PROSIDING Seminar Nasional SISFOTEK ISSN 2597-3584 (Media Online)
Sistem Informasi dan Teknologi
website : www.seminar.iaii.or.id | email : seminar@iaii.or.id

Kombinasi Algoritma Sampling dengan Algoritma Klasifikasi untuk Meningkatkan Performa Klasifikasi Dataset Imbalance

Gagah Gumelar[1], Norlaila[2] , Quratul Ain[3], Riza Marsuciati[4], Silvi Agustanti Bambang[5], Andi Sunyoto[6], M. Syukri Mustafa[7]
[1,2,3,4,5,6,7]Magister Teknik Informatika, Fakultas Ilmu Komputer, Universitas AMIKOM Yogyakarta
[1]gagah.gumelar@students.amikom.ac.id, [2]norlaila@students.amikom.ac.id, [3]ainquratul@students.amikom.ac.id, [4]riza.marsuciati@students.amikom.ac.id, [5]silvibambang97@students.amikom.ac.id, [6]andi@amikom.ac.id, [7]syukri@dipanegara.ac.id

Abstract

A class to be imbalanced when there is a class that has more data than other classes. A comparison between minority classes and the majority class is called Imbalance Ratio (IR). The greater the difference between the minority class and the majority class the value of the Imbalance Ratio (IR) is getting larger. Dataset imbalance in data mining is a serious problem. The application of the classification algorithm regardless of class balance resulted in a good prediction for the majority class and a neglected minority class. Therefore, in this research, the SMOTE algorithm was applied to balance the dataset. The study used 4 datasets with different Imbalance Ratio and used classification algorithms, C45, Naïve Bayes, K-NN, and SVM. Then compared before and after using SMOTE. The research results that have been done accuracy value and value G-mean Naïve Bayes algorithm is consistent with its performance at each level of imbalance ratio, before the implementation has no good performance, whereas after the implemented SMOTE algorithm Naïve Bayes has a consistent increase in accuracy. So it can be concluded that the combination SMOTE + Naïve Bayes most effectively used in the imbalance dataset with different levels in the scheme of 10 fold cross validation and 80% data testing tested as much as 50 times.

Keywords: *classification, imbalance datasets, SMOTE*

Abstrak

Kelas akan seimbang ketika ada kelas yang memiliki lebih banyak data daripada kelas lain. Kelompok kelas dengan banyak data disebut kelas mayoritas, sebaliknya disebut kelas minoritas. Perbandingan antara kelas minoritas dan kelas mayoritas disebut rasio ketidakseimbangan (IR). Semakin besar perbedaan antara kelas minoritas dan kelas mayoritas nilai rasio ketidakseimbangan (IR) semakin besar. Dataset tak seimbang menyebabkan kebingungan atau kesalahan hasil klasifikasi dimana data kelas minoritas sering diklasifikasikan sebagai kelas mayoritas. Aplikasi algoritma klasifikasi terlepas dari keseimbangan kelas menghasilkan prediksi yang baik untuk kelas mayoritas dan kelas minoritas diabaikan. Oleh karena itu dalam penelitian ini diterapkan algoritma SMOTE untuk menyeimbangkan dataset. Penelitian menggunakan 4 datasets dengan rasio ketidakseimbangan yang berbeda dan menggunakan algoritma klasifikasi, C45, Nïve Bayes, KNN, dan SVM. Kemudian dibandingkan sebelum dan sesudah menggunakan SMOTE. Hasil penelitian yang telah dilakukan nilai akurasi dan nilai G Naive Bayes algorithm yang konsisten dengan kinerja pada setiap tingkat dari rasio ketidakseimbangan, sebelum diimplementasi tidak memiliki kinerja yang baik, sedangkan setelah yang diimplementasikan Naive Bayes SMOTE memiliki peningkatan akurasi yang konsisten. Jadi dapat disimpulkan bahwa kombinasi SMOTE + Naïve Bayes paling efektif digunakan dalam dataset ketidakseimbangan dengan tingkat yang berbeda dalam skema 10 fold validasi dan 80% tes data diuji sebanyak 50 kali.

Kata kunci: *classification, imbalance datasets, SMOTE*

1. Pendahuluan

Sebuah kelas pada dataset yang memiliki distribusi kelas tidak seimbang (*class imbalance*) menyebabkan terjadinya klasifikasi lebih condong ke kelas mayoritas daripada kelas minoritas. Ketidakseimbangan kelas dalam sekumpulan dataset merupakan sebuah permasalahan dalam machine learning, dimana jumlah kelas mayoritas (negatif) lebih besar daripada jumlah kelas minoritas (positif) [1]. Dampak negatif dari ketidakseimbangan data pada kinerja pengklasifikasi lebih diperburuk oleh faktor kesulitan data yang melekat seperti tumpang tindih kelas, terputus-putus kecil, adanya kebisingan, dan jumlah pengamatan pelatihan yang tidak mencukupi [7]. Class imbalance muncul terkait dengan evolusi pengembangan ilmu pengetahuan menjadi teknologi terapan dalam machine learning. Masalah class imbalance adalah permasalahan yang

umum ditemukan pada kumpulan dataset dari berbagai bidang, termasuk prediksi cacat software, diagnosis penyakit, deteksi tumpahan minyak dari citra satelit, dan deteksi penipuan kartu kredit online [2].

Pada beberapa kasus dimana dataset yang tidak seimbang memerlukan klasifikasi dua class dalam klasifikasinya, ketika distribusi alami dari dataset didominasi oleh satu kelas atau ketika kelas mayoritas positif melebihi kelas lain (kelas mayoritas atau negatif). Misalnya, dataset pasien tumor payudara menunjukkan masalah klasifikasi kelas ganda untuk diagnosis kanker payudara, yang terdiri dari kanker jinak dan kanker ganas. Dalam kasus medis, ada lebih sedikit pasien yang didiagnosis dengan kanker ganas (kelas minoritas) daripada pasien yang didiagnosis dengan kanker jinak (kelas mayoritas). Kelas minoritas dapat menyebabkan kesulitan dalam mendapatkan klasifikasi yang tepat. Kemiringan distribusi data telah membuat terbentuknya pengklasifikasi yang lebih kuat, yang merupakan masalah dasar dalam data mining [2].

Ada berbagai pendekatan untuk meningkatkan kinerja pemodelan prediktif dalam kumpulan data yang tidak seimbang. Metode pengambilan sampel adalah metode yang meningkatkan jumlah contoh minoritas dengan menghasilkan contoh sintetis, atau mengurangi jumlah contoh mayoritas dengan menghapus beberapa di antaranya. Pendekatan populer lainnya adalah menetapkan biaya kesalahan klasifikasi yang berbeda untuk berbagai kelas, pendekatan ini disebut biaya-sensitif. Dalam metode yang sensitif terhadap biaya, informasi tentang biaya kesalahan klasifikasi untuk setiap kelas diperlukan, sedangkan informasi semacam ini tidak diketahui. Satu-satunya fakta yang diketahui adalah bahwa biaya kesalahan klasifikasi untuk kelas minoritas lebih tinggi daripada biaya kesalahan klasifikasi untuk kelas mayoritas [3].

Biasanya, data level akan dijalankan sebagai langkah pra-pemrosesan untuk memastikan dataset yang tidak seimbang dapat disesuaikan dan kemiringan distribusi dapat dikurangi, dengan menggunakan semua jenis metode pengambilan sampel [2]. Salah satu pendekatan paling umum untuk mengurangi dampak negatif ketidakseimbangan data adalah pra-pemrosesan dataset asli dengan strategi tingkat data [7]. Pendekatan undersampling dan oversampling adalah teknik standar yang digunakan dalam menangani data yang tidak seimbang, namun keduanya memiliki keterbatasan masing-masing. Misalnya, undersampling menyebabkan lebih banyak penghapusan sampel data yang pada akhirnya menyebabkan masalah kekurangan data, dengan peningkatan kemungkinan kehilangan data penting. sementara oversampling menyebabkan duplikasi data asli, sehingga menyebabkan overfitting kelas minoritas [5]. *Overfitting* adalah masalah di mana perilaku pengklasifikasi terlalu sesuai dengan data percobaan. Ini berdampak buruk pada kinerja tes data, karena tidak semua informasi dalam data percobaan berguna [8].

Untuk diagnosis kesalahan yang tidak seimbang, algoritma yang paling umum digunakan adalah metode *oversampling*. Teknik random *oversampling* (RAMO) adalah metode *oversampling* yang paling sederhana. namun, dapat dengan mudah menghasilkan overfitting. Oleh karena itu, algoritma synthetic minority *over-sampling* (SMOTE) menggunakan data asli untuk mensintesis data minoritas baru yang berbeda dari aslinya, sehingga mengurangi dampak overfitting. Namun, SMOTE rentan terhadap generalisasi yang berlebihan dan rentan terhadap *noise*. Dengan demikian diusulkan teknik *oversampling Safe-level-SMOT*E, yang menentukan nilai level aman [4,5].

Hasil dari penelitian ini adalah untuk menangani distribusi kelas yang tidak seimbang pada dataset menggunakan teknik resampling SMOTE yang digunakan untuk menangani ketidakseimbangan kelas pada dataset dan dengan membandingkan antara penerapan teknik resampling dengan dataset yang tidak menggunakan teknik resampling.

Hairani dkk (2019) dengan judul Metode Klasifikasi Data Mining dan Teknik Sampling Smote Menangani *Class Imbalance* untuk Segmentasi *Customer* pada Industri Perbankan dengan menggunakan pendekatan Teknik sampling yaitu algoritma Smote yang dikombinasikan dengan metode J48, SVM dan *Naïve Bayes* menggunakan tools weka dan evaluasi kinerja *confusion matrix* yang membuktikan bahwa kombinasi metode J48 dan Smote mampu memiliki tingkat akurasi dan sensitivity menangani *class imbalance* pada dataset bank direct marketing dibandingkan kombinasi antara metode smote dengan SVM dan *Naïve Bayes*.

Nurul Fitrah dkk (2019) dengan judul *Handling Imbalanced Ratio for Class Imbalance Problem Using Smote* yang bertujuan untuk mengidentifikasi kinerja dari metode KNN dan C4.5 dengan validasi silang 10x lipat dengan hasil pengambilan data smote mengarah pada perbaikan klasifikasi untuk memberikan informasi mengenai skewness data dan mengamati IR ekstrim diambil dari pengambilan sampel.

2. Metode Penelitian

2.1. SMOTE

Synthetic Minority Over-sampling Technique (SMOTE) merupakan teknik Oversampling menyeimbangkan jumlah distribusi dataset pada kelas minoritas dengan cara mensintesis dataset minoritas hingga jumlah dataset menjadi seimbang dengan jumlah dataset pada kelas mayoritas. Penggunaan teknik *Oversampling* dapat menyebabkan *Overfitting*, metode *Synthetic Minority Over-sampling Technique* (SMOTE) ditawarkan untuk menangani *Overfitting* [1], yaitu dengan memanfaatkan

ketetanggaan terdekat (KNN) dari jumlah *Oversampling* yang dikehendaki.

$$Xsyn = Xi + (Xknn - Xi) * \sigma \quad (1)$$

Keterangan :
X syn : data sintesis yang akan diciptakan.
Xi : data yang akan direplikasi.
Xknn : data yang memiliki jarak terdekat dari data yang akan direplikasi.
σ : nilai random antara 0 dan 1.

2.1.1 *Data Splitting*

Data Splitting merupakan tahap pembagian data menjadi data training dan data testing. Berikut tabel 1 merupakan ilustrasi pembagian data secara umum.

Tabel 1. Ilustrasi Pembagian Data Secara Umum

Dataset	
Data Training	*Data Testing*
75%	25%

a. Pembagian data

Data training digunakan oleh algoritma klasifikasi. Model ini merupakan representasi pengetahuan yang akan digunakan untuk prediksi kelas data baru yang belum pernah ada. Data testing digunakan untuk mengukur sejauh mana classifier berhasil melakukan klasifikasi dengan benar .

Selain itu teknik pembagian data juga dapat menggunakan metode *K-fold cross-validation. K-fold cross validation* merupakan metode untuk mengestimasi performa model yang telah dibangun. Metode ini membagi data menjadi training dan testing sebanyak k bagian dari data. Fungsi dari k-fold cross validation agar tidak ada overlapping pada data testing [2]. Menunjukan bahwa percobaan yang dilakukan dengan menggunakan *10-fold cross validation*. Hal ini dapat dilihat berdasarkan jumlah iterasi yang tertera pada gambar. Kotak berwarna merah menunjukan testing set sedangkan kotak berwarna putih training set.

2.1.2 C 4.5 atau *Decision Tree*

Algoritma C4.5 merupakan algoritma yang digunakan untuk membentuk pohon keputusan. Didalam algoritma C4.5 ini, pohon-pohon keputusan yang dibentuk berdasarkan kriteria-kriteria pembentuk keputusan. Di dalam penyelesaian sebuah kasus menggunakan algoritma C4.5 ada beberapa elemen yang harus dipahami yaitu [3]:

a. Entropy
b. Info Gain
c. Split Info
d. Gain Ratio

Secara umum algoritma C4.5 untuk membangun pohon keputusan adalah sebagai berikut :

a. Pilih atribut sebagai akar
b. Buat cabang untuk tiap-tiap nilai
c. Bagi kasus dalam cabang
d. Ulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

Untuk memilih atribut sebagai akar, didasarkan pada nilai Gain Ratio tertinggi dari atribut-atribut yang ada. Untuk menghitung nilai Gain Ratio, digunakan rumus seperti pada persamaan berikut.

Entropy digunakan untuk menentukan seberapa informatif sebuah atribut input untuk menghasilkan atribut output.

Perhitungan nilai *entropy* dapat dilihat dari persamaan berikut.

$$Entropy(S) = \sum_{i=1}^{n} \quad -\pi * log2\pi \ldots\ldots\ldots\ldots (2)$$

Keterangan :

S : himpunan kasus
A : fitur
n : jumlah partisi S
Pi : proporsi dari Si terhadap S

Untuk menghitung gain digunakan rumus seperti tertera dalam persamaan berikut :

$$Gain(S,A) = Entropy(s) - \sum_{i=1}^{n} \frac{|Si|}{|S|} \text{Entropy(Si)} \ldots\ldots (2.3)$$

Keterangan :

S : himpunan kasus
A : atribut
n : jumlah partisi atribut A
|Si| : jumlah kasus pada partisi ke-i
|S| : jumlah kasus dalam S

Untuk mencari nilai Split Info digunakan rumus berikut ini.

$$Split\ Info\ (S,A) = \sum_{i=1}^{n} \quad \frac{Si}{S} x log_2 \frac{Si}{Si} \ldots\ldots\ldots\ldots\ldots (3)$$

Untuk mencari nilai Gain Ratio digunakan rumus berikut ini.

$$Gain\ Ratio\ (A) = \frac{Gain(A)}{Split\ Info\ (A)} \ldots\ldots\ldots\ldots\ldots (4)$$

2.1.3 Naïve Bayes

Naïve bayes adalah metode klasifikasi probabilitas berdasarkan teorema Bayes dengan asumsi independensi antar variabel prediktor. Secara sederhana, pengelompokan Naïve Bayes menganggap adanya suatu fitur tertentu dalam sebuah kelas tidak terkait dengan adanya fitur lainnya [7]. dengan persamaan berikut:

P(C | X)=(P(X|C)P(C))/(P(X))…….. (5)

Keterangan:

P(C|X) : posterior probability kelas (C, target) yang diberikan prediktor (X, atribut).
P(C) probabilitas kelas sebelumnya.
P(X|C) : Kemungkinan yang merupakan probabilitas prediktor kelas yang diberikan. .
P(X) : probabilitas prediktor sebelumnya

2.1.4 KNN

Algoritma KNN adalah sebuah algoritma untuk mengklasifikasikan objek berdasarkan data latih yang mempunyai jarak paling dekat dengan objek tersebut KNN termasuk algoritma supervised learning. Algoritma klasifikasi KNN memiliki konsep sederhana, algoritma ini,bekerja berdasarkan jarak terpendek dari data uji ke data latih untuk menentukan kelas. KNN memiliki beberapa kelebihan yaitu tahan terhadap data yang memiliki noise dan efektif terhadap data latih yang berjumlah besar dan memiliki performa cukup baik [8]. Ada banyak cara untuk mengukur jarak kedekatan antar data pada algoritma KNN diantaranya menggunakan *Euclidean distance*. *Euclidean distance* merupakan cara yang sering digunakan untuk menghitung jarak antar data. Jarak ini digunakan untuk menguji interpretasi kedekatan jarak antara dua objek. Berikut adalah persamaan dari Euclidean distance :

$$dist = \sqrt{\sum_{i=1}^{n} \quad (X_{i2} - X_{i1}) \quad ^2} \text{ ……(6)}$$

Keterangan :

dist : Jarak.
Xi2 : Data uji.
Xi1 : Data Sampel.
i : Atribut, dan
n : Jumlah atribut.

2.1.4 SVM

Support vector machines (SVM) adalah suatu metode yang handal dalam menyelesaikan masalah klasifikasi data. Permasalahan SVM dipecahkan dengan menyelesaikan persamaan Lagrangian yang merupakan bentuk dual dari SVM melalui *quadratic programming* [9].

$$x_n = \frac{0.8*(x-a)}{b-a} + 0.1 \text{…….(7)}$$

Dimana :
Xn = Nilai ke-n
a = Nilai angka terendah
b = Nilai angka tertinggi
0.8 dan 0.1 = Ketepatan

Masalah dasar dari SVM adalah menentukan suatu hyperplane <w, x> + b = 0 memisahkan data xj yang terdiri dari dua kelas, yaitu yi = {+1, -1}, dengan margin maksimal. Margin disini merupakan jarak antara hyperplane ke masing-masing kelas data. Selanjutnya, hyperplane ini akan menjadi fungsi keputusan f(x) untuk masalah klasifikasi dua kelas diatas.

$$f(\phi(x)) = sign(w.\phi(x) + b) = sign\left(\sum_{i=1}^{N} \alpha_i y_i \phi(x_i)^T . \phi(x) + b\right) \quad (8)$$

Keterangan :

w : nilai *weight*
x : nilai variabel input
b : nilai bias Rumus di atas digunakan untuk perhitungan hasil prediksi.

2.1.5 *Confusion Matrix*

Pada data mining untuk mengukur kinerja dari model yang dihasilkan salah satunya menggunakan *Confusion Matrix*. *Confusion Matrix* adalah suatu metode yang digunakan untuk melakukan perhitungan akurasi pada konsep data mining. Presisi atau *confidence* adalah proporsi kasus yang diprediksi positif yang juga positif benar pada data sebenarnya. *Recall* atau *sensitivity* adalah proporsi kasus positif yang sebenarnya diprediksi positif secara benar [6].

Tabel 2. *Predicted Class and Actual Class*

Predicted Class	Actual Class	
	+	-
+	True Positives (TP)	False Positives (FP)
-	False Negatives (FN)	True Negatives (TN)

2.1.6 Confusion matrix

Perhitungan akurasi dengan tabel *Confusion Matrix* adalah sebagai berikut.

Akurasi=(TP+TN)/(TP+TN+FP+FN)……(9)

Presisi didefinisikan sebagai rasio item relevan yang dipilih terhadap semua item yang terpilih. Presisi dapat diartikan sebagai kecocokan antara permintaan informasi dengan jawaban terhadap permintaan tersebut. Rumus presisi adalah:

Presisi=TP/(TP+FP) …….. (10)

Recall didefinisikan sebagai rasio dari item relevan yang dipilih terhadap total jumlah item relevan yang tersedia. Recall dihitung dengan rumus:

Recall atau *sensitivity*=TP/(TP+FN)…… (11)

Presisi dan *Recall* dapat diberi nilai dalam bentuk angka dengan menggunakan perhitungan persentase (1-100%) atau dengan menggunakan bilangan antara 0-1. Sistem rekomendasi akan dianggap baik jika nilai presisi dan recallnya tinggi.

3. Implementasi dan pembahasan

3.1 Alur penelitian

Gambar 1. Alur Penelitian

Pada penelitian kali ini digunakan dataset public yang diambil dari *keels imbalance datase*t, dengan nama ecoli2. berikut isi label dari dataset yang kami gunakan

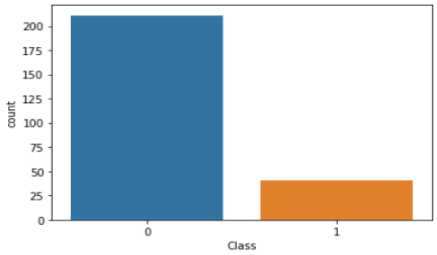
```
Data columns (total 8 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Mcg     336 non-null    float64
 1    Gvh    336 non-null    float64
 2    Lip    336 non-null    float64
 3    Chg    336 non-null    float64
 4    Aac    336 non-null    float64
 5    Alm1   336 non-null    float64
 6    Alm2   336 non-null    float64
 7   Class   336 non-null    int64
dtypes: float64(7), int64(1)
```

Gambar 2. .Isi Label Dataset

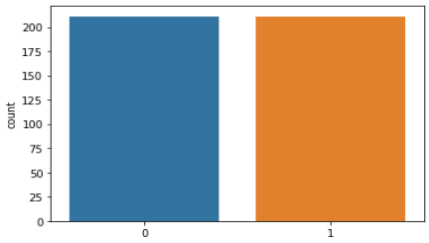
kami menggunakan l*ibrary imblearn* untuk implementasi *oversampling* SMOTE, dan sklearn untuk implementasi dari algoritma klasifikasi

3.2 Pembahasan

Penelitian membandingkan antara algoritma klasifikasi yang langsung diimplementasikan dengan dataset imbalance dan algoritma klasifikasi diimplementasikan ke dataset yang disampling terlebih dahulu. Berikut adalah hasil penelitian yang sudah dilakukan :



Gambar 3. Perbandingan kelas sebelum diimplementasikan sampling SMOTE



Gambar 4. perbandingan kelas setelah dilakukan implementasi SMOTE

Implementasi algoritma klasifikasi langsung ke dataset imbalance tanpa resampling SMOTE :

Tabel 3. Hasil implementasi algoritma tanpa resampling

No	Algoritma	*accuracy*	*precision*	*recall*
1	SVM	0.89	0.625	0.45
2	NB	0.36	0.17	1.0
3	KNN	0.94	0.8	0.72
4	DT	0.92	1.0	0.45

Implementasi algoritma klasifikasi setelah dilakukan resampling SMOTE pada dataset dengan parameter smote (*random_state* = 2) :

Tabel 4. Hasil implementasi algoritma setelah resampling

No	Algoritma	*accuracy*	*precision*	*recall*
1	SVM	0.88	0.53	0.82
2	NB	0.94	0.8	0.73
3	KNN	0.95	0.82	0.82
4	DT	0.94	0.8	0.72

Besar dari data *splitting* yang digunakan untuk data *testing* dan *data training* berpengaruh terhadap hasil dari implementasi algoritma SMOTE dengan algoritma Klasifikasi pada penelitian ini kami menggunakan pembagian 75% untuk data training dan 25% untuk data testing. kemudian nilai dari parameter yang pada implementasi SMOTE dan implementasi Algoritma klasifikasi juga berpengaruh terhadap hasil implementasi. Sehingga perlu diimplementasikan hyperparameter tuning untuk mendapatkan kombinasi parameter yang maksimal.

4. Kesimpulan

Uji perbandingan teknik resampling dilakukan dengan membandingkan aplikasi resampling dengan dataset yang tidak menggunakan resampling. Teknik Resampling SMOTE digunakan untuk mengelola ketidakseimbangan kelas dalam kumpulan data.

Berdasarkan hasil percobaan dan pengujian yang dilakukan, maka dapat diambil kesimpulan sebagai berikut:

1. implementasi resampling smote yang dikombinasikan dengan algoritma klasifikasi dapat meningkatkan akurasi pada algoritma NB sebesar 24%, algoritma KNN sebesar 1%, dan algoritma DT sebesar 2 %.
2. implementasi resampling semote yang dikombinasikan dengan algoritma svm mengalami penurunan akurasi sebesar 1%.

Kritik dan Saran :

Paper yang telah dibuat ini jauh dari kata sempurna karena keterbatasan pengetahuan tim penulis dan saat pembuatan paper ini tim penulis sedang isolasi mandiri karena terkonfirmasi positif virus corona. Berikut adalah saran yang dapat digunakan sebagai acuan penelitian selanjutnya.

1. Perlu penggunaan *hyperparameter tuning* pada implementasi SMOTE dan implementasi algoritma klasifikasi untuk mendapatkan hasil yang maksimal.
2. Perlu dilakukan pengujian pada level i*mbalance* yang lebih tinggi maupun lebih rendah untuk mendapatkan konsistensi hasil dari penerapan kombinasi algoritma SMOTE dengan algoritma Klasifikasi.

Daftar Rujukan

[1] Hairani, Noor Akhmad Setiawan, Teguh Bharata Adji, 2019. Metode Klasifikasi Data Mining dan Teknik Sampling SMOTE Menangani Class Imbalance untuk Segmentasi Customer pada Industri Perbankan. ISBN 978-602-99334-5-1.

[2] Nurulfitrah Noorhalim, Aida Ali and Siti Mariyam Shamsuddin, 2019. Handling Imbalanced Ratio for Class Imbalance Problem Using SMOTE. In: Kor, L.-K., Ahmad, A.-R., Idrus, Z., Mansor, K.A. (Eds.), *Proceedings of the Third International Conference on Computing, Mathematics and Statistics (iCMS2017)*. Springer Nature: Singapore.

[3] S. Piri, D. Delen, and T. Liu, "A synthetic informative minority over-sampling (SIMO) algorithm leveraging support vector machine to enhance learning from imbalanced datasets," *Decis. Support Syst.*, vol. 106, pp. 15–29, 2018, doi: 10.1016/j.dss.2017.11.006.

[4] J. Wei, H. Huang, L. Yao, Y. Hu, Q. Fan, and D. Huang, "New imbalanced bearing fault diagnosis method based on Sample-characteristic Oversampling TechniquE (SCOTE) and multi-class LS-SVM," *Appl. Soft Comput.*, vol. 101, p. 107043, 2021, doi: 10.1016/j.asoc.2020.107043.

[5] Asniar, N. U. Maulidevi, and K. Surendro, "SMOTE-LOF for noise identification in imbalanced data classification," *J. King Saud Univ. - Comput. Inf. Sci.*, no. xxxx, 2021, doi: 10.1016/j.jksuci.2021.01.014.

[6] M. Koziarski, "Potential Anchoring for imbalanced data classification," *Pattern Recognit.*, vol. 120, p. 108114, 2021, doi: 10.1016/j.patcog.2021.108114.

[7] V. P. K. Turlapati and M. R. Prusty, "Outlier-SMOTE: A refined oversampling technique for improved detection of COVID-19," *Intell. Med.*, vol. 3–4, no. July, p. 100023, 2020, doi: 10.1016/j.ibmed.2020.100023.

[8] C. Wang, C. Deng, Z. Yu, D. Hui, X. Gong, and R. Luo, "Adaptive ensemble of classifiers with regularization for imbalanced data classification," *Inf. Fusion*, vol. 69, no. December 2019, pp. 81–102, 2021, doi: 10.1016/j.inffus.2020.10.017.

[9] Z. Xu, D. Shen, T. Nie, and Y. Kou, "A hybrid sampling algorithm combining M-SMOTE and ENN based on Random forest for medical imbalanced data," *J. Biomed. Inform.*, vol. 107, no. May 2019, p. 103465, 2020, doi: 10.1016/j.jbi.2020.103465.