A New SVDD Approach to Reliable and Explainable AI

Alberto Carlevaro, *University of Genoa, 16145, Genoa, Italy*
Maurizio Mongelli, *Italian National Research Council (CNR), Via De Marini, 16149, Genova, Italy*

Safety engineering and artificial intelligence are two fields that still need investigation on their reciprocal interactions. Safety should be guaranteed when autonomous decision may lead to risk for the environment and the human. The present work addresses how support vector data description (SVDD) can be redesigned to detect safety regions in a cyber-physical system with zero statistical error. Rule-based knowledge extraction is also presented, to let the SVDD be understandable. Two applications are considered for performance evaluation: domain name server tunneling detection and region of attraction estimation of dynamic systems. Results demonstrate how the new SVDD and its intelligible representation are both suitable in designing safety regions, still maximizing the space of the working conditions.

The study proposed in this article follows the recent trend dedicated to identifying and handling assurance under uncertainties in artificial intelligence (AI) systems.[29] It falls in the category of improving reliability of prediction confidence. The topic remains a significant challenge in machine learning (ML), as learning algorithms proliferate into difficult real-world pattern recognition applications. The intrinsic statistical error introduced by any ML algorithm may lead to criticism by safety engineers. The topic has received a great interest from industry,[31] in particular in the automotive[33] and avionics[8] sectors. In this perspective, the conformal predictions framework[6] studies methodologies to associate reliable measures of confidence with pattern recognition settings including classification, regression, and clustering. The proposed approach follows this direction, by identifying methods to circumvent data-driven safety envelopes with statistical zero errors. We show how this assurance may limit considerably the size of the safety envelope (e.g., providing collision avoidance by drastically reducing speed of vehicles) and focus on how to find a good balance between the assurance and the safety space.

1541-1672 © 2021 IEEE
Digital Object Identifier 10.1109/MIS.2021.3123669
Date of publication 29 October 2021; date of current version 20 May 2022.

We concentrated our work on a specific ML methods, the support vector data description (SVDD), which by (its) definition is particularly suitable to define safety envelops (see the "Support Vector Data Description" section). To it we have added intelligible models for knowledge extraction with rules: intelligibility means that the model is easily understandable, e.g., when it is expressed by Boolean rules. Decision trees (DTs) are typically used toward this aim. The comprehension of neural network models (and of the largest part of the other ML techniques) reveals to be a hard task (see, e.g., Section 4 of Mongelli *et al.*'s work[14]). Together with DT, we use logic learning machine (LLM), which may show more versatility in rule generation and classification precision.

Our work takes a step forward in these areas due to the following reasons.

- Safety regions are tuned on the basis of the radius of the SVDD hypersphere.
- Simple rule extraction method from SVDD is studied in comparison with LLM and DT.

The rest of this article is organized as follows. First, a detailed introduction of SVDD and negative SVDD is introduced, also focusing on how to choose the best model parameters (see the "Autonomous Detection of SVDD Parameters With RBF Kernel" section) and how to handle large datasets

(see the "Fast Training SVDD" section). Then, the "Rules Extraction" section is devoted to rule extraction: LLM and DT are presented, together with rule extraction from SVDD. An application example is proposed in the "Applications" section. The "Remarks" sections presents some remarks. The "Conclusion and Future Work" section concludes this article.

SUPPORT VECTOR DATA DESCRIPTION

Characterizing a dataset in a complete and exhaustive way is an essential preliminary step for any action you want to perform on it. Having a good description of a dataset means being able to easily understand if a new observation can contribute to the information brought by the rest of the data or be totally irrelevant. The task of the data domain description is precisely to identify a region, a border, in which to enclose a certain type of information in the most precise possible way, i.e., not adding misinformation or empty spaces. This idea is realized mathematically by a circumference (a sphere, a hypersphere depending on the size of the data space) that encloses as many points with as little area (volume) as possible. Indeed, SVDD can be used also to perform a classification of a specific class of target objects, i.e., it is possible to identify a region (a closed boundary) in which objects that should be rejected are not allowed.

This section is organized as follows: SVDD is introduced as in Tax and Duin's work,[34] focusing first on the normal description and then on the description with negative examples.[35] Then, we will focus on two proposed algorithms for solving two problems involving SVDD: fast training of large datasets[7] and autonomous detection of SVDD parameters.[37] Finally, the last section is devoted to two original methods for finding zero false positive rate (FPR) regions with SVDD.

Theory

Let $\{\mathbf{x}_i\}, i = 1, \ldots, N$ with $\mathbf{x}_i \in \mathbb{R}^d$, $d >= 1$, be a training set for which we want to obtain a description. We want to find a sphere (a hypersphere) of radius R and center $\mathbf{a}$ with minimum volume, containing all (or most of) the data objects.

Normal Data Description

For finding the decision boundary that captures the normal instances and at the same time keeps the hypersphere's volume at minimum, it is necessary to solve the following optimization problem:[35]

$$\min_{R,\mathbf{a}} \; F(R,\mathbf{a}) = R^2 \text{ s.t. } ||\mathbf{x}_i - \mathbf{a}||^2 \leq R^2 \quad \forall i. \tag{1}$$

In order to allow the possibility of outliers in the training set, analogously to what happens for the soft-margin SVMs,[1] slack variables $\xi_i \geq 0$ are introduced and the minimization problem changes into that described in Tax and Duin's work[35]

$$\min_{R,\mathbf{a},\xi_i} F(R,\mathbf{a},\xi_i) = R^2 + C\sum_i \xi_i \tag{2}$$

$$\text{s.t.} \begin{cases} ||\mathbf{x}_i - \mathbf{a}||^2 \leq R^2 + \xi_i, & i = 1, \ldots, N \\ \xi_i \geq 0 \end{cases} \tag{3}$$

where the parameter C controls the influence of the slack variables and thereby the tradeoff between the volume and the errors.

The optimization problem is solved by incorporating the constraints (3) into (2) using the method of Lagrange for positive inequality constraints[13]

$$L(R,\mathbf{a},\alpha_i,\gamma_i,\xi_i) = R^2 + C\sum_i \xi_i - \sum_i \alpha_i\left[R^2 + \xi_i - \left(||\mathbf{x_i}||^2 - 2\mathbf{a}\cdot\mathbf{x}_i + ||\mathbf{a}||^2\right)\right] - \sum_i \gamma_i\xi_i \tag{4}$$

with the Lagrange multipliers $\alpha_i \leq 0$ and $\gamma_i \leq 0$. According to Tax and Duin's work,[34] L should be minimized with respect to $R, \mathbf{a}, \xi_i$, and maximized with respect to α_i and γ_i.

Setting partial derivatives of $R, \mathbf{a}$, and ξ_i to zero gives the constraints[11]

$$\frac{\partial L}{\partial R} = 0 : \sum_i \alpha_i = 1, \quad \frac{\partial L}{\partial \mathbf{a}} = 0 : \mathbf{a} = \sum_i \alpha_i \mathbf{x}_i \tag{5}$$

$$\frac{\partial L}{\partial \xi_i} = 0 : C - \alpha_i - \gamma_i = 0 \Rightarrow 0 \leq \alpha_i \leq C \tag{6}$$

and then, substituting (5) into (4) gives the dual problem of (2) and (3)

$$\max_{\alpha_i} \; L = \sum_i \alpha_i(\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_{i,j} \alpha_i\alpha_j(\mathbf{x}_i \cdot \mathbf{x}_j) \tag{7}$$

$$\text{s.t} \begin{cases} \sum_i \alpha_i = 1, \\ 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, N. \end{cases} \tag{8}$$

Maximizing (7) under (8) allows us to determine all α_i and then the parameters $\mathbf{a}$ and ξ_i can be deduced.

A training object $\mathbf{x}_i$ and its corresponding α_i satisfy one of the following conditions:[34,35]

$$||\mathbf{x}_i - \mathbf{a}||^2 < R^2 \Rightarrow \alpha_i = 0 \tag{9}$$

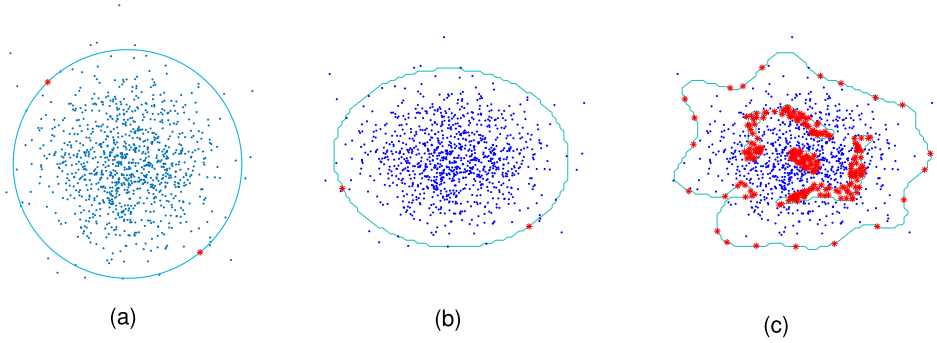


FIGURE 1. SVDD with (a) linear kernel, (b) polynomial kernel, (c) Gaussian kernel, and the respective parameters. The SV (with $\alpha_i < C$) of the description are plotted in red.

$$||\mathbf{x}_i - \mathbf{a}||^2 = R^2 \Rightarrow 0 < \alpha_i < C \tag{10}$$

$$||\mathbf{x}_i - \mathbf{a}||^2 > R^2 \Rightarrow \alpha_i = C. \tag{11}$$

Since $\mathbf{a}$ is a linear combination of the objects with α_i as coefficients, only $\alpha_i > 0$ are needed in the description: this object will therefore be called the *support vectors* of the description (SV). So by definition, R^2 is the distance from the center of the sphere to (any of the SVs on) the boundary, i.e., objects with $0 < \alpha_i < C$. Therefore

$$\begin{aligned} R^2 &= ||\mathbf{x}_k - \mathbf{a}||^2 \\ &= \underbrace{(\mathbf{x}_k \cdot \mathbf{x}_k) - 2\sum_i \alpha_i (\mathbf{x}_k \cdot \mathbf{x}_i) + \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \mathbf{x}_j)}_{T_{\mathbf{a}}(\mathbf{x}_k)} \end{aligned} \tag{12}$$

for any $\mathbf{x}_k \in SV_{<C}$, the set of the SVs have $\alpha_k < C$.

To test a new object $\mathbf{z}$, it is necessary to calculate its distance $T_{\mathbf{a}}(\mathbf{z})$ from the center of the sphere and compare it with R^2

$$\text{sgn}(R^2 - T_{\mathbf{a}}(\mathbf{z})) = \begin{cases} +1 & \text{if } \mathbf{z} \text{ is inside the sphere} \\ -1 & \text{if } \mathbf{z} \text{ is outside the sphere.} \end{cases} \tag{13}$$

As it is common in ML theory,[38] the method can be made more flexible,[34,35] by replacing all the inner products $(\mathbf{x}_i \cdot \mathbf{x}_j)$ with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$ satisfying Mercer's theorem. The data are mapped into a higher dimensional space via a feature map and there the previous spherically classification is computed. The polynomial kernel and the Gaussian kernel are discussed in Tax and Duin's work.[34,35]

An example description by SVDD with different kernel functions for a 2D Gaussian dataset is shown in Figure 1. The 1000 data are generated by a Gaussian distribution with mean [0,0] and variance 1. Figures are handmade drawn using MATLAB and the description bound is shown by a 2D contour plot.

Negative Examples Data Description

When two (or more) classes of data are available and it is necessary to identify a specific one among the others, SVDD can be trained to recognize objects that should be included in the description from those that should be rejected. This task of SVDD can be very useful in real-world applications where, for example, a safety region must be determined (see the "Applications" section).

In the following, the target objects are enumerated by indices i, j and the negative examples by l, m. We assume that target objects are labeled $y_i = 1$ and outlier objects are labeled $y_l = -1$.

In the same way as before, we want to solve this optimization problem

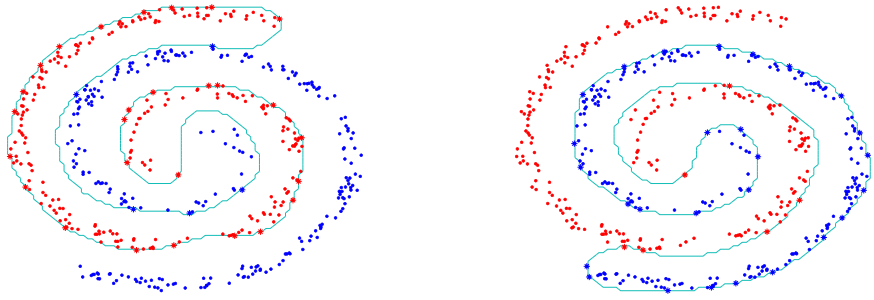
$$\min_{R,\mathbf{a},\xi_i,\xi_l} F(R, \mathbf{a}, \xi_i, \xi_l) = R^2 + C_1 \sum_i \xi_i + C_2 \sum_l \xi_l \tag{14}$$

$$\text{s.t} \begin{cases} ||\mathbf{x}_i - \mathbf{a}||^2 \leq R^2 + \xi_i, \\ ||\mathbf{x}_l - \mathbf{a}||^2 \geq R^2 - \xi_l, \\ \xi_i \geq 0, \quad \xi_l \geq 0 \quad \forall i, l. \end{cases} \tag{15}$$

The constraints are again incorporated in (14) and the Lagrange multipliers $\alpha_i, \alpha_l, \gamma_i, \gamma_l$ are introduced[35]

$$\begin{aligned} L(R, \mathbf{a}, \xi_i, \xi_l, \alpha_i, \alpha_l, \gamma_i, \gamma_l) &= R^2 + C_1 \sum_i \xi_i + C_2 \sum_l \xi_l \\ &- \sum_i \gamma_i \xi_i - \sum_l \gamma_l \xi_l - \sum_i \alpha_i [R^2 + \xi_i - (\mathbf{x}_i - \mathbf{a})^2] \\ &- \sum_l \alpha_l [(\mathbf{x}_l - \mathbf{a})^2 - R^2 + \xi_l] \end{aligned} \tag{16}$$

with $\alpha_i \geq 0, \alpha_l \geq 0, \gamma_i \geq 0, \gamma_l \geq 0$.



(a) (b)

FIGURE 2. Negative SVDD applied to a two-spiral shaped dataset.[24] It is interesting to note that for changing the target objects, it is only necessary to flip the labels. The asterisked points are the SV on the edge, depending on the respective class.

Setting the partial derivatives of L with respect to $R, \mathbf{a}, \xi_i$, and ξ_l to zero gives new constraints[35]

$$\sum_i \alpha_i - \sum_l \alpha_l = 1, \ \mathbf{a} = \sum_i \alpha_i \mathbf{x}_i - \sum_l \alpha_l \mathbf{x}_l \tag{17}$$

$$0 \leq \alpha_i \leq C_1, \ 0 \leq \alpha_l \leq C_2 \quad \forall i, l \tag{18}$$

and substituting (17) in (16), we obtain, similarly to before, the dual problem of (14) and (15):

$$\begin{aligned} \max_{\alpha_i, \alpha_l} \ L = & \sum_i \alpha_i (\mathbf{x}_i \cdot \mathbf{x}_i) - \sum_l \alpha_l (\mathbf{x}_l \cdot \mathbf{x}_l) - \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) \\ & + 2 \sum_{l,j} \alpha_l \alpha_j (\mathbf{x}_l \cdot \mathbf{x}_j) - \sum_{l,m} \alpha_l \alpha_m (\mathbf{x}_l \cdot \mathbf{x}_m) \end{aligned} \tag{19}$$

$$\text{s.t} \begin{cases} \sum_i \alpha_i - \sum_l \alpha_l = 1 \\ 0 \leq \alpha_i \leq C_1 \quad \forall i \\ 0 \leq \alpha_l \leq C_2 \quad \forall l. \end{cases} \tag{20}$$

Again, solving the previous optimization problem allows us to determine α_i and α_l and then we can classify all the dataset objects according to the respective Lagrange coefficient

$$||\mathbf{x}_i - \mathbf{a}||^2 < R^2 \Rightarrow \alpha_i = 0; ||\mathbf{x}_l - \mathbf{a}||^2 < R^2 \Rightarrow \alpha_l = C_2 \tag{21}$$

$$||\mathbf{x}_i - \mathbf{a}||^2 = R^2 \Rightarrow 0 < \alpha_i < C_1 \tag{22}$$

$$||\mathbf{x}_l - \mathbf{a}||^2 = R^2 \Rightarrow 0 < \alpha_l < C_2 \tag{23}$$

$$||\mathbf{x}_i - \mathbf{a}||^2 > R^2 \Rightarrow \alpha_i = C_1; ||\mathbf{x}_l - \mathbf{a}||^2 > R^2 \Rightarrow \alpha_l = 0. \tag{24}$$

Similarly, we test a new point $\mathbf{z}$ based on its distance from the center

$$\begin{aligned} ||\mathbf{z} - \mathbf{a}||^2 = & (\mathbf{z} \cdot \mathbf{z}) - 2 \left(\sum_i \alpha_i (\mathbf{z} \cdot \mathbf{x}_i) - \sum_l \alpha_l (\mathbf{z} \cdot \mathbf{x}_l) \right) \\ & + \sum_{i,j} \alpha_i \alpha_j (\mathbf{x}_i \cdot \mathbf{x}_j) - 2 \sum_{l,j} \alpha_l \alpha_j (\mathbf{x}_l \cdot \mathbf{x}_j) \\ & + \sum_{l,m} \alpha_l \alpha_m (\mathbf{x}_l \cdot \mathbf{x}_m) := T_{\mathbf{a}}(\mathbf{z}) \end{aligned} \tag{25}$$

and we evaluate it compared to the radius squared

$$\text{sgn}(R^2 - T_{\mathbf{a}}(\mathbf{z})) = \begin{cases} +1 & \text{if } \mathbf{z} \text{ is inside the sphere} \\ -1 & \text{if } \mathbf{z} \text{ is outside the sphere} \end{cases} \tag{26}$$

where the radius is calculated as the distance of any SV on the edge ($0 < \alpha_i < C_1, 0 < \alpha_l < C_2$) from the center $\mathbf{a}$

$$R^2 = T_{\mathbf{a}}(\mathbf{x}_k) \text{ for any } \mathbf{x}_k \in SV_{<C_1, <C_2}. \tag{27}$$

Similarly to before, it is possible to replace all the inner products $(\mathbf{x}_i \cdot \mathbf{x}_j)$ with a kernel function $K(\mathbf{x}_i, \mathbf{x}_j)$[34,35,38] to obtain a more flexible description.

An example of negative SVDD is performed in Figure 2: Gaussian kernel with $\sigma = 3$ is used, and the parameters $C1$ and $C2$ are both set to 0.25.

Autonomous Detection of SVDD Parameters With RBF Kernel

Like most ML models, SVDD is massively influenced by the choice of model parameters. It is necessary to find the best tradeoff between error and covering, by choosing suitable C_1 and C_2, and the best kernel parameter σ that avoids overfitting or underfitting issues.

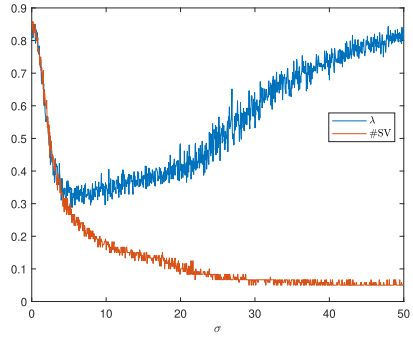


FIGURE 3. For too small or too high values of σ the optimization criterion λ (our metric for the "best error") is high. Also keep in mind the behavior of the SV, which is very similar to the one described in Tax and Duin's work.[34,35]

For this work, we will focus on the RBF kernel since it is well known that it is the kernel function that performs well in application methods.[34]

The method used to find the best model parameters is inspired by Theissler and Dear's work,[37] in which an autonomous detection of the normal SVDD parameters is proposed based only in the training set, since in normal SVDD it is not possible to use cross-validation because only true positives and false negatives can occur during the training. In our work instead, we joined some techniques used in Theissler and Dear's work[37] with the cross-validation method for finding the best C_1, C_2, and σ parameters for negative SVDD.

The regularization parameters C_1 and C_2 are lower bounded by $1/N_1$ and $1/N_2$ respectively, where N_1 is the number of target objects and N_2 the number of negative examples ($N_1 + N_2 = N$).[34,35,37] When in one class of training objects set no errors are expected, we can set $C_i = 1$ ($i = 1, 2$), indicating that all objects of the target class of training set should be accepted ($C_1 = 1$) and all outliers should be rejected ($C_2 = 1$). So the value ranges for C_1 and C_2 are

$$\frac{1}{N_1} \leq C_1 \leq 1, \quad \frac{1}{N_2} \leq C_2 \leq 1. \tag{28}$$

The second parameter to be optimized is the kernel width σ. For high values of σ, the shape of SVDD becomes spherical with the risk of underfitting, while for small values of σ too much objects become SVs and the model is prone to overfitting.

The search for the best parameters is performed by constructing a grid with C_1, C_2, and σ, on which holdout cross-validation is performed. The optimization criterion is chosen according to Theissler and Dear's work,[37] selecting the parameters such that the respective misclassification error e and radius R minimize

$$\lambda = \sqrt{e^2 + |1 - R|^2} \tag{29}$$

for each triple C_1, C_2, and σ in the grid. The idea behind (29) is that minimizing the misclassification error means reducing the number of SVs[34,35] (and so reducing overfitting) while constraining the radius to be close to 1 means choosing small σ[37] (and so reducing underfitting). Then, the balance between these two terms seems the best criterion for finding the best parameters (see Figure 3).

Fast Training SVDD

The curse of dimensionality is a problem that affects many optimization and ML problems, including the SVDD. To overcome this problem, a method based on iterative training of only SV is proposed by Chaudhuri *et al.*[7]

The method iteratively samples from the training dataset with the objective of updating a set of support vectors called the master set of support vectors (SV^*). During each iteration, the method updates SV^* and the corresponding threshold R^2 value and center $\mathbf{a}$. As the threshold value R^2 increases, the volume enclosed by the SV^* increases. The method stops iterating and provides a solution when the threshold value R^2 and the center $\mathbf{a}$ converge. At convergence, the members of the master set of support vectors SV^* characterize the description of the training dataset.

Zero FPR Regions With SVDD

Safety regions research is a well-known task for ML[14–16] and the main focus is to avoid false positives, i.e., including in the safe region unsafe points. In this section, two methods for the research of zero FPR regions are proposed: the first one is simply based on the reduction of the SVDD radius until only safe points are enclosed in the SVDD shape; the second one instead performs successive iterations of the SVDD on the safe region until there are no more negative points.

Radius Reduction

Since, also in the transformed space via feature mapping, the shape of SVDD is a sphere, it is reasonable to think that reducing the volume of the sphere the number of negative points misclassified should reduce. We implemented this simple procedure in MATLAB and we tested it on several datasets (see Figure 4).

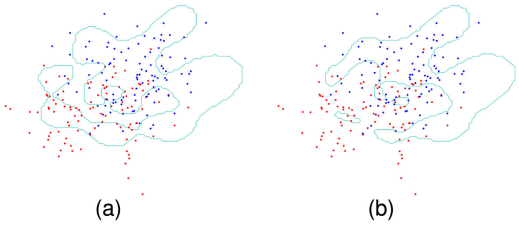


FIGURE 4. Application of Algorithm 1 on a dataset of 400 points sampled from a Gaussian with mean [1,1] and variance 1, 200 target objects and 200 negative examples. The algorithm converged in 12 iterations. (a) FPR = 0.517. (b) FPR = 0.095.

Algorithm 1. RadiusReduction Dataset $\mathcal{X} \times \mathcal{Y}$ is divided in training set $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$ and test set $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$. A threshold ε is set.

1. SVDD-**cross-validation** on $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$
2. $[\mathbf{a}, R^2]$=SVDD($\mathcal{X}_{tr}, \mathcal{Y}_{tr}, C_1, C_2$, param)
3. maxiter=1000;
4. i=1;
5. **while**(i $<$ maxiter)
5.1. $R^2 = R^2 - 10e\text{-}5 * R^2$;
5.2. **Test** SVDD on $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$
5.3. **if**(FPR $< \varepsilon$)
5.3.1. **return** $[\mathbf{a}, R^2]$;
5.4 **end**
6. $i = i + 1$;
7. **end**

Algorithm 2. ZeroFPRSVDD Dataset $\mathcal{X} \times \mathcal{Y}$ is divided in training set $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$ and test set $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$. A threshold ε is set.

1. SVDD-**cross-validation** on $\mathcal{X}_{tr} \times \mathcal{Y}_{tr}$
2. $[\mathbf{a}, R^2]$=SVDD($\mathcal{X}_{tr}, \mathcal{Y}_{tr}, C_{-1}, C_{+1}$, param)
3. **Test** SVDD on $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$
4. maxiter=1000;
5. i=1;
6. **while**(i<maxiter)
6.1. $\mathcal{X}_{tr_i} = \Xi(\mathcal{X}_{ts})$;
6.2. SVDD-**cross-validation** on $\mathcal{X}_{tr_i} \times \mathcal{Y}_{tr_i}$
6.3. $[\mathbf{a}_i, R_i^2]$=SVDD($\mathcal{X}_{tr_i}, \mathcal{Y}_{tr_i}, C_{-1}, C_{+1}$, param)
6.4. **Test** SVDD on $\mathcal{X}_{ts} \times \mathcal{Y}_{ts}$
6.5. **if**(FPR $< \varepsilon$)
6.5.1 **return** $[a^*, R^{*2}] = [\mathbf{a}_i, R_i^2]$;
6.6. **end**
7. $i = i + 1$;
end

SVDD Zero FPR Iterative Procedure

Here, we present another algorithm for finding zero FPR regions with SVDD. The idea is simply to perform successive SVDDs on the safe regions found with a preliminary SVDD to avoid the presence of unsafe points. Again, we achieve convergence when we reach a fixed number of iterations or when the condition on FPR is satisfied.

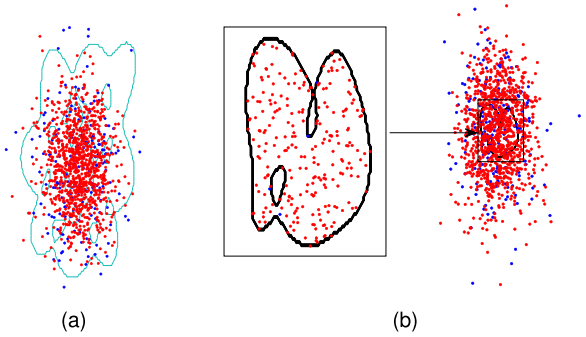


FIGURE 5. Application of Algorithm 2 on a dataset of 2000 target objects sampled from a Gaussian with mean [1,1] and variance 4 and 100 negative examples sampled from a Gaussian with mean [1,1] and variance 5. (a) First iteration of the algorithm. (b) Convergence at the 97th iteration.

We performed this algorithm in MATLAB and tested using data from KEEL.[22] In Figure 5, an example with a 2D Gaussian dataset is reported. It seems clear that the "zeroFPR" algorithm performs better safety regions than "RadiusReduction" since a new SVDD is computed at each iteration and its shape fits the data better. We will confirm this in the "Applications" section, dedicated to applications.

RULES EXTRACTION

We now consider how to make the SVDD explainable in order to explicit the inherent logic and use the extracted rules for further safety envelope tuning as in Mongelli *et al.*'s work.[15]

Let us suppose to have an information vector $\mathbf{I}$ and to have to solve a classification problem depending on two classes $\omega = 0$ or 1. Let $\aleph = \{(\mathbf{I}^k, \omega^k), k = 1, \ldots, \beth\}$ be a dataset corresponding to the collection of events representing a dynamical system evolution (ω) under different system settings ($\mathbf{I}(\cdot)$).

The classification problem consists of finding the best boundary function $f(\mathbf{I}(\cdot), \cdot)$ separating the $\mathbf{I}^k$ points in $\aleph$ according to the two classes $\omega = 0$ or $\omega = 1$. For the case of SVDD, the best boundary f is simply the shape of the hypersphere. Although the shape of the hypersphere may be considered intelligible to some extent (center and radius constitute a good synthesis of it), a rule-based description has a more significant cognitive impact.

Logic Learning Machine

The derivation of $f(\mathbf{I}(\cdot), \cdot))$ in a rule-based shape is made by DT and LLM [the analysis was performed

through the Rulex software suite, developed and distributed by Rulex Inc. (http://www.rulex.ai/)]. They are both based on a set of intelligible rules of the type **if** (*premise*) **then**(*consequence*), where (*premise*) is a logical product (AND, $\wedge$) of conditions and (*consequence*) provides a class assignment for the output. In the present study, the two classes correspond to the presence or the absence of anomalous patterns. LLM rules are obtained through a three-step process. In the first phase (*discretisation and latticisation*), each variable is transformed into a string of binary data in a proper Boolean lattice, using the inverse only-one code binarization. All strings are eventually concatenated in one unique large string per each sample. In the second phase (*shadow clustering*), a set of binary values, called *implicants*, are generated, which allow the identification of groups of points associated with a specific class. (An implicant is defined as a binary string in a Boolean lattice that uniquely determines a group of points associated with a given class. It is straightforward to derive from an implicant an intelligible rule having in its premise a logical product of threshold conditions based on cut-offs obtained during the discretization step. The optimal placement of these cut-offs is, therefore, an important phase to extract the highest information gain before clustering.)[4] During the third phase (*rule generation*), all implicants are transformed into a collection of simple conditions and eventually combined in a set of intelligible rules. The interested reader on shadow clustering and algorithms for efficient rule generation is referred to Muselli and Ferrari's work[18] and references therein.

Rules Extraction From SVDD

The derivation of intelligible rules is made as follows. After that an SVDD has been optimized, a new dataset of observations *sampled around the edge of the SVDD* is provided and the classification via SVDD is registered. The new dataset is then elaborated via LLM. Differently from Carlevaro and Mongelli's work,[5] we need a more refined sampling of SVDD classification to derive the new dataset. The sampling is performed by setting a threshold ε, such that the extracted observations are sufficiently close to the boundary of the trained and tested SVDD. The threshold is set *a priori* and depends on the dataset: given a set $X = \{x_i\}_i$ of synthetic data sampled uniformly from the test set, to extract points close to the radius we evaluate the quantity $t := |\ ||x_i - \mathbf{a}||^2 - R^2||\ |$; therefore, $\varepsilon \in (\min(t), \max(t))$. Values too close to $\min(t)$ do not allow enough samples to be extracted while on the other hand values too close to $\max(t)$ extract too many points away from the edge of the SVDD. A good balance for the choice of ε can then be the average $(\min(t) + \max(t))/2$ or values in a neighborhood of it.

Algorithm 3. ExplainableSVDD Get $\mathbf{a}^*, R^*$ from ZeroFPR algorithms. Fix ε.

1. **Sample** uniformly a new dataset $\mathcal{X}_{new}$ s.t. $x_i \in \mathcal{X}_{new} \Leftrightarrow$ $|\ ||x_i - \mathbf{a}||^2 - R^2\ | < \varepsilon$
2. **Classify** $\mathcal{X}_{new}$ in $\mathcal{Y}_{new}$ through optimal ZeroFPRSVDD (w.r.t. $[\mathbf{a}^*, R^{*2}]$)
3. Solve a classification problem via **LLM** w.r.t. $[\mathcal{X}_{new}, \mathcal{Y}_{new}]$
4. The LLM rule define an explained ZeroFPRSVDD region $\mathcal{R}$
5. **return** $\mathcal{R}$

As in Mongelli *et al.*'s work,[15] we applied these rules with the goal of maximizing the number of safe points (that is, the number of points in the target class), while keeping FPR at zero. This is possible by performing rule tuning as in Mongelli *et al.*'s work,[15] but SVDD allows for much more flexibility.

Figure 6 shows, as an example, a summary of the rules extracted with LLM from SVDD, Algorithm 2, in the case of domain name server (DNS) tunneling (see "DNS Tunneling" section). Each circle represents a rule and the larger this is, the more the respective rule covers a larger number of points. The size of the central hole represents the error of that rule: the larger the hole, the greater the corresponding error. In this example the classification is done in two classes, green and red, and in the outer crown, the input features are shown. The high number of circles (i.e., rules) is an indication of the complexity of the system: with a 2D example, we could say that, in the feature space, a large number of rectangles (i.e., rules) is needed to best approximate the complicated shape of the SVDD. We will discuss these concepts in the next section.

APPLICATIONS

Two applications are now considered. First, we focus on a simple example concerning the stability certification of dynamical systems through region of attraction (ROA),[17] where we want to focus on the performance of rule extraction, and then we move on a much more complex cybersecurity example: the detection of DNS tunneling.[25,26]

ROA Inference

The concept of ROA is fundamental in the stability analysis of dynamical systems[23,40] and it is topical

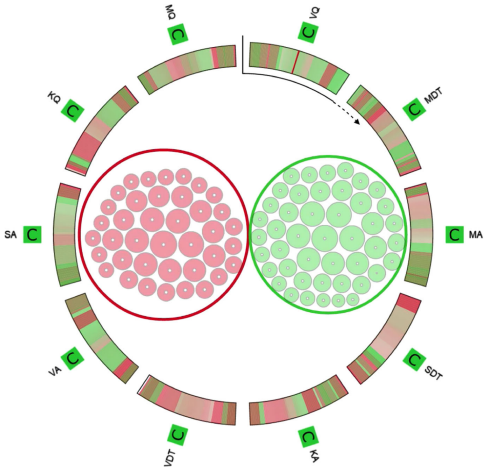


FIGURE 6. Rule Viewer.

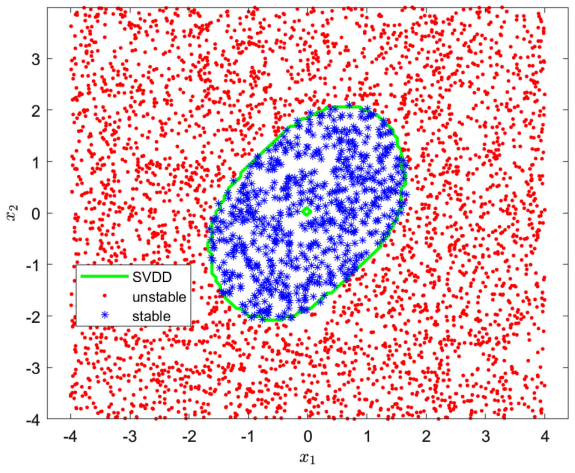


FIGURE 7. ROA of the Van der Pol oscillator. The SVDD shape obtained through fast-SVDD is shown in green, as in the "Fast Training SVDD" section.

when safety of cyber-physical system should be preserved with zero (probabilistic) error.[15,16]

ROA is typically derived through the level sets of Lyapunov functions but in this case, we want to estimate ROA through negative SVDD: we define the target class as the set of stable points and the negative class as the unstable ones.

We consider the Van der Pol oscillator in reverse time

$$\begin{cases} \dot{x}_1 = -x_2 \\ \dot{x}_2 = x_1 + (x_1^2 - 1)x_2 \end{cases} \tag{30}$$

the stability region is depicted in blue in Figure 7. The system has one equilibrium point at the origin and an unstable limit cycle on the border of the true ROA.

The simulation of the dynamical system is developed in C[21] and the dataset is composed by 300,000 points (x_1, x_2) with the relative labels (+1 stable, -1 unstable). Due to the big size of the dataset, a fast SVDD, as in the "Fast Training SVDD" section, is required. We implemented the negative SVDD and tested it over this dataset: we obtained good results (in term of zero FNR) without using Algorithms 1 or 2 due to the good separation between the two classes. In Figure 7, the SVDD shape is shown (in green), and the performance indices are as follows:

$$\mathrm{ACC} = 0.9854 \quad \mathrm{FPR} = 0 \quad \mathrm{FNR} = 0.0542 \tag{31}$$

where $\mathrm{ACC} = \frac{\mathrm{TP}+\mathrm{TN}}{\mathrm{TP}+\mathrm{TN}+\mathrm{FP}+\mathrm{FN}}$ is the accuracy of the model, $\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{FP}+\mathrm{TN}}$ is the false positive rate, and $\mathrm{FNR} = \frac{\mathrm{FN}}{\mathrm{FP}+\mathrm{TN}}$ is the false negative rate.

Then, a set of intelligible rules is extracted, as described in the "Rules Extraction" section (LLM and DT), and they are tested on several extraction of datasets with different sizes, (see Figure 8),[21] with the aim to profile the largest region in term of "safe points," that is related with the precision on the target class $\frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}$.

Here, as example, the first three rules with the highest covering,[a] extracted through Algorithm 3 above

$$\textbf{if } (-1.6 < x_1 \leq 1.2) \wedge (-1.8 < x_2 \leq 1.8) \textbf{ then safe}$$
$$\textbf{if } x_1 \leq -1.6 \textbf{ then unsafe}$$
$$\textbf{if } (-1.6 < x_1 \leq 1.7) \wedge (x_2 \leq -1.8) \textbf{ then unsafe}.$$

We made 10^3 successive extractions from the dataset (with different sizes, from 8% up to 50% of the total points): for each of them, the FPR is almost zero and the precision on the target class is high, i.e., there is a good percentage of safe points. We can see that the performance of the rules extracted with DT after applying SVDD is quite inferior to the others (Figure 8). This is due to the fact that DT generates fewer rules than LLM and the constraint imposed by the shape of SVDD does not allow to generate rules with large coverage (i.e., large rectangles in the features space).

DNS Tunneling

This dataset deals with covert channel detection in cybersecurity;[2] more specifically, the aim is detecting the presence of DNS intruders by an aggregation-based monitoring that avoids packet inspection, in the

[a]The covering of a rule is the percentage of points for which that rule is true.

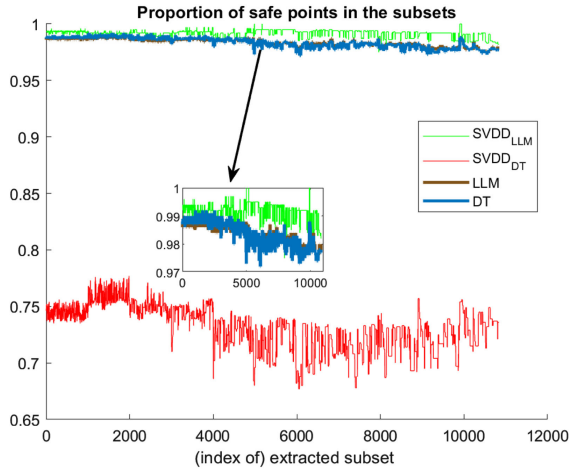


FIGURE 8. Comparison of the percentage of safe points with LLM/DT before and after SVDD, VdP example.

presence of silent intruders and quick statistical fingerprints generation. By modulating the quantity of anomalous packets in the server, we are able to modulate the difficulty of the inherent supervised learning solution via canonical classification schemes (Bayes decision theory, neural networks). However, our goal is to make a good classification even in the cases where the anomalous packets are very much mixed with the legitimate ones, determining the need for more precise and flexible classification methods such as SVDD.

Let q and a be the packet sizes of a query and the corresponding answer, respectively (what answer is related to a specific query can be understood from the packet identifier) and δ the time-interval intercurring between them.

The information vector is composed of the statistics (mean, variance, skewness, and kurtosis) of q, a, and δ for a total number of 12 input features

$$\mathbf{I} = [m_a, m_q, m_\delta, \sigma_a^2, \sigma_q^2, \sigma_\delta^2, s_a, s_q, s_\delta, k_a, k_q, k_\delta].$$

The corresponding vectors are **m**, $\boldsymbol{\sigma}$, **s**, and **k**. High-order statistics give a quantitative indication of the asymmetry (skewness) and heaviness of tails (kurtosis) of a probability distribution; they help improve detection inference.

The training and test sets are built as follows. Let $\{(\mathbf{x}_k, \omega_k), k = 1, \ldots, \aleph\}$ be the training set ($\aleph$ is the training set size), where $\mathbf{x}_k$ is a realization of a vector containing a subset of the features **m**, $\boldsymbol{\sigma}$, **s**, and **k,** and ω_k belongs to $\{0, 1\}$ (the two classes); if the information contained in $\mathbf{x}_k$ corresponds to a DNS data exchange with tunneling: $\omega_k = 1$, $\omega_k = 0$, otherwise.

The classification of the dataset was done through the SVDD algorithms (`RadiusReduction` and `zeroFPRSVDD`) and the results were compared with the DT algorithm and the LLM algorithm (see "Rules Extraction" section), as in the previous section dedicated to the ROA application. As before, our goal is to determine the largest region of parameters with no false positive (i.e., false positive means prediction of tunneling, but not tunneling in reality). To do this, we applied the two algorithms proposed in the "Zero FPR Regions With SVDD" section to the 5000 size sample above (3000 for training and 2000 for test) using $C_1 = 1/\nu_1 N_1$, where $N_1 = \#\{\omega_k = +1\}$ and $\nu_1 = 0.01$ (i.e., we allow the acceptance of up to 1% of negative objects in the target class), $C_2 = 1/\nu_2 N_2$ where $N_2 = \#\{\omega_k = -1\}$ and $\nu_2 = 0.05$ (i.e., we allow up to 5% negative objects to be included in the classifier shape), and RBF kernel with σ determined with cross-validation. The results are shown in Table 1, where FPR is the usual false positive rate, %safe is the percentage of safe points (computed as the precision on the positive class $\frac{\mathrm{TP}}{\mathrm{TP+FP}}$), #iter the number of algorithm iterations, #time (s) the time in second for the convergence, R^2 the squared hypersphere's radius, and #SV the number of determined SVs.

We can observe that the `zeroFPRSVDD` in this case works well than `RadiusReduction`, achieving almost zero FPR with an acceptable large safety region.

Then, we tested the performances of the algorithms in different extractions of 10^3 subsets with different sizes from 8% to 50% of the total points available for test; 11×10^3 trials in total. We compared them with LLM and DT as in Mongelli *et al.*'s work[15] (see Figure 10) and so a rules extraction has been requested (see the "Rules Extraction" section). As an example, here are the first three rules for covering extracted with DT

$$\textbf{if } m_q \leq -0.5 \textbf{ then tunneling}$$
$$\textbf{if } -0.5 < m_q \leq 1.5 \wedge \sigma_q^2 \leq 0.4 \textbf{ then tunneling}$$
$$\textbf{if } m_q > 1.5 \wedge \sigma_q^2 \leq 0.4 \textbf{ then no tunneling}.$$

Native LLM and DT are tuned according to Section 4.4 of Mongelli *et al.*'s work.[15] The procedure has three basic steps. 1) Manual inspection of the most relevant regions for safety. 2) LLM/DT is trained with zero error when developing the rules. 3) Progressive extraction of unsafe points from the original data set until only safe points are obtained. The *native* adjective here means that the algorithms are applied directly, without SVDD interrogation. Due to its intrinsic restriction in modeling data through hyper-rectangles, see, e.g. Grover

TABLE 1. Algorithm statistics for the DNS dataset.

	FPR	% safe	# iter	# time (s)	R^2	#SV
Alg 1	0.0108	80.18	7	65.19	0.7985	61
Alg 2	0.0079	84.71	4	52.13	0.6958	31

et al.'s work,[3] native XAI may not follow the potential tricky nonlinearity that can be chased by SVDD.

The analyses show that the LLM rules extracted from the SVDD model perform better classification than the other methodologies: up to 95% points with near-zero FPR versus only 85% for the classical LLM. The other algorithms perform sufficiently well, more than 50% of the points with near-zero FPR, but, as could be assumed, `zeroFPRSVDD` achieves a better safe region than `RadiusReduction`: this is probably due to the fact that `zeroFPRSVDD` fits the shape of the points better since the algorithm computes a new region at each iteration (see Figure 9) while `RadiusReduction` just rigidly reduces the volume of the SVDD hypersphere until there are points of the other class.

Finally, we report in the following the plot (see Figure 11) concerning the comparison between rule extraction methods with and without the sampling of the points around the edge of the SVDD region (the old algorithm is the one of Carlevaro and Mongelli's work[5]). It is clear that the accuracy of the classification has been improved with the new version of the `ExplainableSVDD` algorithm, thus confirming the observations reported so far.

REMARKS

Zero Statistical Error

Zero statistical error, we have referred to so far, refers to the discovery of the envelope, in the feature space, characterizing the presence of the points of interest of a single class only. We may refer to zero FN when the envelope is a safety envelope as we think to it as the conditions for safety (e.g., no collision in a smart mobility scenario);[41] in that case, the term "positive" means the point is outside of the safety envelope and some risk or danger may be associated to it (a collision). On the other hand, we may refer to zero FP, when we want to discover the envelope, in which

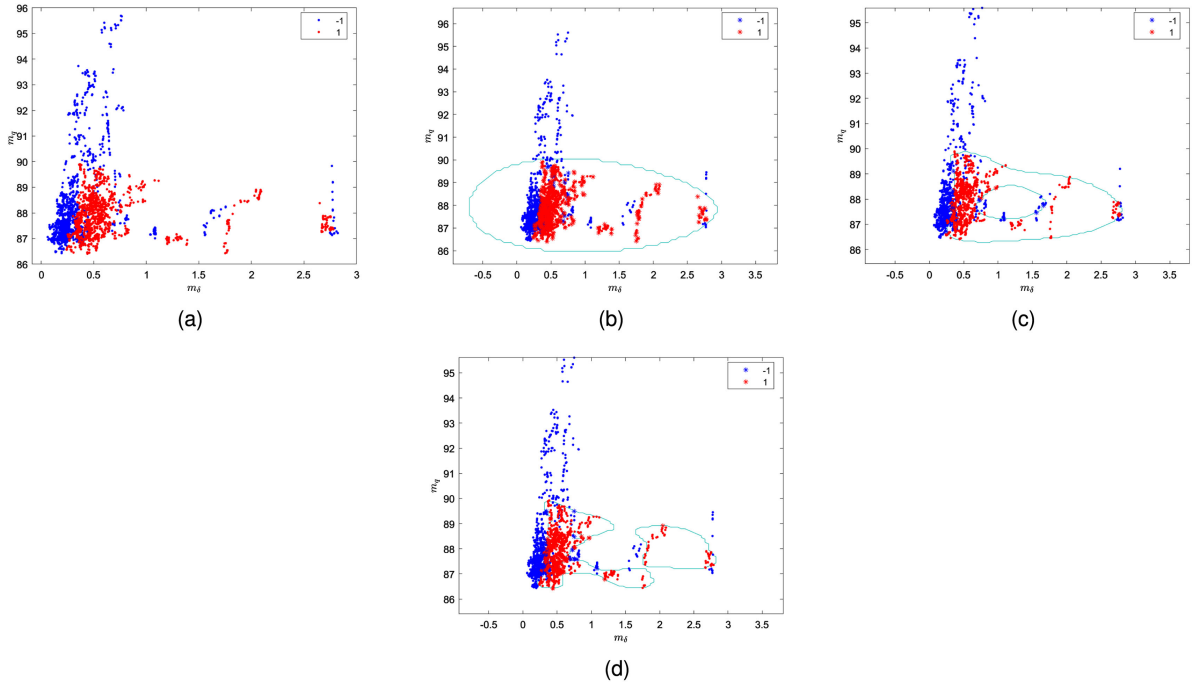


FIGURE 9. 2D graph of the evolution of the "safety region" (the red points are the tunneling ones) with `zeroFPRSVDD`: for this example, we used m_δ (average interarrival time between query and answer packet over 1000 sample) and m_q (average size of query packet) as input features of the DNS tunneling dataset. The star points are the SVs of the description, colored referring their specific label.

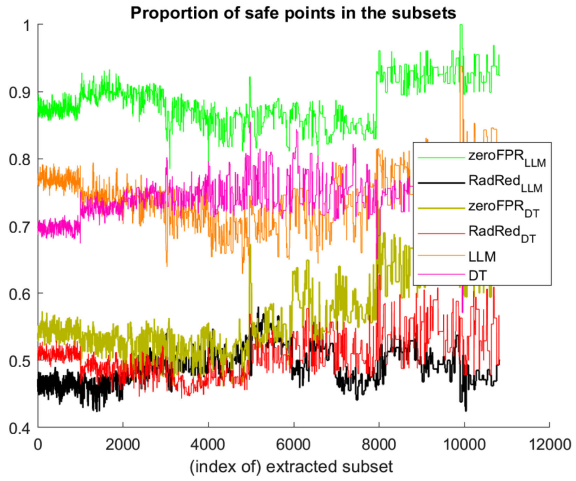


FIGURE 10. Comparison of the percentage of safe points with LLM/DT before and after SVDD, DNS-tunneling example.

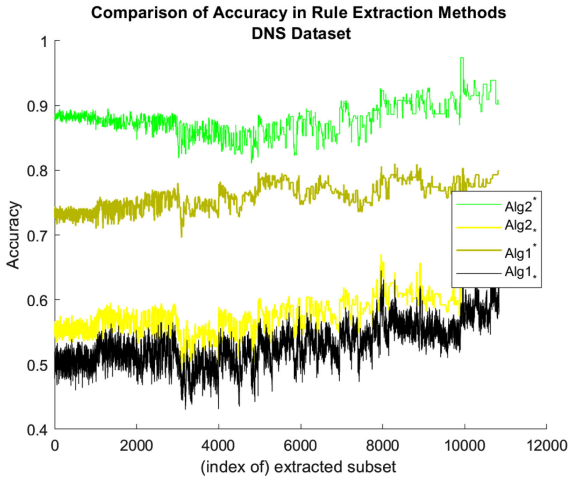


FIGURE 11. Accuracy classification of different extractions of 10^3 subsets of the DNS tunneling dataset. In the legend, the asterisked algorithms at the top (*) refer to those reported in this article, with the rule extraction near the SVDD edge, while those asterisked at the bottom (⁎) refer to the previous version.[5] It is clear that the accuracy of the classification is definitely improved by the new approach.

the risk conditions are certain, namely, all the points of the envelope are anomalous or dangerous; this may be typically associated to the discovery of cyberattacks. For the sake of simplicity, we have followed the zero FPR notation in both algorithm design and performance evaluation. In the DNS tunneling problem, the safety region surrounds the tunneling samples (the red points in Figure 9); in this respect, zero FPR means detected cases are cyberattacks for sure as no legitimate samples lie in the (zero FPR) region.

The term "statistical" is associated to the fact that the metric is still based on measurements performed on the data available; it is not certain as in the formal logic perspective, which is, in turn, a way to certify safety. The two worlds (ML and formal logic), however, may be put in contact; recent studies are dedicated to the formal verification of neural networks[42] and the safety envelope, with zero statistical error property, may be the driver for further formal logic validation.[43]

Data at Production Stage

Results shown in the figures correspond to a validation set, different from the training and test sets used in the cross-validation of the algorithms. Such a validation set would correspond to the production set (i.e., once the ML model is deployed at run time on the "production line," without further retraining), under the assumption that the (unknown) probability distribution generating the data is the same at training and production stages.

The hypothesis may be reasonable or not, depending on the specific application scenario.

In the presented ROA case, the dynamical system is fixed, not affected by noise and no differences are to be considered between training and production stages. Either any variation in the dynamic equations or any environmental noise may be considered during the training phase.

In the DNS case, raw data (from which feature samples are built) derive from the monitoring of a DNS server over a week period, in which traffic variations do not imply significant variations of the ML models (training and test are divided in the proportion of 50%).[45]

CONCLUSION AND FUTURE WORK

This article investigates the use of the SVDD to find envelopes around points of a given class, with zero statistical classification error; the radius of the SVDD is suitable to maintain the largest working conditions, yet with the zero error property. A further interrogation of the SVDD offers support to the intelligibility of the model.

The work on rule extraction is mainly focused on the LLM and the DT algorithms, other approaches may be of interest, such as BEEF[3] and Guidotti *et al.*'s work,[44] which could be the basis for further investigations in intelligible rule extraction.

REFERENCES

1. S. Abe, *Support Vector Machines for Pattern Classification (Advances in Pattern Recognition)*, 2nd ed. London, U.K.: Springer-Verlag, 2010, doi: 10.1007/978-1-84996-098-4.
2. M. Aiello, M. Mongelli, and G. Papaleo, "DNS tunneling detection throughstatistical fingerprints of protocol messages and machine learning," *Int. Nat. J. Commun. Syst.*, vol. 28, no. 14, pp. 1987–2002, 2015, doi: 10.1002/dac.2836.
3. S. Grover, C. Pulice, G. I. Simari, and V. S. Subrahmanian, "BEEF: Balanced English explanations of forecasts," *IEEE Trans. Comput. Social Syst.*, vol. 6, no. 2, pp. 350–364, Apr. 2019, doi: 10.1109/TCSS.2019.2902490.
4. E. Boros, P. L. Hammer, T. Ibaraki, A. Kogan, E. Mayoraz, and I. Muchnik, "An implementation of logical analysis of data," *IEEE Trans. Knowl. Data Eng.*, vol. 12, no. 2, pp. 292–306, Mar./Apr. 2000, doi: 10.1109/69.842268.
5. A. Carlevaro and M. Mongelli, "Reliable AI through SVDD and rule extraction," in *Proc. Int. IFIP Cross Domain (CD) Conf. Mach. Learn. Knowl. Extraction*, 2021, pp. 153–171, doi: 10.1007/978-3-030-84060-0-10.
6. V. N. Balasubramanian, S. S. Ho, and V. Vovk, *Conformal Prediction for Reliable Machine Learning*, 1st ed. Waltham, MA, USA: Morgan Kaufmann Elsevier, 2014.
7. A. Chaudhuri *et al.*, "Sampling method for fast training of support vector data description," *Annu. Rel. Maintainability Symp.*, 2018, doi: 10.1109/RAM.2018.8463127.
8. European Union Aviation Safety Agency, "Concepts of design assurance for neural networks CoDANN," EASA AI Task Force. AG, Mar. 2020.
9. D. Fisch, A. Hofmann, and B. Sick, "On the versatility of radial basis function neural networks: A case study in the field of intrusion detection," *Inf. Sci.*, vol. 180, no. 12, pp. 2421–2439, 2010, doi: 10.1016/j.ins.2010.02.023.
10. J.I. Ge and G. Orosz, "Dynamics of connected vehicle systems with delayed acceleration feedback," *Transp. Res. C, Emerg. Technol.*, vol. 46, pp. 46–64, 2014, doi: 10.1016/j.trc.2014.04.014.
11. G. Huang, H. Chen, Z. Zhou, F. Yin, and K. Guo, "Two-class support vector data description," *Pattern Recognit.* vol. 44, pp. 320–329, 2011, doi: 10.1016/j.patcog.2010.08.025.
12. D. Jia, K. Lu, J. Wang, X. Zhang, and X. Shen, "A survey on platoon-based vehicular cyber-physical systems," *IEEE Commun. Surv. Tut.*, vol. 18, no. 1, pp. 263–284, Jan.–Mar. 2016, doi: 10.1016/j.patcog.2010.08.025.
13. C.A. Jones, "Lecture notes: Math2640 introduction to optimisation 4," Univ. Leeds, Sch. Math., Tech. Rep. 11, 2005.
14. M. Mongelli, M. Muselli, A. Scorzoni, and E. Ferrari, "Accellerating PRISM validation of vehicle platooning through machine learning," in *Proc. 4th Int. Conf. Syst. Rel. Saf.*, 2019, pp. 452–456, doi: 10.1109/ICSRS48664.2019.8987672.
15. M. Mongelli, M. Muselli, E. Ferrari, and A. Fermi, "Performance validation of vehicle platooning via intelligible analytics," *IET Cyber- Phys. Syst.: Theory Appl.*, vol. 4, no. 2, pp. 120–127, 2019, doi: 10.1049/iet-cps.2018.5055.
16. A. Fermi, M. Mongelli, M. Muselli, and E. Ferrari, "Identification of safety regions in vehicle platooning via machine learning," in *Proc. 14th IEEE Int. Workshop Factory Commun. Syst.*, Imperia, Italy, 2018, pp. 1–4, doi: 10.1109/WFCS.2018.8402372.
17. M. Mongelli and V. Orani, "Stability certification of dynamical systems: Lyapunov logic learning machine," in *Applied Soft Computing and Communication Networks* (Lecture Notes in Networks and Systems 187) S. M. Thampi, J. L. Mauri, X. Fernando, R. Boppana, S. Geetha, and A. Sikora, Eds., Singapore: Springer, 2021, doi: 10.1007/978-981-33-6173-7_15.
18. M. Muselli and E. Ferrari, "Coupling logical analysis of data and shadow clustering for partially defined positive Boolean function reconstruction," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 1, pp. 37–50, Jan. 2011, doi: 10.1109/TKDE.2009.206.
19. H. Nunez, C. Angulo, and A. Català, "Rule-based learning systems for support vector machines," *Neural Process. Lett.*, vol. 24, pp. 1–18, 2006, doi: 10.1007/s11063-006-9007-8.
20. S. Oncu, N.vande Wouw, and H. Nijmeijer, "Cooperative adaptive cruise control: Tradeoffs between control and network specifications," in *Proc. 14th Int. IEEE Conf. Intell. Transp. Syst.*, 2011, pp. 2051–2056, doi: 10.1109/ITSC.2011.6082894.
21. M. Mongelli and V. Orani, "Git repository of Lyapunov logic learning machine," [Online]. Available: https://github.com/mopamopa/Liapunov-Logic-Learning-Machine
22. KEEL, "Website: KEEL (knowledge extraction based on evolutionary learning)," Nov. 2012. [Online]. Available: http://sci2s.ugr.es/keel/datasets.php
23. H. Khalil, "Nonlinear Systems," *Encyclopedia of Life Support Systems (EOLSS)*, 3rd ed., Hoboken, NJ, USA: Prentice-Hall, 2002.
24. J. Kools, "6 functions for generating artificial datasets," *MATLAB Central File Exchange*, Apr. 4, 2021. [Online]. Available: https://www.mathworks.com/matlabcentral/fileexchange/41459-6-functions-for-generating-artificial-datasets
25. P. Pop *et al.*, "The safecopecsel project: Safe cooperating cyber-physical systems using wireless communication," in *Proc. Euromicro. Conf. Digit. Syst. Des.*, 2016, pp. 532–538, doi: 10.1109/DSD.2016.25.

26. P. Pop *et al.*, "Safe cooperating cyber-physical systems using wireless communication," *Microprocess. Microsyst.*, vol. 53, pp. 42–50, 2017, doi: 10.1109/DSD.2016.25.
27. M. Ribeiro, S. T. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144, doi: 10.1145/2939672.2939778.
28. Rulex analytics platform. [Online]. Available: https://www.rulex.ai
29. K. Czarnecki and R. Salay, "Towards a framework to manage perceptual uncertainty for safe automated driving," in *Proc. Int. Workshop Artif. Intell. Saf. Eng.*, 2018. [Online]. Available: https://uwaterloo.ca/waterloo-intelligent-systems-engineering-lab/publications/towards-framework-manage-perceptual-uncertainty-safe, doi: 10.1007/978-3-319-99229-7_37.
30. S. Santini, A. Salvi, A. S. Valente, A. Pescapé, M. Segata, and R. Lo Cigno, "A consensus-based approach for platooning with intervehicular communications and its validation in realistic scenarios," *IEEE Trans. Veh. Technol.*, vol. 66, vol. 3, pp. 1985–1999, Mar. 2017, doi: 10.1109/TVT.2016.2585018.
31. *Standardization in the Area of Artificial Intelligence*, ISO/IEC JTC 1/SC 42, 2017. [Online]. Available: https://www.iso.org/committee/6794475.html
32. M. Segata and R. L. Cigno, "Automatic emergency braking: Realistic analysis of car dynamics and network performance," *IEEE Trans. Veh. Technol.*, vol. 62, no. 9, pp. 4150–4161, Nov. 2013, doi: 10.1109/TVT.2013.2277802.
33. *Road Vehicles – Safety of the Intended Functionality*, ISO PAS 21448:2019, International Organization for Standardization, Geneva, Switzerland, 2019.
34. D. M. J. Tax and R. P. W. Duin, "Support vector domain description," *Pattern Recognit. Lett.*, vol. 20, pp. 1191–1199, 1999, doi: 10.1016/S0167-8655(99)00087-2.
35. D. M. J. Tax and R. P. W. Duin, " Support vector data description," *Mach. Learn.*, vol. 54, pp. 45–66, 2004, doi: 10.1023/B:MACH.0000008084.60811.49.
36. D.M. Tax, "One-class classification, concept-learning in the absence of counter-examples," Ph.D. dissertation, Technische Universiteit Delft, Delft Univ. Technol., Delft, The Netherlands, 2001.
37. A. Theissler and I. Dear, "Autonomously determining the parameters for SVDD with RBF kernel from a one-class training set," in *Proc. WASET Int. Conf. Mach. Intell.*, Stockholm, Sweden, 2013, pp. 113–114, doi: 10.5281/zenodo.1087117.
38. V. Vapnik, *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer, 1995, doi: 10.1007/978-1-4757-3264-1.
39. L. Xu, L. Y. Wang, G. Yin, and H. Zhang, "Communication information structures and contents for enhanced safety of highway vehicle platoons," *IEEE Trans. Veh. Technol.*, vol. 63, no. 9, pp. 4206–4220, Nov. 2014, doi: 10.1109/TVT.2014.2311384.
40. C. Zhai and H. D. Nguyen, "Estimating the Region of Attraction for Power Systems Using Gaussian Process and Converse Lyapunov Function," *IEEE Trans. Control Syst. Technol.*, 2019, doi: 10.1109/TCST.2021.3098167.
41. M. Mongelli, "Design of countermeasure to packet falsification in vehicle platooning by explainable artificial intelligence," *Comput. Commun.*, vol. 179, pp. 166–174, 2021, doi: 10.1016/j.comcom.2021.06.026.
42. *Concepts of Design Assurance for Neural Networks CODANN*, European Union Aviation Safety Angency, Daedalean, AG, Standard, Mar. 2020. [Online]. Available: https://www.easa.europa.eu/sites/default/files/dfu/EASA-DDLNConcepts-of-Design-Assurance-for-Neural-Networks-CoDANN.pdf
43. M. Mongelli, M. Muselli, E. Ferrari, and A. Scorzoni, "Accellerating PRISM validation of vehicle platooning through machine learning," in *Proc. 4th Int. Conf. Syst. Rel. Saf.*, 2019, pp. 452–456, doi: 10.1109/ICSRS48664.2019.8987672.
44. R. Guidotti, A. Monreale, F. Giannotti, D. Pedreschi, S. Ruggieri, and F. Turini, "Factual and counterfactual explanations for black box decision making," *IEEE Intell. Syst.*, vol. 34, no. 6, pp. 14–23, Nov./Dec. 2019, doi: 10.1109/MIS.2019.2957223.
45. M. Aiello, M. Mongelli, and G. Papaleo, "DNS tunneling detection through statistical fingerprints of protocol messages and machine learning," *Int. J. Commun. Syst.*, vol. 28, pp. 1987–2002, 2015, doi: 10.1002/dac.2836.

ALBERTO CARLEVARO is currently a Ph.D. student with the Department of Electrical, Electronic and Telecommunications Engineering and Naval Architecture, University of Genoa, Genova, Italy, working on the research topic "Traffic Analysis in the Smart City," in collaboration with CNR and S.M.E. Aitek. He was a research fellow at the Institute of Electronic, Computer and Telecommunications Engineering (IEIIT), National Research Council (CNR), where he worked on machine learning and explainable AI in collaboration with Rulex Inc. His current research interests include machine learning, statistical learning, explainable AI. Carlevaro received the master's degree in applied mathematics from the University of Genoa in May 2020, with 110 out of 110 *cum laude* with a physics-mathematics thesis on the behavior of liquid crystals under electromagnetic fields. He is the corresponding author of this article. Contact him at alberto.carlevaro@edu.unige.it.

MAURIZIO MONGELLI has been a researcher since 2012 at the Institute of Electronics, Computer, and Telecommunication Engineering (IEIIT), National Research Council of Italy (CNR), Rome, Italy, where he deals with machine learning applied to cyber-physical systems and bioinformatics. During his doctorate and in the following years, he worked on the quality of service for military networks with Selex. He is the co-author of more than 100 international scientific papers and two patents. Mongelli received the Ph.D. degree in electronics and computer engineering from the University of Genova, Genova, Italy, in 2004. The doctorate was funded by Selex Communications (Selex). He is a Member of the SAE G-34/EUROCAE WG-114 'AI in Aviation' Committee. Contact him at maurizio.mongelli@ieiit.cnr.it.