Analysis of Tunability of Selected Machine Learning Algorithms

Piotr Tyrakowski, Dorota Rzewnicka

Objectives of Machine Learning Algorithms Analysis

Exploring Tunability and Hyperparameter Sampling Techniques

Primary Goal

Analyze the tunability of three machine learning algorithms: Random Forest, XGBoost, and Extra Trees.

Approach

Utilize four different datasets for comprehensive analysis.

Hyperparameter Sampling Techniques

Compare two hyperparameter sampling techniques: Random Search (RS) and Bayesian Optimization (BO).

Impact on Model Performance

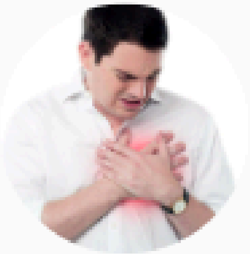
Investigate how hyperparameter tuning affects the performance of machine learning models.

Effect of Sampling Methods

Examine the impact of different sampling methods on the tunability of the algorithms.

Datasets

Overview of Key Datasets for Prediction and Classification



Heart Disease Dataset

Contains 303 observations and 14 features, aimed at predicting the presence of heart disease.

Diabetes Dataset

Comprises 768 observations and 8 features, focused on predicting the onset of diabetes.

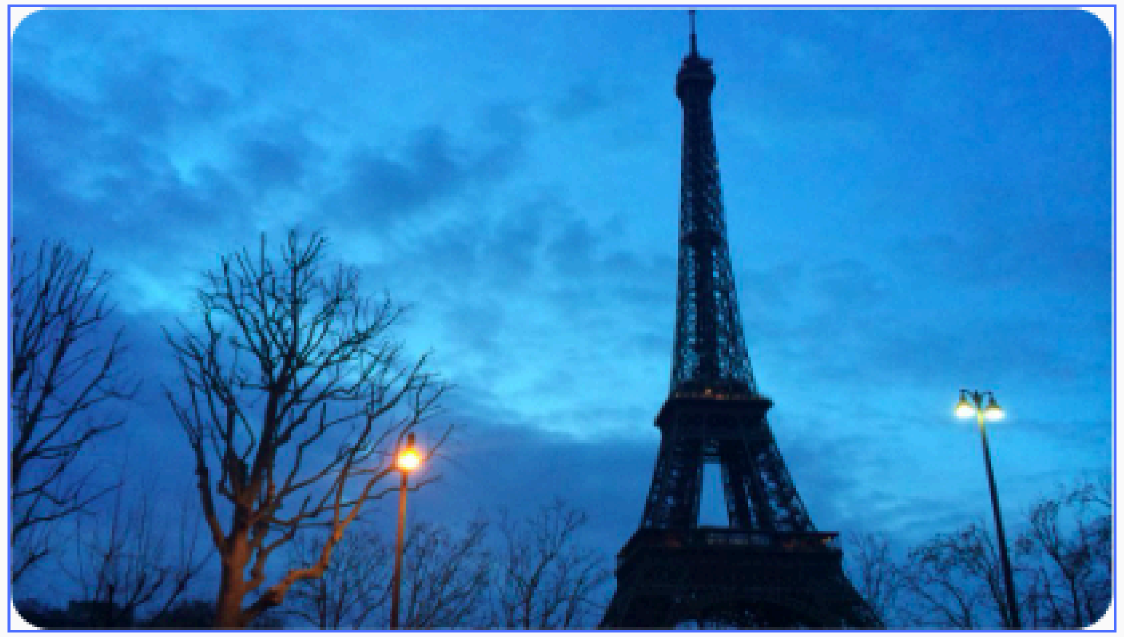
Spambase Dataset

Includes 4601 observations and 57 features, used for classifying emails as spam or not spam.

Phishing Websites Dataset

Encompasses 2456 observations and 30 features, designed for identifying phishing websites.

Algorithms and Hyperparameters



01 Random Forest Hyperparameters

Key hyperparameter ranges for Random Forest include
n_estimators (50-400),
criterion ('gini', 'entropy', 'log_loss'),
max_depth (5-50),
min_samples_leaf (1-14), min_samples_split (2-14),
max_features ('sqrt', 'log2', None)

02 Extra Trees Hyperparameters

Hyperparameter ranges for Extra Trees match those of Random Forest

03 XGBoost Hyperparameters

XGBoost hyperparameters include
n_estimators (50-400),
learning_rate (0.01-0.3),
max_depth (3-14),
subsample (0.5-1.0).

Methods

Hyperparameter Tuning Techniques and Evaluation Metric

Random Search (RS)

Randomly samples hyperparameters from predefined distributions, conducted with 100 iterations.

Bayesian Optimization (BO)

Uses probabilistic models to select promising hyperparameters, conducted with 50 iterations due to computational constraints.

Evaluation Metric

Balanced Accuracy Score accounts for class imbalance and is averaged over 5-fold cross-validation.

Results – Tunability

Impact of Hyperparameter Tuning on Algorithm Performance

01 **Random Forest Tunability**

Tunability varied across datasets, with maximum improvement ranging from 0% to 2.4%.

02 **XGBoost Tunability**

XGBoost demonstrated consistent small improvements, with a maximum improvement around 1.6%.

03 **Extra Trees Tunability**

Extra Trees exhibited slight improvements with tuning, achieving a maximum improvement around 2.1%.

04 **Overall Interpretation**

Some algorithms and datasets benefit more from hyperparameter tuning, and negative tunability values indicate improved performance over default settings.

	Random Forest		XGBoost		Extra Trees	
	RS	BO	RS	BO	RS	BO
Dane 1	−0.016	−0.012	0.000	0.0002	−0.008	−0.019
Dane 2	−0.010	−0.006	−0.013	−0.016	−0.021	−0.020
Dane 3	−0.024	−0.025	−0.005	−0.006	−0.001	−0.004
Dane 4	0.0	0.0	−0.014	−0.010	−0.015	−0.015

Results – Iterations Analysis

Convergence of Methods

Bayesian Optimization Efficiency

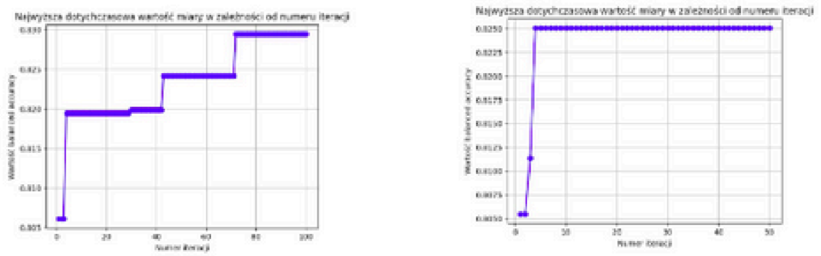
Stabilized performance within approximately 30-40 iterations, demonstrating more efficiency in finding optimal hyperparameters.

Random Search Performance Stabilization

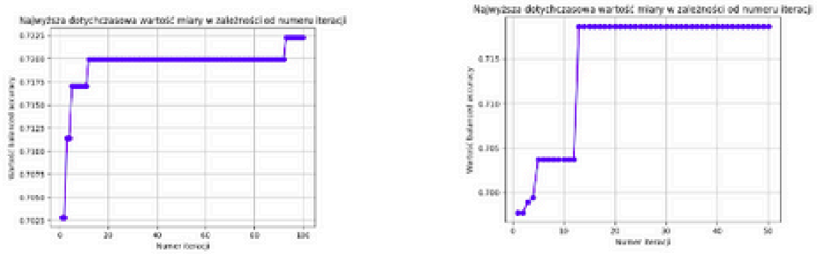
Generally required approximately 70 iterations to stabilize performance.

Comparison of Results

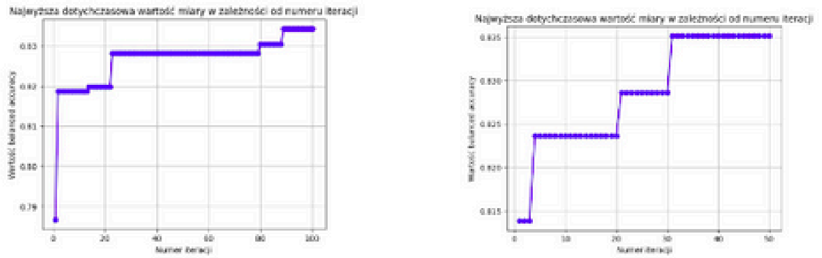
Bayesian Optimization can achieve comparable or better results in fewer iterations compared to Random Search.



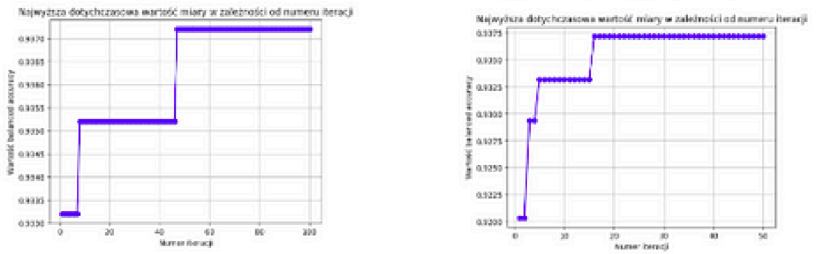
Rysunek 1: Random forest na zbiorze 1, po lewej RS, po prawej BO.



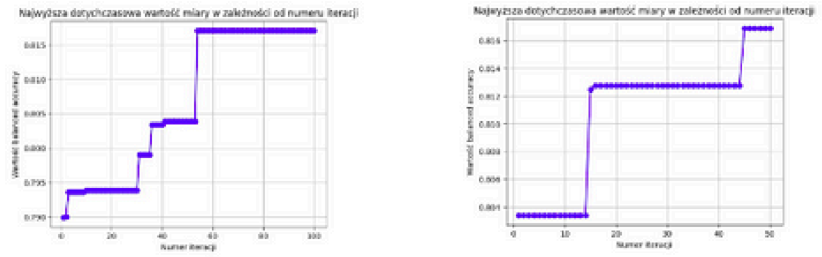
Rysunek 2: Random forest na zbiorze 2, po lewej RS, po prawej BO.



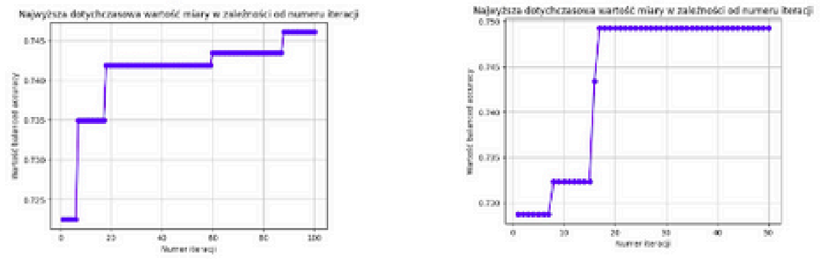
Rysunek 3: Random forest na zbiorze 3, po lewej RS, po prawej BO.



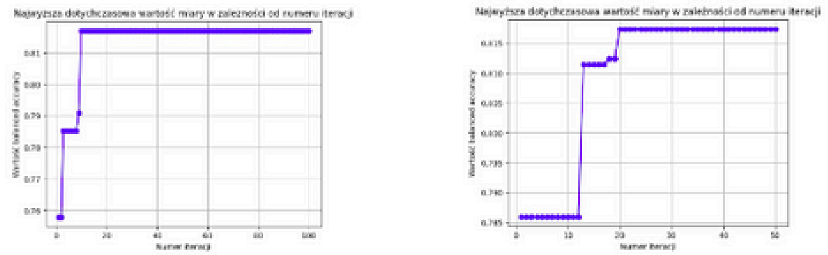
Rysunek 4: Random forest na zbiorze 4, po lewej RS, po prawej BO.



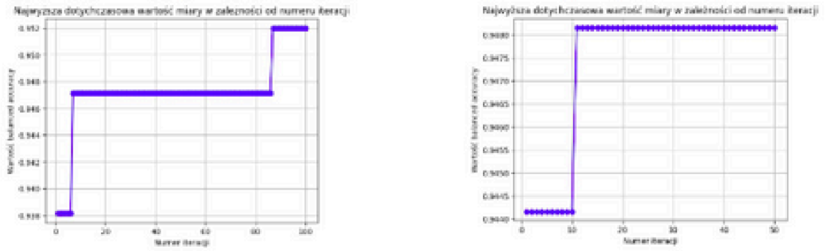
Rysunek 5: XGBoost na zbiorze 1, po lewej RS, po prawej BO.



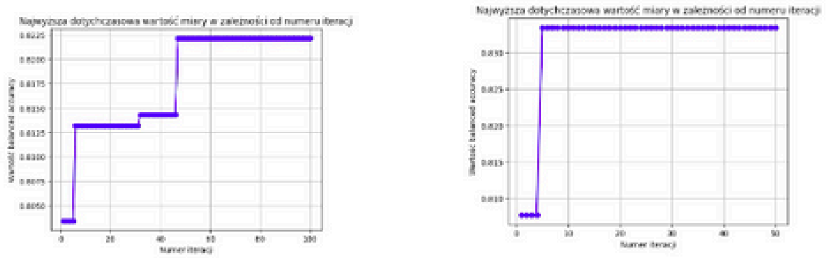
Rysunek 6: XGBoost na zbiorze 2, po lewej RS, po prawej BO.



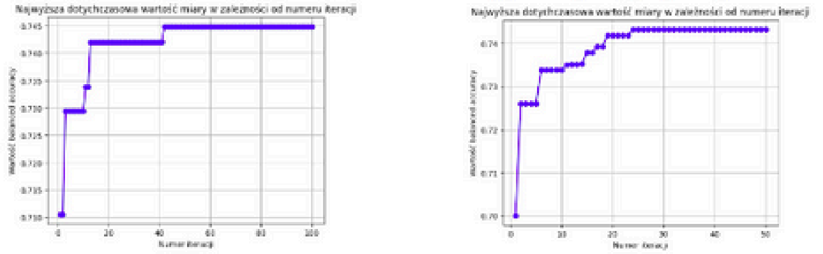
Rysunek 7: XGBoost na zbiorze 3, po lewej RS, po prawej BO.



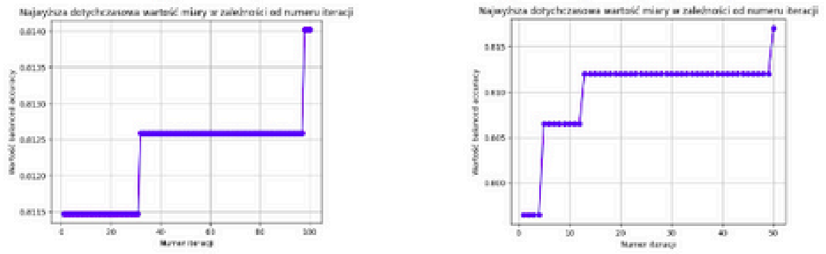
Rysunek 8: XGBoost na zbiorze 4, po lewej RS, po prawej BO.



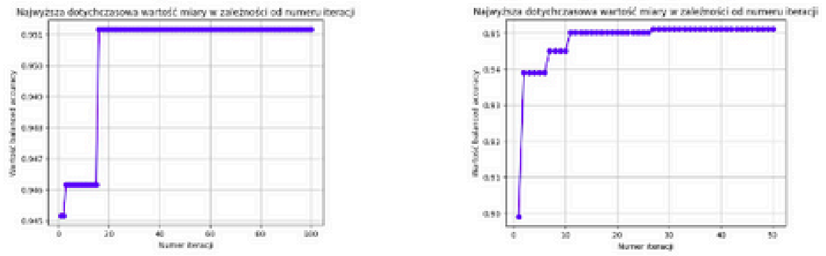
Rysunek 9: Extra trees na zbiorze 1, po lewej RS, po prawej BO.



Rysunek 10: Extra trees na zbiorze 2, po lewej RS, po prawej BO.



Rysunek 11: Extra trees na zbiorze 3, po lewej RS, po prawej BO.



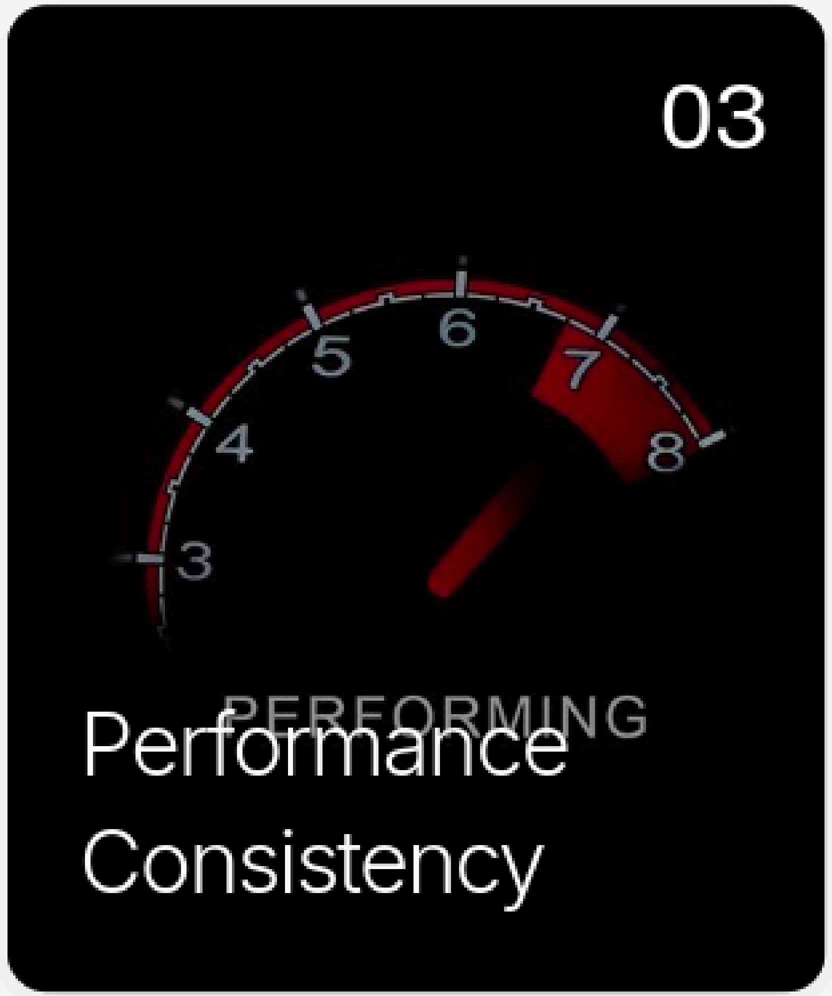
Rysunek 12: Extra trees na zbiorze 4, po lewej RS, po prawej BO.

Bias Sampling

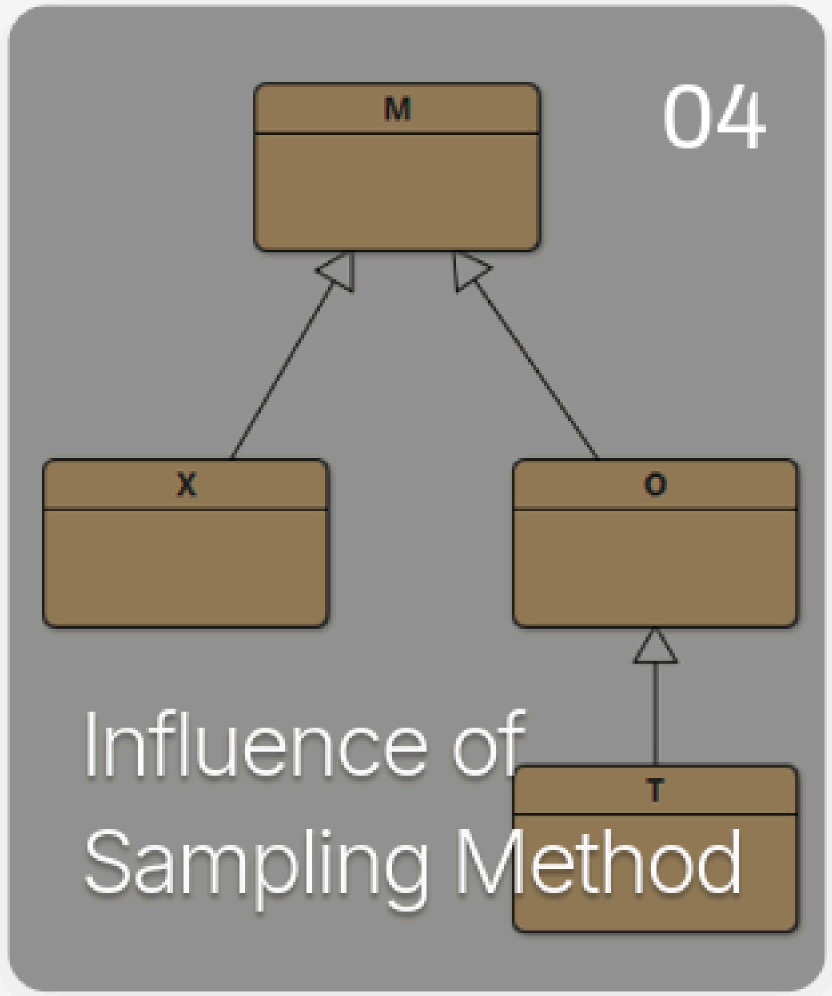
Comparing Random Search and Bayesian Optimization

Random Search involves uniform sampling across the hyperparameter space, leading to greater variability in performance results.

Bayesian Optimization focuses on promising regions of the hyperparameter space, resulting in less dispersed performance improvements.

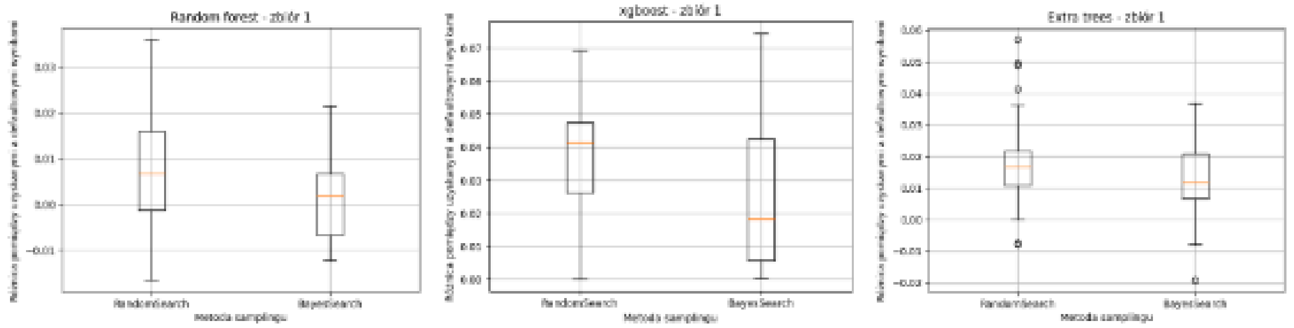


Bayesian Optimization tends to provide more consistent improvements compared to Random Search.



The method of sampling used has not a significant influence on the tunability assessment.

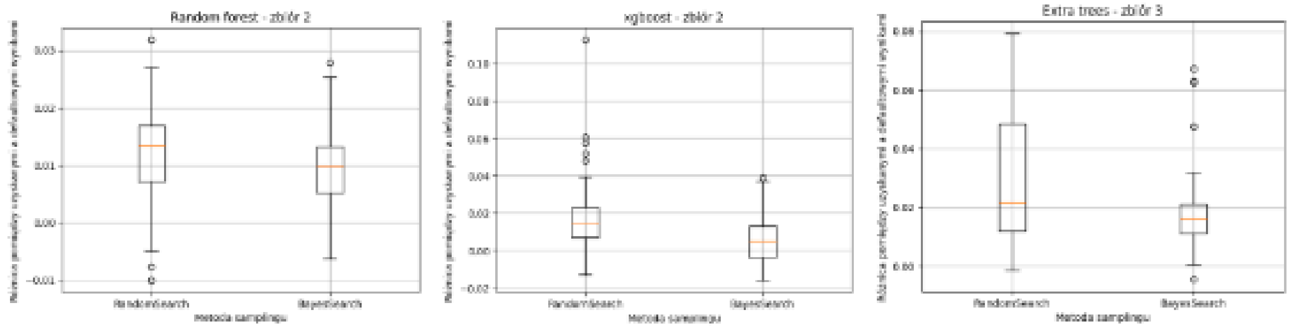
8.2.1 Dataset 1



(a) Random Forest - Dataset 1 (b) XGBoost - Dataset 1 (c) Extra Trees - Dataset 1

Rysunek 13: Boxploty dla Dataset 1

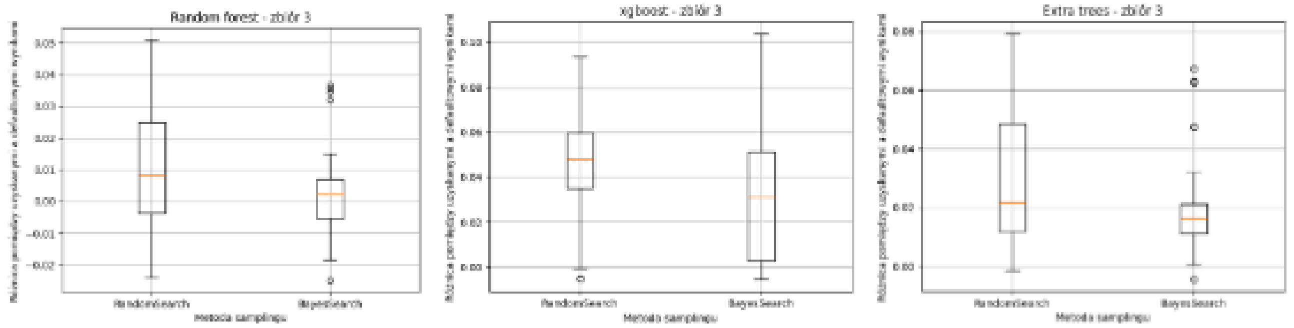
8.2.2 Dataset 2



(a) Random Forest - Dataset 2 (b) XGBoost - Dataset 2 (c) Extra Trees - Dataset 2

Rysunek 14: Boxploty dla Dataset 2

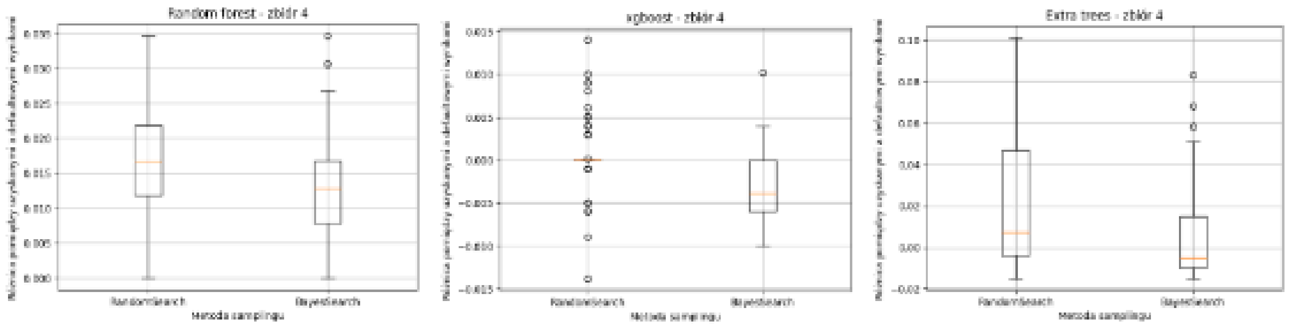
8.2.3 Dataset 3



(a) Random Forest - Dataset 3 (b) XGBoost - Dataset 3 (c) Extra Trees - Dataset 3

Rysunek 15: Boxploty dla Dataset 3

8.2.4 Dataset 4



(a) Random Forest - Dataset 4 (b) XGBoost - Dataset 4 (c) Extra Trees - Dataset 4

Rysunek 16: Boxploty dla Dataset 4