

# Komparasi Teknik Hyperparameter Optimization pada SVM untuk Permasalahan Klasifikasi dengan Menggunakan Grid Search dan Random Search

Muhamad Fajri<sup>1\*</sup>, Aji Primajaya<sup>2\*</sup>

\* Teknik Informatika, Universitas Singaperbangsa Karawang  
[muhammad.fajri18218@student.unsika.a.c.id](mailto:muhammad.fajri18218@student.unsika.a.c.id)<sup>1</sup>, [aji.primajaya@staff.unsika.ac.id](mailto:aji.primajaya@staff.unsika.ac.id)<sup>2</sup>

## Article Info

### Article history:

Received 2022-12-23

Revised 2023-01-08

Accepted 2023-01-24

### Keyword:

Grid Search,  
Hyper Parameter Optimization,  
Random Search,  
SVM

## ABSTRACT

Classification is one of the important tasks in the field of Machine Learning. Classification can be viewed as an Optimization Problem (Optimization Problem) with the aim of finding the best model that can represent the relationship/pattern between data with labels. Support Vector Machine (SVM) Is an algorithm in Machine Learning used to solve problems such as Classification or Regression. The performance of the SVM algorithm is strongly influenced by parameters, for example error prediction in non-linear SVM results in parameters C and gamma. In this study, an analysis of the technique was carried out to obtain good parameter values using Grid search and Random Search on seven datasets. Evaluation is done by calculating the value of accuracy, memory usage and validity test time of the best model by the two techniques.



This is an open access article under the [CC-BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

## I. PENDAHULUAN

Klasifikasi merupakan salah satu tugas yang penting pada bidang *Machine Learning*. Permasalahan yang dapat diselesaikan dengan teknik klasifikasi ini beberapa di antaranya adalah: *data classification*, *text categorization*, *image classification*, *protein structure predictions* dan lain-lain [1]. Klasifikasi dapat dipandang sebagai Permasalahan Optimasi (*Optimization Problem*) dengan tujuan menemukan model terbaik yang dapat merepresentasikan hubungan/pola antar data dengan label [2]. Inti dari permasalahan klasifikasi adalah bagaimana suatu model dapat memisahkan data-data berdasarkan kelas/labelnya. Banyak metode-metode yang digunakan untuk menyelesaikan permasalahan klasifikasi ini seperti: *XGBoost*, *naive Bayes*, *decision tree* dan lain-lain. SVM Menurut beberapa penelitian dilaporkan memiliki performa yang baik untuk permasalahan klasifikasi dalam hal akurasi [1].

*Support Vector Machine* (SVM) dapat digunakan untuk menyelesaikan permasalahan seperti Klasifikasi atau Regresi. SVM bekerja dengan cara menemukan garis pemisah (*hyperplane*) terbaik yang memisahkan data-data yang bernilai positif jika berada pada kanan garis atau negatif jika berada pada kiri garis pemisah. SVM telah digunakan untuk

menyelesaikan berbagai masalah seperti: Klasifikasi, Regresi [4] bahkan Klasterisasi [5]. Pemilihan hyperparameter pada SVM sangat mempengaruhi performa dari algoritma ini. sebagai contoh untuk prediksi *error* pada *non-linear SVM* hasilnya sangat dipengaruhi oleh parameter C dan gamma [6].

Menurut [7] *Hyperparameters* adalah parameter-parameter yang tidak dapat diubah selama proses *training* suatu model. Untuk dapat menentukan *Hyperparameters* yang baik dapat dilakukan dengan cara manual *tuning*. Namun, cara ini membutuhkan pengetahuan dan wawasan yang cukup dalam mengenai parameter apa yang baik bagi model terhadap *dataset* tertentu. Hal ini membuat proses pencarian parameter yang optimal jika dilakukan secara manual membutuhkan banyak waktu [8]. Oleh karena itu penelitian ini bertujuan membandingkan metode yang digunakan untuk menentukan parameter terbaik dengan menggunakan *search algorithm* sebagai strateginya. *Search algorithm* yang digunakan pada penelitian ini adalah: *Grid Search* dan *Random Search*. Evaluasi yang dilakukan sebagai penentu mana metode yang lebih baik yaitu melalui hasil nilai akurasi dan juga *running time* dari tiap *search algorithm* yang diajukan dan *memory usage*.

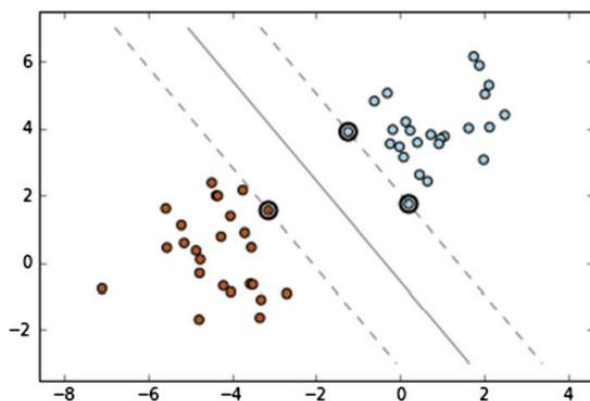
Penelitian terkait dengan permasalahan optimasi *Hyperparameters* dapat ditemukan pada penelitian yang

dilakukan oleh [9] yang membandingkan *Grid Search* dan metode *heuristic-search* yaitu *Genetic Algorithm* (GA) dari hasil penelitian tersebut menyatakan *Grid Search* memiliki performa, dan *running time* yang lebih baik dibandingkan dengan GA untuk *datasets* dengan skala yang tidak besar. Penelitian lain dilakukan [10] yang membandingkan teknik *hyperparameter tuning Random Search* dan *Particle Swarm operation* (PSO) dimana hasil dari *random search* sedikit lebih unggul dibandingkan dengan PSO. Oleh karena itu penelitian yang akan diajukan akan membandingkan metode *Gid Search* dan *Random Search* sebagai teknik *tunning hyperparameter*.

## II. TINJAUAN PUSTAKA

### A. Support Vector Machine

*Support Vector Machine* (SVM) yang dikenal sebagai *optimal margin classifier* [11] adalah suatu teknik klasifikasi dalam *machine learning* yang memanfaatkan label (supervised learning). Pertama kali diperkenalkan pada tahun 1992 oleh Vapnik dan Cortes[3]. SVM memetakan/mentransformasikan *input* vektor ke dimensi yang lebih tinggi di tempat optimal *hyperplane* dibuat dan juga dua buah *hyperplane* lain yang paralel dibuat di tiap sisi dari *optimal hyperplane* yang memisahkan data [2]. SVM memiliki tiga buah parameter yaitu *C*, *Gamma* dan *kernel*. Parameter tersebut dinamakan *Hyperparameters*.



Gambar 1. Ilustrasi Support Vector Machine (SVM)

### B. Grid Search and Random Search

*Grid Search* adalah salah satu algoritma yang dapat digunakan untuk menemukan parameter yang optimal bagi suatu model. algoritma ini bekerja dengan mencoba semua kombinasi yang mungkin berdasarkan pada nilai parameter yang telah disediakan oleh pengguna. metode ini menyimpan hasil parameter-parameter yang telah dikombinasikan ke dalam *grid* kemudian parameter terbaik dipilih dengan mencari parameter dengan nilai galat terkecil [12]. Berbeda dari *Grid Search* yang mencoba semua kombinasi parameter, *Random Search* bekerja dengan cara mengambil beberapa sampel secara acak dari nilai parameter yang ada lalu mengombinasikannya. *Random Search* berfokus pada

eksplorasi nilai parameter yang memiliki dampak yang signifikan pada performa model saja [13].

### C. Missing Value

*Missing value* adalah nilai yang hilang pada atribut suatu data. Hilangnya nilai pada *dataset* tersebut dapat disebabkan kesalahan pada saat proses *input* data. *Missing value* dapat menyebabkan menurunnya kualitas data yang berakibat pada penurunan pada akurasi klasifikasi [14]. Pada penelitian ini akan dilakukan penanganan *Missing value* dengan teknik Imputasi konvensional di mana *Missing value* diganti dengan nilai *central tendency* [15].

### D. Confusion Matrix

*Confusion Matrix* merupakan tabel klasifikasi yang menyatakan hasil perhitungan dari data yang diklasifikasi sebagai *benar* atau *salah* yang dibandingkan dengan label sebenarnya pada data yang telah diklasifikasi. Ilustrasi untuk tabel *Confusion matrix* dapat dilihat pada Tabel I.

TABEL I  
CONFUSION MATRIX

		Actual	
		True	False
Predicted	Positive	True Positive (TA)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Nilai dari *Confusion Matrix* tersebut nantinya akan digunakan untuk menghitung *accuracy*, *precision*, *recall* dan *f-measure* sebagai hasil nilai evaluasi dari suatu model.

$$Accuracy = \frac{TP + TN}{TP + FP + FN + TN} \quad (1)$$

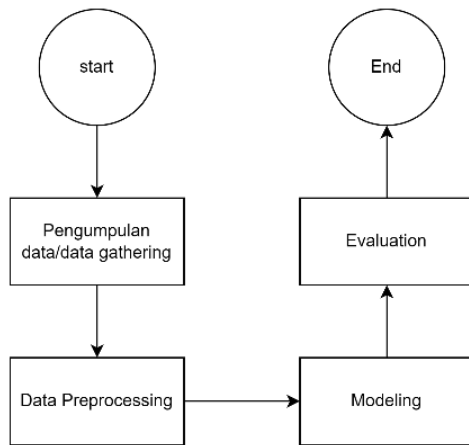
$$Precision = \frac{TP}{TP + FP} \quad (2)$$

$$Recall = \frac{TP}{TP + FN} \quad (3)$$

$$F - measure = \frac{2 * Recall * precision}{Recal + Precision} \quad (4)$$

## III. METODE PENELITIAN

Skema *Datasets* yang digunakan pada penelitian ini merupakan data publik untuk permasalahan klasifikasi. *Datasets* tersebut akan diambil melalui situs *kaggle.com* dengan nama-nama dataset yaitu *Water Quality*, *Liver*, *Water Potability*, *Winconsin Breast Cancer*, *Hear* dan *Iris*. Tahapan-tahapa yang dilakukan dalam penelitian ini adalah pengumpulan data, pemrosesan data, modeling dan evaluasi. Untuk membandingkan *grid search* dan *random search* penelitian ini menggunakan SVM sebagai algoritma untuk mengklasifikasikan datanya dan untuk mengevaluasi performa dari dua metode yang diajukan digunakan 3 alat ukur yaitu *accuracy*, *memory usage* dan *running time*.



Gambar 2. Skema penelitian

#### A. Pengumpulan data

Pada tahap ini dilakukan pengumpulan data sebagai media untuk perbandingan performa *Grid Search* dan *Random Search*. Data yang akan dikumpulkan merupakan data yang bersifat publik yang dapat diperoleh dari internet dan semua data tersebut merupakan data untuk permasalahan klasifikasi saja.

#### B. Data Preprocessing

Pada tahap ini data yang telah dikumpulkan akan diproses agar dapat diterima dengan baik oleh model. Pada tahap ini juga dilakukan pengecekan *missing value*, teknik Imputasi untuk mengisi *missing value* dan encoding label dari kategorik menjadi kelas dengan label 0,1 dan seterusnya jika kelas/label melebihi dua buah.

#### C. Modeling

Pada tahap ini dilakukan proses pemodelan algoritma *Machine Learning* (ML). Algoritma ML yang digunakan pada penelitian ini adalah SVM. Selain itu pada tahap ini juga dilakukan penentuan parameter-parameter yang akan digunakan sebagai *search space* oleh *Grid search* dan *Random search*. Nantinya parameter-parameter tersebut akan digunakan pada proses *hyperparameter tuning* di tahap *cross validation*. Pada tahap ini juga dilakukan *cross validation* dengan menggunakan metode K-Fold cross validation yaitu teknik yang digunakan untuk mendapatkan hasil akurasi yang lebih baik dibanding dengan metode testing biasa. teknik ini memecah set data menjadi K bagian yang tiap bagiannya akan digunakan untuk sebagai set pengujian [16]. Selama uji validitas ini juga dilakukan *tuning hyperparameter* yaitu proses mencari semua kombinasi parameter-parameter yang ada pada *search space* yang telah ditentukan kemudian mencari pasangan parameter dengan hasil model yang optimal berdasarkan nilai rata-rata nilai uji validitas terbaik.

#### D. Evaluasi

Pada tahap ini dilakukan evaluasi terhadap model-model dengan parameter terbaik. Evaluasi dilakukan dengan

memanfaatkan *confusion matrix* yang hasilnya akan digunakan untuk melihat nilai *accuracy*, *recall*, *precision*, *f-measure* dan juga fungsi dari python *standard library* yaitu *memit* dan *timeit* yang digunakan untuk menghitung *memory usage* dan *running time*.

### IV. HASIL DAN PEMBAHASAN

Hasil dari penelitian yang dilakukan adalah komparasi efektivitas penggunaan teknik *hyper parameter optimization* (*Grid Search* dan *Random Search*) dalam memecahkan permasalahan klasifikasi.

#### A. Pengumpulan data

*Dataset* yang digunakan pada penelitian ini sebanyak tujuh buah. Semuanya merupakan data publik yang dapat diperoleh dari situs *kaggle.com*. Semua data yang digunakan merupakan *dataset* untuk permasalahan klasifikasi. Beberapa di antara data yang digunakan merupakan data yang sudah bersih (Tidak memiliki *missing value*, duplikasi data). Namun, beberapa lainnya memiliki *missing value*. Untuk daftar nama *dataset* yang digunakan dapat dilihat pada Tabel II.

#### B. Data Preprocessing

Setelah semua *dataset* yang dibutuhkan sudah terkumpul kemudian dilakukan proses pengecekan *missing value*, duplikasi data ataupun data dengan tipe data yang tidak sesuai. Teknik Imputasi dilakukan dengan cara mengganti *missing value* dengan ketentuan berikut: Apabila data yang hilang memiliki tipe data kategori maka *missing value* akan diisi dengan modus atribut tersebut. Namun apabila data tersebut memiliki tipe data numerik, maka dilakukan pengecekan terdahulu untuk distribusi datanya apa bila distribusi atribut tersebut normal maka *missing value* akan diisi dengan rata-rata dari atribut tersebut. Sebaliknya *missing value* akan diisi dengan median dari atribut tersebut. Data yang sudah dikategorikan sebagai data yang bersih akan dilakukan pisah menjadi data training dan data testing dengan rasio 70/30.

TABEL II  
DATASETS

No	Nama Dataset	Missing value	Banyak sampel	Banyak atribut	Tipe data	class/label
1	Water Quality	Tidak	7,999	21	20 numerik, 1 kategori	0,1
2	Liver	Ya	4,238	16	semua numerik	0,1
3	Water Potability	Ya	3,276	10	semua numerik	0,1
4	Diabetes	Tidak	2,000	9	semua numerik	0,1
5	Wisconsin Breast Cancer	Tidak	569	32	31 numerik, 1 kategorik	M(malignant), B(benign)

6	Heart	Tidak	303	14	semua numerik	0,1
7	Iris	Tidak	150	6	5 numerik, 1 kategori	virginica, setosa, versicolor

*Dataset* diurutkan berdasarkan jumlah sampel terbanyak dan dari semua *dataset* tersebut hanya *dataset* iris yang mempunyai label lebih dari dua. Dimensi atau atribut tertinggi ada pada *dataset* kanker payudara dari Wisconsin dan dimensi terkecil yaitu dimiliki oleh *dataset* iris.

### C. Modeling

Sebelum pembuatan model dilakukan terlebih dahulu penentuan search space yaitu parameter-parameter yang akan digunakan pada proses tuning hyper parameter. Untuk daftar parameter-parameter yang digunakan sebagai search space dapat dilihat pada Tabel III. Setelah *search space* sudah ditentukan kemudian dilakukan proses pembuatan model. pada penelitian ini SVM yang digunakan untuk sebagai model adalah Support Vector Classification (SVC) dari package scikit-learn yang implementasinya berdasar pada LibSVM [17]. SVC adalah SVM yang digunakan untuk menyelesaikan permasalahan klasifikasi dan dapat menyelesaikan permasalahan *binary class classification* ataupun *multi-class classification*. Implementasi yang digunakan untuk melakukan *tuning hyperparameter* dan juga uji validitas yaitu dengan menggunakan *GridSearchCV* dan *RandomSearchCV* yang tersedia pada package *scikit-learn*.

TABEL III  
SEARCH SPACE

No	Parameter	Nilai
1	C	[0.1, 0.3, 0.7, 1, 3, 5, 7, 10, 12, 15, 18, 20]
2	Kernel	[ <i>rbf</i> , <i>linear</i> , <i>poly</i> , <i>sigmoid</i> ]
3	Gamma	[ <i>scale</i> , <i>auto</i> ]

Uji validitas dilakukan dengan cara memanggil fungsi fit pada *instnace* dari objek *GridSearchCV* atau *RandomSearchCV*. Pada uji validitas ini data-data dipisahkan menjadi *5-fold* atau 5 lipatan kemudian akan ada uji validitas dari tiap lipatannya. Jadi untuk satu kali proses *GridSearchCV* ataupun *RandomSearchCV* akan melakukan lima kali uji validitas. Namun berbeda dengan *RandomSearchCV* yang hanya mencoba beberapa kombinasi parameter secara acak. *GridSearchCV* harus mencoba semua kemungkinan parameter yang ada pada *search space*. Untuk hasil parameter terbaik dari tiap teknik dapat dilihat pada Tabel IV.

TABEL IV  
HASIL PARAMETER TERBAIK GRID SEARCH DAN RANDOM SEARCH

Datasets	GS			RS		
	C	gamma	kernel	C	gamma	kernel
Water Quality	3	auto	rbf	7	scale	rbf
Liver	3	scale	poly	1	scale	poly
Water Potability	1	auto	rbf	7	auto	rbf
Diabetes	20	scale	rbf	15	scale	rbf
Wisconsin Breast Cancer	3	auto	rbf	3	auto	rbf
Heart	0.3	scale	rbf	20	auto	rbf
Iris	3	scale	linear	3	scale	rbf

### D. Evaluation

Setelah model dengan parameter terbaik didapatkan kemudian dilakukan evaluasi dengan cara menghitung nilai *accuracy*, *precision*, *recall* dan *F-measure* yang perhitungannya dapat dilihat pada persamaan (1), (2), (3) dan (4) Hasil perhitungannya dapat dilihat pada Tabel V.

TABEL V  
HASIL ACCURACY, PRECISION, RECALL DAN F-MEASURE

SVM	Dataset	Accuracy	Precision	Recall	F-measure
Grid Search	Water Quality	95%	92%	81%	85%
	Liver	86%	82%	86%	82%
	Water Potability	67%	68%	67%	63%
	Diabetes	88%	87%	86%	87%
	Wisconsin Breast Cancer	98%	98%	98%	98%
	Heart	84%	84%	83%	83%
	Iris	93%	94%	93%	93%
Random Search	Water Quality	95%	91%	83%	87%
	Liver	86%	81%	86%	81%
	Water Potability	66%	65%	61%	61%
	Diabetes	87%	86%	85%	86%
	Wisconsin Breast Cancer	98%	86%	85%	86%
	Heart	96%	96%	95%	96%
	Iris	96%	96%	96%	96%

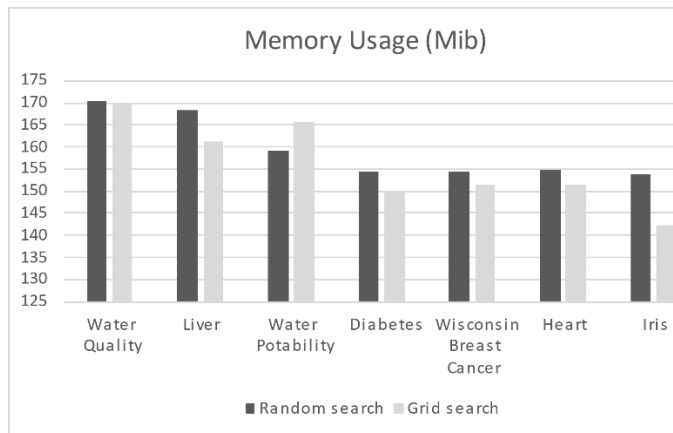
Berdasarkan data yang diperoleh pada Tabel V untuk perbedaan nilai akurasi yang paling signifikan terdapat pada dataset Heart dengan selisih 12% di mana Random search

berhasil membuat prediksi lebih baik dibandingkan dengan Grid search untuk kasus dataset tersebut.

Evaluasi untuk penggunaan memori (*memory usage*) dan lamanya waktu proses (*running time*). Saat uji validitas juga dilakukan untuk menemukan metode yang efektif dan juga efisien dalam menyelesaikan permasalahan klasifikasi. Proses penghitungan penggunaan memori dan waktu ini dilakukan pada perangkat dengan spesifikasi AMD Ryzen 4600H dengan RAM 16GB DDR4 dengan suhu ruangan tanpa AC (~25-30 derajat Celcius) dikarenakan perhitungan performa seperti *running time* yang sangat sensitif terhadap variabel lain seperti spesifikasi laptop atau kondisi ruang oleh karena itu mungkin saja hasil evaluasi ini dapat berbeda dengan hasil evaluasi perangkat dan kondisi lain walaupun dengan dataset dan metode yang sama.

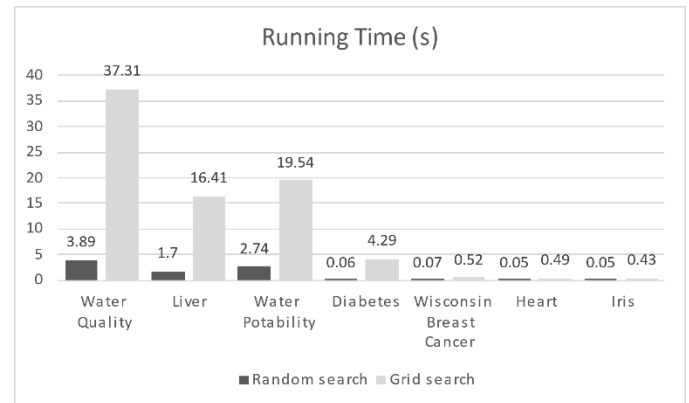
Untuk menghitung memori dilakukan dengan menggunakan *magic line code %memit* yang tersedia pada jupyter notebook. Grafik untuk penggunaan memori dapat dilihat pada gambar 3.

Kemudian dilakukan evaluasi terhadap *running time* untuk proses uji validitas Grid search dan Random search. Untuk mendapatkan hasil yang lebih akurat maka proses uji validitas diulang sebanyak 30 kali untuk setiap dataset kemudian diambil rata-rata sebagai representasi lamanya waktu menjalankan proses tersebut. perhitungan waktu dilakukan dengan menggunakan *magic line code %timeit*. Grafik lamanya waktu dari tiap dataset dapat dilihat pada gambar 4.



Gambar 3 Memory usage Grid search dan Random search

Berdasarkan Grafik pada Gambar 3 dapat dilihat untuk dataset dengan sampel yang kecil Grid search cenderung unggul dalam penggunaan memori. Namun, ketika sampel data semakin banyak maka performa keduanya cenderung tidak memiliki perbedaan yang signifikan.



Gambar 4. Running time Grid search dan Random search

Dapat dilihat pada Gambar 4 dalam segi kecepatan menyelesaikan proses Cross Validation atau uji validitas Random search jauh mengungguli performa Grid search pada setiap dataset bahkan pada data set dengan jumlah sampel yang banyak Random search memiliki kecepatan 10 kali lebih cepat dibandingkan dengan Grid search namun dengan hasil akurasi yang sangat bersaing berdasarkan Tabel V.

### V. KESIMPULAN

Berdasarkan hasil percobaan penelitian ini dapat disimpulkan dari sisi nilai akurasi baik Grid search maupun Random search tidak memiliki perbedaan yang signifikan untuk nilai accuracy dimana nilai rata-rata dari kedua metode tersebut yaitu 84%. Untuk pemakaian memori pun tidak terdapat perbedaan yang signifikan walaupun penggunaan memori untuk Random search secara rata-rata lebih tinggi 3,4 Mebibytes jika dibandingkan Grid Search.

Sedangkan perbedaan yang signifikan dapat ditemukan pada lama waktu proses uji validitas dari kedua metode tersebut. Random search memiliki *running time* yang jauh lebih unggul. Sebagai contoh Random search hampir 10 kali lebih cepat dibandingkan dengan Grid Search untuk datasets Water Quality. Hal ini tentu dikarenakan perbedaan jumlah kombinasi parameter yang dicoba pada proses uji validitas. Grid search harus melakukan semua kemungkinan kombinasi dari parameter-parameter yang telah ditentukan oleh karena itu untuk sekali proses uji validitas Grid search membutuhkan 510 kali fitting sedangkan untuk Random search hanya melakukan pencarian 10 kombinasi parameter dari semua kemungkinan parameter yang ada oleh karena itu untuk tiap uji validitas Random search hanya melakukan 50 kali fitting. Walaupun demikian Random search mampu bersaing dengan Grid search dari segi akurasi.

Untuk penelitian kedepannya penulis berharap penelitian ini dapat dilakukan dengan jumlah dataset yang lebih banyak dan juga merupakan dataset berdimensi tinggi.

### DAFTAR PUSTAKA

[1] D. K Srivastave and L. Bhambu, "Data Classification Using Support Vector Machine," *J. Theor. Appl. Inf. Technol.*, vol. 1, pp. 3–6, 2010, doi: 10.1109/ICTAI.2010.9.

- [2] F. E. B. Otero, A. A. Freitas, and C. G. Johnson, "Inducing decision trees with an ant colony optimization algorithm," *Appl. Soft Comput. J.*, vol. 12, no. 11, pp. 3615–3626, 2012, doi: 10.1016/j.asoc.2012.05.028.
- [3] C. Cortez and V. Vapnik, "Support-Vector Networks," *Mach. Learn.*, vol. 7, no. 5, pp. 63–72, 1995, doi: 10.1109/64.163674.
- [4] R. A. Rizal, I. S. Girsang, and S. A. Prasetyo, "Klasifikasi Wajah Menggunakan Support Vector Machine (SVM) Reyhan," *Ris. dan E-Jurnal Manaj. Inform. Komput.*, vol. 3, no. 2, pp. 275–280, 2019, doi: 10.33480/pilar.v15i2.693.
- [5] Y. Wang *et al.*, "Minimum distribution support vector clustering," *Entropy*, vol. 23, no. 11, 2021, doi: 10.3390/e23111473.
- [6] A. Klein, S. Falkner, S. Bartels, P. Hennig, and F. Hutter, "Fast Bayesian optimization of machine learning hyperparameters on large datasets," *Proc. 20th Int. Conf. Artif. Intell. Stat. AISTATS 2017*, vol. 11, no. June, pp. 4945–4968, 2017.
- [7] T. Yu and H. Zhu, "Hyper-Parameter Optimization: A Review of Algorithms and Applications," pp. 1–56, 2020, [Online]. Available: <http://arxiv.org/abs/2003.05689>.
- [8] I. Syarif, A. Prugel-Bennett, and G. Wills, "SVM Parameter Optimization using Grid Search and Genetic Algorithm to Improve Classification Performance," *TELKOMNIKA (Telecommunication Comput. Electron. Control.)*, vol. 14, no. 4, p. 1502, 2016, doi: 10.12928/telkomnika.v14i4.3956.
- [9] S. Yuanyuan, W. Yongming, G. Lili, M. Zhongsong, and J. Shan, "The comparison of optimizing SVM by GA and grid search," *ICEMI 2017 - Proc. IEEE 13th Int. Conf. Electron. Meas. Instruments*, vol. 2018-Janua, pp. 354–360, 2017, doi: 10.1109/ICEMI.2017.8265815.
- [10] T. Horváth, R. G. Mantovani, and A. C. P. L. F. de Carvalho, "Effects of random sampling on SVM hyper-parameter tuning," *Adv. Intell. Syst. Comput.*, vol. 557, pp. 268–278, 2017, doi: 10.1007/978-3-319-53480-0\_27.
- [11] V. K. Chauhan, K. Dahiya, and A. Sharma, "Problem formulations and solvers in linear SVM: a review," *Artif. Intell. Rev.*, vol. 52, no. 2, pp. 803–855, 2019, doi: 10.1007/s10462-018-9614-6.
- [12] G. H. Saputra, A. H. Wigena, and B. Sartono, "Penggunaan Support Vector Regression Dalam Pemodelan Indeks Saham Syariah Indonesia Dengan Algoritme Grid Search," *Indones. J. Stat. Its Appl.*, vol. 3, no. 2, pp. 148–160, 2019, doi: 10.29244/ijsa.v3i2.172.
- [13] R. Valarmathi and T. Sheela, "Heart disease prediction using hyper parameter optimization (HPO) tuning," *Biomed. Signal Process. Control*, vol. 70, no. August, p. 103033, 2021, doi: 10.1016/j.bspc.2021.103033.
- [14] Moch. Lutfi and Mochamad Hasyim, "Penanganan Data Missing Value Pada Kualitas Produksi Jagung Dengan Menggunakan Metode K-Nn Imputation Pada Algoritma C4.5," *J. Resist. (Rekayasa Sist. Komputer)*, vol. 2, no. 2, pp. 89–104, 2019, doi: 10.31598/jurnalresistor.v2i2.427.
- [15] A. R. Alfarisi, H. Tjandrasa, and I. Arieshanti, "Perbandingan Performa antara Imputasi Metode Konvensional dan Imputasi dengan Algoritma," *Mach. Learn.*, vol. 2, no. 1, pp. 1–4, 2013.
- [16] A. Peryanto, A. Yudhana, and R. Umar, "Klasifikasi Citra Menggunakan Convolutional Neural Network dan K Fold Cross Validation," *J. Appl. Informatics Comput.*, vol. 4, no. 1, pp. 45–51, 2020, doi: 10.30871/jaic.v4i1.2017.
- [17] P. Fabian *et al.*, "Scikit-learn: Machine Learning in Python Fabian," *Mach. Learn. Res.*, vol. 12, no. 9, pp. 2825–2830, 2011, doi: 10.1289/EHP4713.