# Analiza Tunowalności Hiperparametrów w Algorytmach Uczenia Maszynowego

Maciej Kordyaczny
Bartosz Jadczak

21 listopada 2023

## Wstęp

W ramach analizy tunowalności hiperparametrów trzech wybranych algorytmów uczenia maszynowego – Random Forest, Gradient Boosting oraz k-Nearest Neighbors (k-NN), podjęliśmy się zadania optymalizacji ich parametrów na czterech niewielkich zbiorach danych, każdy składający się z około 1000 elementów. Celem naszego eksperymentu jest dogłębne zrozumienie, jak różne techniki losowania punktów wpływają na proces tunowania hiperparametrów, a także jakie są efektywności i charakterystyki optymalizacyjne poszczególnych algorytmów.

### Wybór Algorytmów i Zbiorów Danych

1. **Algorytmy:**
    - **Random Forest:** Wybór tego algorytmu wynika z jego zdolności do obsługi złożonych zbiorów danych i radzenia sobie z problemem overfittingu.
    - **Gradient Boosting:** Jako jeden z najskuteczniejszych algorytmów, gradient boosting stanowi istotną alternatywę, zwłaszcza w kontekście silnych, skomplikowanych modeli.
    - **k-Nearest Neighbors (k-NN):** Algorytm k-NN został uwzględniony z uwagi na jego prostotę i skuteczność w problemach klasyfikacji.
2. **Zbiory Danych:**
    - Wybór czterech niewielkich zbiorów danych o około 1000 elementach pozwoli nam skoncentrować się na efektywnym tunowaniu hiperparametrów bez nadmiernego obciążenia obliczeniowego.
    - Wybraliśmy tak 4 zbiory danych, aby ograniczyć kolumny z wartościami nie numerycznymi w celu ułatwienia procesu przygotowania danych do tunowania hiperparametrów.

### Cel Eksperymentu

Nasz eksperyment ma na celu nie tylko zoptymalizowanie hiperparametrów poszczególnych algorytmów, ale również porównanie różnych metod ich tunowania. Wykorzystamy techniki losowania punktów, takie jak Uniform Grid Search, Random Search, oraz Bayes Optimization, aby uzyskać głębsze zrozumienie wpływu tych strategii na proces optymalizacji. Analiza wyników pozwoli nam ocenić, które hiperparametry są kluczowe dla efektywności modeli w kontekście różnych algorytmów i zbiorów danych. Ostatecznie, nasze badania będą miały na celu dostarczenie praktycznych wskazówek dotyczących tunowalności hiperparametrów w kontekście małych zbiorów danych.

# 1 Metody Samplingu

W tej sekcji przedstawimy dwie wybrane metody samplingu, które zostaną użyte do tunowania hiperparametrów naszych algorytmów: *Random Search* oraz *Bayes Optimization* z wykorzystaniem *BayesSearchCV*.

## 1.1 Random Search

*Random Search* to technika, w której losowo wybierane są różne kombinacje hiperparametrów zdefiniowanych przedziałów. Jest to podejście efektywne, zwłaszcza w przypadku dużych przestrzeni hiperparametrów, gdyż pozwala na znalezienie rozwiązania nawet przy ograniczonych zasobach obliczeniowych.

## 1.2 Bayes Optimization z BayesSearchCV

*Bayes Optimization* to podejście wykorzystujące proces bayesowski do modelowania funkcji celu, co pozwala na skoncentrowanie się na najbardziej obiecujących obszarach przestrzeni hiperparametrów. *BayesSearchCV* jest narzędziem implementującym tę metodę w pakiecie *scikit-optimize*, umożliwiającym skuteczne dostosowanie hiperparametrów do danego problemu.

W naszym eksperymencie skorzystamy z *BayesSearchCV*, aby zoptymalizować hiperparametry naszych algorytmów, mając nadzieję na uzyskanie bardziej efektywnych wyników w porównaniu do *Random Search*.

# 3. Siatka Hiperparametrów

W celu przeprowadzenia optymalizacji hiperparametrów, zdefiniowaliśmy konkretne siatki parametrów dla każdego z wybranych algorytmów. Poniżej przedstawiamy szczegóły tych siatek:

### Random Forest

**max_depth** (1, 20) - Maksymalna głębokość drzewa. Kontroluje, jak głębokie drzewa mogą rosnąć.

**max_features** (1, 10) - Maksymalna liczba cech branych pod uwagę przy podziale węzłów.

**min_samples_split** (2, 10) - Minimalna liczba próbek wymagana do podziału węzła wewnętrznego.

**min_samples_leaf** (1, 20) - Minimalna liczba próbek wymagana do utworzenia liścia.

**bootstrap** [True, False] - Określa, czy próbki są zastępowane przy tworzeniu drzewa.

**criterion** ["gini", "entropy"] - Kryterium pomiaru jakości podziału.

## Gradient Boosting

**loss** ["deviance"] - Funkcja straty do minimalizacji podczas budowy modelu.

**learning_rate** [0.01, 0.025, 0.05, 0.075, 0.1, 0.15, 0.2] - Współczynnik uczenia kontrolujący wpływ każdego drzewa.

**min_samples_split** np.linspace(0.1, 0.5, 12) - Minimalna liczba próbek wymagana do podziału węzła wewnętrznego.

**min_samples_leaf** np.linspace(0.1, 0.5, 12) - Minimalna liczba próbek wymagana do utworzenia liścia.

**max_depth** [3,5,8] - Maksymalna głębokość drzewa.

**max_features** ["log2","sqrt"] - Maksymalna liczba cech branych pod uwagę przy podziale węzłów.

**criterion** ["friedman_mse", "squared_error"] - Kryterium pomiaru jakości podziału.

**subsample** [0.5, 0.618, 0.8, 0.85, 0.9, 0.95, 1.0] - Frakcja próbek używanych do dopasowania każdego drzewa.

**n_estimators** [10] - Liczba drzew w modelu.

## k-Nearest Neighbors (k-NN)

**n_neighbors** [3, 5, 7, 10] - Liczba sąsiadów branych pod uwagę podczas predykcji.

**weights** ["uniform", "distance"] - Wagi używane do prognozy. "Uniform" oznacza równe wagi, "distance" oznacza odwrotność odległości.

**p** [1, 2] - Wybór metryki odległości. 1 dla odległości Manhattan, 2 dla odległości euklidesowej.

**algorithm** ["auto", "ball_tree", "kd_tree", "brute"] - Algorytm używany do obliczeń najbliższych sąsiadów.

# 4. Przebieg Eksperymentu

## Optymalizacja Hiperparametrów

Eksperyment obejmował optymalizację hiperparametrów trzech algorytmów: Random Forest, Gradient Boosting i k-Nearest Neighbors (k-NN).

### Przygotowanie Danych

Staranna analiza czterech niewielkich zbiorów danych (około 1000 elementów), z odpowiednim przetworzeniem i normalizacją.

### Podział na Zbiory

Zbiory treningowe i testowe zachowujące odpowiednie proporcje.

### Tunowanie Hiperparametrów

Użycie Random Search i Bayes Optimization z BayesSearchCV, utrzymując stałą siatkę hiperparametrów.

### Ocena Wyników

Skuteczność modeli oceniana na zbiorze testowym, zbierając miary takie jak dokładność, precyzja, czułość itp.

### Analiza Wyników

Szczegółowa analiza wyników dla obu metod samplingu i algorytmów, z uwzględnieniem różnic w tunowalności.

### Porównanie Technik

Porównanie Random Search i Bayes Optimization, badanie wpływu technik losowania na wyniki tunowania hiperparametrów.

# Wyniki

Wyniki przedstawiliśmy w tabelach, gdzie każdy z wyników dla danej konfiguracji paramsetu, algorytmu oraz datasetu, reprezentowany jest przez parametr *Accuracy*, który został umieszczony w poszczególnych komórkach. Przedstawiliśmy również miarę tunowalności jako różnicę największego średniego *Accuracy* dla danego algorytmu oraz *Accuracy* w danej komórce.

## Bayes Optimization

| | Params Set | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 | Mean |
|---|---|---|---|---|---|---|
| KNN | 1 | 0.76 | 0.7686 | 0.9643 | 0.7078 | 0.800175 |
| | 2 | 0.78 | 0.7934 | 0.9732 | 0.7468 | 0.82335 |
| | 3 | 0.76 | 0.7851 | 0.9554 | 0.7468 | 0.811825 |
| | 4 | 0.7467 | 0.7851 | 0.9286 | 0.7468 | 0.8018 |
| | 5 | 0.7467 | 0.7851 | 0.9464 | 0.7403 | 0.804625 |
| | 6 | 0.7467 | 0.7851 | 0.9554 | 0.7338 | 0.80525 |
| RF | 1 | 0.7597 | 0.7597 | 0.7532 | 0.7857 | 0.764575 |
| | 2 | 0.7597 | 0.7597 | 0.7338 | 0.7532 | 0.7516 |
| | 3 | 0.7792 | 0.7143 | 0.7662 | 0.7403 | 0.75 |
| | 4 | 0.7667 | 0.9504 | 1 | 0.7597 | 0.8692 |
| | 5 | 0.7867 | 0.9256 | 1 | 0.7662 | 0.869625 |
| | 6 | 0.7867 | 0.8264 | 1 | 0.7792 | 0.848075 |
| GB | 1 | 0.7533 | 1 | 1 | 0.7532 | 0.876625 |
| | 2 | 0.7933 | 1 | 1 | 0.7208 | 0.878525 |
| | 3 | 0.72 | 1 | 1 | 0.7403 | 0.865075 |
| | 4 | 0.7533 | 0.6281 | 0.6964 | 0.7857 | 0.715875 |
| | 5 | 0.7733 | 1 | 1 | 0.7662 | 0.884875 |
| | 6 | 0.7667 | 1 | 1 | 0.7468 | 0.878375 |

Tabela 1: Bayes Optimization - ACC

| | Params Set | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 |
|---|---|---|---|---|---|
| KNN | 1 | -0.06335 | -0.05475 | 0.14095 | -0.11555 |
| | 2 | -0.04335 | -0.02995 | 0.14985 | -0.07655 |
| | 3 | -0.06335 | -0.03825 | 0.13205 | -0.07655 |
| | 4 | -0.07665 | -0.03825 | 0.10525 | -0.07655 |
| | 5 | -0.07665 | -0.03825 | 0.12305 | -0.08305 |
| | 6 | -0.07665 | -0.03825 | 0.13205 | -0.08955 |
| RF | 1 | -0.109925 | -0.109925 | -0.116425 | -0.083925 |
| | 2 | -0.109925 | -0.109925 | -0.135825 | -0.116425 |
| | 3 | -0.090425 | -0.155325 | -0.103425 | -0.129325 |
| | 4 | -0.102925 | 0.080775 | 0.130375 | -0.109925 |
| | 5 | -0.082925 | 0.055975 | 0.130375 | -0.103425 |
| | 6 | -0.082925 | -0.043225 | 0.130375 | -0.090425 |
| GB | 1 | -0.131575 | 0.115125 | 0.115125 | -0.131675 |
| | 2 | -0.091575 | 0.115125 | 0.115125 | -0.164075 |
| | 3 | -0.164875 | 0.115125 | 0.115125 | -0.144575 |
| | 4 | -0.131575 | -0.256775 | -0.188475 | -0.099175 |
| | 5 | -0.111575 | 0.115125 | 0.115125 | -0.118675 |
| | 6 | -0.118175 | 0.115125 | 0.115125 | -0.138075 |

Tabela 2: Bayes Optimization - miara tunowalności

## Random Search

| | L.P. | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 | Mean |
|---|---|---|---|---|---|---|
| RF | 1 | 0.7533 | 0.6281 | 0.6964 | 0.7403 | 0.704525 |
| | 2 | 0.7733 | 0.6281 | 0.7768 | 0.7468 | 0.73125 |
| | 3 | 0.7533 | 0.6281 | 0.6964 | 0.6429 | 0.680175 |
| | 4 | 0.64 | 0.9587 | 0.9732 | 0.7403 | 0.82805 |
| | 5 | 0.7533 | 0.6281 | 0.6964 | 0.7532 | 0.70775 |
| | 6 | 0.7533 | 0.6281 | 0.6964 | 0.7403 | 0.704525 |
| | 7 | 0.7533 | 0.6281 | 0.6964 | 0.6429 | 0.680175 |
| | 8 | 0.7 | 0.9817 | 1 | 0.7403 | 0.8555 |
| | 9 | 0.76 | 0.843 | 0.9554 | 0.7273 | 0.821425 |
| | 10 | 0.7133 | 0.9256 | 0.8661 | 0.7532 | 0.81455 |
| | 11 | 0.7267 | 0.9091 | 1 | 0.7208 | 0.83915 |
| | 12 | 0.7667 | 0.7521 | 0.9554 | 0.7338 | 0.802 |
| | 13 | 0.7333 | 0.9001 | 1 | 0.7403 | 0.843425 |
| | 14 | 0.7467 | 0.9008 | 1 | 0.7403 | 0.84695 |
| | 15 | 0.7533 | 0.6281 | 0.6964 | 0.6429 | 0.680175 |
| | 16 | 0.7533 | 0.6281 | 0.6964 | 0.6429 | 0.680175 |
| | 17 | 0.7533 | 0.6281 | 0.6964 | 0.6494 | 0.6818 |
| | 18 | 0.7533 | 0.6281 | 0.6964 | 0.6429 | 0.680175 |
| | 19 | 0.7533 | 0.6281 | 0.6964 | 0.6429 | 0.680175 |
| | 20 | 0.7333 | 0.8595 | 1 | 0.7532 | 0.8365 |
| GB | 21 | 0.7533 | 0.6281 | 0.6518 | 0.6429 | 0.669025 |
| | 22 | 0.7533 | 0.6281 | 0.4464 | 0.6429 | 0.617675 |
| | 23 | 0.7533 | 0.6281 | 0.6964 | 0.7662 | 0.711 |
| | 24 | 0.7533 | 0.6281 | 0.6518 | 0.6429 | 0.669025 |
| | 25 | 0.7533 | 0.6281 | 0.5714 | 0.6429 | 0.648925 |
| | 26 | 0.7533 | 0.6281 | 0.6518 | 0.7078 | 0.68525 |
| | 27 | 0.7533 | 0.6281 | 0.4464 | 0.6429 | 0.617675 |
| | 28 | 0.7533 | 0.6281 | 0.4464 | 0.6429 | 0.617675 |
| | 29 | 0.7533 | 0.6281 | 0.4464 | 0.6429 | 0.617675 |
| | 30 | 0.7533 | 0.6281 | 0.4464 | 0.6429 | 0.617675 |
| | 31 | 0.7533 | 0.6281 | 0.4464 | 0.6429 | 0.617675 |
| | 32 | 0.7533 | 0.6281 | 0.4464 | 0.6429 | 0.617675 |
| | 33 | 0.7533 | 0.6281 | 0.6518 | 0.6429 | 0.669025 |
| | 34 | 0.7533 | 0.6281 | 0.4464 | 0.6429 | 0.617675 |
| | 35 | 0.7533 | 0.6281 | 0.6964 | 0.6429 | 0.680175 |
| | 36 | 0.7533 | 0.6281 | 0.6518 | 0.6429 | 0.669025 |
| | 37 | 0.7533 | 0.6281 | 0.4464 | 0.6429 | 0.617675 |
| | 38 | 0.7533 | 0.6281 | 0.6518 | 0.7208 | 0.6885 |
| | 39 | 0.7533 | 0.6281 | 0.6964 | 0.6429 | 0.680175 |
| | 40 | 0.7533 | 0.6281 | 0.4464 | 0.6429 | 0.617675 |

Tabela 3: Random Search - ACC

| | L.P. | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 | Mean |
|---|---|---|---|---|---|---|
| KNN | 41 | 0.7133 | 0.7025 | 0.9375 | 0.6688 | 0.755525 |
| | 42 | 0.7733 | 0.6942 | 0.9643 | 0.6883 | 0.780025 |
| | 43 | 0.7733 | 0.6281 | 0.8214 | 0.6883 | 0.727775 |
| | 44 | 0.78 | 0.6446 | 0.9643 | 0.7403 | 0.7823 |
| | 45 | 0.7 | 0.719 | 0.9554 | 0.6753 | 0.762425 |
| | 46 | 0.7067 | 0.595 | 0.9464 | 0.6688 | 0.729225 |
| | 47 | 0.7733 | 0.6777 | 0.8482 | 0.6883 | 0.746875 |
| | 48 | 0.78 | 0.6446 | 0.9643 | 0.7403 | 0.7823 |
| | 49 | 0.7 | 0.7438 | 0.9732 | 0.6883 | 0.776325 |
| | 50 | 0.76 | 0.7603 | 0.9554 | 0.7078 | 0.795875 |
| | 51 | 0.7533 | 0.6942 | 0.9554 | 0.6883 | 0.7728 |
| | 52 | 0.7 | 0.719 | 0.9643 | 0.6883 | 0.7679 |
| | 53 | 0.6867 | 0.5702 | 0.9643 | 0.6883 | 0.727375 |
| | 54 | 0.76 | 0.7686 | 0.9732 | 0.7078 | 0.8024 |
| | 55 | 0.76 | 0.6281 | 0.8214 | 0.6494 | 0.714725 |
| | 56 | 0.7 | 0.7025 | 0.8393 | 0.6494 | 0.7228 |
| | 57 | 0.7133 | 0.7355 | 0.875 | 0.6623 | 0.746525 |
| | 58 | 0.7333 | 0.7686 | 0.8839 | 0.6948 | 0.77015 |
| | 59 | 0.6733 | 0.7769 | 0.8839 | 0.6948 | 0.757225 |
| | 60 | 0.6867 | 0.5702 | 0.9643 | 0.6883 | 0.727375 |

Tabela 4: Random Search - ACC

| | L.P. | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 |
|---|---|---|---|---|---|
| KNN | 1 | -0.1022 | -0.2274 | -0.1591 | -0.1152 |
| | 2 | -0.0822 | -0.2274 | -0.0787 | -0.1087 |
| | 3 | -0.1022 | -0.2274 | -0.1591 | -0.2126 |
| | 4 | -0.2155 | 0.1032 | 0.1177 | -0.1152 |
| | 5 | -0.1022 | -0.2274 | -0.1591 | -0.1023 |
| | 6 | -0.1022 | -0.2274 | -0.1591 | -0.1152 |
| | 7 | -0.1022 | -0.2274 | -0.1591 | -0.2126 |
| | 8 | -0.1555 | 0.1262 | 0.1445 | -0.1152 |
| | 9 | -0.0955 | -0.0125 | 0.0999 | -0.1282 |
| | 10 | -0.1422 | 0.0701 | 0.0106 | -0.1023 |
| | 11 | -0.1288 | 0.0536 | 0.1445 | -0.1347 |
| | 12 | -0.0888 | -0.1034 | 0.0999 | -0.1217 |
| | 13 | -0.1222 | 0.0446 | 0.1445 | -0.1152 |
| | 14 | -0.1088 | 0.0453 | 0.1445 | -0.1152 |
| | 15 | -0.1022 | -0.2274 | -0.1591 | -0.2126 |
| | 16 | -0.1022 | -0.2274 | -0.1591 | -0.2126 |
| | 17 | -0.1022 | -0.2274 | -0.1591 | -0.2061 |
| | 18 | -0.1022 | -0.2274 | -0.1591 | -0.2126 |
| | 19 | -0.1022 | -0.2274 | -0.1591 | -0.2126 |
| | 20 | -0.1222 | 0.004 | 0.1445 | -0.1023 |
| RF | 21 | 0.0423 | -0.0829 | -0.0592 | -0.0681 |
| | 22 | 0.0423 | -0.0829 | -0.2646 | -0.0681 |
| | 23 | 0.0423 | -0.0829 | -0.0146 | 0.0552 |
| | 24 | 0.0423 | -0.0829 | -0.0592 | -0.0681 |
| | 25 | 0.0423 | -0.0829 | -0.1396 | -0.0681 |
| | 26 | 0.0423 | -0.0829 | -0.0592 | -0.0032 |
| | 27 | 0.0423 | -0.0829 | -0.2646 | -0.0681 |
| | 28 | 0.0423 | -0.0829 | -0.2646 | -0.0681 |
| | 29 | 0.0423 | -0.0829 | -0.2646 | -0.0681 |
| | 30 | 0.0423 | -0.0829 | -0.2646 | -0.0681 |
| | 31 | 0.0423 | -0.0829 | -0.2646 | -0.0681 |
| | 32 | 0.0423 | -0.0829 | -0.2646 | -0.0681 |
| | 33 | 0.0423 | -0.0829 | -0.0592 | -0.0681 |
| | 34 | 0.0423 | -0.0829 | -0.2646 | -0.0681 |
| | 35 | 0.0423 | -0.0829 | -0.0146 | -0.0681 |
| | 36 | 0.0423 | -0.0829 | -0.0592 | -0.0681 |
| | 37 | 0.0423 | -0.0829 | -0.2646 | -0.0681 |
| | 38 | 0.0423 | -0.0829 | -0.0592 | 0.0098 |
| | 39 | 0.0423 | -0.0829 | -0.0146 | -0.0681 |
| | 40 | 0.0423 | -0.0829 | -0.2646 | -0.0681 |

Tabela 5: Random Search - miara tunowalności

| | L.P. | Dataset 1 | Dataset 2 | Dataset 3 | Dataset 4 |
|---|---|---|---|---|---|
| GB | 41 | -0.0891 | -0.0999 | 0.1351 | -0.1336 |
| | 42 | -0.0291 | -0.1082 | 0.1619 | -0.1141 |
| | 43 | -0.0291 | -0.1743 | 0.019 | -0.1141 |
| | 44 | -0.0224 | -0.1578 | 0.1619 | -0.0621 |
| | 45 | -0.1024 | -0.0834 | 0.153 | -0.1271 |
| | 46 | -0.0957 | -0.2074 | 0.144 | -0.1336 |
| | 47 | -0.0291 | -0.1247 | 0.0458 | -0.1141 |
| | 48 | -0.0224 | -0.1578 | 0.1619 | -0.0621 |
| | 49 | -0.1024 | -0.0586 | 0.1708 | -0.1141 |
| | 50 | -0.0424 | -0.0421 | 0.153 | -0.0946 |
| | 51 | -0.0491 | -0.1082 | 0.153 | -0.1141 |
| | 52 | -0.1024 | -0.0834 | 0.1619 | -0.1141 |
| | 53 | -0.1157 | -0.2322 | 0.1619 | -0.1141 |
| | 54 | -0.0424 | -0.0338 | 0.1708 | -0.0946 |
| | 55 | -0.0424 | -0.1743 | 0.019 | -0.153 |
| | 56 | -0.1024 | -0.0999 | 0.0369 | -0.153 |
| | 57 | -0.0891 | -0.0669 | 0.0726 | -0.1401 |
| | 58 | -0.0691 | -0.0338 | 0.0815 | -0.1076 |
| | 59 | -0.1291 | -0.0255 | 0.0815 | -0.1076 |
| | 60 | -0.1157 | -0.2322 | 0.1619 | -0.1141 |

Tabela 6: Random Search - miara tunowalności

# 4. Wnioski

## Porównanie Metod Tunowania

- Bayes Optimization osiągnął wyższą średnią skuteczność niż Random Search dla wszystkich algorytmów.
- Random Forest miał stabilne wyniki, natomiast Gradient Boosting i k-Nearest Neighbors reagowały różnie na metody tunowania, z korzyścią dla Bayes Optimization.

## Analiza Tunowalności Algorytmów

- Random Forest wykazał stabilność wyników, Gradient Boosting osiągnął najlepsze rezultaty z Bayes Optimization, a k-Nearest Neighbors był bardziej wrażliwy na różnice w metodach tunowania.

## Wpływ Techniki Losowania

- Różnice między Random Search a Bayes Optimization sugerują istotny wpływ techniki losowania.
- Bayes Optimization, opierając się na modelu probabilistycznym, wydaje się bardziej efektywne w przeszukiwaniu przestrzeni hiperparametrów.

## Potencjalne Rozwinięcia

- Użycie testów statystycznych i narzędzi wizualizacyjnych dla pełniejszej analizy różnic między metodami tunowania.
- Dalsze badania nad nowymi technikami tunowania hiperparametrów dostosowanymi do charakterystyki algorytmów.

Podsumowując, eksperyment ujawnił istotne różnice między Random Search a Bayes Optimization, z cennymi wnioskami dla praktycznych zastosowań uczenia maszynowego.

# 8. Podsumowanie

Eksperyment skupił się na tunowalności hiperparametrów trzech algorytmów. Wnioski obejmują wyższą skuteczność Bayes Optimization, różnice w tunowalności algorytmów, wpływ techniki losowania, oraz potencjalne obszary dalszych badań.