–negative method. Both metrics decrease (i.e., performance improves) when the histogram-based method is applied. In addition, the optimization of γ produces a slight increase in FNR (slightly worse detection performance) and a consistent decrease of FPR (big performance improvement). In this sense, a large difference is appreciated between the fixed γ (0.01) and the optimal γ (0.292793). Something similar can be observed for fixed ν and optimal ν.

The optimization of γ also improves a model's performance in the severity discrimination task, i.e., a reduced *ase* can be observed for each of the ν optimization methods. Summarizing the results in detection and severity discrimination, the evident winner is the model with optimized γ and ν optimized with the histogram-based method.



(a) Linear

(b) Cosine

(c) Additive χ^2

Fig. 5. Distance distribution of the best nonparametric kernels for each severity condition.

Similar results are shown for the Laplacian and χ^2 kernels in Tables 3 and 4, respectively. The general results in detection and severity discrimination using χ^2 are slightly worse than those with RBF. The poorest performance is obtained with Laplacian comparing the parametric kernels. However, this highlights the improvement obtained with γ optimization, showing that the

Table 2
Performance of the models for fault detection and fault severity discrimination using RBF kernel.

Optimize γ	γ	Optimize ν	ν	FNR (%)	FPR (%)	*ase* (%)
No	0.010000	Distances	0.95	87.46	0.00	0.54
Yes	0.292793	Distances	0.95	87.43	0.00	0.51
No	0.010000	Histograms	0.15	0.36	6.25	0.65
Yes	0.292793	Histograms	0.25	1.27	0.03	0.55
No	0.010000	None	0.10	0.06	21.45	0.70
Yes	0.292793	None	0.10	0.10	18.14	0.62

Table 3
Performance of the models for fault detection and fault severity discrimination using Laplacian kernel.

Optimize γ	γ	Optimize ν	ν	FNR (%)	FPR (%)	*ase* (%)
No	0.010000	Distances	0.95	87.63	0.00	0.58
Yes	0.069891	Distances	0.95	87.46	0.00	0.58
No	0.010000	Histograms	0.25	1.40	0.17	0.68
Yes	0.069891	Histograms	0.25	1.40	0.12	0.65
No	0.010000	None	0.10	0.06	25.00	0.82
Yes	0.069891	None	0.10	0.10	23.18	0.76

Table 4
Performance of the models for fault detection and fault severity discrimination using χ^2 kernel.

Optimize γ	γ	Optimize ν	ν	FNR (%)	FPR (%)	*ase* (%)
No	0.010000	Distances	0.95	87.24	0.00	0.57
Yes	0.168679	Distances	0.95	87.33	0.00	0.57
No	0.010000	Histograms	0.15	0.36	6.28	0.75
Yes	0.168679	Histograms	0.25	1.36	0.03	0.66
No	0.010000	None	0.10	0.06	21.71	0.82
Yes	0.168679	None	0.10	0.06	18.74	0.75

DFN method is successfully extended to other parametric kernels different to RBF.

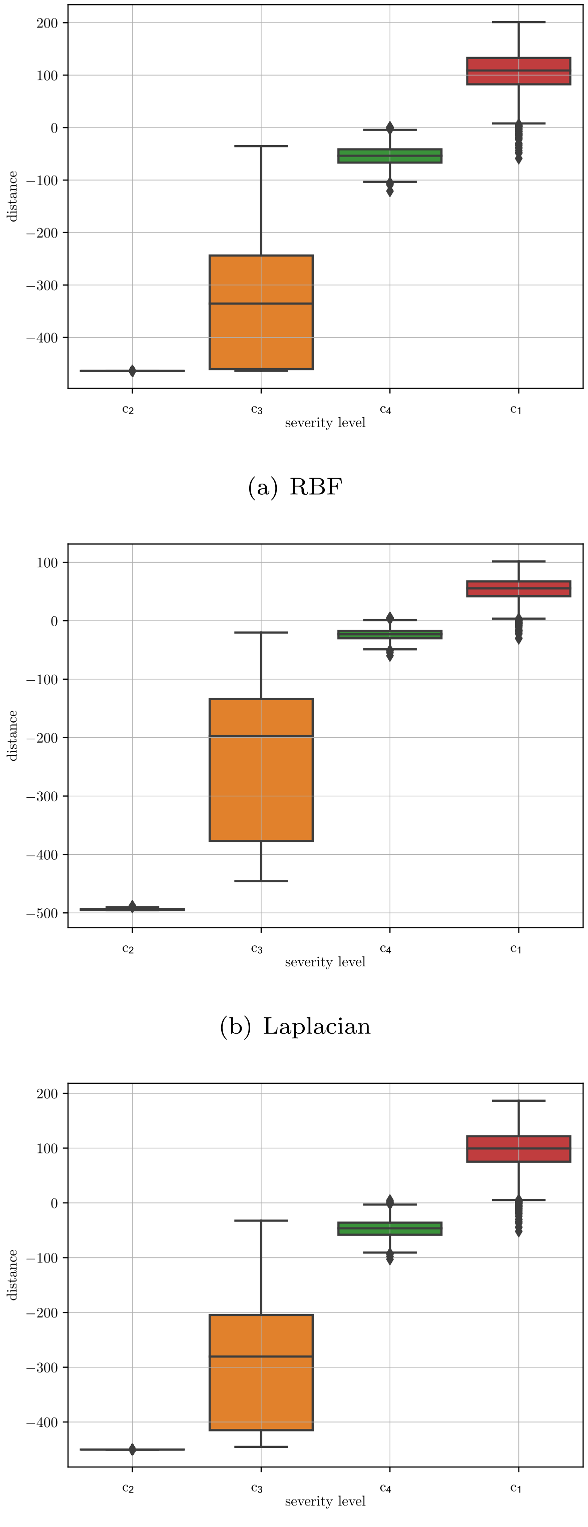
With χ^2, a dependence between γ and ν is evident using histogram-based optimization. The optimal ν is 0.15 without optimized γ, and 0.25 with optimized γ. This difference in ν has a considerable impact on detection performance, especially in the FPR value.

Fig. 6 shows hyperplane distance distributions as an indicator of goodness in the severity discrimination task. Contrary to what was shown in Fig. 5 for the nonparametric kernels, samples from normal condition are projected away from other severity levels in the kernel space, and their distance to the hyperplane is positively increased. As a result, RBF and χ^2 have the best performance due to their mean distance from normal condition to the other severity levels, larger than with the Laplacian kernel.

7. Conclusions

We have proposed OCSVM as a distance-based metric for fault detection and fault severity discrimination, and we have tested it for 3D printers. A set of features was extracted from normal condition signals using a GAN-based approach. An optimized OCSVM model was obtained by tuning the kernel and OCSVM hyperparameters with the proposed extended DFN and histogram-based methods, respectively. The resulting models were evaluated for fault detection, which is typical for OCSVM, and an evaluation in the context of fault severity discrimination was performed as a novel application.

According to the experimental results, γ optimization improves not only the fault detection performance by decreasing the FPR, but fault severity discrimination. Furthermore, this improvement was obtained for all parametric kernels, hence it can be concluded that the DFN method has been successfully extended to other kernels. In the same sense, the correct ν configuration was important to fault detection performance. In effect, the proposed histogram-based optimization approach has allowed us to obtain optimal ν values for parametric and nonparametric kernels, which decreases both FNR and FPR without sacrificing severity discrimination performance.



(a) RBF

(b) Laplacian

(c) χ^2

Fig. 6. Distance distribution of the best parametric kernels for each severity condition.

Machine learning approaches try to reveal the underlying knowledge in the data, and the best case is the availability of all the expected operation conditions. This is not a practical and real scenario, and approaches improving this dependence is a continuous challenge. Our approach is developed by considering that data from only normal state under some operational conditions is available, and the data distribution is representative of such normal case related to the case study. The sensitivity of our approach regarding the use of other test signals different from the circular path, and different operating conditions, will be analyzed in further works.

We expect that this extension of DFN, along with the histogram-based method, will enhance the practical application of OCSVM for fault detection and severity discrimination tasks in different contexts.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] Dhiman C, Vishwakarma DK. A review of state-of-the-art techniques for abnormal human activity recognition. Eng Appl Artif Intell 2019;77:21–45.

[2] Ahmed M, Naser Mahmood A, Hu J. A survey of network anomaly detection techniques. J Netw Comput Appl 2016;60:19–31.

[3] Nissim N, Cohen A, Glezer C, Elovici Y. Detection of malicious PDF files and directions for enhancements: A state-of-the art survey. Comput Secur 2015;48:246–66.

[4] Chen K, Huang C, He J. Fault detection, classification and location for transmission lines and distribution systems: a review on the methods. High Volt 2016;1(1):25–33.

[5] Du X. Fault detection using bispectral features and one-class classifiers. J Process Control 2019;83:1–10.

[6] Zhang Z-H, Li S, Yan H, Fan Q-Y. Sliding mode switching observer-based actuator fault detection and isolation for a class of uncertain systems. Nonlinear Anal Hybrid Syst 2019;33:322–35.

[7] Xu Y, Sun Y, Wan J, Liu X, Song Z. Industrial big data for fault diagnosis: Taxonomy, review, and applications. IEEE Access 2017;5:17368–80.

[8] Zheng J, Wang H, Song Z, Ge Z. Ensemble semi-supervised Fisher discriminant analysis model for fault classification in industrial processes. ISA Trans 2019;92:109–17.

[9] Khan SS, Madden MG. One-class classification: taxonomy of study and review of techniques. Knowl Eng Rev 2014;29(3):345–74.

[10] Yin Z, Hou J. Recent advances on SVM based fault diagnosis and process monitoring in complicated industrial processes. Neurocomputing 2016;174:643–50.

[11] Fisher WD, Camp TK, Krzhizhanovskaya VV. Anomaly detection in earth dam and levee passive seismic data using support vector machines and automatic feature selection. J Comput Sci 2017;20:143–53.

[12] Long J, Zhang S, Li C. Evolving deep echo state networks for intelligent fault diagnosis. IEEE Trans Ind Inform 2020;16(7):4928–37.

[13] Cerrada M, Sánchez R-V, Cabrera D. A semi-supervised approach based on evolving clusters for discovering unknown abnormal condition patterns in gearboxes. J Intell Fuzzy Systems 2018;34(6):3581–93.

[14] Schölkopf B, Platt JC, Shawe-Taylor J, Smola AJ, Williamson RC. Estimating the support of a high-dimensional distribution. Neural Comput 2001;13(7):1443–71.

[15] Moosavi S, Kazemi A, Akbari H. A comparison of various open-circuit fault detection methods in the IGBT-based DC/AC inverter used in electric vehicle. Eng Fail Anal 2019;96:223–35.

[16] Tang X, Zeng W, Shi Y, Zhao L. Brain activation detection by modified neighborhood one-class SVM on fMRI data. Biomed Signal Process Control 2018;39:448–58.

[17] Xi P-P, Zhao Y-P, Wang P-X, Li Z-Q, Pan Y-T, Song F-Q. Least squares support vector machine for class imbalance learning and their applications to fault detection of aircraft engine. Aerosp Sci Technol 2019;84:56–74.

[18] Liu J, Li Y-F, Zio E. A SVM framework for fault detection of the braking system in a high speed train. Mech Syst Signal Process 2017;87:401–9.

[19] Saari J, Strömbergsson D, Lundberg J, Thomson A. Detection and identification of windmill bearing faults using a one-class support vector machine (SVM). Measurement 2019;137:287–301.

[20] Martínez-Rego D, Fontenla-Romero O, Alonso-Betanzos A, Principe JC. Fault detection via recurrence time statistics and one-class classification. Pattern Recognit Lett 2016;84:8–14.

[21] Every PMV, Rodriguez M, Jones CB, Mammoli AA, Martínez-Ramón M. Advanced detection of HVAC faults using unsupervised SVM novelty detection and Gaussian process models. Energy Build 2017;149:216–24.

[22] Cai L, Tian X, Zhang H. Process fault detection method based on time structure independent component analysis and one-class support vector machine. IFAC-PapersOnLine 2015;48(21):1198–203.

[23] Xiao Y, Wang H, Xu W, Zhou J. Robust one-class SVM for fault detection. Chem Intell Lab Syst 2016;151:15–25.

[24] Zeng M, Yang Y, Luo S, Cheng J. One-class classification based on the convex hull for bearing fault detection. Mech Syst Signal Process 2016;81:274–93.

[25] Yin S, Zhu X, Jing C. Fault detection based on a robust one class support vector machine. Neurocomputing 2014;145:263–8.

[26] Beghi A, Cecchinato L, Corazzol C, Rampazzo M, Simmini F, Susto G. A one-class SVM based tool for machine learning novelty detection in HVAC chiller systems. IFAC Proc Vol 2014;47(3):1953–8.

[27] Yan K, Ji Z, Shen W. Online fault detection methods for chillers combining extended kalman filter and recursive one-class SVM. Neurocomputing 2017;228:205–12, Advanced Intelligent Computing: Theory and Applications.

[28] Evangelista PF, Embrechts MJ, Szymanski BK. Some properties of the Gaussian kernel for one class learning. In: Lecture Notes in Computer Science. Springer Berlin Heidelberg; 2007, p. 269–78. http://dx.doi.org/10.1007/978-3-540-74690-4_28.

[29] Wang S, Yu J, Lapira E, Lee J. A modified support vector data description based novelty detection approach for machinery components. Appl Soft Comput 2013;13(2):1193–205.

[30] Khazai S, Homayouni S, Safari A, Mojaradi B. Anomaly detection in hyperspectral images based on an adaptive support vector method. IEEE Geosci Remote Sens Lett 2011;8(4):646–50.

[31] Vamsi I, Sabareesh G, Penumakala P. Comparison of condition monitoring techniques in assessing fault severity for a wind turbine gearbox under non-stationary loading. Mech Syst Signal Process 2019;124:1–20.

[32] Cabrera D, Sancho F, Li C, Cerrada M, Sánchez R-V, Pacheco F, de Oliveira JV. Automatic feature extraction of time-series applied to fault severity assessment of helical gearbox in stationary and non-stationary speed operation. Appl Soft Comput 2017;58:53–64.

[33] Cerrada M, Li C, Sánchez R-V, Pacheco F, Cabrera D, de Oliveira JV. A fuzzy transition based approach for fault severity prediction in helical gearboxes. Fuzzy Sets and Systems 2018;337:52–73, Theme: Applications.

[34] Cerrada M, Sánchez R-V, Li C, Pacheco F, Cabrera D, Valente de Oliveira J, Vásquez RE. A review on data-driven fault severity assessment in rolling bearings. Mech Syst Signal Process 2018;99:169–96.

[35] Cortes C, Vapnik V. Support-vector networks. Mach Learn 1995;20(3):273–97.

[36] Brezhneva OA, Tret'yakov AA, Wright SE. A simple and elementary proof of the karush–kuhn–tucker theorem for inequality-constrained optimization. Optim Lett 2008;3(1):7–10.

[37] Xiao Y, Wang H, Zhang L, Xu W. Two methods of selecting Gaussian kernel parameters for one-class SVM and their application to fault detection. Knowl-Based Syst 2014;59:75–84.

[38] Ratsch G, Mika S, Scholkopf B, Muller K-R. Constructing boosting algorithms from SVMs: an application to one-class classification. IEEE Trans Pattern Anal Mach Intell 2002;24(9):1184–99.

[39] Li C, Cabrera D, Sancho F, Sanchez R-V, Cerrada M, de Oliveira JV. One-shot fault diagnosis of 3D printers through improved feature space learning. IEEE Trans Ind Electron 2020;1.

[40] Sánchez R-V, Lucero P, Vásquez RE, Cerrada M, Cabrera D. A comparative feature analysis for gear pitting level classification by using acoustic emission, vibration and current signals. IFAC-PapersOnLine 2018;51(24):346–52, 10th IFAC Symposium on Fault Detection, Supervision and Safety for Technical Processes SAFEPROCESS 2018.

[41] Hsu C-W, Chang C-C, Lin C-J. A Practical Guide to Support Vector Classification. Technical Report, Department of Computer Science, National Taiwan University; 2003, URL http://www.csie.ntu.edu.tw/~cjlin/papers.html.

[42] Li C, Cabrera D, Sancho F, Sánchez R-V, Cerrada M, Long J, de Oliveira JV. Fusing convolutional generative adversarial encoders for 3D printer fault detection with only normal condition signals. Mech Syst Signal Process 2021;147:107108.

[43] Hempstalk K, Frank E, Witten IH. One-class classification by combining density and class probability estimation. In: Machine Learning and Knowledge Discovery in Databases. Springer Berlin Heidelberg; 2008, p. 505–19. http://dx.doi.org/10.1007/978-3-540-87479-9_51.