An Adversarial Attacker for Neural Networks in Regression Problems

Kavya Gupta, Beatrice Pesquet-Popescu, Fateh Kaakai, Jean-Christophe Pesquet, Fragkiskos D. Malliaros

An Adversarial Attacker for Neural Networks in Regression Problems

Kavya Gupta[1,2*] , **Beatrice Pesquet-Popescu**[2] , **Fateh Kaakai**[2] ,
Jean-Christophe Pesquet[1] and **Fragkiskos D. Malliaros**[1]
[1]Université Paris-Saclay, CentraleSupélec, Inria
Centre de Vision Numérique, Gif-sur-Yvette, France
[2]Air Mobility Solutions BL, Thales LAS France
kavya.gupta100@gmail.com, beatrice.pesquet@thalesgroup.com, fateh.kaakai.e@thalesdigital.io,
{jean-christophe.pesquet, fragkiskos.malliaros}@centralesupelec.fr

Abstract

Adversarial attacks against neural networks and their defenses have been mostly investigated in classification scenarios. However, adversarial attacks in a regression setting remain understudied, although they play a critical role in a large portion of safety-critical applications. In this work, we present an adversarial attacker for regression tasks, derived from the algebraic properties of the Jacobian of the network. We show that our attacker successfully fools the neural network, and we measure its effectiveness in reducing the estimation performance. We present a white-box adversarial attacker to support engineers in designing safety-critical regression machine learning models. We present our results on various open-source and real industrial tabular datasets. In particular, the proposed adversarial attacker outperforms attackers based on random perturbations of the inputs. Our analysis relies on the quantification of the fooling error as well as various error metrics. A noteworthy feature of our attacker is that it allows us to optimally attack a subset of inputs, which may be helpful to analyse the sensitivity of some specific inputs.

1 Introduction

Adversarial machine learning has received an increased attention in the past decade. For all machine learning models, defense against adversarial attacks is important in terms of safety. Adversarial attacks in classification constitute malicious attempts to trick a model classifier. They play a critical role in real-world application domains such as spam/malware detection, autonomous systems [Huang and Wang, 2018], [Eykholt *et al.*, 2018], [Ren *et al.*, 2019], medical systems [Finlayson *et al.*, 2018] etc. Adversarial attacks cause vulnerability in model deployment and specially needs to be taken into account in deployment of security-critical AI applications. Despite the newfound interest of the research community in trustworthy and explainable AI, there are only few works investigating the adversaries in the case of regression tasks.

Input	0	Speed	continuous
	1	Flight Distance	
	2	Departure Delay	
	3	Initial ETE	
	4	Latitude Origin	
	5	Longitude Origin	
	6	Altitude Origin	
	7	Latitude Destination	
	8	Longitude Destination	
	9	Altitude Destination	
	10	Arrival Time Slot	7 slots (categorical)
	11	Departure Time Slot	7 slots (categorical)
	12	Aircraft Category	6 classes (categorical)
	13	Airline Company	19 classes (categorical)
Output	3	Refinement ETE	continuous

Table 1: Input and output variables description for Industrial dataset – A safety critical application.

Current advances in the adversarial machine learning field evolve around the issue of designing attacks and defenses with focus on the use of neural networks in image analysis and computer vision [Goodfellow *et al.*, 2014], [Kurakin *et al.*, 2016]. Much less works concern tabular data. However, most machine learning tasks in the industry rely on tabular data, e.g., fraud detection, product failure prediction, anti-money laundering, recommendation systems, click-through rate prediction, or flight arrival time prediction.

In this paper, we focus on generating adversarial attacks for neural networks in the specific scenario when *i*) a regression task is performed and *ii*) tabular data are employed. Our contributions are the following:

- We propose a simple, novel and flexible method for generating adversarial attacks for regression tasks (a white box attack).
- We show that the proposed attacker allows us to optimally attack on any given subset of input features.
- We explore various error metrics which are useful for analysing these adversarial attacks.
- Our proposed adversarial attacker is generalised for any ℓ_p norm on input and output perturbations.

*Contact Author

- We evaluate our results on open-source regression datasets and an industrial dataset (output and input features described in the Table 1) which lies in the domain of safety critical applications.

In Section 2, we give a brief overview of existing works. In Section 3, we formulate the problem and present our method for generating adversarial examples in regression tasks. In Section 4, we perform numerical experiments on four datasets to demonstrate the effectiveness of the proposed attacker. Some concluding remarks are given in Section 5.

2 Related Work

In [Szegedy *et al.*, 2013] the concept of adversarial attacks was first proposed to fool DNNs. Adding a subtle perturbation to the input of the neural network produces an incorrect output, while human eyes cannot recognize the difference in the modification of the input data. Even though different models have different architectures and might use different training data, the same kind of adversarial attack strategies can be used to attack related models. These attacks pose a huge threat to the performance of DNNs. [Szegedy *et al.*, 2013] paper proposed L-BFGS to construct adversarial attacks and since then there has been plethora of works introducing various adversarial attacks and their defenses for DNNs.

[Goodfellow *et al.*, 2014] proposed a simpler and faster method to construct adversarial attacks (FGSM). The generated images are misclassified by adding perturbations and linearizing the cost function in the gradient direction. This is a non-iterative attack, hence it has a lower computation cost than the previous method. The Fast Gradient Sign Method (FGSM) is an ℓ_∞ bounded attack and is often prone to label leaking.

It may be difficult for FGSM to control the perturbation level in constructing attacks. [Kurakin *et al.*, 2016] proposed an optimized FGSM, termed Iterative Gradient Sign Method (IGSM), which adds perturbations in multiple smaller steps and clips the results after each iteration ensuring that the perturbations are restricted to the neighborhood of the example. [Dong *et al.*, 2018] added momentum to IGSM attacks. [Papernot *et al.*, 2016] proposed the Jacobian-based Saliency Map Attack (JSMA), which is based on the ℓ_0 sparsity measure. The basic idea is to construct a saliency map with the gradients and model the gradients based on the impact of each pixel of the image.

[Moosavi-Dezfooli *et al.*, 2016] proposed a non-targeted attack method based on the ℓ_2-norm, called DeepFool. It tries to find the decision boundary that is the closest to the sample in the image space, and then use the classification boundary to fool the classifier. FGSM, JSMA, and DeepFool are designed to generate adversarial attacks corresponding to single image to fool the trained classifier model. [Moosavi-Dezfooli *et al.*, 2017] proposed a universal image-agnostic perturbation attack method which fools classifier by adding a single perturbation to all images in the dataset. [Carlini and Wagner, 2017] proposed a powerful attack based on L-BFGS. The attack can be generated according to ℓ_1 , ℓ_2, and ℓ_∞ norm which can be targeted or non-targeted. [Liu *et al.*, 2016] proposed an ensemble attack method combining multiple models to construct adversarial attacks. [Rony *et al.*, 2020] proposed a method to generate minimally perturbed adversarial examples based on Augmented Lagrangian for various distance metrics. In [Balda *et al.*, 2018], authors propose a general framework for generation of adversarial examples in both classification and regression tasks for applications in image domain. Similar to our proposed approach, the technique is based on the Jacobian of the neural network. Most of the methods in the literature about adversarial example generation belong to the class of white box attackers, i.e., the attacker has access to the information related to the trained neural network model including the model architecture and its parameters. A black box attacker is introduced in [Su *et al.*, 2019]. Such attackers do not know the model but can interact with it. A byproduct of black-box attack is grey-box attack where attackers might have limited information regarding the model. To the best of our knowledge the only work dealing with adversarial attacks in white box settings for tabular data has been proposed in [Ballet *et al.*, 2019] and this work handles only classification tasks.

In regression tasks there are no natural margins as in the case of classification tasks, and adversarial learning in regression setting is hindered with difficulties to define the adversarial attacks, its success, and evaluation metrics. Despite the number of works in adversarial attack generation, there are few articles dealing with regression tasks.[Tong *et al.*, 2018] looked at adversarial attacks in the setting of an ensemble of multiple learners, investigating the interactions between these linear learners and an attacker in regression setting, modeled as a Multi-Learner Stackelberg Game (MLSG). However, the investigated linear case is not able to capture the larger class of non-linear models. The focus only on specific applications of regression is a common. [Ghafouri *et al.*, 2018] examined an important problem: selecting an optimal threshold for each sensor against an adversary for regression tasks in cyber-physical systems. [Deng *et al.*, 2020] introduced the concept of adversarial threshold which is related to a deviation between the original prediction and the prediction of adversarial example, i.e., an acceptable error range in driving models. In a regression context, [Nguyen and Raff, 2018] introduced a defense that is generically useful to reduce the effectiveness of adversarial attacks. They consider adversarial attacks as a potential symptom of numerical instability in the learned function. In the next section, we propose a general white-box adversarial attacker based on Jacobian of the learned function for regression tasks in tabular data domain.

3 Proposed Method

3.1 Objective

The problem of adversarial attacks is closely related to the robustness issue for a neural network, i.e. its sensitivity to perturbations. Let $T\colon \mathbb{R}^{N_0} \to \mathbb{R}^{N_m}$ be the considered neural network having N_0 scalar inputs and N_m scalar outputs. If $x \in \mathbb{R}^{N_0}$ is a given vector of inputs for some data for which y is the associated target output, the network has been trained to produce an output $T(x)$ close to y. If the input is now perturbed by an additive vector $e \in \mathbb{R}^{N_0}$, the perturbed output

is $T(x+e)$. Attacking the network then amounts to finding a perturbation e of preset magnitude which makes the output of the network to maximally deviate from a reference output. This reference output may be the model output $T(x)$ or the ground truth output y. Since our purpose is to develop an approach which remains efficient even if the accuracy of the network is not very high, we choose y as the reference output when available. In this context, the measures of deviation and of magnitude of the perturbation play an important role in terms of mathematical formulation of the problem. As a standard choice, the measure of perturbation magnitude will be here an ℓ_p-norm where $p \in [1, +\infty]$. For measuring the output deviation, we will similarly consider an ℓ_q-norm where $q \in [1, +\infty]$. It must be emphasized that this choice makes sense when dealing with regression problems. In this context, the ℓ_2 or the ℓ_1 norms are indeed frequently used as loss functions for training. On the other hand, the $\ell_{+\infty}$ norm is also a popular measure when dealing with reliability issues.

3.2 Optimization formulation

In the described setting, the design of the attacker can be formulated as the problem of finding the "worst perturbation" $\widehat{e}$ such that

$$\widehat{e} \in \underset{e \in C_{p,\delta}}{\text{Argmax}} \ \|T(x+e) - y\|_q, \tag{1}$$

where $C_{p,\delta}$ is the closed and convex set defined as

$$C_{p,\delta} = \{e \in \mathbb{R}^{N_0} \mid \|\Sigma^{-1/2} e\|_p \leq \delta\}. \tag{2}$$

$\Sigma \in \mathbb{R}^{N_0 \times N_0}$ is symmetric positive definite matrix. δ is a parameter which controls the maximum allowed perturbation and Σ is a weighting matrix typically corresponding to the covariance matrix of the inputs. For instance, if we assume that it is a diagonal matrix, it simply introduces a normalization of the perturbation components with respect to the standard deviations of the associated inputs.

For standard choices of activation functions, T is a continuous function. By virtue of Weierstrass theorem, the existence of a solution (not necessarily unique) to Problem (1) is then ensured. Although $C_{p,\delta}$ is a relatively simple convex set, this problem appears as a difficult non-convex problem due to the fact that *i*) T is a complex *nonlinear* operator, *ii*) we *maximize* an ℓ_q measure which, in addition, leads to a *nonsmooth* cost function when $q = 1$ or $q = +\infty$. A further difficulty is that we usually need to attack a large dataset to evaluate the robustness of a network and the provided optimization algorithm should therefore be fast.

3.3 Algorithm

We propose to implement a two-step approach.

- **Step 1.** We first perform a linearization based on the following first-order Taylor expansion:

  $$T(x+e) \simeq T(x) + J(x)e, \tag{3}$$

  where $J(x) \in \mathbb{R}^{N_m \times N_0}$ is the Jacobian of the network at x[1]. Note that $J(x)$ can be computed by classical back-propagation techniques. We will make a second approximation, that is $y \simeq T(x)$. Based on these two approximations and after the variable change $e' = \delta^{-1}\Sigma^{-1/2}e$, Problem (1) simplifies to

  $$\underset{e' \in B_p}{\text{maximize}} \ \|J(x)\Sigma^{1/2}e'\|_q, \tag{4}$$

  where B_p is the closed ℓ_p ball centered at 0 and with unit radius. Note that the optimal cost value in (4) is the subordinate norm of matrix $J(x)\Sigma^{1/2}$ when the input space is equipped with the ℓ_p norm and the output space with the ℓ_q one. We recall that this subordinate norm is defined, for every matrix $M \in \mathbb{R}^{N_m \times N_0}$, as

  $$\|M\|_{p,q} = \sup_{z \in \mathbb{R}^{N_0} \setminus \{0\}} \frac{\|Mz\|_q}{\|z\|_p}. \tag{5}$$

  Problem (4) is thus equivalent to find a vector $\widehat{e}'$ for which the value of the cost function is equal to $\|J(x)\Sigma^{1/2}\|_{p,q}$. For values of (p, q) listed below the expression of such vector has an explicit form.

  - If $p = q = 2$, $\widehat{e}'$ is any unit ℓ_2 norm eigenvector of $\Sigma^{1/2}J(x)^\top J(x)\Sigma^{1/2}$ associated with the maximum eigenvalue of this matrix. This vector can be computed by performing a singular value decomposition of $J(x)\Sigma^{1/2}$.
  - If $p = 2$ and $q = +\infty$, $\widehat{e}'$ is any unit ℓ_2 norm vector colinear with a row of $J(x)\Sigma^{1/2}$ having maximum ℓ_2 norm.
  - If $p = +\infty$ and $q = +\infty$, $\widehat{e}'$ is a unit norm vector whose elements are equal to $(\epsilon^{(i)})_{1 \leq i \leq N_0}$ where, for every $i \in \{1, \ldots, N_0\}$, $\epsilon_i \in \{-1, 0, 1\}$ is the sign of the i-th element of a row of $J(x)\Sigma^{1/2}$ with maximum ℓ_1 norm.
  - If $p = 1$ and $q = 1$, $\widehat{e}'$ is a vector which has only one nonzero component equal to ± 1, the index of this component corresponds to the column of $J(x)\Sigma^{1/2}$ with maximum ℓ_1 norm.
  - If $p = 1$ and $q = 2$, $\widehat{e}'$ is a vector with only one nonzero component equal to ± 1. The index of this component corresponds to a column of $J(x)\Sigma^{1/2}$ with maximum ℓ_2 norm.
  - If $p = 1$ and $q = +\infty$, $\widehat{e}'$ is again a vector with only one nonzero component equal to ± 1. The index of this component corresponds to a column of $J(x)\Sigma^{1/2}$ where is located an element of maximum absolute value.

- **Step 2.** In the previous optimization step, the optimal solution is not unique. Indeed if $\widehat{e} = \delta\Sigma^{1/2}\widehat{e}'$ is a solution to Problem (4), then $-\widehat{e}$ is also a solution. In addition, there may exist other reasons for the multiplicity of the solutions. For example, there may be several maximum norm rows for matrix $J(x)\Sigma^{1/2}$. Among all the possible choices, we propose to choose the solution $\widehat{e}$ leading to the maximum deviation w.r.t. the ground truth, that is such that $\|T(x+\widehat{e}) - y\|_q$ is maximum. This requires

[1] We assume that $J(x)$ is defined at x, see [Bolte and Pauwels, 2020] for a justification of this assumption in the nonsmooth case.

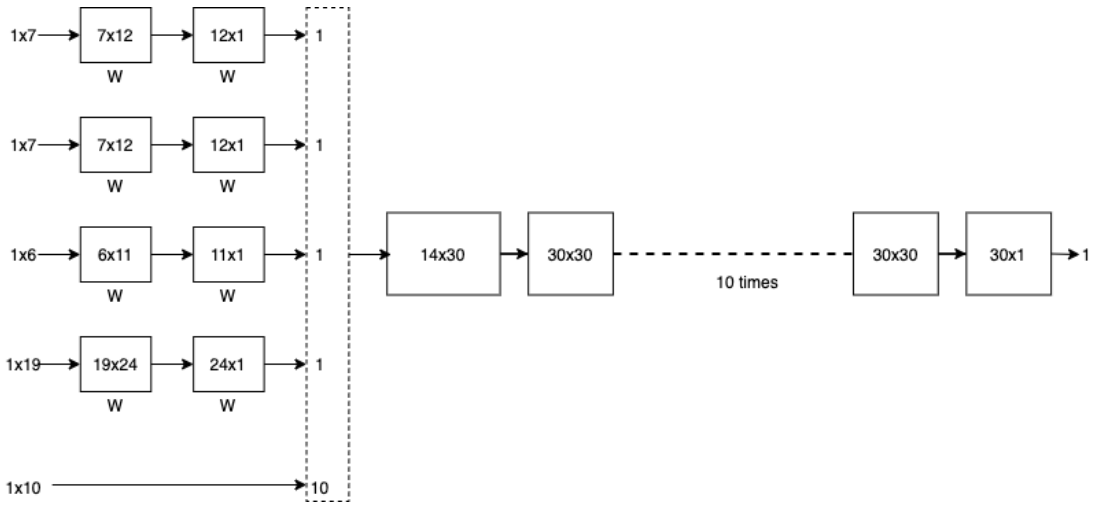

Figure 1: Network Architecture.

to perform a search on a small number of possible candidates. Note that no approximation error is involved in this step. If the ground truth for the output is not available, it can be replaced by the model output.

- **Post-optimization.** If $1 < q < +\infty$ and T is assumed to be differentiable, $e \mapsto \|T(x+e) - y\|_q^q$ is a differentiable function. A further refinement consists of minimizing this function over $C_{p,\delta}$ by using a projected gradient algorithm with Armijo search for the stepsize. The previous estimates of $\widehat{e}$ can then be used to initialize the algorithm. According to our numerical tests, implementing this strategy when $q = 2$ only brings a marginal improvement. Moreover, this approach cannot be used when $q = 1$ or $q = +\infty$.

3.4 Attacking a group of inputs

It can also be interesting to attack only a selected subset of inputs. It may help in identifying the more sensitive inputs of the network. Also, for some inputs like unsorted categorical ones, attacks are often meaningless since they introduce a main change in the informative contents of the dataset, which can be easily detected. Our proposed approach can be adapted to generate such partial attacks. In Problem (4), it is indeed sufficient to replace matrix $\Sigma^{1/2}$ by $D\Sigma^{1/2}D$, where D a masking diagonal matrix whose diagonal elements are equal to 1 when the input is attacked and 0 otherwise. The optimal solutions $\widehat{e}'$ and $\widehat{e} = \delta D\Sigma^{1/2}D\widehat{e}'$ have then their components equal to 0 for the non-attacked inputs. Note that the naive approach which would consist in solving (4) and setting to zero the resulting perturbation components for non-attacked inputs would be suboptimal.

4 Numerical Results

4.1 Dataset and architecture description

Open Source Datasets

We run our experiments on three open source regression datasets. The *Combined Cycle Power Plant [Tüfekci, 2014] dataset* has 4 features with 9,568 instances. The task is to predict the net hourly electrical energy output using hourly average ambient variables. The *Red Wine Quality* dataset [Cortez *et al.*, 2009] contains 1,599 total samples and each instance has 11 features. The features are physicochemical and sensory measurements for wine. The output variable is a quality score ranging from 0 to 10, where 10 represents for best quality and 0 for least quality. For the *Abalone* dataset, the task is to model an Abalone's age based purely on its physical measurements. This would allow Abalone's age estimation without cutting its shell. There are in total 4,177 instances with 8 input variables including one categorical variable. The datasets are divided with a ratio of 4:1 between training and testing data. The categorical attributes are dealt with by using one hot encoding based on the number of categories. The input attributes are normalised by removing their mean and scaling to unit variance.

We train fully connected networks for the estimation of variables from the datasets. The network architecture for the dataset are given below. The values represent the number of hidden neurons in the layers. Activation function at each layer is ReLU except for the last layer.

- Combined cycle Power Plant dataset - $(10, 6, 1)$
- Red Wine Quality dataset - $(100, 100, 100, 10, 1)$
- Abalone Data set - $(256, 256, 256, 256, 1)$

Industrial Dataset – safety critical Application

An industrial application dataset is also considered with 2,219,097 training, 739,639 validation, and 739,891 test samples. The description of the input/output variables of the

Mean Accuracy Error	MAE	=	$\frac{1}{K}\sum_{k=1}^{K}\|T(x_k+e_k)-y_k\|_q$
Fooling Error	E	=	$\frac{1}{K}\sum_{k=1}^{K}\|T(x_k+e_k)-T(x_k)\|_q$
Symmetric Mean Accuracy Percentage Error	SMAPE	=	$\frac{2}{K_+}\sum_{k=1}^{K_+}\frac{\|T(x_k+e_k)-y_k\|_q-\|T(x_k)-y_k\|_q}{\|T(x_k+e_k)-y_k\|_q+\|T(x_k)-y_k\|_q}$

Table 2: Error metrics used for evaluation. The mean value computed for SMAPE is limited to the K_+ positive values of the elements in the summation. e_k is the error generated by the adversarial on the k-th sample in the dataset of length K.

Noise	MAE_{std}	MAE_{gauss}	MAE_{uni}	MAE_{bin}	MAE_{adv}	E_{gauss}	E_{uni}	E_{bin}	E_{adv}	$SMAPE_{gauss}$	$SMAPE_{uni}$	$SMAPE_{bin}$	$SMAPE_{adv}$
Combined Cycle Power Plant Dataset													
$\mathbf{1\times10^{-1}}$	6.4×10^{-3}	6.5×10^{-3}	6.5×10^{-3}	6.5×10^{-3}	$\mathbf{10.3\times10^{-3}}$	1.3×10^{-3}	1.3×10^{-3}	1.4×10^{-3}	$\mathbf{4.0\times10^{-3}}$	0.33	0.34	0.36	**0.62**
$\mathbf{2\times10^{-1}}$	6.4×10^{-3}	6.8×10^{-3}	6.8×10^{-3}	6.9×10^{-3}	$\mathbf{14.2\times10^{-3}}$	2.5×10^{-3}	2.5×10^{-3}	2.7×10^{-3}	$\mathbf{8.0\times10^{-3}}$	0.50	0.52	0.56	**0.87**
Red Wine Quality Dataset													
$\mathbf{1\times10^{-1}}$	0.47	0.46	0.47	0.47	**0.58**	0.04	0.05	0.043	**0.12**	0.34	0.33	0.29	**0.41**
$\mathbf{2\times10^{-1}}$	0.47	0.47	0.48	0.48	**0.66**	0.09	0.09	0.09	**0.21**	0.49	0.51	0.48	**0.56**
Abalone age dataset													
$\mathbf{5\times10^{-2}}$	1.68	1.68	1.68	1.68	**2.04**	0.02	0.02	0.03	**0.36**	0.05	0.04	0.05	**0.38**
$\mathbf{1\times10^{-1}}$	1.68	1.68	1.68	1.68	**2.40**	0.05	0.05	0.05	**0.72**	0.09	0.08	0.09	**0.58**
Industrial Dataset													
$\mathbf{1\times10^{-1}}$	9.2×10^{-3}	9.6×10^{-3}	9.6×10^{-3}	9.6×10^{-3}	$\mathbf{20.9\times10^{-3}}$	2.6×10^{-3}	2.6×10^{-3}	2.7×10^{-3}	$\mathbf{11.8\times10^{-3}}$	0.45	0.46	0.47	**0.96**
$\mathbf{2\times10^{-1}}$	9.2×10^{-3}	$.10.7\times10^{-3}$	10.7×10^{-3}	10.7×10^{-3}	$\mathbf{32.5\times10^{-3}}$	5.1×10^{-3}	5.2×10^{-3}	5.4×10^{-3}	$\mathbf{24.0\times10^{-3}}$	0.65	0.66	0.67	**1.24**

Table 3: Comparison on evaluation metrics random attacker vs. proposed adversarial attacker with variation in perturbation level (ℓ_2 attack).

Noise	E_{adv}	E_{spec}	$SMAPE_{adv}$	$SMAPE_{spec}$
Combined Cycle Power Plant Dataset				
$\mathbf{1\times10^{-1}}$	$\mathbf{4.0\times10^{-3}}$	3.9×10^{-3}	**0.62**	0.60
$\mathbf{2\times10^{-1}}$	$\mathbf{8.0\times10^{-3}}$	7.7×10^{-3}	**0.87**	0.84
Red Wine Quality Dataset				
$\mathbf{1\times10^{-1}}$	**0.12**	0.017	**0.41**	0.12
$\mathbf{2\times10^{-1}}$	**0.21**	0.03	**0.56**	0.17
Abalone age dataset				
$\mathbf{5\times10^{-2}}$	**0.36**	0.11	**0.38**	0.12
$\mathbf{1\times10^{-1}}$	**0.72**	0.23	**0.58**	0.21
Industrial Dataset				
$\mathbf{1\times10^{-1}}$	$\mathbf{11.8\times10^{-3}}$	9.1×10^{-3}	**0.91**	0.49
$\mathbf{2\times10^{-1}}$	$\mathbf{24.0\times10^{-3}}$	17.5×10^{-3}	**1.24**	0.72

Table 4: Results of proposed adversarial techniques. Standard training vs Spectral Normalisation training on ℓ_2 attacks.

Noise	E_{adv}	E_{inp}	$SMAPE_{adv}$	$SMAPE_{inp}$
Combined Cycle Power Plant Dataset				
$\mathbf{1\times10^{-1}}$	$\mathbf{4.0\times10^{-3}}$	3.4×10^{-3}	**0.62**	0.58
$\mathbf{2\times10^{-1}}$	$\mathbf{8.0\times10^{-3}}$	7.0×10^{-3}	**0.87**	0.82
Red Wine Quality Dataset				
$\mathbf{1\times10^{-1}}$	0.12	**0.13**	0.41	**0.44**
$\mathbf{2\times10^{-1}}$	0.21	**0.22**	0.56	**0.60**
Abalone age dataset				
$\mathbf{5\times10^{-2}}$	0.36	0.36	0.38	0.38
$\mathbf{1\times10^{-1}}$	**0.72**	0.71	0.58	**0.59**
Industrial Dataset				
$\mathbf{1\times10^{-1}}$	12.0×10^{-3}	12.0×10^{-3}	0.91	**0.96**
$\mathbf{2\times10^{-1}}$	24.0×10^{-3}	24.0×10^{-3}	1.24	1.24

Table 5: Results of proposed adversarial techniques. Standard training attacking all inputs vs standard training attacking few inputs on ℓ_2 attacks.

dataset is given in Table 1. The variable to be predicted is the Estimation of Arrival time (ETE) of a flight, given variables including the distance and speed, and also an initial estimate of ETE. The dataset is related to flight control, an activity area where safety is critical. The input attributes are normalized by removing their mean and scaling to unit variance. For models, we build fully connected networks with ReLU activation function on all the hidden layers except the last one. The network architecture is shown in the Figure 1.

4.2 Experimental setup

We first train our networks without any constraints using the network architecture presented in the previous section with the aim of reducing the prediction/performance loss on the train dataset. This will be refered to as a standard training procedure.

To understand and analyze the performance of the proposed adversarial attacker, we calculate the three error metrics described in Table 2. We compare the proposed adversarial attacker with random noise attackers generated by i.i.d. perturbations. We use three additive noise distributions—Gaussian, uniform and binary, for comparisons. The output of these attackers have been normalized so as to meet the desired bound on the norm of the perturbation. The metrics are computed on the test samples where K is the total number of samples in the test set. The results on the 4 datasets for varying noise levels are shown in Table 3. We also show the histograms of $(\|T(x_k+e_k)-y_k\|_q-\|T(x_k)-y_k\|_q)_{1\leq k\leq K}$ in Figures 2, 3, 4, and 5, where $(e_k)_{1\leq k\leq K}$ have been generated from various noise distributions and the proposed adversarial attacker.

For safety critical tasks, Lipschitz and performance targets can be specified as engineering requirements, prior to network training. Such a design approach has proven to make the network more stable and robust to adversarial attacks. Imposing a Lipschitz target can be done either by controlling the Lipschitz constant for each layer or for the whole network depending on the application at hand. One such method for

Noise	MAE_{std}	MAE_{gauss}	MAE_{uni}	MAE_{bin}	MAE_{adv}	E_{gauss}	E_{uni}	E_{bin}	E_{adv}	$SMAPE_{gauss}$	$SMAPE_{uni}$	$SMAPE_{bin}$	$SMAPE_{adv}$
ℓ_2 attacks													
$\mathbf{1 \times 10^{-1}}$	9.2×10^{-3}	9.6×10^{-3}	9.6×10^{-3}	9.6×10^{-3}	$\mathbf{20.9 \times 10^{-3}}$	2.6×10^{-3}	2.6×10^{-3}	2.7×10^{-3}	$\mathbf{11.8 \times 10^{-3}}$	0.45	0.46	0.47	**0.96**
$\mathbf{2 \times 10^{-1}}$	9.2×10^{-3}	10.7×10^{-3}	10.7×10^{-3}	10.7×10^{-3}	$\mathbf{32.5 \times 10^{-3}}$	5.1×10^{-3}	5.2×10^{-3}	5.4×10^{-3}	$\mathbf{24.0 \times 10^{-3}}$	0.65	0.66	0.67	**1.24**
ℓ_1 attacks													
$\mathbf{1 \times 10^{-1}}$	9.2×10^{-3}	9.2×10^{-3}	9.2×10^{-3}	9.2×10^{-3}	$\mathbf{18.5 \times 10^{-3}}$	8.4×10^{-4}	8.1×10^{-4}	7.2×10^{-4}	$\mathbf{9.5 \times 10^{-3}}$	0.22	0.21	0.20	**0.87**
$\mathbf{2 \times 10^{-1}}$	9.2×10^{-3}	9.4×10^{-3}	9.3×10^{-3}	9.3×10^{-3}	$\mathbf{28.1 \times 10^{-3}}$	1.7×10^{-3}	1.6×10^{-3}	1.4×10^{-3}	$\mathbf{20.0 \times 10^{-3}}$	0.35	0.34	0.32	**1.15**
ℓ_∞ attacks													
$\mathbf{1 \times 10^{-1}}$	9.2×10^{-3}	10.5×10^{-3}	11.1×10^{-3}	13.0×10^{-3}	$\mathbf{31.1 \times 10^{-3}}$	4.7×10^{-3}	6.0×10^{-3}	9.9×10^{-3}	$\mathbf{22.0 \times 10^{-3}}$	0.63	0.71	0.87	**1.22**
$\mathbf{2 \times 10^{-1}}$	9.2×10^{-3}	13.5×10^{-3}	15.3×10^{-3}	22.0×10^{-3}	$\mathbf{52.5 \times 10^{-3}}$	9.4×10^{-3}	12.0×10^{-3}	19.0×10^{-3}	$\mathbf{45.0 \times 10^{-3}}$	0.84	0.92	1.08	**1.47**

Table 6: Comparison on industrial dataset for ℓ_2, ℓ_1 and ℓ_∞ attacks with variation in perturbation levels.

controlling the Lipschitz has been presented in [Serrurier *et al.*, 2020] using Hinge regularization. In the experiments, we train our networks while using a spectral normalisation technique [Miyato *et al.*, 2018] which has been proven to be very effective in controlling Lipschitz properties in GANs.

Given an m layer fully connected architecture and a Lipschitz target L, we can constrain the spectral norm of each layer to be less than $\sqrt[m]{L}$. This ensures that the upper bound on the global Lipschitz constant is less than L. We keep the network architectures exactly the same for both training procedures. The performance of adversarial attacker on standard and spectrally normalized trained model in terms of Fooling Error (E) and Symmetric Mean Accuracy Percentage error (SMAPE) for various datasets and varying perturbation magnitude is given in Table 4.

All the previous results have been obtained with attack and noise addition on all the input features present in the datasets. As pointed in Section 3.4, the introduced adversarial attacker is capable of attacking a group of inputs. While generating an adversarial attack we avoid attacking the categorical input variables [Ballet *et al.*, 2019], hence in Abalone and industrial datasets we attack only the continuous variables. For the Combined Power plant dataset, we attack 3 out of 4 continuous variables since it does not contain any categorical variables. Similarly, for the Red-wine dataset we attack 8 continuous variables out of 11. The performance of the adversarial attacker, when attacking only few inputs, is shown in Table 5.

As emphasized in Section 3.3, our adversarial attacker is applicable for various measures of perturbation on input and output deviations. The previous results have been obtained for the value $p = q = 2$ termed as ℓ_2 attacks here. We further show results for $p = q = 1$ termed as ℓ_1 attacks and for $p = q = +\infty$ termed as ℓ_∞ attacks in Table 6.

4.3 Result analysis

Some general conclusions can be drawn from the experiments.

- We observe that the proposed adversarial attacker performs better than all the three random noise attackers for the three quantitative measures we have defined. In addition, the histograms in Figures 2, 3, 4, and 5 show that the error may be increased or reduced by random attackers, while this shortcoming does not happen with our adversarial attacker. This observation is verified on the norms - ℓ_2 , ℓ_1 and ℓ_∞ norms in Table 6.
- Spectral normalisation has been proven to robustify the trained models. As in Table 4, we see that the Fooling error (E) and SMAPE are reduced in all the cases when compared to the standard trained model.
- In the considered examples, we observe that categorical data have little effect when attacking the trained model as shown in Table 5. The E and SMAPE measures do not show major differences.

5 Conclusion

In this article, we have introduced a novel easily implementable Jacobian-based adversarial attacker for estimation problems. These regression tasks cover a major portion of safety critical applications. Yet there is lack of works studying and analysing adversarial attacks in this context, as opposed to classification tasks. The present study contributes to filling this gap. We have presented error metrics which help in analysing the effectiveness of the attacker. Our attacker is versatile in the sense that it can handle any measure (ℓ_1, ℓ_2, ℓ_∞) on input or output perturbations according to the target application. Our attacker is also successful in handling attacks focused on subsets of inputs. This feature may be useful when handling specific tabular datasets and may also be insightful when information is available related to the sensitivity or ability to control some inputs. Our tests concentrated on fully connected networks, but it is worth pointing out that the proposed approach can be applied to any network architecture.

References

[Balda *et al.*, 2018] Emilio Rafael Balda, Arash Behboodi, and Rudolf Mathar. Perturbation analysis of learning algorithms: A unifying perspective on generation of adversarial examples. *arXiv preprint arXiv:1812.07385*, 2018.

[Ballet *et al.*, 2019] Vincent Ballet, Xavier Renard, Jonathan Aigrain, Thibault Laugel, Pascal Frossard, and Marcin Detyniecki. Imperceptible adversarial attacks on tabular data. *arXiv preprint arXiv:1911.03274*, 2019.

[Bolte and Pauwels, 2020] Jérôme Bolte and Edouard Pauwels. Conservative set valued fields, automatic differentiation, stochastic gradient methods and deep learning. *Mathematical Programming*, pages 1–33, 2020.

[Carlini and Wagner, 2017] Nicholas Carlini and David Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. IEEE, 2017.

[Cortez *et al.*, 2009] Paulo Cortez, António Cerdeira, Fernando Almeida, Telmo Matos, and José Reis. Modeling wine preferences by data mining from physicochemical properties. *Decision support systems*, 47(4):547–553, 2009.

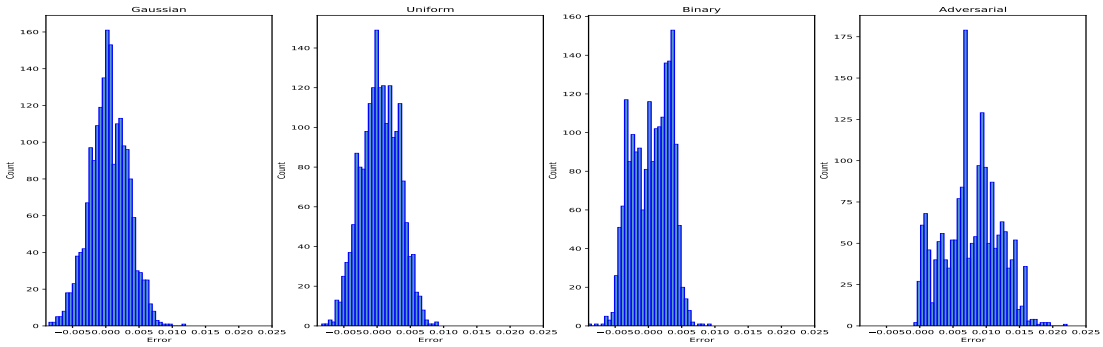

Figure 2: Error distribution of random attacks and proposed adversarial attack on Combined cycle Power Plant dataset for perturbation level of 2×10^{-1} for ℓ_2 attack.

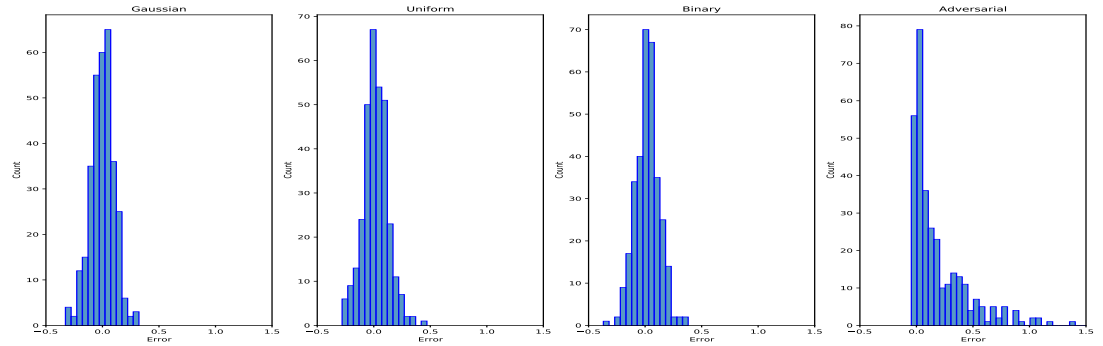

Figure 3: Error distribution of random attacks and proposed adversarial attack on Red-wine dataset for perturbation level of 2×10^{-1} for ℓ_2 attack.

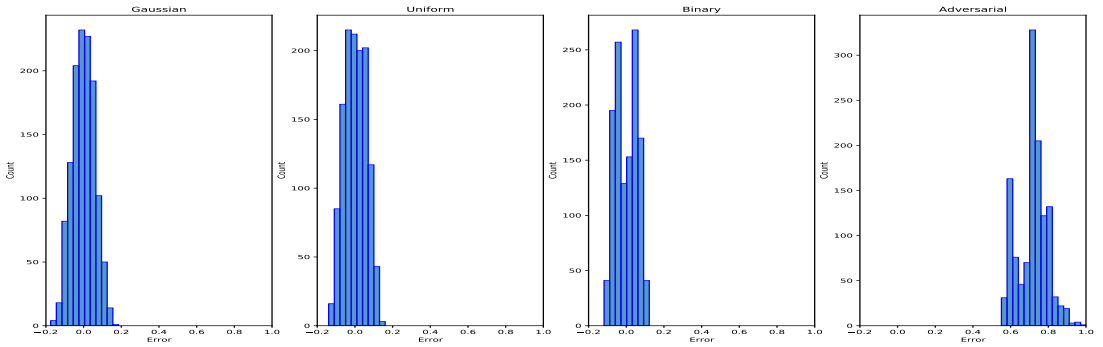

Figure 4: Error distribution of random attacks and proposed adversarial attack on Abalone dataset for perturbation level of 1×10^{-1} for ℓ_2 attack.

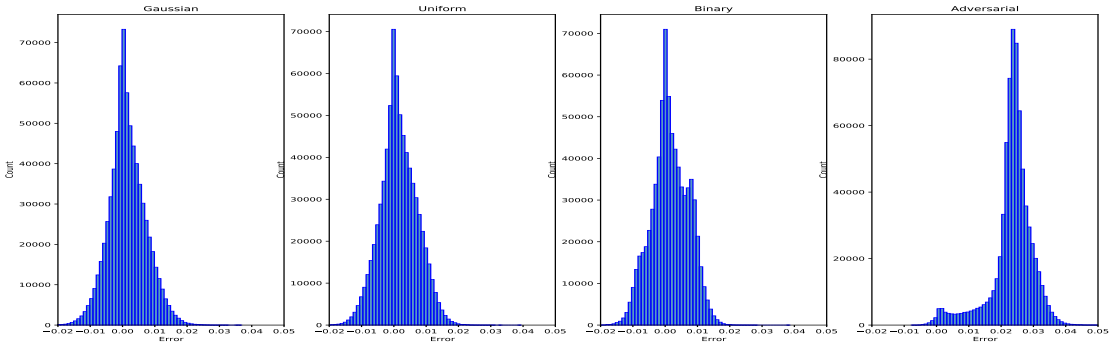

Figure 5: Error distribution of random attacks and proposed adversarial attack on industrial dataset for perturbation level of 2×10^{-1} for ℓ_2 attack.

[Deng *et al.*, 2020] Yao Deng, Xi Zheng, Tianyi Zhang, Chen Chen, Guannan Lou, and Miryung Kim. An analysis of adversarial attacks and defenses on autonomous driving models. In *2020 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 1–10. IEEE, 2020.

[Dong *et al.*, 2018] Yinpeng Dong, Fangzhou Liao, Tianyu Pang, Hang Su, Jun Zhu, Xiaolin Hu, and Jianguo Li. Boosting adversarial attacks with momentum. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 9185–9193, 2018.

[Eykholt *et al.*, 2018] 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. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1625–1634, 2018.

[Finlayson *et al.*, 2018] Samuel G Finlayson, Hyung Won Chung, Isaac S Kohane, and Andrew L Beam. Adversarial attacks against medical deep learning systems. *arXiv preprint arXiv:1804.05296*, 2018.

[Ghafouri *et al.*, 2018] Amin Ghafouri, Yevgeniy Vorobeychik, and Xenofon Koutsoukos. Adversarial regression for detecting attacks in cyber-physical systems. *arXiv preprint arXiv:1804.11022*, 2018.

[Goodfellow *et al.*, 2014] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[Huang and Wang, 2018] Yonghong Huang and Shih-han Wang. Adversarial manipulation of reinforcement learning policies in autonomous agents. In *2018 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2018.

[Kurakin *et al.*, 2016] Alexey Kurakin, Ian Goodfellow, Samy Bengio, et al. Adversarial examples in the physical world, 2016.

[Liu *et al.*, 2016] Yanpei Liu, Xinyun Chen, Chang Liu, and Dawn Song. Delving into transferable adversarial examples and black-box attacks. *arXiv preprint arXiv:1611.02770*, 2016.

[Miyato *et al.*, 2018] Takeru Miyato, Toshiki Kataoka, Masanori Koyama, and Yuichi Yoshida. Spectral normalization for generative adversarial networks. *arXiv preprint arXiv:1802.05957*, 2018.

[Moosavi-Dezfooli *et al.*, 2016] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, and Pascal Frossard. Deepfool: a simple and accurate method to fool deep neural networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2574–2582, 2016.

[Moosavi-Dezfooli *et al.*, 2017] Seyed-Mohsen Moosavi-Dezfooli, Alhussein Fawzi, Omar Fawzi, and Pascal Frossard. Universal adversarial perturbations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1765–1773, 2017.

[Nguyen and Raff, 2018] Andre T Nguyen and Edward Raff. Adversarial attacks, regression, and numerical stability regularization. *arXiv preprint arXiv:1812.02885*, 2018.

[Papernot *et al.*, 2016] Nicolas Papernot, Patrick McDaniel, Somesh Jha, Matt Fredrikson, Z Berkay Celik, and Ananthram Swami. The limitations of deep learning in adversarial settings. In *2016 IEEE European symposium on security and privacy (EuroS&P)*, pages 372–387. IEEE, 2016.

[Ren *et al.*, 2019] Kui Ren, Qian Wang, Cong Wang, Zhan Qin, and Xiaodong Lin. The security of autonomous driving: Threats, defenses, and future directions. *Proceedings of the IEEE*, 108(2):357–372, 2019.

[Rony *et al.*, 2020] Jérôme Rony, Eric Granger, Marco Pedersoli, and Ismail Ben Ayed. Augmented lagrangian adversarial attacks. *arXiv preprint arXiv:2011.11857*, 2020.

[Serrurier *et al.*, 2020] Mathieu Serrurier, Franck Mamalet, Alberto González-Sanz, Thibaut Boissin, Jean-Michel Loubes, and Eustasio del Barrio. Achieving robustness in classification using optimal transport with hinge regularization, 2020.

[Su *et al.*, 2019] Jiawei Su, Danilo Vasconcellos Vargas, and Kouichi Sakurai. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

[Szegedy *et al.*, 2013] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

[Tong *et al.*, 2018] Liang Tong, Sixie Yu, Scott Alfeld, et al. Adversarial regression with multiple learners. In *International Conference on Machine Learning*, pages 4946–4954. PMLR, 2018.

[Tüfekci, 2014] Pınar Tüfekci. Prediction of full load electrical power output of a base load operated combined cycle power plant using machine learning methods. *International Journal of Electrical Power and Energy Systems*, 60:126 – 140, 2014.