

Analisis *Trade-off* Pendekatan *Greedy* dan *Metaheuristic* dalam Seleksi Fitur terhadap Model *Ensemble*

Anak Agung Gede Ngurah Ananda Wirasena^{a1}, I Wayan Supriana^{a2}, I Made Satria Bimantara^{a3}

^aProgram Studi Informatika, Fakultas Matematika dan Ilmu Pengetahuan Alam,
Universitas Udayana
Jalan Raya Kampus Udayana, Bukit Jimbaran, Kuta Selatan, Badung, Bali, Indonesia

¹wirasena.2308561079@student.unud.ac.id

²wayan.supriana@unud.ac.id

³satriabimantara@unud.ac.id

Abstract

The increasing volume and dimensionality of medical data necessitate robust feature selection techniques (FST) to balance predictive performance and computational efficiency. This study analyzes the trade-off between greedy and metaheuristic FST approaches in optimizing tree-based ensemble models. We compare Mutual Information-Sequential Backward Selection (MI-SBS) as the greedy method and Binary Grey Wolf Optimization (BGWO) as the metaheuristic approach. FST internal fitness is evaluated using a Decision Tree Classifier with 5-fold cross-validation, while final classification performance is assessed using AdaBoost and XGBoost across three medical datasets. Empirical results reveal that modern gradient-boosted ensembles inherently handle high-dimensional noise effectively, often allowing the baseline models to achieve peak raw accuracy. However, MI-SBS provides an effective balance by swiftly filtering variables without degrading accuracy. Conversely, BGWO, despite requiring longer initial selection time, executes an aggressive dimensional reduction that yields a highly parsimonious model. This massive reduction significantly accelerates final model training and inference times at the cost of a minor accuracy drop. This research provides insights into selecting appropriate FST based on desired trade-offs between computational efficiency and classification accuracy in health informatics.

Keywords: *feature selection, metaheuristic, greedy heuristic, ensemble machine learning*

1. Pendahuluan

Dewasa ini, pemanfaatan data medis bersamaan dengan penerapan *Artificial Intelligence* (AI) secara masif telah dikembangkan untuk diagnosis, prediksi, dan/atau perencanaan perawatan pasien [1]. Dalam konteks ini, model *ensemble* berbasis *Decision Tree* (DT) seperti *AdaBoost*, *LightGBM*, *ExtremeGradientBoosting* (XGBoost), dan model serupa lainnya, telah terbukti sangat efektif dalam berbagai tugas klasifikasi, menunjukkan performa yang superior, khususnya di bidang medis [2], [3], [4].

Namun, seiring dengan peningkatan volume data medis yang direkam, kita dihadapkan pada tantangan dimensi data yang semakin tinggi. Menggunakan seluruh fitur yang terekam secara langsung meningkatkan kemungkinan fitur yang *noisy*, redundan, dan tidak relevan yang dapat mengurangi kualitas performa klasifikasi [5]. Dengan menerapkan teknik seleksi fitur/*feature selection technique* (FST) yang reliabel, diharapkan dapat mendukung efisiensi baik dari perspektif akurasi, komputasi, serta penyimpanan, utamanya dalam bidang *Health Informatics* (HI).

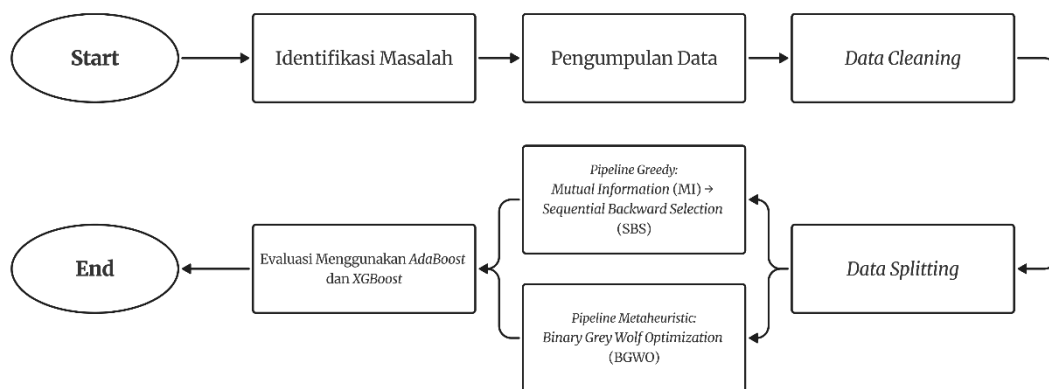
Terdapat tiga jenis utama dalam FST, yaitu Filter, Wrapper, dan Embedded [5]. Beberapa studi terbaru menunjukkan bahwa integrasi teknik seleksi fitur mampu meningkatkan akurasi *ensemble*

learning pada data rekam medis [6], [7]. Meskipun memiliki tujuan yang sama yakni untuk mengidentifikasi fitur yang paling informatif, masih diperlukan analisis lebih lanjut mengenai trade-off antara utilisasi sumber daya sebelum dan/atau selama proses seleksi dengan dampak akhir yang diberikan terhadap performa inferensi model ensemble.

Oleh karena itu, studi ini bertujuan untuk menganalisis performa implementasi FST menggunakan pendekatan *greedy* dan *metaheuristic*. Hasil seleksi fitur dari kedua metode tersebut akan dikomparasikan terhadap *baseline* (*dataset* asli) berdasarkan aspek akurasi dan kecepatan komputasi. Pengujian akhir akan dievaluasi menggunakan model ensemble berbasis DT, seperti AdaBoost dan XGBoost, pada beberapa *dataset* berdomain medis.

2. Metode Penelitian

2.1. Desain Penelitian



Gambar 1. Bagan Alur Penelitian

Penelitian diawali dengan identifikasi permasalahan, kemudian dilanjutkan dengan pengumpulan *dataset* medis dari sumber *open-source* seperti Kaggle. Setelah data terkumpul, tahapan selanjutnya adalah *Data Cleaning* seperti memastikan tipe data tiap kolom, *sampling* data, serta menghilangkan anomali data, untuk memastikan kualitas data. Kemudian diikuti dengan *Data Splitting* untuk memisahkan data pelatihan dan pengujian dan memastikan integritas penelitian. Data yang telah dibagi kemudian akan melalui dua jalur *pipeline* FST yang berbeda yakni, *Pipeline Greedy* yang mengaplikasikan MI dan SBS, serta *Pipeline Metaheuristic* yang menggunakan BGWO. Hasil dari kedua *pipeline* FST ini, beserta *baseline dataset*, akan dievaluasi menggunakan model *ensemble* berbasis *Decision Tree*, yaitu AdaBoost dan XGBoost, untuk menganalisis akurasi dan kecepatan komputasi. Tahap terakhir adalah melakukan analisis terhadap hasil pengujian yang didapat dari seluruh skenario tersebut.

2.2. Pengumpulan dan *Cleaning* Data

Akan digunakan tiga *dataset* medis, yang tersedia secara *open-source* di Kaggle. Dengan keterangan dari setiap *dataset* dapat dilihat pada Tabel 1.

Tabel 1. Karakteristik *Dataset*

Jenis Data	Jumlah Sampel	Jumlah Fitur	Jenis Target	Nilai Kosong
Diabetes	70.692	21	Biner	0
Jantung	299	12	Biner	0
Sinyal Tubuh	991.346	22	Multi Kelas	0

Selanjutnya, *dataset* terpilih akan melalui tahap *cleaning*, untuk menyelaraskan sifat dan bentuk data. Dikarenakan seluruh *dataset* tidak memiliki *missing values*, maka tidak perlu dilakukan imputasi pada aspek tersebut. Sedangkan mengenai jenis target atribut, pada *dataset* Sinyal Tubuh, akan diambil salah satu targetnya saja berupa 'DRK_Y_N' yang mengindikasikan seseorang merupakan peminum atau tidak, dan target atribut lainnya akan dijadikan fitur. Sehingga untuk *dataset* Sinyal Tubuh, akan memiliki 23 fitur.

Selain itu juga pada *dataset* yang sama masih terdapat data non-numerikal yang dapat menghambat tahap pemrosesan sehingga, akan dikonversikan menjadi data nominal. Sedangkan pada *dataset* lainnya seluruh atribut sudah berupa nilai numerikal sehingga tidak perlu ada pemrosesan tambahan. Juga pada *dataset* yang sama akan dilakukan sampling untuk memperkecil ukuran *dataset* menjadi 3% ukurannya, menggunakan *stratified sampling*, untuk menjaga keseimbangan data seperti *dataset* asli.

2.3. Splitting Data

Splitting data atau pembagian data, pada penelitian ini akan dibagi menjadi tiga bagian yakni *train*, *validation*, dan *test*. Hal ini dikarenakan didalam pemilihan fitur, jenis FST yang akan dilakukan condong kepada jenis *Wrapper*, yang menyeleksi fitur setelah mengevaluasi hasil pelatihan model. Sehingga untuk menjaga integritas penelitian, evaluasi akan dilakukan terhadap data *validation* terlebih dahulu. Kemudian pada evaluasi akhir akan menggunakan data *test*.

Tabel 2. Penggalan Kode *Splitting*

No	Potongan Kode
1	<code>seed = np.random.randint(0, 100)</code>
2	<code>X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, stratify=y, random_state=seed)</code>
3	<code>X_train, X_dev, y_train, y_dev= train_test_split(X_train, y_train, test_size=0.2, stratify=y, random_state=seed)</code>

Kode pada Tabel 2, merupakan representasi dari bagaimana kode bekerja. Variabel `random_state` digunakan untuk memastikan kesamaan data ketika melakukan pengujian yang berulang. Sehingga dengan metode ini didapati bahwa ukuran data *train* ialah 64%, data *validation* sebanyak 16%, serta data *test* sebanyak 20% dari data keseluruhan.

2.4. FST Pipeline Greedy

Secara umum, algoritma bertipe *greedy* merupakan algoritma yang membuat keputusan/memilih pada setiap iterasinya, opsi yang paling baik pada saat itu. Meskipun algoritma ini tidak memastikan hasil akhir yang terbaik, tetapi algoritma ini akan unggul pada skenario ketika mencari solusi paling optimal yang eksak tidak diperbolehkan [8]. Menurut Borboudakis [9], pendekatan *Greedy* dalam pemilihan fitur dapat mengikuti *Template for Forward-Backward Selection* (TFBS) yang diajukannya. Secara umum, algoritma ini memiliki dua komponen utama: a) strategi pengurutan variabel, dan b) fungsi *backward phase* untuk menghapus variabel yang kurang relevan setelah pengujian di setiap iterasinya.

Untuk mengadopsi paradigma *greedy* dalam penelitian ini, strategi pengurutan variabel akan menggunakan *Mutual Information* (MI), serta fungsi yang bertanggung jawab untuk menyelksi variabel/attribute menggunakan *Sequential Backward Selection* (SBS).

MI mengukur nilai kepentingan suatu variabel, sedemikian hingga variabel tersebut mengurangi ketidakpastian dari variabel lainnya. Oleh karena itu, MI juga sering digunakan sebagai metode pengukuran dalam teknik *filtering* [10].

Sedangkan SBS akan berperan untuk menghapus satu fitur dari subset fitur yang ada, di setiap iterasinya. Fitur yang dihapus adalah fitur yang, ketika dihilangkan, menghasilkan peningkatan kinerja model berdasarkan metrik evaluasi berupa akurasi. Proses ini berulang hingga sudah mengonsiderasi seluruh atribut secara berurutan dengan algoritma sebagai berikut:

Tabel 3. Penggalan Kode SBS

No	Potongan Kode
1	Inisialisasi F_{ranked} , $F_{removed}$ = Semua fitur dalam X, List kosong untuk fitur yang akan dihapus
2	Definisikan $Model \leftarrow DT_Classifier()$
3	Hitung $Current_Best_Fitness \leftarrow k_cross_val(X, y, Model)$
4	For f in F_{ranked} :
5	$F_{removed}.push(f)$
6	$X_{temp} \leftarrow X.drop(F_{removed})$
7	Definisikan $Model \leftarrow DT_Classifier()$
8	Hitung $Temp_Fitness \leftarrow k_cross_val(X_{temp}, y, Model)$
9	If $Temp_Fitness \geq Current_Best_Fitness$:
10	Perbarui $Current_Best_Fitness \leftarrow Temp_Fitness$
11	Else: $F_{removed}.pop()$
12	EndIf
13	EndFor
14	Kembalikan $F_{removed}$ dan $Current_Best_Fitness$

F_{ranked} merupakan hasil dari mutual information yang telah diurutkan secara *ascending*, sehingga pada SBS akan secara berurutan dikonsiderasi atribut yang paling tidak informatif, jika memiliki *fitness* yang lebih tinggi, maka atribut tersebut akan secara permanen *didrop*.

Hasil *return* dari fungsi ini, berupa $F_{removed}$ merupakan *list* atribut yang akan dihapus atau tidak digunakan berdasarkan konsiderasi dari metode *greedy* tersebut. Sehingga atribut yang terseleksi merupakan seluruh atribut yang dikurangi atribut pada $F_{removed}$.

2.5. FST Pipeline Metaheuristic

Metaheuristic adalah sebuah algoritma optimisasi yang dirancang untuk menemukan solusi yang mendekati optimal (sub-optimal), seringkali terinspirasi oleh fenomena alam ataupun perilaku adaptif. Secara umum *metaheuristic* menyeimbangkan strategi eksplorasi (mencari di area baru) dan eksploitasi (memperbaiki solusi di area menjanjikan) untuk menjelajahi ruang solusi yang lebih luas. Tujuannya adalah untuk meningkatkan peluang menemukan solusi yang lebih baik secara global dalam waktu yang wajar.

Dalam penelitian ini, algoritma *metaheuristic* yang digunakan adalah *Binary Grey Wolf Optimization* (BGWO). BGWO merupakan adaptasi dari algoritma *Grey Wolf Optimization* (GWO) yang terinspirasi oleh perilaku sosial dan strategi berburu serigala. Pemilihan BGWO dibandingkan algoritma *metaheuristic* populer lainnya seperti Genetic Algorithm (GA) atau Particle Swarm Optimization (PSO) didasarkan pada keunggulan BGWO yang memiliki interpretabilitas sekaligus performa yang lebih baik [7]. Perbedaan utamanya diantara BGWO dan GWO terletak pada representasi ruang pencarian dan mekanisme pembaruan posisi serigala. GWO dirancang untuk masalah optimisasi di ruang pencarian kontinu, sedangkan BGWO secara khusus diadaptasi untuk masalah optimisasi di ruang pencarian biner, seperti dalam menyeleksi fitur (1 berarti fitur tidak terpilih, 0 berarti fitur terpilih) [11].

Tabel 4. Penggalan Kode BGWO

No Potongan Kode	
1	Fungsi: <i>BGWO</i> (<i>PopSize</i> , <i>MaxIter</i> , <i>ALLFeatures</i> , <i>X</i> , <i>y</i> , <i>alpha_tradeoff</i>)
2	<i>NumFeatures</i> ← len(<i>ALLFeatures</i>)
3	<i>Populations</i> ← Inisialisasi populasi biner acak (<i>PopSize</i> , <i>NumFeatures</i>)
4	<i>AlphaPos</i> , <i>BetaPos</i> , <i>DeltaPos</i> = Inisialisasi vektor nol
5	<i>AlphaScore</i> , <i>BetaScore</i> , <i>DeltaScore</i> = Inisialisasi -inf
6	<i>ConvergenceCurve</i> ← List kosong
7	For <i>iter_t</i> from 1 to <i>MaxIter</i> :
8	Evaluasi Fitness Populasi:
9	For each <i>wolf_i</i> in <i>Populations</i> :
10	<i>current_fitness</i> ← <i>k_cross_val</i> ()
11	Perbarui <i>Alpha</i> , <i>Beta</i> , <i>Delta</i> berdasarkan <i>current_fitness</i> .
12	End For
13	For each <i>wolf_i</i> in <i>Populations</i> :
14	<i>wolf_i</i> = <i>Binary_Update</i> (<i>AlphaPos</i> , <i>BetaPos</i> , <i>DeltaPos</i> , <i>wolf_i</i>)
15	End For
16	<i>ConvergenceCurve.append</i> (<i>AlphaScore</i>)
17	End For
18	Kembalikan <i>AlphaScore</i> , <i>AlphaPos</i> , <i>ConvergenceCurve</i>

Selanjutnya *AlphaPos* yang dikembalikan merupakan posisi terakhir dari serigala *alpha* atau merupakan serigala terbaik yang merepresentasikan *list* atribut yang dipilih untuk dihapus oleh metode ini, tetapi bentuknya masih dalam *list* biner yang perlu dikonversi menjadi nama atribut, sehingga bentuknya sama seperti metode *greedy* sebelumnya.

2.6. Metode Evaluasi dan Pengujian Akhir

Pada penelitian ini, proses evaluasi dibagi menjadi dua tahap utama: pertama, evaluasi fungsi *fitness* internal selama proses seleksi fitur (MI-SBS dan BGWO); dan kedua, evaluasi kinerja klasifikasi akhir dari dataset yang telah terseleksi fitur menggunakan model *ensemble*. Metode seleksi fitur yang diajukan akan dibandingkan terhadap *baseline* berupa dataset asli tanpa seleksi fitur. Kinerja akan diukur dari aspek akurasi klasifikasi dan efisiensi komputasi.

a. Fungsi *Fitness* untuk Seleksi Fitur

Selama proses seleksi fitur, baik pada MI-SBS maupun BGWO, kinerja dari setiap subset fitur yang diusulkan dievaluasi menggunakan sebuah fungsi *fitness*. Fungsi *fitness* ini bertujuan untuk mengukur seberapa baik subset fitur tersebut dalam memprediksi target. Dalam penelitian ini, fungsi *fitness* dihitung berdasarkan akurasi klasifikasi yang diperoleh dari model Decision Tree Classifier yang dilatih menggunakan subset fitur tersebut. Untuk mendapatkan estimasi kinerja yang stabil dan mengurangi bias, evaluasi juga dilakukan menggunakan *5-fold cross-validation*, di mana nilai akurasi rata-rata dari kelima lipatan akan dijadikan sebagai skor *fitness*.

b. Evaluasi Klasifikasi Akhir

Setelah proses seleksi fitur selesai dan subset fitur optimal untuk masing-masing metode (SBS dan BGWO) serta *baseline* (seluruh fitur) telah didapatkan, tahap selanjutnya

adalah mengevaluasi kinerja klasifikasi akhir. Evaluasi ini menggunakan model *ensemble* yang sering digunakan dalam berbagai tugas klasifikasi. Model yang digunakan dalam penelitian ini adalah:

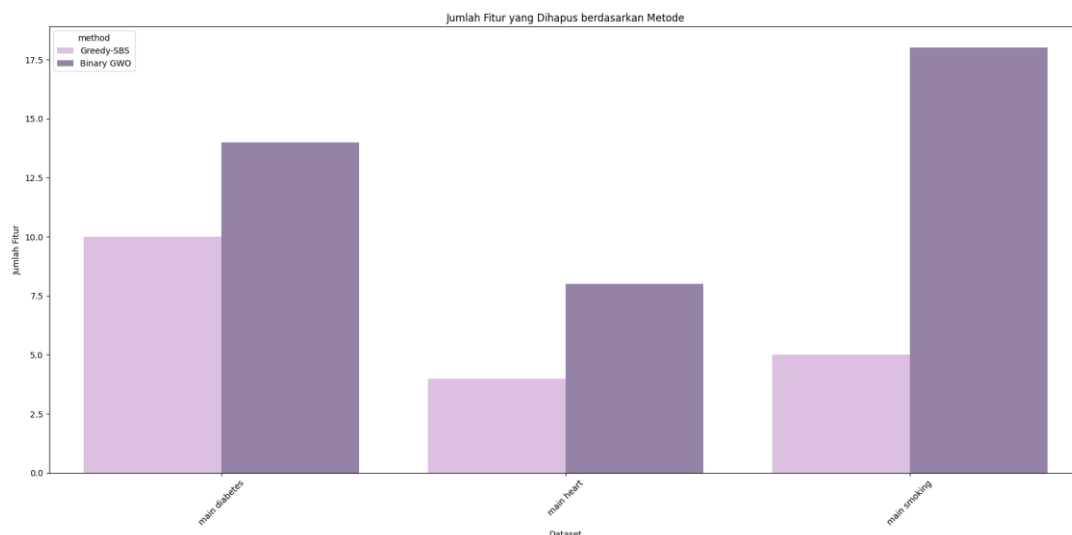
- AdaBoost (Adaptive Boosting) adalah sebuah algoritma *ensemble* yang bekerja secara adaptif untuk meningkatkan performa model. Prosesnya melibatkan pelatihan serangkaian *weak learners* (biasanya Decision Tree dengan *depth* yang dangkal) secara berurutan. Di setiap iterasinya, AdaBoost akan memberikan bobot yang lebih tinggi pada sampel data yang sebelumnya terjadi misklasifikasi, sehingga *weak learners* berikutnya akan lebih fokus pada data dengan prediksi yang salah. Hasil akhir adalah kombinasi bobot dari semua *weak learners* untuk membentuk sebuah *strong learner*.
- XGBoost adalah implementasi dari algoritma *gradient boosting*. Sama seperti AdaBoost, XGBoost membangun model *ensemble* secara aditif, namun ia melatih Decision Tree baru untuk memprediksi residual (kesalahan) dari fungsi *loss* terhadap model *ensemble* yang sudah ada. Perbedaan lainnya yakni optimasi yang lebih baik seperti paralelisasi, *zero value handling*, dan teknik regularisasi (L1 dan L2).

3. Hasil dan Diskusi

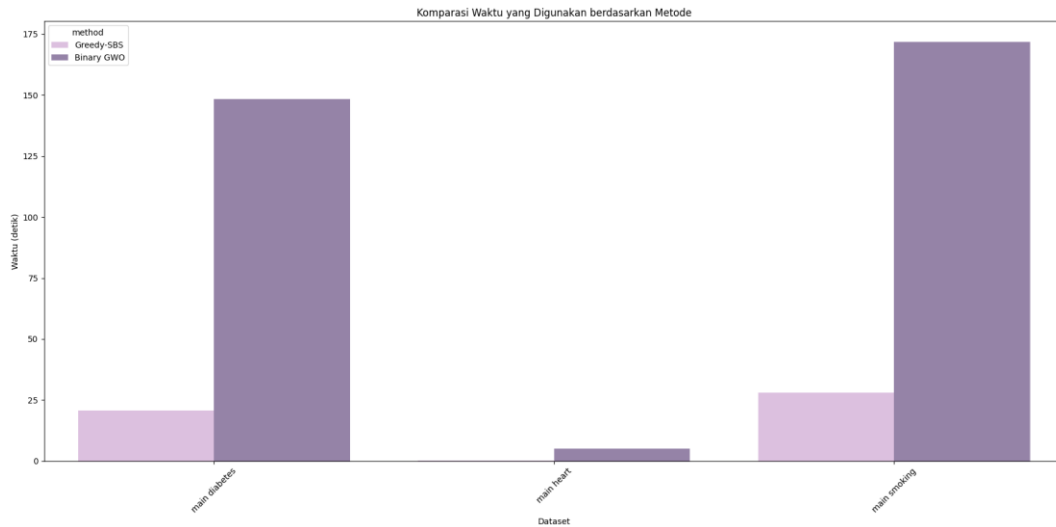
3.1. Hasil FST

Sebelum menguraikan komparasi empiris, perlu diperjelas batasan dan asumsi eksperimen dalam diskusi ini. Pertama, keandalan metode *metaheuristic* bergantung pada karakteristik bawaan dataset, sehingga interpretasi hasil evaluasi pada penelitian ini secara spesifik terbatas pada ruang lingkup dataset yang digunakan. Kedua, subset fitur yang dihasilkan dievaluasi pada model klasifikasi dengan konfigurasi *hyperparameter* bawaan (sebagai kontrol eksperimen), sehingga perilaku akurasi berpotensi berubah apabila diterapkan pada model yang telah melalui proses *tuning* khusus.

Mempertimbangkan batasan tersebut, penelitian ini akan membandingkan performa hasil seleksi fitur dari dua buah metode FST terhadap tiga buah dataset klasifikasi biner. Hasil FST dari masing-masing metode akan disajikan pada **Gambar 1** dan **2**.



Gambar 1. Grafik Komparasi Jumlah Fitur yang Dihapus

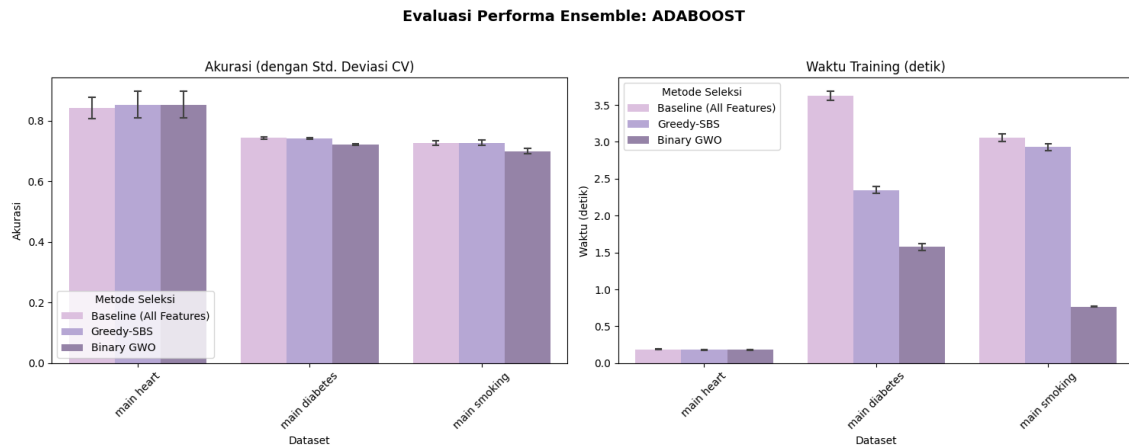


Gambar 2. Grafik Komparasi Waktu yang Digunakan

Tabel 5. Tabel Komparasi Metode

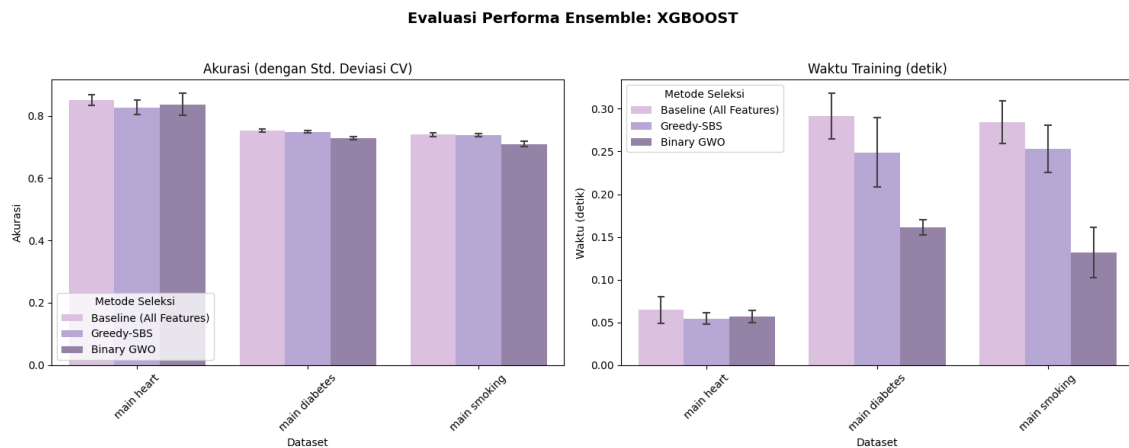
Dataset	Skor Fitting		Atribut Terhapus		Waktu Dibutuhkan	
	MI-SBS	BGWO	MI-SBS	BGWO	MI-SBS	BGWO
Diabetes	0.66	0.72	'DiffWalk', 'PhysHlth', 'PhysActivity', 'HeartDiseas eorAttack', 'Education', 'CholCheck', 'AnyHealthca re', 'Smoker', 'MentHlth', 'Sex'	'HighChol', 'BMI', 'Smoker', 'Stroke', 'HeartDiseaseorAttack', 'PhysActivity', 'Fruits', 'Veggies', 'HvyAlcoholConsump', 'AnyHealthcare', 'MentHlth', 'PhysHlth', 'Education', 'Income'	20.72	148.40
Jantung	0.83	0.84	'serum_creat inine', 'age', 'creatinine_p hosphokinas e', 'platelets'	'age', 'creatinine_phosphokinase', 'diabetes', 'high_blood_pressure', 'platelets', 'serum_creatinine', 'sex', 'smoking'	0.19	5.05
Sinyal Tubuh	0.65	0.69	'sex', 'serum_creat inine', 'sight_left', 'hear_left', 'LDL_chole'	'weight', 'waistline', 'sight_left', 'sight_right', 'hear_left', 'SBP', 'DBP', 'BLDS', 'tot_chole', 'HDL_chole', 'LDL_chole', 'triglyceride', 'hemoglobin', 'urine_protein', 'SGOT_AST', 'SGOT_ALT', 'gamma_GTP', 'SMK_stat_type_cd'	27.96	171.65

3.2. Hasil Pengujian Klasifikasi



Gambar 3. Grafik Evaluasi Akurasi dan Waktu terhadap Model AdaBoost

Selanjutnya, hasil dari masing-masing metode FST dievaluasi menggunakan model *ensemble* (AdaBoost dan XGBoost) dan dikomparasikan dengan *baseline* sebagai kontrol. Kinerja model ditinjau dari dua aspek utama, yakni akurasi klasifikasi, yang dilengkapi dengan *error bar* dari hasil *cross-validation* untuk mengukur stabilitas varians, serta waktu komputasi pelatihan model akhir. Hasil pengujian divisualisasikan pada **Gambar 3** dan **4**



Gambar 4. Grafik Evaluasi Akurasi dan Waktu terhadap Model XGBoost

Berdasarkan visualisasi tersebut, terlihat sebuah pola yang menarik terkait karakteristik model *tree-based ensemble*. Data *baseline* seringkali mampu mencapai tingkat akurasi yang kompetitif. Hal ini wajar terjadi mengingat algoritma seperti XGBoost dan AdaBoost memiliki mekanisme pembobotan dan seleksi fitur implisit (yang merupakan bagian dari arsitektur algoritma tersebut) saat memilih percabangan pada pohon, sehingga model secara mandiri mampu mengabaikan fitur yang kurang relevan dari *dataset* utuh.

Namun, mengevaluasi model klasifikasi tidak hanya sebatas melihat akurasi puncak. Jika ditinjau dari aspek reduksi dimensi, data yang direduksi menggunakan BGWO secara konsisten menampilkan waktu komputasi pelatihan yang lebih rendah. Penurunan waktu ini merupakan hasil langsung dari eliminasi atribut secara masif oleh algoritma BGWO (merujuk pada **Gambar 1**). Meskipun reduksi ekstrem ini berdampak pada penurunan akurasi yang minor, kestabilan varians (terlihat pada *error bar*) tetap terjaga. Hal ini menunjukkan bahwa BGWO sangat unggul

dalam menghasilkan model yang lebih ringan jika ditinjau dari segi komputasinya (penyimpanan dan waktu).

Di sisi lain, data yang telah direduksi menggunakan MI-SBS menunjukkan perspektif yang berbeda. Pendekatan ini memberikan pengurangan waktu komputasi yang moderat dibandingkan *baseline*, namun memiliki keunggulan utama pada retensi informasi. Dari hasil eksperimen, dataset yang diproses dengan MI-SBS secara konsisten mempertahankan akurasi prediksi, dan bahkan pada beberapa skenario berhasil melampaui akurasi *baseline*. Hal ini menjadikannya sebagai metode FST yang aman untuk membuang *noise* tanpa mengurangi keabsahan dan kualitas prediksi model akhir secara agresif.

4. Kesimpulan

Penelitian ini telah menganalisis dan membandingkan performa pendekatan seleksi fitur MI-SBS dan BGWO terhadap model *ensemble* berbasis Decision Tree (AdaBoost dan XGBoost) pada tiga dataset medis.

Dari hasil eksperimen, dapat disimpulkan bahwa kedua pendekatan FST mampu mengurangi dimensi fitur dan mengoptimalkan kinerja model klasifikasi dengan cara yang berbeda. MI-SBS menunjukkan efisiensi komputasi pada tahap seleksi fitur, dengan waktu pemrosesan yang jauh lebih cepat dibandingkan BGWO. MI-SBS terbukti efektif dalam mempertahankan akurasi klasifikasi akhir dibandingkan dengan penggunaan *baseline*, meskipun jumlah fitur yang dihapus relatif lebih sedikit apabila ditinjau dari perspektif efisiensi ruang penyimpanan.

Di sisi lain, BGWO mampu mengeliminasi jumlah fitur yang jauh lebih banyak, yang secara drastis memangkas waktu komputasi pelatihan model akhir. Meskipun pengurangan fitur yang agresif ini menyebabkan sedikit penurunan akurasi klasifikasi, *trade-off* antara efisiensi waktu komputasi dan penurunan akurasi minor ini dapat dipertimbangkan, apabila hendak mengolah data berdimensi besar dan kecepatan inferensi adalah prioritas. BGWO juga secara konsisten mengungguli MI-SBS dalam evaluasi skor *fitness* internal selama proses pencarian.

Secara keseluruhan, kebaruan dari penelitian ini terletak pada pembuktian empiris penyeimbangan antara utilisasi sumber daya dan retensi akurasi dari dua paradigma seleksi fitur yang bertolak belakang, dalam domain informatika kesehatan. Pemilihan metode bergantung pada prioritas sistem, dimana MI-SBS lebih unggul dalam mendahulukan stabilitas akurasi dengan pemrosesan awal yang cepat, sementara BGWO bisa menjadi pilihan dalam menghasilkan model akhir yang ringan.

Untuk penelitian selanjutnya, disarankan untuk mengeksplorasi kombinasi metode FST (misalnya, menggunakan filter MI sebagai pra-pemrosesan untuk membatasi ruang pencarian BGWO), mengevaluasi dampak *hyperparameter tuning* pasca-seleksi fitur, atau menguji algoritma *metaheuristic* generasi terbaru yang mungkin menawarkan keseimbangan *trade-off* yang lebih proporsional.

Daftar Pustaka

- [1] D. Zuo, L. Yang, Y. Jin, H. Qi, Y. Liu, dan L. Ren, "Machine learning-based models for the prediction of breast cancer recurrence risk," *BMC Med. Inform. Decis. Mak.*, vol. 23, no. 1, hlm. 276, Nov 2023, doi: 10.1186/s12911-023-02377-z.
- [2] N. Al Mudawi dan A. Alazeb, "A Model for Predicting Cervical Cancer Using Machine Learning Algorithms," *Sensors*, vol. 22, no. 11, hlm. 4132, Mei 2022, doi: 10.3390/s22114132.
- [3] G. Dharmarathne, T. N. Jayasinghe, M. Bogahawaththa, D. P. P. Meddage, dan U. Rathnayake, "A novel machine learning approach for diagnosing diabetes with a self-explainable interface," *Healthc. Anal.*, vol. 5, hlm. 100301, Jun 2024, doi: 10.1016/j.health.2024.100301.

- [4] T. A. Assegie, R. L. Tulasi, dan N. K. Kumar, "Breast cancer prediction model with decision tree and adaptive boosting," *IAES Int. J. Artif. Intell. IJ-AI*, vol. 10, no. 1, hlm. 184, Mar 2021, doi: 10.11591/ijai.v10.i1.pp184-190.
- [5] P. Usha dan M. P. Anuradha, "Feature Selection Techniques in Learning Algorithms to Predict Truthful Data," *Indian J. Sci. Technol.*, vol. 16, no. 10, hlm. 744–755, Mar 2023, doi: 10.17485/IJST/v16i10.2102.
- [6] M. Shokouhifar, M. Hasanvand, E. Moharamkhani, dan F. Werner, "Ensemble Heuristic–Metaheuristic Feature Fusion Learning for Heart Disease Diagnosis Using Tabular Data," *Algorithms*, vol. 17, no. 1, hlm. 34, Jan 2024, doi: 10.3390/a17010034.
- [7] A. Bilal, A. Alzahrani, A. Almuhaimeed, A. H. Khan, Z. Ahmad, dan H. Long, "Advanced CKD detection through optimized metaheuristic modeling in healthcare informatics," *Sci. Rep.*, vol. 14, no. 1, hlm. 12601, Jun 2024, doi: 10.1038/s41598-024-63292-5.
- [8] A. García, "Greedy algorithms: a review and open problems," *J. Inequalities Appl.*, vol. 2025, no. 1, hlm. 11, Feb 2025, doi: 10.1186/s13660-025-03254-1.
- [9] G. Borboudakis dan I. Tsamardinos, "Extending greedy feature selection algorithms to multiple solutions," *Data Min. Knowl. Discov.*, vol. 35, no. 4, hlm. 1393–1434, Jul 2021, doi: 10.1007/s10618-020-00731-7.
- [10] M. Beraha, A. M. Metelli, M. Papini, A. Tirinzoni, dan M. Restelli, "Feature Selection via Mutual Information: New Theoretical Insights," 17 Juli 2019, *arXiv*: arXiv:1907.07384. doi: 10.48550/arXiv.1907.07384.
- [11] A. K. Dey, G. P. Gupta, dan S. P. Sahu, "A metaheuristic-based ensemble feature selection framework for cyber threat detection in IoT-enabled networks," *Decis. Anal. J.*, vol. 7, hlm. 100206, Jun 2023, doi: 10.1016/j.dajour.2023.100206.