Analysis of Halving Random Search Cross Validation in Machine Learning Model Optimization

Alief Cahyo Utomo, Sucipto, Muhammad Najibulloh Muzaki

Universitas Nusantara PGRI Kediri, Jl. Ahmad Dahlan No.76, Mojoroto, Kec. Mojoroto, Kota Kediri, Jawa Timur, 64112, Indonesia

e-mail: aliepcahyo@gmail.com

Abstract: *This study aims to analyze the effectiveness of the Halving Random Search Cross Validation method as an alternative for hyperparameter optimization in machine learning models compared to Grid Search Cross Validation and Random Search Cross Validation. The dataset used is Internet Service Churn with four algorithms: KNN, Decision Tree, SVM, and Gaussian Naive Bayes. The testing process involves 10-fold cross validation and three repetitions to ensure the validity of the results. The experimental results show that Halving Random Search Cross Validation is able to achieve competitive accuracy, precision, and recall performance (difference < 0.5%) compared to Grid Search in most models, with computational time savings of up to 62–74% on KNN, Decision Tree, and SVM. However, on Gaussian Naive Bayes with a small hyperparameter space, this method is slower due to the successive halving overhead. Random Search shows high speed but less stable on SVM and Gaussian Naive Bayes. The research conclusion states that Halving Random Search Cross Validation is the most balanced method for business cases such as churn prediction, with recommendations for application on complex models and further development using Hyperband or Bayesian Optimization.*

Key Words*:* Halving Random Search Cross-Validation, Hyperparameter Optimization, Machine Learning

Introduction

In the current digital era, machine learning is used in various industrial applications from health, finance, to telecommunications. The performance of the model is highly determined by the optimal hyperparameter configuration. However, the hyperparameter tuning process on large-scale datasets often takes a lot of time and computational resources, especially when using traditional methods like Grid Search Cross Validation which evaluates all parameter combinations.

Some previous studies have compared hyperparameter optimization methods. Bergstra and Bengio (2012) showed that Random Search Cross Validation is much more efficient than Grid Search on large search spaces. Li et al. (2018) then introduced Successive Halving and Hyperband as resource allocation-based approaches that can drastically reduce model evaluation time. scikit-learn since version 0.24 has implemented Halving Random Search Cross Validation as a combination of both, but empirical studies comparing the performance and efficiency of Halving Random Search Cross Validation directly with Grid Search and Random Search on various machine learning algorithms are still very limited.

The novelty of this research lies in the comparative analysis of Halving Random Search Cross Validation compared to Grid Search and Random Search using four different algorithms (KNN, Decision Tree, SVM, Gaussian Naive Bayes) on a telecommunications churn dataset (72,274 rows), with a dual focus on computational time efficiency and performance metrics (accuracy, precision, recall).

The objective of this research is to analyze and prove that Halving Random Search Cross Validation can provide an optimal balance between computational time efficiency and model quality in customer churn prediction cases, so it can be recommended as a standard hyperparameter tuning method in industrial environments with limited computational resources.

Method

The data collection procedure follows the CRISP-DM (Cross-Industry Standard Process for Data Mining) framework as a systematic reference (Nila et al., 2023), starting from the Business Understanding stage which focuses on evaluating the effectiveness of Halving Random Search Cross Validation compared to other methods, followed by Data Understanding to understand the churn dataset, and Data Preparation through the removal of empty data and normalization using Min-Max Scaler. In the Modeling stage, hyperparameter optimization is performed on four algorithms (KNN, Decision Tree, SVM, and Gaussian Naive Bayes) with 10-fold cross validation, where Grid Search and Random Search use discrete hyperparameter ranges as in Table 1 with a total of 456 combinations for Grid and 46 iterations for Random equivalent to 10% of the total, while Halving uses continuous distributions such as loguniform and randint in Table 2 for a wider search space. Each experiment is repeated three times to obtain average values.

Table 1. List Hyperparameter 1

Model	Hyperparameter	Value	Number of unique combinations
KNN	n_neighbor	3, 5, 7, 9, 11, 13, 15, 17, 19, 21	200 Combinations
	algorithm	'ball_tree', 'kd_tree'	
	weight	uniform, distance	
	metric	'cityblock', 'cosine', 'euclidean', 'manhattan', 'nan_euclidean'	
Decision Tree	criterion	'gini', 'entropy', dan 'log_loss'	240 Combinations
	max_depth	None, 3, 5, 7, 8, 9, 10, 15, 20, 25	
	min_sample_leaf	10, 20, 30, 40, 50, 100, 200, 500	
SVM	C	0.001, 1.0	6 Combinations
	kernel	'linear', 'rbf', 'poly'	
Gaussian Naive Bayes	var_smoothing	0.001, 0.00316, 0.01, 0.0316, 0.1, 0.316, 1.0, 3.16, 10.0, 100.0	10 Combinations

Table 2. List Hyperparameter 2

Model	Hyperparameter	Value	Number of unique combinations
KNN	n_neighbor	‘randint(3, 30)’	810 Combinations
	algorithm	'ball_tree', 'kd_tree'	
	weight	uniform, distance	
	metric	'cityblock', 'cosine', 'euclidean', 'manhattan', 'nan_euclidean'	
Decision Tree	criterion	'gini', 'entropy', 'log_loss'	37,500 Combinations
	max_depth	‘[None, range(3, 26)]’	
	min_sample_leaf	‘randint(10, 501)’	
SVM	C	‘loguniform(1e-3, 1e2)’	Unlimited
	kernel	'linear', 'rbf', 'poly'	
Gaussian Naive Bayes	var_smoothing	‘loguniform(1e-3, 1e2)’	Unlimited

Results and Discussion

1. Dataset Exploration Results

The data exploration results show that the dataset consists of 72,274 data points, 10 features, and 1 label with a relatively balanced class distribution (55.4% churn and 44.6% non-churn). Correlation analysis shows that subscription duration and remaining contract are negatively correlated with churn. Some features have outliers, but they are retained because they represent real customer conditions.

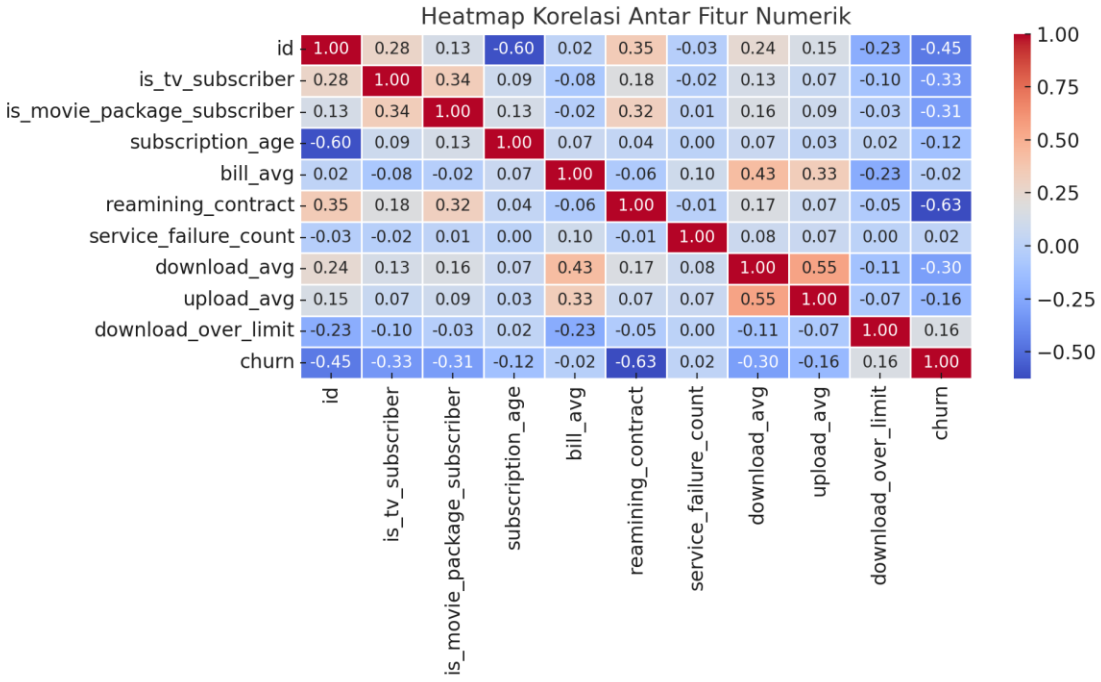


Figure 1. Heatmap of Correlations Between Features

2. Dataset Pre-Processing

The data is cleaned of missing values with median imputation on remaining contract and average on download average and upload average. The id feature is removed because it is not predictively relevant. All numeric features are normalized using Min-Max Scaling to the range 0-1. The dataset is then split into 90% training data and 10% test data.

3. Training Process

The selected models are KNN, Decision Tree, SVM, and Gaussian Naive Bayes. Hyperparameter optimization is performed with three methods, namely Grid Search Cross Validation, Random Search Cross Validation, and Halving Random Search Cross Validation using 10-fold cross validation. Each experiment is repeated three times, and computational time is measured using the perf_counter function to measure the time required by each method.

Table 3. Grid Search Cross Validation Method

Model	Run	Time (seconds)	Accuracy	Precision	Recall
KNN	1	1540.61	0.9208	0.9254	0.9301
	2	1556.37			
	3	1577.01			
Decision Tree	1	487.07	0.9395	0.9505	0.9382
	2	484.42			
	3	483.17			
SVM	1	7319.16	0.8312	0.8156	0.8929
	2	6892.50			
	3	6501.26			
GNB	1	2.22	0.7522	0.7927	0.74
	2	2.91			
	3	2.13			

Table 4. Random Search Cross Validation Method

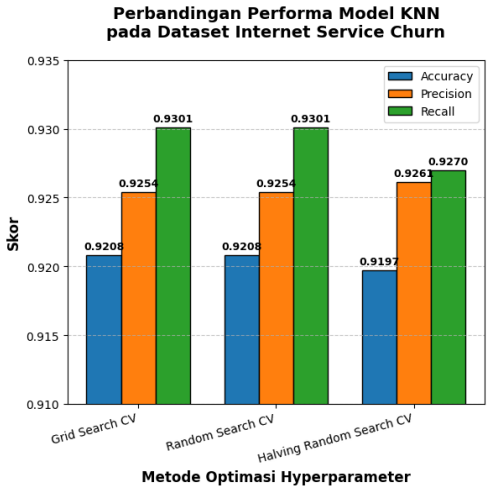
Model	Run	Time (seconds)	Accuracy	Precision	Recall
KNN	1	301.37	0.9208	0.9254	0.9301
	2	299.22			
	3	309.18			
Decision Tree	1	44.67	0.9392	0.9509	0.9372
	2	43.86			
	3	44.56			
SVM	1	1121.51	0.7721	0.74832	0.8784
	2	1261.76			
	3	1014.56			
GNB	1	0.87	0.5464	0.5464	1.0
	2	0.35			
	3	0.34			

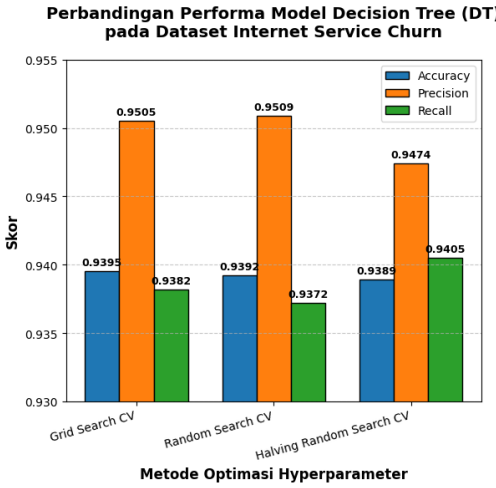
Table 5. Halving Random Search Cross Validation Method

Model	Run	Time (seconds)	Accuracy	Precision	Recall
KNN	1	419.95	0.9197	0.9261	0.9270
	2	412.06			
	3	411.08			
Decision Tree	1	182.32	0.9389	0.9474	0.9405
	2	187.44			
	3	186.31			
SVM	1	1804.39	0.8281	0.8099	0.8956
	2	1800.82			
	3	1794.14			
GNB	1	204.32	0.7281	0.8857	0.5769
	2	200.98			
	3	206.03			

Conclusion

Based on the test results, Halving Random Search Cross Validation has proven to provide an optimal balance between model quality and computational time efficiency compared to Grid Search Cross Validation and Random Search Cross Validation methods. Halving Random Search significantly reduces computational time in most models, especially on SVM, Decision Tree, and KNN. Although on Gaussian Naive Bayes this method becomes relatively slower due to the successive halving overhead on a narrow hyperparameter space. Overall, this method is able to achieve an accuracy difference of less than 0.5% compared to Grid Search and computational time savings of 62–74%.





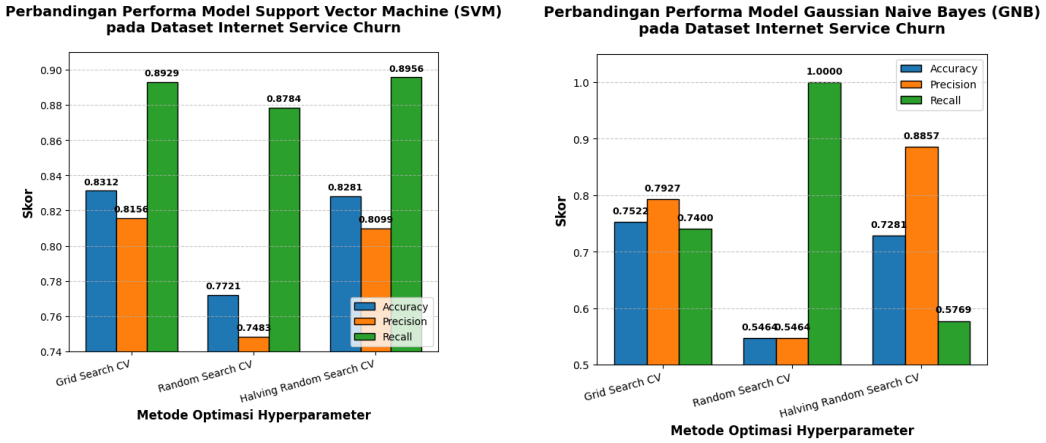


Figure 2. Comparison of Accuracy, Precision, and Recall

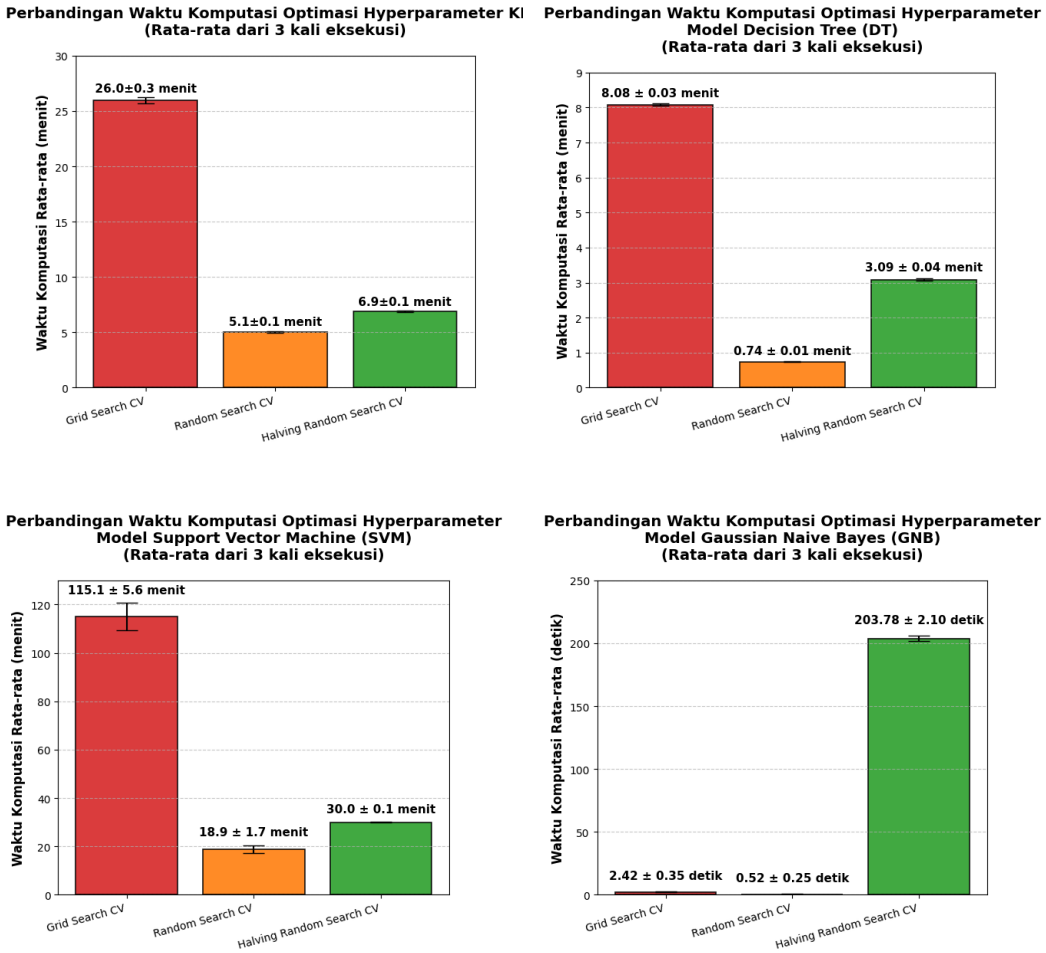


Figure 3. Time Comparison

References

Anggaini, I. Y., Sucipto, S., & Indriati, R. (2018). Cyberbullying detection modelling at Twitter social networking. JUITA: Jurnal Informatika, 6(2), 113–118. https://doi.org/10.30595/juita.v6i2.3350

Aprilliandhika, W., & Fauzi Abdullah, F. (2024). Comparison of K-nearest neighbor and support vector machine algorithm optimization with grid search CV on stroke prediction. Journal

of Information Technology, 5(4), 991–1000. https://doi.org/10.52436/1.jutif.2024.5.4.1951

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of Machine Learning Research, 13, 281–305.

Dirjen, S. K., Riset, P., Pengembangan, D., Dikti, R., Irmanda, H. N., & Astriratma, R. (2017). Klasifikasi jenis pantun dengan metode support vector machines (SVM). RESTI, 1(3), 915–922. https://doi.org/10.29207/resti.v4i5.2313

Irawan, I., Qisthiano, R., Syahril, M., & Jakak, P. M. (2023). Optimasi prediksi kelulusan tepat waktu: Studi perbandingan algoritma random forest dan algoritma K-NN berbasis PSO. Jurnal Pengembangan Sistem Informasi dan Informatika, 4(4). https://doi.org/10.47747/jpsii.v4i4.1374

Jamiluddin, F., Faisal, S., Arum Puspita Lestari, S., Fauzi, A., ... (2024). Implementasi hyperparameter tuning grid search CV pada prediksi produksi padi menggunakan algoritma linear regresi. Journal of Information System Research, 6(1), 490–498. https://doi.org/10.47065/josh.v6i1.5930

Li, L., Jamieson, K., DeSalvo, G., Rostamizadeh, A., & Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization. Journal of Machine Learning Research, 18(185), 1–52.

Misnawati. (2023). ChatGPT: Keuntungan, risiko, dan penggunaan bijak dalam era kecerdasan buatan. Jurnal Prosiding Mateandrau, 2(1). https://doi.org/10.55606/mateandrau.v2i1.221

Muhamad, I., & Matin, M. (2023). Hyperparameter tuning menggunakan GridSearchCV pada random forest untuk deteksi malware. https://doi.org/10.32722/multinetics.v9i1.5578

Munawaroh, S., Rosyidah, U. A., & Yanuarti, R. (2024). Klasifikasi tingkat kecemasan atlet sebelum bertanding menggunakan algoritma K-nearest neighbor (KNN) berbasis website. BIOS: Jurnal Teknologi Informasi dan Rekayasa Komputer, 5(2), 87–94. https://doi.org/10.37148/bios.v5i2.120

Nila, U., Firliana, R., & Sucipto. (2023). Analisis data transaksi penjualan produk pertanian menggunakan algoritma FP-Growth. Prosiding SEMNAS INOTEK, 7. https://doi.org/10.29407/inotek.v7i1.3426

Nugraha, W., & Sasongko, A. (2022). Hyperparameter tuning pada algoritma klasifikasi dengan grid search. SISTEMASI: Jurnal Sistem Informasi, 11(2). https://doi.org/10.32520/stmsi.v11i2.1750

Nugroho, A., Soeleman, Ma., Anggi Pramunendar, R., & Nurhindarto, A. (2023). Peningkatan performa ensemble learning pada segmentasi semantik gambar dengan teknik oversampling untuk class imbalance. https://doi.org/10.25126/jtiik.2023106831

Nurjanah, I., Karaman, J., Widaningrum, I., Mustikasari, D., & Sucipto. (2023). Penggunaan algoritma Naive Bayes untuk menentukan pemberian kredit pada koperasi desa. Journal of Computer Science and Information Technology E-ISSN, 3(2). https://doi.org/10.47065/explorerv3i2.766

Putri, T. A. E., Widiharih, T., & Santoso, R. (2023). Penerapan tuning hyperparameter RandomSearchCV pada adaptive boosting untuk prediksi kelangsungan hidup pasien gagal jantung. Jurnal Gaussian, 11(3), 397–406. https://doi.org/10.14710/j.gauss.11.3.397-406

Rahmat, A., Syafiih, M., & Faid, M. (2023). Implementasi klasifikasi potensi penyakit jantung dengan menggunakan metode C4.5 berbasis website (studi kasus Kaggle.com). INFOTECH Journal, 9(2), 393–400. https://doi.org/10.31949/infotech.v9i2.6295

Sartika, D., & Sensuse, D. I. (2017). Perbandingan algoritma klasifikasi Naive Bayes, nearest neighbour, dan decision tree pada studi kasus pengambilan keputusan pemilihan pola pakaian. JATISI, 3(2). https://doi.org/10.35957/jatisi.v3i2.78

Soper, D. S. (2023). Hyperparameter optimization using successive halving with greedy cross validation. Algorithms, 16(1). https://doi.org/10.3390/a16010017

Sucipto, Prasetya, D. D., & Widiyaningtyas, T. (2025). A supervised hybrid weighting scheme for Bloom's taxonomy questions using category space density-based weighting. Engineering, Technology and Applied Science Research, 15(2), 22102–22108. https://doi.org/10.48084/etasr.10226