Ottimizzazione del design tramite Algoritmi Evolutivi e modello surrogato Machine Learning: caso di studio industriale

Tesi di Laurea magistrale in Fisica dei Sistemi Complessi

di
Giacomo
Deandrea

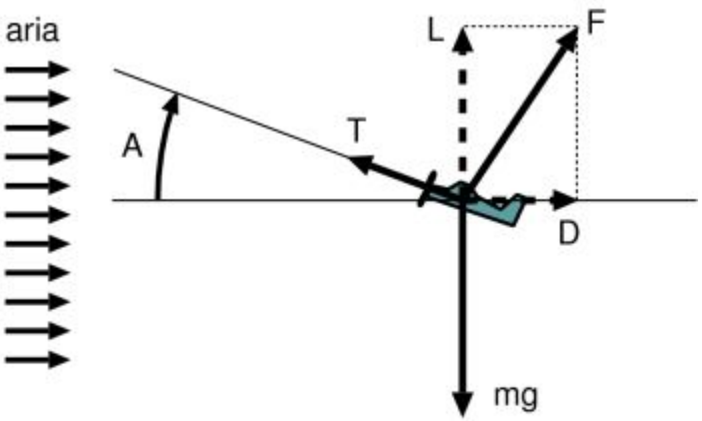
relatore
Prof.
Piero Fariselli

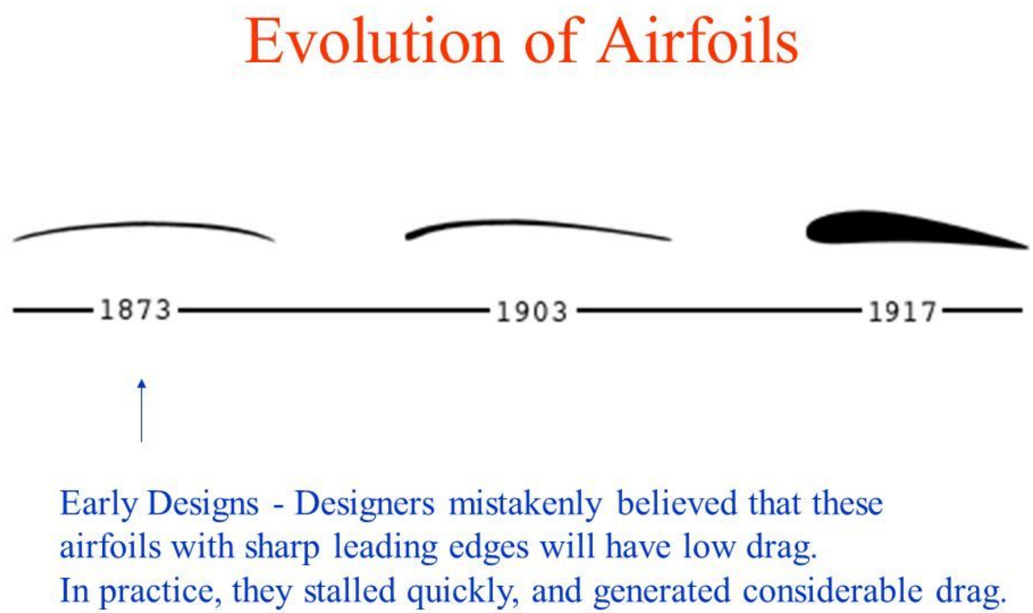
Overview

- **Industrial design optimization**
- Client's material and requirements
- Machine Learning modeling
- Optimization loop
- Results
- Final Remarks

Industrial Design Optimization

- **Iterative process to develop a product that achieves a given task**





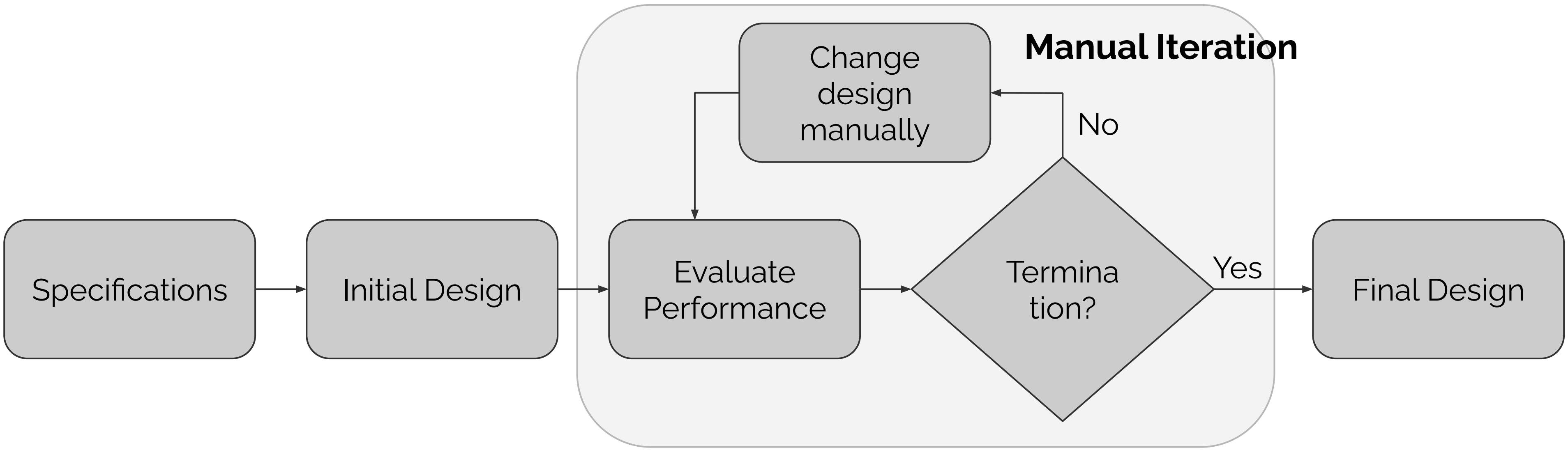
Examples:

- **Finance- Portfolio optimization**
- **Architecture- Structure optimization**
- **Machine Learning - Hyperparameter optimization**

Ref: J. R. R. A. Martins and A. Ning, "Engineering design optimization" 2021.

Industrial Design Optimization

- **Traditional loop**

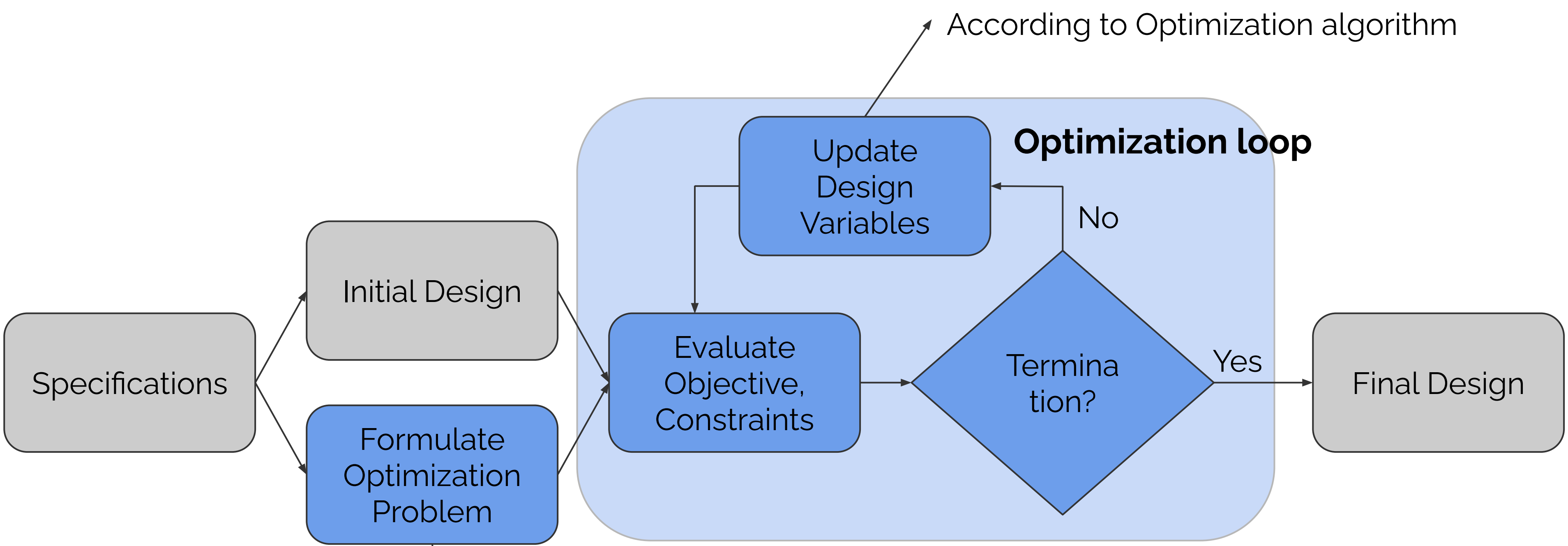


- Specifications and change of design are user defined
- Subjectivity in decisions is a problem to achieve optimal results

Ref: J. R. R. A. Martins and A. Ning, “Engineering design optimization” 2021.

Industrial Design Optimization

Optimization loop



- Design Variables Space $\mathbf{x} \in X \subseteq \mathbb{R}^k$
- Objective Functions $f(\mathbf{x}) : \mathbb{R}^k \longrightarrow \mathbb{R}^q$
- Constraints $g_i(\mathbf{x}) < b_i$ e $h_j(\mathbf{x}) = c_j$

$$i = 1, \ldots, l \qquad j = 1, \ldots, p$$

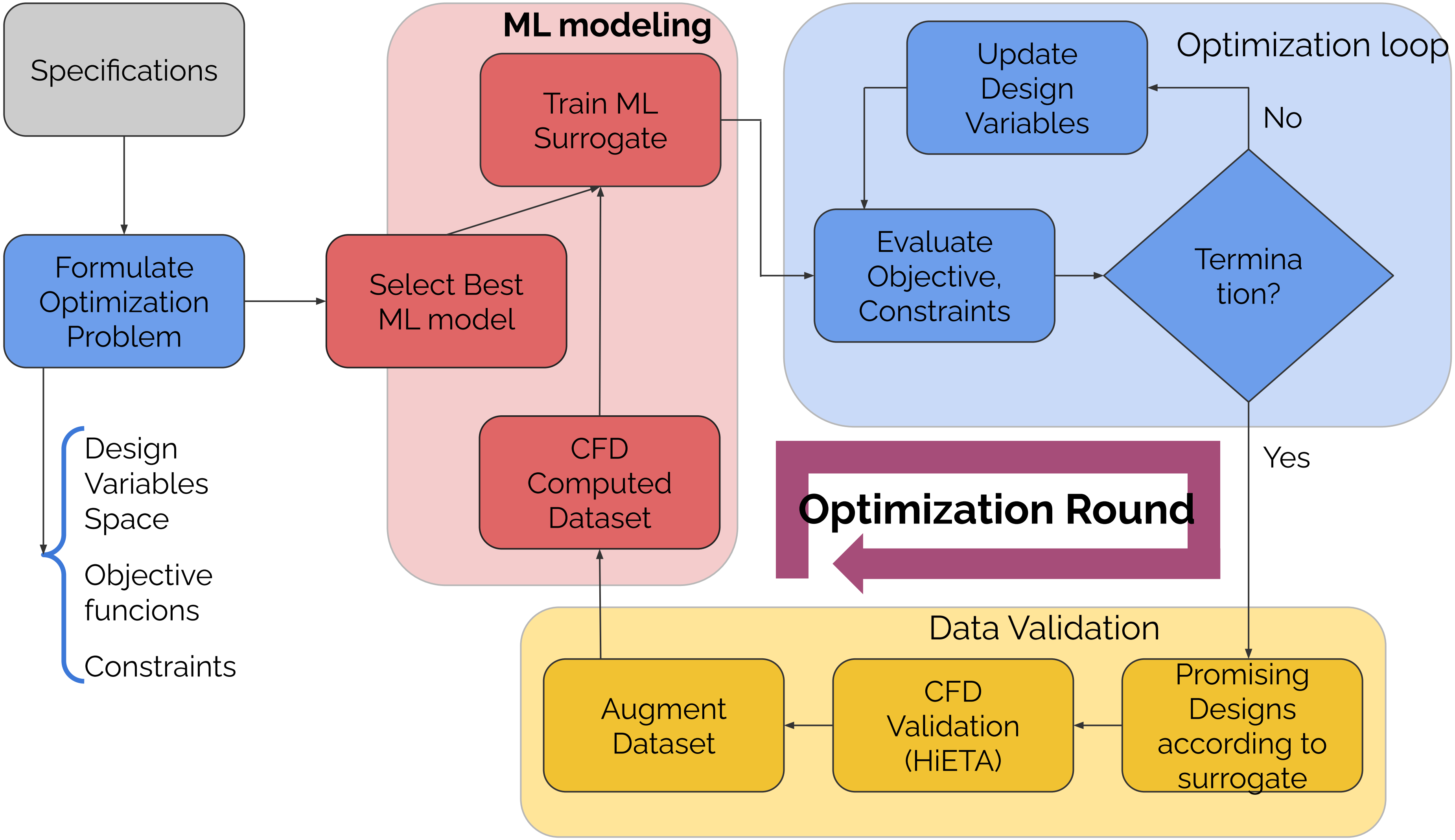
Ref: J. R. R. A. Martins and A. Ning, “Engineering design optimization” 2021.

Client's material and requirements

- Design space: 6 **features**, 3 **targets** + constraints
- Grid search optimization (4096 points) using CFD model of the component as objective function
- CFD very expensive to compute (~1min x design)
- Provide a framework to find better designs for the component using AI

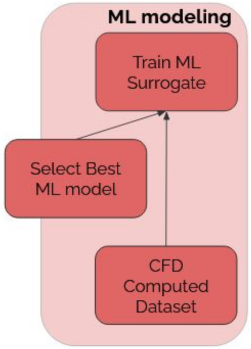
	feature1	feature2	feature3	feature4	feature5	feature6	target1	target2	target3
4091	0.75	0.663927	0.45	0.477041	1.620003	0.786852	0.207774	0.037435	0.081406
4092	0.60	0.704913	0.36	0.485889	1.620003	0.786852	0.311409	0.048445	0.117979
4093	0.65	0.713656	0.39	0.496380	1.620003	0.786852	0.266211	0.044430	0.109598
4094	0.70	0.722399	0.42	0.506872	1.620003	0.786852	0.226846	0.040863	0.109968
4095	0.75	0.731142	0.45	0.517363	1.620003	0.786852	0.220808	0.039623	0.106437

Framework



Specifications
Formulate Optimization Problem
Design Variables Space
Objective funcions
Constraints
ML modeling
Train ML Surrogate
Select Best ML model
CFD Computed Dataset
Optimization loop
Update Design Variables
Evaluate Objective, Constraints
Termina tion?
No
Yes
Optimization Round
Data Validation
Augment Dataset
CFD Validation (HiETA)
Promising Designs according to surrogate

ML modeling



Supervised Machine Learning Problem Definition:

Starting from a dataset $\boldsymbol{D} = \{(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_N, y_N)\}$

Where $y_i = f(\mathbf{x}_i)$

With $f(\mathbf{x}) : \boldsymbol{X} \subseteq \mathbb{R}^k \longrightarrow \boldsymbol{Y} \subseteq \mathbb{R}$ supposed true function

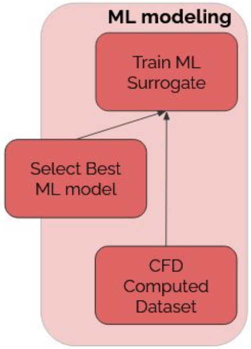
We look to find a function $g(\mathbf{x}, \hat{\theta}) : \boldsymbol{X} \subseteq \mathbb{R}^k \longrightarrow \boldsymbol{Y} \subseteq \mathbb{R}$

Such that $\hat{\theta} = argmin_\theta \left[\sum_{i=1}^{N} L(y_i, \hat{y}_i) \right]$

With $g(\mathbf{x}_i, \theta) = \hat{y}_i$

Ref: P. Mehta, "A high-bias, low-variance introduction to machine learning for physicists," 2018

ML modeling



- **Problem under consideration:**

Supervised: true function values for data in dataset at our disposal

$$y_i = f(\mathbf{x}_i)$$

Multioutput: more than one target depending from same features

$$f(\mathbf{x}) : \mathbb{R}^k \longrightarrow \mathbb{R}^q;\ f(\mathbf{x}) = (f_1(\mathbf{x}), \ldots, f_q(\mathbf{x}))$$

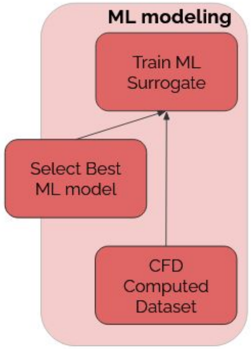
Regression: predicting continuous numerical values

$$L(y_i, \hat{y}_i) = (y_i - \hat{y}_i)^2$$

Moreover:

N = 4096
k = 6
q = 3

ML modeling



- **Model selection:**

  Elastic Net: regularized version of Linear Regression (L1 and L2 penalty)

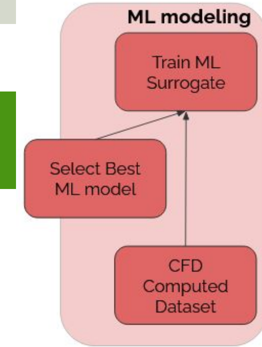
  Support Vector Regression: popular and effective non linear model for regression

  Random Forest: popular and effective ‘bagging’ method (CARTS

  Xtreme Gradient Boosting: popular and effective ‘boosting’ method

- Need to estimate model performance together with best hyperparameters

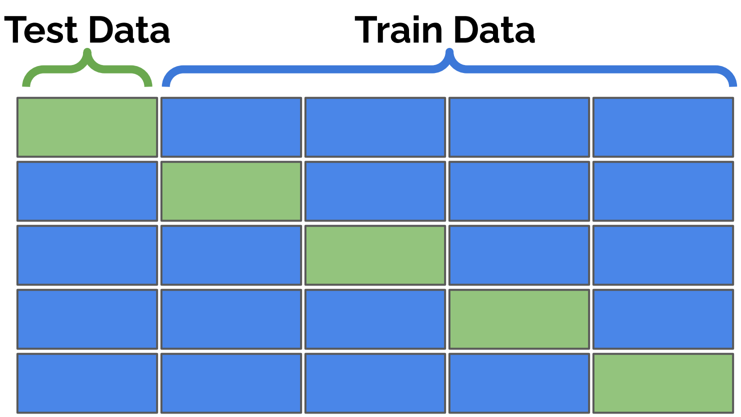
ML modeling



- How to compare different models? Out of sample performance

K-Fold Cross Validation - OOS + hyperparam performance estimation -> Positive Bias

Hyperparams selection + Model evaluation



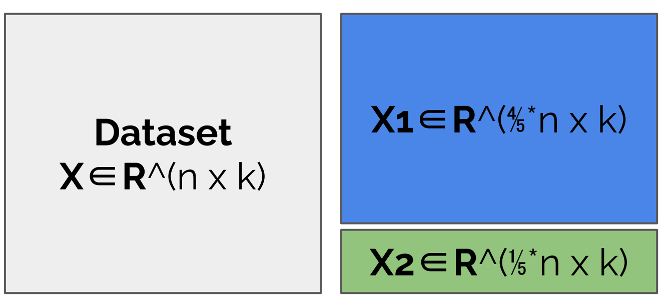
→ Error1 = sqrt(mean[(**y2**-**ŷ2**)^2])

→ Error2

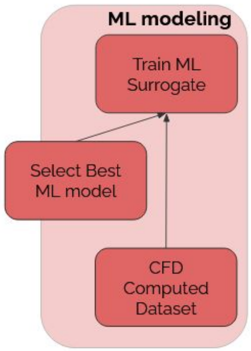
→ Error3

→ Error4

→ Error5



Split data
Select w randomly from search space
Select Train & Test data
Train f(**X1**, w) = **ŷ1**;
w_score = sqrt[mean((**y2**-**ŷ2**)^2)]
(**x5**) ⟶ w_score = mean([w_score])
Repeat **n_iter** times
w = argmin_w[[w_score]]
score = min[[w_score]]

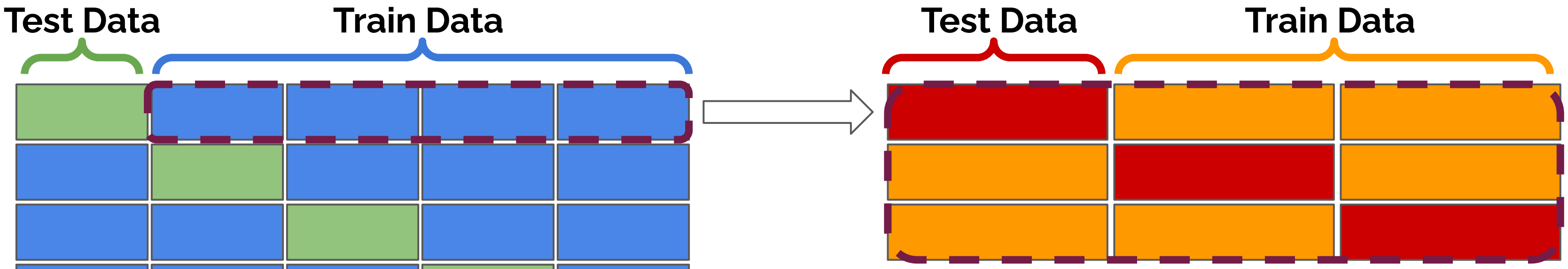


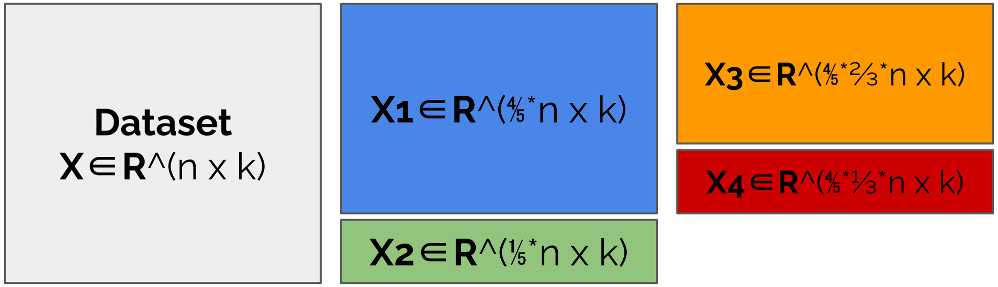
ML modeling

- **Nested K-Fold Cross Validation**

Outer Loop - Model evaluation

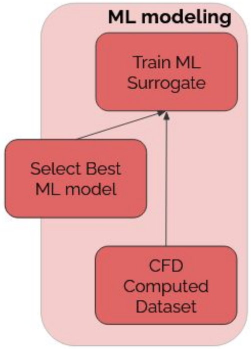
Inner Loop - Hyperparams selection



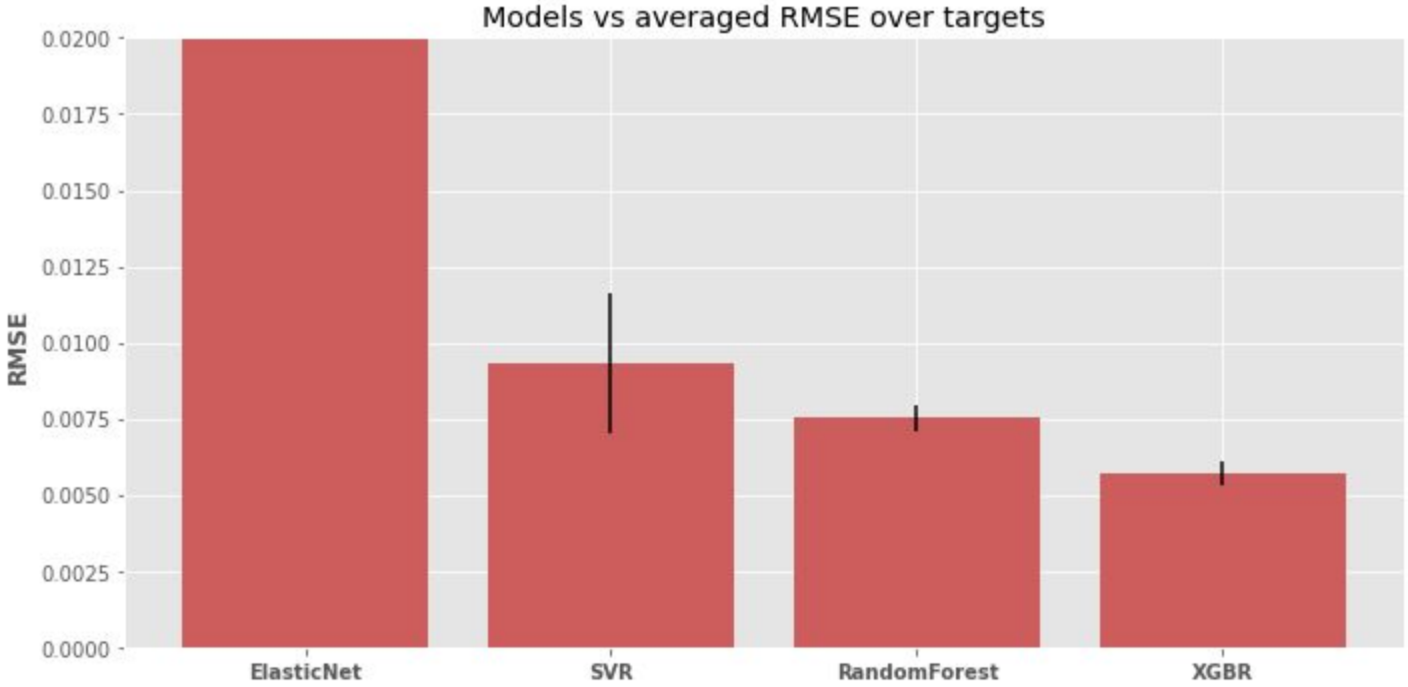


Split data in Outer loop
Select Train & Test data
Select w randomly from search space
Split data in inner loop, selecting Train & Test data
Train f(**X3**, w) = **ŷ3**;
w_score3 = sqrt[mean[(**y3**-**ŷ3**)^2]]
(**x3**) ⟶ w_score = mean[{w_score3}]
Repeat **n_iter** times
Select w* = argmin_w[{w_score}]
Train f(**X1**, w*) = **ŷ1**;
test_score = sqrt[mean[(**y1**-**ŷ1**)^2]];
Repeat **5** times
score = mean[{test_score}]
error_score = (max[{test_score}] - min[{test_score}])/2

ML modeling

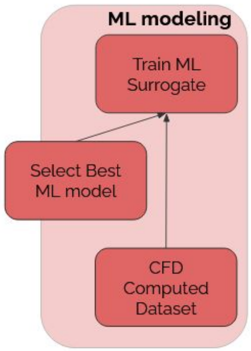


- **cenno su allenamento modello utilizzato come surrogato in loop ottimizzazione**
  cenno calcolo errori



Pred times are for every model (~10^-3 s compared to ~CFD pred time ~60 s)

ML modeling



- **XGBoost** - Xtreme Gradient Boosting

Additive model: $\hat{y}_i(x_i) = \sum_{j=1}^{J} f_j(x_i),\ f_j \in \mathcal{F}$ combines different CART

$$\hat{y}_i^{(t)} = \sum_{j=1}^{t} f_j(x_i) = \hat{y}_i^{(t-1)} + f_t(x_i)$$

$$\begin{cases} g_i = \partial_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \\ h_i = \partial^2_{\hat{y}_i^{(t-1)}} l(y_i, \hat{y}_i^{(t-1)}) \end{cases}$$

$$obj^{(t)} = \sum_{i=1}^{n} l(y_i, \hat{y}_i^{(t-1)} + f_t(x_i)) + \sum_{j=1}^{t} w(f_j) \xrightarrow{\text{Taylor}} obj^{(t)} \approx \sum_{i=1}^{n} [g_i f_t(x_i) + \frac{1}{2} h_i f_t^2(x_i)] + w(f_t)$$

$$f_t(x) = w_{q(x)};\ w \in R^L;\ q(x) : R^k \longrightarrow 1, 2, \ldots, L$$

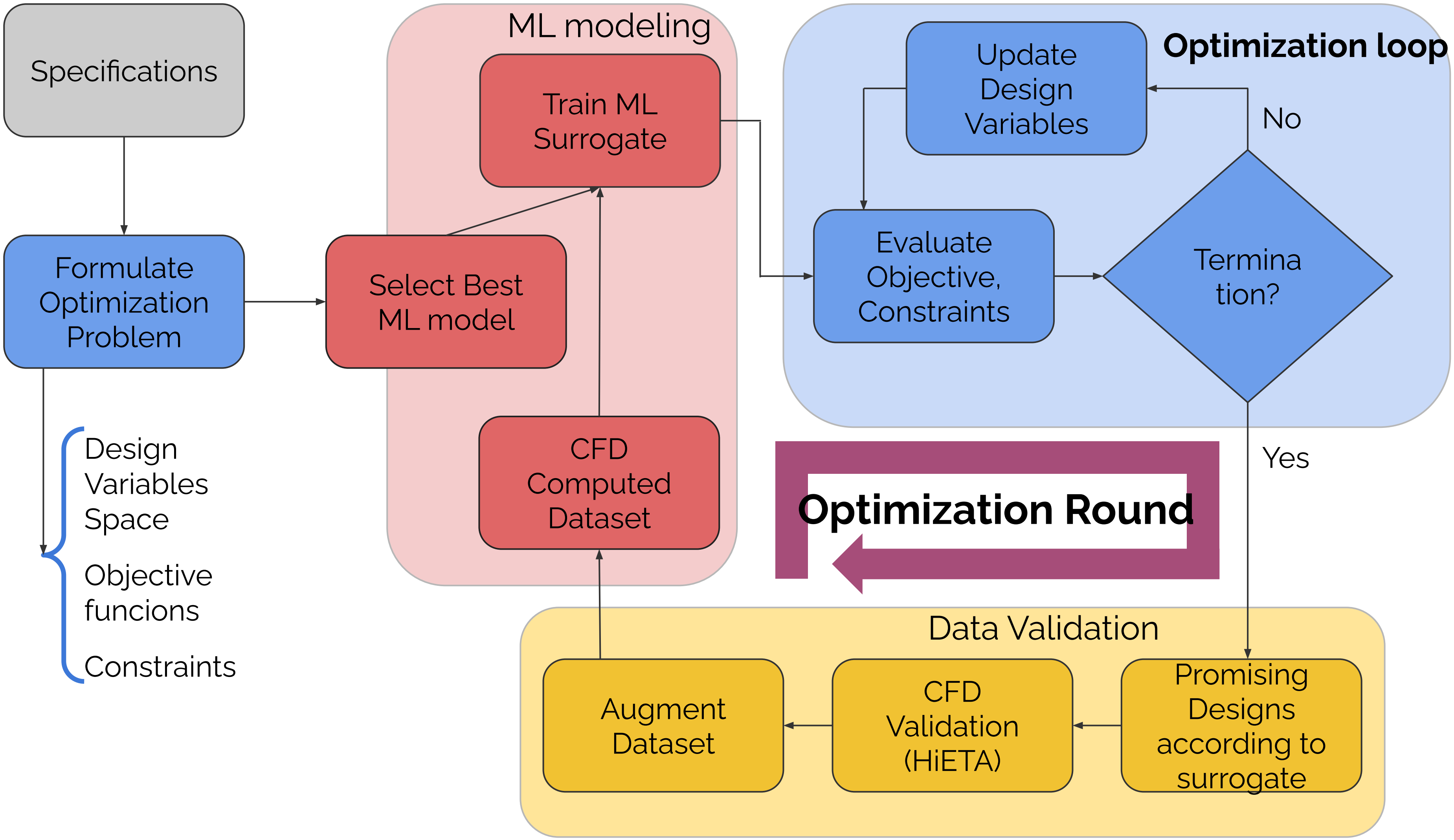
$$obj^{(t)} \approx \sum_{i=1}^{n} [g_i w_{q(x_i)} + \frac{1}{2} h_i w_{q(x_i)}^2] + \gamma L + \frac{1}{2}\lambda \sum_{l=1}^{L} w_l^2$$

$$obj^{(t)} \approx \sum_{l=1}^{L} [(\sum_{i \in I_l} g_i) w_l + \frac{1}{2}(\sum_{i \in I_l} h_i + \lambda) w_l^2] + \gamma L \qquad \begin{cases} G_l = \sum_{i \in I_l} g_i \\ H_l = \sum_{i \in I_l} h_i \end{cases}$$

$$w_l^* = -\frac{G_l}{H_l + \lambda} \qquad obj^* = -\frac{1}{2} \sum_{l=1}^{L} \frac{G_l^2}{H_l + \lambda} + \gamma L$$

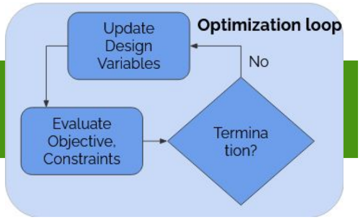
Greedy approx (would need to calculate obj* for every tree structure)

Framework



Specifications
Formulate Optimization Problem
Design Variables Space
Objective funcions
Constraints
ML modeling
Train ML Surrogate
Select Best ML model
CFD Computed Dataset
Optimization loop
Update Design Variables
Evaluate Objective, Constraints
Termina tion?
No
Yes
Optimization Round
Data Validation
Augment Dataset
CFD Validation (HiETA)
Promising Designs according to surrogate

Optimization loop



- **Optimization problem definition:**

Constrained
linear constraints among features

Lagrange Multipliers method?

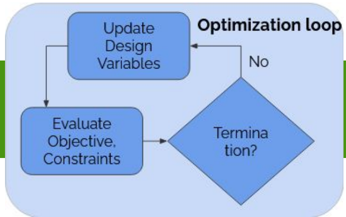
Black Box
no analytical function ties inputs to outputs (XGBoost using CARTs)

Quasi-Newton methods?

Multiobjective
multiple objectives to be minimized simultaneously

Evolutionary Algorithms!

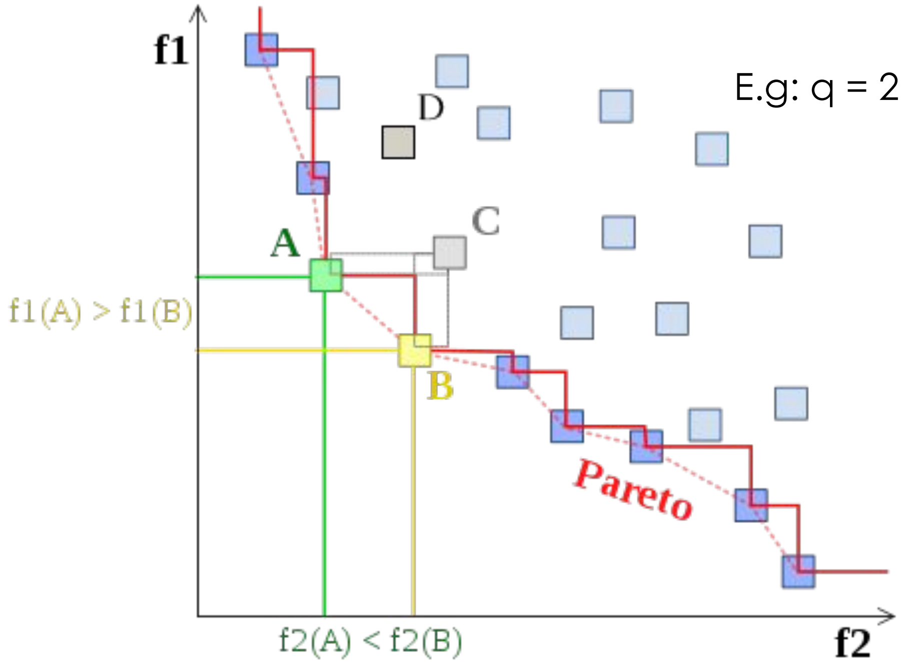
Optimization loop



Pareto Dominance

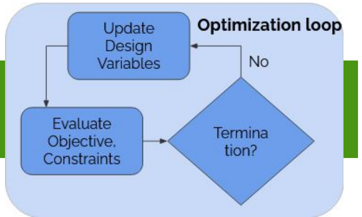
Impossible (in non trivial problems) to tell who is the 'best' according to multiple criteria

$$F(x) : x \in \mathbb{R}^k \longrightarrow y \in \mathbb{R}^q, \; y = (f_1(x), f_2(x), \ldots, f_q(x))$$



$$x^* \in P \iff \nexists\, x \in X \; t.c. \; f_i(x) \leq f_i(x^*) \; \forall\, i = 1, \ldots, q$$

Optimization loop



Evolutionary algorithms

Euristics: lack of mathematical proof of convergence

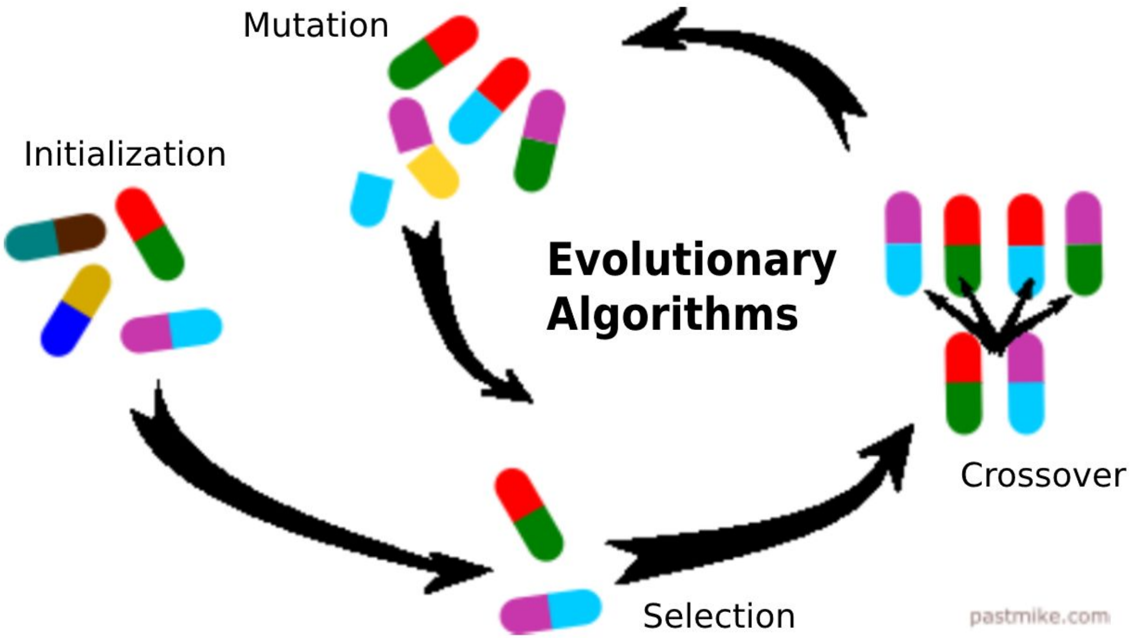
Born to simulate evolution of species

Population based

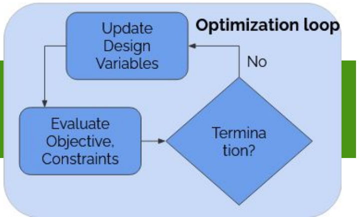
Fittest reproduces
-> mixes genes with other members of the population
-> genetic operators

Gene diversity among members of population - > stochasticity

Adaptable to multiobjective optimization problems!



Optimization loop



Differential Evolution (multiobjective)

Random in feature space
$G\colon\ x_{i.G} \in \mathbb{R}^k;\ i = 1, 2, \ldots, \lambda.$
λ = num of vectors in population

Initialization

While Termination == False:

$\forall\ x_i;\ i = 1, \ldots, \lambda$ 'father' vector

parla di constraints and termination

select $x_{r1}, x_{r2} \neq x_i$ from Pareto front or randomly

select x_{best} in Pareto front if x_i is not, else $x_{best} = x_i$

compute 'donor' vector $v_i = x_i + F \cdot (x_{best} - x_i) + F \cdot (x_{r1} - x_{r2})$

Mutation

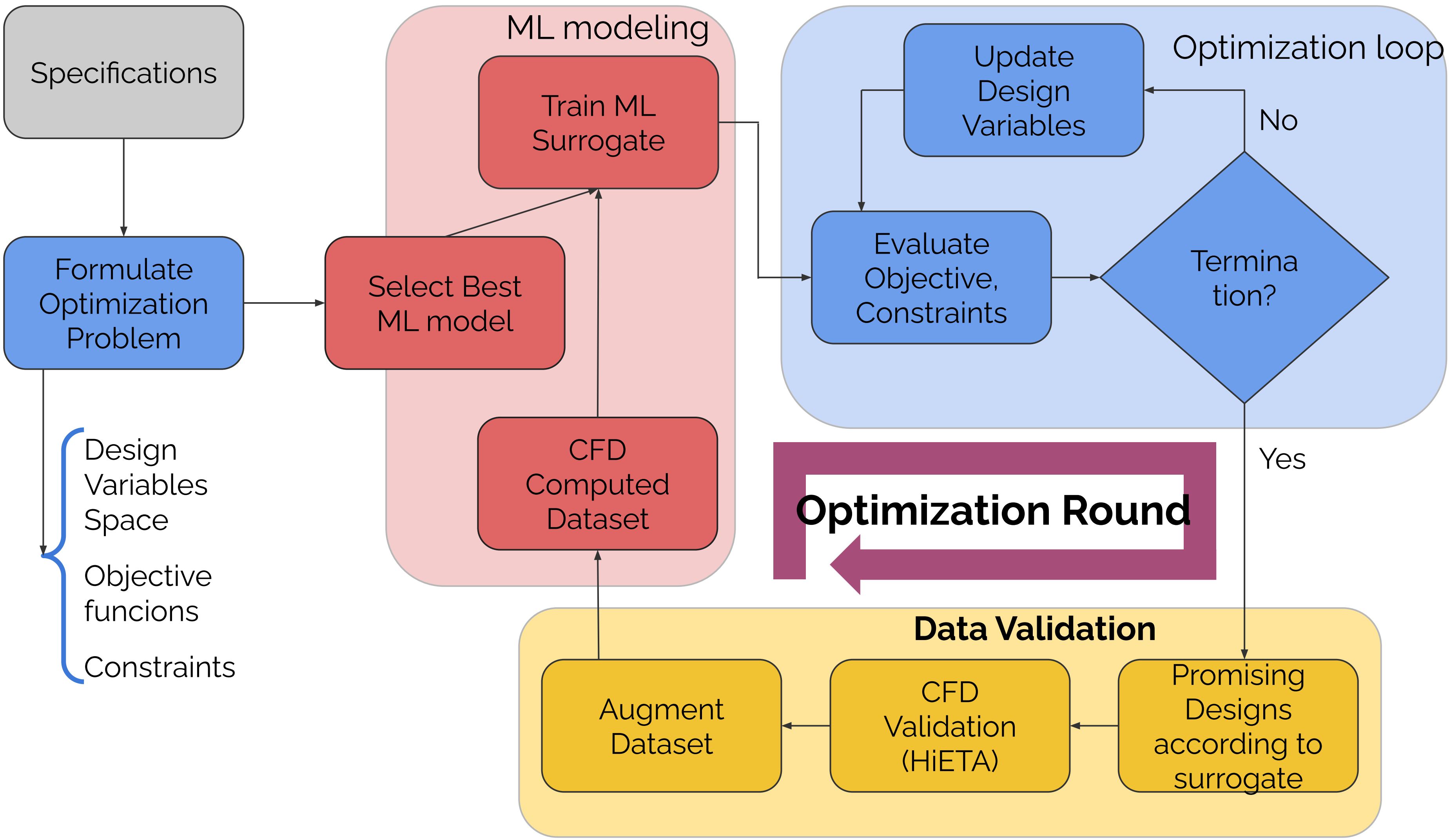
compute 'son' vector
$$u_i = \begin{cases} v_{i,j} \text{ se } j = j_{rand} \vee r_{i,j} \leq CR \\ x_{i,j} \text{ altrimenti} \end{cases}$$
Dove $i = 1, \ldots, \lambda;\ j = 1, \ldots, k;\ r_{i,j} \in \mathcal{U}(0,1)$

Crossover

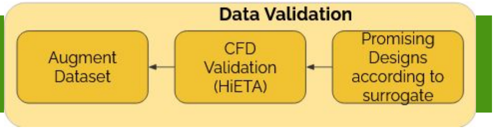
'son' substitute 'father' in population if 'son' in Pareto front while 'father' is not
or with proba (num of better targets/total num of targets)

Selection

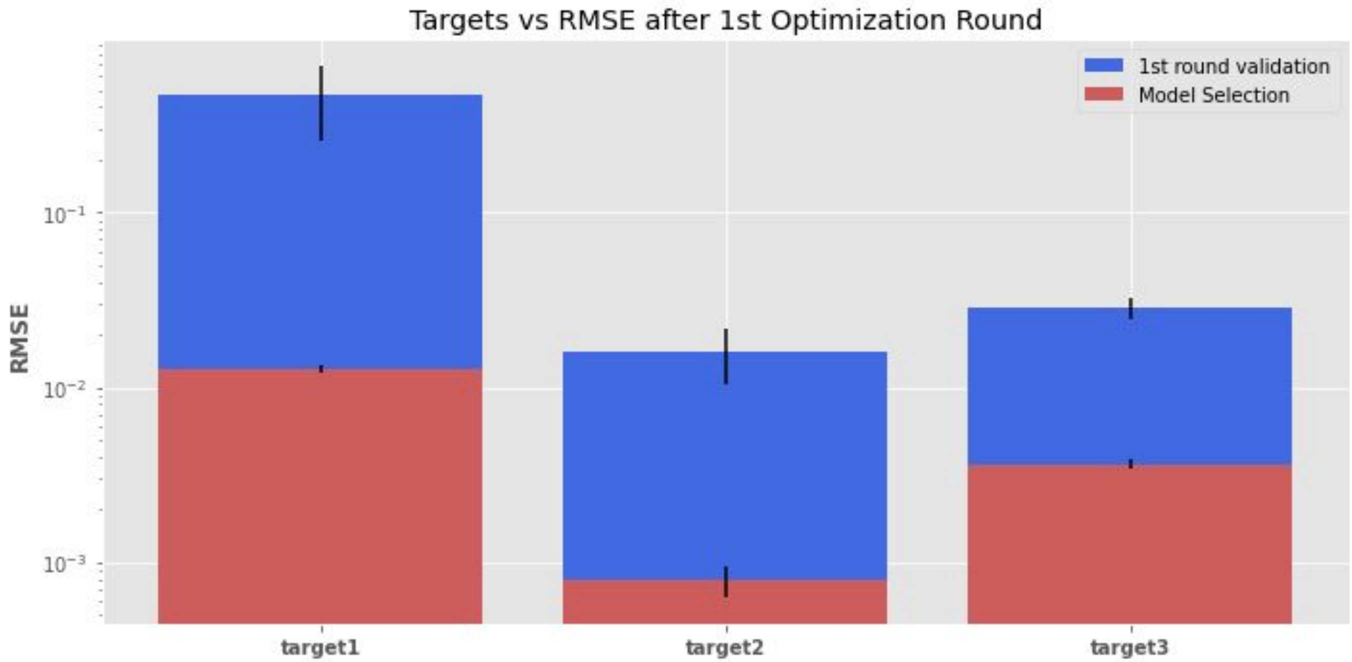
Framework


Specifications
Formulate Optimization Problem
Design Variables Space
Objective funcions
Constraints
ML modeling
Train ML Surrogate
Select Best ML model
CFD Computed Dataset
Optimization loop
Update Design Variables
No
Evaluate Objective, Constraints
Termina tion?
Yes
Optimization Round
Data Validation
Augment Dataset
CFD Validation (HiETA)
Promising Designs according to surrogate

Data Validation - 1st round



Lots of errors

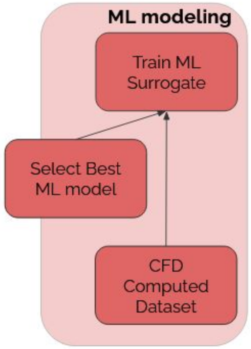


Errors on points sent to CFD validation way bigger than expected

Model not good in predicting points belonging to Pareto front region

(was the model overfitting? even if so...)

ML modeling - 2nd round



- **Change in model training schema**

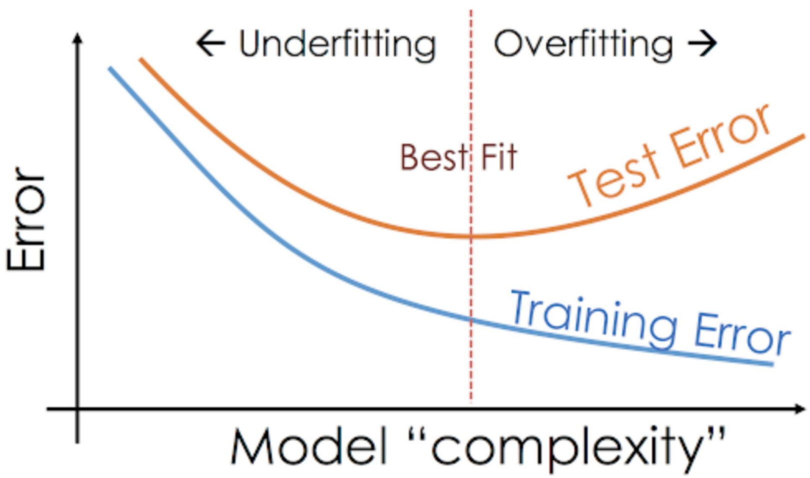
  Points ranked by uniform sum of standardized targets

  first 12% of ranked points defined to be in 'minima region'

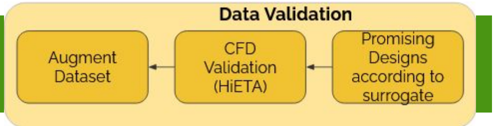
  10% of points in 'minima region' to be used as test set only

  Hyperparam selection
  remaining points in minima region used in 3-fold crossvalidation procedure:
  ⅔ of them as early stopping points
  ⅓ of them for training

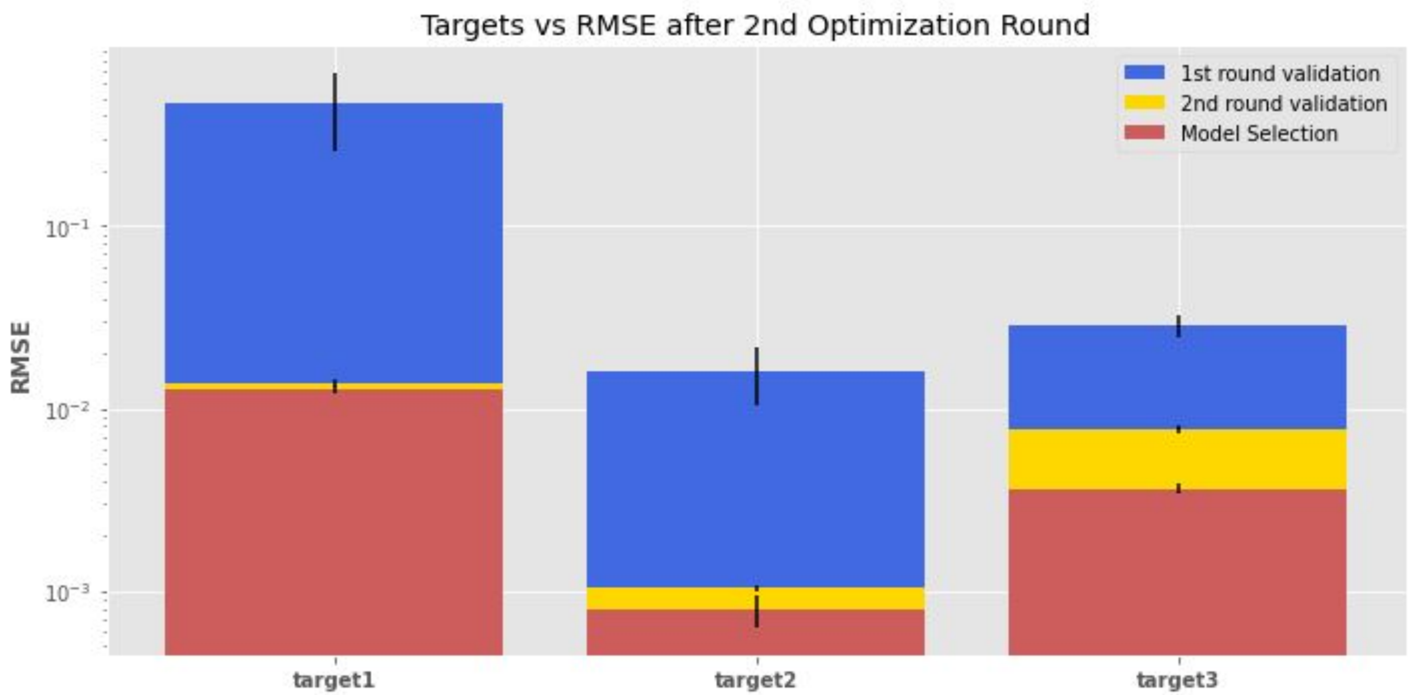
  Model training
  early stopping with ⅔ of the remaining points in minima region



Data Validation - 2nd round



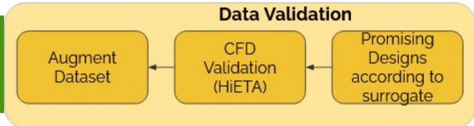
- **Better errors**



Model's new training scheme improved performance

New datapoints have surely helped

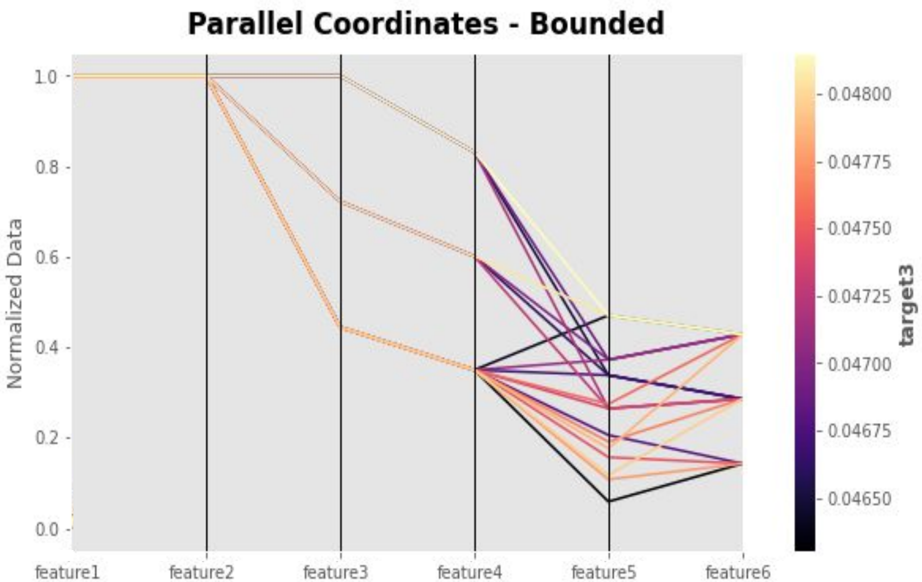
Data Validation - 2nd round

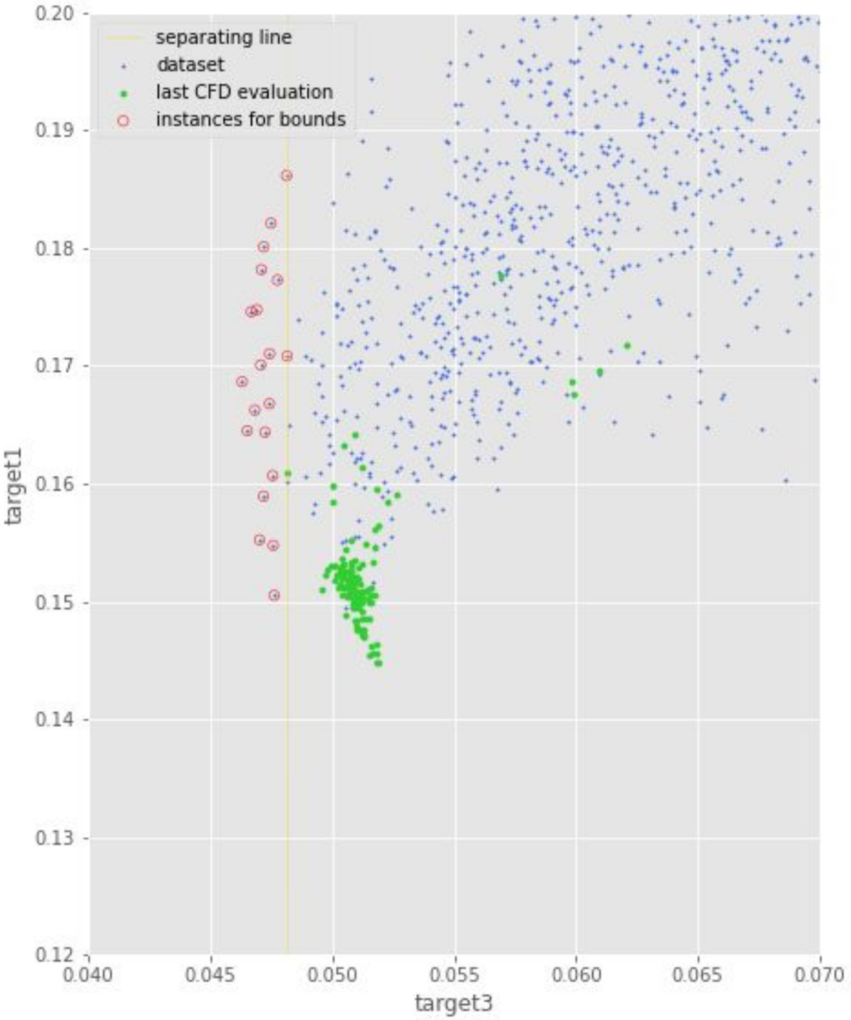


- **Points are not spread in Pareto front (hypervolume measure?)**

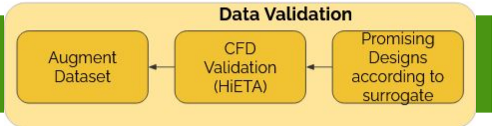
New minima for target 1 found, what about other targets?

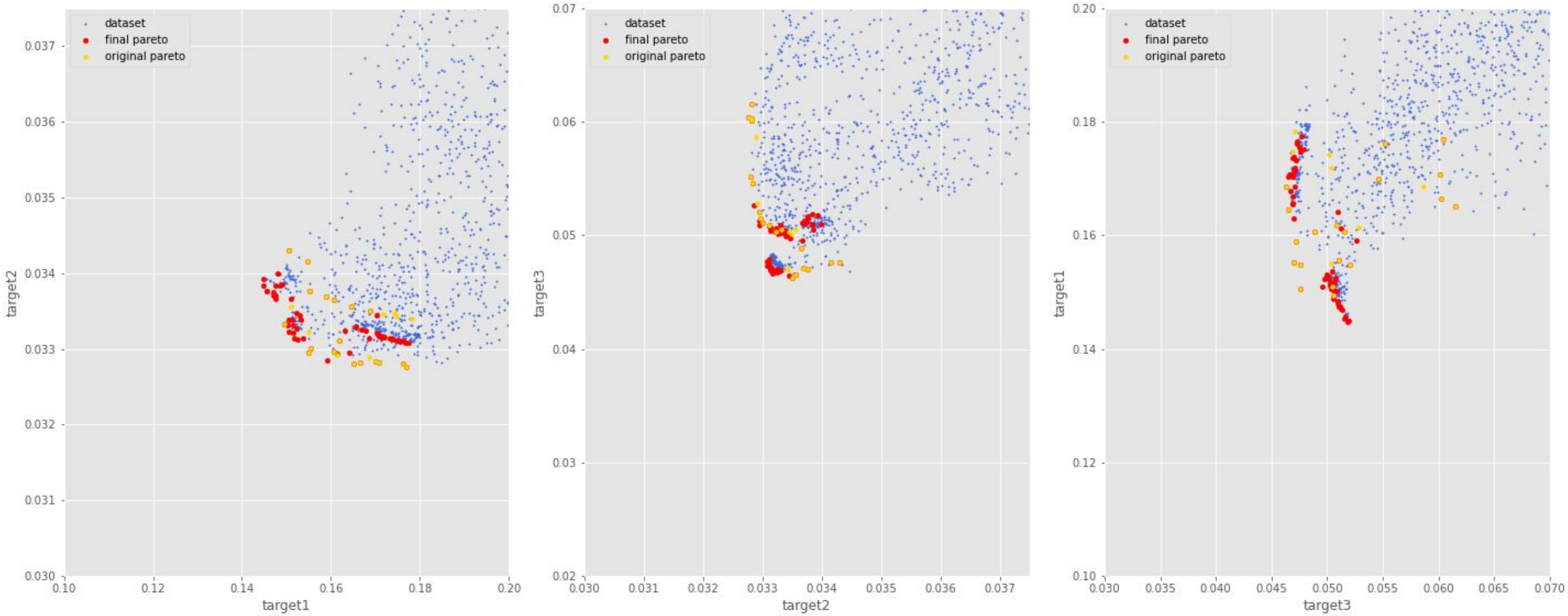
Bounding the search space for target 3





Data Validation - 3rd round






```
Num di punti del fronte di Pareto, 3 targets, dataset a lunghezza minima dopo 1° ciclo ottimizzazione:
 26
Num di punti del fronte di Pareto, 3 targets, dataset a lunghezza minima dopo 2° ciclo ottimizzazione:
 49
Num di punti del fronte di Pareto, 3 targets, dataset a lunghezza minima dopo 3° ciclo ottimizzazione:
 70
Num di punti dell intersezione tra i fronti di Pareto del 3° e 1° ciclo:
 18
Num di punti dell intersezione tra i fronti di Pareto del 3° e 2° ciclo:
 46
```

hypervolume metric?

Conclusioni

Framework for optimization using AI was provided

As result the Pareto front was enlarged when compared to original dataset's Pareto

Better result would have been to find a new Pareto front that dominates the original dataset's Pareto

Tons of ways to make the framework better

initializing optimization algorithm with surrogate's predictions on dataset
compare different optimization techniques (scalarization vs multiobjective)
compare different optimization algorithms
couple CFD with optimization algorithm directly

In the end though, I managed to satisfy the customer :)

```
Confronto tra il punto selezionato nella prima fase e quello nella seconda:
target1  variazione %:  -2.56
target2  variazione %:  0.02
target3  variazione %:  -2.53
```