

Analisis Perbandingan Performa Metode Klasifikasi pada Dataset Multiclass Citra Busur Panah

Performance Comparison Analysis of Classification Methods on the Multiclass Dataset of Bows

Huzain Azis¹, Fadhila Tangguh Admojo², Erma Susanti³

¹Teknik Informatika, Fakultas Ilmu Komputer, Universitas Muslim Indonesia

²Teknik Informatika, STMIK Palcomtech

^{1,2,3}Informatika, Fakultas Teknologi Industri, Institut Sains & Teknologi, AKPRIND

E-mail: ¹huzain.azis@umi.ac.id, ²Fadhila.tangguh@gmail.com, ³erma@akprind.ac.id

Abstrak

Pengujian performa berbagai metode pada sebuah dataset merupakan salah satu cara dalam penetapan metode klasifikasi yang tepat, masalah yang diangkat pada penelitian ini adalah bagaimana membandingkan performa beberapa metode klasifikasi dalam mengelola dataset yang memiliki lebih dari dua label (*multiclass*). Penelitian ini fokus membandingkan hasil performa tujuh metode klasifikasi yaitu *K-Nearest Neighbor* (knn), *Naive Bayes Classifier* (nbc), *Support Vector machine* (svm), *Neural Network* (nn), *Random Forest Classifier* (rfc), *Ada Boost Classifier* (abc) dan *Quadratic Discriminant Analysis* (qdc). Objek pada penelitian ini berupa dataset *multiclass* yaitu dataset citra busur panah, serta performa yang diukur yaitu seluruh nilai *cross-validation* dari akurasi, presisi, *recall* dan *f-measure*. Hasil pada penelitian ini menunjukkan bahwa seluruh metode tidak memperoleh performa yang cukup baik, dan menunjukkan bahwa beberapa metode yang memiliki akurasi yang tinggi tidak menjadi penentu menjadi metode yang baik dikarenakan setelah penerapan *cross-validation* dan visualisasi boxplot ditemukan beberapa nilai akurasi tinggi yang merupakan nilai tidak wajar atau outlier. Kesimpulan menunjukkan metode svm memiliki performa yang lebih baik dibandingkan dengan enam metode lainnya pada kasus dataset multiclass citra busur panah.

Kata kunci: analisis performa, klasifikasi, dataset multiclass, cross-validation, perbandingan metode

Abstract

Testing the performance of various methods on a dataset is one way in determining the appropriate classification method. The problem raised in this study is how to compare the performance of several classification methods in managing datasets that have more than two labels (multiclass). This study focuses on analyzing the results of performance seven classification methods namely K-Nearest Neighbor (knn), Naive Bayes Classifier (nbc), Support Vector Machine (svm), Neural Network (nn), Random Forest Classifier (rfc), Ada Boost Classifier (abc) and Quadratic Discriminant Analysis (qdc). The object is a multiclass dataset, which is the bow image dataset, and the performance measured in this study is all the cross-validation values of accuracy, precision, recall and f-measure. The results of this study indicate that all methods do not get good enough performance, and show that some plans that have high accuracy are not decisive to be suitable methods because after the application of cross-validation and boxplot visualization. Several high accuracy values are identified to be unnatural or outlier, the conclusion shows that the svm method has a better performance compared to the six other ways in the case of the bow multiclass dataset.

Keywords: performance analysis, classification, multiclass dataset, cross-validation, method comparison

1. PENDAHULUAN

Terdapat banyak metode klasifikasi dalam *supervised learning* pada *machine learning*, diantaranya adalah *K-Nearest Neighbor* (knn), *Naive Bayes Classifier* (nbc), *Support Vector Machine* (svm), *Neural Network* (nn), *Random Forest Classifier* (rfc), *Ada Boost Classifier* (abc), serta *Quadratic Discriminant Analysis* (qda). Metode tersebut memiliki kelebihan serta kekurangannya masing-masing, salah satu faktor menjadi keunggulan metode klasifikasi tersebut dilihat dari bagaimana metode tersebut mengolah objek dataset. Dataset *multiclass* merupakan dataset unik yang memiliki lebih dari dua label. Performa sebuah metode dapat diukur melalui pengujian nilai akurasi, presisi, *recall* dan *f-measure*, serta dengan menerapkan *cross-validation* pada pengujian performa maka kekokohan dari performa tersebut dapat terlihat.

Saat ini terdapat banyak penelitian mengenai penerapan atau perbandingan metode klasifikasi serta penerapannya di berbagai jenis dataset[1]. Tabel 1 menunjukkan perbandingan ruang lingkup penelitian sebelumnya dengan penelitian ini. Penelitian ini mencoba mengukur serta membandingkan performa tujuh metode klasifikasi yaitu knn, nbc, svm, nn, rfc, abc dan qda pada dataset *multiclass* citra busur panah yang terdiri dari empat label dimana data yang digunakan untuk klasifikasi merupakan *output* dari deteksi tepi serta ekstraksi fitur citra.

Tabel 1 Gap Penelitian Terkait

No	Tahun Penelitian	Metode	Multiclass Dataset	Cross-validation	Performa			
					akurasi	presisi	recall	F-measure
1	[Penelitian ini] 2020	knn, nbc, svm, nn, rfc, abc dan qda	✓	✓	✓	✓	✓	✓
2	[2] 2020	knn	✓	-	✓	✓	✓	✓
3	[3] 2019	knn	✓	-	✓	✓	✓	✓
4	[4] 2017	knn	-	-	✓	-	-	-
		nn	-	-	✓	-	-	-
		lr	-	-	✓	-	-	-
		nb	-	-	✓	-	-	-
5	[5] 2016	svm	-	-	✓	-	-	-
		dt	-	✓	✓	-	-	-
		knn	-	✓	✓	-	-	-
6	[6] 2015	nb	-	✓	✓	-	-	-
		dt	✓	-	✓	-	-	-
7	[7] 2015	Svmm	✓	-	✓	-	-	-
		Knn	-	✓	✓	-	-	-
		Nbc	-	✓	✓	-	-	-
		j48	-	✓	✓	-	-	-
		Svm	-	✓	✓	-	-	-

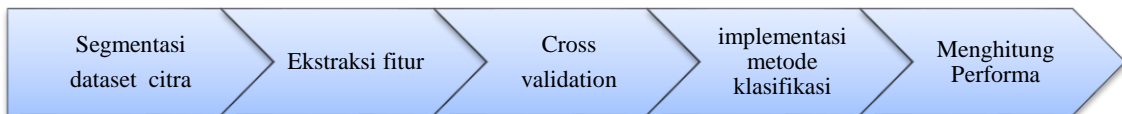
Tabel 1 menunjukkan bahwa keterkaitan serta gap penelitian ini dengan penelitian sebelumnya yaitu terdapat pada jumlah metode klasifikasi yang dibandingkan, jenis dataset yang digunakan (*multiclass* atau bukan), penerapan *crossvalidation*, dan jenis performa yang diukur (akurasi, presisi dan recall). Tahapan penelitian ini dan penelitian-penelitian sebelumnya jelas akan berbeda tergantung pada metode apa yang dibandingkan serta objek dataset yang digunakan.

Tahap yang dilakukan pada penelitian ini adalah deteksi tepi citra menggunakan *canny*, ekstraksi fitur menggunakan *humoment*, *splitting* data *training* dan data *testing* sebanyak 5 *fold* menggunakan *cross-validation*, menerapkan tujuh metode klasifikasi, serta menghitung performa masing-masing metode yang akan diuji. Perbedaan penelitian ini dengan penelitian sebelumnya yaitu dari jenis dataset yang digunakan, jumlah metode yang dibandingkan, serta penerapan *cross-validation* untuk uji kekokohan performa.

2. METODE PENELITIAN

Dataset yang digunakan pada penelitian ini merupakan dataset dengan format citra, yang dikumpulkan dengan melakukan proses *image crawling* pada *google image searcher*, jumlah label (*class*) sebanyak empat yaitu gambar dengan label *recurve bow*, *longbow*, *compound bow*, dan *crossbow*.

Jumlah keseluruhan gambar dataset sebanyak 800 yang dapat diunduh Pada repositori dataset kaggle. Adapun tahap analisis yang dilakukan pada penelitian ini terdiri dari segmentasi, ekstraksi fitur, *cross-validation*, klasifikasi dan menghitung performa, Gambar 1 menunjukkan tahapan analisis pada penelitian ini.



Gambar 1 Tahapan analisis

2.1. Segmentasi dataset citra

Tahap pertama adalah segmentasi citra, pada tahap ini terdapat 2 sub tahapan inti yaitu melakukan deteksi tepi menggunakan *canny* serta *reduce noise* menggunakan *gaussianblur* [6][7], persamaan 1 menunjukkan persamaan deteksi tepi *canny* dan Tabel 2 menunjukkan implementasi deteksi tepi *canny* pada dataset citra.

$$EG(G) = \sqrt{G_x^2 + G_y^2} \quad (1)$$

$$Angel(\theta) = \tan^{-1}\left(\frac{G_y}{G_x}\right)$$

Tabel 2. Implementasi Segmentasi

Citra Asli	Canny	Canny + gaussian blur

2.2. Ekstraksi fitur

Tahap kedua adalah ekstraksi fitur, pada tahap ini metode yang digunakan adalah ekstraksi fitur *humoment*, dimana akan dilakukan proses pengambilan tujuh fitur pada setiap gambar.

Persamaan 2 menunjukkan formula yang digunakan untuk mengambil fitur pada gambar. fitur yang diekstrak pada setiap gambar sebanyak 7 fitur [8][9], sedangkan Tabel 3 menunjukkan lima contoh hasil ekstraksi fitur pada sebuah gambar yang telah melalui

segmenteasi sebelumnya, alasan penggunaan *humoment* dengan pertimbangan bahwa nilai *humoment invariant* baik terhadap skala ukuran objek dan rotasi, menimbang bahwa dataset yg diolah merupakan dataset yang memiliki perbedaan yang berbeda ukuran serta rotasinya.

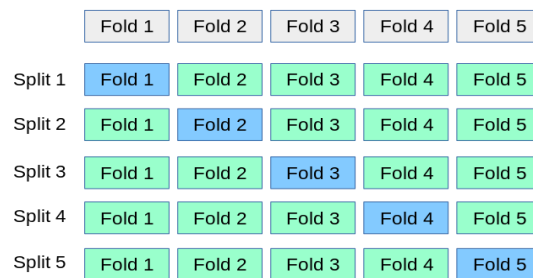
$$\begin{aligned}
 \emptyset_0 &= \eta_{20} + \eta_{02} \\
 \emptyset_1 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{21}^2 \\
 \emptyset_2 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \mu_{03})^2 \\
 \emptyset_3 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \mu_{03})^2 \\
 \emptyset_4 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \\
 &\quad + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} - \eta_{03})^2 \\
 &\quad - (\eta_{21} + \eta_{30})^2] \\
 \emptyset_5 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{30})^2] + 4\eta_{11} \\
 \emptyset_6 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) [(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] \times [3(\eta_{30} + \eta_{12})^2 - (\eta_{21} \\
 &\quad + \eta_{30})^2]
 \end{aligned} \tag{2}$$

Tabel 3. Implementasi Ekstraksi fitur Humoment

\emptyset_0	\emptyset_1	\emptyset_2	\emptyset_3	\emptyset_4	\emptyset_5	\emptyset_6
0.01275	1.31E-04	2.54E-07	4.76E-08	2.58E-15	4.02E-12	-4.55E-15
0.009296	5.01E-07	1.18E-08	2.86E-09	-1.60E-17	-1.93E-12	-4.37E-18
0.00663	1.20E-07	1.24E-09	3.91E-10	-2.20E-19	1.21E-13	-1.60E-19
0.012702	1.42E-04	1.32E-07	2.00E-08	8.03E-17	-7.07E-11	1.03E-15
0.009711	7.99E-05	6.27E-08	3.48E-08	1.60E-15	2.54E-10	2.81E-16

2.3. Cross-validation

Tahap ketiga adalah tahap proses *cross-validation*, sesuai dengan defenisinya bahwa *cross-validation* merupakan metode statistik yang ampuh diterapkan untuk mengevaluasi kinerja model maupun algoritma dimana data dipilah menjadi dua bagian yaitu data proses *training* dan data *testing*[10], [11]. Model atau algoritma dilatih oleh subset pembelajaran dan divalidasi oleh subset validasi agar pengambilan nilai performa dapat disimpulkan dengan baik maka setiap data pada keseluruhan dataset dirancang agar bergilir untuk berperan sebagai data *training* dan data *testing*, penerapan *cross-validation* pada tahap ini dibagi menjadi 5 bagian atau juga disebut sebagai *k-5 fold validation*. Gambar 2 menunjukkan ilustrasi *cross-validation* pada penelitian ini.



Gambar 2 Implementasi *cross-validation*

2.4. Implementasi metode klasifikasi

Tahap selanjutnya adalah melakukan *testing* pada tujuh metode klasifikasi yaitu knn, nbc, svm, nn, rfc, abc dan qda. Metode pertama yang diuji adalah metode knn dimana metode tersebut memutuskan hasil klasifikasi berdasarkan dominan tetangga yang terdekat. Persamaan 3 menunjukkan rumus penerapan metode knn yang mana algoritma jarak yang digunakan pada metode knn adalah *eucliden*[3][2].

$$euc = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (3)$$

Metode klasifikasi kedua adalah *Support Vector Machine* (svm), pada dasarnya terdapat tiga implementasi berbeda dari Regresi Vektor: SVR, NuSVR dan LinearSVR [10], [11]. LinearSVR menyediakan implementasi yang lebih cepat daripada SVR tetapi linearSVR hanya fokus pada kernel linier, sementara NuSVR mengimplementasikan formulasi yang sedikit berbeda dari SVR dan LinearSVR. Penerapan metode svm pada penelitian ini menggunakan LinearSVR yang ditunjukkan pada persamaan 4.

$$\text{Min}_{w,b} \frac{1}{2} w^t w + C \sum_{i=1} \max(0, |y_i - w^t \phi(x_i) + b| - \epsilon) \quad (4)$$

Metode yang ketiga yang adalah *Multi-Layer Perceptron* (MLP) dalam hal ini adalah salah satu pemodelan dalam *Neural Network* [12], formulanya memberikan serangkaian training dimana $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$ dimana $x_1 \in R^n$ dan $y_1 \in \{0,1\}$, satu *hidden layer* satu *hidden neuron* MLP mempelajari fungsinya $f(x) = W_2 g(W_1^T x + b_1) + b_2$ dimana $W_1 \in R^m$ dan $W_2, b_1, b_2 \in R$ adalah parameter model. Namun jika terdapat lebih dari 2 class maka $f(x)$ itu sendiri akan menjadi vektor ukuran (n_classes), dan akan melewati fungsi softmax [13], Formula softmax ditunjukkan pada persamaan 5.

$$\text{softmax}(z)_1 = \frac{\exp(z_i)}{\sum_{l=1}^k \exp(z_l)} \quad (5)$$

Metode keempat adalah *Naïve Bayes Classifier* (nbc), metode yang sangat populer yang melakukan proses klasifikasinya dengan pendekatan probabilitas. dikarenakan data yang diolah merupakan data *numeric* maka fungsi yang diterapkan menggunakan *gaussian naïve bayes* [14]–[16], persamaan 6 menunjukkan fungsi yang digunakan pada *gaussian naïve bayes*.

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} \exp\left(-\frac{(x_i - \mu_y)^2}{2\sigma_y^2}\right) \quad (6)$$

Tiga metode lainnya yaitu rfc, abc dan qda, dimana *Random forests* adalah kombinasi 3 pohon keputusan yang mana setiap pohon bertanggung jawab pada setiap nilai dari random vector yang diberikan begitupun persebarannya akan diberikan kepada pohon keputusan lainnya[10][17]. Sama halnya dengan rfc, abc juga merupakan *ensemble method* yang juga dapat digunakan untuk melakukan proses klasifikasi dataset *multiclass*[18]. Sedangkan qda, merupakan pengklasifikasi dengan batas keputusan kuadratik, dihasilkan dengan menyesuaikan kepadatan *class* dengan data dan menggunakan aturan Bayes[19][20].

2.5. Menghitung performa

Tahap terakhir setelah penerapan metode klasifikasi adalah menghitung performa, adapun persamaan performa menggunakan *confusion matrix* yang mana untuk membantu menghitung nilai akurasi, presisi, *recall* dan *f-measure*[21][22][23]. Tabel 4. menunjukkan model *confusion matrix* yang diterapkan untuk kasus 4 label dimana *Re* untuk label recursive bow, *Lo* untuk Label Long bow, *Co* untuk label Compound Bow dan *Cr* untuk Cross bow.

Tabel 4. Confusion Matrix Multiclass

		classifier			
		Re	Lo	Co	Cr
Actual	Re	ReRe	ReLo	ReCo	ReCr
	Lo	LoRe	LoLo	LoCo	LoCr
	Co	CoRe	CoLo	CoCo	CoCr
	Cr	CrRe	CrLo	CrCo	CrCr

Persamaan yang digunakan untuk menghitung akurasi ditunjukkan pada persamaan 7, penerapan perhitungan akurasi yang digunakan pada penelitian ini adalah *balanced-accuracy* (ba) dimana berfungsi untuk menangani *multiclass* klasifikasi dengan data yang tidak seimbang[24], [25].

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (7)$$

$$b.a = 1/2(\frac{TP}{TP + FN} + \frac{TN}{TN + FP})$$

Proses klasifikasi *multiclass*, perhitungan performa presisi, recall, dan f-measure dapat diterapkan pada setiap label secara independen [18][19]. presisi menggunakan persamaan 8, persamaan 9 menunjukkan perhitungan performa *recall* dan persamaan 10 untuk *f-measure*[28][29].

$$precision = \frac{TP}{TP + FP} \quad (8)$$

$$recall = \frac{TP}{TP + FN} \quad (9)$$

$$f - measure = \frac{precision * recall}{precision + recall} \quad (10)$$

3. HASIL DAN PEMBAHASAN

Setelah melalui keseluruhan Tahapan, maka diperoleh nilai performa untuk semua metode, Tabel 4. Memperlihatkan performa pada setiap metode yang dianalisis pada penelitian ini, performa yang diperlihatkan terdiri dari akurasi, presisi, recall, dan f-measure. dimana nilai tersebut diambil dari nilai tertinggi dari pada setiap simulasi data *5-fold crossvalidation* yang telah diatur sebelumnya. Nilai performa pada setiap *crossvalidation* diilustrasikan pada Gambar 4, dimana ilustrasi visualisasi yang diperlihatkan dalam bentuk *linegraph*

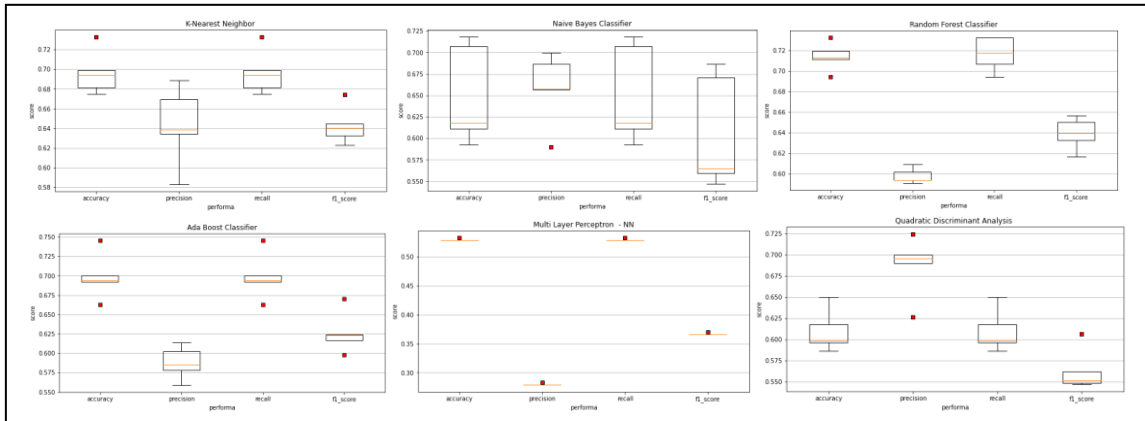
Tabel 4. Hasil Pengujian

No	Metode	Akurasi	Presisi	Recall	F-measure
1	K-Nearest Neighbor (k-nn),	73%	68%	73%	67%
2	Naive Bayes Classifier (nbc),	71%	69%	71%	68%
3	Support Vector Machine (svm),	73%	61%	73%	66%
4	Neural Netowork (nn),	53%	28%	53%	26%
5	Random Forest Classifier (rfc),	73%	60%	73%	65%
6	Ada Boost Classifier (abc),	74%	61%	74%	67%
7	Quadratic Discriminant Analysis (qda).	64%	68%	64%	60%

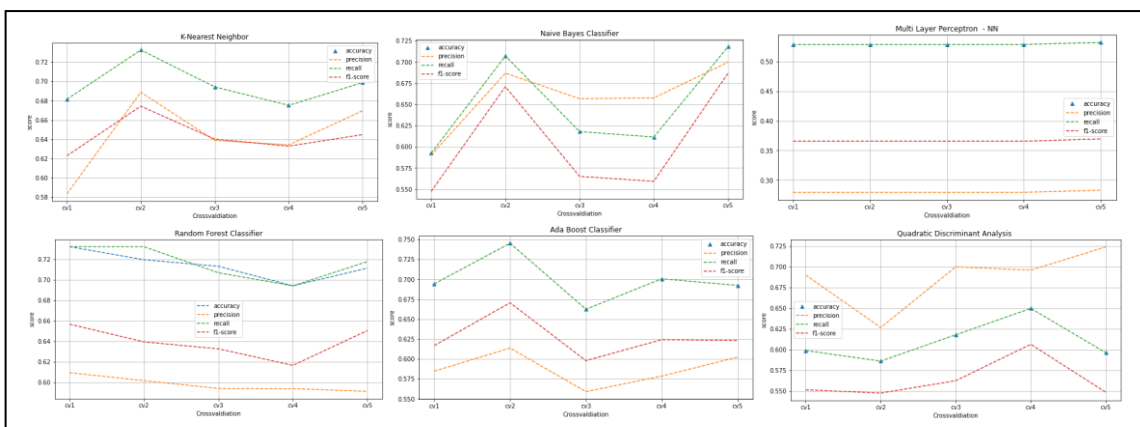
Gambar 3 menunjukkan visualisasi setiap metode dalam format boxplot. Terlihat bahwa setiap metode ilustrasi menunjukkan terdapat nilai outlier, baik terdapat di akurasi, presisi, recall maupun f-measure. Performa sebuah metode tidak cukup jika diukur dari nilai akurasi

saja, namun juga perlu dilihat nilai presisi, recall maupun f-measure. Selain itu dengan melakukan simulasi *crossvalidation* juga dapat membuat nilai performa yang tidak wajar atau juga disebut anomali(*outlier*).

Berdasarkan apa yang ditunjukkan pada Tabel 4 dan jika disandingkan dengan Gambar 3, bisa dilihat bahwa Meskipun Ada Boost Classifier (abc) memiliki nilai akurasi tertinggi sebesar 74%, namun visualisasi dalam bentuk *boxplot* terlihat bahwa nilai akurasi yang tertinggi tersebut merupakan nilai yang tidak wajar atau disebut *outlier* dari nilai hasil *crossvalidation* lainnya. Begitupula terlihat pada metode *knn* dan *rfc* dengan akurasi 73% juga merupakan nilai *outlier* diantara nilai *crossvalidation* lainnya

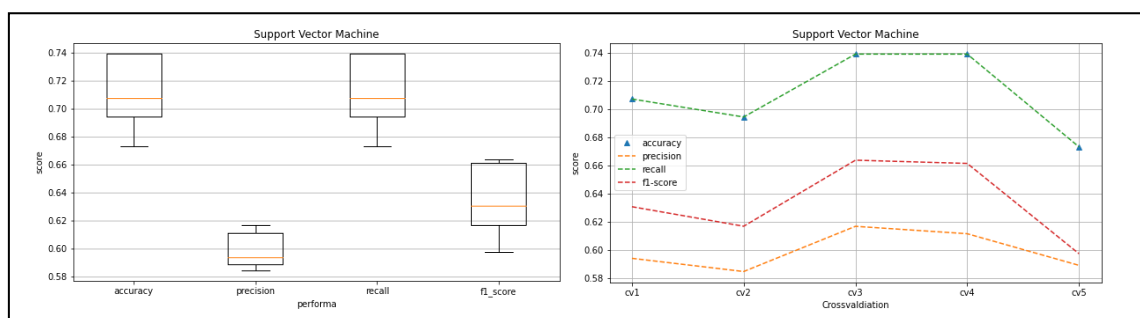


Gambar 3. Ilustrasi visualisasi performa dalam *boxplot*



Gambar 4. *Boxlot* dan *linegraph* metode *svm*

Gambar 4 menunjukkan visualisasi performa metode pada *5-fold crossvalidation*. Sedangkan Gambar 5 menunjukkan *boxplot* serta *linegraph* metode *svm*, Nilai performa yang menjadi *outlier* tidak dapat digunakan sebagai nilai performa yang baik, sehingga penetapan nilai terbaik pada percobaan ini jatuh pada metode *svm* yang memiliki nilai akurasi 73%, jika dibandingkan dengan nilai performa yang lain terlihat bahwa nilai performa pada metode *svm* tidak memiliki satupun nilai *outlier*, baik pada performa akurasi, presisi, *recall* dan *F-Measure*. Sehingga alasan tersebut membuat metode *svm* menjadi metode yang lebih baik dibanding metode lainnya pada studi kasus klasifikasi dataset *multiclass* citra busur panah



Gambar 5 Boxlot dan linegraph metode svm

4. KESIMPULAN DAN SARAN

Setelah melalui berbagai simulasi dapat disimpulkan bahwa meskipun secara keseluruhan semua metode tidak mendapatkan nilai performa yang sangat baik, namun metode svm memiliki nilai performa yang lebih baik dibandingkan metode klasifikasi lainnya dalam studi kasus klasifikasi dataset multiclass busur panah, Serta dapat dibuktikan bahwa jika hanya dengan melihat nilai akurasi maka tidak cukup untuk menunjukkan bagus maupun tidaknya performa metode tersebut. Banyak pertimbangan yang dapat dilakukan dalam menentukan metode yang tepat yaitu dengan menghitung keseimbangan hasil pengujian seperti presisi, recall, f-measure serta dengan melakukan uji coba berbagai simulasi pembagian data ujiannya menggunakan crossvalidation.

DAFTAR PUSTAKA

- [1] N. Fadhillah, H. Azis, and D. Lantara, "Validasi Pencarian Kata Kunci Menggunakan Algoritma Levenshtein Distance Berdasarkan Metode Approximate String Matching," *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 3–7, 2018.
- [2] Hasran, "Klasifikasi Penyakit Jantung Menggunakan Metode K-Nearest Neighbor," *Indones. J. Data Sci.*, vol. 1, no. 1, pp. 1–4, 2020.
- [3] M. M. Baharuddin, T. Hasanuddin, and H. Azis, "Analisis Performa Metode K-Nearest Neighbor untuk Identifikasi Jenis Kaca," *Ilk. J. Ilm.*, vol. 11, no. 28, pp. 269–274, 2019.
- [4] A. Ilham, "Komparasi Algoritma Klasifikasi Dengan Pendekatan Level Data Untuk Menangani Data Kelas Tidak Seimbang," *J. Ilm. Ilmu Komput.*, vol. 3, no. 1, pp. 9–14, 2017.
- [5] M. Yusa, E. Utami, and E. T. Luthfi, "Analisis Komparatif Evaluasi Performa Algoritma Klasifikasi pada Readmisi Pasien Diabetes," *J. Buana Inform.*, vol. 7, no. 4, pp. 293–302, 2016, doi: 10.24002/jbi.v7i4.770.
- [6] Rizky Ade Putranto, Triastiti Wuryandari, and Sudarno, "Perbandingan Analisis Klasifikasi Antara Decision Tree Dan Support Vector Machine Multiclass Untuk Penentuan Jurusan Pada Siswa Sma," *J. Gaussian*, vol. 4, no. 4, pp. 1007–1016, 2015.
- [7] Y. Lukito and A. R. Chrismanto, "Perbandingan Metode-Metode Klasifikasi untuk Indoor Positioning System," *J. Tek. Inform. dan Sist. Inf.*, vol. 1, no. 2, pp. 123–131, 2015, doi: 10.28932/jutisi.v1i2.373.
- [8] S. Niu, J. Yang, S. Wang, and G. Chen, "Improvement and parallel implementation of canny edge detection algorithm based on GPU," *Proc. Int. Conf. ASIC*, no. 6, pp. 641–644, 2011, doi: 10.1109/ASICON.2011.6157287.
- [9] W. Ye, Y. Xia, and Q. Wang, "An Improved Canny Algorithm for Edge Detection," *J. Comput. Inf. Syst.*, vol. 75, pp. 1516–1523, 2011, doi: 10.1109/WCSE.2009.718.
- [10] T. F. Wu, C. J. Lin, and R. C. Weng, "Probability estimates for multi-class classification

- by pairwise coupling,” *J. Mach. Learn. Res.*, vol. 5, pp. 975–1005, 2004.
- [11] K. Crammer, “On the algorithmic implementation of multiclass kernel-based vector machines,” *J. Mach. Learn. Res. - JMLR*, vol. 2, no. 2, pp. 265–292, 2002.
- [12] M. J. Hartmann and G. Carleo, “Neural-Network Approach to Dissipative Quantum Many-Body Dynamics,” *Phys. Rev. Lett.*, vol. 122, no. 25, p. 250502, Jun. 2019, doi: 10.1103/PhysRevLett.122.250502.
- [13] B. Gao and L. Pavel, “On the Properties of the Softmax Function with Application in Game Theory and Reinforcement Learning,” 2017.
- [14] H. Zhang, “The optimality of Naive Bayes,” *Proc. Seventeenth Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2004*, vol. 2, pp. 562–567, 2004.
- [15] V. Metsis, I. Androutsopoulos, and G. Paliouras, “Spam filtering with Naive Bayes - Which Naive Bayes?,” *3rd Conf. Email Anti-Spam - Proceedings, CEAS 2006*, 2006.
- [16] M. Christopher, P. Raghavan, and H. Schütze, *An Introduction to Information Retrieval*. Cambridge University Press, 2009.
- [17] Y. L. Pavlov, “Random forests,” *Random For.*, pp. 1–122, 2019, doi: 10.1201/9780367816377-11.
- [18] T. Hastie, S. Rosset, J. Zhu, and H. Zou, “Multi-class AdaBoost,” *Stat. Interface*, vol. 2, no. 3, pp. 349–360, 2009, doi: 10.4310/sii.2009.v2.n3.a8.
- [19] R. Puri and K. Khamrui, “Application of Quantitative Descriptive Analysis (QDA), Principal Component Analysis (PCA) and Response Surface Methodology (RSM) in standardization of cham-cham making,” 2015.
- [20] A. Tharwat, “Linear vs. quadratic discriminant analysis classifier: a tutorial,” *Int. J. Appl. Pattern Recognit.*, vol. 3, no. 2, p. 145, 2016, doi: 10.1504/ijapr.2016.079050.
- [21] A. Tharwat, “Classification assessment methods,” *Appl. Comput. Informatics*, 2018, doi: 10.1016/j.aci.2018.08.003.
- [22] P. A. Flach and M. Kull, “Precision-Recall-Gain curves: PR analysis done right,” *Adv. Neural Inf. Process. Syst.*, vol. 2015-Janua, pp. 838–846, 2015.
- [23] L. Nurhayati and H. Azis, “Perancangan Sistem Pendukung Keputusan Untuk Proses Kenaikan Jabatan Struktural Pada Biro Kepegawaian,” *Semin. Nas. Teknol. Inf. dan Multimed.*, pp. 6–7, 2016.
- [24] J. D. Kelleher, B. Mac Namee, and A. D. Arcy, *Fundamentals of Machine Learning For Predictive Data Analytics Algorithms, Worked Examples, and Case Studies*. London: The MIT Press, 2015.
- [25] K. H. Brodersen, C. S. Ong, K. E. Stephan, and J. M. Buhmann, “The balanced accuracy and its posterior distribution,” *Proc. - Int. Conf. Pattern Recognit.*, pp. 3121–3124, 2010, doi: 10.1109/ICPR.2010.764.
- [26] A. A. Karim, H. Azis, and Y. Salim, “Kinerja Metode C4.5 dalam Penyaluran Bantuan Dana Bencana 1,” *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 84–87, 2018.
- [27] A. Fitria and H. Azis, “Analisis Kinerja Sistem Klasifikasi Skripsi menggunakan Metode Naïve Bayes Classifier,” *Pros. Semin. Nas. Ilmu Komput. dan Teknol. Inf.*, vol. 3, no. 2, pp. 102–106, 2018.
- [28] S. Paembonan *et al.*, “Combination of K-Means and Profile Matching for Drag Substitution,” in *2018 2nd East Indonesia Conference on Computer and Information Technology (EIConCIT)*, Nov. 2018, pp. 180–183, doi: 10.1109/EIConCIT.2018.8878539.
- [29] D. Chicco and G. Jurman, “The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation,” *BMC Genomics*, vol. 21, no. 1, pp. 1–13, 2020, doi: 10.1186/s12864-019-6413-7.