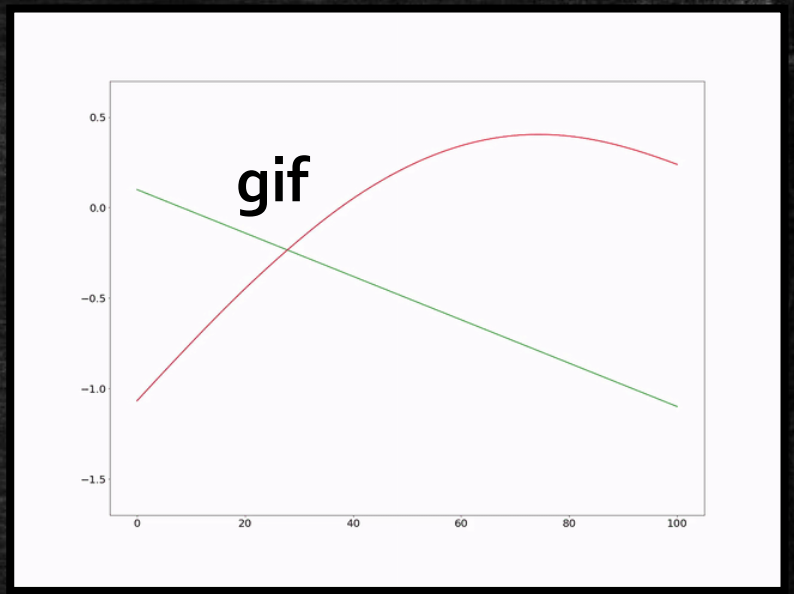


Adversarial attacks on Explainable AI methods

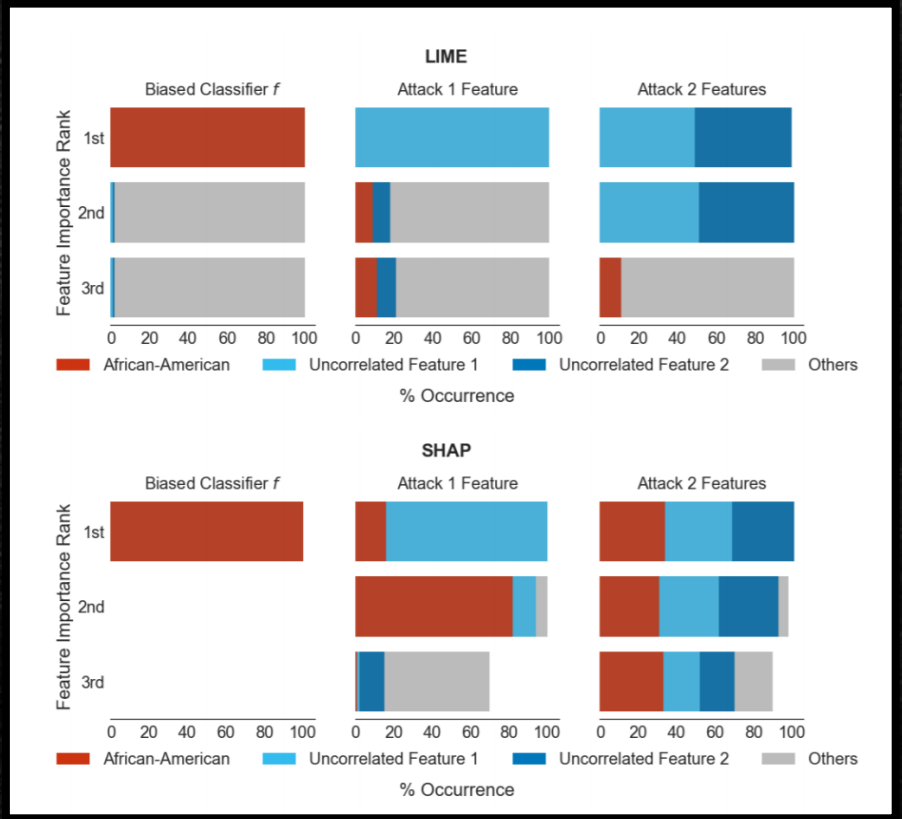
Hubert Baniecki, Wojciech Kretowicz

Plan

- Zarys dziedziny
- Paper 1: ataki na wyjaśnienia predykcji w ML
- Paper 2: ataki na wyjaśnienia predykcji w CV
- Praca inżynierska: ataki na wyjaśnienia modelu w ML

O co chodzi?

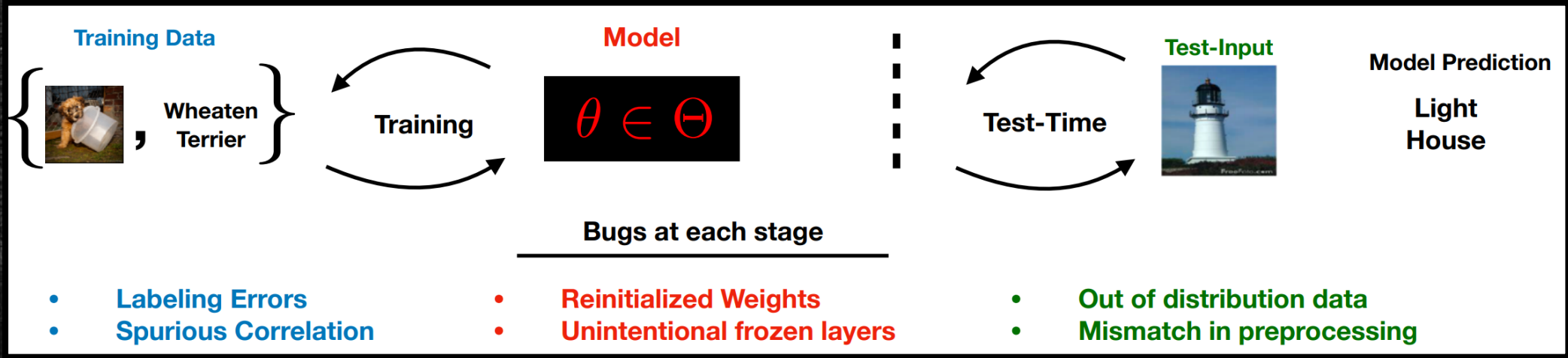
"Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods"
D. Slack et al. *AIES(AAAI)* 2020

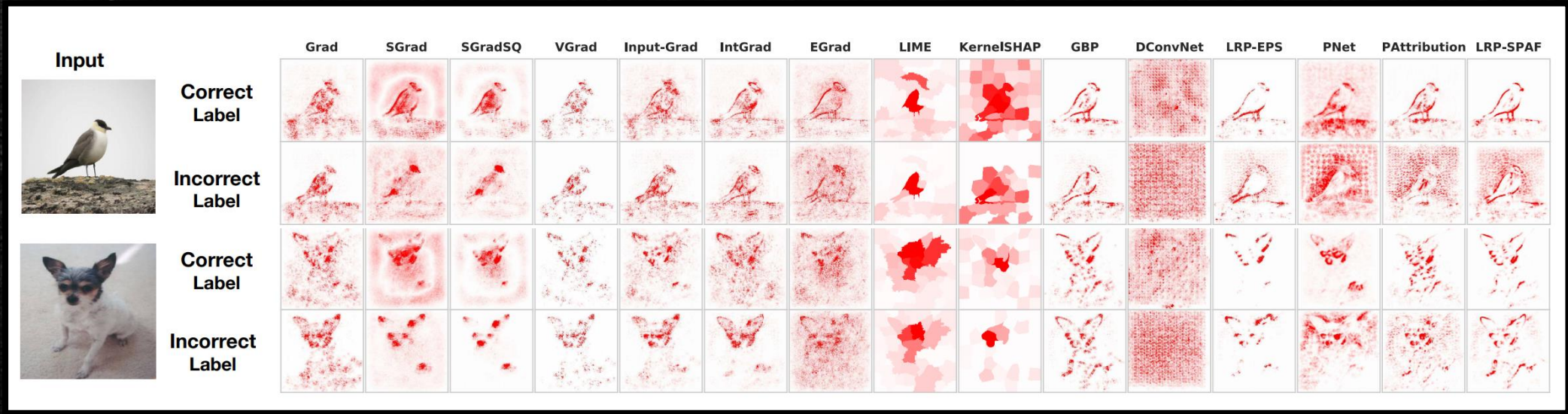


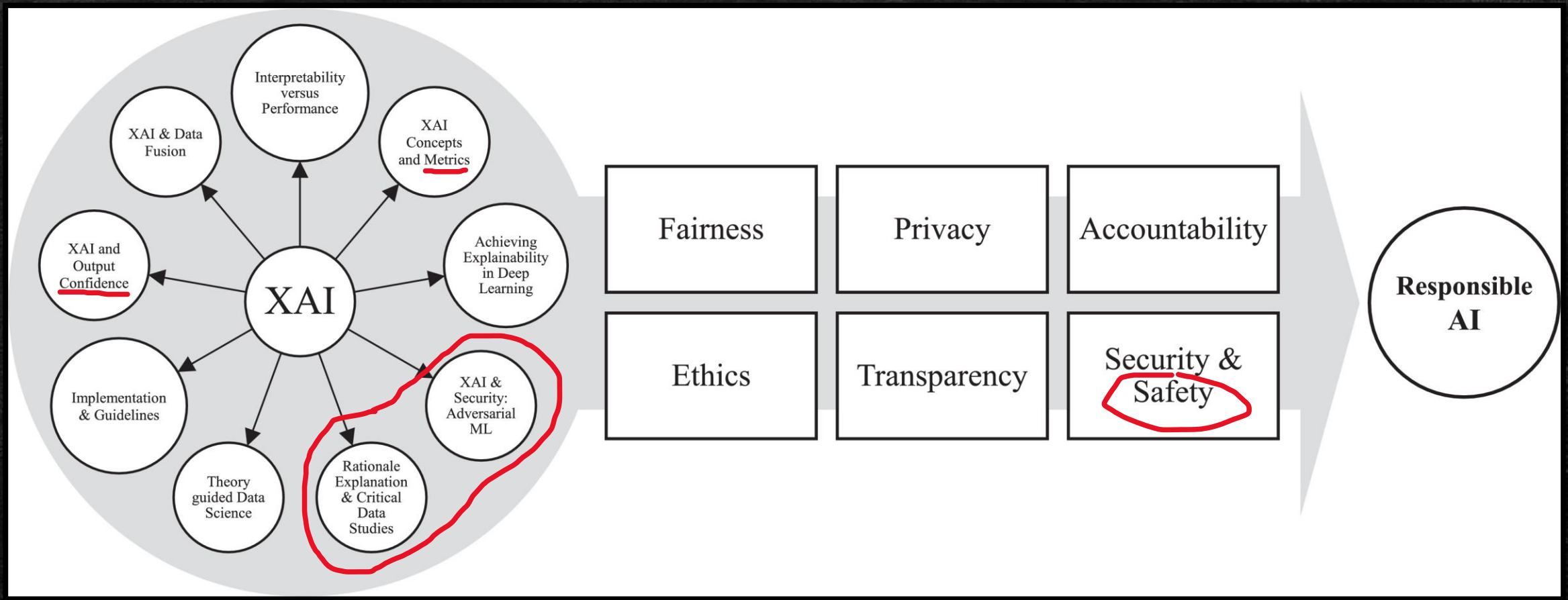
"Explanations can be manipulated and geometry is to blame"
A. K. Dombrowski et al. *NeurIPS* 2019

Ewaluacja wyjaśnień

"Debugging Tests for Model Explanations"
J. Adebayo et al. *NeurIPS* 2020







"XAI: Concepts, taxonomies, opportunities and challenges toward responsible AI"
A. Barredo-Arrieta et al. *Information Fusion* 2019

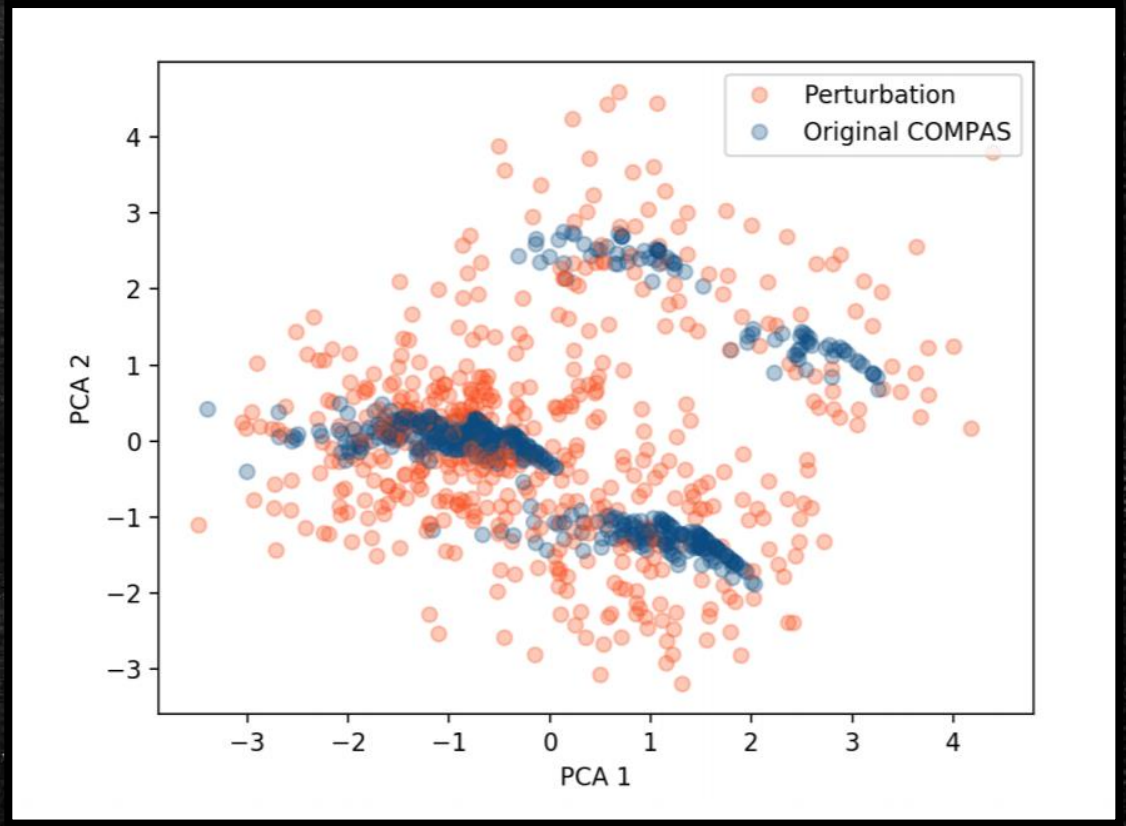
Fooling LIME and SHAP

LIME i Kernel SHAP

$$L(f, g, \pi_x) = \sum_{x' \in X'} [f(x') - g(x')]^2 \pi_x(x')$$

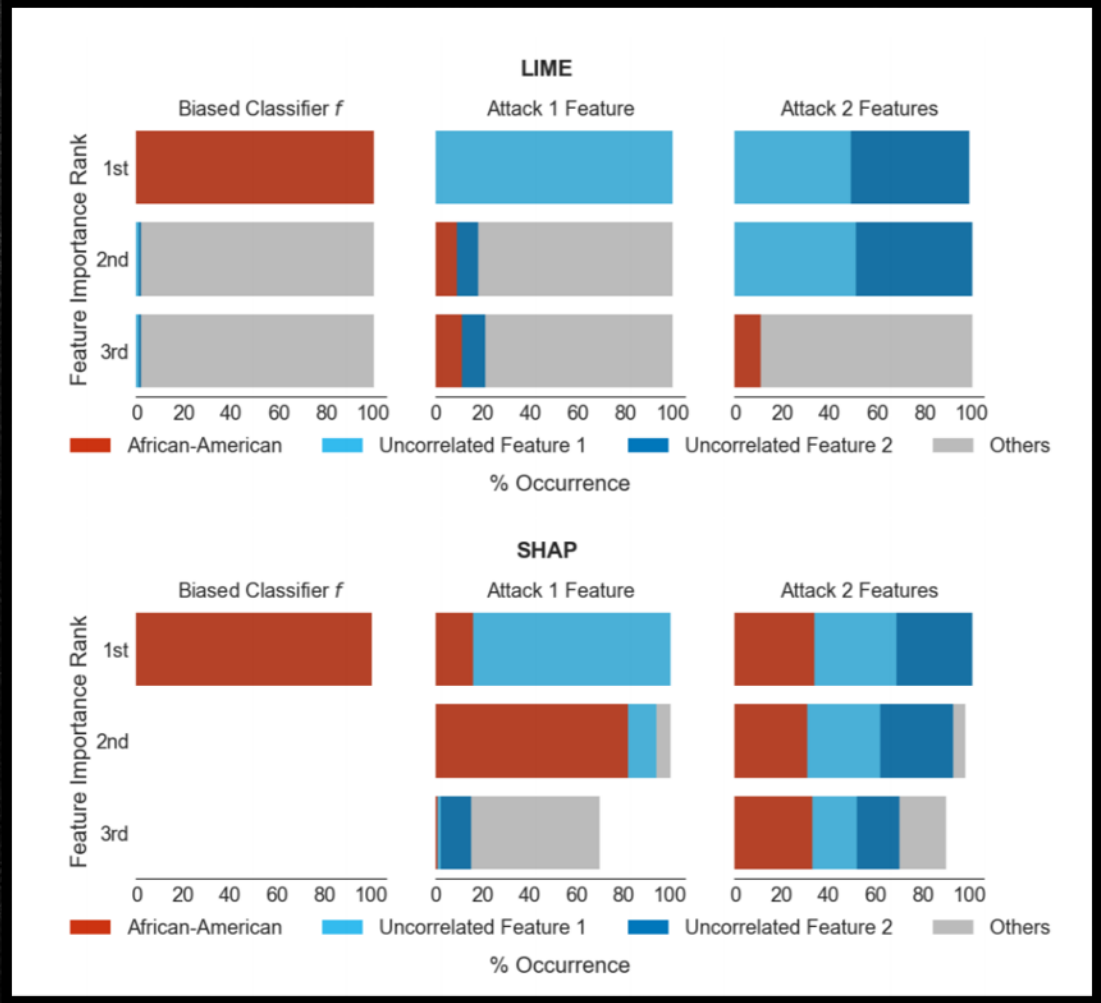
Adversarial model

$$e(x) = \begin{cases} f(x), & \text{if } x \in \mathcal{X}_{dist} \\ \psi(x), & \text{otherwise} \end{cases}$$



"Fooling LIME and SHAP: Adversarial Attacks on Post hoc Explanation Methods"
D. Slack et al. *AIES(AAAI)* 2020

Fooling LIME and SHAP



Ataki na wyjaśnienia predykcji w CV

Original Image

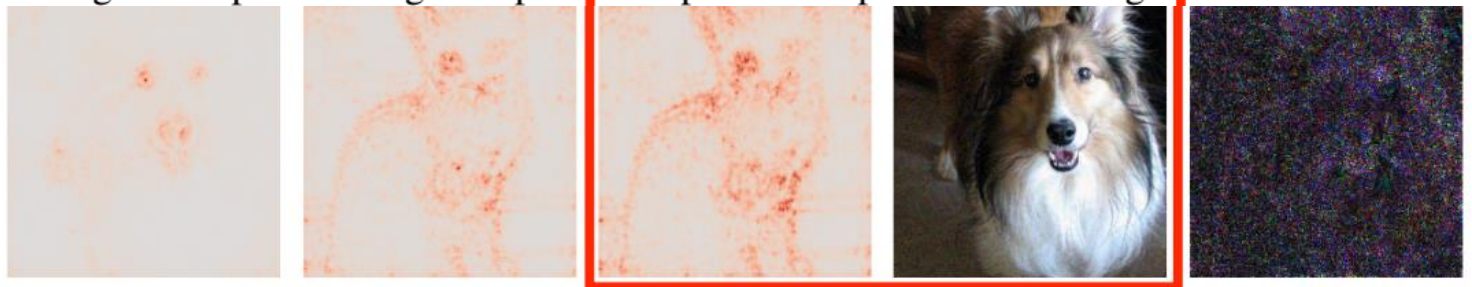
Image used to produce Target

For a given explanation method and specified target $h^t \in \mathbb{R}^d$, a manipulated image $x_{\text{adv}} = x + \delta x$ has the following properties:

1. The output of the network stays approximately constant, i.e. $g(x_{\text{adv}}) \approx g(x)$.
2. The explanation is close to the target map, i.e. $h(x_{\text{adv}}) \approx h^t$.
3. The norm of the perturbation δx added to the input image is small, i.e. $\|\delta x\| = \|x_{\text{adv}} - x\| \ll 1$ and therefore not perceptible.

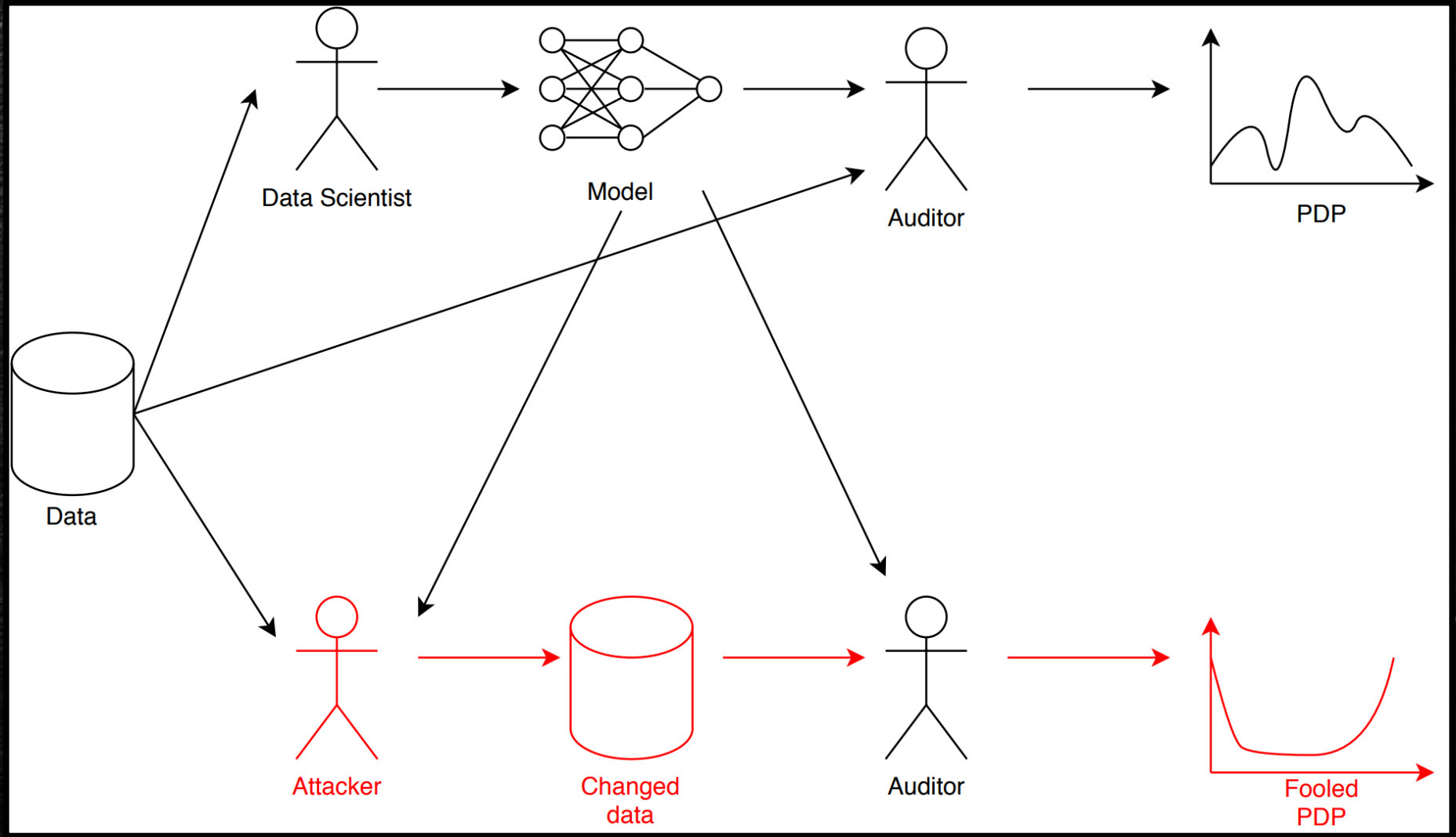
We obtain such manipulations by optimizing the loss function

$$\mathcal{L} = \left\|h(x_{\text{adv}}) - h^t\right\|^2 + \gamma \left\|g(x_{\text{adv}}) - g(x)\right\|^2 , \tag{4}$$



"Explanations can be manipulated and geometry is to blame"
A. K. Dombrowski et al. *NeurIPS* 2019

Atak


Data Scientist
Model
Auditor
PDP
Data
Attacker
Changed data
Auditor
Fooled PDP

global-level, model-agnostic, post-hoc

Partial Dependence Plots (PDP)

$$PDP_c(z) = E_{X_{-c}}\left[f(X^{c|=z}\right]$$

$$PDP_c(z) = \frac{1}{n}\sum_{i=1}^{n} f\left(x_i^{c|=z}\right)$$

Accumulated Local Effects (ALE)

$$ALE_c(z) = \int_{z_0}^{z}\left(E_{X_{-c}}\left[\left\{\frac{\partial f(u)}{\partial u}\right\}_{u=X^{c|=z}}\right]\right)dv + c$$

$$ALE_c(z) = \sum_{k=1}^{K}\left[\frac{1}{\sum_l w_l(z_k)}\sum_{i=1}^{N} w_i(z_k)\left\{f\left(X_i^{c|=z_k}\right) - f\left(X_i^{c|=z_k-\Delta}\right)\right\}\right] - \hat{c}$$

"Greedy Function Approximation: A Gradient Boosting Machine"
Friedman, Jerome H. *Annals of Statistics* 2000.

"Visualising the effects of predictor variables in black box supervised learning models."
Apley, Daniel W. and Jingyu Zhu*Journal of the Royal Statistical Society Series B* 2020.

Metoda

- **Cel**: wizualna zmiana wyniku wyjaśnienia modelu
- **Poprzez**: zmianę zbioru danych na którym liczone są wyjaśnienia
- **Algorytmem**: genetycznym lub gradientowym
- **Strategia**: atak celowany lub '*robustness/sanity check*'
- **Wynik**:
  - sprawdzenie czy wyjaśnienie jest stabilne
  - przekłamanie interpretacji działania modelu:
    - wykazanie fałszywej monotoniczności
    - ukrycie dyskryminacji ze względu na zmienną
    - zwiększenie istotności zmiennej

Metoda

- Optymalizacja funkcji kosztu
- Pierwszy człon opisuje różnicę między wyjaśnieniami
- Kolejne regularyzują:
  - różnicę między zbiorami danych
  - różnicę między predykcjami
- Wyjaśniana/atakowana zmienna pozostaje stała
- Można wybrać, które zmienne powinny pozostać stałe

$$L_1^g(X) = \frac{||g_c(X') - g_c(X)||^2}{|Z|},$$

$$L_2(X) = \frac{||X' - X||^2}{np},$$

$$L_3(X) = \frac{||f(X') - f(X)||^2}{n}.$$

$$L(X) = L_1(X) + \alpha L_2(X) + \beta L_3(X)$$

Algorytm genetyczny

- Model-agnostic
- Explanation-agnostic
- Stosunkowo wolny - narzut czasowy przy predykcji
- Dla 200 osobników, 500 obserwacji w danych, 31 punktów podziału atakowanej zmiennej: ponad 3mln wierszy

ALGORITHM 1: Genetic algorithm.

Data: model, data, targeted variable, **optionally:** target explanation, constant variables, α, β
Result: result explanation, result data
`/* The initialized population consists of original datasets; thus, the order of operations */`
while *iteration* < *max_iterations* **do**
    mutation phase (Algorithm 4)
    crossover phase (Algorithm 5)
    evaluation phase (Algorithm 6)
    selection phase (Algorithm 7) `/* omit selection in the last iteration */`
end

ALGORITHM 2: Mutation phase.

ALGORITHM 3: Crossover phase.

ALGORITHM 4: Evaluation phase.

ALGORITHM 5: Selection phase involving rank selection and basic elitarism.

Data: population
Result: population subset
sort the population by loss values
save top best individuals for the population subset
calculate the rank probabilities for the remaining individuals
sample population subset by these probabilities

Algorytm gradientowy

- Wymaga policzenia pochodnej wyniku po wejściowych danych:
  - dlatego idealnym zastosowaniem dla sieci neuronowych
- Zbiega o wiele szybciej
- Można użyć różnych optymalizatorów:
  - u nas: Adam

Theorem 1 (PDP term not centred derivative). *Let $f : X \longrightarrow Y$ represents the differentible function. Let Z be the set of points that are used to calculate PDP. Then*

$$\nabla_{X_{-c}} L_1^{PDP,notcentred}(X) = \frac{2}{n|Z|} \sum_{z \in Z} \nabla_{X_{-c}} f(X^{c|=z}) \cdot (PDP_c(X,z) - PDP^*(z))$$

Theorem 2 (PDP term cetered derivative). *Let $f : X \longrightarrow Y$ represents the differentible function. Let Z be the set of points that are used to calculate PDP. Then*

$$\nabla_{X_{-c}} L_1^{PDP,centred}(X) =$$

$$\frac{2}{n|Z|} \sum_{z \in Z} \left(\nabla_{X_{-c}} f(X^{c|=z}) - \frac{\sum_{z' \in Z} \nabla_{X_{-c}} f(X^{c|=z'})}{|Z|} \right) \cdot$$

$$\cdot \left(\left(PDP_c(X,z) - \frac{\sum_{z' \in Z} PDP_c(X,z')}{|Z|} \right) - \left(PDP^*(z) - \frac{\sum_{z' \in Z} PDP^*(z')}{|Z|} \right) \right)$$

Theorem 3 (ALE term derivative). *Let $f : X \longrightarrow Y$ represents the differentible function. Let Z be the set of points that are used to calculate ALE. Then*

$$\nabla_{X_{-c}} L_1^{ALE}(X) = \frac{2}{|Z|} \sum_{z \in Z} (ALE_c(X,z) - ALE^*(z)) \nabla_{X_{-c}} ALE_c(X,z)$$

Lemma 1. *Let $f : X \longrightarrow Y$ represents the differentible function. Let $Z = (z_1, ..., z_g)$ be the set of points that are used to calculate ALE. Then*

$$\frac{\partial ALE_c(z_K)}{\partial X_{i,j}} = \sum_{k=1}^{K} \left[\frac{1}{\sum_{l=1}^{n} w_l(z_k)} w_i(z_k) \left(\frac{\partial f}{\partial X_{i,j}} \left(X_i^{c|=z_k} \right) - \frac{\partial f}{\partial X_{i,j}} \left(X_i^{c|=z_{k-1}} \right) \right) \right]$$

Corollary 1.

$$\frac{\partial ALE_c(z_{K+1})}{\partial X_{i,j}} = \frac{\partial ALE_c(z_K)}{\partial X_{i,j}} + \frac{1}{\sum_{l=1}^{n} w_l(z_k)} w_i(z_{K+1}) \left(\frac{\partial f}{\partial X_{i,j}} \left(X_i^{c|=z_{K+1}} \right) - \frac{\partial f}{\partial X_{i,j}} \left(X_i^{c|=z_K} \right) \right)$$

Studium przypadku: heart

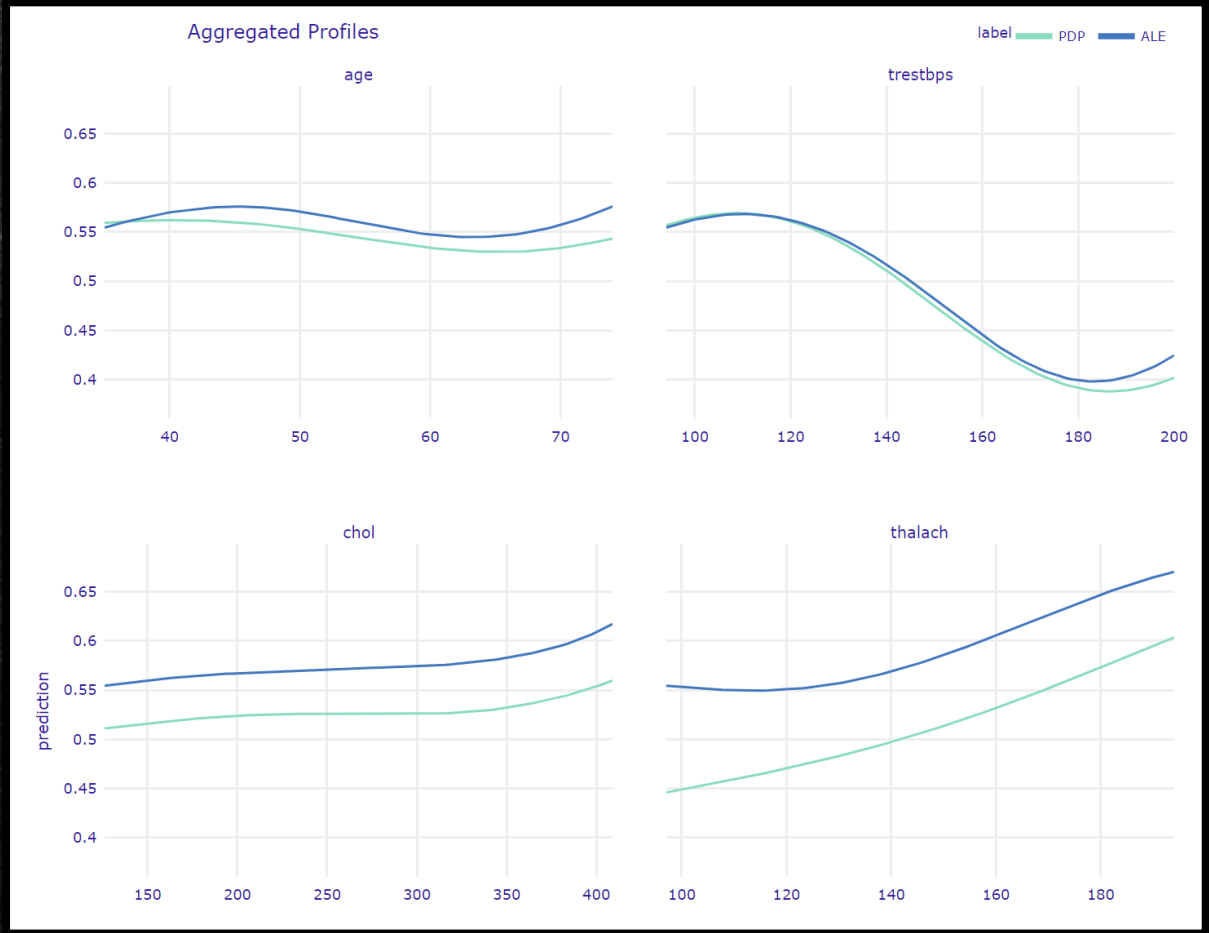

```
data = fool.datasets.load_heart1()
X, y = data.drop("target", axis=1), data.target
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)
```

```
model = Pipeline([('scaler', StandardScaler()), ('estimator', SVC(C=10, probability=True,
model.fit(X_train, y_train)
```

```
explainer = dalex.Explainer(model, X_test, y_test, label="svm-heart", verbose=False)
explainer.model_performance()
```

	recall	precision	f1	accuracy	auc
svm-heart	0.794118	0.870968	0.830769	0.819672	0.905229

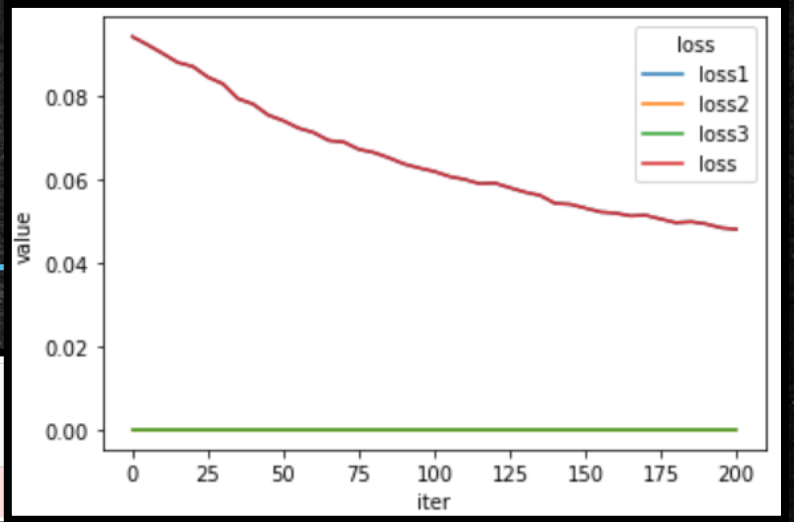
```
explainer.model_profile(label='PDP', verbose=False).plot(
    explainer.model_profile('ale', label='ALE', verbose=False),
    variables=num_variables
)
```

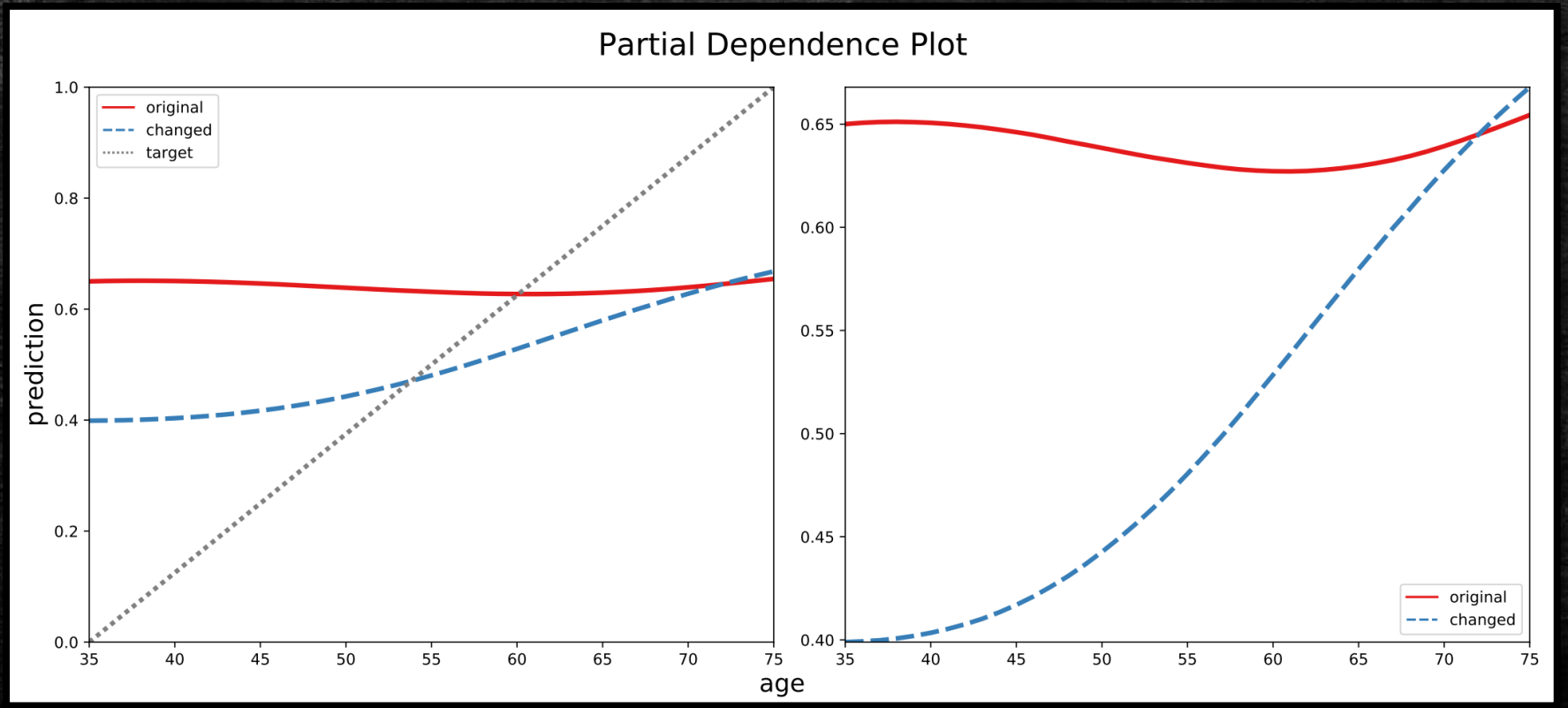


Atak

```
attack = fool.attacks.GeneticAttack(explainer, variable='age', explanation_type='pdp',
                                    constant=cat_variables, pop_count=200)
attack.fool_aim(target=lambda x: (x-35)/40, max_iter=200)
```

```
Iter: 200 || Fitness: 0.04812025287323034: 100%|██████████████████████████████| 200/200
```





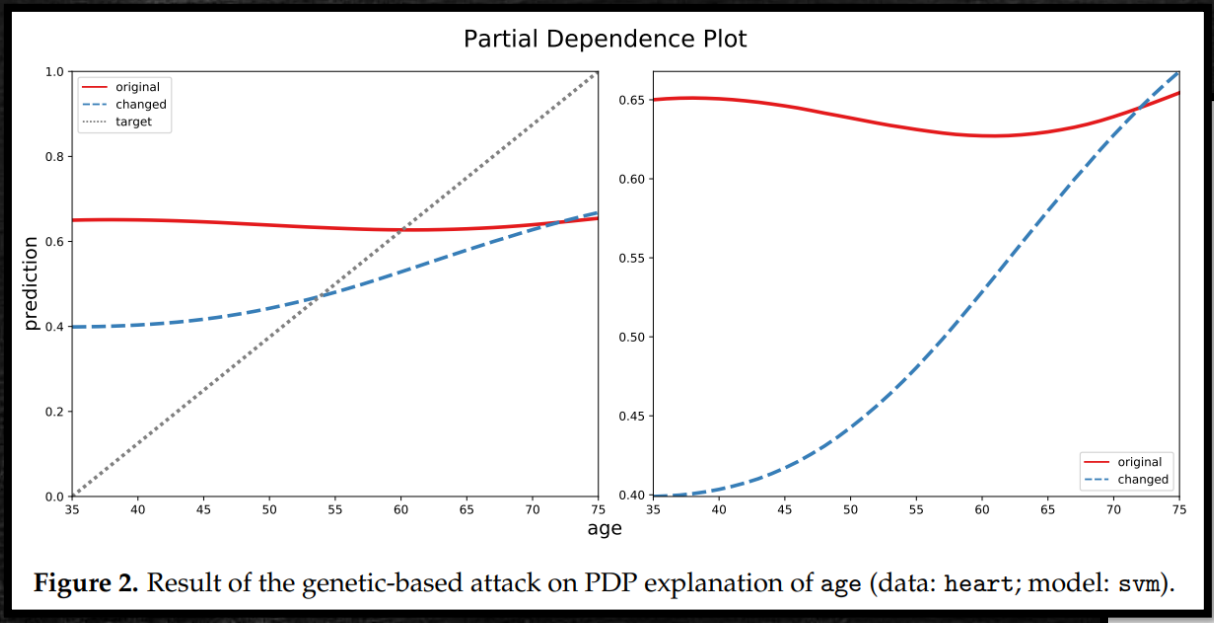


Figure 2. Result of the genetic-based attack on PDP explanation of `age` (data: `heart`; model: `svm`).

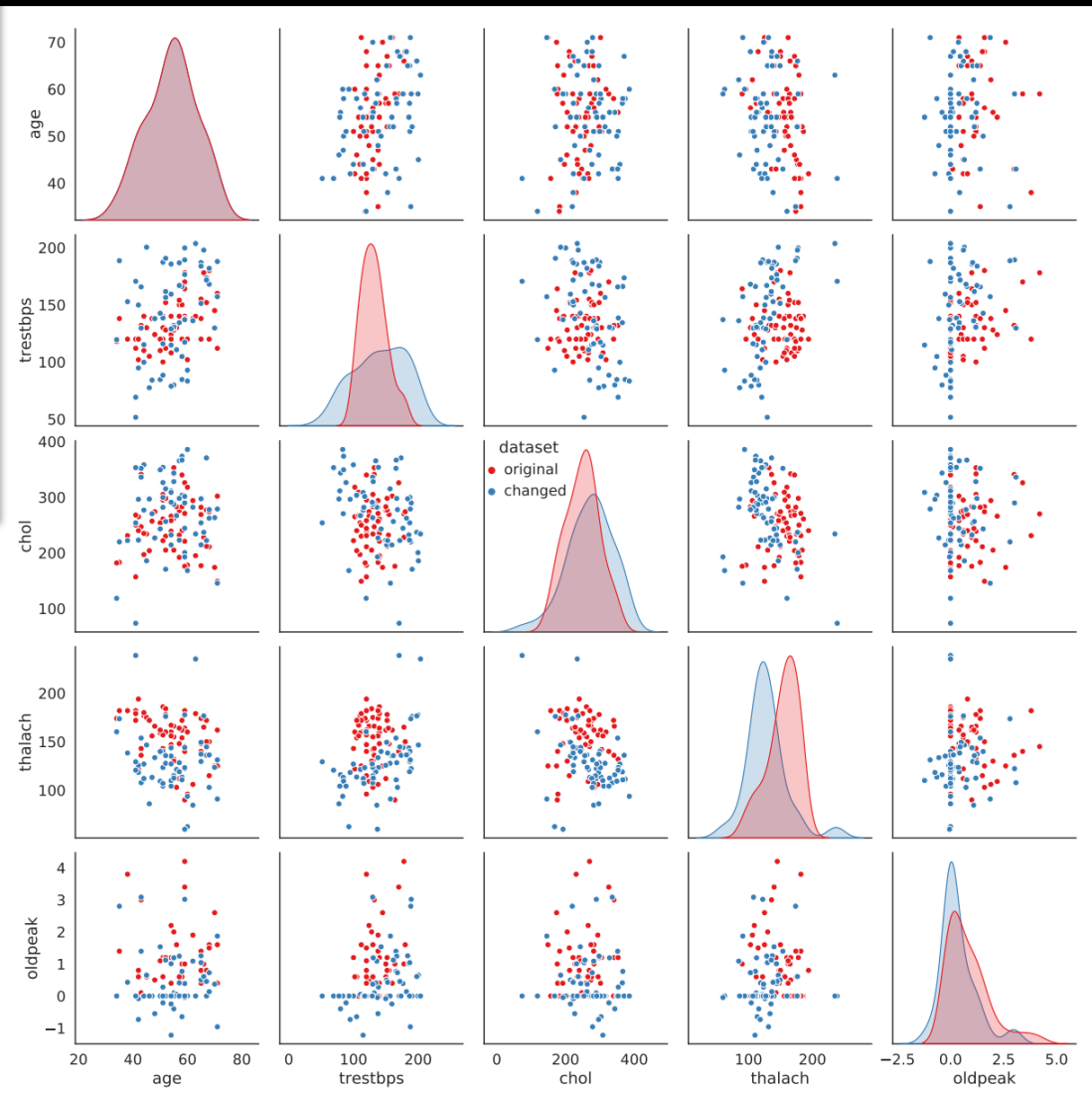


Figure 3. Result of the genetic-based attack changing 4 out of 13 explanatory variables (`age` and remaining variables are constant). This behaviour can be tuned using the α parameter to regularize the cost function - for this example we set α to 0. Some values may drift out of distribution e.g. to negative values, which is a possible improvement for the future works.

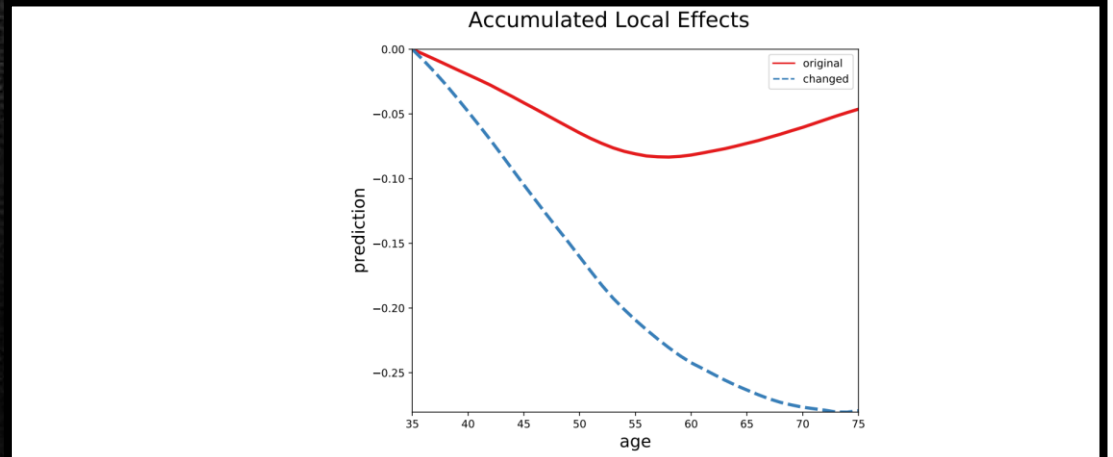


Figure 4. Result of the gradient-based attack on ALE explanation of `age` (data: `heart`; model: `neural n`

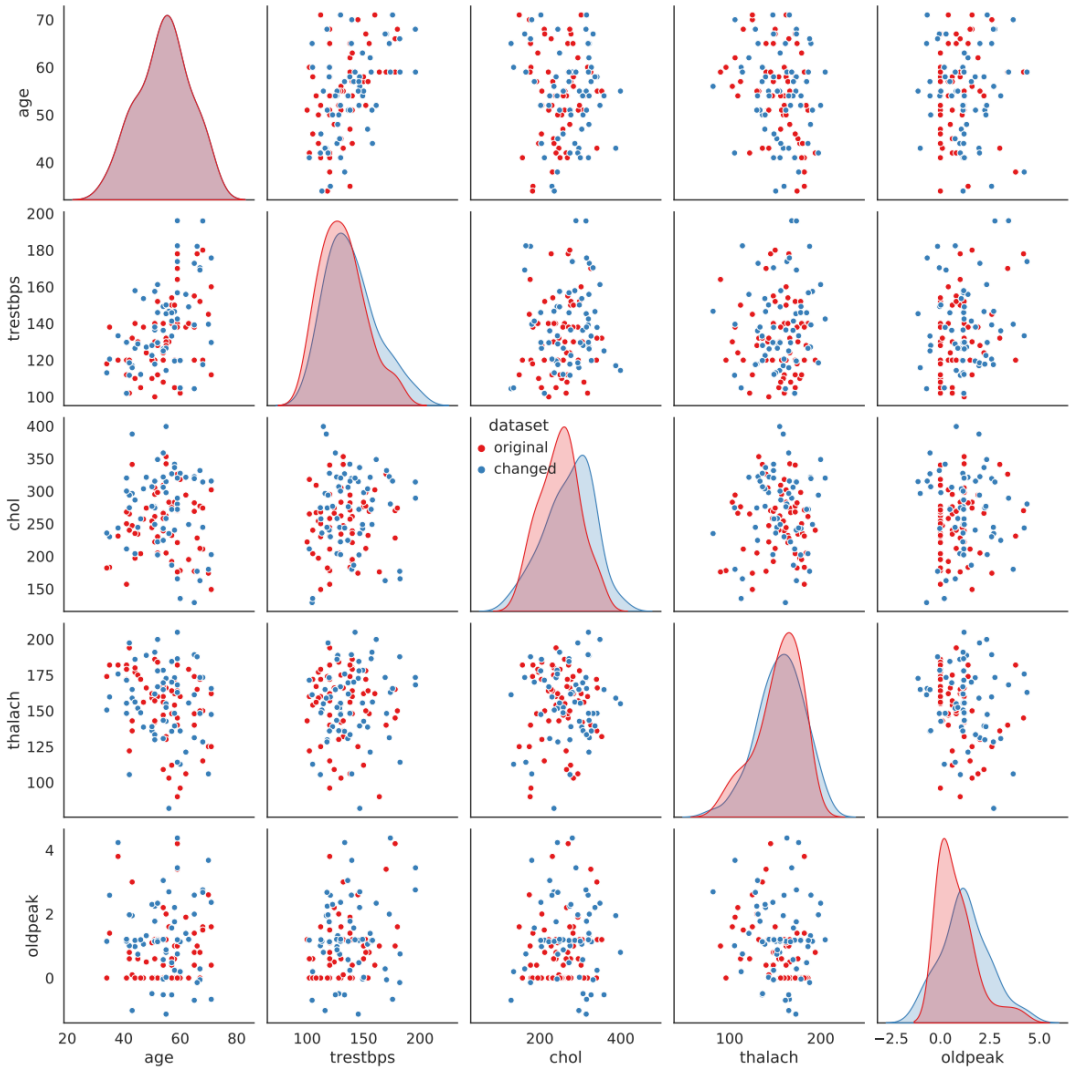
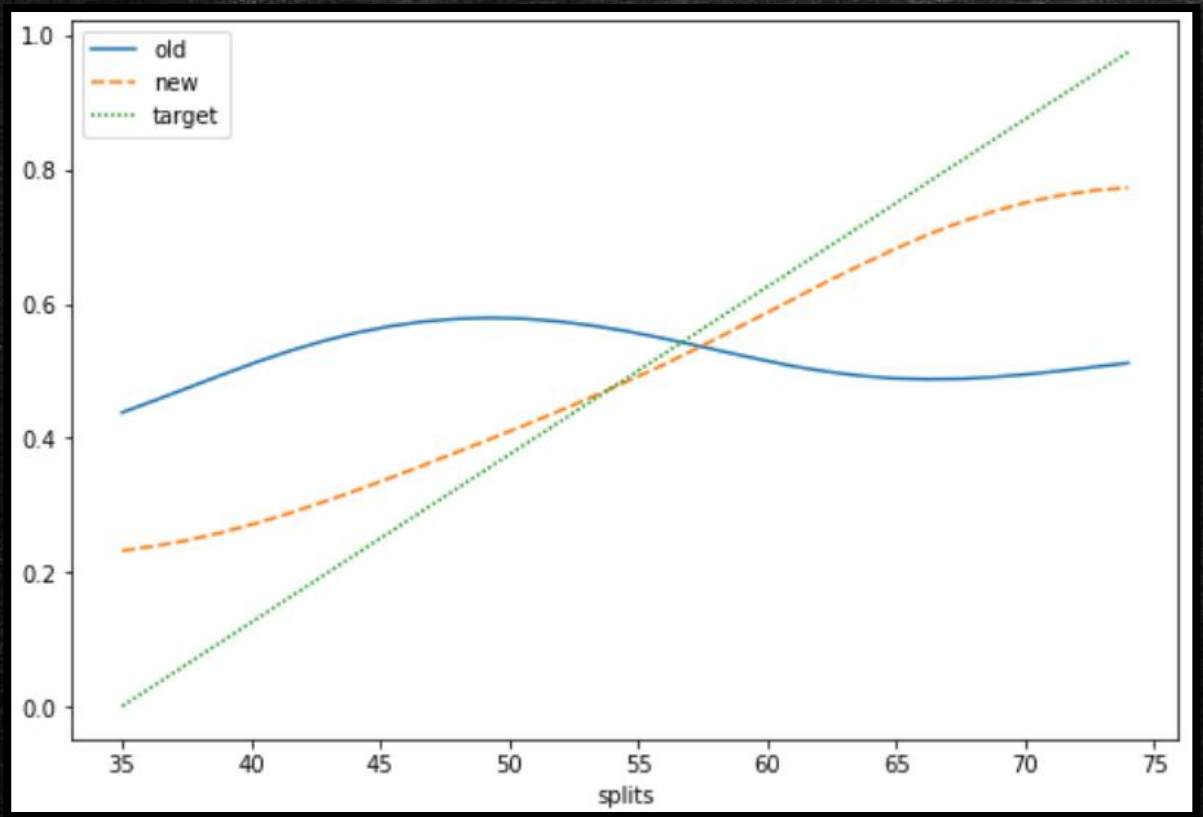
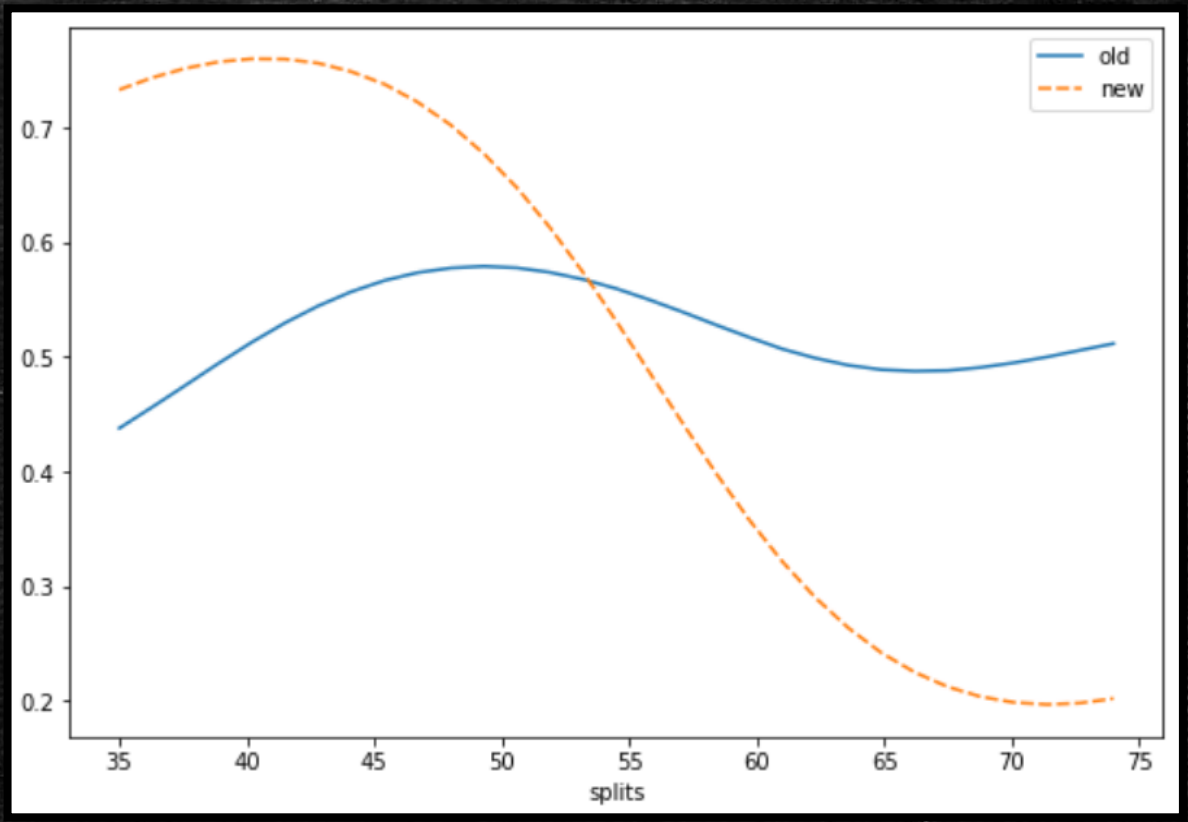


Figure 5. Result of the gradient-based attack changing 4 out of 13 explanatory variables (`age` and remaining variables are constant). This behaviour can be tuned using the α parameter to regularize the cost function - for this example we set α to 0. Some values may drift out of distribution e.g. to negative values, which is a possible improvement for the future works.

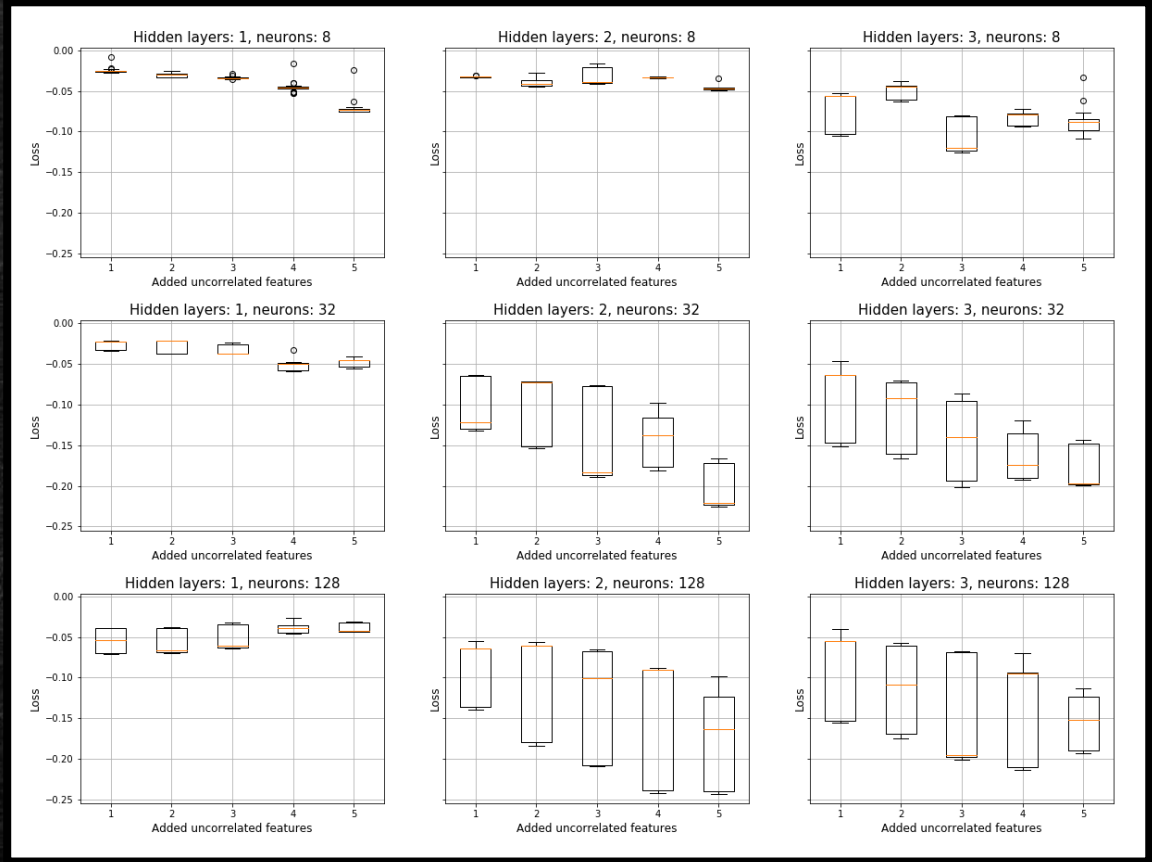
Problem


old
new
target
1.0
0.8
0.6
0.4
0.2
0.0
35
40
45
50
55
60
65
70
75
splits

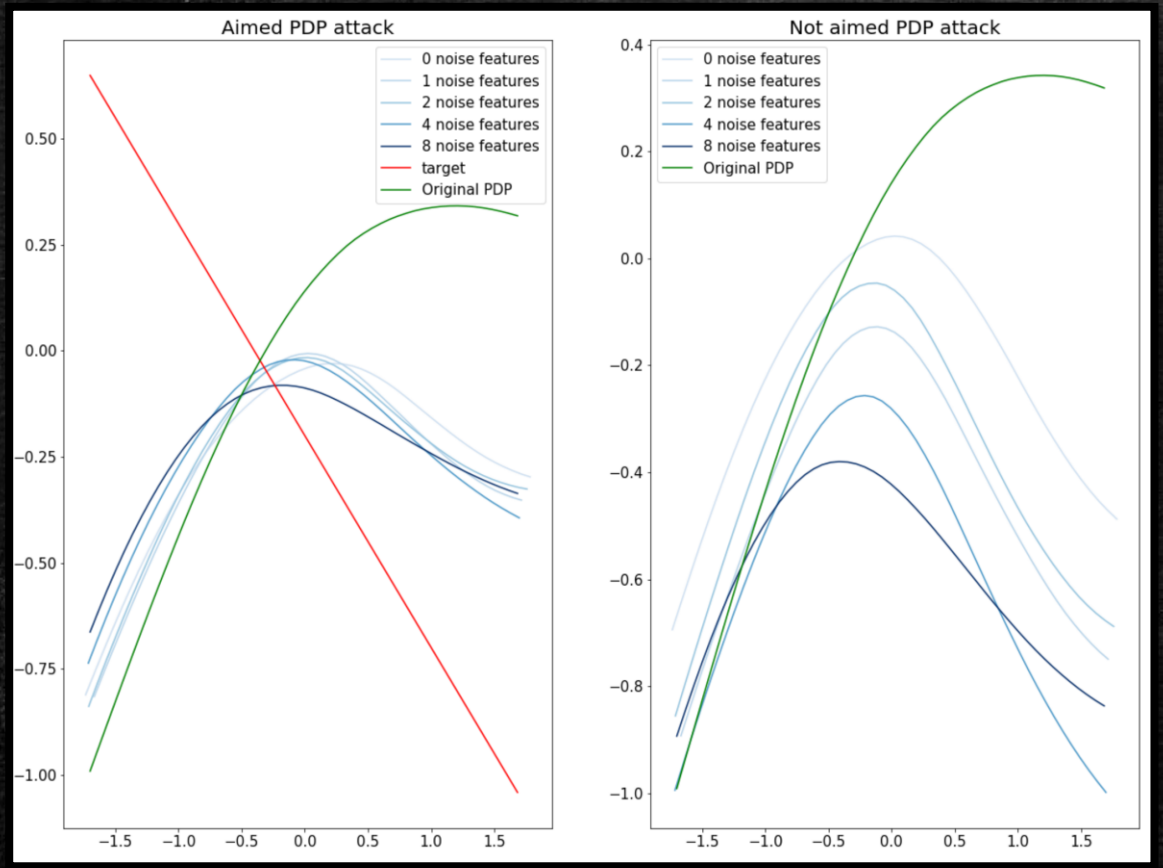

old
new
0.7
0.6
0.5
0.4
0.3
0.2
35
40
45
50
55
60
65
70
75
splits

Eksperymenty

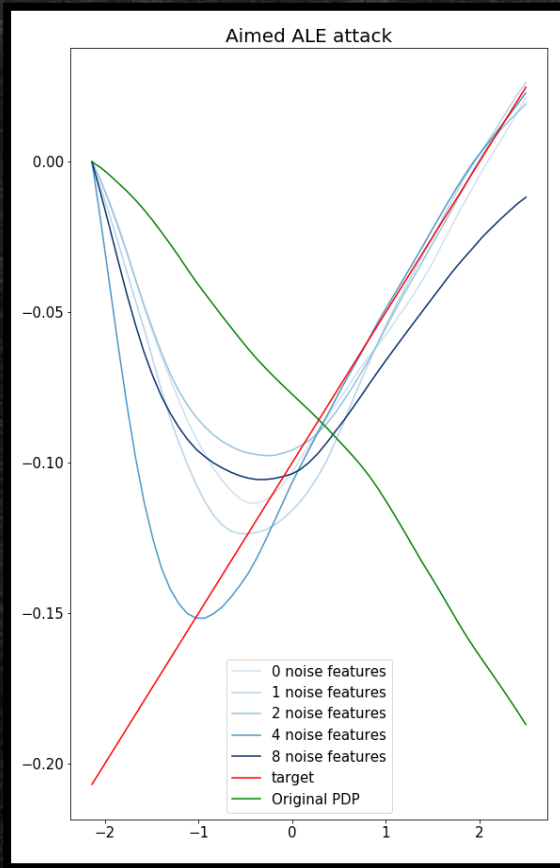
- **Q1**: Jak złożoność modelu wpływa na skuteczność ataku?
- **Q2**: Jak dodanie szumu do danych wpływa na skuteczność ataku?
- **Q3**: Które modele są podatne na ataki, a które odporne?
- Qx: Czy trudniej jest zaatakować ALE niż PDP?
- Qy: Jak wygląda porównanie działania algorytmów dla sieci?

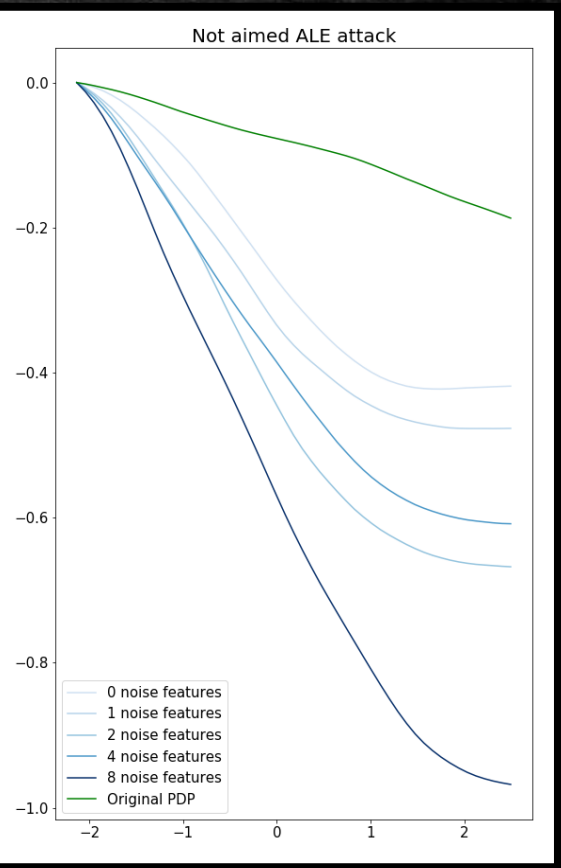


Wyniki


Aimed PDP attack
0 noise features
1 noise features
2 noise features
4 noise features
8 noise features
target
Original PDP
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

Not aimed PDP attack
0 noise features
1 noise features
2 noise features
4 noise features
8 noise features
Original PDP
0.4
0.2
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5

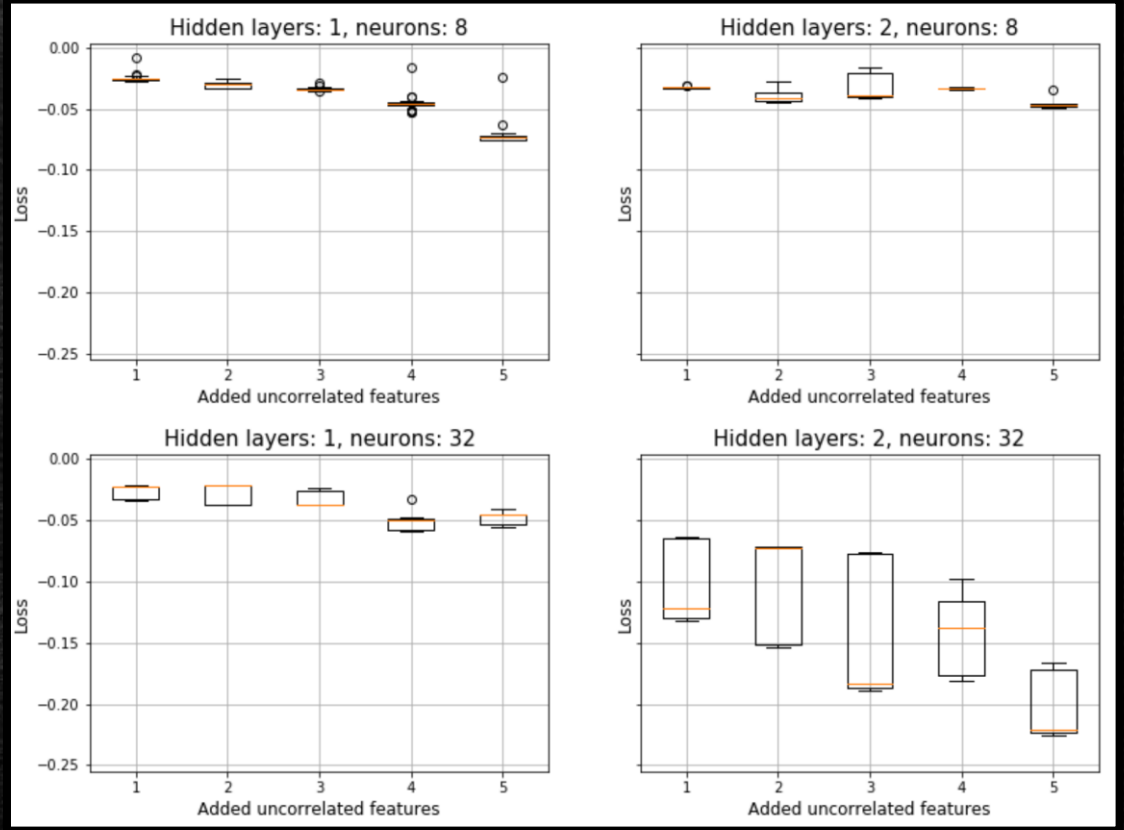

Aimed ALE attack
0 noise features
1 noise features
2 noise features
4 noise features
8 noise features
target
Original PDP
0.00
−0.05
−0.10
−0.15
−0.20
−2
−1
0
1
2


Not aimed ALE attack
0 noise features
1 noise features
2 noise features
4 noise features
8 noise features
Original PDP
0.0
−0.2
−0.4
−0.6
−0.8
−1.0
−2
−1
0
1
2

Wyniki (PDP)

model	aggregate rank of 4 tasks mean	median
SVM	1.500000	1.0
NN	2.833333	2.0
DecisionTree	2.916667	3.0
KNN	3.833333	4.0
RandomForest	4.916667	5.0
Boosting	5.000000	5.0
LM	7.000000	7.0

model	aggregate rank of 4 tasks mean	median
NN16x32x16	1.75	1.5
NN32x64x32	2.00	2.0
NN32x32	2.25	2.5
NN16	6.75	6.0
RF10	7.00	7.5
NN16x16	7.00	5.0
NN32	8.00	7.5
RF20	8.50	9.0
RF40	9.25	10.0
RF320	10.00	9.5
GBM320	11.00	10.5
RF160	11.00	10.5
RF80	11.25	12.0
GBM80	13.50	14.0
GBM160	13.75	15.0
GBM40	14.50	14.5
GBM20	15.50	16.0
GBM10	18.00	18.0



Wartość dodana

1. Ewaluacja wyjaśnień PDP i ALE wykorzystywanych w praktyce badawczej i komercyjnej
2. Rozwinięcie dziedziny zdominowanej dotychczas przez metody dla wyjaśnień modeli klasyfikacji obrazu
3. Propozycja wykorzystania algorytmu genetycznego jako ogólna metoda dla wielu modeli i wyjaśnień niezależnie od struktury
4. Dostępna implementacja Pythonowa kompatybilna z popularnymi pakietami do uczenia maszynowego (ft. dalex)