Algoritma Ensemble yang Efisien untuk Meningkatkan *k*-Performa Klasifikasi Tetangga Terdekat melalui Pengemasan Fitur

Huu-Hoa Nguyen
Sekolah Tinggi Teknologi Informasi dan Komunikasi, Universitas Can Tho, Vietnam

Abstrak-Makalah *ini* mengusulkan algoritma ensemble baru yang bertujuan untuk meningkatkan kinerja klasifikasi *k-Nearest* Neighbors (KNN) dengan menggabungkan teknik feature bagging, yang membantu mengatasi keterbatasan yang melekat pada KNN dalam skenario Big Data. Algoritma yang diusulkan, yang disebut FBE (Feature Bagging-based Ensemble), menggunakan strategi ansambel yang efisien dengan teknik subset fitur yang diurutkan untuk mengurangi kompleksitas waktu dari linier ke logaritmik. Dengan berfokus pada fitur-fitur penting selama pelatihan berulang dan memanfaatkan pencarian biner dalam fase pengujian, FBE meningkatkan efisiensi dan akurasi komputasi dalam set data berdimensi tinggi dan tidak seimbang. Penelitian kami secara ketat mengevaluasi algoritma FBE yang diusulkan terhadap algoritma KNN, Random Forest (RF), dan AdaBoost tradisional di sepuluh dataset benchmark dari UCI Machine Learning Repository. Hasil eksperimen menunjukkan bahwa FBE tidak hanya mengungguli KNN konvensional dan AdaBoost di semua metrik yang dievaluasi (akurasi, presisi, recall, dan skor F1), tetapi juga menunjukkan kinerja yang kompetitif dibandingkan dengan RF. Secara khusus, FBE menunjukkan peningkatan yang luar biasa pada dataset yang ditandai dengan dimensi yang tinggi dan ketidakseimbangan kelas. Kontribusi utama dari penelitian ini termasuk pengembangan kerangka kerja KNN adaptif yang menangani tuntutan komputasi yang khas dan kerentanan terhadap noise pada data, sehingga sangat cocok untuk dataset berskala besar. Metodologi ensemble dalam FBE juga membantu mengurangi overfitting, tantangan umum dalam model KNN standar, dengan mendiversifikasi proses pengambilan keputusan di beberapa subset data. Strategi ini memastikan ketahanan dan keandalan, memposisikan FBE sebagai alat yang cocok untuk tugas klasifikasi dalam berbagai domain seperti perawatan kesehatan dan pemrosesan gambar.

Kata Kunci-Pengelompokan; ansambel; fitur; k-tetangga terdekat

I. PENDAHULUAN

Machine learning (ML) secara signifikan meningkatkan kemampuan kita untuk menganalisis kumpulan data yang besar dan mengekstrak wawasan yang dapat ditindaklanjuti di berbagai sektor, termasuk layanan kesehatan dan layanan keuangan. Namun, mengintegrasikan ML dengan Big Data menghadirkan tantangan yang kompleks, seperti mengelola volume dan jenis data yang sangat besar yang dapat melebihi kemampuan metode pemrosesan tradisional. Tantangan-tantangan ini dapat mempersulit pelatihan dan penyempurnaan model ML, yang berdampak pada skalabilitasnya. Selain itu, Big Data dapat memperburuk masalah seperti overfitting, di mana model berkinerja baik pada data pelatihan tetapi buruk pada data baru. Untuk mengatasi kesulitan-kesulitan ini, ada kebutuhan mendesak untuk algoritma canggih yang disesuaikan untuk data berskala besar, serta teknik-teknik untuk pengurangan dimensi yang efektif dan validasi model yang ketat.

Spektrum model pembelajaran mesin sangat bervariasi, masing-masing dirancang untuk memenuhi karakteristik data dan persyaratan analitis tertentu. Model probabilistik, seperti jaringan Bayesian, unggul dalam mengelola ketidakpastian dan variabilitas data, tetapi membutuhkan sumber daya komputasi yang signifikan [1]. Model regresi sangat penting untuk memprediksi variabel kontinu dan memberikan interpretasi yang jelas, meskipun model ini dapat menyederhanakan hubungan yang kompleks [2]. Model arsitektur, seperti jaringan syaraf, unggul dalam pengenalan pola dan mengatasi tantangan non-linear, tetapi mereka membutuhkan data dan sumber daya komputasi yang besar dan sering kali tidak memiliki kejelasan dalam proses pengambilan keputusan [3]. Demikian pula, model berbasis jarak seperti k-Nearest Neighbors efektif dalam tugas klasifikasi yang bergantung pada ukuran kedekatan, namun kesulitan dengan data berdimensi tinggi karena kutukan dimensi [4]. Setiap jenis model menawarkan keuntungan dan keterbatasan yang unik, sehingga memerlukan pemilihan yang cermat untuk menyelaraskan dengan tujuan dan batasan tertentu.

Dalam penelitian ini, kami fokus pada model ML berbasis jarak/kemiripan, khususnya algoritma k-Nearest Neighbors (KNN) [4]. KNN mengklasifikasikan contoh baru berdasarkan kelas yang paling sering muncul di antara tetangga terdekat dalam ruang fitur. Model pembelajaran yang secara inheren bersifat non-parametrik dan malas ini menghafal data pelatihan daripada membangun model yang pasti, memungkinkan kemampuan beradaptasi yang tinggi dan respons langsung terhadap data baru. Terlepas dari kesederhanaan dan keefektifannya, KNN menghadapi beberapa tantangan. Sebagai lazy learner yang menyimpan seluruh dataset, kebutuhan komputasi KNN meningkat seiring dengan ukuran data, sehingga membatasi penggunaannya pada dataset berskala besar. Keakuratan algoritme juga dikompromikan oleh fitur-fitur yang berisik atau tidak relevan yang dapat mendistorsi pengukuran jarak, yang menyebabkan klasifikasi yang tidak akurat. Selain itu, memilih jumlah tetangga (k) yang optimal sangatlah penting; terlalu sedikit dapat menyebabkan overfitting, sementara terlalu banyak dapat menyebabkan underfitting. Selain itu, KNN kesulitan dengan dataset yang menunjukkan ketidakseimbangan kelas yang signifikan, yang berpotensi membiaskan prediksi ke arah kelas mayoritas.

Penelitian kami mengeksplorasi peningkatan pada pendekatan KNN tradisional untuk mengatasi masalah skalabilitas, sehingga mengoptimalkan efisiensinya tanpa mengorbankan akurasi, membuatnya sangat cocok untuk aplikasi Big Data. Secara khusus, makalah ini memperkenalkan algoritma baru yang disebut FBE (Feature Bagging-based Ensemble), yang dirancang untuk meningkatkan kinerja klasifikasi KNN melalui pengantongan fitur. Metode ini secara signifikan mengurangi kompleksitas waktu model tradisional dari linear menjadi logaritmik dengan menyortir subset data selama fase pelatihan dan menggunakan pencarian biner yang efisien dalam fase pengujian, sehingga sangat cocok untuk Big Data

aplikasi. Kami secara ketat mengevaluasi algoritma FBE yang diusulkan pada sepuluh dataset benchmark dari UCI Machine Learning Repository. Hasil eksperimen menunjukkan peningkatan yang signifikan dalam kinerja klasifikasi dibandingkan dengan KNN tradisional dan AdaBoost, dan kompetitif dengan pengklasifikasi Random Forest. Eksperimen komprehensif kami menyoroti potensi FBE dalam menangani dataset yang kompleks, tidak seimbang, atau berdimensi tinggi. Algoritme ini secara eksperimental unggul dalam berbagai metrik, termasuk akurasi, presisi, recall, dan skor F1, menggarisbawahi ketangguhan dan kemampuan beradaptasi. Melalui evaluasi terperinci dan perbandingan dengan model pembelajaran mesin standar, FBE telah membuktikan keefektifan dan keserbagunaannya dalam menangani berbagai macam set data yang menantang.

Bagian selanjutnya dari makalah ini disusun sebagai berikut. Bagian II mengeksplorasi survei literatur dan sintesis. Bagian III merinci algoritma yang diusulkan, menguraikan metodologi dan dasar-dasar teoritisnya, sedangkan Bagian IV didedikasikan untuk validasi eksperimental. Akhirnya, Bagian V menyimpulkan makalah ini dengan ringkasan temuan kami dan penelitian di masa depan.

II. Survei dan Sintesis Literatur

Dalam bidang pembelajaran mesin, khususnya yang berkaitan dengan algoritma k-Nearest Neighbors (KNN), banyak kemajuan yang telah dicapai dalam mengatasi tantangan komputasi yang melekat pada KNN. Bagian ini membahas berbagai metode yang dikembangkan untuk meningkatkan kinerja dan efisiensi KNN.

Di antara berbagai strategi untuk meningkatkan KNN, pengurangan dimensi adalah yang paling berdampak. Hal ini memainkan peran penting dalam meningkatkan efisiensi KNN dengan mengubah data berdimensi tinggi menjadi format yang lebih mudah dikelola tanpa kehilangan informasi yang signifikan. Salah satu pendekatan [5] menggunakan Extreme Learning Machine (ELM) untuk menyederhanakan data yang kompleks menjadi ruang fitur yang lebih mudah diakses. ELM, sebuah metode pembelajaran mesin yang diawasi dengan satu lapisan tersembunyi, terkenal karena kemampuannya memproses data dengan cepat. Namun, metode ini juga sensitif terhadap noise dan sangat bergantung pada pemilihan bobot dan bias secara acak, yang dapat membatasi keefektifannya. Metode lain [6] menggunakan Mutual Information (MI) untuk meningkatkan efisiensi pengurangan dimensi dan menggunakan General Purpose Graphics Processing Units untuk memparalelkan proses pencarian tetangga terdekat. Meskipun efektif, metode ini membutuhkan sumber daya perangkat keras tambahan, yang mungkin tidak praktis dalam semua pengaturan.

Untuk mengurangi tantangan konsumsi sumber daya yang terkait dengan kNN, banyak peneliti telah mengeksplorasi solusi berbasis pohon sebagai strategi umum. Metode-metode ini biasanya melibatkan pemilihan kriteria pemisahan untuk membangun sebuah pohon, sering kali berupa pohon biner, yang mengorganisir dataset dengan cara yang mempercepat pencarian tetangga terdekat. Beberapa model berbasis pohon yang inovatif telah dikembangkan, seperti Combi Tree, yang menawarkan pendekatan adaptif untuk mengoptimalkan KNN.

Combi Tree, yang dikembangkan dari binary search tree [7], memilah-milah titik data menjadi beberapa kluster dan menggunakan tabel hash untuk mengompres setiap kluster. Klaster-klaster ini kemudian digabungkan untuk membentuk Pohon Kombinasi. Akan tetapi, pendekatan ini beroperasi secara eksklusif

dalam ruang Hamming, sebuah ruang berdimensi tinggi yang cocok untuk data biner, sehingga kurang efektif untuk jenis data lain atau ukuran kemiripan di luar ruang ini. Strategi lain membangun Binary Search Tree (BST) berdasarkan norma-norma titik data [8]. Ini melibatkan skema partisi yang menggunakan norma-norma untuk mendistribusikan titik-titik data secara merata di dalam BST, meskipun metode ini mungkin mengalami kesulitan dengan kemencengan dalam distribusi data.

Selain itu, metode BST yang baru [9] menggabungkan faktor penskalaan untuk meningkatkan kecepatan pencarian, terutama bermanfaat untuk mengelola kumpulan data yang besar di mana pohon pencarian biner tradisional mungkin tidak praktis dan tidak efisien. Metode ini menyesuaikan ukuran interval pencarian menggunakan faktor penskalaan saat pencarian berlangsung, sehingga memungkinkan penyempitan ruang pencarian dengan cepat dan mencapai kompleksitas waktu logaritmik. Namun, validasi metode ini masih terbatas pada data sintetis, dan efektivitasnya dalam skenario dunia nyata masih belum dikonfirmasi.

Sementara metode berbasis pohon berfokus pada optimasi struktural, pendekatan lain melibatkan penyempurnaan data itu sendiri melalui pengelompokan. Salah satu metode [10] menggunakan algoritma pengelompokan k-means untuk mengelompokkan dataset, menghapus titik data yang memiliki dampak minimal pada akurasi. Selama tahap pengujian, metode ini menentukan cluster yang menjadi milik sebuah instance dan melakukan KNN dalam subset tertentu. Namun, teknik pemangkasan ini mungkin tidak mencapai hasil yang optimal dalam skenario di mana batas keputusannya tidak linier atau dataset secara inheren berisik.

Untuk lebih menyempurnakan strategi ini, penelitian lain [11] menggunakan pengelompokan untuk mengurangi jumlah titik data yang diperlukan untuk setiap kueri. Metode ini memperkenalkan teknik pembagian wilayah untuk lebih membatasi ruang pencarian. Metode ini membagi ruang menjadi beberapa wilayah yang lebih kecil dan hanya mempertimbangkan titik-titik data di dalam wilayah yang berisi titik kueri untuk pencarian KNN. Namun demikian, pengelompokan dapat menjadi berat secara komputasi dengan data berdimensi tinggi, sehingga membatasi skalabilitas metode ini seiring dengan bertambahnya jumlah fitur.

Membangun ide pengelompokan, metode Pohon KNN [12] menggabungkan strategi berbasis pohon dan pengelompokan untuk lebih meningkatkan efisiensi. Algoritma ini membangun pohon keputusan (DT) hingga kedalaman tertentu dan kemudian menerapkan KNN pada subset yang tersisa dari kumpulan data. Algoritma hibrida ini secara efektif mengurangi jumlah sampel yang diperlukan untuk KNN, sehingga meningkatkan efisiensi model. Secara khusus, metode ini melampaui kinerja DT mandiri dan KNN tradisional, menunjukkan peningkatan yang signifikan dalam mengelola dataset berskala besar.

Untuk mengontekstualisasikan kemajuan ini, Tabel I membandingkan berbagai pendekatan ini, menggarisbawahi pertukaran dan peningkatan yang diperkenalkan oleh algoritme kami. Perbandingan ini menjelaskan pertukaran yang melibatkan kecepatan, skalabilitas, sensitivitas noise, dan kekhususan data di antara berbagai metode. Meskipun setiap metode secara signifikan meningkatkan efisiensi KNN, metode-metode tersebut juga memiliki keterbatasan spesifik yang dapat memengaruhi kegunaannya, tergantung pada skenario aplikasi.

TABEL. I. ANALISIS PERBANDINGAN BERBAGAI METODE TERKAIT

Metode	Keuntungan	Kelemahan
Berbasis ELM [5]	Meningkatkan kecepatan pemrosesan dengan menyederhanakan representasi data.	Sensitif terhadap noise, yang memengaruhi ketahanan klasifikasi.
Pknn-mifs [6]	Mempercepat proses KNN melalui pemrosesan paralel.	Membutuhkan sumber daya perangkat keras tambahan, sehingga meningkatkan biaya.
Pohon kombi [7]	Mencapai kompleksitas logaritmik, cocok untuk data biner.	Dibatasi untuk aplikasi dalam ruang Hamming saja.
Berbasis norma [8]	Efektif di lingkungan dengan distribusi data yang seragam.	Efektivitas yang terbatas pada kumpulan data yang tidak seragam atau miring.
Berbasis BST [9]	Mencapai kompleksitas logaritmik, mengoptimalkan waktu pencarian.	Performa terutama divalidasi pada data sintetis, bukan data dunia nyata.
EDP [10]	Meningkatkan kecepatan dengan memangkas data yang tidak perlu secara efisien.	Performa menurun dengan skenario data non-linear.
SRBC [11]	Memberikan eksekusi yang lebih cepat dibandingkan dengan KNN tradisional.	Tidak dapat menskalakan dengan baik dengan data berdimensi tinggi.
KNNTree [12]	Mencapai kompleksitas waktu logaritmik, meningkatkan efisiensi.	Performa sangat bergantung pada penyetelan hiper-parameter yang tepat.

III. ALGORITMA YANG DIUSULKAN (FBE)

A. Gagasan Umum tentang FBE

Ide umum di balik algoritma FBE yang diusulkan adalah untuk mengubah KNN yang mahal secara komputasi menjadi model yang lebih efisien dan dapat diskalakan dengan menggunakan kekuatan pemilahan berbasis pendekatan dan pembelajaran ensemble. Di bawah ini, kami mengeksplorasi konsep dasar yang mendukung FBE.

1) Pemilihan dimensi dan penyortiran data: FBE dimulai dengan pemilihan strategis dan penggunaan dimensi data. Sebagai contoh, pertimbangkan dataset yang direpresentasikan dalam ruang tiga dimensi, seperti yang diilustrasikan pada Gbr. 1. Dalam kerangka kerja FBE, dimensi dipilih secara acak, seperti dimensi D1 dan D3 untuk iterasi tertentu. Pemilihan dimensi sangat penting karena mempengaruhi pengurutan dataset selanjutnya. Jika D3 menunjukkan korelasi yang lebih kuat dengan label kelas daripada D1, maka D3 menjadi sumbu utama untuk pengurutan. Proses penyortiran ini, diilustrasikan pada Gbr. 2, sangat penting karena proses ini mengatur ulang dataset untuk menyelaraskan lebih dekat dengan struktur data yang melekat, sehingga memungkinkan pencarian yang lebih efisien selama fase pengujian.

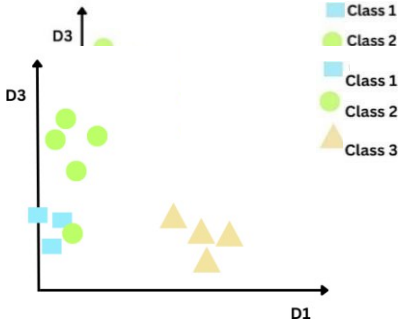
2) Teknik pencarian dan ansambel berbasis aproksimasi: FBE menggunakan teknik pencarian berbasis perkiraan untuk mengurangi kompleksitas waktu yang secara tradisional terkait dengan KNN, biasanya di mana *n* adalah jumlah contoh dan *d* adalah dimensi data. Dengan mengurutkan data sesuai dengan dimensi yang dipilih yang menunjukkan hubungan yang konsisten dengan variabel target, algoritme ini membuka jalan untuk pencarian biner. Metode pencarian ini secara signifikan mengurangi ruang pencarian dari kompleksitas waktu linear ke logaritmik, sehingga memungkinkan untuk menangani kumpulan data yang besar secara efektif.

Integrasi teknik ensemble semakin meningkatkan algoritma FBE. Dengan mengulangi proses pemilihan dan penyortiran beberapa kali, masing-masing dengan dimensi yang berpotensi berbeda, algoritme ini menciptakan serangkaian tampilan data yang beragam, masing-masing disusun berdasarkan fitur yang paling informatif dari iterasi tersebut. Ansambel himpunan bagian yang diurutkan ini tidak hanya mengurangi risiko bias dalam model, tetapi juga meningkatkan akurasi secara keseluruhan melalui pengambilan keputusan kolektif selama fase pengujian.

3) Manfaat sinergis: Sinergi antara pemilihan subset terurut dan strategi ensemble menghasilkan algoritme yang kuat yang tidak hanya meningkatkan efisiensi komputasi, tetapi juga mempertahankan, bahkan meningkatkan efektivitas klasifikasi. Hasil

Pendekatan ensemble mengurangi varians dan potensi overfitting dengan mengintegrasikan beberapa evaluasi independen dari tetangga terdekat, masing-masing dari perspektif data yang sedikit berbeda. Hasilnya, FBE menyajikan alternatif yang menarik untuk KNN konvensional, terutama dalam skenario yang melibatkan dataset berskala besar dengan hubungan yang kompleks dan tidak linier antar fitur.

Gbr. 1. Titik data dalam tiga dimensi



Gbr. 2. Titik data dalam dua dimensi yang dipilih secara acak

B. FBE dalam Tahap Pelatihan

Fase pelatihan FBE secara metodis diuraikan dalam Algoritma 1 dan terdiri dari tiga langkah utama, sebagai berikut.

1) Langkah 1: Inisialisasi: Algoritma dimulai dengan menginisialisasi himpunan kosong S_F yang pada akhirnya akan menyimpan subset dari data pelatihan di samping pengurutan yang sesuai
fitur. Set ini memainkan peran penting dalam strategi ensemble, memfasilitasi beragam set data yang disederhanakan untuk pencarian tetangga yang efisien selama fase pengujian.

2) Langkah 2: Pemrosesan berulang: Inti dari Algoritma 1 beroperasi melalui beberapa iterasi, yang mencerminkan sifat ansambel FBE. Setiap iterasi dirancang untuk membuat subset data yang unik, dengan fokus pada fitur yang berbeda untuk menangkap berbagai karakteristik data:

- Pemilihan Fitur: Dalam setiap iterasi, subset fitur, X', dipilih secara acak dari kumpulan fitur X. Keacakan ini memperkenalkan keragaman dalam fitur yang dipertimbangkan di seluruh iterasi yang berbeda, yang merupakan hal mendasar dalam pendekatan ensemble.
- Penentuan Fitur Terbaik: Untuk setiap fitur yang dipilih a, informasi timbal baliknya dengan label target y dihitung. Informasi timbal balik, dilambangkan sebagai MI (a, y), mengukur jumlah informasi yang dikandung oleh satu variabel tentang variabel lainnya, sehingga membantu mengidentifikasi fitur yang paling fitur prediktif. Secara matematis, MI didefinisikan sebagai:

$$MI(X,Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right),$$

di mana $p(x,y)$ adalah distribusi probabilitas gabungan dari X dan Y, dan $p(x)$ dan $p(y)$ masing-masing adalah distribusi marjinal dari X dan Y.

- Fitur *a* dengan nilai *MI* tertinggi dipilih sebagai fitur *b* terbaik. Fitur ini dianggap paling efektif dalam mengklasifikasikan titik data untuk iterasi tersebut, memandu proses penyortiran.

3) Langkah 3: Menyortir dan menyimpan

- Setelah fitur terbaik *b* diidentifikasi, subset dari data X_t, sesuai dengan X', bersama dengan label y_t, diurutkan berdasarkan *b*. Pengurutan ini sangat penting karena mengatur ulang titik-titik data sehingga nilai-nilai yang mirip (dan dengan demikian berpotensi memiliki kelas yang sama) diposisikan lebih dekat satu sama lain, yang secara drastis meningkatkan efisiensi pencarian tetangga dalam ruang dimensi tinggi.

Subset yang telah diurutkan bersama dengan metadata (fitur yang digunakan dan fitur terbaik) dienkapsulasi ke dalam sebuah tuple $val = (X_t, y_t, X', b)$ dan ditambahkan S_F. Setiap tuple dalam S_F

mewakili "pandangan" atau "model" yang berbeda dari dataset, yang dioptimalkan untuk pencarian cepat dalam kerangka kerja metode ensembel yang diusulkan.

Pada akhir iterasi, S_F berisi beberapa pengurutan versi subset dari set pelatihan, masing-masing dioptimalkan secara berbeda berdasarkan fitur yang dipilih. Pengaturan ini memungkinkan algoritme FBE selama fase pengujian untuk dengan cepat menemukan tetangga terdekat dengan memanfaatkan struktur data yang telah diproses sebelumnya dan diurutkan secara efisien, sehingga sangat mengurangi overhead komputasi dan kompleksitas waktu dibandingkan dengan pendekatan KNN tradisional.

Pada intinya, Algoritma 1 meletakkan dasar untuk FBE, memastikan bahwa metode ensemble tidak hanya mempertahankan kinerja klasifikasi yang tinggi tetapi juga mengatasi masalah skalabilitas yang sering dikaitkan dengan KNN, terutama dalam set data yang besar.

Algoritma 1: FBE dalam fase pelatihan

Input:

- $X = \{x_1, x_2, ..., x_n\}$: Kumpulan vektor fitur
- $y = \{y_1, y_2, ..., y_n\}$: Label yang sesuai
- *k*: Jumlah tetangga terdekat
- *m*: Jumlah iterasi untuk ansambel
- *g:* Parameter anugerah

Keluaran:

- S_F : Kumpulan fitur yang diurutkan dan metadata terkait

Prosedur:

1. Inisialisasi S_F set fitur yang diurutkan ke set kosong.

2. Untuk setiap iterasi *i* dari 1 sampai *m*:

- Pilih subset fitur secara X' acak dari *X*.
- Inisialisasi fitur terbaik *b* menjadi tidak ada dan informasi timbal balik maksimum *Max_MI* menjadi -1.
- Untuk setiap fitur *a* dalam X' :
  - Hitunglah informasi timbal balik *MI* antara *a* dan *y*.
  - Jika *MI* lebih besar dari *Max_MI*:
    - Perbarui *Max_MI* dengan *MI*.
    - Tetapkan fitur terbaik *b* ke *a*.
- Pilih himpunan bagian dari *X* yang sesuai dengan X' dan X_t $y_t = y$ sebagai dan himpunan .
- Mengurutkan X_t dan y_t berdasarkan nilai fitur terbaik *b*.
- Buat $val = (X_t, y_t, X', b)$ tuple dan tambahkan ke S_F

3. Kembali

C. FBE dalam Tahap Pengujian

Fase pengujian FBE secara metodis dijelaskan dalam Algoritma 2. Representasi Algoritma 2 ini selaras dengan Algoritma 1 dengan secara langsung memanfaatkan himpunan bagian terurut yang dihasilkan selama fase pelatihan, memastikan bahwa metode ensembel secara efisien menggunakan data yang telah diproses sebelumnya untuk meningkatkan akurasi prediksi dan efisiensi komputasi. Fase pengujian ini terdiri dari empat tugas utama, sebagai berikut.

Algoritma 2: FBE dalam tahap pengujian

Input:

- X_i : Titik data pengujian yang diberikan.

Keluaran:

- C_i : Label kelas yang diprediksi untuk titik data pengujian X_i.

Prosedur:

1. Inisialisasi daftar kosong prediksi *P* untuk menyimpan label prediksi dari setiap iterasi.

2. Untuk setiap iterasi *i* dari 1 hingga *m* (seperti yang ditetapkan dalam fase pelatihan):

- Tetapkan batas pencarian awal *rendah* = 1 dan *tinggi* = *n*, di mana *n* adalah jumlah total titik data dalam subset data.
- Ekstrak subset terurut dari fitur X_t danlabel yang sesuai y_i dariset fitur terurut S_F yang disiapkan pada fase pelatihan
  - $X_t = S_F[i]["features"]$
  - $y_t = S_F[i]["labels"]$
  - $X' = S_F[i][b]$
  -
- Tentukan indeks X_t yang paling X_i cocok X' dengan menggunakan pencarian biner:
  - Sementara *rendah* < *tinggi:*
    - Hitung *pertengahan* = *rendah* + (*tinggi* - *rendah*)/2.
    - $X_t[X'][mid] < X_i[X']$ Jika maka tetapkan *rendah* = *pertengahan* + 1, jika tidak tetapkan *tinggi* = *pertengahan* - 1.
- Setelah menemukan wilayah terdekat, tentukan interval pencarian dalam data yang diurutkan:
  - *kiri* = maks (0, *rendah* - *k* - *g*)
  - *kanan* = min (*n*, *rendah* + *k* + *g*) $X_t[left : right]$ $y_t[left : right]$
- Gunakan KNN tradisional untuk memprediksi label kelas dari subset dan dan tambahkan hasilnya ke prediksi *P*. C_i

3. Setelah semua iterasi, tentukan label kelas mayoritas dari *P* dan kembalikan sebagai .

Legenda:

- *k*: Jumlah tetangga terdekat (ditentukan dalam fase pelatihan).
- *g*: Parameter Grace (ditentukan dalam fase pelatihan).
- S_F : Set fitur yang diurutkan, berisi tupel dari subset fitur yang diurutkan dan label yang sesuai dari fase pelatihan.
- *m*: Jumlah iterasi ensemble, selaras dengan jumlah himpunan bagian yang diurutkan dalam S_F.

1) Inisialisasi dan pengumpulan prediksi

- Algoritme dimulai dengan menginisialisasi daftar prediksi kosong, *P*, yang dirancang untuk mengumpulkan hasil dari setiap iterasi. Hal ini memfasilitasi pendekatan ensembel di mana beberapa prediksi dikumpulkan untuk menentukan kelas yang paling mungkin untuk contoh pengujian yang diberikan, X_i.
- Metode ensembel yang digunakan di sini memastikan ketahanan dalam prediksi dengan merata-rata bias yang mungkin ada di setiap subset yang diurutkan dari data pelatihan.

2) Pencarian biner berulang pada himpunan bagian yang diurutkan

- Untuk setiap iterasi dalam jumlah iterasi ensembel *m* yang telah ditentukan sebelumnya, algoritme

memproses subset data yang telah diurutkan dan disimpan selama fase pelatihan. Himpunan bagian ini diindeks dari kumpulan

set terstruktur S_F , yang berisi informasi kunci seperti subset fitur, label yang sesuai, dan fitur yang menunjukkan informasi timbal balik tertinggi dengan hasil selama fase pelatihan.

- Inti dari tahap pengujian adalah pencarian biner pada subset yang dipilih menggunakan fitur terbaik *b* yang diidentifikasi selama pelatihan. Hal ini memandu pencarian untuk menemukan segmen dataset di mana contoh pengujian mungkin berada, berdasarkan kemiripan fitur.
- Algoritma pencarian biner menyesuaikan penunjuk *rendah* dan *tinggi* berdasarkan perbandingan antara X_i $[X']$ dan nilai titik tengah X_t $[X']$. Metode ini secara drastis mengurangi jumlah perbandingan yang diperlukan untuk menemukan wilayah terdekat dalam kumpulan data dari mana tetangga dipilih.

3) Penentuan tetangga terdekat setempat

- Setelah perkiraan lokasi X_i ditentukan dengan tepat di

larik terurut, lingkungan lokal didefinisikan di sekitar ini terbaik
titik. Ukuran lingkungan ini disesuaikan dengan parameter *k* dan *g*, di mana *k* menunjukkan jumlah tetangga terdekat yang biasanya dipertimbangkan dalam KNN, dan *g* memungkinkan untuk penyangga pencarian yang diperluas untuk mengurangi risiko kehilangan potensi tetangga terdekat karena efek batas atau wilayah yang jarang dalam dataset.

- KNN tradisional kemudian diterapkan dalam segmen dataset yang terlokalisasi ini untuk memprediksi kelas berdasarkan suara mayoritas di antara k-tetangga *terdekat* yang ditemukan di wilayah ini. Hal ini memastikan prinsip dasar KNN dalam mengklasifikasikan berdasarkan titik data terdekat tetap dipertahankan, bahkan dalam metode ensemble.

4) Agregasi dan prediksi akhir

- Setelah melalui semua iterasi, prediksi dari setiap subset digabungkan untuk menentukan label kelas akhir untuk X_i . Metode agregasi biasanya melibatkan pemilihan kelas mayoritas dari daftar prediksi, menggunakan keragaman ansambel untuk memberikan prediksi yang lebih akurat dan stabil.
- Sistem pemungutan suara mayoritas di berbagai model prediktif ini mengurangi varians dan meningkatkan keandalan klasifikasi, terutama dalam kasus-kasus di mana model individu mungkin memiliki bias atau berkinerja buruk dalam kondisi data tertentu.

Pada intinya, Algoritma 2 meningkatkan pendekatan KNN tradisional dengan mengintegrasikan pencarian biner yang efisien dalam subset yang telah diproses secara strategis dan menggunakan metodologi ensemble untuk mendapatkan prediksi yang kuat dan akurat. Pendekatan ini tidak hanya mempercepat proses klasifikasi secara signifikan dengan mengurangi jumlah perhitungan jarak yang biasanya diperlukan dalam KNN, tetapi juga memanfaatkan keragaman beberapa model untuk meningkatkan akurasi prediksi secara keseluruhan. Kombinasi dari strategi ini membuat FBE sangat cocok untuk dataset berskala besar di mana KNN tradisional mungkin mengalami kesulitan dalam hal skalabilitas dan kinerja.

D. Analisis Kompleksitas Algoritma FBE

Memahami kompleksitas komputasi FBE sangat penting untuk mengevaluasi efisiensinya, terutama jika dibandingkan dengan metode berbasis KNN tradisional. Bagian ini membahas kompleksitas fase pelatihan dan pengujian FBE, menyoroti peningkatan yang dilakukan dengan menggabungkan himpunan bagian yang diurutkan dan teknik ensemble.

1) Analisis kompleksitas FBE pada fase pelatihan: Fase pelatihan FBE melibatkan pemilihan subset fitur, perhitungan mutual information (*MI*) untuk mengidentifikasi fitur yang paling informatif, dan mengurutkan subset ini untuk pencarian yang efisien selama pengujian. Misalkan *n* adalah jumlah instance, *d* adalah jumlah dimensi, *k* adalah jumlah tetangga terdekat, *m* adalah jumlah iterasi, dan *g* adalah parameter grace.

- Pemilihan fitur dan perhitungan informasi timbal balik:

Pada setiap iterasi, subset fitur dipilih secara acak, yang memperkenalkan variabilitas tetapi juga memerlukan penilaian ulang struktur data untuk setiap subset. Saling perhitungan informasi, yang membantu dalam memilih fitur

untuk pengurutan, biasanya memiliki kompleksitas $O(n)$ per fitur. Dengan mempertimbangkan bahwa setiap atau semua fitur dapat terlibat dalam kasus terburuk, kompleksitas untuk langkah ini adalah $O(n.d)$.

- Penyortiran:

Setelah mengidentifikasi fitur terbaik, subset diurutkan berdasarkan fitur ini. Mengurutkan daftar n elemen umumnya menghabiskan waktu $O(n \log n)$. Karena hal ini dilakukan untuk setiap dimensi
subset yang dikurangi (secara efektif setiap fitur dalam kasus terburuk), kompleksitasnya menjadi $O(nd \log n)$.

- Kompleksitas keseluruhan dalam fase pelatihan:

Dengan menggabungkan faktor-faktor ini, kompleksitas untuk setiap iterasi adalah $O(nd + nd \log n)$, yang disederhanakan menjadi $O(nd \log n)$. Di seluruh m iterasi, hal ini menghasilkan kompleksitas total sebesar $O(mnd \log n)$
,
mewakili kebutuhan komputasi yang signifikan tetapi masih lebih mudah dikelola daripada perbandingan berpasangan yang lengkap di semua fitur dan contoh.

2) Analisis kompleksitas FBE dalam fase pengujian: Fase pengujian menggunakan himpunan bagian yang sudah diurutkan sebelumnya dan mekanisme pencarian biner untuk menemukan tetangga terdekat yang potensial dengan cepat, sehingga secara signifikan mengurangi waktu yang diperlukan untuk setiap kueri.

- Pencarian biner:

Pencarian biner pada subset yang diurutkan memiliki kompleksitas
$O(\log n)$, yang tidak bergantung pada dimensi d karena pencarian terbatas pada dimensi yang diurutkan yang diidentifikasi selama fase pelatihan.

- Komputasi KNN lokal:

Setelah perkiraan lokasi contoh uji ditentukan, pencarian KNN lokal dilakukan dalam segmen yang ditentukan oleh k dan g. Biaya komputasi untuk pencarian lokal ini bergantung pada ukuran segmen tetapi tetap lebih kecil daripada pencarian seluruh dataset. Kerumitan untuk hal ini

Langkah ini diperkirakan sebagai $O(kd + dg + k^2 + kg)$, yang disederhanakan menjadi $O(k^2)$ dalam skenario praktis di mana k jauh lebih kecil dari n.

- Keseluruhan kompleksitas dalam fase pengujian:

Kompleksitas gabungan dari fase pengujian untuk m iterasi adalah $O(m(\log n + k^2))$. Hal ini menyoroti hal yang substansial efisiensi dibandingkan metode yang memerlukan pemindaian dataset lengkap.

3) Analisis kompleksitas ruang dari FBE: Kompleksitas ruang dari FBE terutama ditentukan oleh penyimpanan yang diperlukan untuk subset yang diurutkan dan metadata terkait. Untuk m iterasi, menyimpan setiap subset dan fitur-fiturnya membutuhkan ruang $O(mnd)$, sedikit lebih tinggi daripada KNN tradisional tetapi dibenarkan oleh peningkatan yang signifikan dalam kinerja waktu kueri.

Singkatnya, algoritma FBE menyajikan sebuah pendekatan untuk KNN, dengan kompleksitas waktu $O(mnd \log n)$ untuk pelatihan dan $O(m(\log n + k^2))$ untuk pengujian. Peningkatan ini membuat FBE sangat cocok untuk dataset berskala besar di mana keseimbangan antara akurasi, kecepatan komputasi, dan Pemanfaatan sumber daya sangat penting. Algoritme ini secara efektif memanfaatkan kekuatan metode ensembel dan struktur data yang diurutkan untuk meningkatkan skalabilitas pencarian tetangga terdekat.

IV. Validasi Eksperimental

A. Deskripsi Dataset

Untuk mengevaluasi efektivitas FBE, sepuluh set data benchmark yang dipilih secara komprehensif, yang menekankan keragaman dan kompleksitas yang melekat pada data dunia nyata. Dataset ini, yang sebagian besar bersumber dari UCI Machine Learning Repository, sangat cocok untuk menunjukkan kemampuan FBE yang kuat karena tantangan dan karakteristiknya yang beragam.

Di antara set data yang dipilih, lima di antaranya berpusat pada aplikasi medis, sebuah domain di mana kompleksitas data dan kebutuhan akan ketepatan dan keandalan sangat penting. Dataset ini meliputi EKG, Diabetes, Limfografi, Kesuburan, dan Kanker Payudara, yang masing-masing menghadirkan tantangan unik karena ketidakseimbangan dan sifat hasil yang ingin diprediksi. Sebagai contoh, dataset EKG, dengan rangkaian fiturnya yang luas, menguji kemampuan algoritme untuk menangani data berskala besar dalam situasi di mana keakuratan dalam memprediksi kondisi jantung dapat menyelamatkan nyawa. Di sisi lain, dataset seperti Diabetes dan Kanker Payudara membutuhkan model untuk mengelola data yang tidak seimbang, di mana prevalensi satu kelas di atas kelas lainnya dapat membiaskan proses pembelajaran, yang berpotensi menyebabkan diagnosis yang tidak akurat. Kompleksitas lebih lanjut diperkenalkan dengan dimasukkannya dataset seperti MNIST, yang berbeda secara signifikan dalam hal dimensi dan struktur kelas. Dataset MNIST, dengan ruang dimensi tinggi yang terdiri dari gambar digit tulisan tangan, menantang model untuk secara efisien memproses dan mengklasifikasikan pola visual yang kompleks. Selain itu, pemilihan dataset dengan ukuran sampel yang lebih kecil mencerminkan skenario umum di mana klasifikasi berkinerja tinggi harus dicapai meskipun ketersediaan data terbatas.

Rincian dataset yang dipilih, termasuk fitur-fitur spesifik dan distribusi kelas, dikatalogkan dengan cermat dalam Tabel II. Tabel ini berfungsi sebagai titik acuan untuk memahami tantangan data yang beragam yang dirancang untuk ditangani oleh FBE, yang menggambarkan penerapan algoritme yang luas dan kinerja yang kuat di berbagai skenario kompleks.

B. Pengaturan Eksperimental

1) Alat komputasi: Eksperimen dilakukan pada sistem operasi Linux Fedora 32, menggunakan Intel Core i7-4790 CPU pada 3,6 GHz dan dilengkapi dengan RAM 32 GB. Konfigurasi ini, mirip dengan pengaturan komputasi pribadi kelas atas, menyediakan lingkungan yang stabil dan seimbang yang cocok untuk fase pengembangan dan evaluasi.

TABEL. II. Dataset yang Digunakan untuk Eksperimen

ID.	Nama Dataset	Contoh	Fitur	Kelas
1	Kanker Payudara	569	30	2
2	Limfografi	148	18	4
3	Kesuburan	100	8	2
4	EKG	109.446	187	5
5	Diabetes	768	8	2
6	Cmc	150	4	3
7	Spambase	110.201	4	2
8	MNIST	70.000	784	10
9	Kaca	214	9	6
10	Sihir			

Python dipilih sebagai bahasa pemrograman utama karena dukungannya yang luas untuk pembelajaran mesin. Pustaka Python utama yang digunakan dalam penyiapan meliputi:

- Numpy: Memfasilitasi komputasi numerik yang efisien dengan dukungan untuk array dan matriks multi-dimensi yang besar. Pustaka ini sangat penting untuk pengoptimalan kinerja dalam aplikasi yang intensif data.
- Panda: Menawarkan kemampuan manipulasi data yang kuat yang menyederhanakan pembersihan, transformasi, dan analisis data, yang penting untuk menyiapkan set data untuk pembelajaran mesin.
- Scikit-Learn: Menyediakan beragam algoritme dan alat pembelajaran mesin, sehingga sangat diperlukan untuk pelatihan, evaluasi, dan perbandingan model.

2) Pembandingan model dan pengaturan parameter: Untuk mengkontekstualisasikan kinerja FBE, FBE dibandingkan dengan tiga pengklasifikasi yang sudah dikenal luas: *k-Nearest* Neighbors (KNN), Random Forest (RF), dan AdaBoost. Pengklasifikasi ini dipilih karena popularitas dan rekam jejak yang telah terbukti di lingkungan akademis dan industri, yang berfungsi sebagai dasar yang kuat untuk perbandingan. Konfigurasi parameter untuk eksperimen dipilih dengan cermat untuk menyeimbangkan antara kompleksitas model dan kinerja prediktif:

- KNN dan FBE: Dikonfigurasikan dengan tiga tetangga terdekat ($k=3$), pengaturan standar untuk KNN yang menawarkan keseimbangan antara underfitting dan overfitting. Untuk FBE, parameter tambahan termasuk tiga iterasi ($m=3$) untuk menguji efek ensemble dan parameter rahmat ($g=0$) untuk mengevaluasi dampaknya terhadap sensitivitas dan spesifisitas model.

- Hutan Acak: Menggunakan 100 pohon keputusan yang telah berkembang sepenuhnya untuk memaksimalkan efek ensemble, meningkatkan kemampuan model untuk melakukan generalisasi di berbagai set data.
- AdaBoost: Mirip dengan RF dalam hal jumlah pohon keputusan, namun dengan pohon yang dipangkas pada level satu untuk fokus pada pengurangan overfitting, sehingga meningkatkan generalisasi model.

3) Metrik evaluasi kinerja: Metrik komprehensif dipilih untuk mengevaluasi performa model yang diuji secara komprehensif:

- Akurasi: Memberikan ukuran umum ketepatan model di semua kelas, berguna untuk penilaian awal keampuhan model.
- Presisi: Sangat penting untuk aplikasi di mana biaya positif palsu cukup signifikan, membantu mengukur keandalan prediksi positif.
- Mengingat: Terutama penting dalam aplikasi medis atau keuangan di mana kegagalan mendeteksi hal positif dapat menimbulkan konsekuensi serius, hal ini mengukur kemampuan model untuk menangkap semua contoh yang relevan.
- Skor F1: Menggabungkan presisi dan recall ke dalam satu metrik yang mengukur akurasi model dalam mengidentifikasi hanya contoh yang relevan, yang sangat penting untuk mengevaluasi kinerja dalam set data yang tidak seimbang.

4) Protokol evaluasi: Untuk memastikan evaluasi menyeluruh terhadap FBE dan membandingkan kinerjanya dengan sistem konvensional.

model, kami menggunakan metodologi validasi silang yang kuat. Secara khusus, data dibagi dalam rasio 7:3, dengan 70% digunakan untuk melatih model dan 30% sisanya didedikasikan untuk pengujian. Rasio pembagian yang diterima secara luas ini memungkinkan data pelatihan yang substansial sekaligus menyediakan data pengujian yang cukup untuk menilai generalisasi model secara efektif. Selanjutnya, proses validasi silang diulang sebanyak 10 kali untuk memastikan keandalan dan stabilitas metrik kinerja. Setiap iterasi secara acak mendistribusikan ulang data sesuai dengan rasio pelatihan dan pengujian 7:3, meminimalkan bias dan variabilitas dalam evaluasi. Metrik kinerja dihitung untuk setiap kali proses, dan hasilnya kemudian dirata-ratakan di seluruh 10 iterasi untuk menghasilkan ukuran kinerja akhir.

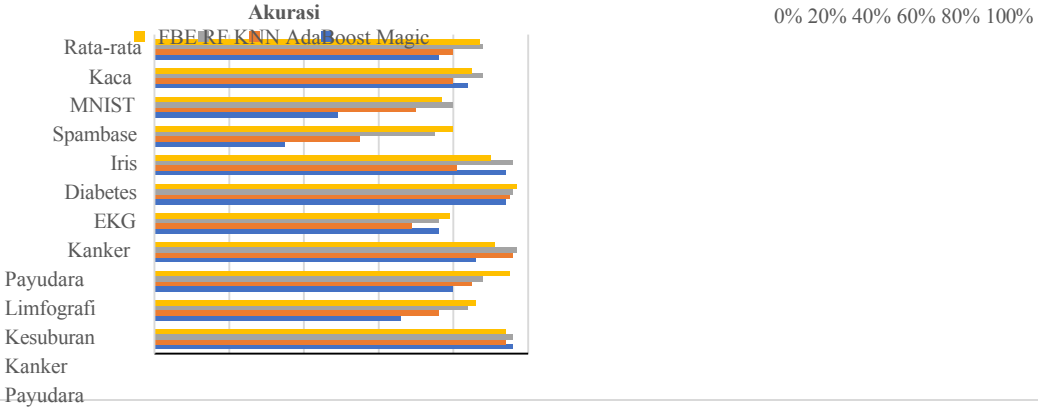
C. Analisis dan Perbandingan Kinerja

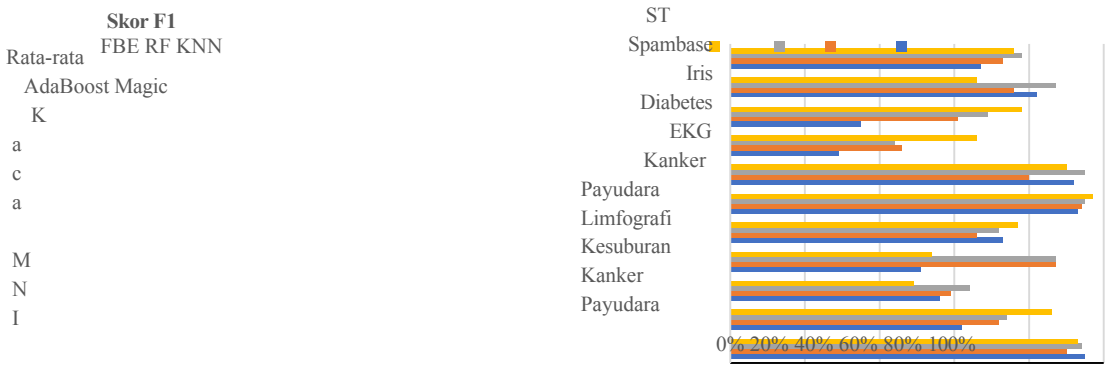
Hasil eksperimen dirinci secara menyeluruh dalam Tabel III dan IV. Representasi visual dari hasil-hasil ini disajikan pada Gbr. 3 dan 4.

Tabel III dan Gambar 3 menunjukkan bahwa FBE secara konsisten mencapai akurasi yang tinggi di seluruh dataset yang diuji, dengan kinerja yang menonjol pada dataset seperti Iris (97%), Fertility (95%), dan MNIST (80%). Hasil ini menunjukkan kemampuan FBE yang kuat untuk menangani struktur data yang sederhana dan kompleks. Secara rata-rata, FBE mencapai akurasi 87%, yang mana 7% lebih tinggi dari KNN dan 11% lebih tinggi dari AdaBoost, dan mengungguli RF dengan selisih hanya 1%. Hal ini menunjukkan bahwa FBE memberikan alternatif yang kuat untuk model yang lebih mapan, terutama dalam menangani berbagai jenis data secara efektif.

TABEL. III. AKURASI DAN PERFORMA SKOR F1 MODEL YANG DIBANDINGKAN PADA BERBAGAI DATASET

ID.	Dataset	Akurasi				Skor F1			
		AdaBoost	*KNN*	*RF*	*FBE*	*AdaBoost*	*KNN*	*RF*	*FBE*
1	Kanker Payudara	0.96	0.94	0.96	0.94	0.95	0.90	0.94	0.93
2	Limfografi	0.66	0.76	0.84	0.86	0.62	0.72	0.74	0.86
3	Kesuburan	0.80	0.85	0.88	0.95	0.56	0.59	0.64	0.49
4	EKG	0.86	0.96	0.97	0.91	0.51	0.87	0.87	0.54
5	Diabetes	0.76	0.69	0.76	0.79	0.73	0.66	0.72	0.77
6	Iris	0.94	0.95	0.96	0.97	0.93	0.94	0.95	0.97
7	Spambase	0.94	0.81	0.96	0.90	0.92	0.80	0.95	0.90
8	MNIST	0.35	0.55	0.75	0.80	0.29	0.46	0.44	0.66
9	Kaca	0.49	0.70	0.80	0.77	0.35	0.61	0.69	0.78
10	Sihir	0.84	0.80	0.88	0.85	0.82	0.76	0.87	0.66
Rata-rata		0.76	0.80	0.88	0.87	0.67	0.73	0.78	0.76





Gbr. 3. Akurasi dan performa F1 Score dari model yang dibandingkan di berbagai set data

Tabel dan gambar ini juga menggambarkan bagaimana F1 Score menyoroti kinerja FBE yang seimbang dalam hal presisi dan recall. Khususnya, FBE mencapai skor 78% pada dataset Glass, secara signifikan mengungguli AdaBoost yang mencapai 35% dan KNN yang mencapai 61%. Pada dataset Iris, FBE mencapai 97% yang mengesankan. Hasil ini menggarisbawahi keefektifan model dalam skenario di mana menyeimbangkan positif palsu dan negatif palsu sangat penting. Dengan skor F1 rata-rata 76%, FBE melampaui KNN dan AdaBoost, menunjukkan kemampuannya yang unggul untuk menyelaraskan recall dan presisi di berbagai aplikasi.

Dalam hal presisi, seperti yang diuraikan pada Tabel IV dan Gambar 4, FBE menunjukkan hasil yang patut dicontoh, terutama pada dataset seperti Iris (97%) dan Fertility (95%), di mana akurasi pada nilai prediktif positif sangat penting. Ketepatan rata-rata FBE di semua dataset mencapai 86%, lebih tinggi daripada AdaBoost dan KNN, menggarisbawahi keandalannya dalam mengklasifikasikan contoh dengan benar.

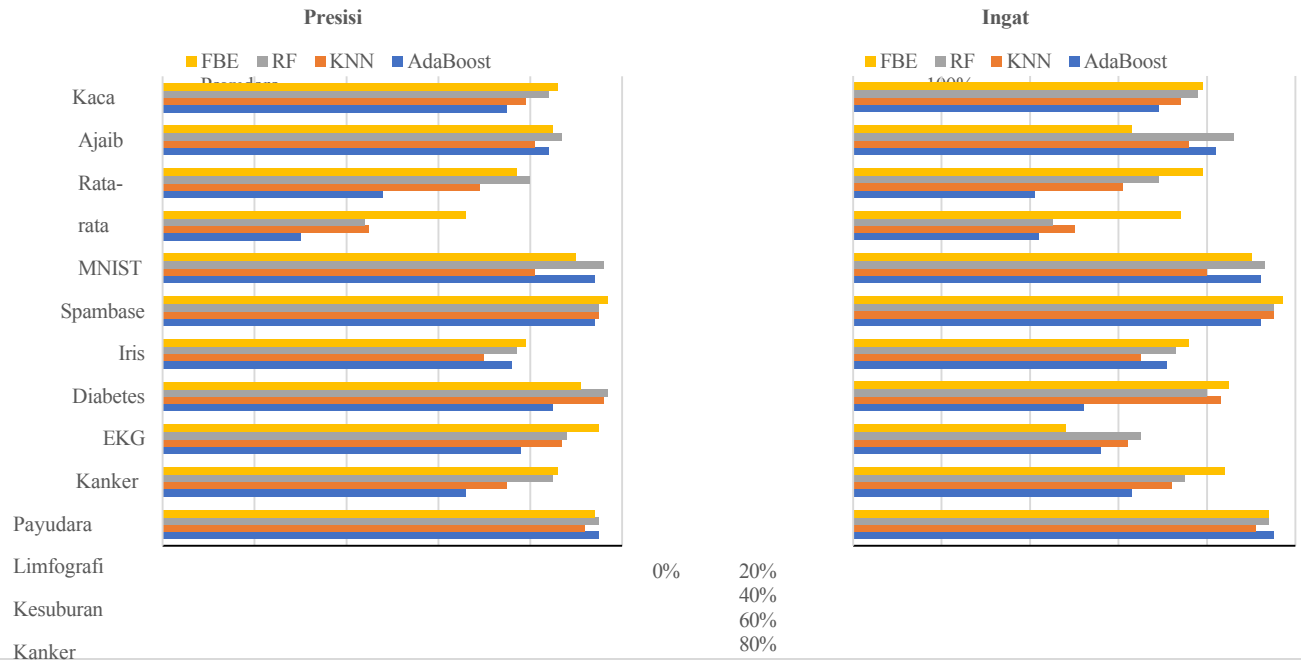
Untuk metrik penarikan kembali, Tabel IV dan Gambar 4 mengungkapkan kekuatan FBE dalam hal sensitivitas, terutama yang menonjol pada set data seperti ECG (85%) dan Glass (79%). FBE mempertahankan penarikan rata-rata 79%, yang menunjukkan keefektifannya dalam mengidentifikasi semua contoh yang relevan di berbagai kumpulan data. Kemampuan ini sangat penting untuk aplikasi di mana kehilangan sebuah instance dapat berdampak signifikan.

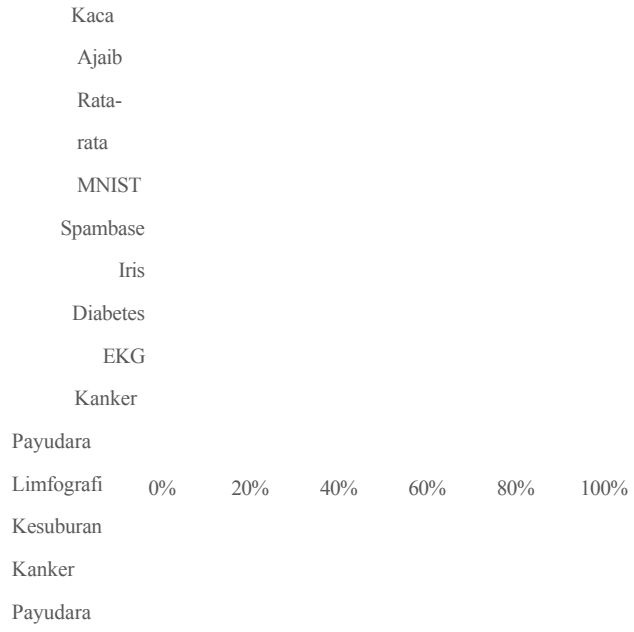
Analisis komparatif mengungkapkan bahwa FBE tidak hanya bersaing ketat dengan, tetapi dalam banyak kasus mengungguli model tradisional. Hal ini terutama terlihat dari keunggulannya yang konsisten atas AdaBoost dan sering mengungguli KNN. Meskipun RF sering menunjukkan metrik yang sedikit lebih tinggi, kesenjangannya kecil, menunjukkan bahwa FBE dapat menawarkan kinerja yang sebanding dengan manfaat tambahan berupa efisiensi dalam pemrosesan dan kesederhanaan model.

Efektivitas FBE dapat dikaitkan dengan pendekatan inovatifnya dalam menangani set data. Dengan berfokus pada fitur yang paling informatif melalui metode subset dan ensemble yang diurutkan, FBE mengurangi dampak dari fitur-fitur yang berisik atau tidak relevan yang biasanya memengaruhi algoritme KNN. Prioritas fitur ini tidak hanya meningkatkan akurasi tetapi juga meningkatkan kemampuan model untuk menggeneralisasi di berbagai jenis data, menghindari overfitting yang biasa terjadi pada model tradisional.

TABEL. IV. PERFORMA PRESISI DAN RECALL DARI MODEL YANG DIBANDINGKAN PADA BERBAGAI DATASET

ID.	Dataset	Presisi				Ingat			
		AdaBoost	*KNN*	*RF*	*FBE*	*AdaBoost*	*KNN*	*RF*	*FBE*
1	Kanker Payudara	0.95	0.92	0.95	0.94	0.95	0.91	0.94	0.94
2	Limfografi	0.66	0.75	0.85	0.86	0.63	0.72	0.75	0.84
3	Kesuburan	0.78	0.87	0.88	0.95	0.56	0.62	0.65	0.48
4	EKG	0.85	0.96	0.97	0.91	0.52	0.83	0.80	0.85
5	Diabetes	0.76	0.70	0.77	0.79	0.71	0.65	0.73	0.76
6	Iris	0.94	0.95	0.95	0.97	0.92	0.95	0.95	0.97
7	Spambase	0.94	0.81	0.96	0.90	0.92	0.80	0.93	0.90
8	MNIST	0.30	0.45	0.44	0.66	0.42	0.50	0.45	0.74
9	Kaca	0.48	0.69	0.80	0.77	0.41	0.61	0.69	0.79
10	Sihir	0.84	0.81	0.87	0.85	0.82	0.76	0.86	0.63
Rata-rata		0.75	0.79	0.84	0.86	0.69	0.74	0.78	0.79





Gbr. 4. Performa Presisi dan Recall dari model yang dibandingkan di berbagai dataset

Ada banyak alasan di balik peningkatan kinerja FBE:

- Keuntungan ensemble: FBE menggunakan beberapa subset yang diurutkan, mengurangi varians dan meningkatkan keandalan melalui rata-rata ensemble. Pendekatan ini mengurangi dampak dari titik data pencilan dan noise fitur, yang secara signifikan dapat memengaruhi model seperti KNN dan AdaBoost.
- Pemilihan fitur: Dengan berfokus secara iteratif pada fitur yang paling informatif, FBE meminimalkan tantangan dimensi dan noise fitur yang tidak relevan, yang sering menjadi masalah dalam dataset berdimensi tinggi seperti MNIST.

Singkatnya, hasil terperinci menyoroti potensi tinggi FBE untuk menangani data berdimensi tinggi atau data yang tidak seimbang. Kinerjanya di semua metrik menunjukkan ketangguhan dan kemampuan beradaptasi dalam mengatasi berbagai tantangan klasifikasi. Evaluasi komprehensif FBE terhadap model standar menunjukkan efektivitas dan keserbagunaannya di berbagai set data. Pemilihan fitur yang sistematis dan penggunaan metode ensembel meningkatkan akurasi dan keandalannya dalam tugas klasifikasi yang kompleks, yang mencakup berbagai bidang, mulai dari perawatan kesehatan hingga pemrosesan gambar.

V. KESIMPULAN DAN ARAH MASA DEPAN

Dalam penelitian ini, kami telah memperkenalkan algoritma ensemble yang kuat yang bertujuan untuk meningkatkan kinerja klasifikasi *k-nearest* neighbors melalui penggunaan fitur bagging yang inovatif. Metode kami melibatkan pemilihan subset fitur, menentukan fitur yang paling informatif dalam subset ini menggunakan metrik informasi timbal balik, dan memanfaatkan fitur ini untuk mengurutkan subset data. Penyortiran ini memfasilitasi pencarian biner yang efisien selama fase pengujian untuk dengan cepat menemukan perkiraan tetangga terdekat, dan proses ini diulang beberapa kali untuk meningkatkan kinerja dan keandalan klasifikasi. Algoritma yang diusulkan juga menjalani analisis kompleksitas yang ketat dalam fase pelatihan dan pengujian. Analisis ini menegaskan bahwa pendekatan kami tidak hanya meningkatkan metrik kinerja, tetapi juga mengurangi overhead komputasi secara signifikan, dari kompleksitas linear ke logaritmik.

Validasi eksperimental kami menunjukkan bahwa algoritme yang diusulkan secara signifikan mengungguli *k-nearest* neighbor tradisional dan AdaBoost dalam hal akurasi, presisi, recall, dan skor F1 di berbagai dataset, termasuk dataset yang memiliki data berdimensi tinggi dan data yang tidak seimbang. Khususnya, pendekatan kami menunjukkan peningkatan yang nyata pada dataset seperti MNIST, di mana *k-nearest* neighbor tradisional biasanya mengalami kesulitan karena adanya masalah dimensi dan sensitivitas noise. Algoritma ensemble secara konsisten mencapai tingkat akurasi yang lebih tinggi, sering kali melebihi kinerja Random Forest dalam skenario tertentu, terutama dengan set data yang tidak seimbang.

Penelitian di masa depan akan memperluas studi ketahanan algoritme kami di berbagai set data yang lebih luas, terutama mengeksplorasi kinerjanya di bawah kondisi distribusi data yang ekstrem dan ketidakseimbangan kelas. Penelitian ini membuka jalan bagi penelitian di masa depan untuk mengeksplorasi pendekatan hibrida yang menggabungkan feature bagging dengan teknik pembelajaran mesin lainnya untuk lebih meningkatkan kinerja klasifikasi dan efisiensi komputasi.

UCAPAN TERIMA KASIH

Penelitian ini didukung oleh Sekolah Tinggi Teknologi Informasi dan Komunikasi (CICT) di Universitas Can Tho. Kami mengucapkan terima kasih yang sebesar-besarnya kepada Laboratorium Big Data dan Komputasi Seluler CICT atas bantuan mereka yang tak ternilai. Selain itu, kami menerima dukungan dari program penelitian dan inovasi Horizon Uni Eropa di bawah perjanjian hibah MSCA-SE (Marie Skłodowska-Curie Actions Staff Exchange) 101086252; Telp: HORIZON-MSCA-2021-SE-01;
Judul proyek: STARWARS (Manajemen Data Heterogen Berbasis AI untuk Jaringan Air Hujan dan Air Limbah).

REFERENSI

[1] M. Magris dan A. Iosifidis, "Pembelajaran Bayesian untuk jaringan syaraf: survei algoritmik", Tinjauan Kecerdasan Buatan, 56.10 (2023): 11773-
11823, 2023.

[2] I.H. Sarker, "Pembelajaran mesin: Algoritme, aplikasi dunia nyata, dan arah penelitian", Springer Nature Computer Science, 2.3 (2021): 160, 2021.

[3] Y. Eren dan I. Kucukdemiral, "Sebuah tinjauan komprehensif tentang pendekatan pembelajaran mendalam untuk peramalan beban jangka pendek", Renewable and Sustainable Energy Reviews, 189 (2024): 114031, 2024.

[4] P. Cunningham dan S.J. Delany, "*k-nearest* neighbour classifiers - a tutorial", ACM Computing Surveys (CSUR), 54.6, pp. 1-25, 2021.

[5] A. Shokrzade, M. Ramezani, F.A. Tab dan M.A. Mohammad, "Metode klasifikasi knn berbasis mesin pembelajaran ekstrim baru untuk menangani data besar", Sistem Pakar dengan Aplikasi, 183, 115293 (2021).

[6] S. Shekhar, N. Hoque dan D.K. Bhattacharyya, "Pknn-mifs: Pengklasifikasi knn paralel atas subset fitur yang optimal", Intelligent Systems with Applications, 14, 200073 (2022).

[7] P. Gupta, A. Jindal, Jayadeva dan S. Debarka, "Combi: Pohon pencarian biner terkompresi untuk perkiraan pencarian k-nn di ruang hamming", Big Data Research, 25, 100223 (2021).

[8] A.B. Hassanat, "Pohon pencarian biner berbasis norma untuk mempercepat klasifikasi data besar knn", Computers, 7(4), 54 (2018).

[9] P. Pappula, "Sebuah metode pohon pencarian biner baru untuk menemukan sebuah item menggunakan penskalaan ", International Arab Journal of Information Technology, 19(5),
Hal. 713-720, 2022.

[10] H. Saadatfar, S. Khosravi, J.H. Joloudari, A. Mosavi, S. Shamshirband, "Pengklasifikasi *k-nearest* neighbour yang baru untuk data besar berdasarkan pemangkasan data yang efisien", Mathematica 8(2), 286, 2020.

[11] H. Wang, P. Xu dan J. Zhao, "Algoritma knn yang ditingkatkan untuk wilayah bola berdasarkan pengelompokan dan pembagian wilayah", Alexandria Engineering Journal 61(5), pp. 3571-3585, 2022.

[12] N. Islam, M. Fatema-Tuj-Jahra, M.T. Hasan and D.M. Farid, "Knntree: Metode baru untuk memperbaiki klasifikasi *k-nearest* neighbour menggunakan pohon keputusan", Dalam Konferensi Internasional Teknik Elektro, Komputer dan Komunikasi (ECCE), hal. 1-6, IEEE, 2023.