Prediksi Retweet Berdasarkan Fitur Pengguna, Konten, dan Waktu Menggunakan Metode Klasifikasi ANN-Cat Swarm Optimization

1st Muhammad Hasan Syadzily
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
[1]hasansyadzily@student.telkomuniversity.ac.id

2nd Jondri
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
jondri@telkomuniversity.ac.id

3rd Kemas Muslim Lhaksmana
Fakultas Informatika
Universitas Telkom
Bandung, Indonesia
kemasmuslim@telkomuniversity.ac.id

Abstrak **- Twitter merupakan salah satu sarana microblogging populer saat ini yang memungkinkan penggunanya untuk mengirim pesan berupa teks, gambar, atau video, serta berbagi informasi dengan cepat. Salah satu fitur utama di Twitter adalah retweet, dengan fitur ini pengguna dapat memposting ulang pesan yang diunggah oleh orang lain. Penelitian ini bertujuan untuk membangun model prediksi retweet dengan metode klasifikasi ANN yang dioptimasi oleh algoritma CSO menggunakan fitur berbasis pengguna, konten, dan waktu. Masalah yang dihadapi dalam penelitian ini yaitu ketidakseimbangan kelas yang sering terjadi pada data retweet. Untuk mengatasi masalah tersebut, digunakan teknik oversampling dan undersampling. Hasil evaluasi pada penelitian ini menunjukkan bahwa proses klasifikasi ANN dengan CSO dapat mencapai nilai akurasi sebesar 86.70% dan F1-Score sebesar 86.61% dengan melakukan teknik undersampling.**

Kata kunci : **retweet, prediksi, ANN, CSO, undersampling**

I. PENDAHULUAN

A. Latar Belakang

Pada masa digital ini, informasi dapat menyebar secara luas dan cepat melalui media sosial. Saat ini, Twitter merupakan salah satu situs jejaring sosial yang paling banyak digunakan. Salah satu mekanisme difusi informasi pada Twitter adalah *retweet*[1]. Retweet adalah kegiatan dimana pengguna memposting ulang *tweet* pengguna lainnya. Dengan memprediksi *tweet* yang akan mendapatkan banyak *retweet* dapat membantu menyelesaikan masalah seperti analisis bisnis dan pemasaran yang membutuhkan *engagement* yang besar[2].

Pada penelitian sebelumnya, Thi Bich Ngoc Hoang dkk. Pada tahun 2017 melakukan penelitian dengan tujuan untuk melakukan prediksi *retweet*. Menggunakan metode klasifikasi Random Forest dengan fitur *user-based, time-based,* dan *content-based*[3]. Penelitian selanjutnya, Rakes dkk. Membuat sistem prediksi *retweet* berdasarkan fitur pengguna dengan metode klasifikasi SVM pada tahun 2021[4]. Penelitian berikutnya, Hamidan dkk. Melakukan studi mengenai prediksi *retweet* menggunakan metode klasifikasi ANN pada tahun 2021, dengan fitur berbasis pengguna dan konten[5].

Tujuan penelitian ini, yaitu untuk membangun model prediksi *retweet* dengan metode klasifikasi ANN yang dioptimasi algoritma Cat Swarm Optimization (CSO) menggunakan fitur pengguna, konten, dan waktu. CSO berpotensi untuk meningkatkan performansi model prediksi *retweet* dengan mencari kombinasi parameter terbaik sehingga memperoleh struktur jaringan ANN yang optimal. CSO merupakan algoritma pengoptimalan yang dikembangkan berdasarkan perilaku umum kucing saat berburu mangsanya. Peneliti memilih metode ini karena pada penelitian sebelumnya ANN-CSO dapat memperoleh hasil performansi yang baik dibandingkan algoritma optimasi lainnya[6]–[8].

B. Topik dan Batasannya

Berdasarkan hal tersebut, topik pada penelitian ini membahas tentang pembuatan model prediksi *retweet* menggunakan fitur berbasis pengguna, konten, dan waktu dengan memanfaatkan metode klasifikasi ANN yang dioptimasi algoritma CSO. Batasan dari penelitian ini adalah *dataset* yang diambil dengan metode *crawling data* menggunakan kata kunci “capres”. Data yang diambil terdiri dari *tweet* berbahasa Indonesia dalam rentang waktu bulan Juni – Juli 2023.

C. Tujuan

Penelitian ini bertujuan untuk meningkatkan performansi model prediksi *retweet* dengan metode klasifikasi ANN yang dioptimasi CSO menggunakan fitur pengguna, konten, dan waktu. Serta mengetahui dampak penggunaan metode *resampling* terhadap performansi model klasifikasi dalam mengatasi ketidakseimbangan kelas pada data *tweet*

D. Organisasi Tulisan

Penelitian ini akan membahas tentang metodologi dan studi terkait serupa dari penelitian sebelumnya yang berkaitan dengan topik *retweet*. Bagian berikutnya, akan menjelaskan urutan langkah yang akan diambil untuk membangun model prediksi *retweet* menggunakan informasi data pengguna, konten, dan waktu. Setelah itu, berbagai skenario pengujian akan dilakukan untuk mengevaluasi dan menganalisis hasil penelitian. Kesimpulan dan saran untuk penelitian lebih lanjut disajikan pada bagian terakhir

II. STUDI TERKAIT

Thi Bich Ngoc Hoang dkk. Pada tahun 2017 melakukan penelitian dengan judul "*Predicting Information Diffusion on Twitter – Analysis of predictive features*". Dengan tujuan memprediksi apakah sebuah *tweet* akan di-*retweet* atau tidak. Serta mengetahui potensi postingan akan tersebar. Fitur yang mereka gunakan dalam penelitian ini didasarkan oleh tiga jenis fitur yaitu, *user-based, content-based,* dan *time-based* dengan menggunakan algoritma Random Forest sebagai modelnya. Hasil eksperimen dari penelitian ini cukup baik dan mendapat nilai akurasi diantara 70-82%[3].

Rakes dkk. Pada tahun 2021 melakukan penelitian dengan judul "*Prediksi Retweet Berdasarkan Feature User-Based Menggunakan Metode Klasifikasi Support Vector Machine*". Dengan menggunakan teknik Support Vector Machine, Rakes mengembangkan sistem untuk memprediksi *retweet* berdasarkan informasi data pengguna. Penelitian ini menunjukkan bahwa teknik SVM dapat mencapai nilai akurasi sebesar 70%[4].

Hamidan dkk. Pada tahun 2021 melakukan penelitian dengan judul "*Prediksi Retweet Menggunakan Fitur Berbasis Pengguna dan Fitur Berbasis Konten dengan Metode Klasifikasi ANN*". Studi yang dilakukan Hamidan menggunakan klasifikasi ANN untuk memprediksi *retweet* berdasarkan informasi data pengguna dan konten. Hasil eksperimen dari penelitian ini menunjukkan dengan menggunakan metode klasifikasi ANN dapat meraih tingkat akurasi sebesar 86%[5].

A. Twitter

Salah satu sarana *microblogging* yang populer, yaitu Twitter memungkinkan penggunanya untuk berbagi informasi dalam bentuk teks, gambar, dan video. Retweet merupakan salah satu mekanisme difusi informasi pada Twitter. Retweet adalah kegiatan dimana pengguna memposting ulang *tweet* yang serupa dengan *tweet* asli pengguna lainnya. Tidak hanya sebatas memposting ulang, pengguna juga bisa melakukan *quote retweet*, yaitu melakukan *retweet* beserta tambahan komentar dari pengguna yang melakukan *retweet*. Prediksi *retweet* memberikan pemahaman bagaimana informasi menyebar di Twitter[9], [10].

B. Fitur Retweet

Penelitian ini memanfaatkan fitur retweet yang dibagi menjadi beberapa kelompok, seperti yang telah dilakukan pada penelitian sebelumnya[3]. Fitur-fitur tersebut meliputi fitur pengguna, konten, dan waktu. Selain itu, ditambahkan juga fitur baru, yaitu "status_retweet". Seperti yang dapat dilihat pada Tabel 2.1.

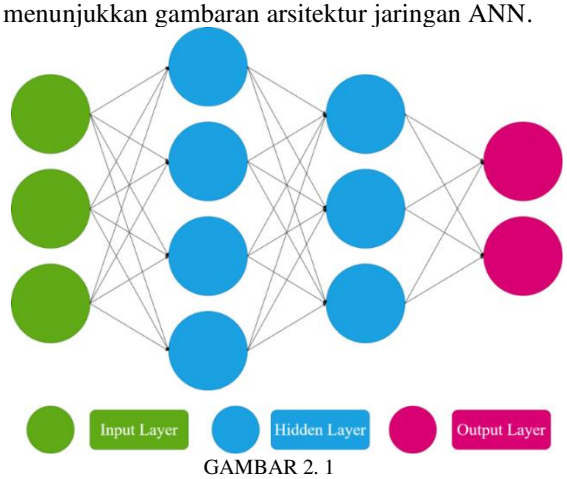
TABEL 2. 1
Fitur Retweet

No	Jenis Fitur	Nama Fitur	Tipe	Deskripsi
1	User-Based	Total_of_tweets	Numeric	Total *tweet* yang sebelumnya diposting pengguna di *timeline*
2		No_of_followers	Numeric	Jumlah orang yang mengikuti pengguna
3		No_of_followees	Numeric	Jumlah orang yang diikuti pengguna
4		Age_of_account	Numeric	Jumlah hari sejak akun dibuat
5		No_of_favourite	Numeric	Jumlah *tweet* yang disukai pengguna di timeline
6		Aver_favou_per_day	Numeric	Rata-rata favorit per hari
7		Aver_tweets_per_day	Numeric	Rata-rata *tweet* per hari
8		User_name_len	Numeric	Panjang nama pengguna
9	Content-Based	Contain_location	Boolean	Tweet berisi nama lokasi
10		Contain_org	Boolean	Tweet berisi nama organisasi
11		Contain_tvshow	Boolean	Tweet berisi nama acara TV
12		Sentiment_level	{positive, negative, neutral}	Tweet diklasifikasikan ke dalam level sentimen
13		Contain_video	Boolean	Tweet berisi video
14		Contain_picture	Boolean	Tweet berisi gambar
15		Contain_upper	Boolean	Tweet berisi huruf kapital
16		Contain_number	Boolean	Tweet berisi nomor
17		Contain_excl	Boolean	Tweet berisi tanda seru
18		Contain_user_mentioned	Boolean	Tweet menyebutkan nama pengguna lain
19		Contain_url	Boolean	Tweet yang mengandung link URL
20		Contain_hashtag	Boolean	Tweet mengandung tagar

21		Opt_length	Boolean	Panjang konten diantara 70 sampai 100 karakter
22		Len_of_text	Boolean	Panjang dari konten
23		Status_retweet	Boolean	Tweet yang mendapat *retweet*
24	Time-Based	Is_post_at hol	Boolean	Tweet diposting pada hari libur
25		Is_posted_at_noon	Boolean	Tweet diposting dari jam 11:00 sampai 13:00
26		Is_posted_at_eve	Boolean	Tweet diposting dari jam 18:00 sampai 21:00
27		Is_posted_at_wee	Boolean	Tweet diposting pada akhir pekan

C. Artificial Neural Network

ANN merupakan teknik klasifikasi untuk membuat model sistem yang memiliki sifat non-linier dan kompleks. Karakteristik dari ANN dipengaruhi oleh fungsi aktivasi, arsitektur jaringan, dan proses pelatihan. Neuron yang ditempatkan arsitektur jaringan pada suatu lapisan terdiri dari *input layer, output layer,* dan *hidden layer*, yang memiliki jumlah satu atau lebih[11]. Gambar 2.1 menunjukkan gambaran arsitektur jaringan ANN.



GAMBAR 2. 1
Struktur Jaringan ANN

1. Input Layer

Pada lapisan ini berisi *neuron-neuron* yang berfungsi untuk menerima *input* data sebagai langkah awal dari proses *learning* dan *recognition* oleh ANN[12].

2. Hidden Layer

Pada lapisan ini berisi *neuron-neuron* yang berperan dalam mengolah dan mengganti kontribusi informasi dari *input layer* sebelum akhirnya menghasilkan *output layer*[12].

3. Output Layer

Pada lapisan ini berisi *neuron-neuron* yang memberikan respons atau keluaran berdasarkan hasil dari proses pengolahan pada *hidden layer*, sebagai hasil akhir dari proses ANN terhadap data *input* yang diberikan[12].

D. Cat Swarm Optimization

Cat Swarm Optimization (CSO) merupakan algoritma pengoptimalan berdasarkan Swarm Intelligence (SI). Algoritma CSO dikembangkan berdasarkan perilaku umum kucing. Kucing menghabiskan sebagian besar waktunya untuk beristirahat dan mengamati lingkungannya daripada mengejar sesuatu karena hal ini menyebabkan sumber energi yang berlebihan. Metode ini dibagi menjadi dua mode yaitu, "*seeking mode*" dan "*tracing mode*" yang melakukan dua proses berbeda dalam algoritma untuk meniru perilaku kucing. Seeking mode memodelkan perilaku kucing saat beristirahat dan mengamati lingkungannya, sedangkan *tracing mode* memodelkan perilaku kucing saat belari mengejar target[6], [7].

1. Seeking Mode

Prosedur *seeking mode* memiliki empat faktor penting diantaranya adalah: Seeking Memory Pool (SMP), Seeking Range of the selected Dimension (SRD), Counts of Dimension to Change (CDC), dan Self Position Consideration (SPC) seperti yang dijelaskan oleh Chu dkk.[6], [7]. Seeking mode dari algoritma CSO dapat digambarkan sebagai berikut:

a. Langkah-1: Buat *j copies* dari posisi cat_k, *where* $j = SMP$.
If value SPC = true, let j = (SMP-1) then pertahankan posisi saat ini sebagai salah satu kandidat.
b. Langkah-2: Untuk setiap *copy* berdasarkan CDC, tambahkan secara acak atau kurangi persentase SRD ke nilai saat ini dan mengganti nilai sebelumnya.
c. Langkah-3: Menghitung Fitness Score (FS) dari semua poin kandidat.
d. Langkah-4: Jika semua FS tidak sama persis, hitunglah pemilihan probabilitas dari setiap poin kandidat, jika tidak atur semua probabilitas pemilihan masing-masing kandidat poin ke 1.
e. Langkah-5: Pilih secara acak titik pindah tujuan dari poin kandidat, dan ganti posisi cat_k.

$$P_i = \frac{|FS_i - FS_b|}{FS_{max} - FS_{min}} \quad (1) \\ < i < j$$

2. Tracing Mode

Tracing mode dari algoritma CSO memodelkan perilaku kucing saat mengejar target. Begitu seekor kucing memasuki *tracing mode*, ia bergerak sesuai dengan *velocities* untuk setiap *dimension*. Aksi *tracing mode* menurut Chu dkk. dapat digambarkan sebagai berikut[6], [7]:

a. Langkah-1: Perbaharui *velocities* untuk setiap *dimension*.
b. Langkah-2: Periksa apakah *velocities* berada di *range of maximum velocity*. Jika *velocity* baru mengalami

overrange, maka akan diatur sama dengan batas.
Langkah-3: Perbaharui posisi dari cat_k.

$$v_{k,d} = v_{k,d} + r_1 * c_1 * (x_{best,d} - x_{k,d}), d = 1,2,\dots,M \quad (2)$$

dimana $x_{best,d}$ adalah posisi kucing, yang memiliki *fitness value* terbaik, $x_{k,d}$ adalah posisi dari cat_k, c_1 adalah konstan, dan r_1 adalah *random value* dalam *range* [0,1].

$$x_{k,d} = x_{k,d} + v_{k,d} \quad (3)$$

3. CSO Process

Tahapan CSO dapat digambarkan dalam 6 langkah berikut:

a. Langkah-1: Membuat *N cat* ke dalam proses.
b. Langkah-2: Secara acak sebarkan *cat* ke dalam M-dimensional solution space dan secara acak memilih *value*, yang berada dalam *range of maximum velocity,* kepada *velocities* dari setiap *cat*. Kemudian secara acak memilih jumlah *cat* dan mengaturnya ke *tracing mode* berdasarkan Mixture Ratio (MR), dan sisa yang lainnya diatur ke *seeking mode*.
c. Langkah-3: Evaluasi *fitness value* dari setiap *cat* dengan menerapkan posisi *cat* ke dalam *fitness function*, dan simpan *cat* terbaik ke dalam memori. Perhatikan posisi *cat* (x_{best}) yang mewakili solusi terbaik.
d. Langkah-4: Pindahkan *cat* sesuai dengan *flags*, if cat_k is in *seeking mode*, terapkan *cat* ke dalam proses *seeking mode*, jika tidak terapkan ke dalam proses *tracing mode*.
e. Langkah-5: Pilih Kembali jumlah *cat* dan atur mereka ke *tracing mode* berdasarkan Mixture Ratio (MR), lalu atur *cat* lainnya ke *seeking mode*.
f. Langkah-6: Periksa *termination condition*, jika memenuhi, hentikan program, jika tidak ulangi Langkah-3 sampai Langkah-5.

E. Confusion Matrix

Confusion Matrix digunakan untuk menentukan kinerja model prediksi, pengukuran dilakukan dalam beberapa skenario. Beberapa metrik digunakan untuk mengukur *retweet*. Confusion Matrix digunakan untuk evaluasi, dengan mengambil data klasifikasi dalam bentuk True Negative, True Positive, False Negative, False Positive[13]. Tabel 2.2 merupakan gambaran *confusion matrix* secara umum.

TABEL 2. 2
Confusion Matrix

		Predicted Values	
		Positive	**Negative**
Actual Values	**Positive**	TP (True Positive)	FP (False Positive)
	Negative	FN (False Negative)	TN (True Negative)

Accuracy, merupakan tingkat kesesuaian antara nilai prediksi dengan nilai asli dalam perbandingan dengan jumlah keseluruhan data. Akurasi tinggi menandakan tingkat performansi yang baik. Berikut adalah persamaan untuk menghitung nilai *accuracy*:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (4)$$

Recall, merupakan rasio dari True Positive dibagi dengan jumlah True Positive ditambah False Negative. Recall mengukur sejauh mana model mampu mendeteksi data positif secara keseluruhan. Berikut adalah persamaan untuk menghitung nilai *recall*:

$$Recall = \frac{TP}{TP + FN} \quad (5)$$

Precision, merupakan rasio jumlah True Positive dibagi dengan jumlah total prediksi yang diklasifikasikan dalam suatu kelas tertentu. Presisi mengukur sejauh mana prediksi positif yang dibuat oleh model adalah benar. Berikut adalah persamaan untuk menghitung nilai *precision*:

$$Precision = \frac{TP}{TP + FP} \quad (6)$$

F1-Score, merupakan ukuran harmonis yang menyesuaikan nilai *precision* dan *recall* dalam sebuah formula tunggal. Ini memberikan gambaran keseluruhan tentang kinerja model dengan mempertimbangkan baik *precision* maupun *recall*. Berikut adalah persamaan untuk menghitung nilai F1-Score:

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precision + Recall} \quad (7)$$

F. SMOTE

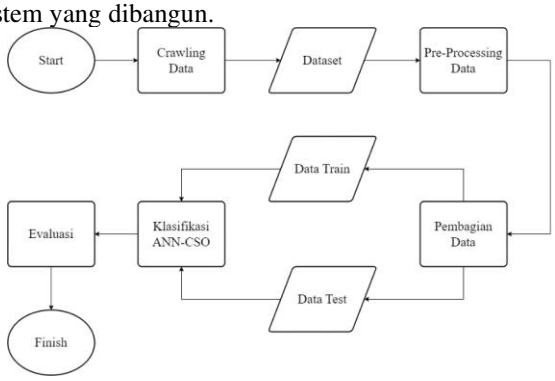
SMOTE (Synthetic Minority Over-sampling Technique) digunakan untuk menyeimbangkan data yang bias terhadap satu kelompok[14]. SMOTE menciptakan *instance* baru dari kelas minoritas dengan mereplikasi data saat ini dan menggabungkannya dengan tetangga terdekatnya [14]. Pendekatan ini membantu meningkatkan jumlah sampel dalam kelas minoritas tanpa mengubah jumlah kelas mayoritas, sehingga menciptakan keseimbangan dalam data [15].

G. NearMiss

NearMiss adalah sekelompok metode *undersampling* yang memilih sampel berdasarkan seberapa jauh jarak mereka dari kelas mayoritas. Varian dari NearMiss terdiri dari NearMiss-1, NearMiss-2, dan NearMiss-3. Tujuan utama NearMiss-1 yaitu memilih tiga bagian terdekat dari kelas minoritas dengan jarak rata-rata terpendek[16].

III. SISTEM YANG DIBANGUN

Pada penelitian ini, metode klasifikasi Artificial Neural Network (ANN) dioptimasi dengan Cat Swarm Optimization (CSO) yang digunakan untuk mengembangkan model prediksi *retweet* pada data Twitter. Gambar 3.1 menampilkan *flowchart* perancangan sistem yang dibangun.



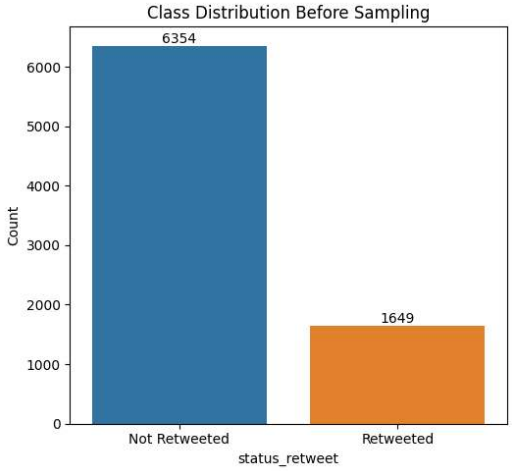
GAMBAR 3. 1
Flowchart Rancangan Sistem

A. Crawling Data

Pengumpulan data pada penelitian ini, menggunakan metode *crawling data* dengan memanfaatkan *platform* netlytic. Pencarian data dilakukan dengan menggunakan kata kunci "capres". Data yang diambil terdiri dari *tweet* berbahasa Indonesia pada rentang waktu bulan Juni - Juli 2023. Pengumpulan data menghasilkan 10.329 data *tweet*.

B. Pre-Processing Data

Setelah data dikumpulkan, data tersebut dibersihkan dengan menghapus data duplikat dan memastikan tidak ada *missing value*. Setelah proses pembersihan, jumlah data menjadi 8003 data *tweet*. Selanjutnya, fitur dalam dataset diubah ke dalam bentuk vektor ciri yang dikelompokkan menjadi 3 jenis fitur, yaitu *user-based, content-based,* dan *time-based*. Data yang terkumpul kemudian diperiksa untuk kelas yang tidak seimbang, dengan kelas “Retweeted” mewakili *tweet* yang di-*retweet* dan kelas “Not Retweeted” mewakili *tweet* yang tidak di-*retweet*. Hasil pengecekan menunjukkan ketidakseimbangan kelas, di mana kelas "Not Retweeted" memiliki data yang lebih banyak (6354 data) dibandingkan dengan kelas "Retweeted" (1649 data). Kemudian, menggunakan dua metode *resampling* dari *library* imblearn untuk mengatasi masalah ketidakseimbangan kelas ini, pertama menggunakan metode *oversampling* dengan SMOTE, kedua menggunakan metode *undersampling* dengan NearMiss.



GAMBAR 3. 2
Distribusi Kelas Retweet

C. Klasifikasi ANN-CSO

Algoritma Cat Swarm Optimization (CSO) digunakan untuk melatih Artificial Neural Network (ANN) dengan tujuan untuk mencari struktur jaringan ANN yang optimal. Data train dan data test akan digunakan untuk mencari nilai bobot dan bias terbaik dalam ANN dengan meminimalisir fungsi objektif menggunakan algoritma CSO. Dalam proses perulangan ANN-CSO, nilai bobot dan bias yang dioptimalkan akan diiterasi dan dihubungkan dengan struktur jaringan ANN. Pada tahap ini, dilakukan pengukuran *accuracy* dan F1-score menggunakan metode klasifikasi ANN dengan CSO sebagai algoritma optimasinya.

IV. EVALUASI

Penelitian ini menggunakan dataset yang terdiri dari 8003 data *tweet* dengan fitur *user-based, content-based,* dan *time-based*. Evaluasi dilakukan untuk menganalisis performansi model klasifikasi pada tiga skenario berbeda. Pertama, pada kondisi data saat *imbalanced class*. Kedua, setelah dilakukan *resampling* dengan metode *oversampling* menggunakan SMOTE. Dan ketiga, setelah dilakukan *resampling* dengan metode *undersampling* menggunakan NearMiss. Evaluasi bertujuan untuk menilai efektivitas model dalam mengatasi masalah ketidakseimbangan kelas pada data *tweet*.

A. Hasil Pengujian

1. Skenario Pengujian 1

Pada skenario pengujian pertama, evaluasi performansi model klasifikasi dilakukan pada kondisi data saat *imbalanced class* tanpa melakukan *resampling*. Hasil pengujian menunjukkan bahwa *default model* memiliki rata-rata akurasi sebesar 61.31% dan F1-Score sebesar 64.88%. Setelah dioptimasi CSO, model mengalami peningkatan performansi dengan rata-rata akurasi terbaik mencapai 80.37% dan F1-Score terbaik sebesar 72.06%. Pengujian dilakukan 10 kali karena penggunaan algoritma CSO dapat menghasilkan variasi hasil yang berbeda dalam *hyperparameter tuning*.

Penggunaan *cross-validation 2-fold* selama 100 iterasi untuk menjaga keseimbangan antara informasi evaluasi yang cukup dan efisiensi komputasi. Selain melibatkan CSO dalam proses *hyperparameter tuning*, juga melibatkan parameter *default* ANN dari *library* Scikit-learn. Parameter-parameter ini meliputi *hidden layer sizes* sebanyak 100, *activation* relu, dan *solver* adam. Hasil dari model yang menggunakan parameter default ini digunakan sebagai pembanding dalam proses evaluasi hasil *hyperparameter tuning* dengan CSO.

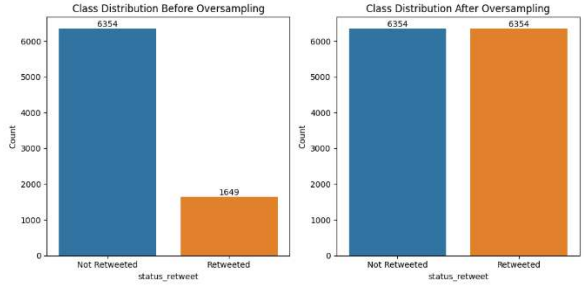
Penggunaan CSO membantu mengidentifikasi kombinasi *hyperparameter* yang optimal, serta meningkatkan kinerja model dalam mengklasifikasikan data yang tidak seimbang, sehingga merespons dengan lebih baik pada kelas minoritas, yang terlihat dalam peningkatan F1-Score pada *best model*. Kombinasi parameter terbaik untuk skenario pengujian pertama adalah *hidden layer sizes* (17, 25, 25, 20) dengan *activation* relu dan *solver* adam.

TABEL 4. 1
Hasil Skenario Pengujian 1

Uji	Default Model Accuracy	Best Model Accuracy	Default Model F1-Score	Best Model F1-Score
1	67.58%	80.64%	70.58%	74.72%
2	56.15%	80.32%	60.30%	71.56%
3	63.96%	80.32%	67.61%	71.56%
4	53.65%	80.45%	57.98%	71.86%
5	67.71%	80.32%	70.64%	71.56%
6	65.90%	80.32%	68.96%	72.25%
7	51.22%	80.32%	55.30%	71.56%
8	64.71%	80.39%	67.56%	72.39%
9	63.90%	80.32%	67.25%	71.56%
10	58.34%	80.32%	62.57%	71.56%
Average Score	**61.31%**	**80.37%**	**64.88%**	**72.06%**

2. Skenario Pengujian 2

Pada skenario pengujian kedua, evaluasi performansi model klasifikasi dilakukan setelah dilakukan *resampling* dengan metode *oversampling* (SMOTE). Sebelum *resampling*, jumlah data *tweet* pada kelas Retweeted sebanyak 1649, sementara pada kelas Not Retweeted sebanyak 6354. Setelah dilakukan *oversampling*, kedua kelas berhasil diimbangi menjadi masing-masing 6354 data *tweet*. Seperti yang ditunjukkan pada gambar 4.1.



GAMBAR 4. 1
Hasil Oversampling

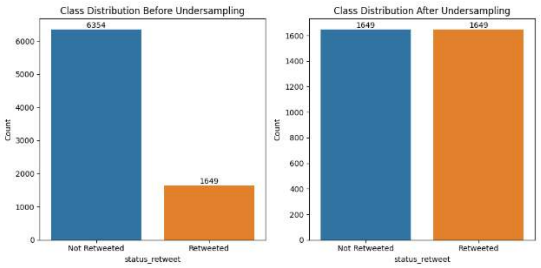
Tabel 4.2 menampilkan hasil perbandingan antara *best model* (setelah optimasi) dan *default model* pada masing-masing pengujian dari 1 hingga 10. Rata-rata akurasi dan F1-Score pada *best model* mengalami peningkatan yang konsisten dari hasil *default model*. Pada skenario 2 terlihat bahwa *default model* memiliki rata-rata akurasi lebih tinggi, yaitu 63.28%, dan F1-Score menjadi lebih rendah, yaitu 60.23%. Setelah dilakukan optimasi, *best model* mengalami peningkatan performansi dari *default model* dengan rata-rata akurasi 66.49% dan F1-Score 65.47%. Penggunaan algoritma CSO dalam optimasi *hyperparameter* berkontribusi pada peningkatan F1-Score, yang menunjukkan bahwa CSO dapat membantu memperoleh kombinasi *hyperparameter* terbaik. Kombinasi parameter terbaik untuk skenario pengujian kedua adalah *hidden layer sizes* (23, 30, 33, 20) dengan *activation* relu dan *solver* adam.

TABEL 4. 2
Hasil Skenario Pengujian 2

Uji	Default Model Accuracy	Best Model Accuracy	Default Model F1-Score	Best Model F1-Score
1	65.30%	68.84%	65.05%	68.56%
2	64.16%	65.34%	61.25%	64.21%
3	59.28%	63.34%	53.21%	58.90%
4	58.34%	62.90%	53.33%	62.32%
5	69.47%	74.39%	67.98%	74.38%
6	64.59%	65.07%	62.15%	64.86%
7	58.38%	69.00%	53.24%	68.71%
8	68.45%	70.30%	68.30%	69.92%
9	64.16%	64.56%	61.36%	64.01%
10	60.62%	61.13%	56.41%	58.84%
Average Score	**63.28%**	**66.49%**	**60.23%**	**65.47%**

3. Skenario Pengujian 3

Pada skenario pengujian ketiga, evaluasi performansi model klasifikasi dilakukan setelah dilakukan *resampling* dengan metode *undersampling* menggunakan NearMiss. Metode *undersampling* ini digunakan untuk mengatasi ketidakseimbangan kelas dengan mengurangi jumlah data pada kelas mayoritas (Not Retweeted) sehingga seimbang dengan jumlah data pada kelas minoritas (Retweeted).



GAMBAR 4. 2
Hasil Undersampling

Dapat dilihat pada Tabel 4.3 menunjukkan perbandingan performansi *default model* dan *best model* pada masing-masing uji dari 1 hingga 10. Pada skenario ini, *default model* memiliki rata-rata akurasi sebesar 78.55% dan F1-Score sebesar 77.86%. Setelah dilakukan optimasi dengan CSO, *best model* mencapai rata-rata akurasi sebesar 86.70% dan F1-Score sebesar 86.61%.

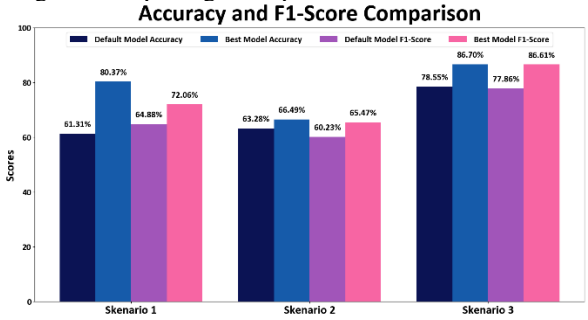
Dapat disimpulkan bahwa pada skenario 3, *resampling* dengan metode *undersampling* menggunakan NearMiss berhasil meningkatkan performansi model klasifikasi. Best model mencapai tingkat akurasi yang lebih baik daripada default model. Kemampuan CSO dalam menyesuaikan *hyperparameter* model membantu mengoptimalkan kinerja model dalam menghadapi masalah ketidakseimbangan kelas. Sehingga, performansi model meningkat secara signifikan. Kombinasi parameter terbaik untuk skenario pengujian ketiga adalah *hidden layer sizes* (16, 19, 26, 10) dengan *activation* relu dan *solver* adam.

TABEL 4. 3
Hasil Skenario Pengujian 3

Uji	Default Model Accuracy	Best Model Accuracy	Default Model F1-Score	Best Model F1-Score
1	78.48%	85.91%	78.21%	85.80%
2	87.27%	87.42%	87.26%	87.40%
3	73.03%	89.85%	71.20%	89.82%
4	78.79%	83.79%	78.49%	83.57%
5	79.24%	87.27%	78.81%	87.25%
6	83.03%	86.82%	82.84%	86.72%
7	85.30%	86.21%	85.28%	86.10%
8	81.67%	86.21%	81.23%	86.15%
9	69.85%	87.73%	68.66%	87.68%
10	68.79%	85.76%	66.58%	85.58%
Average Score	**78.55%**	**86.70%**	**77.86%**	**86.61%**

B. Analisis Hasil Pengujian

Hasil analisis dari ketiga skenario pengujian menunjukkan bahwa penggunaan metode *resampling* mempengaruhi performansi model klasifikasi pada data *tweet* dengan ketidakseimbangan kelas. Pada Skenario 1, *default model* memiliki akurasi rata-rata 61.31% dan F1-Score 64.88%. Setelah dilakukan optimasi dengan CSO, *best model* mencapai akurasi rata-rata terbaik 80.37% dan F1-Score terbaik 72.06%. Pada Skenario 2, *default model* memiliki akurasi rata-rata lebih tinggi, yaitu 63.28%, dan F1-Score lebih rendah, yaitu 60.23%. Setelah optimasi, *best model* meningkatkan performansi dengan akurasi rata-rata 66.49% dan F1-Score 65.47%. Namun, performansi Skenario 2 masih lebih rendah dari Skenario 1. Pada Skenario 3, *default model* memiliki akurasi rata-rata 78.55% dan F1-Score 77.86%. Setelah optimasi, *best model* mencapai akurasi rata-rata 86.70% dan F1-Score 86.61%. Hasil ini menunjukkan bahwa Skenario 3 memiliki performansi paling baik dibandingkan dengan skenario lainnya, karena *best model* mencapai akurasi dan F1-Score yang lebih tinggi. Undersampling dengan NearMiss berhasil meningkatkan performansi model klasifikasi dan memberikan hasil akurasi dan F1-Score tertinggi di antara ketiga skenario. Pemilihan metode *resampling* yang tepat, seperti *undersampling* dengan NearMiss, berdampak signifikan dalam meningkatkan performansi model klasifikasi untuk mengatasi ketidakseimbangan kelas pada data *tweet*. Algoritma CSO berperan dalam memperoleh *hyperparameter* yang menghasilkan nilai *accuracy* dan F1-Score terbaik. CSO membantu mengoptimalkan parameter-parameter model secara adaptif, sehingga memungkinkan model lebih responsif terhadap kelas minoritas, sehingga menghasilkan peningkatan performansi.



GAMBAR 4. 3
Hasil Setiap Skenario Pengujian

V. KESIMPULAN

Pada penelitian ini, dapat disimpulkan bahwa algoritma CSO dapat meningkatkan kinerja model prediksi *retweet* melalui metode klasifikasi ANN dengan fitur pengguna, konten, dan waktu. Algoritma CSO secara efektif mengoptimasi parameter model ANN dan menghasilkan peningkatan yang signifikan dalam akurasi dan F1-Score. Penggunaan metode *resampling* berdampak besar terhadap performa model klasifikasi dalam mengatasi ketidakseimbangan kelas pada data *tweet*. Skenario pengujian terbaik adalah dengan menggunakan metode *undersampling* NearMiss pada Skenario 3, yang menghasilkan akurasi rata-rata 86.70% dan F1-Score 86.61%.

Untuk penelitian mendatang, diharapkan adanya penggunaan algoritma *swarm intelligence* lainnya, seperti Moth Flame Optimizer (MFO) atau Krill Herd Algorithm (KHA), sebagai alternatif untuk mengoptimasi model klasifikasi. Selain itu, penggunaan metode klasifikasi lainnya, seperti Decision Tree, Random Forest, atau Support Vector Machine (SVM), perlu dipertimbangkan untuk menganalisis performansi model secara komprehensif.

REFERENSI

[1] S. N. Firdaus, C. Ding, and A. Sadeghian, "Retweet: A popular information diffusion mechanism – A survey paper," *Online Soc. Networks Media*, vol. 6, pp. 26–40, 2018, doi: 10.1016/j.osnem.2018.04.001.

[2] S. N. Firdaus, C. Ding, and A. Sadeghian, "Retweet prediction considering user's difference as an author and retweeter," *Proc. 2016 IEEE/ACM Int. Conf. Adv. Soc. Networks Anal. Mining, ASONAM 2016*, pp. 852–859, 2016, doi: 10.1109/ASONAM.2016.7752337.

[3] T. B. N. Hoang and J. Mothe, "Predicting information diffusion on Twitter – Analysis of predictive features," *J. Comput. Sci.*, vol. 28, pp. 257–264, 2018, doi: 10.1016/j.jocs.2017.10.010.

[4] Rakes, Jondri, and K. M. Lhaksamana, "Prediksi

retweet berdasarkan feature user-based menggunakan metode klasifikasi Support Vector Machine," vol. 8, no. 5, pp. 11183–11191, 2021.

[5] H. Amarullah Purwaatmaja Ash-Shidiq EFSA and K. Muslim Lhaksmana, "Prediksi Retweet Menggunakan Fitur Berbasis Pengguna dan Fitur Berbasis Konten dengan Metode Klasifikasi ANN," vol. 8, no. 5, pp. 11207–11215, 2021.

[6] S. C. Chu and P. W. Tsai, "Computational intelligence based on the behavior of cats," *Int. J. Innov. Comput. Inf. Control*, vol. 3, no. 1, pp. 163–173, 2007.

[7] S. Chu, P. Tsai, and J. Pan, "PRICAI 2006: Trends in Artificial Intelligence," *PRICAI 2006 Trends Artif. Intell.*, no. March 2014, 2006, doi: 10.1007/11801603.

[8] J. C. Hwang, J. C. Chen, J. S. Pan, and Y. C. Huang, "CSO and PSO to solve optimal contract capacity for high tension customers," *Proc. Int. Conf. Power Electron. Drive Syst.*, no. 245, pp. 246–251, 2009, doi: 10.1109/PEDS.2009.5385703.

[9] S. N. Firdaus, C. Ding, and A. Sadeghian, "Retweet Prediction based on Topic, Emotion and Personality," *Online Soc. Networks Media*, vol. 25, no. August, p. 100165, 2021, doi: 10.1016/j.osnem.2021.100165.

[10] Ghina Khoerunnisa, Jondri, and Widi Astuti, "Prediction of Retweets Based on User, Content, and Time Features Using EUSBoost," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 6, no. 3, pp. 442–447, 2022, doi: 10.29207/resti.v6i3.4125.

[11] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. E. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018, doi: 10.1016/j.heliyon.2018.e00938.

[12] E. Sutoyo and M. A. Fadlurrahman, "Penerapan SMOTE untuk Mengatasi Imbalance Class dalam Klasifikasi Television Advertisement Performance Rating Menggunakan Artificial Neural Network," *J. Edukasi dan Penelit. Inform.*, vol. 6, no. 3, p. 379, 2020, doi: 10.26418/jp.v6i3.42896.

[13] I. P. Dewi, J. Jondri, and K. M. Lhaksmana, "Prediksi Retweet Menggunakan Metode Bernoulli Dan Gaussian Naive Bayes Di Media Sosial Twitter Dengan Topik Vaksinasi Covid-19," *eProceedings Eng.*, vol. 8, no. 5, pp. 11216–11225, 2021, [Online]. Available: https://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15627/15340%0Ahttps://openlibrarypublications.telkomuniversity.ac.id/index.php/engineering/article/view/15627

[14] N. V. Chawla, W. K. Bowyer, L. O. Hall, and P. W. Kegelmeyer, "SMOTE: Synthetic Minority Over-sampling Technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321–357, 2002.

[15] P. Lu and B. Li, "SMOTE," *learn.microsoft.com*, 2021. https://learn.microsoft.com/en-us/azure/machine-learning/component-reference/smote (accessed Jul. 24, 2023).

[16] J. Brownlee, "Undersampling Algorithms for Imbalanced Classification," *machinelearningmastery.com*, 2021. https://machinelearningmastery.com/undersampling-algorithms-for-imbalanced-classification/ (accessed Jul. 24, 2023).