JKT Jurnal Komputer Terapan
ISSN : 2443-4159 (Print) ISSN: 2460-5255 (Online)
Politeknik Caltex Riau
Empower You to Global Competition

OVERSAMPLING MENGGUNAKAN PENDEKATAN LATIN HYPERCUBE SAMPLING DAN K-NEAREST NEIGHBORS UNTUK MENINGKATKAN KINERJA KLASIFIKASI

Sapriadi*[1], Mardiah Nasution[2]
Institut Kesehatan Helvetia, Medan, Indonesia[1]
Program Studi Sistem Informasi, STMIK Logika, Medan, Indonesia[2]
sapriadi@helvetia.ac.id[1], mardiahnst.logika2@gmail.com[2]

**Penulis Koresponden*

ABSTRAK

Ketidakseimbangan kelas pada data (*imbalanced class*) merupakan tantangan signifikan dalam pengembangan model machine learning, yang sering kali menyebabkan penurunan kinerja model. Masalah ini sering ditemui dalam data nyata, di mana proporsi data antara kelas mayoritas dan minoritas sangat tidak seimbang. Salah satu pendekatan yang umum digunakan untuk mengatasi masalah ini adalah oversampling, yang berfungsi untuk menyeimbangkan distribusi kelas dengan menambahkan data sintetis ke kelas minoritas. Teknik oversampling yang paling populer adalah Synthetic Minority Oversampling Technique (SMOTE), meskipun metode ini memiliki kelemahan seperti menghasilkan data yang kurang beragam dan kemungkinan munculnya outlier. Sebagai solusi alternatif, penelitian ini mengusulkan penggunaan metode Latin-Hypercube Sampling (LHS) yang dikombinasikan dengan k-Nearest Neighbor (k-NN) untuk meningkatkan kinerja klasifikasi pada data yang tidak seimbang. Kombinasi LHS dan k-NN diharapkan dapat menghasilkan data sintetis yang lebih berkualitas, sehingga meningkatkan performa model klasifikasi yang diukur menggunakan confusion matrix. Data yang digunakan dalam penelitian ini berasal dari berbagai online repository seperti KEEL, Kaggle, UCI, serta satu dataset penjurusan siswa SMK di Pekanbaru.

Kata kunci: *Imbalanced Class; Kinerja Klasifikasi; k-Nearest Neighbors; Latin Hypercube Sampling; Oversampling*

ABSTRACT

Class imbalance in datasets is a significant challenge in machine learning, often leading to a decline in model performance. This issue is frequently encountered in real-world data, where the proportion between majority and minority classes is highly imbalanced. One common approach to address this problem is oversampling, which aims to balance class distribution by adding synthetic data to the minority class. The most popular oversampling technique is the Synthetic Minority Oversampling Technique (SMOTE), although this method has drawbacks such as producing less diverse data and the potential generation of outliers. As an alternative solution, this study proposes the use of the Latin Hypercube Sampling (LHS) method combined with k-Nearest Neighbor (k-NN) to enhance classification performance on imbalanced datasets. The combination of LHS and k-NN is expected to produce higher quality synthetic data, thereby improving the performance of classification models measured using the confusion matrix. The data used in this study is sourced from various online repositories such as KEEL, Kaggle, UCI, as well as the student specialization of vocational high school (SMK) students in Pekanbaru.

Keywords: *Imbalanced Class; k-Nearest Neighbors; Latin Hypercube Sampling; Oversampling; Performance of Classification*

Histori Artikel
Diserahkan: 21 Agustus 2024 Diterima setelah Revisi: 30 Oktober 2024 Diterbitkan: 28 November 2024

1. PENDAHULUAN

Kebanyakan algoritma *machine learning* mengalami penurunan kinerja pada data yang memiliki jumlah

kelas tidak seimbang atau ketika proporsi data dari satu kelas lebih banyak daripada kelas lainnya yang biasa disebut dengan *Imbalanced Class* [1], [2], [3], [4], [5], [6], [7]. Menurut beberapa penelitian diantaranya [8], [9], [10] mengatakan bahwa *imbalanced class* adalah masalah umum dan sering terjadi pada data yang bersumber dari kasus-kasus nyata di lapangan, dan permasalahan ini merupakan tantangan yang menarik dan harus segera diselesaikan.

Dikutip dari [11], [12], [13] salah satu cara mengatasi masalah *imbalanced class* adalah dengan cara melakukan *oversampling*. *Oversampling* adalah melakukan sintetis data sehingga rasio jumlah data pada setiap kelas menjadi sama atau seimbang [14], [15]. Cara ini dinilai mampu memberikan hasil yang lebih baik pada saat klasifikasi, hal ini telah terbukti pada beberapa penelitian yang dilakukan [1], [6], [16], [17] mengatakan bahwa hasil klasifikasi pada data *imbalanced class* yang telah dilakukan *oversampling* memberikan peningkatan kinerja pada metode klasifikasi yang digunakan, hal ini juga senada dengan hasil penelitian yang dilakukan oleh [8], [10], [18]. Ada banyak metode *oversampling* yang bisa digunakan, yang paling populer adalah *Synthetic Minority Oversampling Technique* (SMOTE) [15], [19], [20], [21]. SMOTE menggunakan prinsip dasar dari metode k-Nearest Neighbor (k-NN). k-Nearest Neighbors (k-NN) merupakan salah satu algoritma paling populer di *machine learning*. Hal tersebut dikarenakan k-NN salah satu algoritma yang *simple*, mudah untuk diimplementasikan dan memiliki hasil yang cukup baik dalam melakukan klasifikasi [22], [23], [24], [25]. Akan tetapi metode SMOTE memiliki beberapa kelemahan diantaranya, masalah keragaman data, konsistensi data yang disintetis, dan sering menghasil data yang berupa *outlier* atau *noise* [14], [19], [20].

Pada penelitian [26] telah menawarkan solusi alternatif untuk permasalahan di atas, yaitu dengan cara melakukan tahapan *oversampling* menggunakan metode *Latin-Hypercube Sampling* (LHS). LHS dinilai mampu memberikan hasil data sampling atau sintetis yang lebih baik, yang mana hal terbukti mampu meningkatkan hasil klasifikasi pada algoritma-algoritma *machine learning* yang digunakan [26], [27].

Berdasarkan pembahasan dari paragraf-paragraf di atas, pada penelitian kali ini, penulis akan mengusulkan metode *oversampling* dengan menggunakan pendekatan *Latin Hypercube Sampling* (LHS) dan *k-Nearest Neighbor* (k-NN), diharapkan kombinasi metode ini mampu meningkatkan kinerja pada model klasifikasi yang digunakan, dimana untuk mengukur kinerja model klasifikasi tersebut penelitian ini menggunakan *confusion matrix*. Sedangkan data yang digunakan pada penelitian ini bersumber dari *online repository* seperti keel, Kaggle, UCI, dan satu dataset penjurusan siswa SMK di Pekanbaru.

2. TINJAUAN PUSTAKA

2.1 OVERSAMPLING

Oversampling adalah teknik yang digunakan dalam pemrosesan data untuk mengatasi ketidakseimbangan kelas atau kelompok dalam dataset yang tidak seimbang [17], [28]. Ketidakseimbangan kelas terjadi ketika jumlah sampel dalam satu kelas jauh lebih sedikit dibandingkan dengan jumlah sampel dalam kelas lainnya [12], [18].

Oversampling bertujuan untuk menyeimbangkan jumlah sampel antara kelas minoritas dan mayoritas dengan meningkatkan jumlah sampel dalam kelas minoritas [13], [19]. Dengan melakukan proses penyeimbangan jumlah data ini diharapkan dapat meningkatkan pengurangan bias yang mungkin timbul pada analisis atau pemodelan [8], [15].

Ada beberapa metode oversampling yang umum digunakan:

i) Duplikasi (repetisi): Metode ini melibatkan duplikasi sampel-sampel dari kelas minoritas sehingga jumlahnya sebanding dengan kelas mayoritas. Pendekatan ini sederhana namun dapat menyebabkan *overfitting* pada data pelatihan [29].
ii) SMOTE (*Synthetic Minority Over-sampling Technique*): Metode ini menciptakan sampel sintetis baru untuk kelas minoritas dengan menggunakan teknik interpolasi. Pada dasarnya, SMOTE menggabungkan fitur-fitur dari sampel minoritas yang berdekatan untuk menciptakan sampel baru di antara mereka. Ini membantu memperluas variasi dan mencegah overfitting [18], [19].
iii) ADASYN (*Adaptive Synthetic Sampling*): Metode ini adalah pengembangan dari SMOTE yang menghasilkan sampel sintetis dengan bobot yang berbeda. ADASYN memberikan penekanan lebih pada sampel yang sulit diklasifikasikan dengan benar oleh model saat ini, sehingga

meningkatkan fokus pada area yang lebih menantang [3], [19].

iv) *Random Oversampling*: Metode ini melibatkan pemilihan sampel acak dari kelas minoritas dan menggandakannya untuk mencapai keseimbangan kelas. Pendekatan ini sederhana dan mudah diimplementasikan, tetapi dapat menyebabkan overfitting pada data pelatihan [30].

Oversampling harus digunakan dengan hati-hati, dan evaluasi yang cermat diperlukan untuk memastikan kualitas dan keakuratan hasilnya.

2.2 LATIN HYPERCUBE SAMPLING (LHS)

Latin Hypercube Sampling (LHS) adalah metode statistik yang digunakan dalam analisis sensitivitas, pemodelan, dan eksperimen yang melibatkan sampel acak. Pendekatan ini menggabungkan keuntungan dari dua teknik sampling yaitu random sampling dan stratified sampling [26], [27], [28].

Pada dasarnya, LHS adalah metode untuk menghasilkan sampel acak yang lebih merata dan efisien. Tujuan utamanya adalah untuk mendistribusikan nilai-nilai parameter secara merata di seluruh rentang yang mungkin, sehingga mencakup variasi yang lebih luas dan meminimalkan bias dalam estimasi [26], [27], [31].

Proses Latin Hypercube Sampling melibatkan langkah-langkah berikut:

Langkah-1 : Tentukan rentang nilai untuk setiap parameter sampling.
Langkah-2 : Tentukan segmen sebanyak jumlah sampel yang diinginkan.
Langkah-3 : Tentukan satu nilai acak untuk setiap segmen. Nilai-nilai ini harus unik sehingga menghasilkan sampel acak yang merata dan efisien.
Langkah-4 : Ulangi Langkah ini sampai menghasilkan satu set sampel yang merata di seluruh rentang nilai parameter.

Manfaat utama dari LHS salahs atunya mampu menghasilkan sampel yang merata secara statistik dan mencakup variasi yang lebih luas, sehingga meningkatkan keakuratan dan keandalan analisis sensitivitas dan pemodelan [26], [27], [28], [31]

2.3 K-NEAREST NEIGHBOR (K-NN)

k-Nearest Neighbor (k-NN) pertama kali diperkenalkan sekitar tahun 1950, k-NN merupakan salah satu metode yang paling banyak digunakan pada permasalahan text categorization, pengenalan pola, pengklasifikasian, dan lain-lain [22], [32], [33], [34], [35], [36], [37], [38]. Hal ini disebabkan k-NN memiliki karakteristik yang cukup atraktif, mudah untuk diterapkan, intuitif, adaktif, serta sederhana [39], [40], [41], [42].

k-NN merupakan algoritma yang termasuk kedalam kategori distance-based algorithms [42]. Distance-Based Algorithms adalah algoritma yang menentukan kemiripan data atau objek berdasarkan pada kedekatan jarak antar data ke suatu kelas atau label atau kelompok data lainnya. Kemiripan antar data pada k-NN ditentukan dengan menggunakan pengukuran model jarak. Adapun beberapa model jarak yang umum digunakan adalah:

$$\mathrm{d}(X, X') = \sqrt{\sum_{i=1}^{y} (x'_i - x_i)^2} \quad (1)$$

k-NN bekerja dengan tujuan menentukan kelas data baru dengan menggunakan data training sebagai acuan. Pada proses training, data yang sudah memiliki label akan diproses dan dipisahkan berdasarkan kemiripannya (gambar 1), lalu data baru akan dihitung jarak terdekat terhadap data training. Setelahnya penentuan kelas data yang didasari pada kelas mayoritas untuk k tetangga terdekat. Alur kerja dari k-NN adalah sebagai berikut[43], [44]:

Langkah-1 : Tentukan nilai k
Langkah-2 : Hitung jarak antar data
Langkah-3 : Pilih k tetangga terdekat berdasarkan urutkan jarak terdekat
Lagnkah-4 : Hitung label mayoritas dan jadikan label mayoritas untuk label data baru

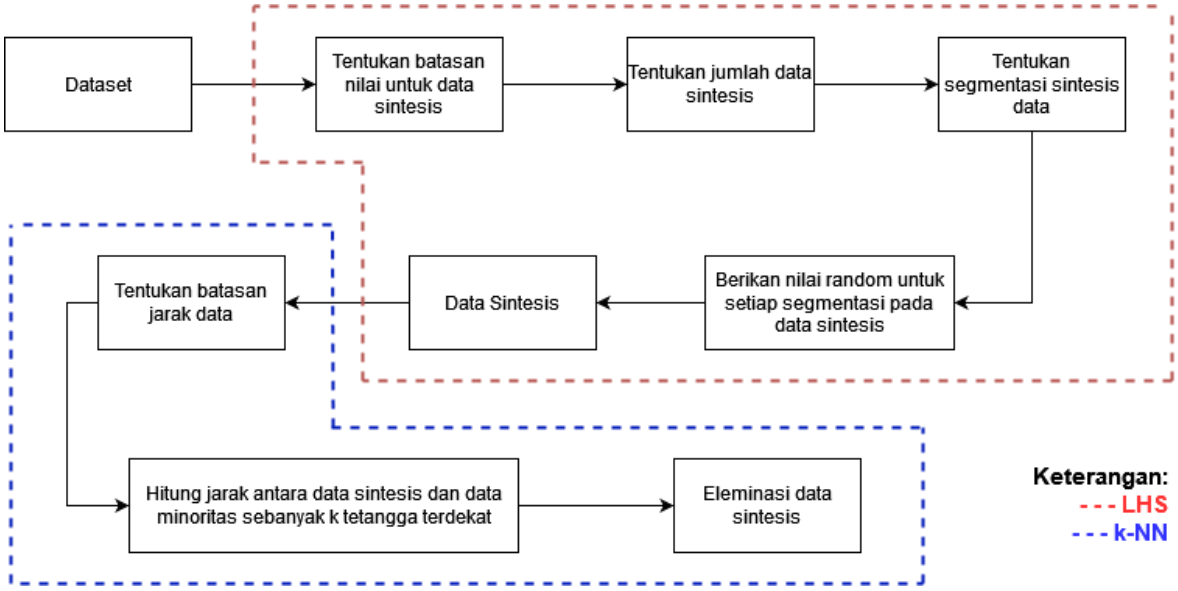
[45] dalam bukunya menyatakan dengan segala kekurangan dan kelebihannya, k-NN merupakan salah

satu dari top ten algoritma in data mining, terutama pada proses klasifikasi.

3. METODE YANG DIUSULKAN

Adapun metode gambaran dari metode yang diusulkan pada penelitian ini secara garis besar dapat dilihat pada gambar 1. Berdasarkan gambar 1, dapat dijelaskan bahwa tahapan pertama dari metode yang diusulkan adalah:

i) Penentuan batasan nilai untuk data yang akan disintetis
   Tahapan ini adalah proses penentuan nilai maksimum dan nilai minimum untuk setiap segmentasi data yang akan disintetis berdasarkan nilai maksimum dan minimum dari data minoritas pada dataset.
ii) Tentukan jumlah data sintesis
   Tentukan jumlah data yang akan disintetis berdasarkan ketimpangan kelas data
iii) Tentukan segmentasi sintetis data
   Tentukan segmentasi sintetis data berdasarkan data minoritas dari dataset
iv) Bangkitkan nilai random untuk setiap segmentasi dari data
   Tentukan nilai random antara nilai minimum dan maksimum untuk setiap segmentasi
v) Data hasil sintetis dengan metode LHS
vi) Tentukan batasan jarak antar data
   Proses ini untuk menentukan sebaran dan besaran varian data yang akan terbentuk.
vii) Hitung Jarak antar data sintetis dan data minoritas
   Proses pada tahapan ini akan melakukan perhitungan jarak antar data yang baru saja disintetis dengan data minoritas dengan model jarak *Euclidean*.
viii) Eliminasi data sintetis
   Eliminasi data sintetis berdasarkan batasan jarak antar data, sehingga mampu menghasilkan data sintetis yang lebih baik atau minim *outlier*



Gambar 1. Kombinasi LHS dan k-NN

Penelitian ini membawa kontribusi baru dalam mengatasi masalah keragaman data, konsistensi data yang disintetis, dan *outlier* dari data yang disintetis dengan mengkombinasikan pendekatan LHS dan k-NN untuk melakukan *oversampling* pada data.

Selain itu, penelitian ini juga melakukan evaluasi kinerja model klasifikasi setelah penerapan metode *oversampling* yang diusulkan. Penggunaan *confusion matrix* diharapkan memberikan pemahaman yang lebih baik tentang kinerja model klasifikasi pada kondisi ketidakseimbangan data.

Keunikan lain dari penelitian ini adalah penggunaan beragam dataset dari online repository, termasuk dataset penjurusan siswa SMK di Pekanbaru. Dengan demikian, penelitian ini tidak hanya memberikan kontribusi dalam pengembangan metode *oversampling* yang baru, akan tetapi juga memberikan

wawasan yang lebih luas tentang efektivitas metode tersebut dalam berbagai konteks dataset yang berbeda.

Sebagai tambahan, penelitian ini menyediakan landasan bagi penelitian lanjutan, baik untuk mengeksplorasi variasi parameter dalam metode *oversampling* yang diusulkan maupun untuk mencoba metode *oversampling* lainnya untuk membandingkan kinerjanya. Hal ini menciptakan peluang untuk terus meningkatkan pemahaman dan solusi dalam mengatasi masalah ketidakseimbangan kelas pada data.

4. HASIL DAN PEMBAHASAN

4.1 PERSIAPAN

Penelitian ini menggunakan data wine quality, glass, palmer penguins, anaemia, food category, date fruit, dry bean yang bersumber dari UCI, keel, dan Kaggle, dan satu dataset lapangan yaitu penjurusan siswa SMK Pekanbaru. Adapun rincian data yang digunakan dapat dilihat pada tabel 1.

Table 1. Dataset yang digunakan

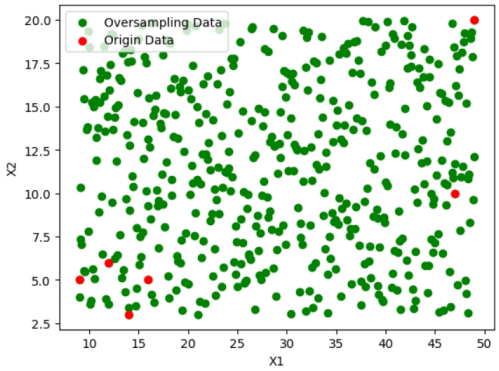
No	Nama Dataset	Jumlah Data per Kelas		No	Nama Dataset	Jumlah Data per Kelas	
1	Wine Quality	class-3	: 6	5	Anaemia	Yes	: 26
		class-4	: 33			No	: 78
		class-5	: 483				
		class-6	: 462				
		class-7	: 143				
		class-8	: 16				
2	Glass	class-1	: 69	6	Food Category	group-1	: 551
		class-2	: 76			group-2	: 319
		class-3	: 17			group-3	: 571
		class-5	: 13			group-4	: 232
		class-6	: 9			group-5	: 722
		class-7	: 29				
3	Palmer Penguins	Adelie	: 146	7	Date Fruit	Berhi	: 65
		Chinstrap	: 68			Deglet	: 98
		Gentoo	: 119			Dokol	: 204
						Iraqi	: 72
						Rotana	: 166
						Safavi	: 199
						Sogay	: 94
4	Penjurusan siswa SMK Pekanbaru	OTKP	: 180	8	Dry Bean	Barbunya	: 1322
		AKT	: 107			Bombay	: 522
		RPL	: 95			Cali	: 1630
		TKJ	: 98			Dermason	: 3546
		BDP	: 53			Horoz	: 1860
						Seker	: 2027
						Sira	: 2636

Dataset yang dipilih pada penelitian ini seluruh fitur atau atributnya bernilai numerik atau angka. Dataset yang digunakan nantinya akan dilakukan *oversampling* dengan metode yang diusulkan, adapun kelas data yang akan dilakukan *oversampling* adalah jumlah kelas data dengan *imbalanced ratio* (ir) besar sama dengan 1.5 dari kelas data mayoritas.

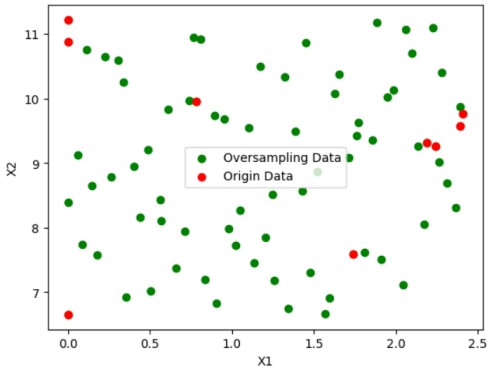
Pada penelitian ini, untuk melihat lebih sejauh mana mentode yang diusulkan ini mampu melakukan *oversampling* data dengan baik, maka dilakukan proses klasifikasi dengan menggunakan metode klasifikasi k-NN dengan nilai k = 3, *Support Vector Machine* (SVM), *Decision Tree* (DT), dan *Naïve Bayes* (NB). Agar kinerja dari metode klasifikasi dapat memberikan gambaran yang lebih baik maka dataset yang diguakan akan dibagi menjadi 2 bagian, yaitu sebanya 75% dari data digunakan sebagai data latih dan 15% digunakan sebagai data uji. Proses implementasi dari metode yang diusulkan dilakukan dengan menggunakan bahasa pemrograman phyton dengan bantuan platform Kaggle.

4.2 OVERSAMPLING DENGAN METODE YANG DIUSULKAN

Berikut adalah beberapa plot persebaran data hasil *oversampling* dari setiap kelas minoritas untuk setiap dataset wine kelas-3 dapat dilihat pada gambar 2, dan dataset glass untuk kelas data 6 dapat dilihat pada gambar 3.

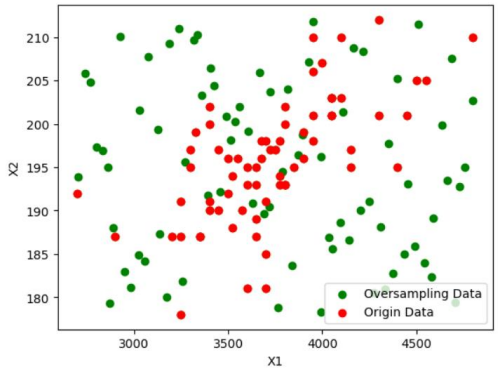


Gambar 2. Oversampling Kelas-3 Dataset Wine

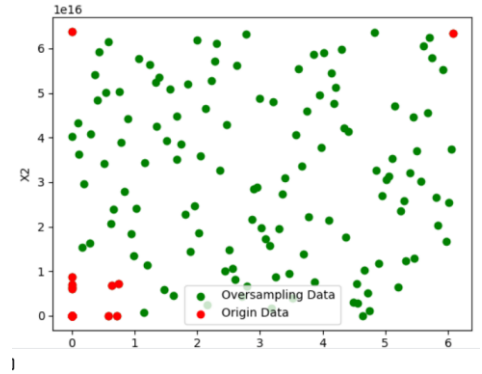


Gambar 3. Oversampling Kelas-6 Dataset Glass

Gambar 4 merupakan plot persebaran data hasil *oversampling* untuk kelas Chinstrap pada dataset palmer penguins, dan gambar 5 merupakan plot hasil *oversampling* untuk kelas BDP pada data penjurusan siswa.

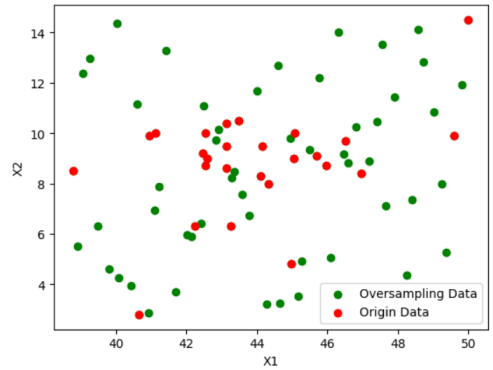


Gambar 4. Oversampling Kelas Chinstrap Dataset Palmer Penguins

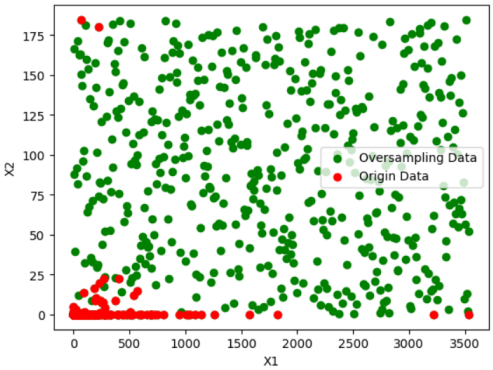


Gambar 5. Oversampling Kelas BDP Dataset Penjurusan Siswa SMK

Gambar 6 merupakan plot untuk kelas YES pada dataset anaemia, dan gambar 7 merupakan plot hasil *oversampling* group-4 untuk dataset *food category*.

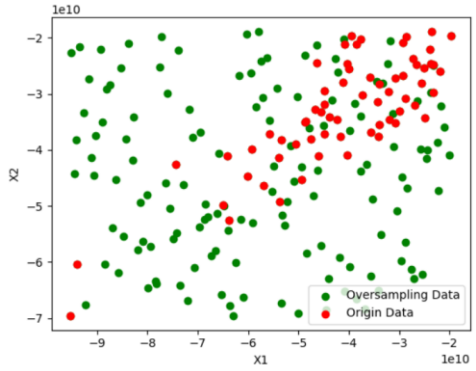


Gambar 6. Oversampling Kelas YES Dataset Anaemia

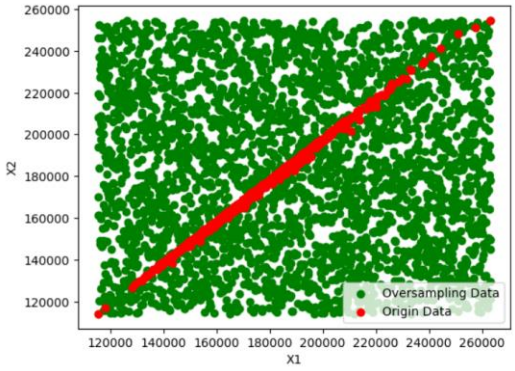


Gambar 7. Oversampling Kelas Group-4 Dataset Food Category

Plot data hasil oversampling pada dataset date fruit untuk Kelas berhi dapat dilihat pada gambar 8, dan kelas data bombay dari dataset dry bean dapat dilihat pada gambar 9.



Gambar 8. Oversampling Kelas Berhi Dataset Date Fruit



Gambar 9. Oversampling Kelas Bombay Dataset Dry Bean

Pada tahapan ini setiap kelas dari dataset akan terlebih dahulu dicek apakah memenuhi batasan *imbalanced ratio* yang besar sama dengan 1.5. Artinya apabila terdapat 10 buah data pada kelas-1 dan selanjutnya terdapat 20 buah data pada kelas lainnya, maka data pada kelas-1 termasuk kedalam kelas minoritas dan akan dilakukan *oversampling* dengan menggunakan metode yang diusulkan.

Misalkan pada data wine ini terdapat 6 kelas data, dimana kelas minoritas pada data ini berjumlah sebanyak 4 kelas data, diantaranya kelas data-3, 4, 7, dan kelas data-8. Kelas data minoritas ini akan dilakukan proses *oversampling* dengan metode yang diusulkan. *Oversampling* dilakukan pada kelas minoritas dengan mengikut jumlah maksimum dari salah satu kelas data. Pada dataset ini kelas data dengan jumlah terbanyak dimiliki oleh kelas data-5 yaitu berjumlah 483 data, sehingga data pada kelas minoritas akan dilakukan *oversampling* sampai berjumlah 483 data. Adapun hasil dari *oversampling* untuk setiap dataset dapat dilihat pada tabel 2.

Table 2. Jumlah Data Sebelum dan Sesudah *Oversampling*

No	Nama Dataset	Nama Kelas	Jumlah	Imbalanced Ratio	Setelah Oversampling
1	Wine Quality	class-3	6	80.50	483
		class-4	33	14.64	483
		class-5	483	1.00	483
		class-6	462	1.05	462
		class-7	143	3.38	483
		class-8	16	30.19	483
2	Glass	class-1	69	1.10	69
		class-2	76	1.00	76
		class-3	17	4.47	73
		class-5	13	5.85	76
		class-6	9	8.44	76
		class-7	29	2.62	76
3	Palmer Penguins	Adelie	146	1.00	146
		Chinstrap	68	2.15	68
		Gentoo	119	1.23	119
4	Penjurusan siswa SMK Pekanbaru	OTKP	180	1.00	180
		AKT	107	1.68	107
		RPL	95	1.89	95
		TKJ	98	1.84	98
		BDP	53	3.40	53
5	Anaemia	Yes	26	3.00	26
		No	78	1.00	78
6	Food Category	group-1	551	1.31	551
		group-2	319	2.26	319

No	Nama Dataset	Nama Kelas	Jumlah	Imbalanced Ratio	Setelah Oversampling
		group-3	571	1.26	571
		group-4	232	3.11	232
		group-5	722	1.00	722
7	Date Fruit	Berhi	65	3.14	65
		Deglet	98	2.08	98
		Dokol	204	1.00	204
		Iraqi	72	2.83	72
		Rotana	166	1.23	166
		Safavi	199	1.03	199
		Sogay	94	2.17	94
8	Dry Bean	Barbunya	1322	2.68	1322
		Bombay	522	6.79	522
		Cali	1630	2.18	1630
		Dermason	3546	1.00	3546
		Horoz	1860	1.91	1860
		Seker	2027	1.75	2027
		Sira	2636	1.35	2636

4.3 EVALUASI KINERJA KLASIFIKASI

Setelah data dilakukan proses *oversampling* selanjunya data tersebut akan digunakan untuk proses klasifikasi. Adapun kinerja metode klasifikasi pada dataset wine dapat dilihat pada tabel 3.

Table 3. Kinerja Metode Klasifikasi Pada Dataset Wine

No	Metode Klasifikasi	Kinerja Sebelum Oversampling				Kinerja Setelah Oversampling			
		Akurasi	Precision	Recall	F1	Akurasi	Precision	Recall	F1
1	k-NN	0.5523	0.2632	0.2655	0.2643	0.6967	0.6976	0.6884	0.6918
2	SVM	0.5988	0.2398	0.2777	0.2573	0.7407	0.7605	0.7298	0.7391
3	DT	0.5813	0.3425	0.3555	0.3483	0.7662	0.7592	0.7657	0.7615
4	Naïve Bayes	0.5697	0.2927	0.2931	0.2863	0.7569	0.7853	0.7571	0.7660

Berdasarkan tabel 3, dapat dilihat bahwa data hasil *oversampling* memberikan peningkatan kinerja terhadap proses klasifikasi pada hampir seluruh metode klasifikasi yang digunakan. Peningkatan akurasi dan recall tertingg terjadi pada metode naïve bayes dengan peningkatan sebesar 0.1872 dan 0.464, sedangkan untuk peningkatan precision dan F1 terjadi pada metode SVM yaitu sebesar 0.5207 dan 0.4818.

Adapun kinerja metode klasifikasi pada dataset glass dapat dilihat pada tabel 4.

Table 4. Kinerja Metode Klasifikasi Pada Dataset Glass

No	Metode Klasifikasi	Kinerja Sebelum Oversampling				Kinerja Setelah Oversampling			
		Akurasi	Precision	Recall	F1	Akurasi	Precision	Recall	F1
1	k-NN	0.7187	0.6278	0.5854	0.5974	0.8676	0.8657	0.8629	0.8590
2	SVM	0.6250	0.5377	0.4484	0.4494	0.6470	0.7056	0.7174	0.6568
3	DT	0.6875	0.5444	0.6000	0.5661	0.8971	0.9013	0.9058	0.9032
4	Naïve Bayes	0.5625	0.4514	0.5463	0.4903	0.6912	0.7431	0.7524	0.7100

Berdasarkan tabel 4, dapat dilihat bahwa hasil *oversampling* memberikan peningkatan kinerja terhadap proses klasifikasi, dimana peningkatan akurasi, precision, recall, dan F1 score yang tertinggi terjadi pada metode *decision tree* dengan peningkatan sebesar 0.2096 untuk akurasi, 0.3569 untuk precision, 0.3058 untuk recall, dan 0.3371 F1 score.

Adapun kinerja metode klasifikasi pada dataset palmer penguins dapat dilihat pada tabel 5.

Table 5. Kinerja Metode Klasifikasi Pada Dataset Palmer Penguins

No	Metode Klasifikasi	Kinerja Sebelum Oversampling				Kinerja Setelah Oversampling			
		Akurasi	Precision	Recall	F1	Akurasi	Precision	Recall	F1
1	k-NN	1.0000	1.0000	1.0000	1.0000	0.9516	0.9667	0.9500	0.9554
2	SVM	0.9800	0.9861	0.9762	0.9806	0.9516	0.9667	0.9500	0.9554
3	DT	0.9800	0.9778	0.9855	0.9811	0.9516	0.9554	0.9586	0.9567
4	Naïve Bayes	0.9800	0.9861	0.9762	0.9806	0.9839	0.9881	0.9833	0.9854

Berdasarkan pada tabel 5, data hasil oversampling hanya mengalami peningkatan pada metode klasifikasi naïve bayes, yaitu 0.0039 untuk akurasi, 0.0020 untuk precision, 0.0071 untuk recall, dan 0.0048 untuk F1-score.

Adapun kinerja metode klasifikasi pada dataset penjurusan siswa SMK Pekanbaru dapat dilihat pada tabel 6.

Table 6. Kinerja Metode Klasifikasi Pada Dataset Penjurusan Siswa SMK Pekanbaru

No	Metode Klasifikasi	Kinerja Sebelum Oversampling				Kinerja Setelah Oversampling			
		Akurasi	Precision	Recall	F1	Akurasi	Precision	Recall	F1
1	k-NN	0.5750	0.5252	0.5285	0.5144	0.7481	0.7601	0.7618	0.7523
2	SVM	0.6125	0.5800	0.5631	0.5688	0.7185	0.7417	0.7372	0.7235
3	DT	0.6250	0.5870	0.5588	0.5678	0.7556	0.7738	0.7684	0.7638
4	Naïve Bayes	0.3250	0.2869	0.3509	0.2457	0.5556	0.7615	0.5880	0.5717

Melalui tabel 6 dapat dilihat bahwa seluruh metode klasifikasi mengalami peningkatan kinerja, terutama pada metode klasifikasi naïve bayes yang mengalami peningkatan tertinggi pada seluruh indikator kinerja, diantaranya 0.2306 untuk nilai akurasi, 0.4746 untuk precision, 0.2370 untuk nilai recall, dan 0.3260 untuk F1-score.

Adapun kinerja metode klasifikasi pada dataset Anaemia dapat dilihat pada tabel 7.

Table 7. Kinerja Metode Klasifikasi Pada Dataset Anaemia

No	Metode Klasifikasi	Kinerja Sebelum Oversampling				Kinerja Setelah Oversampling			
		Akurasi	Precision	Recall	F1	Akurasi	Precision	Recall	F1
1	k-NN	1.0000	1.0000	1.0000	1.0000	0.8333	0.8252	0.8444	0.8286
2	SVM	1.0000	1.0000	1.0000	1.0000	0.8750	0.8643	0.8778	0.8693
3	DT	1.0000	1.0000	1.0000	1.0000	0.9167	0.9111	0.9111	0.9111
4	Naïve Bayes	1.0000	1.0000	1.0000	1.0000	0.8750	0.8643	0.8778	0.8693

Pada dataset anaemia, seluruh metode klasifikasi mengalami penurunan kinerja, yang mana penurunan kinerja tertinggi terjadi pada metode k-NN dengan nilai penurunan akurasi sebesar 0.1667, precision sebesar 0.1748, recall sebesar 0.1556, dan F1-score sebesar 0.1714.

Adapun kinerja metode klasifikasi pada dataset Food Category dapat dilihat pada tabel 8.

Berdasarkan tabel 8 terlihat bahwa seluruh metode klasifikasi yang digunakan mengalami peningkatan kinerja, dengan metode klasifikasi SVM mendapatkan peningkatan kinerja tertinggi untuk seluruh indikator kinerja. Peningkatan akurasi pada SVM terjadi sebesar 0.1168, precision 0.2809, recall 0.2053, dan F1-score sebesar 0.2309.

Table 8. Kinerja Metode Klasifikasi Pada Dataset Food Category

No	Metode Klasifikasi	Kinerja Sebelum Oversampling				Kinerja Setelah Oversampling			
		Akurasi	Precision	Recall	F1	Akurasi	Precision	Recall	F1
1	k-NN	0.6861	0.6952	0.6607	0.6691	0.7368	0.7382	0.7345	0.7357
2	SVM	0.4278	0.4669	0.3303	0.3132	0.5445	0.7478	0.5356	0.5442
3	DT	0.6611	0.6532	0.6463	0.6467	0.7085	0.7068	0.7064	0.7050
4	Naïve Bayes	0.4667	0.4440	0.3720	0.3559	0.5769	0.6964	0.5594	0.5636

Adapun kinerja metode klasifikasi pada dataset Date Fruit dapat dilihat pada tabel 9.

Table 9. Kinerja Metode Klasifikasi Pada Dataset Date Fruit

No	Metode Klasifikasi	Kinerja Sebelum Oversampling				Kinerja Setelah Oversampling			
		Akurasi	Precision	Recall	F1	Akurasi	Precision	Recall	F1
1	k-NN	0.9111	0.8542	0.8573	0.8506	0.8413	0.8509	0.8491	0.8470
2	SVM	0.9481	0.9343	0.9196	0.9243	0.9038	0.9145	0.9035	0.9073
3	DT	0.8815	0.7976	0.8040	0.7991	0.8702	0.8758	0.8659	0.8693
4	Naïve Bayes	0.8741	0.8063	0.8125	0.7974	0.9135	0.9180	0.9112	0.9133

Berdasarkan tabel 9 hanya metode naïve bayes yang mengalami peningkatan kinerja, yaitu sebesar 0.0394 untuk akurasi, 0.1117 untuk precision, 0.0987 untuk recall, dan 0.1159 untuk F1-score.

Adapun kinerja metode klasifikasi pada dataset Dry Bean dapat dilihat pada tabel 10.

Table 10. Kinerja Metode Klasifikasi Pada Dataset Dry Bean

No	Metode Klasifikasi	Kinerja Sebelum Oversampling				Kinerja Setelah Oversampling			
		Akurasi	Precision	Recall	F1	Akurasi	Precision	Recall	F1
1	k-NN	0.9158	0.9299	0.9255	0.9270	0.9035	0.9040	0.9006	0.9017
2	SVM	0.9218	0.9363	0.9320	0.9335	0.9089	0.9068	0.9065	0.9066
3	DT	0.8976	0.9108	0.9109	0.9107	0.8939	0.8910	0.8912	0.8911
4	Naïve Bayes	0.8967	0.9066	0.9083	0.9065	0.8798	0.8791	0.8786	0.8777

Berdasakan tabel 10, seluruh metode klasifikasi mengalami penurunan kinerja. Metode yang paling besar mengalami penurunan pada indikator akurasi, recall, dan F1-score adalah SVM yaitu sebesar 0.0168, 0.0297, dan 0.0288, dan untuk precision yaitu sebesar 0.0295 terjadi pada metode klasifikasi SVM.

Berdasarkan data pada tabel-tabel sebelumnya terlihat bahwa metode yang diusulkan mampu melakukan *oversampling* dan dapat memberikan kinerja yang lebih baik pada seluruh metode klasifikasi pada dataset *wine*, *glass*, penjurusan siswa SMK, dan *Food Category*. Pada dataset palmer penguins, dan date fruit, metode klasifikasi k-NN, SVM, dan Decision Tree mengalami penurunan kinerja, hanya naïve bayes yang tetap mengalami peningkatan kinerja. Sedangkan pada dataset *dry bean* seluruh metode klasifikasi mengalami penurunan.

Adapun rangkuman peningkatan rata-rata kinerja dari metode klasifikasi yang digunakan dapat dilihat pada tabel 11.

Table 11. Rata-rata Peningkatan Kinerja Metode Klasifikasi

Metode	Akurasi	Precision	Recall	F1-Score
k-NN	0.0275	0.0891	0.0961	0.0936
SVM	0.0220	0.1159	0.1138	0.1094
Decisiont Tree	0.0557	0.1201	0.1140	0.1177
Naïve Bayes	0.0698	0.1827	0.1311	0.1493

Berdasarkan data-data yang telah disajikan dapat disimpulkan bahwa metode yang diusulkan mampu melakukan *oversampling* dengan baik dan memberikan peningkatan kinerja pada metode klasifikasi dengan peningkatan rata-rata kinerja tertinggi terjadi pada metode naïve bayes.

5. KESIMPULAN

Berdasarkan pembahasan pada sub chapter sebelumnya dapat disimpulkan bahwa metode yang diusulkan dapat melakukan *oversampling* dengan baik dan mampu meningkatkan kinerja metode klasifikasi terutama pada dataset yang sebelumnya memiliki hasil kinerja klasifikasi yang rendah. Peningkatan kinerja dari metode klasifikasi tertinggi terjadi pada model klasifikasi Naïve Bayes dengan peningkatan akurasi sebesar 0.0698, precision sebesar 0.1827, recall sebesar 0.1311, dan F1-score sebesar 0.1493.

DAFTAR PUSTAKA

[1] H. Wang and H. Huang, "Feature Space Oversampling Technique for Imbalanced Classification," *2019 6th Int. Conf. Information, Cybern. Comput. Soc. Syst. ICCSS 2019*, pp. 93–99, 2019, doi: 10.1109/ICCSS48103.2019.9115430.

[2] R. Sauber-Cole and T. M. Khoshgoftaar, "The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey," *J. Big Data*, vol. 9, no. 1, 2022, doi: 10.1186/s40537-022-00648-6.

[3] A. Puri and M. K. Gupta, "Improved Hybrid Bag-Boost Ensemble with K-Means-SMOTE-ENN Technique for Handling Noisy Class Imbalanced Data," *Comput. J.*, vol. 65, no. 1, pp. 124–138, 2022, doi: 10.1093/comjnl/bxab039.

[4] H. Mardiansyah, R. Widia Sembiring, and S. Efendi, "Handling Problems of Credit Data for Imbalanced Classes using SMOTEXGBoost," *J. Phys. Conf. Ser.*, vol. 1830, no. 1, 2021, doi: 10.1088/1742-6596/1830/1/012011.

[5] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, 2019, doi: 10.1186/s40537-019-0192-5.

[6] W. Ustyannie and S. Suprapto, "Oversampling Method To Handling Imbalanced Datasets Problem in Binary Logistic Regression Algorithm," *IJCCS (Indonesian J. Comput. Cybern. Syst.*, vol. 14, no. 1, p. 1, 2020, doi: 10.22146/ijccs.37415.

[7] S. Mutmainah, "Penanganan Imbalance Data Pada Klasifikasi," *SNATi*, vol. 1, pp. 10–16, 2021.

[8] I. Kunakorntum, W. Hinthong, and P. Phunchongharn, "A Synthetic Minority Based on Probabilistic Distribution (SyMProD) Oversampling for Imbalanced Datasets," *IEEE Access*, vol. 8, pp. 114692–114704, 2020, doi: 10.1109/ACCESS.2020.3003346.

[9] L. Zhang *et al.*, "A class imbalance loss for imbalanced object recognition," *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 13, pp. 2778–2792, 2020, doi: 10.1109/JSTARS.2020.2995703.

[10] Z. Wang and H. Wang, "Global Data Distribution Weighted Synthetic Oversampling Technique for Imbalanced Learning," *IEEE Access*, vol. 9, pp. 44770–44783, 2021, doi: 10.1109/ACCESS.2021.3067060.

[11] C. Liu *et al.*, "Constrained Oversampling: An Oversampling Approach to Reduce Noise Generation in Imbalanced Datasets With Class Overlapping," *IEEE Access*, vol. 10, no. July 2020, pp. 91452–91465, 2022, doi: 10.1109/ACCESS.2020.3018911.

[12] J. Engelmann and S. Lessmann, "Conditional Wasserstein GAN-based oversampling of tabular data for imbalanced learning," *Expert Syst. Appl.*, vol. 174, no. December 2020, p. 114582, 2021, doi: 10.1016/j.eswa.2021.114582.

[13] Y. Il Kang and S. Won, "Weight decision algorithm for oversampling technique on class-imbalanced learning," *ICCAS 2010 - Int. Conf. Control. Autom. Syst.*, pp. 182–186, 2010, doi: 10.1109/iccas.2010.5669889.

[14] C. Liu, X. Wang, K. Wu, J. Tan, F. Li, and W. Liu, "Oversampling for imbalanced time series classification based on generative adversarial networks," *2018 IEEE 4th Int. Conf. Comput. Commun. ICCC 2018*, pp. 1104–1108, 2018, doi: 10.1109/CompComm.2018.8780808.

[15] S. Korkmaz, M. A. Şahman, A. C. Cinar, and E. Kaya, "Boosting the oversampling methods based on differential evolution strategies for imbalanced learning," *Appl. Soft Comput.*, vol. 112, p. 107787, 2021, doi: 10.1016/j.asoc.2021.107787.

[16] S. K. Lee, S. J. Hong, and S. Il Yang, "Oversampling for Imbalanced Data Classification Using Adversarial Network," *9th Int. Conf. Inf. Commun. Technol. Converg. ICT Converg. Powered by Smart Intell. ICTC 2018*, pp. 1255–1257, 2018, doi: 10.1109/ICTC.2018.8539543.
[17] V. A. Briones-Segovia, V. Jiménez-Villar, J. A. Carrasco-Ochoa, and J. F. Martínez-Trinidad, "A new oversampling method in the string space," *Expert Syst. Appl.*, vol. 183, no. November 2020, 2021, doi: 10.1016/j.eswa.2021.115428.
[18] G. Douzas, R. Rauch, and F. Bacao, "G-SOMO: An oversampling approach based on self-organized maps and geometric SMOTE," *Expert Syst. Appl.*, vol. 183, no. May, p. 115230, 2021, doi: 10.1016/j.eswa.2021.115230.
[19] S. Feng, J. Keung, X. Yu, Y. Xiao, and M. Zhang, "Investigation on the stability of SMOTE-based oversampling techniques in software defect prediction," *Inf. Softw. Technol.*, vol. 139, no. August 2020, p. 106662, 2021, doi: 10.1016/j.infsof.2021.106662.
[20] H. Zhou, X. Dong, S. Xia, and G. Wang, "Weighted oversampling algorithms for imbalanced problems and application in prediction of streamflow[Formula presented]," *Knowledge-Based Syst.*, vol. 229, p. 107306, 2021, doi: 10.1016/j.knosys.2021.107306.
[21] J. Liu, "A minority oversampling approach for fault detection with heterogeneous imbalanced data," *Expert Syst. Appl.*, vol. 184, no. July, p. 115492, 2021, doi: 10.1016/j.eswa.2021.115492.
[22] K. U. Syaliman, A. Labellapansa, and A. Yulianti, "Improving the Accuracy of Features Weighted k-Nearest Neighbor using Distance Weight," no. ICoSET 2019, pp. 326–330, 2020, doi: 10.5220/0009390903260330.
[23] Z. Pan, Y. Wang, and W. Ku, "A new general nearest neighbor classification based on the mutual neighborhood information," *Knowledge-Based Syst.*, vol. 121, pp. 142–152, 2017, doi: 10.1016/j.knosys.2017.01.021.
[24] Ö. F. Ertuğrul and M. E. Tağluk, "A novel version of k nearest neighbor: Dependent nearest neighbor," *Appl. Soft Comput. J.*, vol. 55, pp. 480–490, 2017, doi: 10.1016/j.asoc.2017.02.020.
[25] A. A. Nababan, O. S. Sitompul, and Tulus, "Attribute Weighting Based K-Nearest Neighbor Using Gain Ratio Attribute Weighting Based K-Nearest Neighbor Using Gain Ratio," 2018.
[26] X. Sun *et al.*, "Smart Sampling for Reduced and Representative Power System Scenario Selection," *IEEE Open Access J. Power Energy*, vol. 8, no. May, pp. 293–302, 2021, doi: 10.1109/OAJPE.2021.3093278.
[27] Q. Wang *et al.*, "Modified Algorithms for Fast Construction of Optimal Latin-Hypercube Design," *IEEE Access*, vol. 8, pp. 191644–191658, 2020, doi: 10.1109/ACCESS.2020.3032122.
[28] L. Zhu *et al.*, "Pendekatan Resampling Data Untuk Menangani Masalah Ketidakseimbangan Kelas," *IEEE Access*, vol. 8, no. 1, pp. 31–38, 2021, doi: 10.35508/jicon.v10i1.6554.
[29] X. Wang, J. Xu, T. Zeng, and L. Jing, "Local distribution-based adaptive minority oversampling for imbalanced data classification," *Neurocomputing*, vol. 422, pp. 200–213, 2021, doi: 10.1016/j.neucom.2020.05.030.
[30] T. Kurbiel, H. G. Gckler, and D. Alfsmann, "A novel approach to the design of oversampling low-delay complex-modulated filter bank Pairs," *EURASIP J. Adv. Signal Process.*, vol. 2009, 2009, doi: 10.1155/2009/692861.
[31] J. Mendes, M. Freitas, H. Siqueira, A. Lazzaretti, S. Stevan, and S. Pichorim, "Comparative Analysis Among Feature Selection of sEMG Signal for Hand Gesture Classification by Armband," *IEEE Lat. Am. Trans.*, vol. 18, no. 6, pp. 1135–1143, 2020.
[32] A. Islam, S. B. Belhaouari, A. U. Rehman, and H. Bensmail, "K Nearest Neighbor OveRsampling approach: An open source python package for data augmentation," *Softw. Impacts*, vol. 12, no. February, p. 100272, 2022, doi: 10.1016/j.simpa.2022.100272.
[33] M. Kumar, N. K. Rath, A. Swain, and S. K. Rath, "Feature Selection and Classification of Microarray Data using MapReduce based ANOVA and K-Nearest Neighbor," *Procedia Comput. Sci.*, vol. 54, pp. 301–310, 2015, doi: 10.1016/j.procs.2015.06.035.
[34] P. Nair and I. Kashyap, "Classification of medical image data using k nearest neighbor and finding the optimal k value," *Int. J. Sci. Technol. Res.*, vol. 9, no. 4, pp. 221–226, 2020.
[35] G. I. Okolo, S. Katsigiannis, and N. Ramzan, "IEViT: An enhanced vision transformer architecture for chest X-ray image classification," *Comput. Methods Programs Biomed.*, vol. 226, p. 107141, 2022, doi: 10.1016/j.cmpb.2022.107141.
[36] J. A. Romero-del-Castillo, M. Mendoza-Hurtado, D. Ortiz-Boyer, and N. García-Pedrajas,

"Local-based k values for multi-label k-nearest neighbors rule," *Eng. Appl. Artif. Intell.*, vol. 116, no. June, p. 105487, 2022, doi: 10.1016/j.engappai.2022.105487.

[37] S. Suyanto, P. E. Yunanto, T. Wahyuningrum, and S. Khomsah, "A multi-voter multi-commission nearest neighbor classifier," *J. King Saud Univ. - Comput. Inf. Sci.*, vol. 34, no. 8, pp. 6292–6302, 2022, doi: 10.1016/j.jksuci.2022.01.018.

[38] X. Zhang, H. Xiao, R. Gao, H. Zhang, and Y. Wang, "K-nearest neighbors rule combining prototype selection and local feature weighting for classification," *Knowledge-Based Syst.*, vol. 243, 2022, doi: 10.1016/j.knosys.2022.108451.

[39] N. García-Pedrajas and D. Ortiz-Boyer, "Boosting k-nearest neighbor classifier by means of input space projection," *Expert Syst. Appl.*, vol. 36, no. 7, pp. 10570–10582, 2009, doi: 10.1016/j.eswa.2009.02.065.

[40] S. Ougiaroglou and G. Evangelidis, "Fast and accurate k-nearest neighbor classification using prototype selection by clustering," *Proc. 2012 16th Panhellenic Conf. Informatics, PCI 2012*, no. i, pp. 168–173, 2012, doi: 10.1109/PCi.2012.69.

[41] Z. Pan, Y. Wang, and W. Ku, "A new k-harmonic nearest neighbor classifier based on the multi-local means," *Expert Syst. Appl.*, vol. 67, pp. 115–125, 2017, doi: 10.1016/j.eswa.2016.09.031.

[42] J. Wang, P. Neskovic, and L. N. Cooper, "Improving nearest neighbor rule with a simple adaptive distance measure," *Pattern Recognit. Lett.*, vol. 28, no. 2, pp. 207–213, 2007, doi: 10.1016/j.patrec.2006.07.002.

[43] K. U. Syaliman, E. B. Nababan, and O. S. Sitompul, "Improving the accuracy of k-nearest neighbor using local mean based and distance weight," *J. Phys. Conf. Ser.*, vol. 978, no. 1, pp. 1–6, 2018, doi: 10.1088/1742-6596/978/1/012047.

[44] Y. Yuliska and K. U. Syaliman, "Peningkatan Akurasi K-Nearest Neighbor Pada Data Index Standar Pencemaran Udara Kota Pekanbaru," *IT J. Res. Dev.*, vol. 5, no. 1, pp. 11–18, 2020, doi: 10.25299/itjrd.2020.vol5(1).4680.

[45] X. Wu *et al.*, *Top 10 algorithms in data mining*, vol. 14, no. 1. 2008. doi: 10.1007/s10115-007-0114-2.