Inexact Restoration method for solving Hinge loss problems

Master thesis

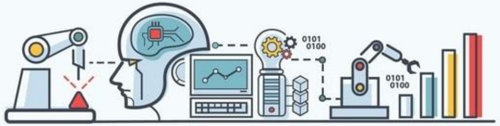
Hunor Tot-Bagi

Agenda

- Motivation
- Machine learning
- The Hinge loss function $f(x)$
- The algorithm
- Numerical results

Motivation

- The non-smooth Hinge loss function
- Second order derivatives
- Big data & sample size
- Inexact restoration



MACHINE LEARNING

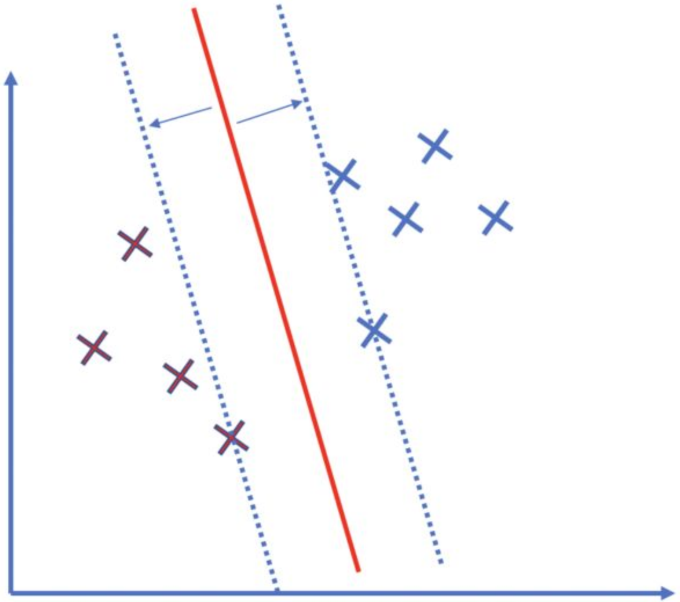
- L2 - regularized binary hinge loss

- Hinge loss for anomaly detection

L2 - Regularized binary hinge loss

- Width of the margin $\frac{2}{||x||}$
- If we want to maximize it, we need to minimize $||x||$
- Optimization problem

$$\min_x \frac{1}{2}||x||^2, \text{ subject to } z_i x^T \omega_i \geq 1,\ i = 1, \ldots, N$$

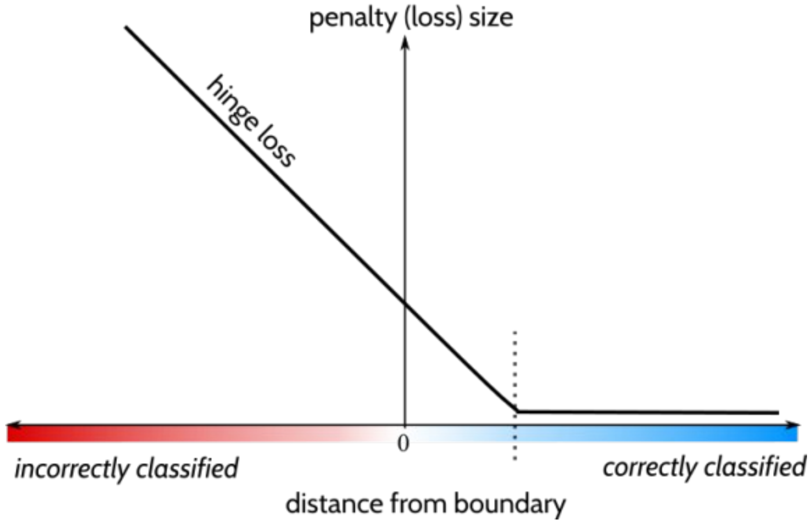


Hyperplanes in $\mathbb{R}^2$

L2 - Regularized binary hinge loss

$$f(x) := \frac{\lambda}{2}||x||^2 + \frac{1}{N}\sum_{i=1}^{N} l(\omega_i, z_i, x)$$

$$f_{N_k}(x) = \frac{\lambda}{2}||x||^2 + \frac{1}{N_k}\sum_{i=1}^{N_k} \max\left(0, 1 - z_i x^T \omega_i\right)$$



Hinge loss function

Hinge loss for anomaly detection

$$\min_{x,r} \frac{1}{2}||x||^2 - r, \quad \text{subject to } \; x^T\omega_i \geq |r|, i = 1, \ldots, N$$

The samples ω_i for which $x^T w_i \leq \; |r|$ is true, will be considered as anomalies

$$\min_{x,r} f(x,r) = \min_{x,r} \left(\frac{\lambda||x||^2}{2} - \lambda r + \frac{1}{N} \sum_{i=1}^{N} \max\{0, r - x^T w_i\} \right)$$

The algorithm

- BFGS update
- Descent direction
- Inexact Restoration approach

BFGS update

- Broyden-Fletcher-Goldfarb-Shanno
- Iterative method
- Has an advantage if the function is convex
- $B_{k+1} = (I - \rho_k s_k y_k^T) B_k (I - \rho_k y_k s_k^T) + \rho_k s_k s_k^T$
- We skip the update if $s_k^T y_k \geq 10^{-4} ||y_k||^2$

Descent direction algorithm

- What is a descent direction

$$g^T p < 0 \text{ for all } g \in \partial\psi(x)$$

- Minimize the pseudo-quadratic model

$$Y(p) = \frac{1}{2} p^T B^{-1} p + \sup_{g \in \partial\phi(x)} g^T p$$

- For nonsmooth function

$$p_k = -B_k g_k$$

Inexact restoration algorithm

- S1 - Restoration phase

$$\text{Find } \tilde{N}_{k+1} \geq N_k \text{ such that } h\left(\tilde{N}_{k+1}\right) \leq rh(N_k)$$

- S2 - Updating parameter theta

- S3 - Optimization phase
    a. Calculating new descent direction
    b. Searching for step size with backtracking

Inexact restoration algorithm

- S4 - Update for solution vector

  Set $s_k = \alpha_k p_k$ and $x_{k+1} = x_k + s_k$

- S5 - Choosing the next subgradient and update the next BFGS

- S6 - Increase the counter and go to S1

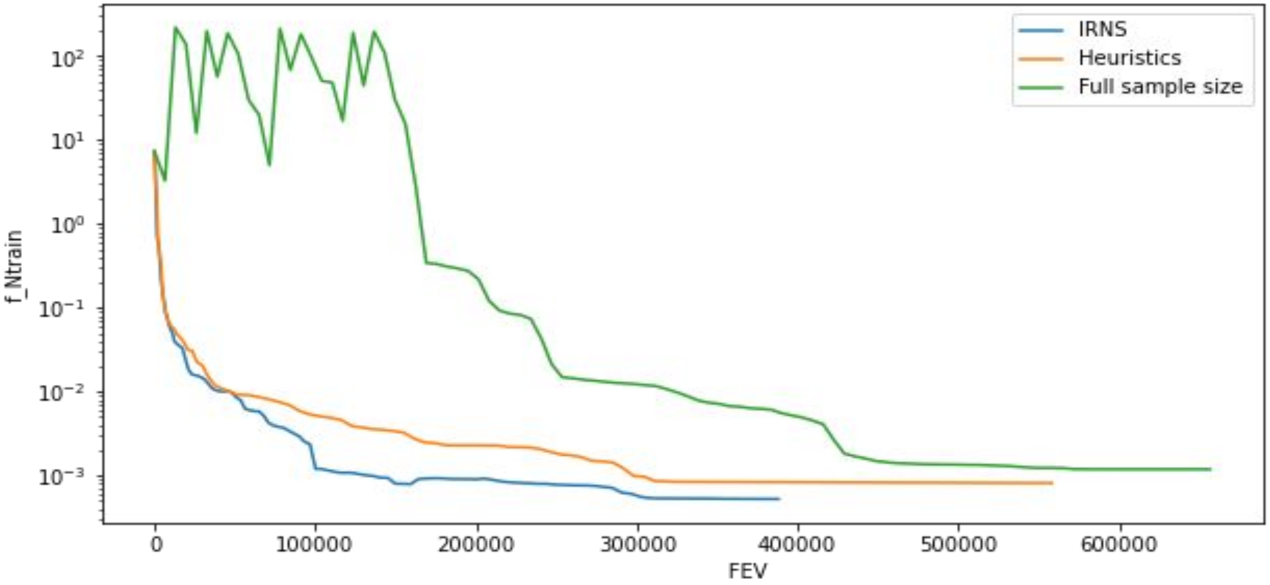
Numerical results

Properties of the dataset used in the experiments

	Dataset	N	n	N_{train}	N_{test}	Max_{FEV}
1	Splice	3175	60	2540	635	10^5
2	Mushrooms	8124	112	6500	1624	10^5
3	Adult	32561	123	26049	6512	10^6

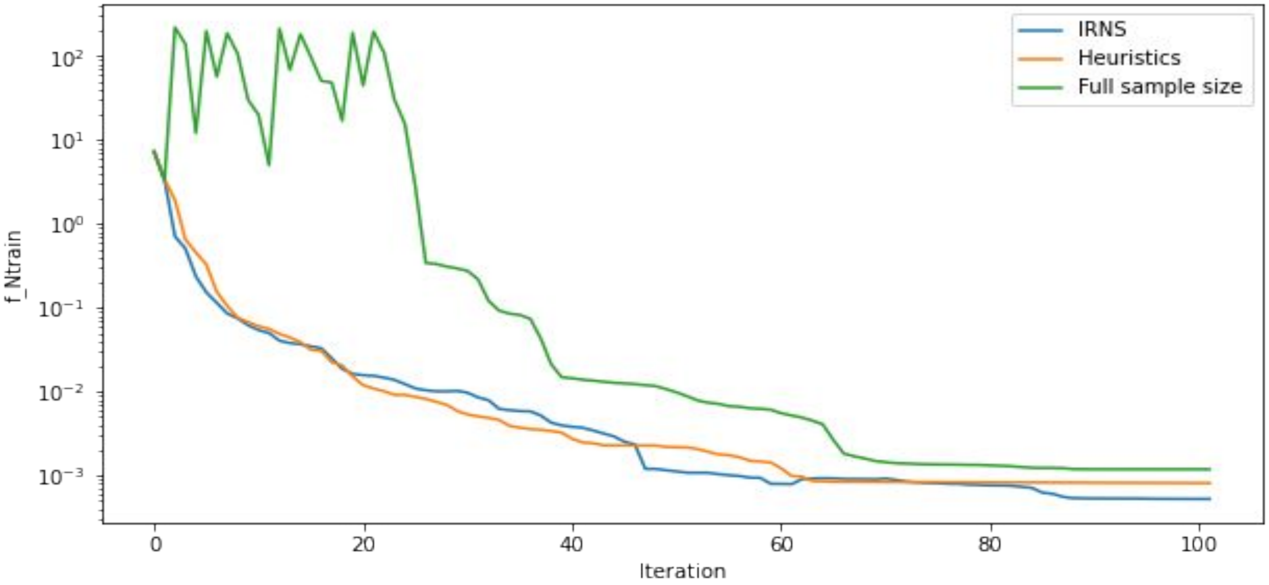
Mushroom dataset

- Edible (1) or not (-1)
- Training loss versus FEV



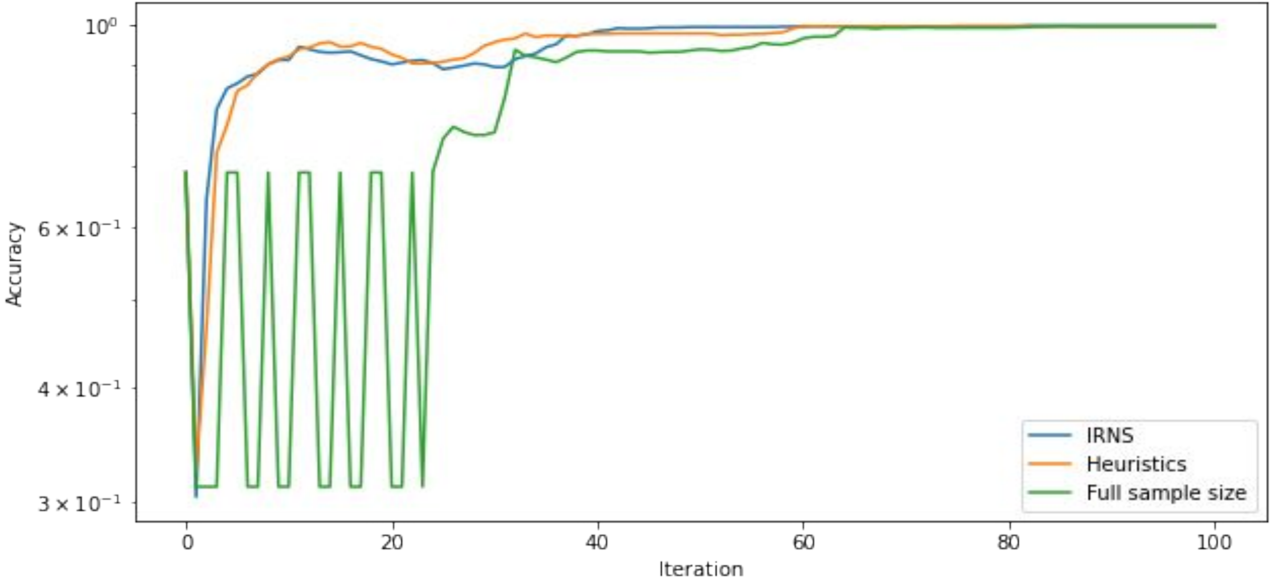
Mushroom dataset

- Training loss versus iteration



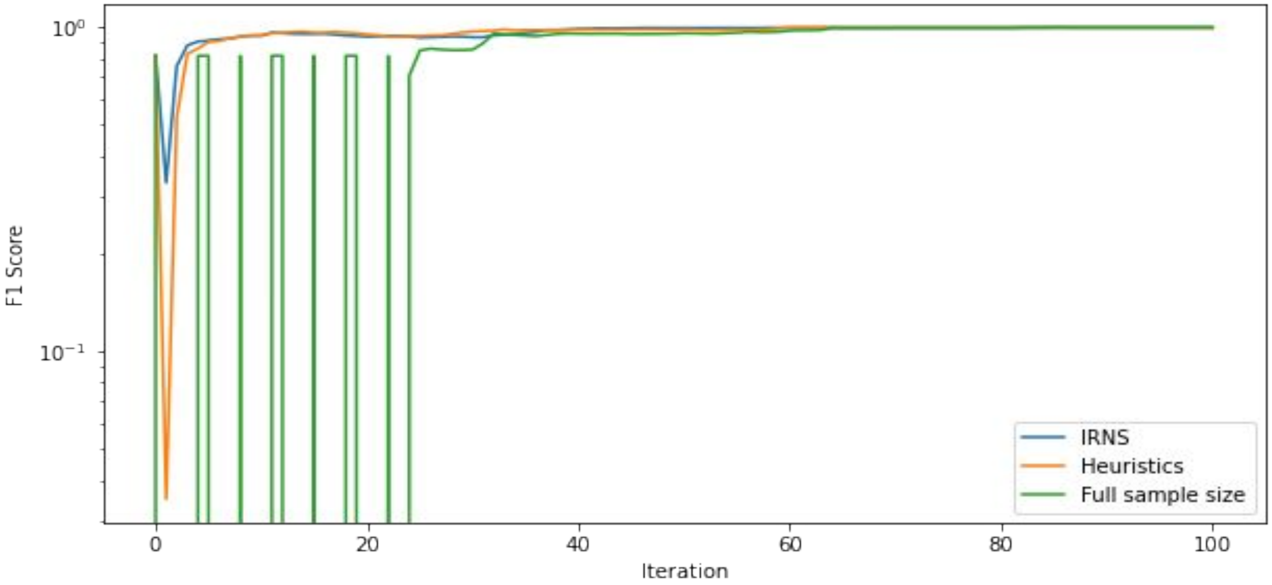
Mushroom dataset

- Accuracy versus iteration

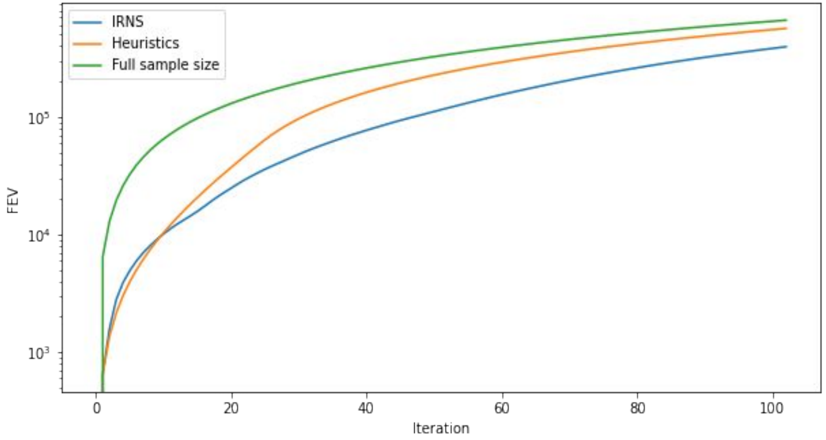


Mushroom dataset

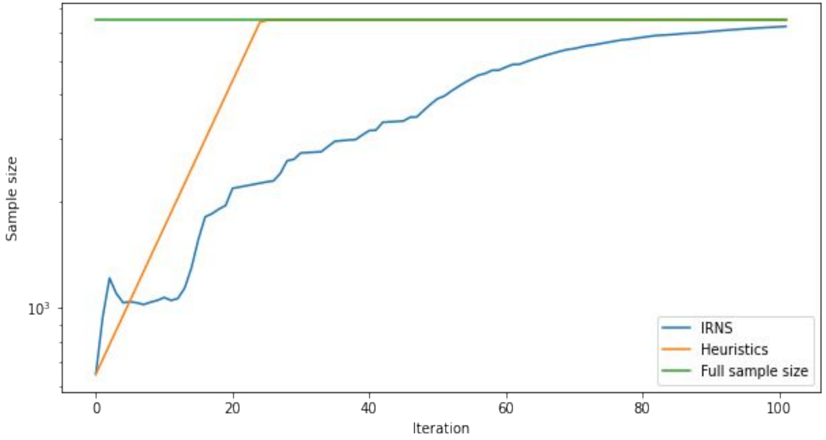
- F1 Score versus iteration



Mushroom dataset



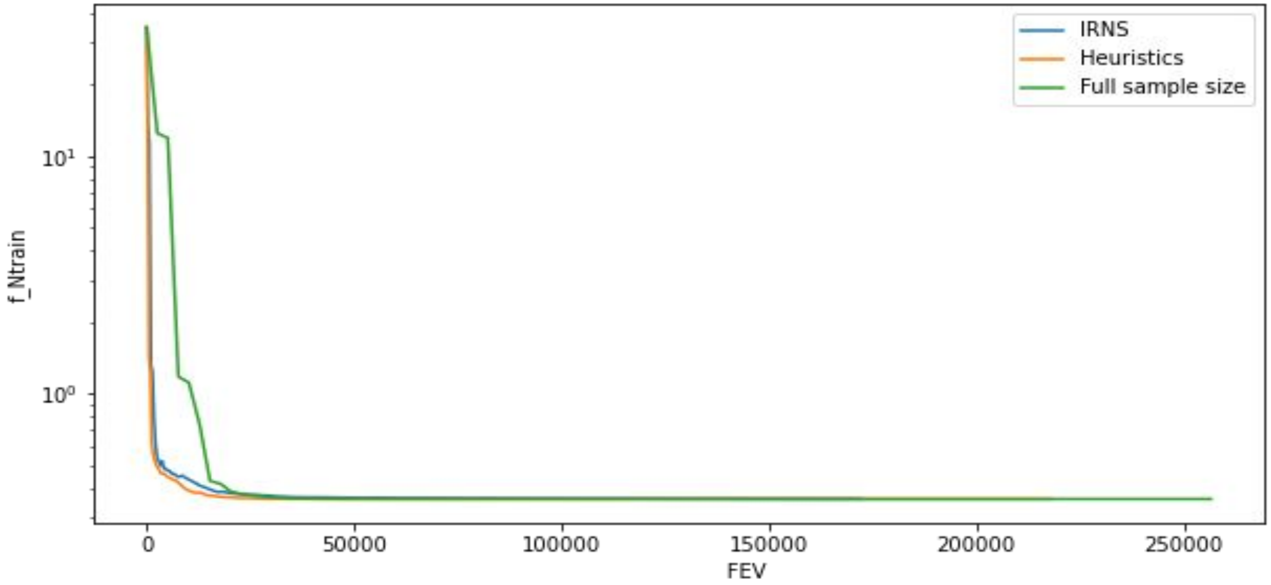
FEV versus iteration



Sample size representation

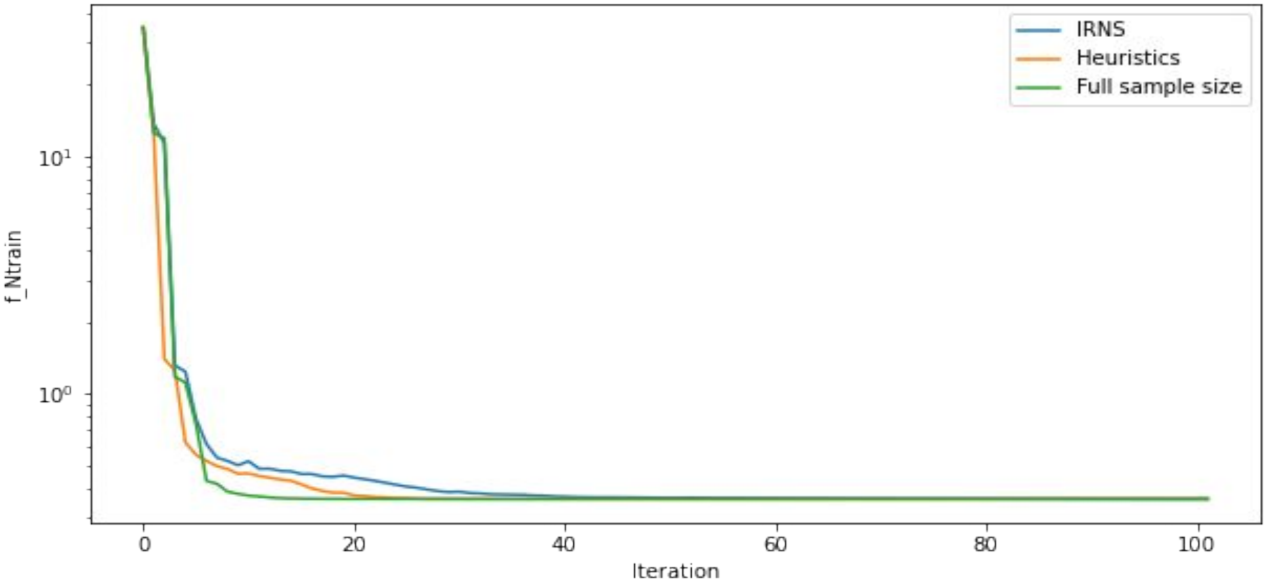
Splice dataset

- Training loss versus FEV



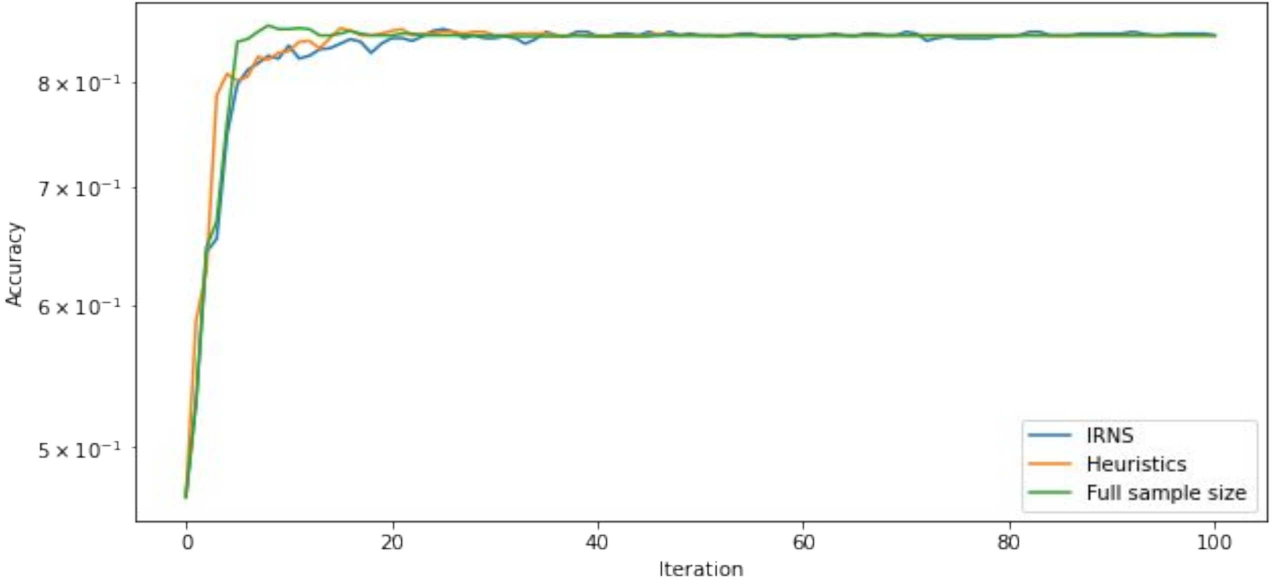
Splice dataset

- Training loss versus iteration



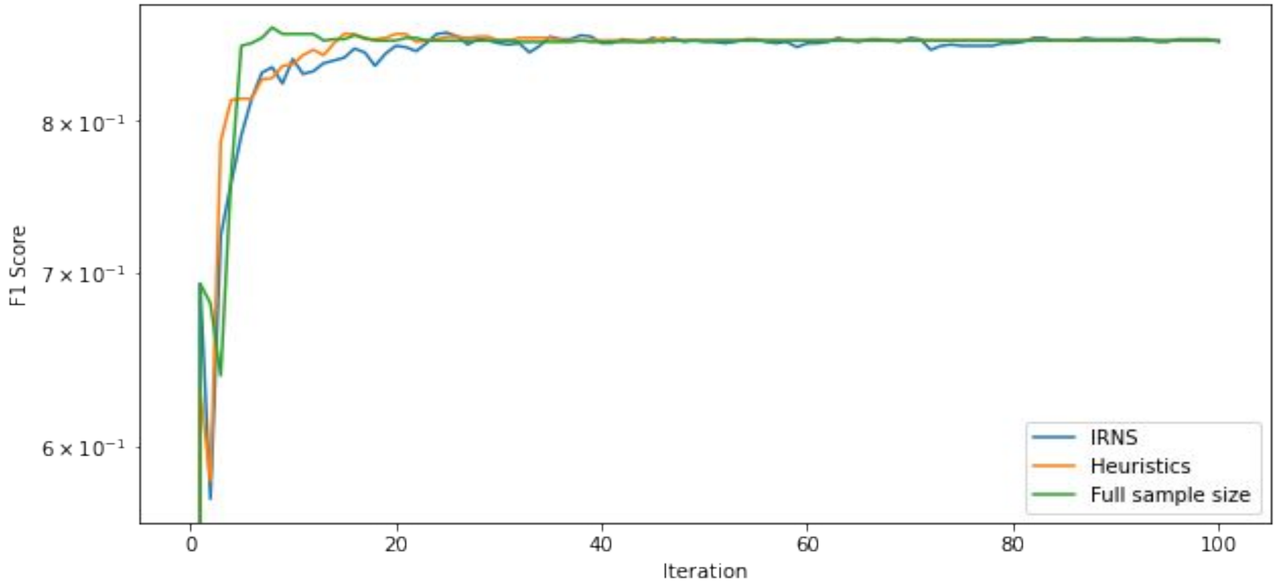
Splice dataset

- Accuracy versus iteration

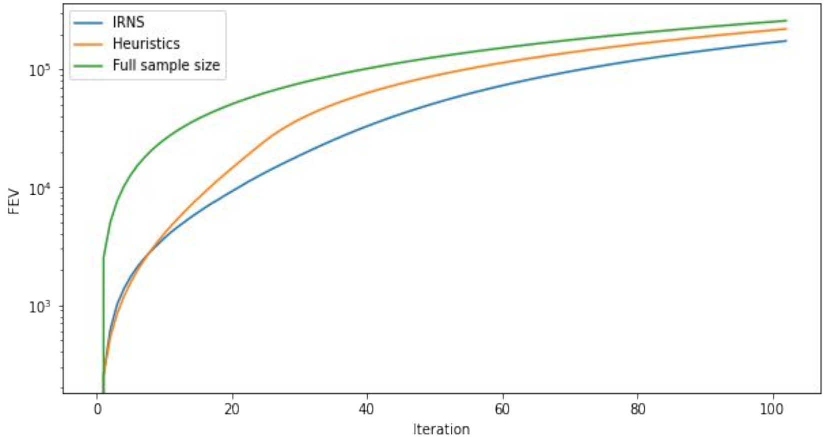


Splice dataset

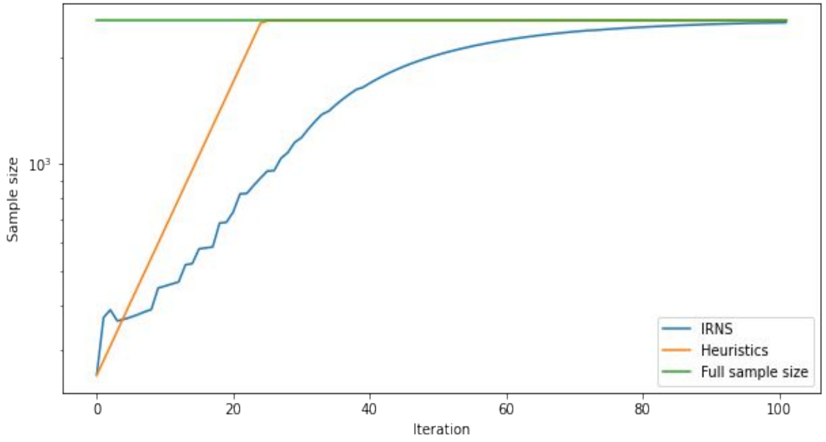
- F1 Score versus iteration



Splice dataset



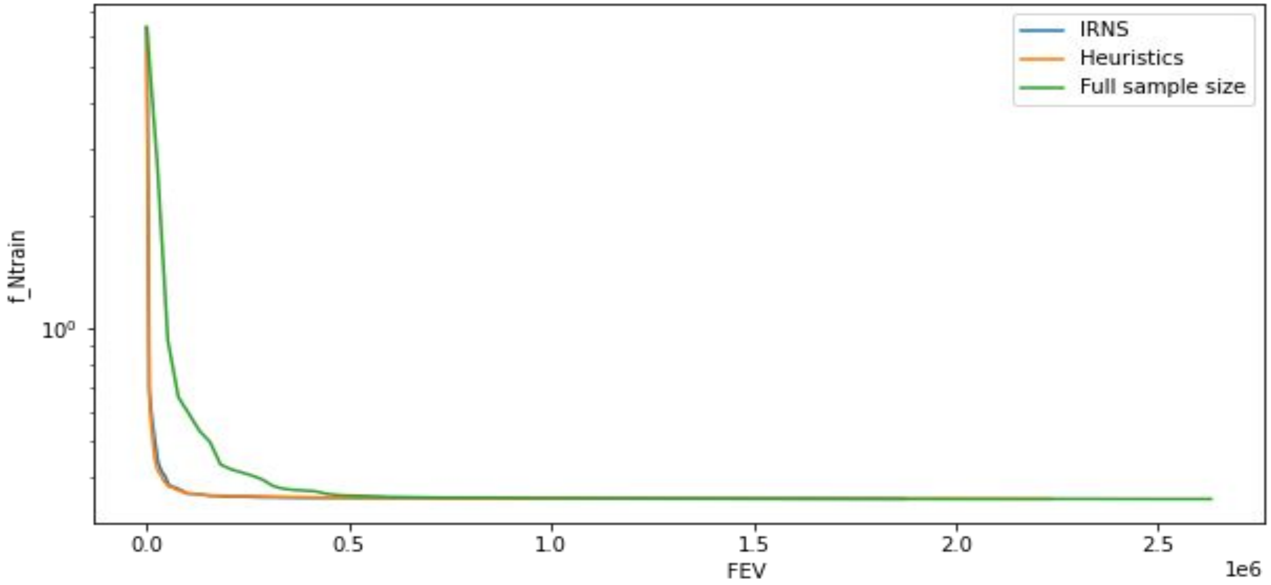
FEV versus iteration



Sample size representation

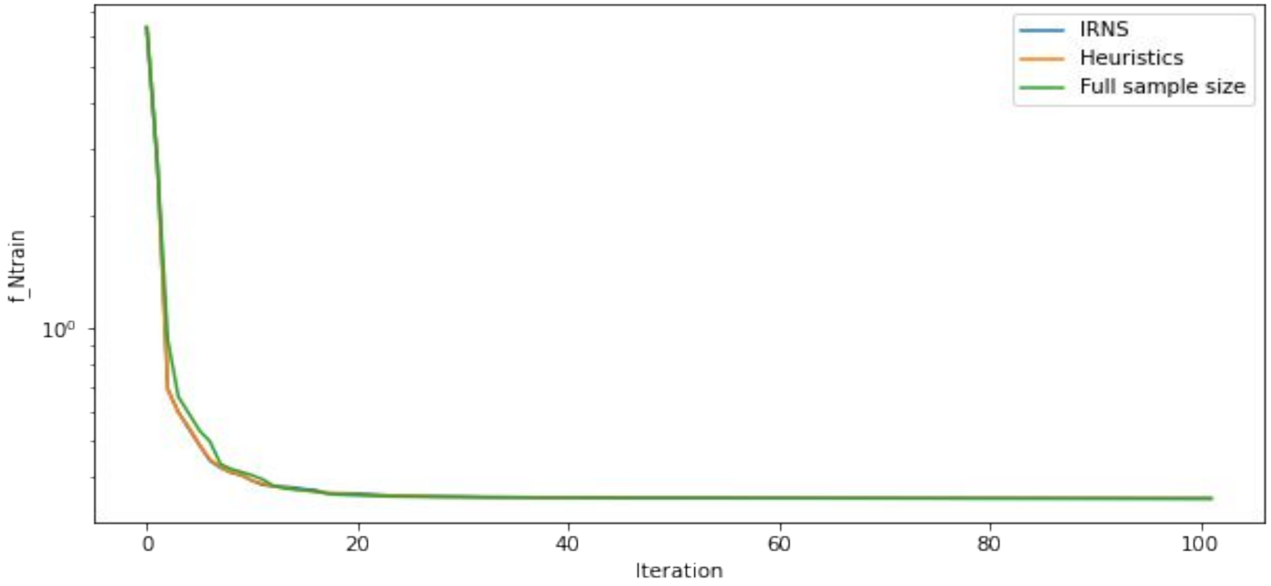
Adult dataset

- Training loss versus FEV

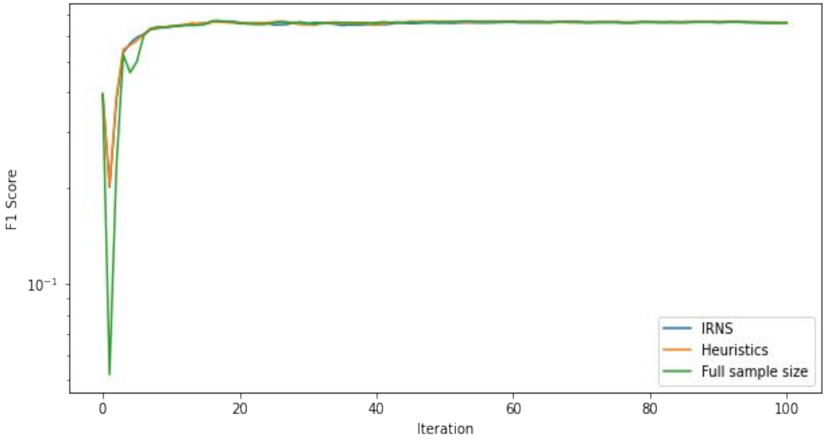


Adult dataset

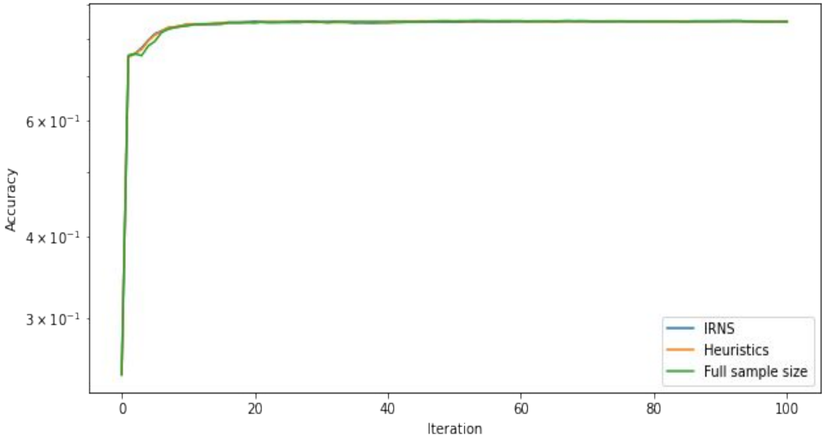
- Training loss versus iteration



Adult dataset

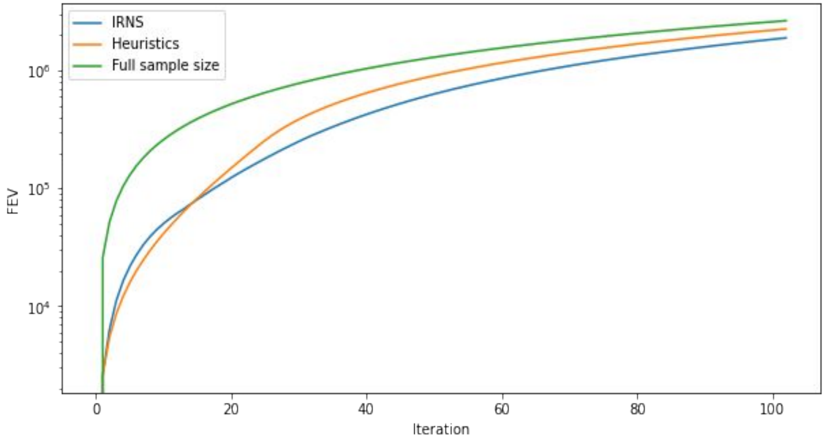


F1 Score versus iteration

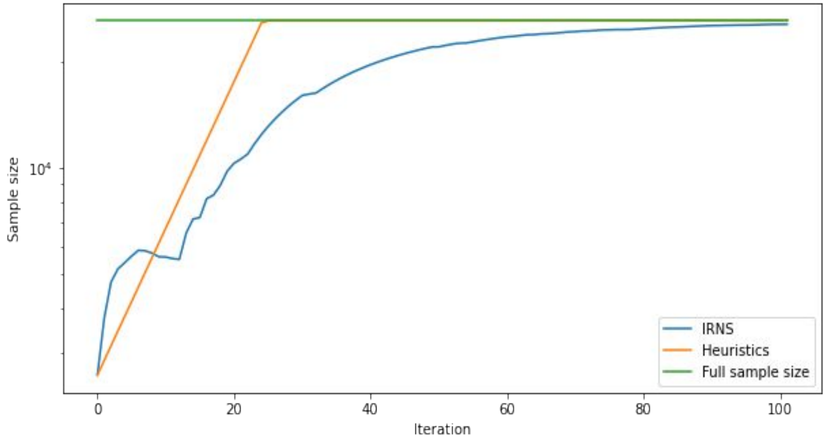


Accuracy versus iteration

Adult dataset



FEV versus iteration



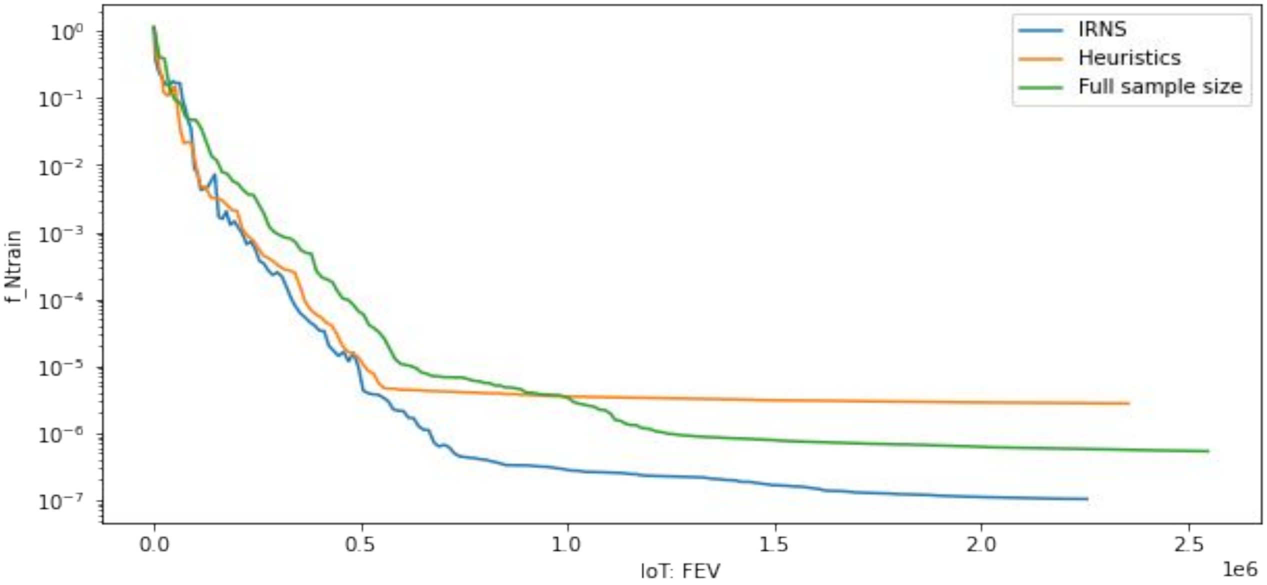
Sample size representation

IoT dataset

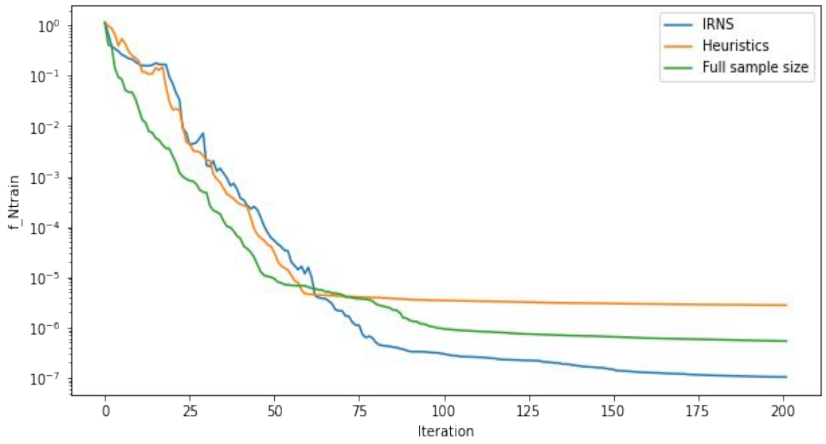
- H2020 C4IIoT project - Cyber security 4.0
- Data was generated using NB-IoT edge nodes
- Box-shaped container inside a transport vehicle in Novi Sad
- 12678 samples for train and 1571 samples for test
- Timestamps, 13 attributes

IoT dataset

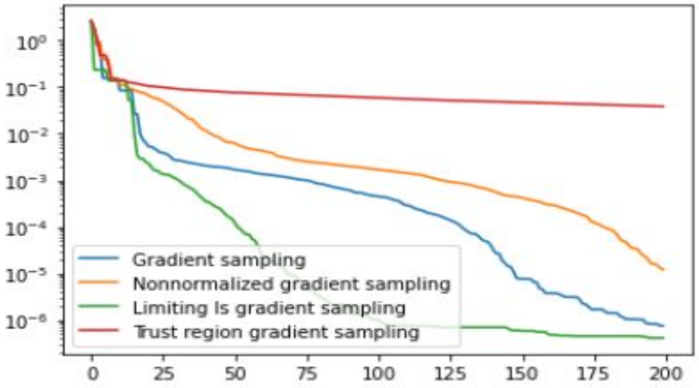
- Training loss versus FEV



IoT dataset



Training loss versus iteration



Representing the training loss of the gradient sampling method

IoT dataset

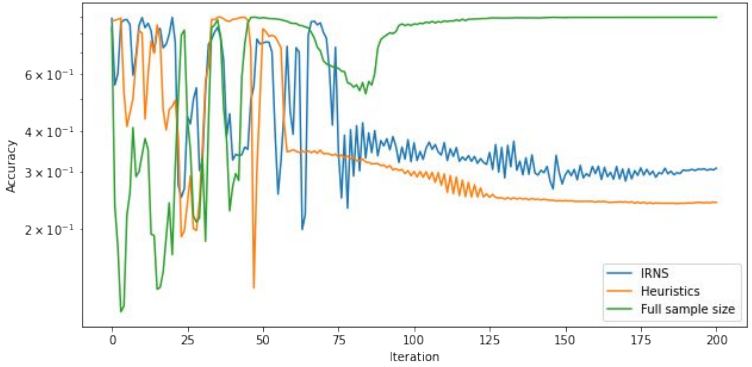
Confusion matrix

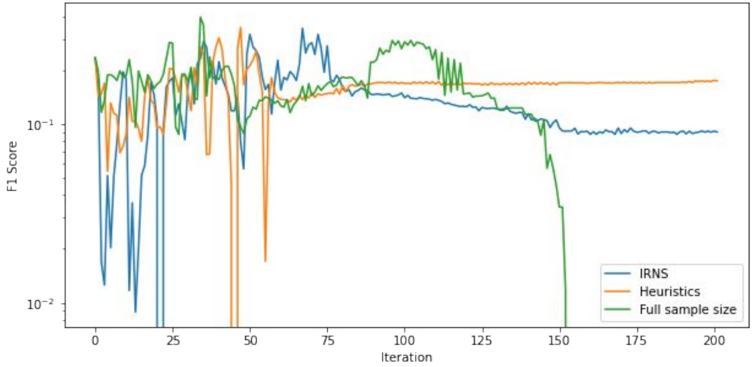
	IRNS	Heuristics	Ful sample size
TP	95	118	88
FP	336	395	189
FN	71	48	178
TN	1069	1010	1216

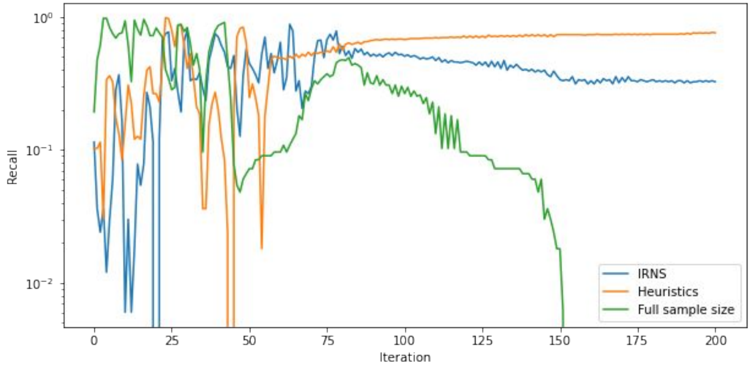
Classification results

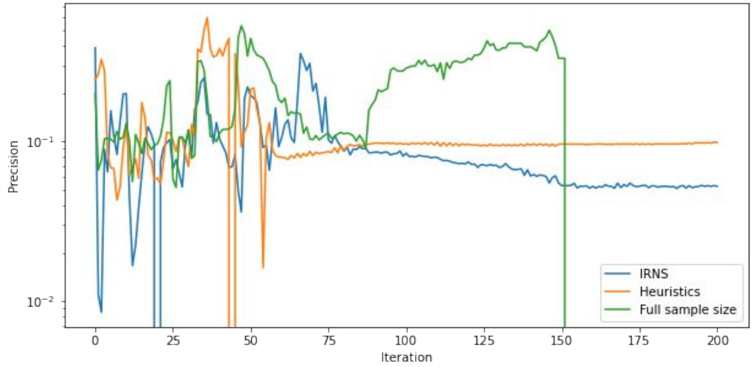
	IRNS	Heuristics	Full sample size
Accuracy	0.741	0.718	0.83
Precision	0.220	0.230	0.318
Recall	0.572	0.711	0.53
F1 score	0.318	0.348	0.397

IoT dataset - Classification results









Conclusions

- Advantages of IRNS in terms of FEV
- Second order information
- Better vicinity of the solution than Gradient Sampling
- Classification results