Optimal Feature Manipulation Attacks Against Linear Regression

Fuwei Li, Lifeng Lai, and Shuguang Cui

Abstract—**In this paper, we investigate how to manipulate the coefficients obtained via linear regression by adding carefully designed poisoning data points to the dataset or modifying the original data points. Given the energy budget, we first provide the closed-form solution of the optimal poisoning data point when our target is modifying one designated regression coefficient. We then extend the analysis to a more challenging scenario where the attacker aims to change one particular regression coefficient while making others to be changed as small as possible. For this scenario, we introduce a semidefinite relaxation method to design the best attack scheme. Finally, we study a more powerful adversary who can perform a rank-one modification on the feature matrix. We propose an alternating optimization method to find the optimal rank-one modification matrix. Numerical examples are provided to illustrate the analytical results obtained in this paper.**

Index Terms—**Linear regression, adversarial robustness, poisoning attack, non-convex optimization.**

I. Introduction

Linear regression plays a fundamental role in machine learning and is used in a wide spectrum of applications [2]–[6]. In linear regression, one assumes that there is a simple linear relationship between the explanatory variables and the response variable. The goal of linear regression is to find out the regression coefficients through the methods of ordinary least square (OLS), ridge regression, Lasso [7], etc. Having the regression coefficients learned from the data points, one can predict the response values given the values of the explanatory variables. The regression coefficients also help us explain the variation in the response variable that can be attributed to the variation in the explanatory variables. They can quantify the strength of the relationship between certain explanatory variables and the response variable. A large magnitude of the regression coefficient usually indicates a strong relationship, while a small valued regression coefficient means a weak relationship. This is especially true when linear regression is accomplished by the parameter regularized method such as ridge regression and Lasso. In addition, the sign of the regression coefficient indicates whether the value of the response variable increases or decreases when the value of an explanatory variable changes, which is very important in biologic science [8], financial analysis [9], and environmental science [10].

Machine learning is being used in various applications, including security and safety critical applications such as medical image analysis [11] and autonomous driving [12]. For these applications, it is important to understand the robustness of machine learning algorithms in adversarial environments [13]–[18]. In such an environment, there may exist a malicious adversary. Depending on the adversary's knowledge about the data samples, the learning algorithm, and the defense strategy of the learning system, the adversary can carry out white-box, grey-box, and black-box attacks. In the white-box attack, the adversary has the full knowledge of the machine learning system and has the ability to observe the whole data points. After seeing the data points, the adversary can add some carefully designed poisoning data points or directly modify the data points so as to corrupt the learning system or leave a backdoor in this system [19]. If the adversary knows nothing about the data samples, learning algorithms, and defense strategies, the adversary can also carry out black-box attacks, where it gains information of the system by repeatedly sending queries to the system [20]. If the adversary only has partial knowledge of the data samples, learning algorithms, and defense strategies, the adversary can perform grey-box attacks, in which it uses surrogate data samples or classifiers to mimic the original ones [21]. In this paper, we focus on the white-box attacks.

The goal of this paper is to investigate the optimal way to attack linear regression methods. In the considered linear regression system, there exists an adversary who can observe the whole dataset and then inject carefully designed poisoning data points or directly modify the original dataset in order to manipulate the regression coefficients. The manipulated regression coefficients can later be used by the adversary as a backdoor of this learning system or mislead our interpretation of the linear regression model. For example, changing the magnitude of a regression coefficient to be small makes us believe that its corresponding explanatory variable is irrelevant. Similarly, the adversary can change the magnitude of a regression coefficient to a larger value to increase its importance. Furthermore, changing the sign of a regression coefficient can also lead us to misinterpret the correlation between its explanatory and response variables.

Fuwei Li and Lifeng Lai are with the Department of Electrical and Computer Engineering, University of California, Davis, CA, 95616 (e-mail: fli@ucdavis.edu; lflai@ucdavis.edu). Shuguang Cui is currently with the School of Science and Engineering, Shenzhen Research Institute of Big Data and Future Network of Intelligence Institute (FNii), the Chinese University of Hong Kong, Shenzhen, China, 518172 (e-mail: shuguangcui@cuhk.edu.cn).

The work of F. Li and L. Lai was supported in part by the National Science Foundation with Grants CCF-1717943, CCF-1908258 and ECCS-2000415. The work of S. Cui was supported in part by the Key Area R&D Program of Guangdong Province with grant No. 2018B030338001, by Shenzhen Outstanding Talents Training Fund 202002, and by Guangdong Research Projects No. 2017ZT07X152 and No. 2019CX01X104. This paper was presented in part at IEEE International Workshop on Machine Learning for Signal Processing, Espoo, Finland, Sep. 2020 [1].

Depending on the objective of the adversary and the way the adversary changes the regression coefficients, we have different problem formulations. We first consider a scenario where the adversary tries to manipulate one specific regression coefficient by adding one carefully designed poisoning data point that has a limited energy budget to the dataset. We show that finding the optimal attack data point is equivalent to solve an optimization problem where the objective function is a ratio of two quadratic functions with a quadratic inequality constraint. Even though this type of problem is non-convex in general, our particular problem has a hidden convex structure. With the help of this convex structure, we further convert the optimization problem into a quadratic constrained quadratic program (QCQP). Since strong duality exists in this problem [22], we manage to identify its closed-form optimal solutions from its Karush-Kuhn-Tucker (KKT) conditions.

We next consider a more sophisticated objective where the attacker aims to change one particular regression coefficient while making others be changed as small as possible. We show that finding the optimal attack data point is equivalent to solving an optimization problem where the objective function is a ratio of two fourth order multivariate polynomials with a quadratic inequality constraint. This optimization problem is much more complicated than the optimization above. We introduce a semidefinite relaxation method to solve this problem. The numerical examples show that we can find the globally optimal solutions with a very low relaxation order. Hence, the complexity of this method is low in practical problems.

Finally, we consider a more powerful adversary who can directly modify the feature matrix. Particularly, we consider a rank-one modification attack [23], where the attacker carefully designs a rank-one matrix and adds it to the existing data matrix. A rank-one modification attack is general enough to capture the most common modifications, such as modifying one feature, deleting or adding one data point, changing one entry of the data matrix, etc. Hence, studying the rank-one modification provides us universal bounds on these kinds of attacks. By leveraging the rank-one structure, we develop an alternating optimization method to find the optimal modification matrix. We also prove that the solution obtained by the proposed optimization method is one of the critical points of the optimization problem.

Our study is related to several recent works on adversarial machine learning. For example, Pimentel-Alarcón et al. studied how to add one adversarial data point in order to maximize the error of the subspace estimated by principal component [24] and Li et al. derived a closed-form optimal modification to the original dataset in order to maximize the subspace distance between the original one the one after modification [23]. These two works focused on the robustness of subspace learning algorithms that are based on principal component analysis (PCA). PCA is an unsupervised learning method. By contrast, we study the robustness of linear regression, which is a supervised learning method. Alfeld et al. studied how to manipulate the training data to increase the validation or test error for the linear regression task [25], [26] and Biggio et al. used a gradient based algorithm to design one poisoning data point with the aim of worsening the testing error in a support vector machine (SVM) learning system, and they also proposed a heuristic approach to flip parts of the training labels in order to achieve a similar goal [27], [28]. These works aimed to deteriorate the performance on a specific data set. However, we concentrate on the explanation of the linear regression model. By manipulating the regression coefficient, we can mislead the interpretation of the dependency between the features and response value. Furthermore, a series of works focused on the adversarial robustness of deep learning networks. Kurakin et al. proposed a gradient based method to design adversarial noise [13], [14], [29]. By adding this noise to the test data, it makes the machine learning system make the wrong prediction. By contrast, we focus on adding or modifying training data samples to maneuver the regression coefficient. Biggio et al. corrupted the deep learning system by inserting delicately designed poisoning data samples into the training data [19], [30], [31]. Due to the complexity of deep neural networks, it is hard to know whether the designed poisoning data samples are optimal. Nevertheless, our method is proven to be optimal with respect to certain specific goals discussed in this paper.

In addition, there are recent works that focus on the adversarial robustness of machine learning in various other applications. For example, Kwon et al. proposed a gradient based method to generate adversarial audio examples [32], Li et al. presented an ensemble method to enhance the robustness of the malware detection system against adversarial attacks [18], and Flowers et al. demonstrated the vulnerability of communication systems against adversarial noises [33]. These works are limited to their specific applications. In our paper, we target maneuvering the interpretation of a general linear regression model by adding poisoning data points or modifying the original data.

The work that is most relevant to our paper is [34], where the authors develop a bi-level optimization framework to design the attack matrix. [34] used the projected gradient descent method to solve the bi-level optimization problem. However, a general bi-level problem is known to be NP-hard, and solving it depends on the convexity of the lower level problem. In addition, the convergence of projected gradient descent for a non-convex problem is not clear. Compared with [34], we obtain the globally optimal solution to the case for adding one poisoning data point, and we also prove that the proposed alternating optimization method converges to one of the critical points for the case where the attacker can perform a rank-one modification attack. Furthermore, for the projected gradient descent method, different datasets need different parameters, which means we must do parameter tuning before applying this algorithm. By contrast, we provide a closed-form solution to the case for adding one poisoning data point to attack one of the regression coefficients, and the designed alternating optimization method for the case of rank-one attack does not need parameter tuning. Furthermore, compared with the projected gradient descent method, our alternating optimization method provides smaller objective values, faster convergence rate, and more stable behavior.

The remainder of this paper is organized as follows. In Section II, we consider the scenario where the attacker adds one carefully designed poisoning data point to the dataset. In Section III, we investigate the rank-one attack strategy. Numerical examples are provided in Section IV to illustrate the results we obtained in this paper. Finally, we provide concluding remarks in Section V.

II. Attacking with one adversarial data point

In this section, we consider the scenario where the attacker can add one carefully crafted data point to the existing dataset. We will extend the analysis to the case with more sophisticated attacks in Section III.

A. Problem formulation

Consider a dataset with n data samples, $\{y_i, \mathbf{x}_i\}_{i=1}^n$, where y_i is the response variable, $\mathbf{x}_i \in \mathbb{R}^m$ is the feature vector, where each component of $\mathbf{x}_i$ represents an explanatory variable. In this section, we consider an adversarial setup in which the adversary first observes the whole dataset $\{\mathbf{y}, \mathbf{X}\}$, in which $\mathbf{y} := [y_1, y_2, \dots, y_n]^\top$ and $\mathbf{X} := [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^\top$, and then carefully designs an adversarial data point, $\{y_0, \mathbf{x}_0\}$, and adds it into the existing data samples. After inserting this adversarial data point, we have the poisoned dataset $\{\hat{\mathbf{y}}, \hat{\mathbf{X}}\}$, where $\hat{\mathbf{y}} := [y_0, y_1, y_2, \dots, y_n]^\top$, $\hat{\mathbf{X}} := [\mathbf{x}_0, \mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n]^\top$.

From the dataset, we intend to learn a linear regression model. From the poisoned dataset, the learned model is obtained by solving

$$\underset{\boldsymbol{\beta}}{\text{argmin}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \tag{1}$$

where $\|\cdot\|$ denotes the ℓ_2 norm for a vector and the induced 2-norm for a matrix throughout this paper. Let $\hat{\boldsymbol{\beta}}$ be the optimal solution to problem (1). The goal of the adversary is to minimize some objective function, $f(\hat{\boldsymbol{\beta}})$, by carefully designing the adversarial data point. The form of $f(\hat{\boldsymbol{\beta}})$ depends on the specific goal of the attacker. For example, the attacker can try to reduce the importance of feature i by setting $f(\hat{\boldsymbol{\beta}}) = |\hat{\beta}_i|$, in which $\hat{\beta}_i$ is the ith component of $\hat{\boldsymbol{\beta}}$. Or the attacker can try to increase the importance of feature i by setting $f(\hat{\boldsymbol{\beta}}) = -|\hat{\beta}_i|$. To make the problem meaningful, in this paper, we impose the energy constraint on the adversarial data point. Since one data point contains a feature vector and a response value, we put ℓ_2 norm constraint on the concatenated vector $[\mathbf{x}_0^\top, y_0]^\top$. With the objective $f(\hat{\boldsymbol{\beta}})$ and the energy constraint of the adversary data point, our problem can be formulated as

$$\begin{aligned} \min_{\|[\mathbf{x}_0^\top, y_0]\| \le \eta} &: \quad f(\hat{\boldsymbol{\beta}}) \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \end{aligned} \tag{2}$$

where η is the energy budget. The objective function, $f(\hat{\boldsymbol{\beta}})$, depends on the poisoning data point, $\{\mathbf{x}_0, y_0\}$, not in a direct way, but through a lower level optimization problem. What makes this problem even harder is the complication of the objective function. Depending on the goal of the adversary, the objective can be in various forms. In the following two subsections, we will discuss two important objectives and their solutions, respectively. The methods and insights obtained from these two cases could then be extended to cases with other objectives.

B. Attacking one regression coefficient

In this subsection, the goal of the adversary is to design the adversarial data point $\{y_0, \mathbf{x}_0\}$ to decrease (or increase) the importance of a certain explanatory variable. If the goal is to decrease the importance of explanatory variable i, we can set $f(\hat{\boldsymbol{\beta}}) = |\hat{\beta}_i|$, and the optimization problem can be written as

$$\begin{aligned} \min_{\|[\mathbf{x}_0^\top, y_0]\|_2 \le \eta} &: \quad |\hat{\beta}_i| \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2. \end{aligned} \tag{3}$$

Similarly, if the goal of the adversary is to increase the importance of the explanatory variable i, we can set our objective as

$$\min : -|\hat{\beta}_i| \tag{4}$$

withe the same constraints as in problem (3).

To solve the optimization problems (3) and (4), we first solve the following two optimization problems

$$\min_{\|[\mathbf{x}_0^\top, y_0]\| \le \eta} : \hat{\beta}_i \tag{5}$$

$$\text{s.t.} \quad \hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \tag{6}$$

and

$$\max_{\|[\mathbf{x}_0^\top, y_0]\| \le \eta} : \hat{\beta}_i \tag{7}$$

$$\text{s.t.} \quad \hat{\boldsymbol{\beta}} = \min_{\boldsymbol{\beta}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2. \tag{8}$$

It is easy to check that the solutions to problems (3) and (4) can be obtained from the solutions to problem (5) and (7). In particular, let $(\hat{\beta}_i^*)_{\min}$ and $(\hat{\beta}_i^*)_{\max}$ be optimal values of problem (5) and (7) respectively. Then, if $\hat{\beta}_i \ge 0$, we can check that $\max\{0, (\hat{\beta}_i^*)_{\min}\}$ and $\max\{|(\hat{\beta}_i^*)_{\min}|, |(\hat{\beta}_i^*)_{\max}|\}$ are the solutions to problem (3) and (4) respectively. Similar arguments can be made if $\hat{\beta}_i < 0$.

In the following, we will focus on solving the minimization problem (5). The solution to the maximization problem (7) can be obtained by using a similar approach. To solve this bi-level optimization problem, we can first solve the optimization problem in the subjective. Assume $\mathbf{X}$ is full column rank. Problem (6) is just an ordinary least squares problem, which has a simple closed-form solution: $\hat{\boldsymbol{\beta}} = (\hat{\mathbf{X}}^\top\hat{\mathbf{X}})^{-1}\hat{\mathbf{X}}^\top\hat{\mathbf{y}}$. Substitute in $\hat{\mathbf{X}} = [\mathbf{x}_0, \mathbf{X}^\top]^\top$ and $\hat{\mathbf{y}} = [y_0, \mathbf{y}^\top]^\top$, and we have

$$\hat{\boldsymbol{\beta}} = (\mathbf{X}^\top\mathbf{X} + \mathbf{x}_0\mathbf{x}_0^\top)^{-1}[\mathbf{x}_0, \mathbf{X}^\top][y_0, \mathbf{y}^\top]^\top.$$

According to the Sherman-Morrison formula [35], we have

$$(\mathbf{X}^\top\mathbf{X} + \mathbf{x}_0\mathbf{x}_0^\top)^{-1} = \mathbf{A} - \frac{\mathbf{A}\mathbf{x}_0\mathbf{x}_0^\top\mathbf{A}}{1 + \mathbf{x}_0^\top\mathbf{A}\mathbf{x}_0}, \tag{9}$$

where

$$\mathbf{A} = (\mathbf{X}^\top\mathbf{X})^{-1}. \tag{10}$$

The inverse of $\mathbf{X}^\top\mathbf{X} + \mathbf{x}_0\mathbf{x}_0^\top$ always exists because $1 + \mathbf{x}_0^\top\mathbf{A}\mathbf{x}_0 \neq 0$ and $\mathbf{X}^\top\mathbf{X}$ is invertible. Plug this inverse in the expression of $\hat{\boldsymbol{\beta}}$, we get

$$\hat{\boldsymbol{\beta}} = \boldsymbol{\beta}_0 + \frac{\mathbf{A}\mathbf{x}_0(y_0 - \mathbf{x}_0^\top\boldsymbol{\beta}_0)}{1 + \mathbf{x}_0^\top\mathbf{A}\mathbf{x}_0}, \tag{11}$$

where

$$\boldsymbol{\beta}_0 = \mathbf{A}\mathbf{X}^\top \mathbf{y}. \tag{12}$$

We can observe that $\boldsymbol{\beta}_0$ is the coefficient that is obtained from the clean data. Problem (5) is equivalent to

$$\begin{aligned} \min_{\mathbf{x}_0, y_0} : \quad & \frac{\mathbf{a}^\top \mathbf{x}_0 (y_0 - \mathbf{x}_0^\top \boldsymbol{\beta}_0)}{1 + \mathbf{x}_0^\top \mathbf{A}\mathbf{x}_0} \\ \text{s.t.} \quad & \|[\mathbf{x}_0^\top, y_0]\| \le \eta, \end{aligned} \tag{13}$$

where $\mathbf{a}$ is the ith column of $\mathbf{A}$. The optimization problem (13) is the ratio of two quadratic functions with a quadratic constraint. To further simplify this optimization problem, we can write our objective and subjective in a more compact form by performing variable change: $\mathbf{u} = [\mathbf{x}_0^\top, y_0]^\top$. Using this compact representation, the optimization problem (13) can be written as

$$\begin{aligned} \min_{\mathbf{u}} : \quad & \frac{\frac{1}{2}\mathbf{u}^\top \mathbf{H}\mathbf{u}}{1 + \mathbf{u}^\top [\begin{smallmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & 0 \end{smallmatrix}]\mathbf{u}} \\ \text{s.t.} \quad & \mathbf{u}^\top \mathbf{u} \le \eta^2, \end{aligned} \tag{14}$$

in which

$$\mathbf{H} = \begin{bmatrix} -\mathbf{a}\boldsymbol{\beta}_0^\top - \boldsymbol{\beta}_0 \mathbf{a}^\top & \mathbf{a} \\ \mathbf{a}^\top & 0 \end{bmatrix}. \tag{15}$$

(14) is a non-convex optimization problem. To solve this problem, we employ the technique introduced in [36]. We first perform variable change $\mathbf{u} = \frac{\mathbf{z}}{s}$ by introducing variable $\mathbf{z}$ and scalar s. Inserting this into problem (14), adding constraint 1 to the denominator of the objective and moving it to the subjective, we have a new optimization problem

$$\begin{aligned} \min_{\mathbf{z}, s} : \quad & \frac{1}{2}\mathbf{z}^\top \mathbf{H}\mathbf{z} && (16) \\ \text{s.t.} \quad & s^2 + \mathbf{z}^\top [\begin{smallmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & 0 \end{smallmatrix}]\mathbf{z} = 1, && (17) \\ & \mathbf{z}^\top \mathbf{z} \le s^2 \eta^2. && (18) \end{aligned}$$

To validate the equivalence between problem (14) and (16), we only need to check if the optimal value of problem (14) is less than the optimal value of problem (16) when $s = 0$ [36]. Firstly, since $\mathbf{H}$ is not positive semi-definite (which will be shown later), the optimal value of problem (14) is less than zero. Secondly, when $s = 0$, the optimal value of problem (16) is zero, which is apparently larger than the optimal value of problem (14). Therefore, the two problems are equivalent.

To solve problem (16), we substitute s^2 in equation (17) for that in equation (18) and obtain

$$\begin{aligned} \min_{\mathbf{z}} : \quad & \frac{1}{2}\mathbf{z}^\top \mathbf{H}\mathbf{z} && (19) \\ \text{s.t.} \quad & \frac{1}{2}\mathbf{z}^\top \mathbf{D}\mathbf{z} \le \eta^2, && (20) \end{aligned}$$

where

$$\mathbf{D} = 2\left(\mathbf{I} + \eta^2 \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}\right). \tag{21}$$

Notice that $\mathbf{H}$ is not positive semi-definite; hence problem (19) is not a standard convex QCQP problem [22]. However, it is proved that strong duality holds for this type of problem [22], [37]. Hence, to solve this problem, we can start by investigating its KKT necessary conditions. The Lagrangian of problem (19) is

$$\mathcal{L}(\mathbf{z}, \lambda) = \frac{1}{2}\mathbf{z}^\top \mathbf{H}\mathbf{z} + \lambda\left(\frac{1}{2}\mathbf{z}^\top \mathbf{D}\mathbf{z} - \eta^2\right),$$

where λ is the dual variable. According to the KKT conditions, we have

$$\begin{aligned} (\mathbf{H} + \lambda \mathbf{D})\,\mathbf{z} = \mathbf{0}, && (22) \\ \frac{1}{2}\mathbf{z}^\top \mathbf{D}\mathbf{z} \le \eta^2, && (23) \\ \lambda\left(\frac{1}{2}\mathbf{z}^\top \mathbf{D}\mathbf{z} - \eta^2\right) = 0, && (24) \\ \lambda \ge 0. && (25) \end{aligned}$$

By inspecting the complementary slackness condition (24), we consider two cases based on the value of λ.
Case 1: $\lambda = 0$. In this case, we must have $\mathbf{H}\mathbf{z} = \mathbf{0}$ according to (22). As a result, the objective value of (19) is zero, which contradicts the fact that the optimal value should be negative. Hence, this case is not possible.
Case 2: $\lambda > 0$. In this case, equality in (23) must hold based on (24). According to the stationary condition (22), if the matrix $\mathbf{H} + \lambda \mathbf{D}$ is full rank, we must have $\mathbf{z} = \mathbf{0}$, for which equality in (23) cannot hold. Hence, $\mathbf{H} + \lambda \mathbf{D}$ is not full-rank and we have $\det(\mathbf{H} + \lambda \mathbf{D}) = 0$. As $\mathbf{D}$ is positive definite, we also have $\det(\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2} + \lambda \mathbf{I}) = 0$. Since $\lambda > 0$, this equality tells us that $-\lambda$ belongs to one of the negative eigenvalues of $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$. In the following, we will show that $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$ has one and only one negative eigenvalue.

By definition, $\mathbf{D}$ is a block diagonal matrix. Hence, its inverse is also block diagonal. Let us define $\mathbf{D}^{-1/2} = \text{diag}\{\mathbf{G}, g\}$, where $\mathbf{G} = 1/\sqrt{2}(\mathbf{I} + \eta^2 \mathbf{A})^{-1/2}$ and $g = 1/\sqrt{2}$. Thus, we have

$$\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2} = \begin{bmatrix} -\mathbf{c}\mathbf{h}^\top - \mathbf{h}\mathbf{c}^\top & g\mathbf{c} \\ g\mathbf{c}^\top & 0 \end{bmatrix},$$

where $\mathbf{c} = \mathbf{G}\mathbf{a}$ and $\mathbf{h} = \mathbf{G}\boldsymbol{\beta}_0$. Define ξ as one eigenvalue of $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$, and compute its eigenvalues by computing the characteristic polynomial:

$$\begin{aligned} & \det\left(\xi \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}\right) \\ & = \xi^{m-1}\left(\xi^2 + 2\xi \mathbf{c}^\top \mathbf{h} + \mathbf{c}^\top \mathbf{h}\mathbf{h}^\top \mathbf{c} - g^2 \mathbf{c}^\top \mathbf{c} - \mathbf{c}^\top \mathbf{c}\mathbf{h}^\top \mathbf{h}\right). \end{aligned}$$

Thus, the eigenvalues of $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$ are $\xi = 0$ ($(m-1)$ multiplicities) and $\xi = -\mathbf{c}^\top \mathbf{h} \pm \|\mathbf{c}\|\sqrt{g^2 + \mathbf{h}^\top \mathbf{h}}$. Since $\|\mathbf{c}\|\sqrt{g^2 + \mathbf{h}^\top \mathbf{h}} > |\mathbf{c}^\top \mathbf{h}|$, the eigenvalues of $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$ satisfy: $\xi_{m+1} < 0, \quad \xi_m = \xi_{m-1} = \cdots = \xi_2 = 0, \quad \xi_1 > 0$. Now, it is clear that $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$ has one and only one negative eigenvalue and one positive eigenvalue, respectively. Thus, we have $\lambda = -\xi_{m+1}$. Assume $\boldsymbol{\nu}_1$ and $\boldsymbol{\nu}_{m+1}$ are two eigenvectors corresponding to eigenvalues ξ_1 and ξ_{m+1}. Through simple calculation, we have

$$\boldsymbol{\nu}_i = k_i\left[-\frac{\mathbf{c}^\top \mathbf{h} + \xi_i}{\mathbf{c}^\top \mathbf{c}}\mathbf{c}^\top + \mathbf{h}^\top, \frac{g\mathbf{c}^\top}{\xi_i}\left(-\frac{\mathbf{c}^\top \mathbf{h} + \xi_i}{\mathbf{c}^\top \mathbf{c}}\mathbf{c} + \mathbf{h}\right)\right]^\top, \tag{26}$$

Algorithm 1 Optimal Adversarial Data Point Design

```
Input: the data set, $\{y_i, \mathbf{x}_i\}_{i=1}^n$, energy budget $\eta$, and the index of feature to be attacked.
Steps:
compute $\mathbf{A}$ according to equation (10), compute $\boldsymbol{\beta}_0$ according to (12).
compute $\mathbf{H}$ and $\mathbf{D}$ according to (15) and (21), respectively.
compute the smallest eigenvalue, $\xi_{m+1}$, of $\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}$ and its corresponding eigenvector according to (26).
design the adversarial data point, $\{\mathbf{x}_0, y_0\}$, according to equations (27), (28), and (29).
Output: return the optimal adversarial data point $\{\mathbf{x}_0, y_0\}$ and the optimal value $\eta^2\xi_{m+1} + (\boldsymbol{\beta}_0)_i$.
```

where $i = 1$, $m+1$ and scalar k_i is the normalization constant to guarantee the eigenvectors to be of unit length. According to (22), we have

$$(\mathbf{H} + \lambda\mathbf{D})\,\mathbf{z} = \mathbf{D}^{1/2}\left(\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2} + \lambda\mathbf{I}\right)\mathbf{D}^{1/2}\mathbf{z} = 0;$$

thus the solution to problem (19) is

$$\mathbf{z}^* = k \cdot \mathbf{D}^{-1/2}\boldsymbol{\nu}_{m+1}. \tag{27}$$

Since $\frac{1}{2}\mathbf{z}^\top\mathbf{D}\mathbf{z} = \eta^2$, we have $k = \sqrt{2}\eta$. Having the expression of the optimal $\mathbf{z}^*$, we can then compute s according to equation (17):

$$s = \pm\sqrt{1 - (\mathbf{z}^*_{1:m})^\top\mathbf{A}\,\mathbf{z}^*_{1:m}}, \tag{28}$$

where $\mathbf{z}^*_{1:m}$ is the vector that comprises the first m elements of $\mathbf{z}^*$. Hence, the corresponding solution to problem (13) is

$$\mathbf{x}_0^* = \mathbf{z}^*_{1:m}/s, \quad y_0^* = z^*_{m+1}/s. \tag{29}$$

We now compute the optimal value of problem (16). Since our objective function is $\frac{1}{2}(\mathbf{z}^*)^\top\mathbf{H}\mathbf{z}^*$, substituting $\mathbf{z}^*$ in (27) leads to the objective value: $\eta^2\boldsymbol{\nu}_{m+1}^\top\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}\boldsymbol{\nu}_{m+1}$. Since $\boldsymbol{\nu}_{m+1}^\top\mathbf{D}^{-1/2}\mathbf{H}\mathbf{D}^{-1/2}\boldsymbol{\nu}_{m+1} = \xi_{m+1}$, our optimal objective value is $\eta^2\xi_{m+1}$.

Following similar analysis as above, we can find the optimal $\mathbf{z}^*$ for problem (7), which is $\mathbf{z}^* = \sqrt{2}\eta\mathbf{D}^{-1/2}\boldsymbol{\nu}_1$. Also, we can compute the optimal $\mathbf{x}_0^*$ and y_0^* according to equation (29) and its optimal objective value, which is $\eta^2\xi_1$.

In summary, the optimal values for problems (5) and (7) are $\eta^2\xi_{m+1} + (\boldsymbol{\beta}_0)_i$ and $\eta^2\xi_1 + (\boldsymbol{\beta}_0)_i$ respectively. We have summarized the process to design the optimal adversarial data point in Algorithm 1 with respect to objective (5) and the process with respect to objective (7) can be obtained accordingly. Based on our optimal values of problems (5) and (7), we can further decide the optimal values of problems (3) and (4) as discussed at the beginning of this section. From our analysis we can see that the main computation is to compute $\mathbf{A}$ in (10). Hence, the complexity of our algorithm is $\mathcal{O}(m^3)$.

Moreover, if we use the ridge regression method in linear regression, there is only a slight difference in the matrix $\mathbf{A}$ in problem (13) and the whole analysis remains the same.

One may concern that the proposed adversarial data point may behave as an outlier and can be easily detected by the learning system. We can mitigate this by a simple repeating strategy, in which we repeat the proposed adversarial data point K times and shrink the magnitude of these poisoning data by $\sqrt{K}$. This can be simply verified by

$$\begin{aligned}\hat{\boldsymbol{\beta}} &= (\hat{\mathbf{X}}^\top\hat{\mathbf{X}})^{-1}\hat{\mathbf{X}}\hat{\mathbf{y}}\\ &= \left(\mathbf{X}^\top\mathbf{X} + \mathbf{x}_0\mathbf{x}_0^\top\right)^{-1}\left(\mathbf{X}^\top\mathbf{y} + \mathbf{x}_0 y_0\right)\\ &= \left(\mathbf{X}^\top\mathbf{X} + \sum_{i=1}^{k}\frac{1}{\sqrt{K}}\mathbf{x}_0\frac{1}{\sqrt{K}}\mathbf{x}_0^\top\right)^{-1}\Bigg(\mathbf{X}^\top\mathbf{y}\\ &\quad + \sum_{i=1}^{K}\frac{1}{\sqrt{K}}\mathbf{x}_0\frac{1}{\sqrt{K}}y_0\Bigg)\\ &= (\tilde{\mathbf{X}}^\top\tilde{\mathbf{X}})^{-1}\tilde{\mathbf{X}}^\top\tilde{\mathbf{y}},\end{aligned}$$

where $\tilde{\mathbf{X}} = [\mathbf{X}^\top, \underbrace{\frac{1}{\sqrt{K}}\mathbf{x}_0, \ldots, \frac{1}{\sqrt{K}}\mathbf{x}_0}_{K \text{ times}}]^\top$ and $\tilde{\mathbf{y}} = [\mathbf{y}^\top, \underbrace{\frac{1}{\sqrt{K}}y_0, \ldots, \frac{1}{\sqrt{K}y_0}}_{K \text{ times}}]^\top$. By shrinking the poisoning data points, it will make the detection of these points more difficult, especially when the dataset is standardized.

We now analyze the impact of parameters, such as η, on the objective value. Even though we have a closed-form solution to the optimal adversarial data point, the objective is a complex function of the original dataset. Hence, it will be difficult to analyze this for the general case. Instead, we will focus on some special cases. In particular, we analyze how the energy budget affects the value of objective function in the large data sample scenario. As our analysis shows, our optimal values are $\eta^2\xi$, where $\xi = -\mathbf{c}^\top\mathbf{h} \pm \|\mathbf{c}\|\sqrt{g^2 + \mathbf{h}^\top\mathbf{h}}$, $\mathbf{c} = \mathbf{G}\mathbf{a}$, $\mathbf{h} = \mathbf{G}\boldsymbol{\beta}_0$, $\mathbf{G} = 1/\sqrt{2}(\mathbf{I} + \eta^2\mathbf{A})^{-1/2}$, $g = 1/\sqrt{2}$, $\mathbf{A} = (\mathbf{X}^\top\mathbf{X})^{-1}$, and $\boldsymbol{\beta}_0$ is the original regression coefficient. In the large data sample limit and the assumption that the features are independent and standardized, we have the approximation $\mathbf{A} = \mathbf{I}$. Recall that $\mathbf{a}$ is the ith column of $\mathbf{A}$, $\mathbf{a} = \mathbf{e}_i$. As the result, the objective value is $\eta^2\xi = \frac{1}{2}\frac{\eta^2}{1+\eta^2}\left[-\boldsymbol{\beta}_0^i \pm \sqrt{\eta^2 + 1 + ||\boldsymbol{\beta}_0||^2}\right]$. For objective (5) with optimal value $\frac{1}{2}\frac{\eta^2}{1+\eta^2}\left[-\boldsymbol{\beta}_0^i - \sqrt{\eta^2 + 1 + ||\boldsymbol{\beta}_0||^2}\right]$, this function is monotonically decreasing with η. For the objective (7) with optimal value $\frac{1}{2}\frac{\eta^2}{1+\eta^2}\left[-\boldsymbol{\beta}_0^i + \sqrt{\eta^2 + 1 + ||\boldsymbol{\beta}_0||^2}\right]$, it is a monotonically increasing function of η.

C. Attacking with small changes of other regression coefficients

In Section II-B, we have discussed how to design the adversarial data points to attack one specific regression coefficient. However, as we only focus on one particular regression coefficient, other regression coefficients may also be changed as well. In this subsection, we consider a more complex objective function, where we aim to make the changes to other regression coefficients to be as small as possible while attacking one of the regression coefficients.

Suppose our objective is to minimize the ith regression coefficient (the scenario of maximize the ith regression coefficient can be solved using similar approach), i.e., to minimize $\|\hat{\beta}_i\|^2$.

Algorithm 2 Optimal Adversarial Data Point Design while Making Small Changes to Other Regression Coefficients

```
**Input**: the data set, $\{y_i, \mathbf{x}_i\}_{i=1}^n$, energy budget $\eta$, and the index of feature to be attacked, the trade-off parameter $\lambda$.
**Steps**:
compute $\mathbf{A}$ according to equation (10), compute $\boldsymbol{\beta}_0$ according to (12), compute $\mathbf{A}_2$ according to (32).
follow the steps (30), (31), (33), and (34), and formulate our problem as a polynomial optimization problem (37).
use Lasserre's relaxation method to solve problem (37) and get the optimal solution $\mathbf{x}^*$ and optimal value $p^*$.
compute $\mathbf{w}^* = \mathbf{U}^\top \mathbf{x}^*$, where $\mathbf{I} + \eta^2 \mathbf{A}_2 = \mathbf{U}\mathbf{U}^\top$.
compute $s^* = \pm\sqrt{1 - (\mathbf{w}^*)^\top \mathbf{A}_2 \mathbf{w}^*}$.
calculate the optimal solution $\mathbf{x}_0^* = \mathbf{w}_{1:m}^*/s^*, y_0^* = w_{m+1}^*/s^*$.
**Output**: return the optimal adversarial data point $\{y_0^*, \mathbf{x}_0^*\}$ and the optimal value $p^*$.
```

At the same time, we would also like to minimize the changes to the rest of the regression coefficients, i.e., to minimize $\|\boldsymbol{\beta}_0^{-i} - \hat{\boldsymbol{\beta}}^{-i}\|^2$, where $\boldsymbol{\beta}_0^{-i} = [\beta_0^1, \ldots, \beta_0^{i-1}, 0, \beta_0^{i+1}, \ldots, \beta_0^m]^\top$ and $\hat{\boldsymbol{\beta}}^{-i} = [\hat{\beta}_1, \ldots, \hat{\beta}_{i-1}, 0, \hat{\beta}_{i+1}, \hat{\beta}_m]^\top$. Combine the two objectives, we have our new objective function

$$f(\hat{\boldsymbol{\beta}}) = \frac{1}{2}\left\|\boldsymbol{\beta}_0^{-i} - \hat{\boldsymbol{\beta}}^{-i}\right\|^2 + \frac{\lambda}{2}\left\|\hat{\beta}_i\right\|^2,$$

where λ is the trade-off parameter. The larger the λ is, the more effort will be made to keep the ith regression coefficient small. A negative λ means the adversary attempts to make the magnitude of the ith regression coefficient large. Again, we assume that the attack energy budget is η. As the result, we have the following optimization problem

$$\begin{aligned} \min_{\|[\mathbf{x}_0^\top, y_0]\| \le \eta} : \quad & \frac{1}{2}\left\|\boldsymbol{\beta}_0^{-i} - \hat{\boldsymbol{\beta}}^{-i}\right\|^2 + \frac{\lambda}{2}\left\|\hat{\beta}_i\right\|^2 \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2. \end{aligned} \tag{30}$$

As the objective function is a quadratic function with respect to $\hat{\boldsymbol{\beta}}$, we can write it in a more compact form: $\frac{1}{2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0^{-i})^\top \boldsymbol{\Lambda}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0^{-i})$, where $\boldsymbol{\Lambda} = \text{diag}(1, 1, \ldots, \lambda, \ldots, 1)$ and λ is at the ith coordinate. With this compact form, our optimization problem can be written as

$$\begin{aligned} \min_{\|[\mathbf{x}_0^\top, y_0]\| \le \eta} : \quad & \frac{1}{2}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0^{-i})^\top \boldsymbol{\Lambda}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}_0^{-i}) \\ \text{s.t.} \quad & \hat{\boldsymbol{\beta}} = \operatorname*{argmin}_{\boldsymbol{\beta}} : \|\hat{\mathbf{y}} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2. \end{aligned} \tag{31}$$

To solve this problem, same as in the previous subsection, we start by solving the lower level optimization problem. Since we have the same lower level problem as in (5), substitute $\hat{\boldsymbol{\beta}}$ in the objective with the expression (11), and we have the one level optimization problem

$$\begin{aligned} \min_{\mathbf{x}_0, y_0} : \quad & \frac{1}{2}\mathbf{g}^\top \boldsymbol{\Lambda} \mathbf{g} \\ \text{s.t.} \quad & \left\|[\mathbf{x}_0^\top, y_0]\right\| \le \eta, \end{aligned}$$

where $\mathbf{g} = \frac{\mathbf{A}\mathbf{x}_0(y_0 - \mathbf{x}_0^\top \boldsymbol{\beta}_0)}{1 + \mathbf{x}_0^\top \mathbf{A}\mathbf{x}_0} - \mathbf{b}$ with $\mathbf{A}$ and $\boldsymbol{\beta}_0$ defined in (10) and (12) respectively and $\mathbf{b} = \boldsymbol{\beta}_0^{-i} - \boldsymbol{\beta}_0$. To further simplify our problem, let us define

$$\mathbf{A}_1 = [\mathbf{A}, \mathbf{0}],\ \mathbf{A}_2 = \begin{bmatrix} \mathbf{A} & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix},\ \mathbf{c} = \begin{bmatrix} -\boldsymbol{\beta}_0 \\ 1 \end{bmatrix},\ \mathbf{z} = \begin{bmatrix} \mathbf{x}_0 \\ y_0 \end{bmatrix}, \tag{32}$$

where $\mathbf{A}_1 \in \mathbb{R}^{m \times (m+1)}$ and $\mathbf{A}_2 \in \mathbb{R}^{(m+1) \times (m+1)}$. With the new defined variables, we can write our problem more compactly as:

$$\begin{aligned} \min_{\mathbf{z}} : \quad & \frac{1}{2}\left(\frac{\mathbf{A}_1 \mathbf{z}\mathbf{c}^\top \mathbf{z}}{1 + \mathbf{z}^\top \mathbf{A}_2 \mathbf{z}} - \mathbf{b}\right)^\top \boldsymbol{\Lambda}\left(\frac{\mathbf{A}_1 \mathbf{z}\mathbf{c}^\top \mathbf{z}}{1 + \mathbf{z}^\top \mathbf{A}_2 \mathbf{z}} - \mathbf{b}\right) \\ \text{s.t.} \quad & \|\mathbf{z}\| \le \eta. \end{aligned} \tag{33}$$

Since the objective is a ratio of two quartic functions, similar to the process we carried out from (14) to (16), we perform variable change $\mathbf{z} = \frac{\mathbf{w}}{s}$ by introducing the new variable $\mathbf{w}$ and scalar s. Insert it into problem (33) and follow the same argument we have made to transform problem (14) to problem (16), problem (33) is equivalent to the following problem

$$\begin{aligned} \min_{\mathbf{w}, s} : \quad & \frac{1}{2}\left(\mathbf{A}_1 \mathbf{w}\mathbf{c}^\top \mathbf{w} - \mathbf{b}\right)^\top \boldsymbol{\Lambda}\left(\mathbf{A}_1 \mathbf{w}\mathbf{c}^\top \mathbf{w} - \mathbf{b}\right) && (34) \\ \text{s.t.} \quad & (s^2 + \mathbf{w}^\top \mathbf{A}_2 \mathbf{w})^2 = 1, && (35) \\ & \mathbf{w}^\top \mathbf{w} \le s^2 \eta^2. && (36) \end{aligned}$$

According to the definition of $\mathbf{A}_2$, it is positive semidefinite. Hence, we have $s^2 = 1 - \mathbf{w}^\top \mathbf{A}_2 \mathbf{w}$. Plug in the expression of s^2 into (36), the constraints in problem (34) can be simplified to $\mathbf{w}^\top(\mathbf{I} + \eta^2 \mathbf{A}_2)\mathbf{w} \le \eta^2$. Let $\mathbf{U}^\top \mathbf{U} = \mathbf{I} + \eta^2 \mathbf{A}_2$ be the Cholesky decomposition of $\mathbf{I} + \eta^2 \mathbf{A}_2$. Define $\mathbf{H} = \mathbf{A}_1 \mathbf{U}^{-1}$, $\mathbf{e} = \mathbf{U}^{-\top}\mathbf{c}$, and $\mathbf{x} = \mathbf{U}\mathbf{w}$, we can simplify problem (34) further as:

$$\begin{aligned} \min_{\mathbf{x}} : \quad & \frac{1}{2}\left(\mathbf{H}\mathbf{x}\mathbf{e}^\top \mathbf{x} - \mathbf{b}\right)^\top \boldsymbol{\Lambda}\left(\mathbf{H}\mathbf{x}\mathbf{e}^\top \mathbf{x} - \mathbf{b}\right) \\ \text{s.t.} \quad & \mathbf{x}^\top \mathbf{x} \le \eta^2. \end{aligned} \tag{37}$$

This is an optimization problem with a quartic objective function and with a quadratic constraint. Recent progress in multivariate polynomial optimization has made it possible to solve this problem using the sum of squares technology [38]–[41]. This method finds the globally optimal solutions by solving a sequence of convex linear matrix inequality problems. Even though this sequence might be infinitely long, in practice, a very short sequence is enough to guarantee its global optimality. Hence, in this subsection, we will resort to Lasserre's relaxation method [38]. Algorithm 2 summarizes the process to design the adversarial data point. The complexity of Algorithm 2 is dominant by the solving of the relaxation semidefinite problem. Hence, the computational complexity of Algorithm 2 is $\mathcal{O}(s(N)^{4.5})$, where N is the relaxation order and $s(N) = \binom{N+m}{N}$ [42]. Numerical examples using this method to solve our problem with real data will be provided in Section IV.

In this subsection, we put an ℓ_2 norm constraint on the adversarial data point. It is possible to extend our work to

TABLE I
CONFIGURATIONS OF $\mathbf{c}$ AND $\mathbf{d}$ AND THEIR CORRESPONDING MODIFICATIONS.

Modification	Configurations of $\mathbf{c}$ and $\mathbf{d}$
delete the ith data sample	$\mathbf{c} = -\mathbf{e}_i$, $\mathbf{b} = \mathbf{X}_{i,:}$
delete feature i	$\mathbf{c} = \mathbf{X}^\top_{:,i}$, $\mathbf{d} = -\mathbf{e}_i$
add one adversarial data sample	$\mathbf{X} \leftarrow [\mathbf{X}, \mathbf{0}]$, $\mathbf{c} = \mathbf{e}_{n+1}$, $\mathbf{d} = \mathbf{x}^\top_{n+1}$
modify one entry	$\mathbf{c} = \eta \cdot \mathbf{e}_i$, $\mathbf{d} = \mathbf{e}_j$

other kinds of norm constraints, such as ℓ_1 and ℓ_∞ norm constraints. Suppose we put ℓ_p ($p = 1$ or $p = \infty$) norm constraint on the adversarial data sample with objective (30), following similar steps in this subsection, we can obtain objective (34) with constraint (35) and the norm cone constraint $||\mathbf{w}||_p \leq s\eta$. When $p = 1$, the norm cone constraint can be transformed to the inequalities constraints $\sum_{i=1}^{m+1} a_i \leq s\eta$ and $-a_i \leq w_i \leq a_i$ for $i = 1, \ldots, m+1$, where a_i is the auxiliary variable. When $p = \infty$, we can transform the norm cone constraint to $b \leq s\eta$ and $-b\mathbf{1} \preccurlyeq \mathbf{w} \preccurlyeq b\mathbf{1}$, where b is a auxiliary variable. Both cases lead to linear inequality constraints, which are special polynomial inequalities. Hence, we can still use the Lasserre's relaxation method to obtain the optimal solution.

III. RANK-ONE ATTACK ANALYSIS

In Section II, we have discussed how to design one adversarial data point to attack the regression coefficients. In this section, we consider a more powerful adversary who can modify the whole dataset in order to attack the regression coefficients. In particular, we will consider a rank-one attack on the feature matrix [23]. This type of attack covers many practical scenarios, for example, modifying one entry of the feature matrix, deleting one feature, changing one feature, replacing one feature, etc. We summarize the these modifications and their corresponding configurations of $\mathbf{c}$ and $\mathbf{d}$ in Table I, where $\mathbf{cd}^\top$ is the rank one modification matrix, $\mathbf{X}_{i,:}$ denotes the ith row of the feature matrix $\mathbf{X}$, $\mathbf{X}_{:,i}$ represents the ith column of the feature matrix, $\mathbf{e}_i$ is the standard basis vector, and η is the scalar which denotes the modification energy budget. Hence, the analysis of the rank-one attack provides a universal bound for all of these kinds of modifications. Specifically, we will consider the objective in problem (3) and (4) where the adversary attacks one particular regression coefficient. In the following, we will first formulate our problem and then provide our alternating optimization method to solve this problem.

In the considered rank one attack model, the attacker will carefully design a rank-one feature modification matrix $\mathbf{\Delta}$ and add it to the original feature matrix $\mathbf{X}$. As the result, the modified feature matrix is $\hat{\mathbf{X}} = \mathbf{X} + \mathbf{\Delta}$. As $\mathbf{\Delta}$ has rank one, we can write $\mathbf{\Delta} = \mathbf{cd}^\top$, where $\mathbf{c} \in \mathbb{R}^n$ and $\mathbf{d} \in \mathbb{R}^m$. Similar to the previous section, we restrict the adversary to having constrained energy budget, η. Here, we use the Frobenius norm to measure the energy of the modification matrix. Hence, we have $\|\mathbf{\Delta}\|_\mathrm{F} \leq \eta$, where $\|\cdot\|_\mathrm{F}$ denotes the Frobenius norm of a matrix. If the attacker's goal is to increase the importance of feature i, our problem can be written as

$$\begin{aligned} \max_{\|\mathbf{cd}^\top\|_\mathrm{F} \leq \eta} \ :& \quad |\hat{\beta}_i| \\ \text{s.t.} & \quad \hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}} \, \|\mathbf{y} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \\ & \quad \hat{\mathbf{X}} = \mathbf{X} + \mathbf{cd}^\top. \end{aligned} \tag{38}$$

If the adversary is trying to minimize the magnitude of the ith regression coefficient, our problem is

$$\begin{aligned} \min_{\|\mathbf{cd}^\top\|_\mathrm{F} \leq \eta} \ :& \quad |\beta_i| \\ \text{s.t.} & \quad \hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}} : \ \|\mathbf{y} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \\ & \quad \hat{\mathbf{X}} = \mathbf{X} + \mathbf{cd}^\top. \end{aligned} \tag{39}$$

Similar to Section II-B, the solutions to problems (38) and (39) can be obtained by the solutions to the following two problems:

$$\max_{\|\mathbf{cd}^\top\|_\mathrm{F} \leq \eta} \ : \quad \hat{\beta}_i \tag{40}$$

and

$$\min_{\|\mathbf{cd}^\top\|_\mathrm{F} \leq \eta} \ : \quad \hat{\beta}_i \tag{41}$$

with the same constraints as in (38) and (39).

We can further write the above two problems in a more unified form:

$$\begin{aligned} \min_{\|\mathbf{cd}^\top\|_\mathrm{F} \leq \eta} \ :& \quad \mathbf{e}^\top \hat{\boldsymbol{\beta}} \\ \text{s.t.} & \quad \hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\text{argmin}} : \ \|\mathbf{y} - \hat{\mathbf{X}}\boldsymbol{\beta}\|^2, \\ & \quad \hat{\mathbf{X}} = \mathbf{X} + \mathbf{cd}^\top. \end{aligned} \tag{42}$$

If $\mathbf{e} = \mathbf{e}_i$, in which $\mathbf{e}_i$ is a vector with the ith entry being 1 and all other entries being zero, problem (42) is equivalent to problem (41). If $\mathbf{e} = -\mathbf{e}_i$, problem (42) is equivalent to problem (40). Hence, in the following part, we will focus on solving this unified problem (42).

To solve problem (42), we can first solve the lower level optimization problem in the constraints. It admits a simple solution that $\hat{\boldsymbol{\beta}} = \hat{\mathbf{X}}^\dagger \mathbf{y}$ and $\hat{\mathbf{X}}^\dagger$ is the pseudo-inverse of $\hat{\mathbf{X}}$. This pseudo-inverse can be written as $\hat{\mathbf{X}}^\dagger = \mathbf{X}^\dagger + \mathbf{G}$ [43], where

$$\begin{aligned} \mathbf{G} =& \frac{1}{\gamma}\mathbf{X}^\dagger \mathbf{n} \mathbf{w}^\top - \frac{\gamma}{\|\mathbf{n}\|^2\|\mathbf{w}\|^2 + \gamma^2} \cdot \\ & \left(\frac{\|\mathbf{w}\|^2}{\gamma}\mathbf{X}^\dagger \mathbf{n} + \mathbf{v}\right)\left(\frac{\|\mathbf{n}\|^2}{\gamma}\mathbf{w} + \mathbf{n}\right)^\top, \end{aligned} \tag{43}$$

$\gamma = 1 + \mathbf{d}^\top \mathbf{X}^\dagger \mathbf{c}$, $\mathbf{v} = \mathbf{X}^\dagger \mathbf{c}$, $\mathbf{n} = (\mathbf{X}^\dagger)^\top \mathbf{d}$, and $\mathbf{w} = (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger)\mathbf{c}$.

Since $\hat{\boldsymbol{\beta}} = \hat{\mathbf{X}}^\dagger \mathbf{y} = (\mathbf{X}^\dagger + \mathbf{G})\mathbf{y}$ and $\mathbf{X}^\dagger$ does not depend on $\mathbf{c}$ and $\mathbf{d}$, our problem is equivalent to

$$\begin{aligned} \min_{\mathbf{c},\mathbf{d}} : & \quad \mathbf{e}^\top \mathbf{G} \mathbf{y} \\ \text{s.t.} & \quad \|\mathbf{c} \cdot \mathbf{d}^\top\|_\mathrm{F} \leq \eta. \end{aligned} \tag{44}$$

Suppose $(\mathbf{c}^*, \mathbf{d}^*)$ is the optimal solution of (44), it is easy to see that for nonzero k, $(k\mathbf{c}^*, \mathbf{d}^*/k)$ is also a valid optimal solution. To avoid the ambiguity, it is necessary and possible to further reduce the feasible region. Hence, we put an extra constraint on $\mathbf{c}$, where we restrict the norm of $\mathbf{c}$ to be less than or equal to 1. As the result, our problem can be further written as

$$\begin{aligned} \min_{\mathbf{c},\mathbf{d}} : \quad & \mathbf{e}^\top \mathbf{G} \mathbf{y} \\ \text{s.t.} \quad & \|\mathbf{c}\| \leq 1, \quad \|\mathbf{d}\| \leq \eta, \end{aligned} \tag{45}$$

in which we use the identity $\|\mathbf{c}\mathbf{d}^\top\|_{\mathrm{F}} = \|\mathbf{c}\|\|\mathbf{d}\|$. It is clear that problem (44) and problem (45) have the same optimal objective value.

Since $\mathbf{G}$ is determined by $\mathbf{c}$, $\mathbf{d}$, and $\mathbf{X}$, different values of $\mathbf{c}$ and $\mathbf{d}$ may result in different objective functions. Before further discussion, let us assume the singular value decomposition of the original feature matrix is $\mathbf{X} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^\top$, where $\mathbf{\Sigma} = [\text{diag}(\sigma_1, \sigma_2, \cdots, \sigma_m), \mathbf{0}]^\top$ and $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_m > 0$. With this decomposition, we have $\mathbf{X}^\dagger = \mathbf{V}\mathbf{\Sigma}^\dagger\mathbf{U}^\top$, where $\mathbf{\Sigma}^\dagger = [\text{diag}(\sigma_1^{-1}, \sigma_2^{-1}, \cdots, \sigma_m^{-1}), \mathbf{0}]$. In (43), if $\eta \geq \sigma_m$, by letting $\gamma \to 0$, we have our objective being minus infinity by setting $(\mathbf{c}, \mathbf{d}) = (\mathbf{u}_m, -\sigma_m \mathbf{v}_m)$ or $(\mathbf{c}, \mathbf{d}) = (-\mathbf{u}_m, \sigma_m \mathbf{v}_m)$, where $\mathbf{u}_m$ and $\mathbf{v}_m$ are the mth column of matrices $\mathbf{U}$ and $\mathbf{V}$, respectively. Hence, we conclude that, when $\eta \geq \sigma_m$, the optimal value of problem (45) is unbounded from below. As the result, throughout this section, we assume $\eta < \sigma_m$. Thus, we also have $\gamma = 1 + \mathbf{d}^\top \mathbf{X}^\dagger \mathbf{c} \geq 1 - \|\mathbf{c} \cdot \mathbf{d}^\top\| \|\mathbf{X}^\dagger\| \geq 1 - \frac{\eta}{\sigma_m} > 0$. We note that when η approaches σ_m, it does not mean to kill all of the signals in the feature matrix but only some signals with the energy equal to the smallest singular value of the feature matrix.

Let h denote our objective $h(\mathbf{c}, \mathbf{d}) = \mathbf{e}^\top \mathbf{G} \mathbf{y}$, plug in the expression of $\mathbf{G}$, and we have

$$\begin{aligned} h(\mathbf{c}, \mathbf{d}) = & \frac{1}{\|\mathbf{n}\|^2 \|\mathbf{w}\|^2 + \gamma^2} \big(\gamma \mathbf{e}^\top \mathbf{X}^\dagger \mathbf{n} \mathbf{w}^\top \mathbf{y} - \gamma \mathbf{e}^\top \mathbf{v} \mathbf{n}^\top \mathbf{y} \\ & - \|\mathbf{w}\|^2 \mathbf{e}^\top \mathbf{X}^\dagger \mathbf{n} \mathbf{n}^\top \mathbf{y} - \|\mathbf{n}\|^2 \mathbf{e}^\top \mathbf{v} \mathbf{w}^\top \mathbf{y} \big). \end{aligned} \tag{46}$$

We need to optimize $h(\mathbf{c}, \mathbf{d})$ over $\mathbf{c}$ and $\mathbf{d}$ with the constraint $\|\mathbf{c}\| \leq 1$ and $\|\mathbf{d}\| \leq \eta$. However, $h(\mathbf{c}, \mathbf{d})$ is a ratio of two quartic functions, which is known to be a hard non-convex problem in general. To solve this problem, similar to [34], we can use the projected gradient descent method. However, it is hard to choose a proper step-size and its convergence is not clear when the projected gradient descent is applied to a non-convex problem. In the following, we provide an alternating optimization algorithm with provable convergence.

The enabling observation of our approach is that even though the optimization problem is a complex non-convex problem, for a fixed $\mathbf{c}$, h is a ratio of two quadratic functions with respect to $\mathbf{d}$. Similarly, for a fixed $\mathbf{d}$, h is a ratio of two quadratic functions with respect to $\mathbf{c}$. A ratio of two quadratic functions admits a hidden convex structure [44]. Inspired by this, we decompose our optimization variables into $\mathbf{c}$ and $\mathbf{d}$, and then use alternating optimization algorithm described in Algorithm 3 to sequentially optimize $\mathbf{c}$ and $\mathbf{d}$.

Algorithm 3 Optimal Rank-one Attack Matrix Design via the Alternating Optimization Algorithm

```
Input: data set ${y_i, \mathbf{x}_i}_{i=1}^n$ and energy budget $\eta$.
Initialize: randomly initialize $\mathbf{c}^0$ and $\mathbf{d}^0$, set number of iterations $k = 0$.
compute $\mathbf{G}$ according to (43).
plug in the expression of $\mathbf{G}$ into (45), and obtain our objective, $h(\mathbf{c}, \mathbf{d})$, as in (46).
Do
    update $\mathbf{c}^k$ by solving: $\mathbf{c}^k = \text{argmin}_{\|\mathbf{c}\| \leq 1} : h(\mathbf{c}, \mathbf{d}^{k-1})$,
    update $\mathbf{d}^k$ by solving: $\mathbf{d}^k = \text{argmin}_{\|\mathbf{d}\| \leq \eta} : h(\mathbf{c}^k, \mathbf{d})$,
    set $k = k + 1$,
While convergence conditions are not meet.
compute the modification matrix $\mathbf{\Delta} = \mathbf{c}^k (\mathbf{d}^k)^\top$.
Output: return the modification matrix, $\mathbf{\Delta}$.
```

The core of this algorithm is to solve the following two problems

$$\mathbf{c}^k = \underset{\|\mathbf{c}\| \leq 1}{\text{argmin}} : h(\mathbf{c}, \mathbf{d}^{k-1}), \tag{47}$$

and

$$\mathbf{d}^k = \underset{\|\mathbf{d}\| \leq \eta}{\text{argmin}} : h(\mathbf{c}^k, \mathbf{d}). \tag{48}$$

For a fixed $\mathbf{d}$, the objective of problem (47) becomes $h(\mathbf{c}, \mathbf{d}) = h_1(\mathbf{c})/h_2(\mathbf{c})$, where we omit the superscript of $\mathbf{d}$,

$$\begin{aligned} h_1(\mathbf{c}) = & \, \mathbf{c}^\top \big[\mathbf{e}^\top \mathbf{X}^\dagger \mathbf{n} \mathbf{n} \mathbf{y}^\top (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) - \mathbf{n}^\top \mathbf{y} \mathbf{n} \mathbf{e}^\top \mathbf{X}^\dagger \\ & - \mathbf{e}^\top \mathbf{X}^\dagger \mathbf{n} \mathbf{n}^\top \mathbf{y} (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) - \|\mathbf{n}\|^2 (\mathbf{X}^\dagger)^\top \mathbf{e} \mathbf{y}^\top (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) \big] \mathbf{c} \\ & + \big[\mathbf{e}^\top \mathbf{X}^\dagger \mathbf{n} (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) \mathbf{y} - \mathbf{n}^\top \mathbf{y} (\mathbf{X}^\dagger)^\top \mathbf{e} \big]^\top \mathbf{c}, \end{aligned} \tag{49}$$

and

$$h_2(\mathbf{c}) = \mathbf{c}^\top \big[\|\mathbf{n}\|^2 (\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) + \mathbf{n}\mathbf{n}^\top \big] \mathbf{c} + 2\mathbf{n}^\top \mathbf{c} + 1. \tag{50}$$

Hence, problem (47) can be written as:

$$\min_{\mathbf{c}} : \quad \frac{h_1(\mathbf{c})}{h_2(\mathbf{c})} \tag{51}$$

$$\text{s.t.} \quad \|\mathbf{c}\| \leq 1, \tag{52}$$

where the forms of $h_i(\mathbf{c}) = \mathbf{c}^\top \mathbf{A}_i \mathbf{c} + 2\mathbf{b}_i^\top \mathbf{c} + l_i$, $i = 1, 2$ and $\mathbf{A}_i$, $\mathbf{b}_i$ and l_i can be derived from (49) and (50). The objective of this problem is the ratio of two quadratic functions. Even though it is non-convex, it has certain hidden convex structures. The following theorem characterizes its optimal solution by solving a semidefinite programming [44].

Theorem 1. *([44]) If there exists $\mu > 0$ such that*

$$\begin{bmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{b}_2^\top & l_2 \end{bmatrix} + \mu \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -1 \end{bmatrix} \succ \mathbf{0}, \tag{53}$$

the optimal value of problem (51) *is equivalent to the following optimal value*

$$\max_{\alpha, \, \nu \geq 0} : \quad \alpha \tag{54}$$

$$\textit{s.t.} \quad \begin{bmatrix} \mathbf{A}_1 & \mathbf{b}_1 \\ \mathbf{b}_1^\top & l_1 \end{bmatrix} \succeq \alpha \begin{bmatrix} \mathbf{A}_2 & \mathbf{b}_2 \\ \mathbf{b}_2^\top & l_2 \end{bmatrix} - \nu \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{0} & -1 \end{bmatrix}$$

Proof. Please see [44] for detail. □

We now show that our problem (51) satisfies condition (53). As the result, we can find the solution to problem (51) by solving problem (54).

To prove the left hand side of (53) is positive definite, we can show the following two inequalities are true according to Schur complement condition for positive definite matrix

$$l_2 - \mu > 0, \tag{55}$$

$$\mathbf{A}_2 + \mu\mathbf{I} - \frac{1}{1-\mu}\mathbf{b}_2\mathbf{b}_2^\top \succ \mathbf{0}, \tag{56}$$

where $l_2 = 1$. Plug in the expression of $\mathbf{A}_2$, the left hand of inequality (56) can be written as

$$\begin{aligned}&\mathbf{A}_2 + \mu\mathbf{I} - \frac{1}{1-\mu}\mathbf{b}_2\mathbf{b}_2^\top \\ &= \|\mathbf{n}\|^2(\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger) + \mu\mathbf{I} - \frac{\mu}{1-\mu}\mathbf{n}\mathbf{n}^\top.\end{aligned}$$

Since $\mathbf{I} - \mathbf{X}\mathbf{X}^\dagger$ is a projection matrix, it is positive semi-definite. So, we only need to prove

$$\mu\mathbf{I} - \frac{\mu}{1-\mu}\mathbf{n}\mathbf{n}^\top \succ \mathbf{0}. \tag{57}$$

Since $\mathbf{n}\mathbf{n}^\top$ is rank-one and its non-zero eigenvalue is $\|\mathbf{n}\|^2$, it equals to proving $\|\mathbf{n}\|^2/(1-\mu) < 1$. To guarantee this inequality, we only need to make sure $\mu < 1 - \|\mathbf{n}\|^2$. Since $\|\mathbf{X}^\dagger\| \leq 1/\sigma_m$ and $\|\mathbf{d}\| \leq \eta$, we get $\|\mathbf{n}\|^2 = \|(\mathbf{X}^\dagger)^\top\mathbf{d}\|^2 \leq \|\mathbf{X}^\dagger\|^2\|\mathbf{d}\|^2 \leq \eta^2/\sigma_m^2 < 1$. By choosing $0 < \mu < 1 - \|\mathbf{n}\|^2 < 1$, we can ensure (55) and (56) are both satisfied, and hence inequality (53) is satisfied.

From Theorem 1, we know the optimal value of (51) is equivalent to the optimal value of problem (54). Problem (54) is a semidefinite programming problem, which is convex and can be easily solved by modern tools such as [45] and [46]. We now discuss how to find the optimal $\mathbf{c}$ which achieves this value. Suppose the optimal solution of problem (54) is (α^*, ν^*). Since, $h_2(\mathbf{c}) > 0$, we have $h_1(\mathbf{c}) \geq \alpha^* h_2(\mathbf{c})$ for any feasible $\mathbf{c}$. Hence, we can compute the optimal solution of problem (51) by solving

$$\underset{\mathbf{c}}{\text{argmin}}: \quad h_1(\mathbf{c}) - \alpha^* h_2(\mathbf{c}) \tag{58}$$

$$\text{s.t.} \quad \|\mathbf{c}\|^2 \leq 1 \tag{59}$$

This problem is just a trust region problem. There are several existing methods to solve it efficiently. In this paper, we employ the method described in [47].

Now, we turn to solve problem (48). Since (48) and (47) have similar structure, we can employ the methods described in Theorem 1 and (58) to find its optimal value and optimal solution for problem (48).

Until now, we have fully described how to solve the intermediate problems in the alternating optimization method. The following theorem shows that the proposed alternating optimization algorithm will converge. Suppose the generated sequence of solution is $\{\mathbf{c}^k, \mathbf{d}^k\}$, $k = 0, 1, \cdots$, and we have the following corollary:

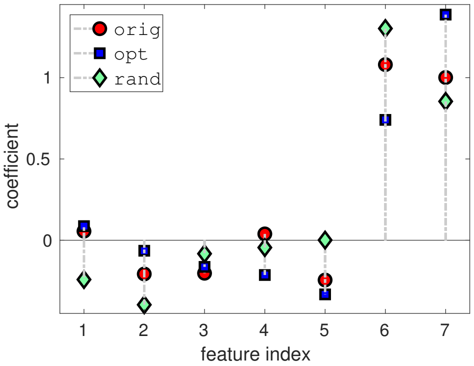


Fig. 1. The regression coefficients before and after attacking the fourth regression coefficient with objective (5).

Corollary 1. *The sequence* $\{\mathbf{c}^k, \mathbf{d}^k\}$ *admits a limit point* $\{\bar{\mathbf{c}}, \bar{\mathbf{d}}\}$ *and we have*

$$\lim_{k\to\infty} h(\mathbf{c}^k, \mathbf{d}^k) = h(\bar{\mathbf{c}}, \bar{\mathbf{d}}). \tag{60}$$

Furthermore, every limit point is a critical point, which means

$$\nabla h(\bar{\mathbf{c}}, \bar{\mathbf{d}})^\top \begin{bmatrix}\mathbf{c} - \bar{\mathbf{c}} \\ \mathbf{d} - \bar{\mathbf{d}}\end{bmatrix} \geq 0, \tag{61}$$

for any $\|\mathbf{c}\| \leq 1$ *and* $\|\mathbf{d}\| \leq \eta$.

Proof. We first give the proof of (60). Since the sequence $\{\mathbf{c}^k, \mathbf{d}^k\}$ lies in the compact set, $\{(\mathbf{c}, \mathbf{d}) \,|\, \|\mathbf{c}\| \leq 1, \|\mathbf{d}\| \leq \eta\}$, and according to the Bolzano-Weierstrass Theorem [48], $\{\mathbf{c}^k, \mathbf{d}^k\}$ must have limit points. Hence, there is a subsequence of $\{h^k\}$ which converges to $h(\bar{\mathbf{c}}, \bar{\mathbf{d}})$. As the objective is a continuous function with respect to $\mathbf{c}$ and $\mathbf{d}$, the compactness of the constraint also implies the sequence of the objective value, $\{h^k\}$, is bounded from below. In addition, $\{h^k\}$ is a non-increasing sequence, which indicates that the sequence of the function value must converge. In summary, the sequence $\{h^k\}$ must converge to $h(\bar{\mathbf{c}}, \bar{\mathbf{d}})$. For the rest of the proof, please refer to Corollary 2 of [49] for more details. □

IV. Numerical Examples

In this section, we test our adversarial attack strategies on practical regression problems. In the first regression task, we use seven international indexes to predict the returns of the Istanbul Stock Exchange [50]. The data set contains 536 data samples, which are the records of the returns of Istanbul Stock Exchange with seven other international indexes starting from Jun. 5, 2009 to Feb. 22, 2011. Also, we demonstrate how our attack impacts the quality of a regression task using the wine dataset [51].

A. Attacking one specific regression coefficient

In this experiment, we attack the fourth regression coefficient of the Istanbul Stock Exchange dataset and try to make its magnitude large by solving problem (4). We use two strategies to attack this coefficient with a fixed energy budget $\eta = 0.2$. The first strategy is the one proposed in this paper. As a comparison, we also use a random strategy to approximate the exhaustive search algorithm. In the random

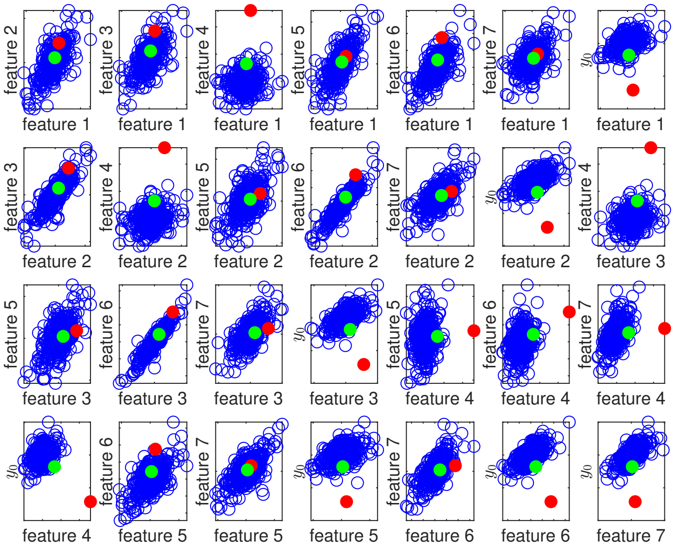


Fig. 2. The scatter plot of the original data, the designed poisoning data, and the poisoning data after the repeating strategy. The x-axis and y-axis are two features that are specified by their corresponding axes labels (including the response value). The blue circle represents the original data, the solid red dot denotes the data point designed by our proposed method in Algorithm 1, and the solid green circle indicates our proposed poisoning data after 16 times of repeating.

strategy, we randomly generate the adversarial data point with each entry being i.i.d. generated from a standard normal distribution. Then, we normalize its energy to be η. We repeat this random attack 10000 times and select the one with the smallest objective value. Hence, the random strategy is an approximation of the exhaustive search algorithm.

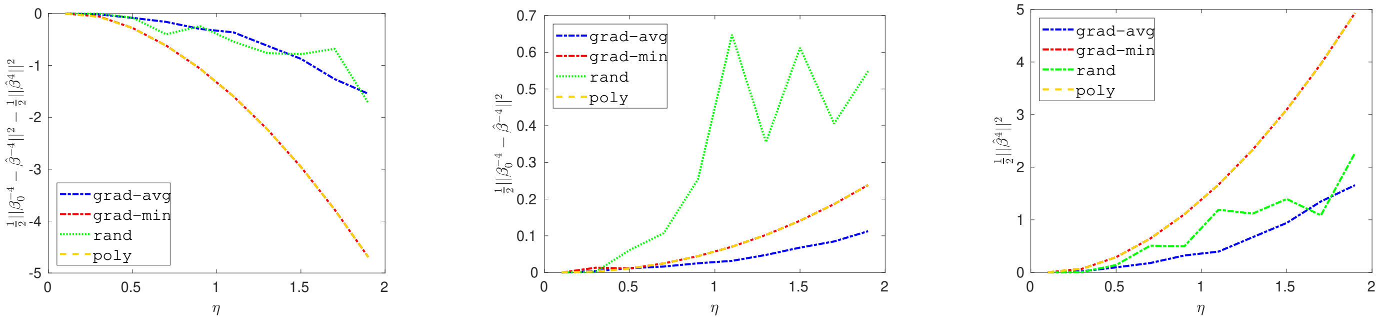
Fig. 1 shows the regression coefficients before and after our attack. The x-axis denotes the index of the regression coefficients and the y-axis indicates the value of the regression coefficients. In this figure, the ‘orig’ denotes the original regression coefficient, ‘opt’ represents the regression coefficient after attacking by our proposed optimal attack strategy, and ‘rand’ indicates the regression coefficient after attacking by the random attack strategy. From the figure we can see that our proposed adversarial attack strategy is much more efficient than the random attack strategy. One can also observe that by only adding one adversarial example, designed by the approach characterized in this paper, one can dramatically change the value of a regression coefficient and hence change the importance of that explanatory variable.

Fig. 2 shows the original data points (in blue), the optimal adversarial data point (in red), and the adversarial data points after the 16 times repeating strategy (in green) in this experiment. The figure demonstrates that the proposed adversarial data point may behave as an outlier. However, after our simple repeating strategy, the adversarial data points act just like normal data points. Hence, our repeating strategy can mitigate the adversarial data point being detected by the regression system.

B. Attacking without changing untargeted regression coefficients too much

From the numerical examples in the previous subsection, we can see the untargeted regression coefficients may change greatly while attacking one specific regression coefficient with an adversarial data point. For example, as demonstrated in Fig. 1, the sixth and seventh regression coefficients change significantly when we attack the fourth regression coefficient. To mitigate the undesirable changes of untargeted regression coefficients, we need more sophisticated attacking strategies. In this subsection, we will test different strategies with a more general objective function as demonstrated in Section II-C. We also use the same data set as described in the previous subsection. We first try to attack the fourth regression coefficient to increase its importance while making only small changes to the rest of the regression coefficients. To accomplish this task, we aim to solve problem (30) with $\lambda = -1$. Given the energy budget, firstly, we use our semidefinite relaxation based algorithm to solve problem (37), and then follow Algorithm 2 to find the adversarial data point. For comparison, we also carry out the random attack strategy, in which we randomly generate the data point with each entry being i.i.d. according to the standard normal distribution. Then, we normalize its energy being η and added it to the original data points. We repeat these random attacks 10000 times and select the one with the smallest objective value. The third strategy is the projected gradient descent based strategy, where we use the projected gradient descent algorithm to solve (37) and follow similar steps of Algorithm 2 to find the adversarial data point. Projected gradient descent works much like the gradient descent except with an additional operation that projects the result of each step onto the feasible set after moving in the direction of negative gradient [52]. In our experiment, we use diminishing step-size, $1/(t+1)$. Since the projected gradient descent algorithm depends on the initial points heavily, given the energy budget, we repeat it 100 times with different random initial points and treat the average of its objective values as the objective value of this algorithm. Also, among the 100 times attacks, we record the one with the smallest objective value.

Fig. 3 shows the objective values under different energy budgets with different attacking strategies and Fig. 4 demonstrates the regression coefficients after one of the attacks of different strategies with $\eta = 1$. In these figures, ‘orig’ is the original regression coefficient, ‘rand’ means the random strategy, ‘poly’ indicates our semidefinite relaxation strategy, ‘grad-avg’ is the average objective value of the 100 times attacks based on the projected gradient descent algorithm, and ‘grad-min’ is the one with the smallest objective value among the 100 times attacks based on the projected gradient descent algorithm. From these two figures, we can see that our semidefinite relaxation based strategy performs much better than the other two strategies. Among the 100 times attacks based on the projected gradient descent, the minimal one can achieve similar objective values as our proposed attacks based on the semidefinite relaxation. In addition, in our experiment, our semidefinite relaxation method with relaxation order 2 or



(a) (b) (c)

Fig. 3. Attack the fourth regression coefficient with objective (30) and $\lambda = -1$ under different energy budgets.

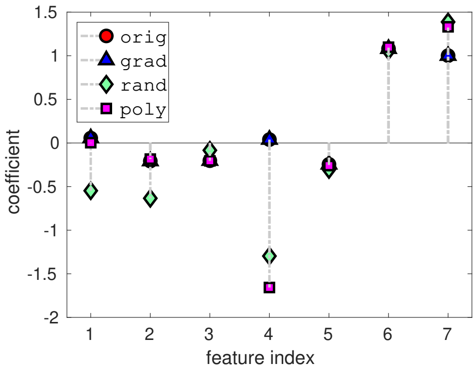


Fig. 4. The regression coefficients before and after different kinds of strategies that attack the fourth regression coefficient with energy budget $\eta = 1$.

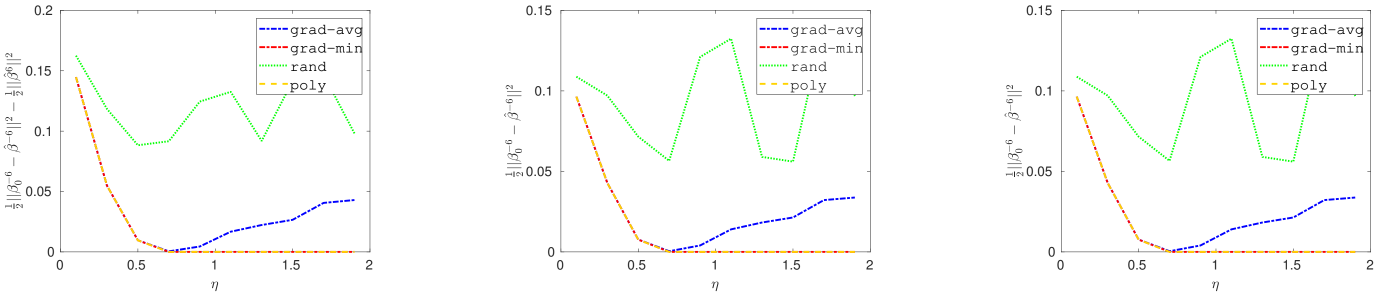
3 can always lead to globally optimal solutions. Hence, the computational complexity of this method is still low. Fig. 4 also shows that our relaxation based method leads to the largest magnitude of the fourth regression coefficient while keeping other regression coefficients almost unchanged.

In the second experiment, we attack the sixth regression coefficient and attempt to make its magnitude small while keeping the change of the rest of the coefficients to be small. So, we set $\lambda = 1$ in problem (30) to achieve this goal. The settings of each strategy are similar to the ones in the first experiment. Fig. 5 shows the objective values with different strategies under different energy budgets and Fig. 6 demonstrates the regression coefficients after one of the attacks of those strategies respectively with energy budget $\eta = 1$. From Fig. 5 we know the projected gradient descent based strategy and the semidefinite relaxation based strategy achieve much lower objective values compared to the random attack strategy. Specifically, when the energy budget is smaller than 0.7, both strategies behave similarly. However, when the energy budget is larger than 0.7, the projected gradient descent based strategy leads to larger objective values as the energy budget grows. This is because the projected gradient descent algorithm tends to find solutions at the boundary of the feasible set. Only some attacks with good initialization can lead to the global minimum. By contrast, our semidefinite relaxation based strategy can find the globally optimal solutions with relaxation order 2 or 3. Thus, it gives the best performance among the three strategies. Fig. 6 also demonstrates our relaxation based method achieves the global optimum when $\eta = 1$ as it leads the sixth regression coefficient to zero and other regression coefficients to be unchanged.

C. Rank-one attack

In this subsection, we carry out different rank-one attack strategies. Our goal is to minimize the magnitude of the fourth regression coefficient with objective (41). We compare two strategies: the projected gradient descent based strategy discussed in Section IV-B and our proposed alternating optimization based strategy. For the projected gradient descent based strategies, we use different step sizes: $1/(1+t)$, $10/(1+t)$, and $100/(1+t)$. As our analysis shows, when the energy budget is larger than the smallest singular value, our objective can be minus infinity. Hence, in our experiment, we vary the energy budget from 0 to the smallest singular value, which is 0.053. Given a specific energy budget, we set all the algorithms with the same randomly initialized point and run them until they stop with the same convergence condition: two consecutive function values change too small, or the algorithm reaches the maximal allowable iterations. We repeat this process 100 times and record their average objective values.

Fig. 7 (a) shows the averaged run times and Fig. 7 (b) illustrates objective values of the four algorithms, where ‘GD-1’, ‘GD-10’ and ‘GD-100’ stand for the projected gradient descent with stepsizes $1/(1+t)$, $10/(1+t)$, and $100/(1+t)$, respectively, and ‘AO’ denotes the proposed alternating optimization method. We carry out this experiment on a PC with four Intel E3 CPUs. All the four algorithms have the same convergence condition: the absolute value of the difference of two consecutive objective values is less than 10^{-5}. Fig. 7 (a) shows that, as the energy budget increases, the run times of the alternating optimization, GD-1, and GD-10 increase. However, as the energy budget increases, the run times of GD-100 first decrease and then increase. This is due to the fact that a larger stepsize will result in a faster convergence rate while it may cause oscillation. Fig. 7 (b) shows that when the energy budget increases, the objectives decrease for both of these algorithms. Furthermore, the proposed alternating optimization based algorithm provides much smaller objective values, especially when the energy budget approaches the smallest singular value. When the energy budget approaches the smallest singular value, the gradient descent based algorithm becomes very



(a) (b) (c)

Fig. 5. Attack the sixth regression coefficient with objective (30) and $\lambda = 1$ under different energy budgets.

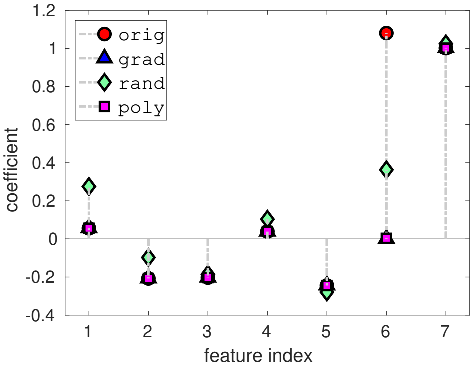
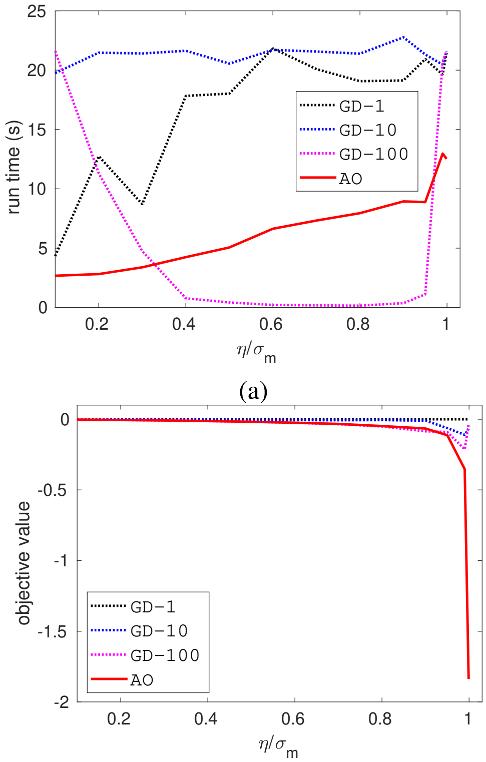


Fig. 6. The regression coefficients after different kinds of strategies that attack the sixth regression coefficient with energy budget $\eta = 1$.



(a)

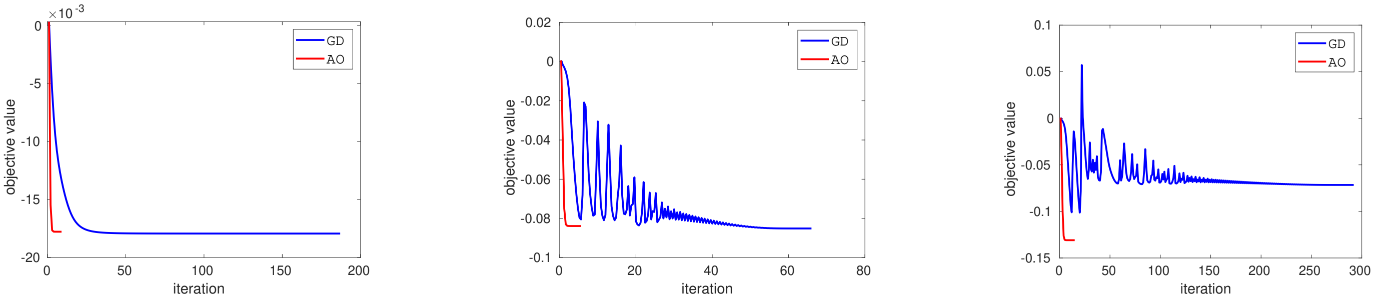
(b)

Fig. 7. The averaged run times (Subfigure (a)) and the objective values (Subfigure (b)) of the projected gradient descent and the proposed alternating optimization method with different stepsizes.

unstable. This is because when the energy budget is large, the objective is very sensitive to the energy budget. So, a small stepsize may result in significant objective value change. This phenomena can be observed in Fig. 8, where it depicts the evolution of the objective values of ‘AO’ and ‘GD-100’ with the energy budget being $\eta/\sigma_m = 0.5$, $\eta/\sigma_m = 0.9$ and $\eta/\sigma_m = 0.95$, respectively, and σ_m is the smallest singular value of the original feature matrix. From this figure we can see the alternating optimization based algorithm converges very fast while the projected gradient descent based algorithm becomes unstable when the energy budget is large. This is due to the fact that the objective of our alternating optimization based algorithm is guaranteed to be monotonically decreasing.

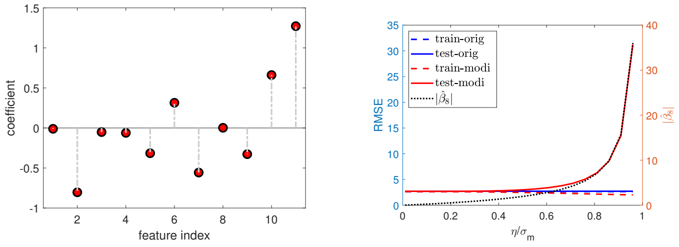
In the second experiment, we test our rank-one attack strategy on the wine dataset [51], which includes 11 chemical analysis of the red wine and its corresponding quality (ranging from 3 to 8). In this dataset, we have 1599 data samples, and we randomly choose 80 percent of the data as the training set and the rest as the test data. We use linear regression to learn the regression coefficients on the training data and then use these regression coefficients on the test data to predict the quality of the test data. We use the root mean square error (RMSE) to measure the goodness of predicting on the training and test data. We use the rank-one attack strategy proposed in this paper on the training data with the target of maximizing the eighth regression coefficient (corresponding to the density feature). We carry out the attack with different energy budgets ranging from 0 to the smallest singular value of the feature matrix of training data.

Fig. 9 (a) illustrates the original regression coefficients without attack. The magnitude of the eighth regression coefficient is very small. It reveals that the eighth feature is not important compared to other features. Fig. 9 (b) shows the RMSE on the training data and test data and the magnitude of the eighth modified regression coefficient under different energy budgets. ‘train-orig’ and ‘test-orig’ represent the RMSE on the training and test data without attacking the training data. ‘train-modi’ and ‘test-modi’ denote the RMSE on the training and test data when we conduct our rank-one attack on the training dataset. This figure demonstrates that, even though the RMSE on the attacked training data is low, the model based on the attacked features performs extremely badly on the test data. It illustrates



(a) (b) (c)

Fig. 8. These figures show the evolution of function values as the iteration increases with one typical run of projected gradient descent and alternating optimization algorithm, where (a) is with $\eta/\sigma_m = 0.5$, (b) is with $\eta/\sigma_m = 0.9$, and (c) is with $\eta/\sigma_m = 0.95$ and σ_m is the smallest singular value of the original feature matrix.



(a) (b)

Fig. 9. Figure (a) shows the regression coefficient on the original data set. Figure (b) illustrates the RMSE on the training and test data set and the magnitude of the eighth modified regression coefficient with different attack energy budgets.

that attacking the regression coefficient not only misleads the interpretation of the model but also has a significant impact on the performance of the model. We note that a large value of RMSE shown in Fig. 9 (b) does not necessarily imply that the user can detect the attack easily. In particular, in this experiment, in order to calculate RMSE, we assume that we know the ground-truth response values in the test data. However, in practice, when the user utilizes the trained model to perform testing, the testing data usually does not contain response values. As a result, the user will not know the RMSE and hence will not be able to use RMSE to detect whether there is an attack or not without additional information.

V. Conclusion

In this paper, we have investigated the adversarial robustness of linear regression problems. Particularly, we have given the closed-form solution when we attack one specific regression coefficient with a limited energy budget. Furthermore, we have considered a more complex objective where we attack one of the regression coefficients while trying to keep the rest of the regression coefficients to be unchanged. We have formulated this problem as a multivariate polynomial optimization problem and introduced the semidefinite relaxation method to solve it. Finally, we have studied a more powerful adversary who can make a rank-one modification on the feature matrix. To take advantage of the rank-one structure, we have proposed an alternating optimization algorithm to solve this problem. The numerical examples demonstrated that our proposed closed-form solution and the semidefinite relaxation based strategies could find the globally optimal solutions, and the alternating optimization based strategy provides better solutions, faster convergence, and more stable behavior compared to the projected gradient descent based strategy. We should also note that the solutions are "optimal" under the specific objectives mentioned in the paper. Clearly, if the goal of the attacker is changed, then the optimal attack strategy will be different.

In terms of future work, it is of interest to study how to design multiple adversarial data points and efficiently design the modification matrix without the rank-one constraint. Another interesting future research direction is to study the defense strategies to mitigate this kind of attack. If we consider the defense strategy, one possible problem formulation is

$$\hat{\boldsymbol{\beta}} = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \quad \ell_{def}(\hat{\mathbf{X}}, \hat{\mathbf{y}}, \boldsymbol{\beta})$$
$$\text{s.t.} \quad \hat{\mathbf{X}}, \hat{\mathbf{y}} = \underset{\mathbf{X}\in\mathcal{C}_x,\, \mathbf{y}\in\mathcal{C}_y}{\operatorname{argmin}} \quad \ell_{adv}(\mathbf{X}, \mathbf{y}),$$

where $\ell_{def}(\cdot)$ is the objective of the defender , $\ell_{adv}(\cdot)$ is the objective of the adversary, and $\mathcal{C}_x$ and $\mathcal{C}_y$ are the modification constraints of the feature matrix and response values, respectively. We should also note that $\ell_{adv}(\cdot)$ may also depend on the defense strategy, which will then render the problem as a competing game between the defender and attacker. With an appropriately designed loss function of the defender, solving this optimization problem leads to the best defense strategy under the optimal attack strategy. The complexity of this problem depends on the forms of $\ell_{def}(\cdot)$, $\ell_{adv}(\cdot)$ and their relationship. In some special cases, we can analyze this problem. For example, our paper solved this problem when $\ell_{def}(\cdot)$ is the MSE loss function and $\ell_{adv}(\cdot)$ is the objective of manipulating one of the regression coefficients. When $\ell_{def}(\cdot) = -\ell_{adv}(\cdot)$, it is a minmax problem and Jagielski et al. studied this problem when $\ell_{def}(\cdot) = -\ell_{adv}(\cdot)$ and $\ell_{def}(\cdot)$ equals to the MSE loss function [26]. Generally, this problem is very complicated as the upper-level and lower-level optimization problems are interconnected. We will investigate this important problem with more sophisticated objectives of the defender in our future work.

References

[1] F. Li, L. Lai, and S. Cui, "On the adversarial robustness of linear regression," in *Proc. IEEE International Workshop on Machine Learning*

for Signal Processing (MLSP), Espoo, Finland, Sep. 2020, pp. 1–6.

[2] X. Yan and X. Su, *Linear regression analysis: theory and computing*. World Scientific, 2009.

[3] G. Papageorgiou, P. Bouboulis, and S. Theodoridis, "Robust linear regression analysis— a greedy approach," *IEEE Transactions on Signal Processing*, vol. 63, no. 15, pp. 3872–3887, Aug. 2015.

[4] X. Jiang, W. Zeng, H. C. So, A. M. Zoubir, and T. Kirubarajan, "Beamforming via nonconvex linear regression," *IEEE Transactions on Signal Processing*, vol. 64, no. 7, pp. 1714–1728, Apr. 2016.

[5] J. Chien and J. Chen, "Recursive Bayesian linear regression for adaptive classification," *IEEE Transactions on Signal Processing*, vol. 57, no. 2, pp. 565–575, Feb. 2009.

[6] T. Gustafsson and B. D. Rao, "Statistical analysis of subspace-based estimation of reduced-rank linear regressions," *IEEE Transactions on Signal Processing*, vol. 50, no. 1, pp. 151–159, Jan. 2002.

[7] R. Tibshirani, "Regression shrinkage and selection via the lasso," *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[8] J. H. McDonald, *Handbook of biological statistics*. Sparky House Publishing, 2009.

[9] O. E. Barndorff-Nielsen and N. Shephard, "Econometric analysis of realized covariation: High frequency based covariance, regression, and correlation in financial economics," *Econometrica*, vol. 72, no. 3, pp. 885–925, May 2004.

[10] C. J. ter Braak and S. Juggins, "Weighted averaging partial least squares regression (WA-PLS): an improved method for reconstructing environmental variables from species assemblages," in *Proc. International Diatom Symposium*, Renesse, The Netherlands, Aug. 1993, pp. 485–502.

[11] S. G. Finlayson, J. D. Bowers, J. Ito, J. L. Zittrain, A. L. Beam, and I. S. Kohane, "Adversarial attacks on medical machine learning," *Science*, vol. 363, no. 6433, pp. 1287–1289, Mar. 2019.

[12] A. E. Sallab, M. Abdou, E. Perot, and S. Yogamani, "Deep reinforcement learning framework for autonomous driving," *Electronic Imaging*, vol. 2017, no. 19, pp. 70–76, Jan. 2017.

[13] I. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv:1412.6572*, Dec. 2014.

[14] A. Kurakin, I. J. Goodfellow, and S. Bengio, "Adversarial machine learning at scale," in *Proc. International Conference on Learning Representations*, Toulon, France, Apr. 2017.

[15] I. Goodfellow, P. McDaniel, and N. Papernot, "Making machine learning robust against adversarial inputs," *Communications of the ACM*, vol. 61, no. 7, pp. 56–66, Jun. 2018.

[16] L. G. Hafemann, R. Sabourin, and L. S. Oliveira, "Characterizing and evaluating adversarial examples for offline handwritten signature verification," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2153–2166, Aug. 2019.

[17] W. Tang, B. Li, S. Tan, M. Barni, and J. Huang, "CNN-based adversarial embedding for image steganography," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 8, pp. 2074–2087, Aug. 2019.

[18] D. Li and Q. Li, "Adversarial deep ensemble: Evasion attacks and defenses for malware detection," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 3886–3900, Jun. 2020.

[19] X. Chen, C. Liu, B. Li, K. Lu, and D. Song, "Targeted backdoor attacks on deep learning systems using data poisoning," *arXiv:1712.05526*, Dec. 2017.

[20] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami, "Practical black-box attacks against machine learning," in *Proc. ACM on Asia Conference on Computer and Communications Security*, Abu Dhabi, United Arab Emirates, Apr. 2017, pp. 506–519.

[21] D. Meng and H. Chen, "Magnet: a two-pronged defense against adversarial examples," in *Proc. ACM SIGSAC Conference on Computer and Communications Security*, Dallas, TX, Oct. 2017, pp. 135–147.

[22] S. Boyd and L. Vandenberghe, *Convex optimization*. Cambridge University Press, 2004.

[23] F. Li, L. Lai, and S. Cui, "On the adversarial robustness of subspace learning," *IEEE Transactions on Signal Processing*, vol. 68, pp. 1470–1483, Mar. 2020.

[24] D. L. Pimentel-Alarcón, A. Biswas, and C. R. Solís-Lemus, "Adversarial principal component analysis," in *Proc. IEEE International Symposium on Information Theory*, Aachen, Germany, Jun. 2017, pp. 2363–2367.

[25] S. Alfeld, X. Zhu, and P. Barford, "Data poisoning attacks against autoregressive models," in *Proc. AAAI Conference on Artificial Intelligence*, Phoenix, Arizona, Feb. 2016, pp. 1452–1458.

[26] M. Jagielski, A. Oprea, B. Biggio, C. Liu, C. Nita-Rotaru, and B. Li, "Manipulating machine learning: Poisoning attacks and countermeasures for regression learning," in *Proc. IEEE Symposium on Security and Privacy*, San Francisco, CA, May 2018, pp. 19–35.

[27] B. Biggio, B. Nelson, and P. Laskov, "Poisoning attacks against support vector machines," in *Proc. International Conference on Machine Learning*, Edinburgh, Scotland, Jun. 2012, pp. 1807–1814.

[28] ——, "Support vector machines under adversarial label noise," in *Proc. Asian Conference on Machine Learning*, Taoyuan, Taiwan, Nov. 2011, pp. 97–112.

[29] S.-M. Moosavi-Dezfooli, A. Fawzi, and P. Frossard, "Deepfool: A simple and accurate method to fool deep neural networks," in *Proc. Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, Jun. 2016, pp. 2574–2582.

[30] B. Biggio, A. Demontis, A. Paudice, V. Wongrassamee, E. C. Lupu, and F. Roli, "Towards poisoning of deep learning algorithms with back-gradient optimization," in *Proc. ACM Workshop on Artificial Intelligence and Security*, Dallas, TX, Oct. 2017, pp. 27–38.

[31] A. Shafahi, W. R. Huang, M. Najibi, O. Suciu, C. Studer, T. Dumitras, and T. Goldstein, "Poison frogs! targeted clean-label poisoning attacks on neural networks," in *Proc. Conference on Neural Information Processing Systems*, Montréal, Canada, Dec. 2018, pp. 6106–6116.

[32] H. Kwon, Y. Kim, H. Yoon, and D. Choi, "Selective audio adversarial example in evasion attack on speech recognition system," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 526–538, Jun. 2019.

[33] B. Flowers, R. M. Buehrer, and W. C. Headley, "Evaluating adversarial evasion attacks in the context of wireless communications," *IEEE Transactions on Information Forensics and Security*, vol. 15, pp. 1102–1113, Aug. 2019.

[34] S. Mei and X. Zhu, "Using machine teaching to identify optimal training-set attacks on machine learners," in *Proc. AAAI Conference on Artificial Intelligence*, Austin, Texas, Jan. 2015, pp. 2871–2877.

[35] R. A. Horn and C. R. Johnson, *Matrix analysis*. Cambridge University Press, 2012.

[36] A. Beck and M. Teboulle, "On minimizing quadratically constrained ratio of two quadratic functions," *Journal of Convex Analysis*, vol. 17, no. 3, pp. 789–804, 2010.

[37] A. Konar and N. D. Sidiropoulos, "Fast approximation algorithms for a class of non-convex QCQP problems using first-order methods," *IEEE Transactions on Signal Processing*, vol. 65, no. 13, pp. 3494–3509, Apr. 2017.

[38] J. B. Lasserre, "Global optimization with polynomials and the problem of moments," *SIAM Journal on Optimization*, vol. 11, no. 3, pp. 796–817, 2001.

[39] M. Laurent, "Sums of squares, moment matrices and optimization over polynomials," in *Emerging Applications of Algebraic Geometry*. Springer, 2009, pp. 157–270.

[40] T. Weisser, J. B. Lasserre, and K.-C. Toh, "Sparse-BSOS: a bounded degree SOS hierarchy for large scale polynomial optimization with sparsity," *Mathematical Programming Computation*, vol. 10, no. 1, pp. 1–32, 2018.

[41] M. J. Wainwright and M. I. Jordan, "Log-determinant relaxation for approximate inference in discrete markov random fields," *IEEE transactions on signal processing*, vol. 54, no. 6, pp. 2099–2109, Jun. 2006.

[42] L. Porkolab and L. Khachiyan, "On the complexity of semidefinite programs," *Journal of Global Optimization*, vol. 10, no. 4, pp. 351–365, 1997.

[43] K. B. Petersen and M. S. Pedersen, "The matrix cookbook," *Technical University of Denmark*, 2008.

[44] A. Beck and M. Teboulle, "A convex optimization approach for minimizing the ratio of indefinite quadratic functions over an ellipsoid," *Mathematical Programming*, vol. 118, no. 1, pp. 13–35, Apr. 2009.

[45] F. Rendl, "A matlab toolbox for semidefinite programming," *The program can be found at ftp://orion. uwaterloo. ca/pub/henry/teaching/co769g*, 1994.

[46] J. Lofberg, "Yalmip: A toolbox for modeling and optimization in matlab," in *Proc. International Conference on Robotics and Automation*, New Orleans, LA, Apr. 2004, pp. 284–289.

[47] A. Beck, A. Ben-Tal, and M. Teboulle, "Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares," *SIAM Journal on Matrix Analysis and Applications*, vol. 28, no. 2, pp. 425–445, 2006.

[48] R. G. Bartle and D. R. Sherbert, *Introduction to real analysis*. Wiley New York, 2000.

[49] L. Grippo and M. Sciandrone, "On the convergence of the block nonlinear Gauss–Seidel method under convex constraints," *Operations Research Letters*, vol. 26, no. 3, pp. 127–136, Apr. 2000.

[50] O. Akbilgic, H. Bozdogan, and M. E. Balaban, "A novel hybrid RBF neural networks model as a forecaster," *Statistics and Computing*, vol. 24, no. 3, pp. 365–375, May 2014.

[51] D. Dua and C. Graff, “UCI machine learning repository,” 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[52] N. Parikh and S. Boyd, “Proximal algorithms,” *Foundations and Trends® in Optimization*, vol. 1, no. 3, pp. 127–239, Jan. 2014.