Countering adversarial evasion
in regression analysis

David Benfield, Phan Tu Vuong, and Alain Zemkoho

School of Mathematical Sciences
University of Southampton
SO17 1BJ Southampton, United Kingdom
db3g17@soton.ac.uk, t.v.phan@soton.ac.uk, a.b.zemkoho@soton.ac.uk

Abstract

Adversarial machine learning challenges the assumption that the underlying distribution remains consistent throughout the training and implementation of a prediction model. In particular, adversarial evasion considers scenarios where adversaries adapt their data to influence particular outcomes from established prediction models; such scenarios arise in applications such as spam email filtering, malware detection and fake-image generation, where security methods must be actively updated to keep up with the ever-improving generation of malicious data. Game theoretic models have been shown to be effective at modelling these scenarios and hence training resilient predictors against such adversaries. In this paper, we introduce a pessimistic bilevel optimsiation model, based on Stackelberg leader-follower games, to counter adversarial evasion in regression analysis. Unlike in some existing literature, we do not make assumptions (such as lower-level convexity or uniqueness of optimal solutions) on the adversary's optimal strategy to avoid hindering their capacity and hence capture their antagonistic nature, leading to more resilient prediction models. Further to this, through the introduction of lower-level constraints, we measure and restrict the adversary's movement, which, unlike its unrestricted counterpart in the current literature, prevents drastic transformations that do not accurately represent reality.

Keywords: Adversarial learning, game theory, bilevel optimisation, regression

1 Introduction

Adversarial machine learning considers the exploitable vulnerabilities of machine learning models and the strategies needed to counter or mitigate such threats [24]. By considering these vulnerabilities during the development stage of our machine learning models, we can work to build resilient methods [6, 8] such as protection from credit card fraud [26] or finding the optimal placement of air defence systems [15]. In particular, we consider the model's sensitivity to changes in the distribution of the data. The way the adversary influences the distribution can fall under numerous categories, see [16] for a helpful taxonomy that categorises these attacks. We focus on the specific case of evasion attacks, which consider the scenarios where adversaries attempt to modify their data to influence particular outcomes from prediction models. Such attacks might occur in security scenarios such as malware detection [2] and network intrusion traffic [23]. In a similar vein, and more recently, vulnerabilities in deep neural networks (DNN) are being discovered, particularly in the field of computer vision and image classification; small perturbations in the data can lead to incorrect classifications by the DNN [25, 14]. These vulnerabilities raise concerns about the robustness of the machine learning technology that is being adopted and, in some cases, in how safe relying on their predictions could be in high-risk scenarios such as autonomous driving [11] and medical diagnosis [12]. By modelling the adversary's behaviour and anticipating these attacks, we can train classifiers that are resilient to such changes in the distribution before they occur.

Game theory provides an effective and popular technique to model adversarial evasion scenarios. These games see one player, the learner, attempt to train a prediction model while another player, the adversary,

modifies their data in an attempt to influence particular outcomes from the learner's predictor. See, for example [7] for early work in modelling adversarial machine learning. The precise structure of such a model then depends on a number of factors, such as whether the attack occurs either at training time [21], or implementation time [4]. Further to this, while some games assume the players act simultaneously [3, 22], others allow for sequential play where either the adversary acts first [19, 20, 5, 18] or the learner acts first [16, 4]. We focus our attention on attacks made at implementation time, where the adversary seeks to influence an already established prediction model. Sequential evasion games naturally take on the form of a bilevel optimisation program. In this formulation, the learner, taking the role of the leader in the upper-level, train a prediction function while anticipating how the adversary, taking the role of the follower in the lower-level, will react. The solution to this program then results in a resilient prediction function. Pessimistic bilevel optimisation in particular has proved particularly effective. When multiple optimal solutions are available to the adversary, the pessimistic model, unlike its optimistic variant, assumes no cooperation between the adversary and the learner. This formulation better captures the antagonistic nature of adversarial evasion scenarios [1, 27, 4].

A pessimistic bilevel formulation for evasion attacks was initially proposed in [4] to suit classification tasks, which was later extended in [1, 27] by relaxing assumptions on convexity and uniqueness of solutions, again for classification problems. In this article, we present a novel pessimistic bilevel program to train resilient predictors against evasion attacks in regression problems. We consider scenarios where we expect adversaries to modify their data in an attempt to influence a particular label from a prediction model. For example, consider a real estate surveyor who uses a prediction model to evaluate the selling price of houses. A homeowner might provide falsified information about their house, such as it's size or distance from public services, in order to influence a higher valuation from the surveyor. Consider a simple example of two houses, H_1 and H_2. These houses are identical in every aspect except for their location. While H_1 is located just 1km from the nearest train station, H_2 is located further away at 4km. Consequently, while H_1 is evaluated at £500k, H_2 would receive a considerably lower evaluation due to its inaccessibility to public transport. To maximise the selling price, the owner of H_2 might lie about the distance of their house as closer to the train station than it actually is. Decisions like this are simulated by the adversary in the lower-level of our bilevel program. Then, the upper-level trains the surveyor's prediction model while considering these adversarial influences and hence trains a more resilient prediction model. Within our model, we make no assumptions about the convexity of the lower-level problem or the uniqueness of its solution, retaining the benefit provided by the pessimistic formulation.

Further to this, we introduce constraints on the lower-level optimisation problem which restrict the extent to which the adversary can modify their data. Consider again our simple housing market of houses H_1 and H_2. We established that the owner of H_2 can lie about the distance of their house from the nearest train station in order to gain a higher evaluation form the surveyor. Without restrictions on their movement, the owner of H_2 would simply state that their house is next to the train station, since a smaller distance leads to a higher evaluation. However, it is incredibility easy to spot this lie due to the large discrepancy between the true distance and the distance stated by the owner. A smaller change in distance, on the other hand, could enable the owner to receive a higher evaluation price while remaining plausibly realistic enough that the surveyor does not notice. Clearly, an adversary which takes smaller movements better reflects the strategies played by adversaries in the real world. However, without restrictions on their movement, the adversary will construct data which, while optimising their objective function, is not plausible or possibly nonsensical. The constraints we introduce here ensure that the similarity between the modified data and its original value is above some pre-defined threshold. To the best of our knowledge, existing game-theoretic approaches to adversarial regression analysis, see, for example, [28, 29, 30], do not propose a pessimistic bilevel program with lower-level constraints on the adversary's movements that makes no assumptions on convexity or uniqueness of solutions.

Moreover, where existing pessimistic approaches rely on feature maps, such as principal component analysis (PCA), or contextual embeddings to represent the adversary's data, the adversary is allowed to manipulate their data in its original feature space. Consequently, we can view the explicit values of the adversary's data at their solution. These values can give us insight into the strategies played by the adversary and allows us to identify the extent to which each feature needs to be modified in order to adequately influence the prediction model. From this, we can identify which features are most vulnerable. For example, if a certain feature required only a minor perturbation, then we might consider it to be particularly vulnerable to adversarial influence. On the other hand, if a feature required large transformations to trick the learner, then we might consider it to be considerably more resilient to attacks.

The contributions of this article can be summarised as follows. We present a novel pessimistic bilevel

optimisation program to train regression prediction functions that are resilient to strategic evasion attempts. This bilevel program, unlike existing approaches to adversarial regression, restricts the adversary's movement through lower-level constraints, while also making no assumptions on the convexity of the lower-level problem, or the uniqueness of its solution. We design experiments to assess the performance of our model and showcase its ability to outperform existing methods. Finally, we demonstrate a key benefit of our model over existing pessimistic approaches. Where existing approaches transform the adversary's data before analysis, removing the ability to analyse the adversary's data at their solution, we keep it in its original feature space. This grants us the ability to infer which features require the greatest transformation in order to effectively influence the learner's prediction function.

The remainder of this article is outlined as follows; we begin in Section 2 by presenting the pessimistic bilevel model with lower-level constraints to train resilient prediction functions in regression analysis before outlining the solution method. Following this, we assess the performance of the bilevel model on two datasets in Section 3 before concluding our findings in Section 4.

2 Training resilient prediction functions

In this section we present the pessimistic bilevel program with lower-level constraints to model adversarial regression. In particular, we demonstrate how the process of adversarial training can be used to train resilient predictors under the scenario of adversaries attempting to strategically modify their data in an attempt to be assigned incorrect labels from established prediction models. Such scenarios might arise, for example, in property valuation, as outlined in the introduction. Through suitable choices of objective functions, we construct a constrained pessimistic bilevel program that sees a learner, in the upper-level, train a prediction function while considering how an adversary, in the lower-level, might alter their data towards a target outcome from the prediction model. The adversary is assigned some existing data as a starting point and the distance that the adversary diverges from this point is measured and restricted by lower-level constraints. We demonstrate, through a motivating example, how these constraints ensure that the adversary's data remain plausibly realistic and prevent instances of data losing their intended meaning. In this way, the bilevel model more realistically simulates adversarial transformations, leading to improved performance. To the best of our knowledge, a pessimistic bilevel program with constrained adversarial movement and with no assumptions on convexity or uniqueness of lower-level solution has not previously been proposed for adversarial evasion scenarios in regression analysis.

Let $x \in \mathbb{R}^q$ be a sample (row) of data containing $p \in \mathbb{N}$ features with corresponding label $y \in \mathbb{R}$. For some weights $w \in \mathbb{R}^q$, the learner's prediction function, $\sigma : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, as the linear weighted sum of the features:

$$\sigma(w, x) := w^T x. \tag{1}$$

This is a common prediction function in linear regression. We then construct a loss function, $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$, to train the prediction function and identify appropriate weights. As in linear regression, we can achieve this by minimising the squared error between the learner's prediction of x and the label, y,

$$\mathcal{L}(\sigma(w, x), y) := (\sigma(w, x) - y)^2 = (w^T x - y)^2. \tag{2}$$

The adversary, in the lower-level, transforms some data with the aim of having the learner's prediction function, σ output a desired target label. For example, consider again the example of an adversary lying about features of a housing property to influence a higher valuation than deserved. If, for example, the actual valuation price is $y \in \mathbb{R}$, the adversary might set their target, $z \in \mathbb{R}$, as $z = y + \nu$, where $\nu > 0$. The lower-level (adversary's) loss function is then defined as the squared distance between the learner's prediction and the target label:

$$\ell(\sigma(w, x), z) := (\sigma(w, x) - z)^2 = (w^T x - z)^2. \tag{3}$$

With the loss functions established, we now work them into objective functions that accommodate a full dataset. We divide the training data into two sets, a static set as would typically be used in the training of a machine learning predictor, and then a second set of data which can be manipulated by the adversary. Let $D \in \mathbb{R}^{n \times q}$ be the static set of $n \in \mathbb{N}$ instances of data where each $D_i \in \mathbb{R}^q$, $i = 1, \ldots, n$, is a row vector containing the values of $q \in \mathbb{N}$ features, and let $\gamma \in \mathbb{R}^n$ be the corresponding corresponding collection of labels. Let $X \in \mathbb{R}^{m \times q}$ be the set of $m \in \mathbb{N}$ instances of the same q features which can be manipulated by the adversary with corresponding labels $Y \in \{0, 1\}^m$. For convenience, we collapse

the adversary's data into a single column vector in order to reduce its dimensionality. Therefore, the adversary's data, $X \in \mathbb{R}^{mq}$ is refined as

$$X := \begin{pmatrix} X_1^T \\ \vdots \\ X_m^T \end{pmatrix},$$

where each X_i, $i = 1, \ldots, m$, is a row vector of features. For consistency, we also redefine the static data in the same way, $D \in \mathbb{R}^{nq}, D := (D_1, \ldots, D_n)^T$.

The upper level objective of the bilevel program sees the learner minimising their loss over both sets of data. Note that the learner considers the true labels of the adversary's data, where $F : \mathbb{R}^p \times \mathbb{R}^{mp} \to \mathbb{R}$ is the upper-level objective defined by

$$F(w, X) := \frac{1}{n} \sum_{i=1}^{n} \mathcal{L}(\sigma(w, D_i), \gamma_i) + \frac{1}{m} \sum_{i=1}^{m} \mathcal{L}(\sigma(w, X_i), Y_i). \tag{4}$$

Then, given a set of weights, w of the learner's prediction function, the adversary, in the lower-level seeks to modify their data such that the learner's prediction moves towards their target labels, Z. Let $f : \mathbb{R}^p \times \mathbb{R}^{mp} \to \mathbb{R}$ be the adversary's objective, which is defined as the sum of the adversary's loss over their data:

$$f(w, X) := \frac{1}{m} \sum_{i=1}^{m} \ell(\sigma(w, X_i), Z_i). \tag{5}$$

As we have highlighted through a motivating real estate example in Section 1, an adversary who takes smaller movements better reflects the strategies played by adversaries in the real world. However, the unconstrained objective functions do not take this under consideration. The optimal adversary under the objective function in (5) will overestimate the abilities of the adversary, leading unrealistic data transformations. In order to construct a realistic adversary within the bilevel program, we introduce a set of lower-level constraints on the adversary's optimization problem which measure the similarity between the true value of the adversary's data and the value seen by the learner. Let X^0 be the true (original) value of the adversary's data, and let $g : \mathbb{R}^{mq} \to \mathbb{R}^m$ be the vector of constraint functions

$$g(X) := \begin{pmatrix} g_1(X) \\ \vdots \\ g_m(X) \end{pmatrix}. \tag{6}$$

We define each constraint function component $g_i : \mathbb{R}^q \to \mathbb{R}$, for each instance of the adversary's data $i = 1, \ldots, m$, by

$$g_i(X) := \delta - d(X_i, X_i^0), \tag{7}$$

where $d : \mathbb{R}^q \times \mathbb{R}^q \to \mathbb{R}$ is some similarity function and δ is the minimum required similarity threshold. The constraint $g(X) \leq 0$ then restrict the adversary to produce data whose similarity is greater than δ. The work in [27] demonstrated the effectiveness of the cosine similarity to measure the extent to which the adversary has modified their data and so we use the same here,

$$d(X_i, X_i^0) := \frac{X_i \cdot X_i^0}{\|X_i\| \|X_i^0\|}, \ i = \{1, \ldots, m\}. \tag{8}$$

Combining the lower-level objective 5 with the constraints in 8 gives the adversary's optimisation problem, the solution to which provides data that trick the learner into giving the best possible label assignment while remaining plausibly realistic. Given a set of weights of the learner's prediction function, w, we defined $S(w)$ as this set of optimal data, given by

$$S(w) := \underset{X \in \mathbb{R}^{mq}}{\operatorname{argmin}} \left\{ f(w, X) \mid g(X) \leq 0 \right\}. \tag{9}$$

Note that the adversary's objective function under the linear regression loss in (3) and prediction function (1) possess multiple optimal solutions. This can be directly shown using [27, Proposition 1] since the prediction function, $\sigma(w, x) = w^T x$, is a linear combination of w and x. Note also that it has already been demonstrated in [27, Proposition 2] that the feasible region defined by the constraints in (8) is non-convex. Therefore, in the case of multiple optimal solutions to the adversary's problem, given by the set $S(w)$, we make the pessimistic assumption that the adversary will not act cooperatively, namely,

they will choose the solution which maximises the learner's objective, while the learner seeks to minimise it. The complete bilevel program is then given by

$$\min_{w \in \mathbb{R}^p} \max_{X \in S(w)} F(w, X). \tag{10}$$

A solution to the problem described by (9) - (10) will comprise the weights of a prediction that has accounted for adversarial manipulation during its training process. Consequently, the prediction function will be more resilient to evasion attacks made after implementation. We say that a set of weights $\bar{w}$ is a local optimal solution for problem (9) - (10) if there is a neighbourhood W of the point such that

$$\varphi_p(\bar{w}) \leq \varphi_p(w) \quad \text{for all} \quad w \in W, \tag{11}$$

where the φ_p denotes the following *two-level value function* (concept first introduced and studied in [10]):

$$\varphi_{\mathrm{p}}(w) := \max\{F(w, X) \mid X \in S(w)\}.$$

Note that the the two-level value function φ_{p} is typically non-convex. Hence, problem (9) - (10) is a nonconvex optimization problem. For such problems, algorithms usually only compute stationary points. Therefore, we aim to calculate stationary points for problem (9) - (10). If we have a local optimal solution, w, of the bilevel program, then, based on results in [9, 31] (and [27, Theorem 1] for more precise relevant calculations), under a suitable framework, we can find a Lagrange multiplier vector $(\lambda, \beta, \hat{\beta})$ such that the following stationarity conditions are satisfied:

$$\nabla_w F(w, X) = 0, \tag{12a}$$
$$\nabla_X F(w, X) - \lambda \nabla_X f(w, X) - \nabla g(X)^\top \beta = 0, \tag{12b}$$
$$\nabla_X f(w, X) + \nabla g(X)^\top \hat{\beta} = 0, \tag{12c}$$
$$\hat{\beta} \geq 0, \quad g(X) \leq 0, \quad \hat{\beta}^\top g(X) = 0, \tag{12d}$$
$$\lambda \geq 0, \ \beta \geq 0, \quad g(X) \leq 0, \quad \beta^\top g(X) = 0, \tag{12e}$$

for some data X belonging to the adversary. We can solve this system by first transforming it into a system of equations. To simplify the notation, we introduce the block variables

$$z := \begin{bmatrix} w \\ X \end{bmatrix} \in \mathbb{R}^{q+mq} \quad \text{and} \quad \xi := \begin{bmatrix} \beta \\ \hat{\beta} \\ \lambda \end{bmatrix} \in \mathbb{R}^{2b+1},$$

as well as the block functions

$$\mathcal{G}(z) := \begin{bmatrix} g(X) \\ g(X) \\ 0 \end{bmatrix} \quad \text{and} \quad \mathcal{H}(z, \xi) := \begin{bmatrix} \nabla_w F(w, X) \\ \nabla_X L_X^{\mathrm{P}}(z, \xi) \\ \nabla_X \ell^{\mathrm{P}}(z, \xi) \end{bmatrix},$$

where $L_X^{\mathrm{P}}, \ell^{\mathrm{P}} \colon (\mathbb{R}^q \times \mathbb{R}^{mq}) \times \mathbb{R}^{2b+1} \to \mathbb{R}$ are upper and lower-level Lagrangian type functions:

$$L_X^{\mathrm{P}}(z, \xi) := F(w, X) - \lambda f(w, X) - \beta^\top g(X),$$
$$\ell^{\mathrm{P}}(z, \xi) := f(w, X) + \hat{\beta}^\top g(X).$$

Based on this notation, the system described by (12a)–(12e) can then be restated as

$$\zeta \geq 0, \ \mathcal{G}(z) \leq 0, \ \zeta^T \mathcal{G}(z) = 0, \ \mathcal{H}(z, \xi) = 0, \tag{13}$$

which can equivalently be written as the following system of equations

$$\begin{cases} \mathcal{H}(z, \zeta) = 0, \\ \vartheta_{\mathrm{FB}}\left(\zeta_i, -\mathcal{G}_i(z, \zeta)\right) = 0, \ i = 1, \ldots, 2m. \end{cases} \tag{14}$$

Here, ϑ_{FB} corresponds to the so-called Fischer-Burmeister function [13], which is defined by

$$\vartheta_{\mathrm{FB}}(a, b) := \sqrt{a^2 + b^2} - (a + b) \quad \text{for} \quad (a, b) \in \mathbb{R}^2.$$

Thanks to this function, the second equation in (13) is equivalent to the complementarity conditions in (12d)–(12e), written in compact form in (13), while considering them in pairs.

Clearly, the stationary conditions (12a)–(12e) of problem (9) - (10) have been written as a system of equations in (14). This system is overdetermined, with m more equations than variables, making the Levenberg-Marquardt method, specifically, the global nonsmooth Levenberg–Marquardt method for mixed nonlinear complementarity systems developed in [17], a suitable choice to solve it. A version of this algorithm which has been appropriately modified for the pessimistic adversarial bilevel program by introducing additional stopping criteria, is given in [27, Algorithm 2]. We apply this same algorithm here to solve system (14), using the derivative formula provided in Appendix A, and extract the solution $z^* = (w^*, X^*)$. This solution comprises the weights, w^*, of a prediction function that has accounted for adversarial influence during the training process as well as the corresponding transformed adversary's data, X^*. Unlike the framework proposed in the previous work in [27], it is not required that the adversary's data is embedded ahead of analysis. Consequently, we can use X^* to observe the value of the adversary's solution in its original feature space and investigate the nature of their transformation.

3 Numerical experiments

We assess the resilience of the bilevel model to strategic adversarial modifications. To do so, we design experiments on two datasets, the first of which, named *Wine Quality* records 11 features of 4898 bottles of red wine such as acidity and alcohol content. A corresponding quality score between $1 - 10$ is measured and recorded for each bottle. For this dataset, we consider scenarios where wine producers might lie when reporting the features of the wine or perhaps bribe quality assessors to alter the results in order to appear to be producing wine of a higher quality than they actually are. The second dataset, named *Real Estate*, contains the values of 6 features of 414 houses such as their age and distance to the nearest train station along with the corresponding sale price. We consider scenarios where sellers might lie about features of the house in order to be able to charge a higher price than deserved. Further to this, we use the numerical experiments in this section to explore a particularly useful aspect of the constrained model which, due to the requirement of feature transformations, was not exploitable in previous works. Specifically, we demonstrate how, unlike existing pessimistic bilevel approaches to adversarial training, our models allow us to investigate which features of the adversary's data were changed and hence give us insight into the most vulnerable aspects of the prediction models.

We divide the data into a training, and test set by the ration of 80% and 20% respectively and simulate adversarial attacks to inject the into the test set. Let $X^{\text{test}} \in \mathbb{R}^{T \times p}$ be the test set containing T samples. Before evaluation, we simulate adversarial influence on the test set by transforming a portion of the instances towards target values $Z^{\text{test}} = Y^{\text{test}} + \Delta$, defined as a perturbation of the ground truth test labels Y^{test}, where $\Delta \in \mathbb{R}$ is a perturbation. Let $t <= T$ be the portion of data modified by an adversary and let $I \subset \{1, \ldots, T\}$ be the corresponding indexes of the test data which are manipulated. We simulate a range of adversarial abilities. We generate adversarial test data by modifying X_I^{test} towards the adversary's target labels. To simulate a range of abilities, we randomly generate each adversary their own similarity threshold from the range $(0.8, 1)$. Let $\delta^{\text{test}} \in (-1, 1)^t$ be the set of similarity thresholds where $\delta_i \sim U(0.8, 1) \, \forall \, i \in I$ and U denotes the uniform distribution. The instances of adversarial test data and then found by solving the problem

$$X_i^{\text{test}} := \operatorname*{argmin}_{x \in \mathbb{R}^q} \left\{ \ell\left(\sigma(w^{\text{init}}, x), Z_i^{\text{test}}\right) \,\middle|\, \delta_i^{\text{test}} - \frac{x \cdot Z_i^{\text{test}}}{\|x\| \|Z_i^{\text{test}}\|} \leq 0 \right\}, \, \forall i \in I.$$

where $x^0 \in \mathbb{R}^q$ is the initial values of x. We set T to be 10% of the size of the test set. All features and labels are normalised to the range $(0, 1)$ and we set the adversarial perturbation to be $\Delta = 2\text{std}(Y^{\text{test}})$ where $\text{std}(Y^{\text{test}})$ is the standard deviation of the test labels.

We compare our model to that of a typically trained linear regression predictor, named *LinReg*. While the pessimistic model in [4] was intended for use on classification tasks, its prediction function and loss functions can be easily substituted for linear regression variants which still satisfy their assumptions about strict convexity. We name this *B&S* and use it as a comparison to the existing pessimistic approach to adversarial regression analysis. The mean square error for these models as well as our model for various values of the adversary's sample size, m, and the similarity threshold, δ, are plotted in Figure 1. We can see for both datasets a similar trend of an initially high mean square error (MSE) for low values of m, before a decrease to an optimal value. Following this, we observe an increase in MSE, before plateauing around an MSE roughly equal to that of the traditional predictor. Both experiments see an optimal MSE when $\delta = 0.95$ and when m is fairly low. For the *wine* dataset, the optimal value falls at $m = 1$, while on the *real estate* dataset, we see the optimal value fall at $m = 2$. Although, we note that the

real estate dataset also sees similarly good performance for m in the range of $13 - 23$. In general, we observe a general pattern in that the adversary should be granted enough freedom and influence over the training process to capture the nature of adversarial attacks, while not too much influence that the model becomes over-estimates the power of the adversary and hence suffers in performance.
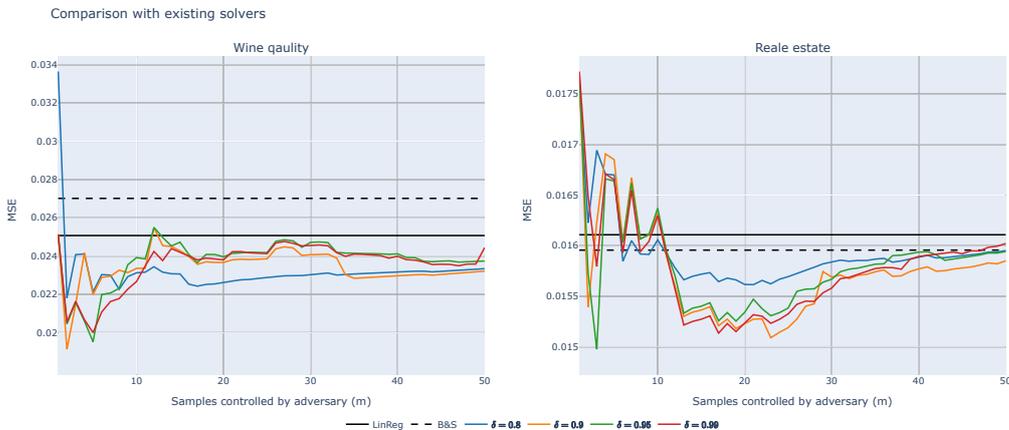

Figure 1: Comparison of the pessimistic bilevel model with existing solvers

We now investigate the movement of each feature by the adversary during the training process to gain insight into which features are most vulnerable. For each feature (column), we calculate the average absolute distance between the adversary's solution X^* and the original position X^0,

$$\frac{1}{m} \sum_{j=1}^{m} \left| X_{ij}^* - X_{ij}^0 \right|, \quad i = 1, \ldots, m.$$

Under this measure, a larger distance implies that the adversary needed to transform the data by a larger extent in order to sufficiently fool the leaner's prediction model. We plot the average movements for each feature and for each dataset in Figure 2. For the *Wine quality*, we see the most movement in residual sugar followed by PH and sulfur dioxide. These results give the insight into how much an adversary must modify each feature in order to fool the learner. It is clear, for example, that the alcohol content must be modified considerably to generate the most effective wine. While chlorides, on the other hand, required little change to achieve a value sufficient to fool the learner's predictor. Chlorides, therefore, might be considered a particularly vulnerable feature that could easily be exploited and perhaps should be treated with more scrutiny. In the *real estate* data, we see the distance to the nearest MRT station as the most resilient feature, requiring relatively substantial change to affect the learner's predictor. Meanwhile, the transaction data and longitude appear to be the easiest to exploit.
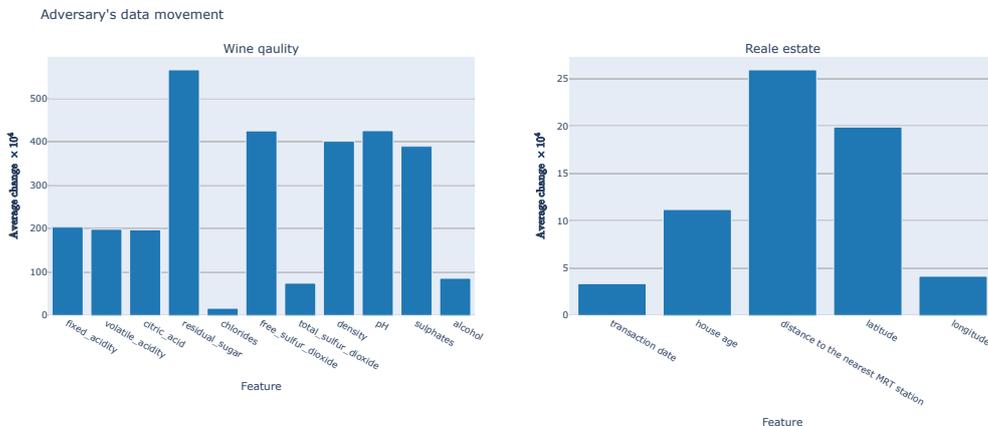

Figure 2: Feature modifications made by the adversary to trick the learner

It is clear from the experimental results that the pessimistic bilevel model with constrained adversarial movement provides an effective method of accounting for adversarial manipulation during the training

process of regression scenarios. The lower-level successfully anticipates the movements of an adversary, which leads to improved performance by the prediction model trained in the upper-level. We see a significant change in performance as we increase the number of instances available to be manipulated by the adversary. In particular, we note that a balance needs to be struck between allowing the adversary enough instances that they have sufficient influence over then training process, while not too many that the model overestimates the influence of the adversary. Finally, we demonstrated how, unlike existing works, we could investigate the values of the adversary's data at the solution. This allowed us to identify which features were most vulnerable to exploitation.

4   Conclusion

In this article, we proposed a pessimistic bilevel program to model adversarial evasion in regression scenarios. In particular, by anticipating adversarial movement in the lower-level, we can train prediction functions that are resilient to adversaries who attempt to influence particular outcomes from a prediction model by strategically transforming their data. For example, in the context of property valuation, an adversary might purposefully provide falsified information to achieve higher valuations and hence charge a higher selling price. While there exist pessimistic bilevel approaches to adversarial evasion scenarios, these models exploit strong assumptions about the convexity of the adversary's problem and the uniqueness of their solution to reformulate the program into its optimistic variant, which, while easier to solve, oversimplifies the capabilities of the adversary. Since our model makes no such assumptions, we retain the pessimistic aspect of the bilevel model, allowing us to accurately capture the antagonistic nature of these scenarios. Furthermore, with the introduction of lower-level constraints, which restrict the adversary's movement, preventing nonsensical transformations and ensuring the adversary's data remains plausibly realistic, leading to more accurate prediction functions.

We simulated adversarial influence to create test sets for wine quality appraisal and real estate pricing. Numerical experiments demonstrated the ability of the bilevel model to train resilient predictors that provided improved performance over existing methods. Further to this, we investigated varying the number of samples available to the adversary and observed a clear pattern in the form of a trade-off between allowing the adversary enough freedom and influence to generate sufficient movement that impacts the training process, while not too much such that the adversary provides too much influence over the training process, leading to overly pessimistic predictors. Additionally, we investigated a feature of the pessimistic bilevel model which previous works were not able to exploit. Specifically, since we do not transform the adversary's data before analysis, we were able to observe the adversary's solution in its original feature space and measure the extent to which the adversary modified each feature. From this we could identify which features were most vulnerable to attacks.

References

References

[1] David Benfield, Stefano Coniglio, Martin Kunc, Phan Tu Vuong and Alain Zemkoho. Classification under strategic adversary manipulation using pessimistic bilevel optimisation. (2024) arXiv: https://arxiv.org/abs/2410.20284.

[2] Battista Biggio, Igino Corona, Davide Maiorca, Blaine Nelson, Nedim ˘Srndi´c, Pavel Laskov, Giorgio Giacinto, and Fabio Roli. Evasion Attacks against Machine Learning at Test Time. *Machine Learning and Knowledge Discovery in Databases.* Springer Berlin Heidelberg, (2013), p. 387-402.

[3] Michael Br¨uckner and Tobias Scheffer. Nash Equilibria of Static Prediction Games. *Proceed- ings of the 22nd International Conference on Neural Information Processing Systems.* NIPS'09, (2009).

[4] Michael Br¨uckner and Tobias Scheffer. Stackelberg games for adversarial prediction problems. *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining - KDD '11.* ACM Press, (2011).

[5] Aneesh Sreevallabh Chivukula and Wei Liu. Adversarial learning games with deep learning models. *2017 International Joint Conference on Neural Networks.* IJCNN, (2017).

[6] Joana C. Costa, Tiago Roxo, Hugo Proença, and Pedro Ricardo Morais Inácio. How Deep Learning Sees the World: A Survey on Adversarial Attacks & Defense. *IEEE Access.* (2024).

[7] Nilesh Dalvi, Pedro Domingos, Mausam, Sumit Sanghai, and Deepak Verma. Adversarial Classification. *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* KDD '04. (2004).

[8] Prithviraj Dasgupta and Joseph B. Collins. A Survey of Game Theoretic Approaches for Adversarial Machine Learning in Cybersecurity Task. (2019) *AI Mag.* 40.2 (2019) pp. 31–43.

[9] Stephan Dempe, Boris S Mordukhovich, and Alain B Zemkoho. Necessary optimality conditions in pessimistic bilevel programming. *Optimization.* 63.4. (2014). p. 505-533.

[10] Stephan Dempe, Boris S Mordukhovich, and Alain B Zemkoho. Sensitivity analysis for two-level value functions with applications to bilevel programming. *SIAM Journal on Optimization.* 22.4. (2012). p. 1309-1343.

[11] Kevin Eykholt, Ivan Evtimov, Earlence Fernandes, Bo Li, Amir Rahmati, Chaowei Xiao, Atul Prakash, Tadayoshi Kohno, and Dawn Song. Robust Physical-World Attacks on Deep Learning Visual Classification. *Proc. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition.* (2018). p. 1625-1634.

[12] Samuel Finlayson et a. Adversarial Attacks Against Medical Deep Learning Systems. (2019) arXiv: https://arxiv.org/abs/1804.05296.

[13] Andreas Fische. A special Newton-type optimization method. *Optimization.* (1992). p. 269-284.

[14] Ian J. Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and Harnessing Adversarial Examples. (2015) *Proc. 2015 International Conference on Learning Representations.* (2015).

[15] Chan Y. Han, Brian J. Lunday, and Matthew J. Robbins. A Game Theoretic Model for the Optimal Location of Integrated Air Defense System Missile Batteries. *INFORMS Journal on Computing.* 28.3 (2016), pp. 405–416.

[16] Ling Huang, Anthony D. Joseph, Blaine Nelson, Benjamin I.P. Rubinstein, and J. D. Tygar. Adversarial Machine Learning. *Proceedings of the 4th ACM Workshop on Security and Artificial Intelligence.* Association for Computing Machinery (2011) pp. 43-58.

[17] Lateef O Jolaoso, Patrick Mehlitz, and Alain B Zemkoho. A fresh look at nonsmooth Levenberg–Marquardt methods with applications to bilevel optimization. *Optimization.* 74.12 (2025) pp. 2745-2792.

[18] Murat Kantarcıoğlu, Bowei Xi, and Chris Clifton. Classifier evaluation and attribute selection against active adversaries. *Data Mining and Knowledge Discovery.* 22.1-2 (2010) pp. 291-335.

[19] Wei Liu and Sanjay Chawla. A Game Theoretical Model for Adversarial Learning. *Proc. 2009 IEEE International Conference on Data Mining Workshops.* (2009) pp. 25-30.

[20] Wei Liu, Sanjay Chawla, James Bailey, Christopher Leckie, and Kotagiri Ramamohanarao. An Efficient Adversarial Learning Strategy for Constructing Robust Classification Boundaries. *AI 2012: Advances in Artificial Intelligence.* (2012) pp. 469-660.

[21] Shike Mei and Xiaojin Zhu. Using machine teaching to identify optimal training-set attacks on machine learner. *Proc. Twenty-Ninth AAAI Conference on Artificial Intelligence.* AAAI Press (2015) pp. 2871-2877.

[22] Dale Schuurmans and Martin Zinkevich. Deep Learning Games. *Proceedings of the 30th International Conference on Neural Information Processing Systems.* NIPS'16 (2016) pp. 1686-1694.

[23] Robin Sommer and Vern Paxson. Outside the Closed World: On Using Machine Learning for Network Intrusion Detection. *Proc. 2010 IEEE Symposium on Security and Privacy.* (2010) pp. 305-316.

[24] Aneesh Sreevallabh Chivukula, Xinghao Yang, Bo Liu, Wei Liu, and Wanlei Zhou. Adversarial Machine Learning: Attack Surfaces, Defence Mech- anisms, Learning Theories in Artificial Intelligence. *Proc. 2010 IEEE Symposium on Security and Privacy.* Springer (2023).

[25] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. (2014) *Proc. 2014 International Conference on Learning Representations.* (2014).

[26] Shixiang Zhu, Henry Shaowu Yuchi, Minghe Zhang, and Yao Xie. Sequential Adversarial Anomaly Detection for One-Class Event Data. *INFORMS Journal on Data Science.* 2.1 (2023). pp. 45-59.

[27] David Benfield, Stefano Coniglio, Vuong Phan and Alain Zemkoho. Facing Adversarial Data Manipulation via Constrained Pessimistic Bilevel Optimization. (2025) arXiv: https://arxiv.org/abs/2510.03254.

[28] Scott Alfeld, Xiaojin Zhu, and Paul Barford. Data poisoning attacks against autoregressive model. *Proc. of the Thirtieth AAAI Conference on Artificial Intelligence.* AAAI Press (2016) pp. 1452–1458

[29] Michael Großhans, Christoph Sawade, Michael Br¨uckner, and Tobias Scheffer. Bayesian Games for Adversarial Regression Problems. *Proc. of the 30th International Conference on Machine Learning.* Proceedings of Machine Learning Research (2013) pp. 55–63

[30] Liang Tong, Sixie Yu, Scott Alfeld, and yevgeniy vorobeychik. Adversarial Regression with Multiple Learners. *Proc. of the 35th International Conference on Machine Learning.* Proceedings of Machine Learning Research (2018) pp. 4946–4954

[31] Stephan Dempe, Boris S Mordukhovich, and Alain B Zemkoho. Two-level value function approach to non-smooth optimistic and pessimistic bilevel programs. *Optimization.* 68.2-3 (2019) pp. 433–455.

A   Derivatives for the leader and follower

Let $x \in \mathbb{R}^q$ be a sample of data with corresponding label $y \in \mathbb{R}$ and let $w \in \mathbb{R}^q$ be the weights of the learner's prediction function, $\sigma : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ defined as in (1). We define the upper-level (learner's) loss function $\mathcal{L} : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ as it is in (2). Let $z \in \mathbb{R}$ be the adversary's target label.

Let $D \in \mathbb{R}^{mq}$ be the static data with corresponding labels $\gamma \in \mathbb{R}^n$ and let $X \in \mathbb{R}^{mq}$ be the adversary's data with corresponding labels $Y \in \mathbb{R}^m$. The upper-level objective $F : \mathbb{R}^q \times \mathbb{R}^{mq} \to \mathbb{R}$ is defined as it is in (4). The derivative of the upper-level (leader's) loss with respect to the learner's weights is given by,

$$\nabla_w F(w, X) = \frac{2}{n} D^T (w^T D - \gamma) + \frac{2}{m} X^T (w^T X - Y) + \frac{2}{\rho} w.$$

The second derivative of the upper-level loss with respect to the learner's weights is given by

$$\nabla_{ww}^2 F(w, X) = \frac{2}{n} D^T D + \frac{2}{m} X^T X + \frac{2}{\rho}$$

The derivative of the upper-level objective function with respect to the adversary's data is given by

$$\frac{\partial F}{\partial X_{ij}}(w, X) := \frac{1}{m} \sum_{k=1}^{m} \frac{\partial \mathcal{L}}{\partial X_{ij}}(\sigma(w, X_k), Y_k) = \frac{1}{m} \frac{\partial \mathcal{L}}{\partial X_{ij}}(\sigma(w, X_i), Y_i),$$

where

$$\frac{\partial \mathcal{L}(w, X)}{\partial X_{ij}} = \frac{2}{m} w_j (w^T X_i - y_i), \quad i = 1, \ldots, m, \ j = 1, \ldots, q.$$

The second derivative with respect to the adversary's data is given by the following cases,

$$\frac{\partial^2 \mathcal{L}}{\partial X_{ij} \partial X_{kl}} = \begin{cases} \frac{2}{m} w_j w_l & i = k \\ 0 & \text{otherwise.} \end{cases}$$

The derivative of the upper-level objective function with respect to the adversary's data is given by

$$\frac{\partial f}{\partial X_{ij}}(w, X) = \frac{1}{m} \sum_{i=k}^{m} \frac{\partial \ell}{\partial X_{ij}}(\sigma(w, X_k), Y_k) = \frac{1}{m} \frac{\partial \ell}{\partial X_{ij}}(\sigma(w, X_k), Y_k).$$

where the lower-level (adversary's) loss function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ is defined as it is in (3). Note that the adversary's loss function can be expressed in terms of the leader's loss function,

$$\ell(\sigma(w, x), z) = \mathcal{L}(\sigma(w, x), z),$$

for some weights $w \in \mathbb{R}^q$, data $x \in \mathbb{R}^q$ and target label $z \in \mathbb{R}$. Therefore, we can express the derivative of the lower-level objective function with respect to the adversary's data as follows

$$\frac{\partial f}{\partial X_{ij}}(w, X) = \frac{1}{m} \frac{\partial \mathcal{L}}{\partial X_{ij}}(\sigma(w, X_i), Z_i) \quad i = 1, \ldots, m, \ j = 1, \ldots, q.$$

The second derivative with respect to the adversary's data is then given as

$$\frac{\partial^2 f}{\partial X_{ij} \partial X_{kc}}(w, X) = \frac{1}{m} \frac{\partial^2 \mathcal{L}}{\partial X_{ij} \partial X_{jk}}(\sigma(w, X_i), Z_i) \quad i, k = 1, \ldots, m, \ j, c = 1, \ldots, q.$$

Finally, the derivative with respect to the learner's weights and the adversary's data is given by

$$\frac{\partial^2 f}{\partial w_i \partial X_{jk}}(w, X) = \frac{1}{m} \frac{\partial^2 \mathcal{L}}{\partial w_i \partial X_{jk}}(\sigma(w, X_i), Z_i) \quad i, j = 1, \ldots, m, \ k = 1, \ldots, q.$$

Let $X^0 \in \mathbb{R}^{mq}$ be the start point of the adversary's data and let the lower-level constraints $g : \mathbb{R}^{mq} \to (-1, 1)$ be defined as in (6), where each constraint function $g_i(X) : \mathbb{R}^q \to \mathbb{R}$, $i = 1, \ldots, m$ measures the cosine similarity between the adversary's data and its original position, as given by (8), where $\delta \in \mathbb{R}$ is the similarity threshold. The derivative of the constraints with respect to the classifier weights is 0. The derivative with respect to the adversary's data is obtained as

$$\frac{\partial g_i(X)}{\partial X_{jk}} = \begin{cases} \frac{X_{ik}^0}{\|X_i\| \cdot \|X_i^0\|} - d(X_i, X_i^0) \frac{X_{ik}}{\|X_i\|^2} & i = j, \\ 0 & i \neq j. \end{cases}$$

The second derivative with respect to the adversary's data is given by the cases

$$\frac{\partial^2 g_i(X)}{\partial X_{jk} \partial X_{lc}} = \begin{cases} \frac{X_{ic} X_{ik}^0 + X_{ik} X_{ic}^0}{\|X_i\|^3 \|X_i^0\|} - \frac{3 X_{ik} X_{ic} d(X_i, X_i^0)}{\|X_i\|^4} & i = j = l, \ k \neq c, \\[2mm] \frac{2 X_{ik} X_{ik}^0}{\|X_i\|^3 \|X_i^0\|} - \frac{3 X_{ik}^2 d(X_i, X_i^0)}{\|X_i\|^4} + \frac{d(X_i, X_i^0)}{\|X_i\|^2} & i = j = l, \ k = c, \\[2mm] 0 & \text{otherwise.} \end{cases}$$