Wrocław University of Science and Technology

Building transparent classification models

dr inż. Bogdan Gulowaty

Artificial Intelligence in Research and Applications Seminar (AIRA)

29.05.2025

unite! University Network for Innovation, Technology and Engineering

HR EXCELLENCE IN RESEARCH

Wroclaw University of Science and Technology

Outline

Outline

Wroclaw University of Science and Technology

Hypothesis

For a given classification task, it is possible to build such a transparent or explainable model whose quality is not worse than a similarly applied black-box model

Objectives

- Development of a novel ensemble "glass-box" model extraction method (like[1])
- Development of novel transparent, "glass-box" models
- Utilization of data complexity metrics in developing novel transparent classification method
- Experimental evaluation of proposed algorithms
- Designing and implementing a programming library

[1] Omer Sagi and Lior Rokach. "Explainable decision forest: Transforming a decision forest into an interpretable tree". en. In: *Information Fusion* (2020).

Examples of not interpretable models

- Neural networks
- Ensemble models (Random forest)
- Non-linear Support vector machines

Examples of interpretable models

- Decision trees
- Rules lists
- Linear regression
- Nearest neighbors model

XAI taxonomies

- Intrinsic/Post-hoc
- Global/Local
- Model agnostic/Model specific

XAI taxonomies

- Intrinsic/**Post-hoc**
- **Global**/Local
- Model agnostic/**Model specific**

Outline

Experimental evaluation setup

1. Fixed number of binary datasets (16)
2. Fixed number of complexity metrics (22) (from Lorena's paper[2])
3. 5x2 cross-validation
4. Statistical analysis with the sign test
5. Base classifiers were pretrained on hold-off datasets to find optimal parameters

[2] Ana C. Lorena et al. "How Complex Is Your Classification Problem? A Survey on Measuring Classification Complexity". In: *ACM Comput. Surv.* (2019).

Wroclaw University of Science and Technology

Used datasets

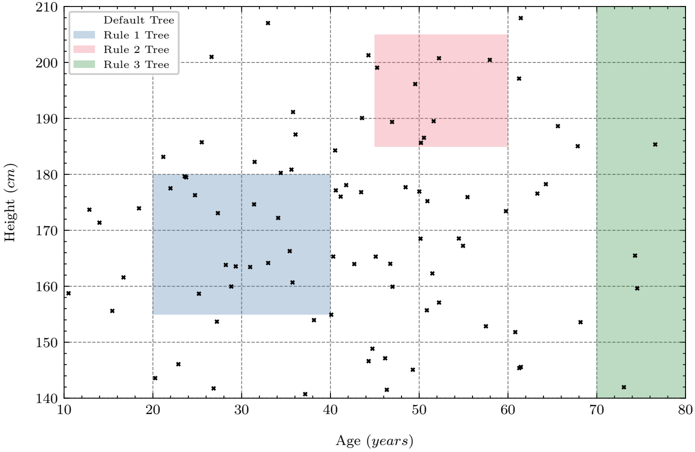
name	#instances	#features	#integer	#real	#nominal
appendicitis	106	7	0	7	0
australian	690	14	5	3	6
bands	365	19	6	13	0
breast	277	9	0	0	9
bupa	345	6	5	1	0
crx	653	15	3	3	9
haberman	306	3	3	0	0
heart	270	13	12	1	0
hepatitis	80	19	17	2	0
housevotes	232	16	0	0	16
ionosphere	351	33	1	32	0
mammographic	830	5	5	0	0
monk-2	432	6	6	0	0
saheart	462	9	3	5	1
tic-tac-toe	958	9	0	0	9
wisconsin	683	9	9	0	0

Complexity metrics

metric	low	medium	high
F1	< 0.49	$0.49 - 0.757$	> 0.757
F2	< 0.215	$0.215 - 0.659$	> 0.659
F3	< 0.824	$0.824 - 0.948$	> 0.948
F4	< 0.629	$0.629 - 0.896$	> 0.896
F1V	< 0.145	$0.145 - 0.417$	> 0.417
C1	< 0.009	$0.009 - 0.069$	> 0.069
C2	< 0.024	$0.024 - 0.172$	> 0.172
L1	< 0.179	$0.179 - 0.266$	> 0.266
L2	< 0.194	$0.194 - 0.297$	> 0.297
L3	< 0.14	$0.14 - 0.301$	> 0.301
N1	< 0.111	$0.111 - 0.195$	> 0.195
N2	< 0.493	$0.493 - 0.54$	> 0.54
N3	< 0.219	$0.219 - 0.377$	> 0.377
N4	< 0.155	$0.155 - 0.25$	> 0.25
T1	< 0.588	$0.588 - 0.822$	> 0.822
T2	< 0.035	$0.035 - 0.096$	> 0.096
T3	< 0.022	$0.022 - 0.043$	> 0.043
T4	< 0.421	$0.421 - 0.667$	> 0.667
clsCoef	< 0.081	$0.081 - 0.368$	> 0.368
hubs	< 0.505	$0.505 - 0.738$	> 0.738
density	< 0.718	$0.718 - 0.879$	> 0.879
LSC	< 0.96	$0.96 - 0.986$	> 0.986
size	< 306.0	$306 - 462.0$	> 462
features_count	< 9.0	$9 - 14$	> 14

Outline

NOTE competence areas visualization



NOTE algorithm (simplified)

1. Extract trees from random forest
2. Extract rules from the trees
3. Check each rule pair for overlapping
4. Model rules as a graph, where each node is rule and edge (relation) means that rules do not overlap with each other
5. Using evaluation metric (e.g., accuracy) and cross-validation find best performing clique (and set of rules)
6. Build final model by training decision tree in boundaries set by each rule + one default tree

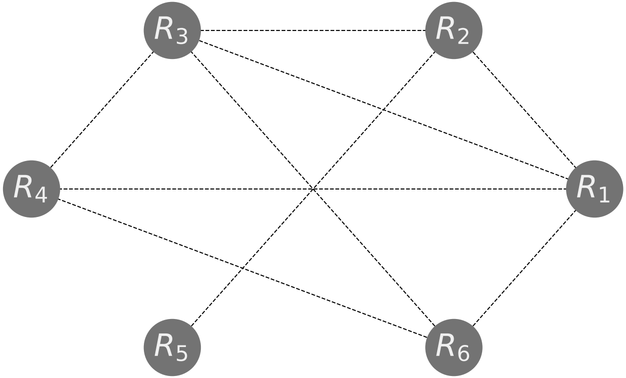
NOTE algorithm (simplified)

1. Extract trees from random forest
2. Extract rules from the trees
3. Check each rule pair for overlapping
4. Model rules as a graph, where each node is rule and edge (relation) means that rules do not overlap with each other
5. Using evaluation metric (e.g., accuracy) and cross-validation find best performing clique (and set of rules)
6. Build final model by training decision tree in boundaries set by each rule + one default tree

NOTE algorithm (simplified)

1. Extract trees from random forest
2. Extract rules from the trees
3. Check each rule pair for overlapping
4. Model rules as a graph, where each node is rule and edge (relation) means that rules do not overlap with each other
5. Using evaluation metric (e.g., accuracy) and cross-validation find best performing clique (and set of rules)
6. Build final model by training decision tree in boundaries set by each rule + one default tree

NOTE rules distillation visualization



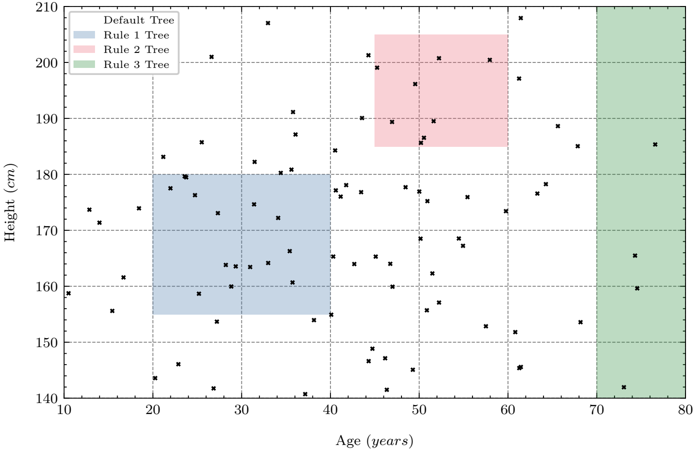
NOTE algorithm (simplified)

1. Extract trees from random forest
2. Extract rules from the trees
3. Check each rule pair for overlapping
4. Model rules as a graph, where each node is rule and edge (relation) means that rules do not overlap with each other
5. Using evaluation metric (e.g., accuracy) and cross-validation find best performing clique (and set of rules)
6. Build final model by training decision tree in boundaries set by each rule + one default tree

NOTE algorithm (simplified)

1. Extract trees from random forest
2. Extract rules from the trees
3. Check each rule pair for overlapping
4. Model rules as a graph, where each node is rule and edge (relation) means that rules do not overlap with each other
5. Using evaluation metric (e.g., accuracy) and cross-validation find best performing clique (and set of rules)
6. Build final model by training decision tree in boundaries set by each rule + one default tree

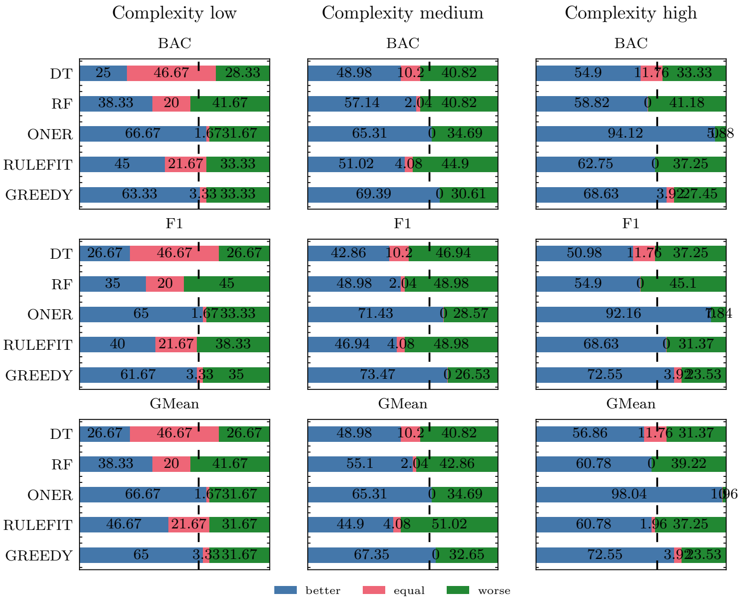
NOTE competence areas visualization



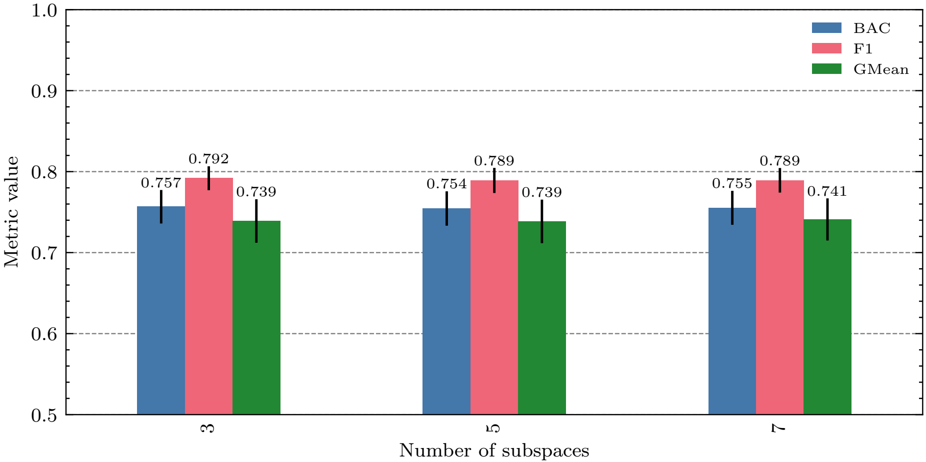
NOTE experiments - tested parameters

- Random Forest sizes of 3, 5, 7
- Number of subspaces (clique size) of 3, 5, 7
- Decision Tree as base classifier

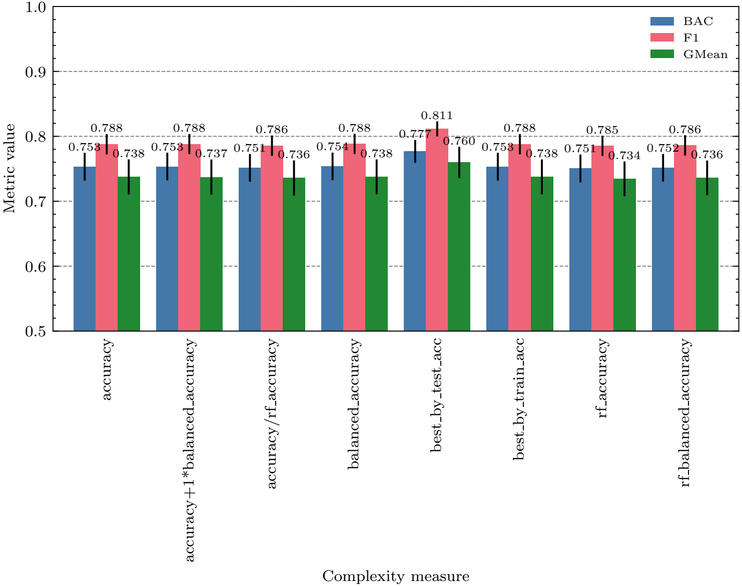
NOTE - overall performance across complexities



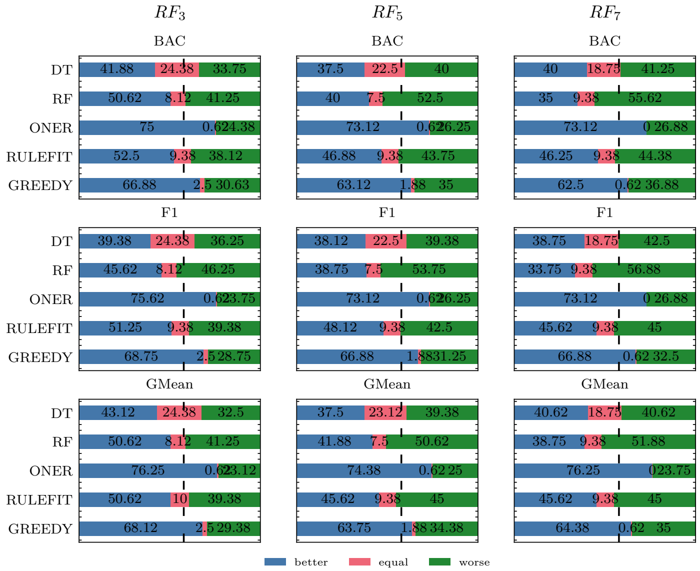
NOTE - dependencies on the subspace number



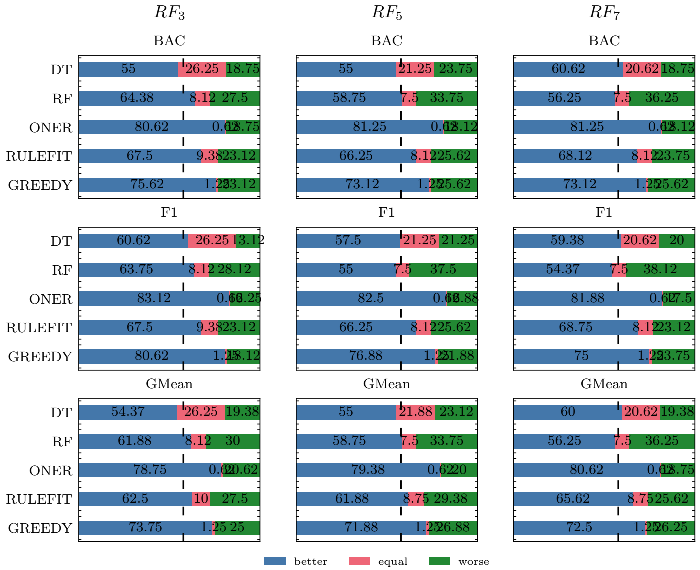
NOTE - dependencies on the evaluation metric



NOTE results - different forest size



NOTE results - different forest size (test)



NOTE - summary

- NOTE creates an explainable tree ensemble that can be competitive to explain Random Forest
- It shows better performance when presented with more complex data
- There is space for finding better clique evaluation metrics, as shown by the test data

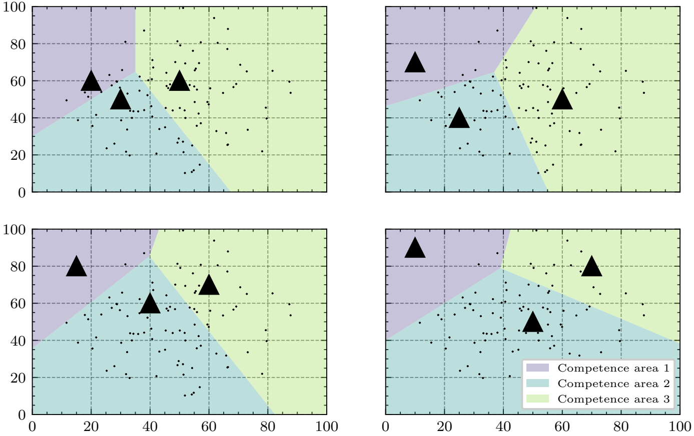
Wroclaw University of Science and Technology

Outline

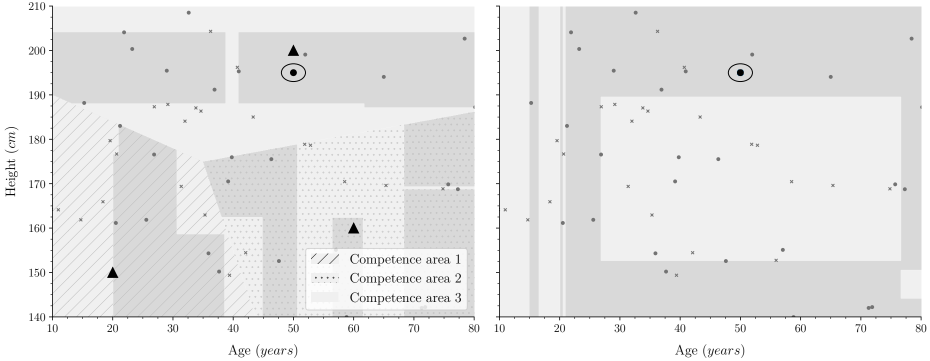
Optimal Centroids algorithm

- Use Genetic Algorithm to find the best position of centroids.
  In each iteration: Build and evaluate a Decision Tree using samples that fall into a given centroid area
- Build final model after optimization procedure ceases

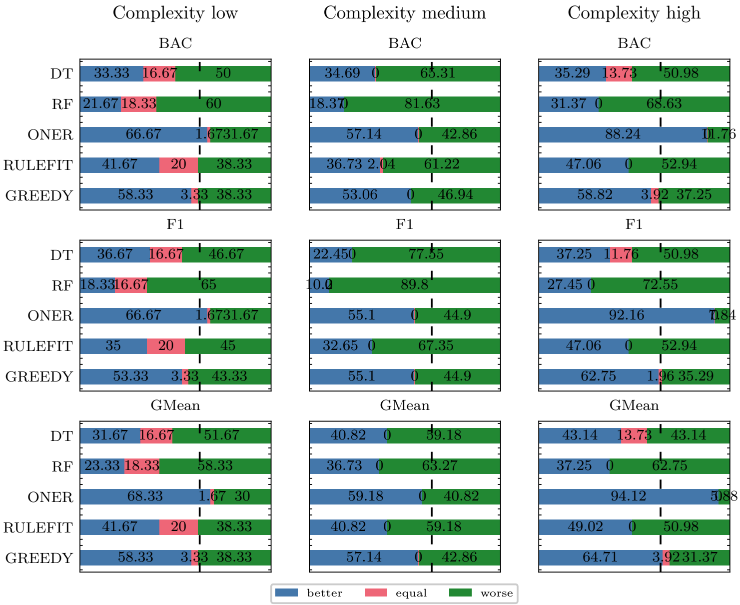
Optimal Centroid learning procedure



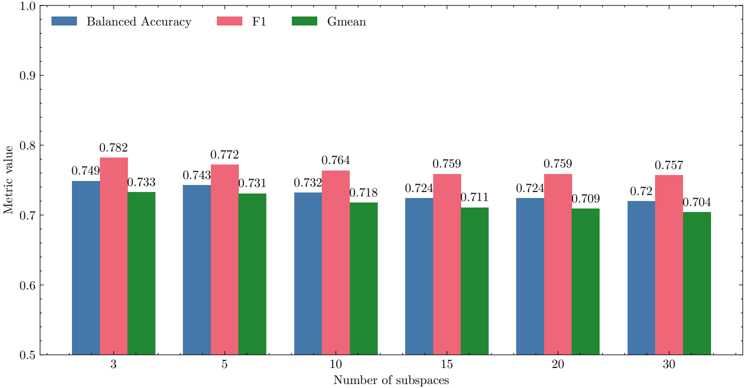
DT vs OC decision boundaries



OC results - overall performance across complexities



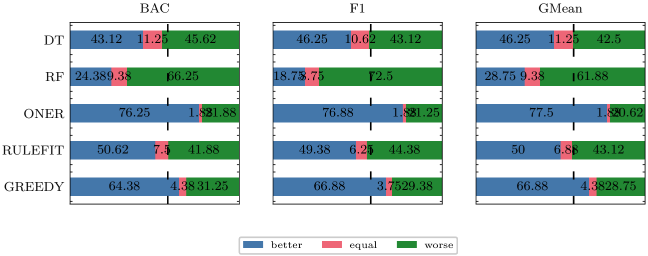
OC results - overall performance for different number of subspaces

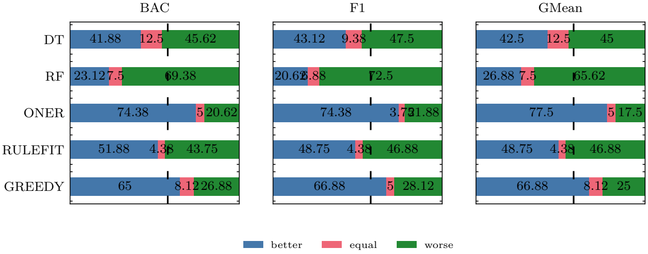


Two "XAI" variations

- For each subspace - select a tree from within the explained Random Forest using the additional GA feature
- Pick the best tree from Random Forest for each centroid

OC XAI variations - overall performance



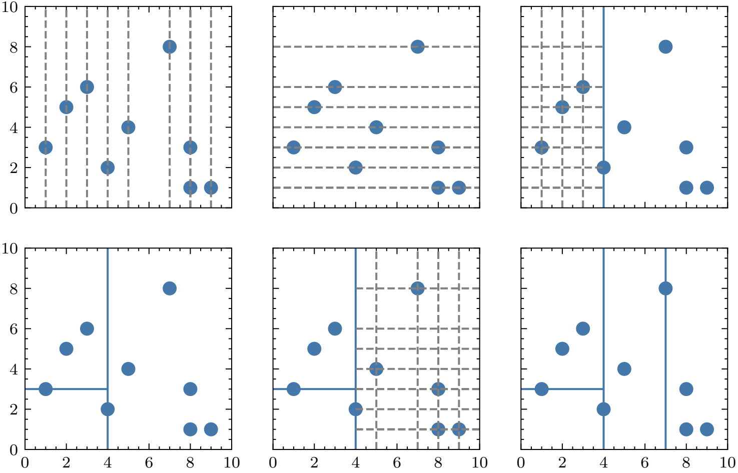


Optimal Centroids - summary

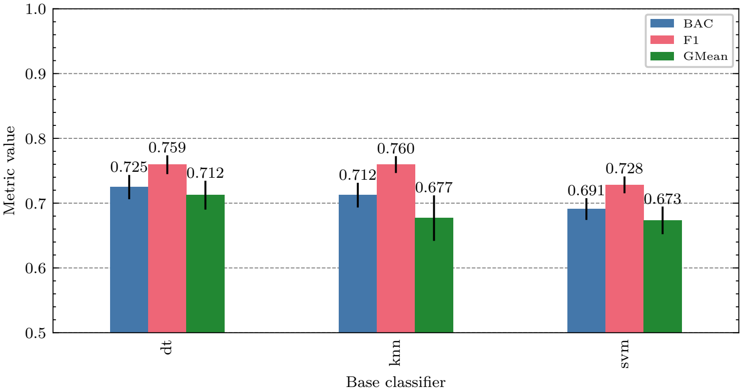
- Optimal Centroid uses Genetic Algorithm for finding optimal distribution of decision spaces for tree ensemble
- It performs best for datasets of low and high complexity
- Both XAI variations perform roughly the same
- The more centroids are picked, the worse the accuracy

Outline

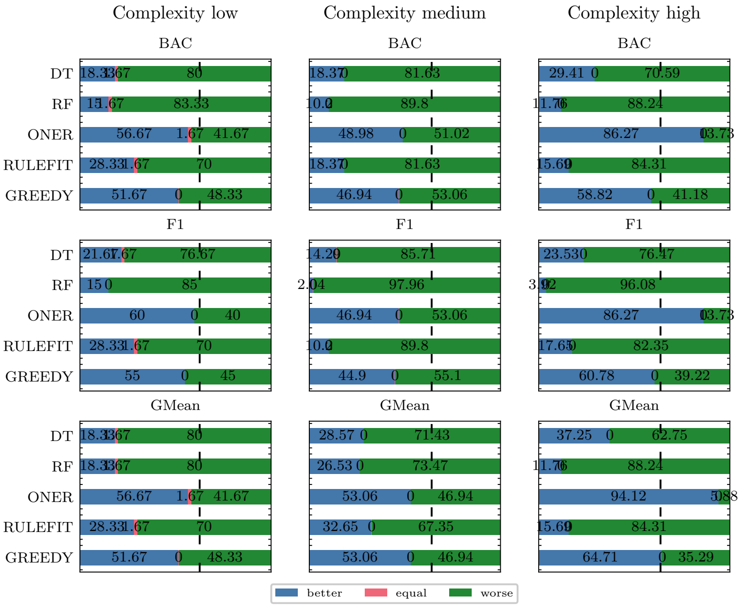
Quad split learning procedure



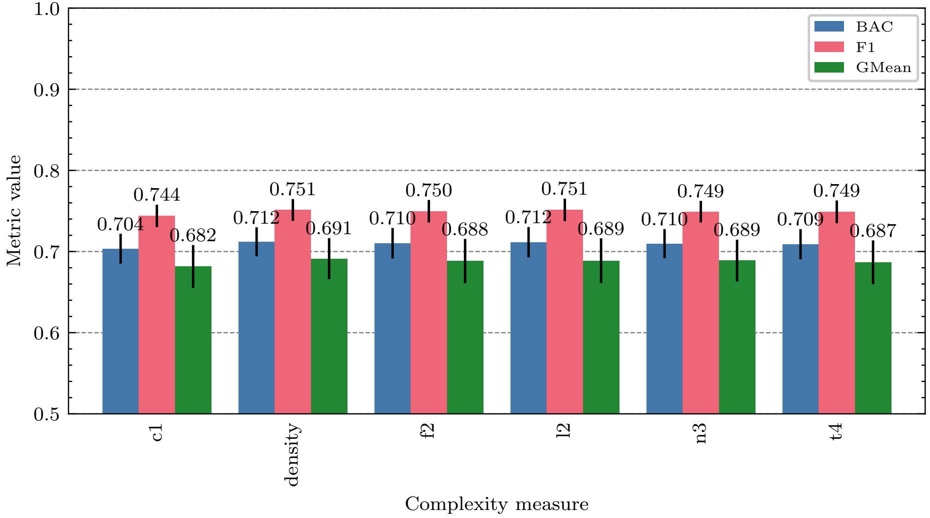
Quad split learning results - different base classifiers



Quad split results



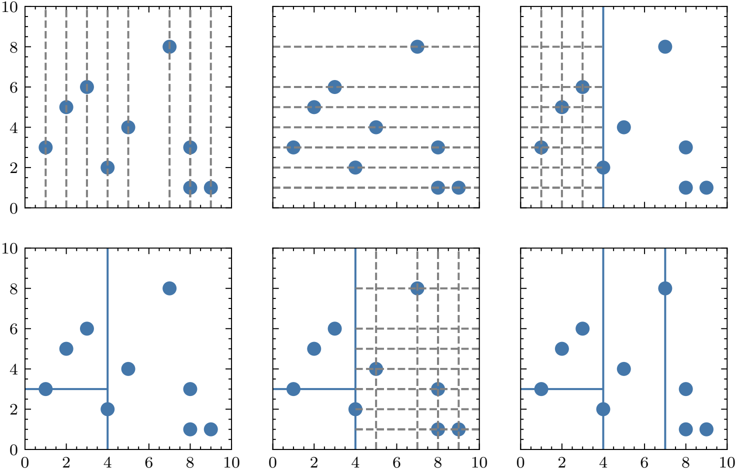
Quad split results - different complexity metrics



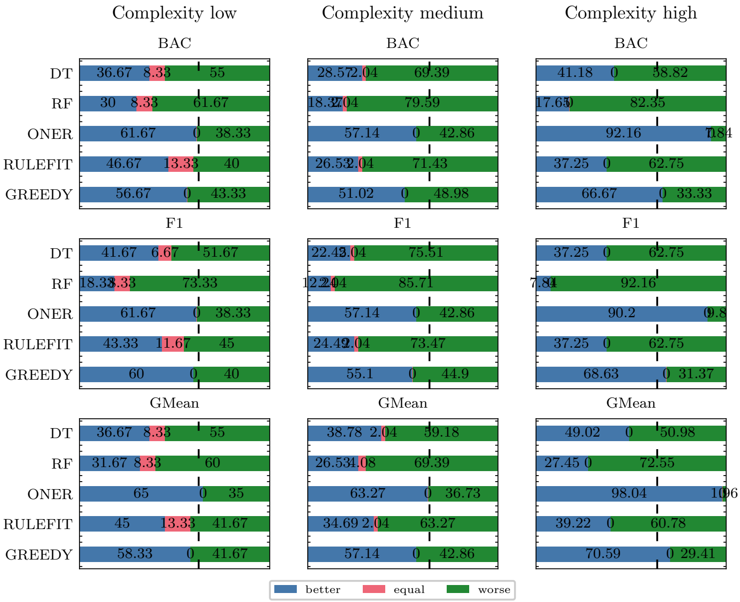
Quad Split - two "XAI" variations

- Pick the best tree from the tree ensemble
- Train on RF output data

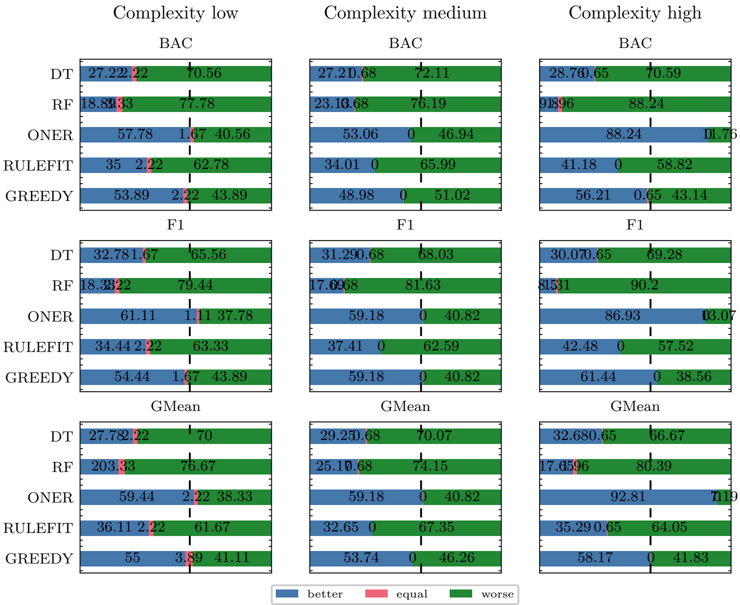
Quad Split - two "XAI" variations



QS with picking trees - overall performance



QS with training on RF's output - overall performance



Quad Split summary

- Quad Split algorithm can act as a viable solution for explaining Random Forest by picking the best trees from the ensemble
- It behaves best with low-complexity datasets
- Quad Split is an inherently transparent model

Outline

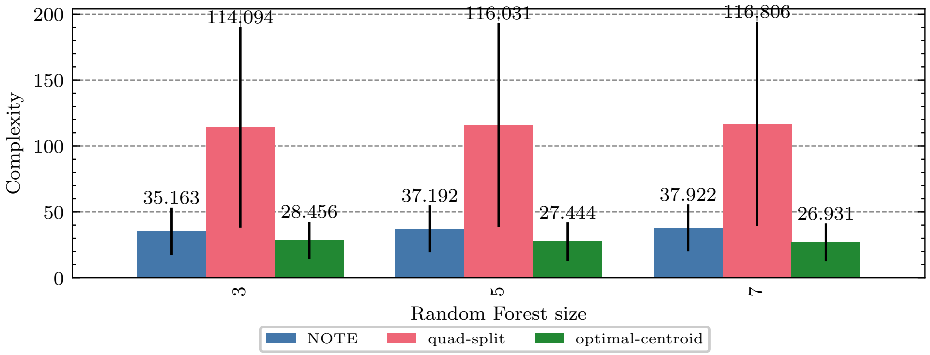
Models compared

1. Optimal Centroids with tree selection (additional GA individual parameter)
2. Quad Split with picking the best tree from RF
3. NOTE

Measuring model's complexity

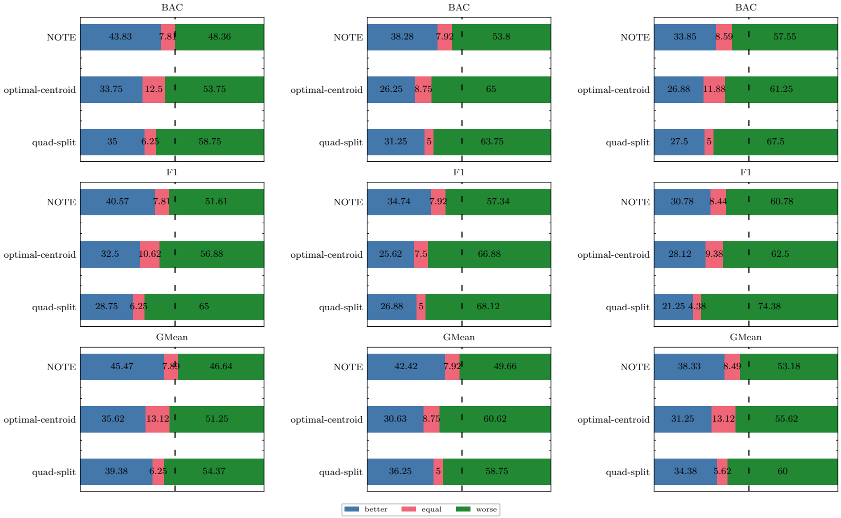
model	complexity formula
Quad Split	$\sum^{\text{trees}} \#$ of leaves $+ \#$ of rules
Optimal Centroids	$\sum^{\text{trees}} \#$ of leaves $+ \#$ of centroids
NOTE	$\sum^{\text{trees}} \#$ of leaves $+ \#$ of rules

Comparison of models' complexity



Performance against explained RF

RF_3 RF_5 RF_7



Wroclaw University of Science and Technology

Outline

Hypothesis

For a given classification task, it is possible to build such a transparent or explainable model whose quality is not worse than a similarly applied black-box model

Wroclaw University of Science and Technology

Objectives

- Development of a novel ensemble "glass-box" model extraction method ✔
- Development of novel transparent, "glass-box" models ✔
- Utilization of data complexity metrics in developing novel transparent classification method ✔
- Experimental evaluation of proposed algorithms ✔
- Designing and implementing a programming library ✔
  - https://github.com/bgulowaty/non-overlapping-rules-ensemble
  - https://github.com/bgulowaty/optimal-centroids
  - https://github.com/bgulowaty/quad-split
  - https://github.com/bgulowaty/ml-utils

Future works

1. Evaluate the stability of models
2. Evaluate the performance of proposed models against other tree ensemble methods, such as XGBoost CatBoost
3. Validate models' performance against other XAI methods, such as EBM, LORE, EXPLAN
4. Run experiments with human end-users/consumers of explanations

Future works

1. Evaluate the stability of models
2. Evaluate the performance of proposed models against other tree ensemble methods, such as XGBoost CatBoost
3. Validate models' performance against other XAI methods, such as EBM, LORE, EXPLAN
4. Run experiments with human end-users/consumers of explanations

Wroclaw University of Science and Technology

Future works

1. Evaluate the stability of models
2. Evaluate the performance of proposed models against other tree ensemble methods, such as XGBoost CatBoost
3. Validate models' performance against other XAI methods, such as EBM, LORE, EXPLAN
4. Run experiments with human end-users/consumers of explanations

Wrocław University of Science and Technology

Future works

1. Evaluate the stability of models
2. Evaluate the performance of proposed models against other tree ensemble methods, such as XGBoost CatBoost
3. Validate models' performance against other XAI methods, such as EBM, LORE, EXPLAN
4. Run experiments with human end-users/consumers of explanations

Building transparent classification models

dr inż. Bogdan Gulowaty

Artificial Intelligence in Research and Applications Seminar (AIRA)

29.05.2025

Publications

- Bogdan Gulowaty, Michał Woźniak, Extracting interpretable decision tree ensemble from random forest, International Joint Conference on Neural Networks IJCNN 2021 (CORE rank A, 140 p. MNiSzW)
- Bogdan Gulowaty, Michał Woźniak, Search-based framework for transparent non-overlapping ensemble models, International Joint Conference on Neural Networks IJCNN 2022 (CORE rank A, 140 p. MNiSzW)