Journal of Computational and Applied Mathematics 454 (2025) 116201

Contents lists available at ScienceDirect

Journal of Computational and Applied Mathematics

journal homepage: www.elsevier.com/locate/cam

# Difference of Convex programming in adversarial SVM

Annabella Astorino [a], Manlio Gaudioso [a], Enrico Gorgone [b,*], Benedetto Manca [b]

[a] *Dipartimento di Ingegneria Informatica, Modellistica, Elettronica e Sistemistica, Università della Calabria, Via P. Bucci Edificio 42C piano 6, Rende, 87036, Italy*

[b] *Dipartimento di Matematica e Informatica, Università degli Studi di Cagliari, Palazzo delle Scienze, Via Ospedale, 72, Cagliari, 09124, Italy*

ARTICLE INFO

*Keywords:*
DC programming
Adversarial machine learning
Sparse optimization

ABSTRACT

We present two models in adversarial machine learning, focussing on the Support Vector Machine framework. In particular, we consider both an evasion and a poisoning problem. The first model is aimed at constructing effective *sparse* perturbation of the dataset samples, while the objective of the second is to induce a substantial *rotation* of the hyperplane defining the classifier. The two models are formulated as Difference of Convex nonsmooth optimization problems. Numerical results on both synthetic and real life datasets are reported.

## 1. Introduction

Adversarial machine learning (AML) is a research area that has experienced a dramatic development in the last two decades. It is concerned with the malicious data manipulation, considered under two conflicting points of view, that of the *defender*, whose objective is to protect the correct behaviour of any learning mechanism, and that of the *attacker*, whose aim is to mislead the same mechanism.

The research in AML has stemmed from application areas of crucial importance, such as image processing, spam filtering and malware detection [1]. In particular most of the research has been stimulated by the vulnerability evidence of many Machine Learning models, mainly in deep network applications [2,3]. In addition, it is worth noting that machine learning applications are the weakest link in the security chain of complex information-communication systems [4]. Consequently, protection of such applications is a crucial issue in cyber-security.

The AML literature is extremely rich and we cite here, on the attacker side, papers like [5,6], which provide classification of problems and methods as well as useful historical details. On the defender side we mention, in particular, [7].

In this paper we focus on adversarial attacks in the framework of supervised binary classification, a relevant chapter of machine learning.

Supervised binary classification is a decision-making technique whose objective is to assign exactly one of two possible classes to each sample in a population. The number of applications is practically infinite. A sample can be a patient and the decision is whether he/she belongs either to the class of people affected by a certain pathology or to that of the not-affected ones. Analogously, a sample can be a mail message and the decision is about being it a spam or a not-spam message.

Each sample is represented by a vector containing the numerical values of several attributes (*features*). The supervised binary classification process primarily needs a dataset of *labelled* samples whose class membership is known (the *training* set). On the base of such an information, the *classifier* is built as a mathematical tool able to predict the class membership of any newly incoming

* Corresponding author.
*E-mail addresses:* annabella.astorino@dimes.unical.it (A. Astorino), manlio.gaudioso@unical.it (M. Gaudioso), gorgone@unica.it (E. Gorgone), bmanca@unica.it (B. Manca).

https://doi.org/10.1016/j.cam.2024.116201
Received 10 January 2024; Received in revised form 28 May 2024
Available online 8 August 2024

sample. A distinct set of labelled samples, which play no role in the construction of the classifier, is finally employed to check the quality of the designed classifier. Such set is usually referred to as the *testing* one.

Construction of the classifier requires in general the solution of a numerical optimization problem to detect an "optimal" surface to separate the two classes of samples in the space of the attributes. We recall here the pioneering works [8–10]. An interesting overview is provided in [11].

Most of the classification methods are based on the use of a hyperplane as the separating surface and belong to the well known family of Support Vector Machine (SVM) models [12,13]. On the other hand, separation by a variety of nonlinear surfaces has been extensively discussed in the literature (see [14–20] and the references therein).

As for the adversarial classification, two types of attacks are commonly considered, the *evasion* and the *poisoning* ones, according to their timing, either at the *testing* or at the *training* phase, respectively. Such types of attacks are related to attacker's ability to manipulate either the testing or the training set. A significant distinction among the types of attack is in fact connected to the quality of the classification system knowledge at attacker's disposal, which can range from being perfect to black-box.

In this paper we consider the SVM model. As previously mentioned, SVM is about separation of two discrete point-sets in $\mathbb{R}^n$ (the *labelled samples*) by means of a hyperplane with an associate separation margin.

As for the impact of adversarial attacks on SVM systems, the literature offers a wide body of research results, according to the different objective pursued, the assumptions on the specific framework at hand and the available methodologies.

On the defender side, the studies on SVM robustness provide a significant contribution [21,22], while defence against *label* noise is analysed in [23].

In the area of the attack models we recall [24], where the objective (the evasion) is to transform a malicious sample into a legitimate one by means of a bounded perturbation, while trying to avoid that the perturbed sample falls into low density legitimate areas.

Poisoning attacks are considered in [25], on the basis of the distinction between free-range and restrained attacks. The problem tackled in [26] is to design a sample point whose insertion into the dataset produces the maximal deterioration of the classification accuracy.

In the following we consider both the evasion and the poisoning problems in SVM systems.

As for the evasion, we focus on *sparse* perturbations, thus we model effective adversarial attacks acting on the smallest number of sample features.

Sparse attacks have been considered in [27], where the use of $\ell_1$ regularization is suggested for defence purposes. In our approach, instead, we take inspiration from the literature on the Feature Selection [28] and we enforce perturbation sparsity by using the $\ell_0$ pseudo-norm. We recall that the $\ell_0$ pseudo-norm [29] (denoted as $\|.\|_0$) counts the number of non-zero components of any vector.

Summing up, we come out with a novel sparse optimization problem which is tackled by adopting the $k$-norm representation of the $\ell_0$ pseudo-norm [30–34].

As for the poisoning problem, we consider the case where the Attacker tries to deteriorate the accuracy of a given SVM classifier by acting on a subset of the dataset. The objective is to induce the SVM model to choose a fairly different separation hyperplane w.r.t. the original one. The novelty of our approach is in introducing a measure of the hyperplane *rotation* produced by the perturbation.

Both our evasion and poisoning models require the solution of a nonconvex nonsmooth optimization problem of the Difference of Convex (DC) type. DC problems have been intensively studied from both the theoretical and the numerical point ov view in [35–39]. In our implementation we adopt the Descent–Ascent algorithm (DADC) described in [40].

Throughout the paper, we consider a standard SVM framework for binary classification. In particular we suppose that two point-sets in $\mathbb{R}^n$, $\mathcal{A} = \{a_i\}$, $i \in I = \{1, \dots, m_a\}$ and $\mathcal{B} = \{b_l\}$, $l \in L = \{1, \dots, m_b\}$, are given. We indicate by $\mathcal{X}$ the entire dataset ($\mathcal{X} = \mathcal{A} \cup \mathcal{B}$) and by $m = m_a + m_b$ its cardinality; consequently $x_j$, $j = 1, \dots, m$, is any element of the dataset, independently of its class membership.

The Defender constructs the classifier (see [12,13]) by stating the following problem:

$$\left|\begin{array}{ll} \min\limits_{w,\gamma,\xi,\zeta} & \frac{1}{2} w^\top w + C\Big(\sum\limits_{i\in I} \xi_i + \sum\limits_{l\in L} \zeta_l\Big) \\ s.t. & a_i^\top w - \gamma + 1 \le \xi_i \quad i \in I \\ & -b_l^\top w + \gamma + 1 \le \zeta_l, \quad l \in L \\ & \xi_i \ge 0, \quad i \in I \\ & \zeta_l \ge 0, \quad l \in L \end{array}\right. \tag{1}$$

where a separating hyperplane, defined by the couple ($w \in \mathbb{R}^n, \gamma \in \mathbb{R}$), is calculated in view of minimizing the sum of the classification errors $\xi_i, i \in I$ and $\zeta_l, l \in L$ for the two sets $\mathcal{A}$ and $\mathcal{B}$, respectively, while maximizing the separation margin, thanks to the quadratic term in the objective function. The (tunable) parameter $C > 0$ represents the trade-off between the two objectives. Observe that the above quadratic problem can be reformulated in the following way as an unconstrained convex nonsmooth optimization problem:

$$\min_{w,\gamma} \frac{1}{2} w^\top w + C\big(\sum_{i\in I} \max\{0, a_i^\top w - \gamma + 1\} + \sum_{l\in L} \max\{0, -b_l^\top w + \gamma + 1\}\big) \tag{2}$$

Fig. 1 shows an example of a separating hyperplane obtained via SVM.

The paper is organized as follows. We describe in Section 2 our sparse evasion strategy. The poisoning model is treated in Section 3, while the results of some numerical experiments are reported in Section 4.

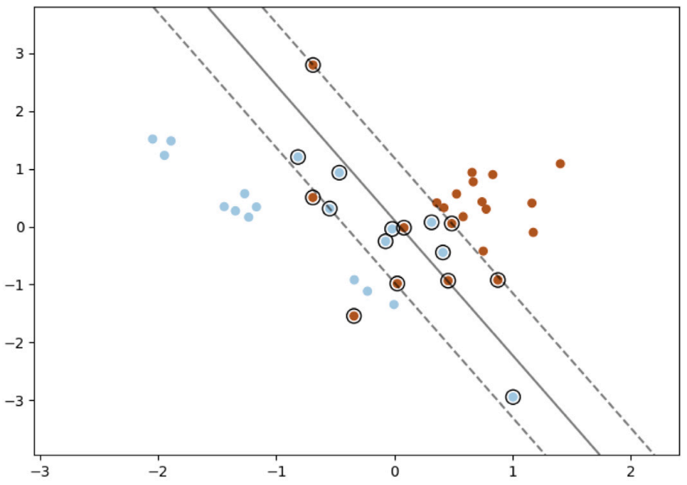

**Fig. 1.** An example of a separating hyperplane having classification errors obtained as a solution to the SVM problem (1).

## 2. The sparse evasion model

Suppose that the problem (1) has been solved at the Defender site and let the hyperplane $H$ associated to the optimal couple $(w', \gamma')$ be

$$H = \left\{x \in \mathbb{R}^n \,|\, {w'}^\top x = \gamma'\right\},$$

In the sequel $w'$ will be referred to as the *normal* to the separating hyperplane $H$. We assume also that the Attacker has complete knowledge of the classifier. Now let $a_i$ $(b_l)$ be any correctly classified point of $\mathcal{A}$ $(\mathcal{B})$ which the Attacker is willing to perturb to evade the classifier. Since the inequality ${a'}_i^\top w' \le \gamma' - 1$ $(b_l^\top w' \ge \gamma' + 1)$ holds, any perturbation $\delta_i \in R^n$ $(\delta_l \in R^n)$ makes point $a_i(b_l)$ misclassified, provided the inequality $(a_i + \delta_i)^\top w' > \gamma' - 1$ $((b_l + \delta_l)^\top w' < \gamma' + 1)$ is satisfied. Such inequalities can be rewritten as:

$$ {w'}^\top \delta_i > \rho_i = -a_i^\top w' + \gamma' - 1 \ge 0, \quad i \in I \tag{3}$$

and

$$ {w'}^\top \delta_l < \rho_l = -b_l^\top w' + \gamma' + 1 \le 0, \quad l \in L \tag{4}$$

As usual in SVM formulation, we assign the label 1 to the samples in $\mathcal{A}$ and $-1$ to those in $\mathcal{B}$. Thus, letting $x_j$ be any well classified sample either in $\mathcal{A}$ or $\mathcal{B}$, with label $y_j = 1$ if $x_j \in \mathcal{A}$ and $y_j = -1$ otherwise, the condition on perturbation $\delta_j$ ensuring evasion of sample $x_j \in \mathcal{X}$ can be written in the form

$$y_j({w'}^\top \delta_j) \ge y_j \rho_j,$$

where $\rho_j$ takes the form indicated in formula (3) or (4) according to the fact that sample $x_j$ is in $\mathcal{A}$ or in $\mathcal{B}$.

The rationale of our *sparse* approach is to minimize the norm of $\delta_j$ (we adopt the $\ell_\infty$ norm), while keeping the number of perturbed components of $x_j$ below a fixed threshold. Finally, we come out with the following sparsity constrained nonlinear program $(P_j)$:

$$\left|\begin{array}{ll} \min\limits_{\delta_j} & \|\delta_j\|_\infty \\ s.t. & y_j {w'}^\top \delta_j \ge y_j \rho_j \\ & \|\delta_j\|_0 \le k \end{array}\right. \tag{5}$$

for an appropriately selected value of $k$, $1 \le k \le n$.

In order to treat the cardinality constraint $\|\delta_j\|_0 \le k$, we resort to a $k$-norm-based reformulation. We recall (see [30,31]) that the $k$-norm of any vector $x \in \mathbb{R}^n$, indicated by $\|x\|_{[k]}$, is the sum of $k$ maximal components (in modulus) of $x$, $k = 1, \dots, n$.

The $k$-norm enjoys the following properties:

(i) lower semi-continuity;
(ii) $\|x\|_\infty = \|x\|_{[1]} \le \cdots \le \|x\|_{[k]} \le \cdots \|x\|_{[n]} = \|x\|_1$;
(iii) $\|x\|_0 \le k \;\; \Rightarrow \;\; \|x\|_1 - \|x\|_{[s]} = 0, \;\; k \le s \le n.$

In our approach we exploit the fundamental equivalence holding for any $k = 1, \dots, n$:

$$\|x\|_0 \le k \;\; \Leftrightarrow \;\; \|x\|_1 - \|x\|_{[k]} = 0,$$

which allows us to reformulate problem (5) as:

$$\left|\begin{array}{ll}\min\limits_{\delta_j} & \|\delta_j\|_\infty \\ s.t. & y_j w'^\top \delta_j \geq y_j \rho_j \\ & \|\delta_j\|_1 - \|\delta_j\|_{[k]} \leq 0\end{array}\right. \tag{6}$$

We observe that the original sparsity constraint $\|\delta_j\|_0 \leq k$ has been substituted by the nonconvex, nonsmooth constraint of the DC type $\|\delta_j\|_1 - \|\delta_j\|_{[k]} \leq 0$.

In the sequel we will refer to the constraints $y_j w'^\top \delta_j \geq y_j \rho_j$ and $\|\delta_j\|_1 - \|\delta_j\|_{[k]} \leq 0$ of problem (6) as the *classification* and *k-norm* constraints, respectively.

In the numerical treatment of the above problem, we adopt a penalty function approach, by introducing the penalty parameter $\sigma > 0$. We come out with the following DC problem,

$$\left|\begin{array}{ll}\min\limits_{\delta_j} & \|\delta_j\|_\infty + \sigma(\|\delta_j\|_1 - \|\delta_j\|_{[k]}) \\ s.t. & y_j w'^\top \delta_j \geq y_j \rho_j,\end{array}\right. \tag{7}$$

which we treat by applying the Descent–Ascent DC algorithm (DADC) introduced in [40].

## 3. The poisoning model

In our poisoning model we consider the same framework as in previous Section, thus we assume that the Defender has found the optimal separating hyperplane $H$ associated to the couple $(w', \gamma')$. The Attacker is able to modify the attributes of a subset of samples, taken both from $\mathcal{A}$ and $\mathcal{B}$, corresponding to the index sets $I_M \subset I$ and $L_M \subset L$, respectively. Thus Attacker's objective is to induce the maximal deterioration of the classification accuracy acting on the samples at hand. Such an objective is pursued by pushing the SVM model to choose a hyperplane whose normal $w$ is as much as possible different from the original one $w'$. In other words, the objective is to significantly *rotate* the normal of the separating hyperplane.

Such an objective can be expressed in terms of minimization of the modulus of the cosine between the new normal $w$ and the original $w'$. The decision variables, together with the couple $(w, \gamma)$, are the perturbations $\delta_i$, $i \in I_M$ and $\delta_l$, $l \in L_M$. Keeping in mind the SVM nonsmooth formulation (2), we state the following problem, where minimization of the modulus of the scalar product between the normal $w$ to the perturbed hyperplane and $w'$ is aimed at maximizing the poisoning effect. The positive parameters $\theta_i$ and $\theta_l$ provide the bounds on the $\ell_2$ norm of the perturbations.

$$\left|\begin{array}{ll}\min\limits_{w,\gamma,\delta} & \frac{1}{2}w^\top w + C_1|w^\top w'| + C_2\Big(\sum\limits_{i\in I\setminus I_M} \max\{0, a_i^\top w - \gamma + 1\} \\ & + \sum\limits_{l\in L\setminus L_M} \max\{0, -b_l^\top w + \gamma + 1\} + \sum\limits_{i\in I_M} \max\{0, (a_i+\delta_i)^\top w - \gamma + 1\} \\ & + \sum\limits_{l\in L_M} \max\{0, -(b_l+\delta_l)^\top w + \gamma + 1\}\Big) \\ s.t. & \\ & \|\delta_i\|_2^2 \leq \theta_i, \quad i \in I_M \\ & \|\delta_l\|_2^2 \leq \theta_l, \quad l \in L_M\end{array}\right. \tag{8}$$

**Remark 1.** Note that the objective function is nonsmooth and nonconvex. In particular the term $|w^\top w'|$ is convex and nonsmooth, while the bilinear terms $\delta_i^\top w$ and $\delta_l^\top w$ under the max operator result in nonsmoothness and non convexity. The latter can be however put in DC form by observing that any function $f(x,y) = \max\{0, x^\top y\}$ can be rewritten as

$$f(x,y) = \max\{0, \frac{1}{4}(\|x+y\|^2 - \|x-y\|^2)\} = \frac{1}{4}\max\{\|x+y\|^2, \|x-y\|^2\} - \|x-y\|^2$$

**Remark 2.** To calculate a possible indicator of the effectiveness of our poisoning strategy, we proceed as follows. After a solution $(w^*, \gamma^*, \delta^*)$ of (8) is obtained, we then train a standard SVM model using the dataset $\tilde{\mathcal{X}} = \{a_i, i \in I \setminus I_M\} \cup \{a_i + \delta_i, i \in I_M\} \cup \{b_l, l \in L \setminus L_M\} \cup \{b_l + \delta_l, l \in L_M\}$ to obtain a new separating hyperplane $(\tilde{w}, \tilde{\gamma})$ which we use to have a quantitative evaluation of the accuracy deterioration on the non manipulated points.

## 4. Numerical experiments

In this section we report the numerical experiments for the problems described in Sections 2 and 3. We have tested the two models for both synthetic and real datasets on a IMac 2019 with a 3.7 GHz 6-core Intel Core i5 with 16 GB RAM. The synthetic datasets have been generated using the Python package scikit-learn [41] which allows to generate $m$ random points in $\mathbb{R}^n$ inside a hyper-cube whose size is determined by the parameter class_sep (higher values correspond to an easier classification task on the dataset). The real datasets have been taken from the LIBSVM repository [42].

For each dataset, either synthetic or real, we have first applied the linear SVM classifier implemented in the scikit-learn package of Python, equipped with a 10-fold cross validation grid-search aimed at model selection. This has allowed us to obtain the appropriate

**Table 1**
Details on the synthetic datasets used to test the sparse evasion model.

| Dataset | class_sep | $m$ | $n$ | $\|w'\|_0$ | wellc | misc |
|---|---|---|---|---|---|---|
| 1 | 0.5 | 150 | 10 | 10 | 131 | 19 |
| 2 | 0.5 | 150 | 30 | 30 | 129 | 21 |
| 3 | 0.5 | 300 | 10 | 10 | 224 | 76 |
| 4 | 0.5 | 300 | 30 | 30 | 241 | 59 |
| 5 | 1.0 | 150 | 10 | 10 | 146 | 4 |
| 6 | 1.0 | 150 | 30 | 30 | 144 | 6 |
| 7 | 1.0 | 300 | 10 | 10 | 273 | 27 |
| 8 | 1.0 | 300 | 30 | 30 | 282 | 18 |

**Table 2**
Details on the real datasets used to test the sparse perturbation model.

| Dataset | $m$ | $n$ | $\|w'\|_0$ | wellc | misc |
|---|---|---|---|---|---|
| australian | 690 | 14 | 12 | 593 | 97 |
| breast | 683 | 10 | 10 | 660 | 23 |
| diabetes | 768 | 8 | 8 | 596 | 172 |
| heart | 270 | 13 | 13 | 230 | 40 |
| ionosphere | 351 | 34 | 33 | 319 | 32 |
| sonar | 208 | 60 | 60 | 190 | 18 |
| splice | 1000 | 60 | 60 | 818 | 182 |

value of the hyper-parameter $C$, which in turn has been utilized on each entire dataset, giving us the separating hyperplane with normal $w' \in \mathbb{R}^n$ as well as the list of well-classified and misclassified points.

For both models (7) and (8) we have adopted the DCDA algorithm proposed in [40], which is designed for unconstrained DC problems. Therefore, both of them have been reformulated as unconstrained problems by constraints penalization.

In calculating the $\ell_0$ pseudo-norm of any vector $x$, we assume equal to zero any component $x_i$ satisfying the condition:

$$\frac{|x_i|}{\|x\|_1} < 10^{-4}. \tag{9}$$

## 4.1. Numerical results for the sparse evasion model

In this subsection we report the numerical results obtained by applying the sparse perturbation model (7) on the synthetic and real datasets.

In Tables 1 and 2 are reported the dimension of both the synthetic and real datasets, together with the $\ell_0$ pseudo-norm of $w'$, the total number of well-classified points (wellc) and misclassified points (misc) by the linear SVM. For the synthetic datasets the indicator (class_sep) is reported too.

For each dataset we have applied the sparse perturbation model (7) to each well-classified sample in $\mathcal{X}$. We have adopted different values of $k$, the target number of sample components to be perturbed, letting $k$ range between 50% and 100% of $n$. As mentioned before, we have penalized also the linear constraint $y_j w^\top \delta_j \geq y_j \rho_j$, by introducing the penalty parameter $\beta > 0$, $\forall i$. We have considered a tuning grid for both $\beta$ and $\sigma$, in particular $\beta \in \{1, 1.5, \dots, 20\}$ and $\sigma \in \{0.05, 0.1\} \cup \{0.5, 1, \dots, 5\}$. For evaluation purposes we have considered the following indexes:

- the average ratio of the $\ell_\infty$ norm of the optimal perturbation $\delta_j^*$ and that of the sample, $\|x_j\|_\infty$:

$$\frac{1}{\text{wellc}} \sum_j \frac{\|\delta_j^*\|_\infty}{\|x_j\|_\infty}, \tag{10}$$

  which shows how small the perturbation is, on average, w.r.t the perturbed sample;
- the average $\ell_0$ pseudo-norm of $\delta_j^*$, calculated according to (9), to evaluate how much the $\ell_0$ pseudo-norm of the perturbation differs from the target value $k$;
- the average relative violation of both the classification and the $k$-norm constraints of problem (6), indicated as $s_j^{(1)}$ and $s_j^{(2)}$, respectively. This indicator is useful to evaluate the effectiveness of our penalty approach to problem (6). For the classification constraints $y_j w^\top \delta_j \geq y_j \rho_j$ we report also $s_{max}^{(1)}$, the maximum value of the violation.

For each dataset we have selected the best setting of the tuning parameters $\beta$ and $\sigma$. The results are reported in Tables 3 and 4, for the synthetic and the LIBSVM datasets, respectively.

To evaluate the results, we observe first that the average percentage classification error $s_j^{(1)}$ is below 1% in 79 out of a total of 89 cases (both synthetic and real datasets). As for $s_j^{(2)}$, the average percentage error in the $k$-norm constraints, we observe that in general, as expected, it decreases when $k$ increases. In fact this phenomenon accounts for the tradeoff between *small* perturbations and *severe* sparsity constraints, associated to small values of $k$.

The results indicate the effectiveness of the $\ell_0$ pseudo-norm approach to ensure sparse adversarial perturbations. As expected, the more restrictive is the sparsity requirement, the bigger is the infinity norm of the perturbation.

**Table 3**
Results obtained by the sparse perturbation model on synthetic datasets generated using the scikit-learn package for Python.

| *Dataset* | $k$ | Avg. $\frac{\|\delta_j^*\|_\infty}{\|x_j\|_\infty}$ | Avg. $\|\delta_j^*\|_0$ | Avg. $s_j^{(1)}$ (%) | $\max s_j^{(1)}$ (%) | Avg. $s_j^{(2)}$ (%) |
|---|---|---|---|---|---|---|
| | 5 | 0.89 | 4.98 | 0.00015 | 0.00370 | 0.0 |
| 1 | 6 | 0.82 | 5.93 | 0.00203 | 0.14909 | 0.03 |
| | 7 | 0.68 | 7.02 | 0.00037 | 0.02838 | 0.02 |
| $\beta = 5$ | 8 | 0.67 | 7.88 | 0.00452 | 0.54300 | 0.04 |
| $\sigma = 1$ | 9 | 0.63 | 8.58 | 0.00133 | 0.15409 | 0.03 |
| | 10 | 0.61 | 9.29 | 0.00272 | 0.27751 | 0.0 |
| | 15 | 0.45 | 16.25 | 0.08187 | 1.70317 | 6.05 |
| 2 | 18 | 0.37 | 20.05 | 0.04667 | 1.86666 | 5.81 |
| | 21 | 0.35 | 21.43 | 0.55308 | 11.58854 | 3.41 |
| $\beta = 5$ | 24 | 0.38 | 24.55 | 2.79404 | 29.86817 | 3.21 |
| $\sigma = 0.5$ | 27 | 0.42 | 25.80 | 4.00520 | 49.43933 | 1.27 |
| | 30 | 0.49 | 25.04 | 3.22775 | 65.55182 | 0.0 |
| | 5 | 0.65 | 5.04 | 0.00460 | 0.68758 | 0.03 |
| 3 | 6 | 0.59 | 6.00 | 0.00315 | 0.09290 | 0.03 |
| | 7 | 0.52 | 6.88 | 0.00176 | 0.08784 | 0.02 |
| $\beta = 5$ | 8 | 0.48 | 7.80 | 0.00448 | 0.55922 | 0.01 |
| $\sigma = 1$ | 9 | 0.45 | 8.79 | 0.00381 | 0.67791 | 0.0 |
| | 10 | 0.42 | 9.46 | 0.00109 | 0.17895 | 0.0 |
| | 15 | 0.40 | 16.86 | 0.06208 | 2.64140 | 5.90 |
| 4 | 18 | 0.33 | 20.16 | 0.27654 | 9.89262 | 5.73 |
| | 21 | 0.30 | 21.88 | 0.20773 | 10.40370 | 3.59 |
| $\beta = 5$ | 24 | 0.26 | 25.64 | 0.32443 | 13.79725 | 3.40 |
| $\sigma = 0.5$ | 27 | 0.26 | 27.48 | 0.58900 | 15.33243 | 1.27 |
| | 30 | 0.27 | 29.39 | 0.86579 | 17.16624 | 0.0 |
| | 5 | 0.82 | 5.01 | 0.01299 | 0.58852 | 0.05 |
| 5 | 6 | 0.74 | 5.99 | 0.00245 | 0.09832 | 0.02 |
| | 7 | 0.69 | 6.85 | 0.00207 | 0.09623 | 0.03 |
| $\beta = 5$ | 8 | 0.58 | 7.79 | 0.00053 | 0.03778 | 0.01 |
| $\sigma = 1.5$ | 9 | 0.57 | 8.62 | 0.00141 | 0.08188 | 0.03 |
| | 10 | 0.56 | 8.97 | 0.00304 | 0.37367 | 0.0 |
| | 15 | 0.53 | 16.05 | 0.06267 | 2.02768 | 4.78 |
| 6 | 18 | 0.51 | 20.36 | 0.06299 | 4.55998 | 5.83 |
| | 21 | 0.41 | 21.85 | 0.06613 | 2.03794 | 3.55 |
| $\beta = 5$ | 24 | 0.45 | 25.24 | 0.85427 | 24.51329 | 3.29 |
| $\sigma = 0.5$ | 27 | 0.44 | 26.25 | 1.00030 | 19.52448 | 1.48 |
| | 30 | 0.56 | 26.68 | 2.50319 | 54.62499 | 0.0 |
| | 5 | 0.77 | 4.91 | 0.00209 | 0.13033 | 0.03 |
| 7 | 6 | 0.70 | 5.75 | 0.00272 | 0.15844 | 0.02 |
| | 7 | 0.65 | 6.67 | 0.00293 | 0.35313 | 0.02 |
| $\beta = 5$ | 8 | 0.58 | 7.55 | 0.00119 | 0.16090 | 0.01 |
| $\sigma = 1$ | 9 | 0.58 | 8.22 | 0.01430 | 0.80727 | 0.01 |
| | 10 | 0.56 | 8.73 | 0.00145 | 0.18044 | 0.0 |
| | 15 | 0.44 | 14.89 | 0.09823 | 1.80670 | 3.98 |
| 8 | 18 | 0.43 | 18.59 | 0.55711 | 10.30213 | 4.60 |
| | 21 | 0.42 | 21.99 | 1.65538 | 20.95362 | 4.52 |
| $\beta = 5$ | 24 | 0.37 | 24.86 | 2.63614 | 20.07741 | 3.02 |
| $\sigma = 0.5$ | 27 | 0.36 | 26.65 | 4.18073 | 25.68798 | 1.42 |
| | 30 | 0.35 | 26.48 | 3.73799 | 28.19092 | 0.0 |

### 4.2. *Numerical results for the poisoning model*

In this subsection we report the numerical results obtained by applying the classifier perturbation model (8) to the synthetic and real datasets described in Tables 1 and 2.

Each dataset has been randomly split into manipulated and non manipulated samples. The percentage of the former ones is determined by the parameter $p$ selected in the range $\{0.1, 0.3, 0.5\}$. In order to use the DCDA software [40], we have penalized the norm-constraints $\|\delta_i\|^2 \leq \theta_i$ and $\|\delta_l\|^2 \leq \theta_l$ using the same penalty parameter $\varepsilon$. Moreover, for all $i \in I_M$ and $l \in L_M$, we have chosen $\theta_i$ and $\theta_l$ as

$$\theta_i = \theta\|a_i\|_2^2, \quad \theta_l = \theta\|b_l\|_2^2$$

Also in this case we have adopted a parameter grid. In particular we have taken

- $C_1 \in \{10^{-2}, 10^{-1}, 10^1, 10^2\}$,

**Table 4**
Results obtained by the sparse perturbation model on real datasets taken from the LIBSVM repository.

| *Dataset* | $k$ | Avg. $\frac{\|\delta_j^*\|_\infty}{\|x_j\|_\infty}$ | Avg. $\|\delta_j^*\|_0$ | Avg. $s_j^{(1)}$ (%) | max $s_j^{(1)}$ (%) | Avg. $s_j^{(2)}$ (%) |
|---|---|---|---|---|---|---|
| | 7 | 7.16 | 8.00 | 0.00010 | 0.00025 | 6.55 |
| australian | 8 | 6.95 | 9.00 | 0.0 | 0.0 | 5.72 |
| | 10 | 6.31 | 11.00 | 0.0 | 0.0 | 5.02 |
| $\beta = 5$ | 11 | 6.81 | 12.00 | 0.0 | 0.0 | 4.60 |
| $\sigma = 0.05$ | 13 | 6.43 | 13.52 | 0.00043 | 0.00964 | 2.09 |
| | 14 | 5.68 | 14.00 | 0.00186 | 0.08016 | 0.0 |
| | 5 | 2.76 | 4.87 | 0.02008 | 6.27529 | 0.05 |
| breast | 6 | 2.67 | 5.52 | 0.09136 | 4.67346 | 0.05 |
| | 7 | 2.21 | 6.86 | 0.01806 | 5.64580 | 0.05 |
| $\beta = 8$ | 8 | 2.29 | 7.55 | 0.02266 | 0.94199 | 0.06 |
| $\sigma = 5$ | 9 | 2.60 | 8.07 | 0.00815 | 2.39941 | 0.04 |
| | 10 | 1.25 | 9.34 | 0.70987 | 23.45253 | 0.0 |
| | 4 | 1.01 | 4.04 | 0.01188 | 3.53105 | 0.03 |
| diabetes | 5 | 1.55 | 5.07 | 0.00288 | 1.30048 | 0.03 |
| $\beta = 10$ | 6 | 1.26 | 6.00 | 0.00125 | 0.76304 | 0.02 |
| $\sigma = 5$ | 7 | 0.97 | 6.97 | 0.01063 | 1.68377 | 0.0 |
| | 8 | 0.84 | 7.98 | 0.00287 | 0.73366 | 0.0 |
| | 6 | 2.96 | 6.12 | 0.00420 | 0.34518 | 0.09 |
| heart | 8 | 2.51 | 8.17 | 0.00148 | 0.14296 | 0.17 |
| | 9 | 2.35 | 8.93 | 0.00090 | 0.11371 | 0.07 |
| $\beta = 5$ | 10 | 2.29 | 9.50 | 0.00020 | 0.02716 | 0.06 |
| $\sigma = 0.5$ | 12 | 1.98 | 11.76 | 0.00019 | 0.01428 | 0.03 |
| | 13 | 1.84 | 12.54 | 0.00099 | 0.20321 | 0.0 |
| | 17 | 2.92 | 20.75 | 0.33392 | 14.71793 | 6.86 |
| ionosphere | 20 | 2.63 | 22.82 | 0.15472 | 21.14294 | 4.93 |
| | 24 | 2.42 | 25.84 | 0.00619 | 0.58580 | 1.27 |
| $\beta = 10$ | 27 | 2.65 | 28.25 | 0.0 | 0.0 | 0.60 |
| $\sigma = 3$ | 31 | 2.37 | 31.13 | 0.0 | 0.0 | 0.10 |
| | 34 | 2.22 | 33.95 | 0.0 | 0.0 | 0.0 |
| | 30 | 2.38 | 31.55 | 0.03283 | 0.85168 | 2.77 |
| sonar | 36 | 2.67 | 36.73 | 0.07780 | 15.65649 | 2.09 |
| | 42 | 2.30 | 42.17 | 0.07780 | 15.65649 | 0.61 |
| $\beta = 20$ | 48 | 2.30 | 48.13 | 0.07780 | 15.65649 | 0.77 |
| $\sigma = 5$ | 54 | 2.30 | 54.01 | 0.0 | 0.0 | 0.11 |
| | 60 | 1.43 | 60.00 | 0.01831 | 0.52649 | 0.0 |
| | 30 | 1.84 | 40.67 | 3.79595 | 39.45041 | 18.88 |
| splice | 36 | 1.63 | 37.09 | 0.10904 | 22.65131 | 2.88 |
| | 42 | 1.50 | 42.61 | 0.00026 | 0.25964 | 1.88 |
| $\beta = 4.5$ | 48 | 1.41 | 48.96 | 0.30414 | 12.83523 | 1.57 |
| $\sigma = 1$ | 54 | 1.42 | 55.01 | 0.67063 | 11.42268 | 1.11 |
| | 60 | 1.43 | 59.30 | 0.65001 | 10.18288 | 0.0 |

- $C_2 \in \{10^{-1}, 10^1, 10^2\} \cup \{1, 3, 5, 30, 50, 80\}$,
- $\theta \in \{10^{-2}, 10^{-1}, 10^1, 10^2\}$,
- $\varepsilon \in \{10^{-2}, 10^{-1}, 10^1, 10^2\}$.

We have considered the following indexes involving the original hyperplane $(w', \gamma')$ and the solution $(w^*, \gamma^*)$ to problem (8) which shows the quality of the solution to the optimization problem:

- the absolute value of the scalar product of the optimal normal $w^*$ and that of the original hyperplane $w'$ ($| w^* \cdot w' |$);
- the cosine of the angle $\varphi$ between $w^*$ and $w'$ ($\cos(\varphi)$);
- The average relative violation of the constraints of problem (8), indicated with $Avg\, s_i$ and $Avg\, s_l$, respectively.

The first two indexes provide information on the position of the poisoned hyperplane $w^*$ with respect to the original hyperplane $w'$.

The following indexes, instead, are defined by considering also the hyperplane $(\tilde{w}, \tilde{\gamma})$ obtained by training SVM on the dataset $\tilde{\mathcal{X}}$ which allow to estimate the effectiveness of our poisoning strategy:

- the classification accuracies (in percentage) obtained on the non manipulated points with respect to the original hyperplane $w'$ ($Acc_{w'}^{nm}$) and the hyperplane $\tilde{w}$ ($Acc_{\tilde{w}}^{nm}$) obtained by training SVM on the dataset $\tilde{\mathcal{X}}$;
- the euclidean norm of the difference between $\tilde{w}$ and $w^*$ ($\|\tilde{w} - w^*\|$), which measures the variation between the hyperplane $\tilde{w}$ and $w^*$.

**Table 5**
Results obtained by the classifier perturbation model on synthetic datasets generated using the scikit-learn package for Python.

| Dataset | $p$ | $\lvert w^* \cdot w' \rvert$ | $\cos(\varphi)$ | Avg. $s_I$ | Avg. $s_L$ | $Acc^{nm}_{w'}$ | $Acc^{nm}_{\tilde{w}}$ | $\lVert \tilde{w} - w^* \rVert$ |
|---|---|---|---|---|---|---|---|---|
| 1 | 0.05 | 0.0 | 0.0 | 0.0 | 0.0 | 100.00 | 71.40 | 0.266 |
|  | 0.10 | 0.0 | −0.0 | 0.0 | 0.0 | 93.30 | 73.30 | 1.141 |
|  | 0.15 | 0.0 | 0.0 | 0.0 | 0.0 | 95.50 | 81.80 | 0.828 |
| 2 | 0.05 | 0.0 | 0.0 | 0.0 | 0.0 | 85.70 | 71.40 | 0.534 |
|  | 0.10 | 0.0 | −0.0 | 0.0 | 0.0 | 86.70 | 86.70 | 0.878 |
|  | 0.15 | 0.0 | −0.0 | 0.0 | 0.0 | 90.90 | 72.70 | 0.564 |
| 3 | 0.05 | 0.0 | 0.0 | 0.185 | 0.133 | 80.00 | 80.00 | 0.706 |
|  | 0.10 | 0.0 | −0.001 | 0.004 | 0.0 | 80.00 | 56.69 | 0.809 |
|  | 0.15 | 0.0 | −0.001 | 0.0 | 0.0 | 82.19 | 64.40 | 0.555 |
| 4 | 0.05 | 0.001 | −0.002 | 0.0 | 0.0 | 86.70 | 60.00 | 0.470 |
|  | 0.10 | 0.0 | 0.0 | 0.0 | 0.0 | 80.00 | 63.30 | 0.665 |
|  | 0.15 | 0.0 | −0.0 | 0.0 | 0.0 | 77.80 | 66.70 | 0.818 |
| 5 | 0.05 | 0.0 | 0.0 | 0.0 | 0.0 | 100.00 | 71.40 | 0.375 |
|  | 0.10 | 0.0 | −0.0 | 0.0 | 0.0 | 100.00 | 80.00 | 0.829 |
|  | 0.15 | 0.0 | −0.001 | 0.0 | 0.0 | 100.00 | 81.80 | 0.536 |
| 6 | 0.05 | 0.0 | 0.001 | 0.0 | 0.016 | 100.00 | 71.40 | 0.754 |
|  | 0.10 | 0.0 | 0.0 | 0.0 | 0.0 | 100.00 | 66.70 | 1.385 |
|  | 0.15 | 0.0 | 0.0 | 0.0 | 0.0 | 100.00 | 77.30 | 0.612 |
| 7 | 0.05 | 0.0 | 0.0 | 0.0 | 0.0 | 93.30 | 80.00 | 0.689 |
|  | 0.1 | 0.0 | 0.0 | 0.0 | 0.0 | 90.00 | 73.30 | 0.584 |
|  | 0.15 | 0.0 | 0.001 | 0.0 | 0.0 | 93.30 | 77.80 | 1.056 |
| 8 | 0.05 | 0.0 | 0.0 | 0.0 | 0.0 | 93.30 | 66.70 | 0.561 |
|  | 0.1 | 0.0 | −0.001 | 0.0 | 0.0 | 93.30 | 73.30 | 0.632 |
|  | 0.15 | 0.0 | −0.0 | 0.0 | 0.008 | 91.10 | 68.89 | 0.915 |

**Table 6**
Results obtained by the classifier perturbation model on real datasets taken from the LIBSVM repository.

| Dataset | $p$ | $\lvert w^* \cdot w' \rvert$ | $\cos(\varphi)$ | Avg. $s_I$ | Avg. $s_L$ | $Acc^{nm}_{w'}$ | $Acc^{nm}_{\tilde{w}}$ | $\lVert \tilde{w} - w^* \rVert$ |
|---|---|---|---|---|---|---|---|---|
| australian | 0.05 | 0.003 | −0.001 | 0.0 | 0.0 | 76.50 | 52.90 | 1.376 |
|  | 0.10 | 0.004 | −0.005 | 0.0 | 0.0 | 81.20 | 50.70 | 0.298 |
|  | 0.15 | 0.002 | −0.003 | 0.0 | 0.0 | 78.60 | 53.40 | 0.248 |
| breast | 0.05 | 0.0 | −0.001 | 0.0 | 0.006 | 91.20 | 64.70 | 0.252 |
|  | 0.10 | 0.0 | −0.0 | 0.001 | 0.005 | 94.10 | 61.80 | 0.311 |
|  | 0.15 | 0.0 | 0.0 | 0.0 | 0.0 | 96.10 | 62.70 | 0.231 |
| diabetes | 0.05 | 0.001 | −0.0 | 1.582 | 1.624 | 81.60 | 63.20 | 1.184 |
|  | 0.10 | 0.0 | 0.0 | 31.150 | 43.990 | 80.30 | 69.69 | 1.456 |
|  | 0.15 | 0.003 | 0.001 | 1.831 | 1.801 | 79.10 | 67.80 | 0.772 |
| heart | 0.05 | 0.0 | −0.0 | 0.0 | 0.0 | 92.30 | 53.80 | 0.774 |
|  | 0.10 | 0.0 | −0.0 | 0.042 | 0.060 | 92.60 | 59.30 | 0.438 |
|  | 0.15 | 0.0 | 0.0 | 0.0 | 0.0 | 92.50 | 57.49 | 0.597 |
| ionosphere | 0.05 | 0.001 | 0.0 | 0.024 | 0.010 | 94.10 | 64.70 | 0.634 |
|  | 0.10 | 0.0 | −0.0 | 0.0 | 0.0 | 91.40 | 74.30 | 0.430 |
|  | 0.15 | 0.0 | 0.0 | 0.380 | 0.151 | 94.19 | 88.50 | 3.059 |
| sonar | 0.05 | 0.001 | −0.0 | 0.0 | 0.0 | 100.00 | 50.00 | 0.305 |
|  | 0.10 | 0.0 | 0.0 | 0.002 | 0.0 | 100.00 | 85.00 | 1.304 |
|  | 0.15 | 0.0 | −0.0 | 0.012 | 0.003 | 100.00 | 48.40 | 0.431 |
| splice | 0.05 | 0.0 | −0.0 | 0.0 | 0.0 | 86.00 | 54.00 | 0.354 |
|  | 0.10 | 0.001 | −0.001 | 0.146 | 0.0 | 86.00 | 53.00 | 1.470 |
|  | 0.15 | 0.001 | 0.001 | 0.193 | 0.216 | 86.70 | 55.30 | 0.701 |

The results are reported in Tables 5 and 6 and in Tables 7 and 8 we have reported the optimal values for the hyper-parameters $C_1, C_2, \theta$ and $\varepsilon$ used for each dataset.

The Tables 5 and 6 show that, for both synthetic and real datasets, the constraint violation is negligible (apart from the Diabetes dataset). The new separating hyperplane is almost orthogonal to the one provided by the SVM model. Moreover, the accuracy obtained on the non manipulated points with respect to the new hyperplane $(\tilde{w}, \tilde{\gamma})$ is significantly lower than the one obtained with respect to the hyperplane $(w', \gamma')$. Such poisoning effect, as expected, grows with $p$, the percentage of manipulated points. In Table 9, finally, we have reported, for the synthetic datasets, the average accuracy loss (in percentage) on the non manipulated points for every value of the parameter $p$ we have considered. This table shows that, on average for the synthetic datasets, there is a loss in accuracy for the non manipulated points between 17% and 20%.

**Table 7**
Optimal hyper-parameters found for the classifier perturbation model on synthetic datasets.

| Dataset | $p$ | $C_1$ | $C_2$ | $\theta$ | $\varepsilon$ |
|---|---|---|---|---|---|
| | 0.05 | 0.01 | 1.0 | 0.10 | 0.10 |
| 1 | 0.10 | 0.01 | 100.0 | 0.10 | 0.10 |
| | 0.15 | 0.01 | 100.0 | 0.10 | 0.10 |
| | 0.05 | 0.01 | 3.0 | 0.10 | 0.10 |
| 2 | 0.10 | 0.01 | 30.0 | 100.00 | 0.01 |
| | 0.15 | 0.01 | 50.0 | 10.00 | 0.01 |
| | 0.05 | 0.01 | 5.0 | 0.01 | 0.01 |
| 3 | 0.10 | 0.01 | 80.0 | 0.10 | 0.01 |
| | 0.15 | 0.01 | 3.0 | 0.10 | 0.10 |
| | 0.05 | 0.01 | 80.0 | 10.00 | 0.01 |
| 4 | 0.10 | 0.01 | 50.0 | 0.01 | 0.10 |
| | 0.15 | 0.01 | 100.0 | 0.10 | 0.10 |
| | 0.05 | 0.01 | 50.0 | 0.01 | 0.10 |
| 5 | 0.10 | 0.01 | 50.0 | 0.01 | 0.10 |
| | 0.15 | 0.01 | 30.0 | 0.10 | 0.10 |
| | 0.05 | 0.01 | 80.0 | 0.01 | 0.01 |
| 6 | 0.10 | 0.01 | 50.0 | 10.00 | 0.10 |
| | 0.15 | 0.01 | 80.0 | 0.10 | 0.01 |
| | 0.05 | 0.01 | 80.0 | 0.10 | 0.01 |
| 7 | 0.10 | 0.01 | 30.0 | 0.10 | 0.10 |
| | 0.15 | 0.01 | 100.0 | 10.00 | 0.01 |
| | 0.05 | 0.01 | 80.0 | 10.00 | 0.01 |
| 8 | 0.10 | 0.01 | 80.0 | 0.10 | 0.10 |
| | 0.15 | 0.01 | 30.0 | 0.01 | 0.01 |

**Table 8**
Optimal hyper-parameters found for the classifier perturbation model on real datasets.

| Dataset | $p$ | $C_1$ | $C_2$ | $\theta$ | $\varepsilon$ |
|---|---|---|---|---|---|
| | 0.05 | 0.01 | 80.0 | 10.00 | 0.01 |
| australian | 0.10 | 0.01 | 80.0 | 0.10 | 0.10 |
| | 0.15 | 0.01 | 80.0 | 0.10 | 0.10 |
| | 0.05 | 0.01 | 100.0 | 0.01 | 0.10 |
| breast | 0.10 | 0.01 | 100.0 | 0.01 | 0.10 |
| | 0.15 | 0.01 | 100.0 | 100.00 | 0.01 |
| | 0.05 | 0.01 | 3.0 | 0.10 | 0.01 |
| diabetes | 0.10 | 0.10 | 80.0 | 0.01 | 0.01 |
| | 0.15 | 0.01 | 80.0 | 0.10 | 0.01 |
| | 0.05 | 0.01 | 100.0 | 0.10 | 0.10 |
| heart | 0.10 | 0.01 | 100.0 | 0.10 | 0.01 |
| | 0.15 | 0.01 | 5.0 | 10.00 | 0.01 |
| | 0.05 | 0.01 | 30.0 | 0.01 | 0.10 |
| ionosphere | 0.10 | 0.01 | 80.0 | 10.00 | 0.01 |
| | 0.15 | 0.01 | 50.0 | 0.10 | 0.01 |
| | 0.05 | 0.01 | 50.0 | 10.00 | 0.01 |
| sonar | 0.10 | 0.01 | 100.0 | 0.10 | 0.01 |
| | 0.15 | 0.01 | 100.0 | 0.10 | 0.01 |
| | 0.05 | 0.01 | 100.0 | 0.10 | 0.01 |
| splice | 0.10 | 0.10 | 30.0 | 0.10 | 0.01 |
| | 0.15 | 0.01 | 30.0 | 0.01 | 0.01 |

**Table 9**
Average accuracy loss on non manipulated points for every value of $p$.

| $p$ | Avg. Accuracy loss |
|---|---|
| 0.05 | 20.83 |
| 0.10 | 18.75 |
| 0.15 | 17.42 |

**Table 10**
Details on the large real datasets we have considered.

| Dataset | $m$ | $n$ | $\|w'\|_0$ | wellc | misc |
|---|---|---|---|---|---|
| MP_carcinoma | 36 | 7457 | 6863 | 36 | 0 |
| MP_DLBCL | 77 | 7129 | 6864 | 77 | 0 |
| MP_tumor1 | 60 | 7129 | 6936 | 53 | 7 |

**Table 11**
Results obtained by the classifier perturbation model on large real datasets.

| Dataset | $p$ | $\|w^* \cdot w'\|$ | $\cos(\varphi)$ | Avg. $s_I$ | Avg. $s_L$ | $Acc_{w'}^{nm}$ | $Acc_{\tilde{w}}^{nm}$ | $\|\tilde{w} - w^*\|$ |
|---|---|---|---|---|---|---|---|---|
| MP_carcinoma | 0.05 | 0.0 | −0.001 | 0.0 | 0.0 | 100.00 | 0.0 | 0.843 |
| | 0.10 | 0.0 | −0.006 | 0.0 | 0.0 | 100.00 | 66.70 | 0.549 |
| | 0.15 | 0.0 | −0.005 | 0.0 | 0.0 | 100.00 | 60.00 | 0.665 |
| MP_DLBCL | 0.05 | 0.0 | 0.0 | 0.0 | 0.0 | 100.00 | 66.70 | 1.779 |
| | 0.10 | 0.002 | −0.002 | 0.0 | 0.0 | 100.00 | 57.09 | 3.163 |
| | 0.15 | 0.009 | 0.016 | 0.0 | 0.0 | 100.00 | 54.50 | 2.397 |
| MP_tumor1 | 0.05 | 0.0 | 0.001 | 0.0 | 0.0 | 100.00 | 100.00 | 0.416 |
| | 0.10 | 0.0 | 0.0 | 0.0 | 0.0 | 100.00 | 66.70 | 0.581 |
| | 0.15 | 0.0 | 0.001 | 0.0 | 0.0 | 100.00 | 66.70 | 0.457 |

**Table 12**
Optimal hyper-parameters found for the classifier perturbation model on large real datasets.

| Dataset | $p$ | $C_1$ | $C_2$ | $\theta$ | $\varepsilon$ |
|---|---|---|---|---|---|
| MP_carcinoma | 0.05 | 0.01 | 10.0 | 10.0 | 0.01 |
| | 0.10 | 0.01 | 100.0 | 10.0 | 0.10 |
| | 0.15 | 0.01 | 80.0 | 0.1 | 0.10 |
| MP_DLBCL | 0.05 | 0.01 | 80.0 | 100.0 | 0.10 |
| | 0.10 | 0.01 | 5.0 | 10.0 | 0.01 |
| | 0.15 | 0.01 | 3.0 | 100.0 | 0.10 |
| MP_tumor1 | 0.05 | 10.00 | 3.0 | 0.1 | 0.01 |
| | 0.10 | 0.01 | 30.0 | 10.0 | 0.10 |
| | 0.15 | 0.01 | 5.0 | 10.0 | 0.10 |

### *4.3. Some results on large datasets*

We have tested our adversarial approach also on some high dimension benchmark datasets (large number of features and relatively small number of samples) [31,43]. The details are given in Table 10.

We have focussed only on the poisoning model. In fact the sparse evasion model is not very much significant for the datasets under consideration, where the separating hyperplane in the SVM model is *naturally* sparse [31,43].

We report the results in Table 11, while in Table 12 the best hyper-parameters found after the grid-search phase are given.

The results obtained confirm those related to smaller dimension datasets The hyperplane $w^*$ is almost perpendicular to the initial one $w'$, the average constraints violation is zero and there is a significant decrease in the accuracy score on the non manipulated points.

## 5. Conclusions

We have introduced two models (evasion and poisoning, respectively) for adversarial learning in the SVM framework. Both models require solution of nonconvex and nonsmooth optimization problems of the DC type. The results indicate that adversarial modelling can benefit from the use of such advanced optimization techniques.

## Data availability

The data used are open source.

## Acknowledgments

The first author was supported by a grant awarded by European Union Next Generation EU under the PRIN 2022 project: SMOTION (ID 2022 EAECWJ), CUP H53D23002000006.

The fourth author is member of the Gruppo Nazionale per l'Analisi Matematica, la Probabilità e le loro Applicazioni-Istituto Nazionale di Alta Matematica (GNAMPA-INdAM) and was supported by PON R&I 2014–2020 (CUP F25F21002210003).

## References

[1] N. Dalvi, P. Domingos, Mausam, S. Sumit, D. Verma, Adversarial classification, in: KDD-2004 – Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2004, pp. 99–108.

[2] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, R. Fergus, Intriguing properties of neural networks, in: International Conference on Learning Representations, (ICLR), 2014.

[3] I.J. Goodfellow, J. Shlens, C. Szegedy, Explaining and harnessing adversarial examples, in: 3rd International Conference on Learning Representations, (ICLR), 2015.

[4] L. Muñoz-González, E.C. Lupu, The Security of Machine Learning Systems, Springer, Berlin, 2018.

[5] M. Barreno, B. Nelson, A.D. Joseph, J.D. Tygar, The security of machine learning, Mach. Learn. 81 (2) (2010) 121–148.

[6] B. Biggio, F. Roli, Wild patterns: Ten years after the rise of adversarial machine learning, Pattern Recognit. 84 (2018) 317–331.

[7] A. Kurakin, I.J. Goodfellow, S. Bengio, Adversarial machine learning at scale, in: 5th International Conference on Learning Representations, (ICLR), 2017.

[8] J.B. Rosen, Pattern separation by convex programming, J. Math. Anal. Appl. 10 (1965) 123–134.

[9] O.L. Mangasarian, Multisurface method of pattern separation, IEEE Trans. Inform. Theory 14 (6) (1968) 801–807.

[10] L. Grippo, A class of unconstrained minimization methods for neural network training, Optim. Methods Softw. 4 (2) (1994) 135–150.

[11] S. Sra, S. Nowozin, S.J. Wright, Optimization for Machine Learning, MIT Press, Cambridge (Mass.), 2012.

[12] V. Vapnik, The Nature of the Statistical Learning Theory, Springer Verlag, 1995.

[13] N. Cristianini, J. Shawe-Taylor, 20 0 0. an Introduction to Support Vector Machines and Other Kernel-Based Learning Methods, Cambridge University Press, 2000.

[14] A. Astorino, M. Gaudioso, Ellipsoidal separation for classification problems, Optim. Methods Softw. 20 (2005) 261–270.

[15] A. Astorino, A. Frangioni, E. Gorgone, B. Manca, Ellipsoidal classification via semidefinite programming, Oper. Res. Lett. 51 (2) (2023) 197–203.

[16] A. Astorino, M. Gaudioso, A fixed-center spherical separation algorithm with kernel transformations for classification problems, Comput. Manag. Sci. 6 (2009) 357–372.

[17] A. Astorino, M. Gaudioso, E. Gorgone, D. Pallaschke, Data preprocessing in semi-supervised SVM classification, Optimization 60 (1–2) (2011) 143–151.

[18] A. Astorino, M. Gaudioso, A. Seeger, Conic separation of finite sets I. The homogeneous case, J. Convex Anal. 21 (1) (2014) 1–28.

[19] A. Astorino, A. Fuduli, Support vector machine polyhedral separability in semisupervised learning, J. Optim. Theory Appl. 164 (2015) 1039–1050.

[20] A. Astorino, M. Di Francesco, M. Gaudioso, E. Gorgone, B. Manca, Polyhedral separation via difference of convex (DC) programming, Soft Comput. 25 (19) (2021) 12605–12613.

[21] L. Xu, K. Crammer, D. Schuurmans, Robust support vector machine training via convex outlier ablation, in: Proceedings of the National Conference on Artificial Intelligence, vol. 1, 2006, pp. 536–542.

[22] H. Xu, C. Caramanis, S. Mannor, Robustness and regularization of support vector machines, J. Mach. Learn. Res. 10 (2009) 1485–1510.

[23] B. Biggio, B. Nelson, P. Laskov, Support vector machines under adversarial label noise, J. Mach. Learn. Res. 20 (2011) 97–112.

[24] B. Biggio, et al., Evasion attacks against machine learning at test time, in: H. Blockeel, K. Kersting, S. Nijssen, F. Železný (Eds.), Machine Learning and Knowledge Discovery in Databases, ECML PKDD 2013, in: Lecture Notes in Computer Science, vol. 8190, Springer, Berlin, Heidelberg, 2013.

[25] Y. Zhou, M. Kantarcioglu, B. Thuraisingham, B. Xi, Adversarial support vector machine learning, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2012, pp. 1059–1067.

[26] B. Biggio, B. Nelson, P. Laskov, Poisoning attacks against support vector machines, in: Proceedings of the 29th International Conference on Machine Learning (ICML), vol. 2, 2012, pp. 1807–1814.

[27] Z. Yin, F. Wang, W. Liu, S. Chawla, Sparse feature attacks in adversarial learning, IEEE Trans. Knowl. Data Eng. 30 (6) (2018) 1164–1177.

[28] F. Zhang, et al., Adversarial feature selection against evasion attacks, IEEE Trans. Cybern. 46 (3) (2016) 766–777.

[29] G.A. Watson, Linear best approximation using a class of polyhedral norms, Numer. Algorithms (1992) 2321–336.

[30] J.ì. Gotoh, A. Takeda, K. Tono, DC formulations and algorithms for sparse optimization problems, Math. Program., Ser. B 169 (1) (2018) 141–176.

[31] M. Gaudioso, E. Gorgone, J.-B. Hiriart-Urruty, Feature selection in SVM via polyhedral $k$-norm, Optim. Lett. 14 (2019) 19–36.

[32] M. Gaudioso, J.-B. Hiriart-Urruty, Deforming $\|.\|_1$ into $\|.\|_\infty$ via polyhedral norms: A pedestrian approach, SIAM Rev. 64 (3) (2022) 713–727.

[33] M. Gaudioso, G. Giallombardo, G. Miglionico, Sparse optimization via vector $k$-norm and DC programming with an application to feature selection for support vector machines, Comput. Optim. Appl. 86 (2) (2023) 746–766.

[34] M. Gaudioso, G. Giallombardo, J.-B. Hiriart-Urruty, Dual formulation of the sparsity constrained optimization problem: application to classification, Optim. Methods Softw. (2023).

[35] J.-B. Hiriart-Urruty, Generalized Differentiability/ Duality and Optimization for Problems Dealing with Differences of Convex Functions, in: Lecture Notes in Economic and Mathematical Systems, vol. 256, Springer Verlag, 1986, pp. 37–70.

[36] A.S. Strekalovsky, Global optimality conditions for nonconvex optimization, J. Global Optim. 12 (1998) 415–434.

[37] H.A. Le Thi, T. Pham Dinh, The DC (difference of convex functions) programming and DCA revisited with DC models of real world nonconvex optimization problems, Ann. Oper. Res. 133 (1–4) (2005) 23–46.

[38] M. Gaudioso, G. Giallombardo, G. Miglionico, A.M. Bagirov, Minimizing nonsmooth DC functions via successive DC piecewise-affine approximations, J. Global Optim. 71 (1) (2018) 37–55.

[39] M. Bagirov, M. Gaudioso, N. Karmitsa, M. Mäkeä, S. Taheri, Numerical Nonsmooth Optimization, State of the Art Algorithms, Springer, Berlin, 2020.

[40] P. d'Alessandro, M. Gaudioso, G. Giallombardo, G. Miglionico, The descent-ascent algorithm for DC programming, INFORMS J. Comput. 36 (2) (2024) 657–671.

[41] F. Pedregosa, et al., Scikit-learn: machine learning in python, J. Mach. Learn. Res. 12 (2011) 2825–2830, r.

[42] C.C. Chang, J. Lin, LIBSVM: A library for support vector machines, ACM Trans. Intell. Syst. Technol. (TIST) 2 (3) (2011) 1–27.

[43] M. Gaudioso, E. Gorgone, M. Labbé, A.M. Rodríguez-Chía, Lagrangian relaxation for SVM feature selection, Comput. Oper. Res. 87 (2017) 137–145.