FORMAT: Jurnal Ilmiah Teknik Informatika
Vol. 15, No. 1, Januari 2026, pp. 6-21
Akreditasi Sinta 5 RISTEKBRIN No. 164/E/KPT/2021
DOI: 10.22441/format.2026.v15.i1.002
E-ISSN 2722-7162
P-ISSN 2089-5615

Perbandingan Algoritma CART Dan AdaBoost Pada Klasifikasi Demensia

Muhammad Arya All Fajri[1], M Aldi Saputra[2], Anita Desiani[3*], Bambang Suprihatin[4], Herlina Hanum[5]

[a] Matematika, Universitas Sriwijaya, Jl. Raya Palembang – Prabumulih No. Km. 32, Ogan Ilir 3086, Sumatera Selatan

[1]alfajrimaria@gmail.com, [2]08011382126096@student.unsri.ac.id, [3*]anita_desiani@unsri.ac.id, [4]bambangs@unsri.ac.id, [5]herlina@mipa.unsri.ac.id

*Penulis Korespondensi

ABSTRAK

Demensia merupakan gangguan kesehatan ditandai dengan penurunan daya ingat, kemampuan kognitif, dan perilaku yang mengganggu aktivitas pada kehidupan sehari-hari. Masyarakat kurang mendapatkan informasi mengenai deteksi dini demensia yang disebabkan terbatasnya fasilitas kesehatan. Klasifikasi menggunakan *data mining* dapat membantu deteksi dini demensia. Penelitian ini bertujuan membandingkan algoritma CART dan AdaBoost untuk melihat metode yang paling efektif digunakan pada klasifikasi demensia. Pembagian data dilakukan menggunakan metode *percentage split* dan *k-fold cross-validation*. *Percentage split* membagi data menjadi dua bagian dengan 70% data pelatihan dan 30% data pengujian. *K-fold cross-validation* mengelompokkan data dengan 1 kelompok data menjadi data pengujian dan 9 kelompok data lainnya menjadi data pengujian yang dilakukan berulang pada setiap kelompok data sebanyak 10 kali. ADASYN digunakan untuk menyeimbangkan data pada setiap kelas. Hasil evaluasi kinerja pada kedua algoritma menunjukkan AdaBoost menggunakan ADASYN dan *k-fold cross-validation* memiliki nilai tertinggi untuk akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC masing-masing sebesar 92.52%, 92.11%, 92.52%, 91.46%, dan 96.85%. Hasil ini menunjukkan bahwa algoritma AdaBoost sangat baik dalam memprediksi seluruh demensia dengan benar, mempertahankan keseimbangan antara presisi dan *recall*, dan membedakan tiga kelas demensia. Hasil penelitian menunjukkan keunggulan pendekatan *ensemble learning* dalam menangani variasi data dan meningkatkan stabilitas model klasifikasi demensia. Penelitian ini menunjukkan bahwa AdaBoost memiliki performa yang sangat baik dibandingkan CART pada klasifikasi demensia.

Article Info

Kata Kunci:
dementia
CART
AdaBoost
percentage split
k-fold cross-validation

Riwayat artikel:
Submit 27 Nov 2024
Revisi 11 Nov 2025
Diterima 30 Nov 2025

1. PENDAHULUAN

Demensia merupakan gangguan kesehatan ditandai dengan penurunan daya ingat, kemampuan kognitif, dan perilaku yang mengganggu aktivitas pada kehidupan sehari-hari [1]. Data WHO (2025) menunjukkan bahwa terdapat lebih dari 55 juta penderita demensia di dunia, dengan peningkatan 10 juta kasus baru setiap tahun [2]. Demensia dapat menyebabkan kecacatan serta ketergantungan yang mengganggu kehidupan orang yang terkena dampak, pengasuh, dan keluarga [1]. Masyarakat kurang mendapatkan informasi mengenai deteksi dini demensia yang disebabkan terbatasnya fasilitas kesehatan. Deteksi dini demensia perlu dilakukan untuk mengetahui penyebab serta menentukan langkah penanganan yang tepat agar demensia tidak bertambah parah. Deteksi demensia dapat dilakukan menggunakan bantuan komputer dengan melakukan klasifikasi menggunakan *data mining*. *Data mining* merupakan serangkaian metode yang digunakan untuk menganalisis kumpulan data besar dan mengekstrak informasi untuk menemukan pola yang bermakna [3]. Salah satu algoritma pada *data mining* adalah *Classification and Regresion Tree* (CART).

E-ISSN 2722-7162
P-ISSN 2089-5615

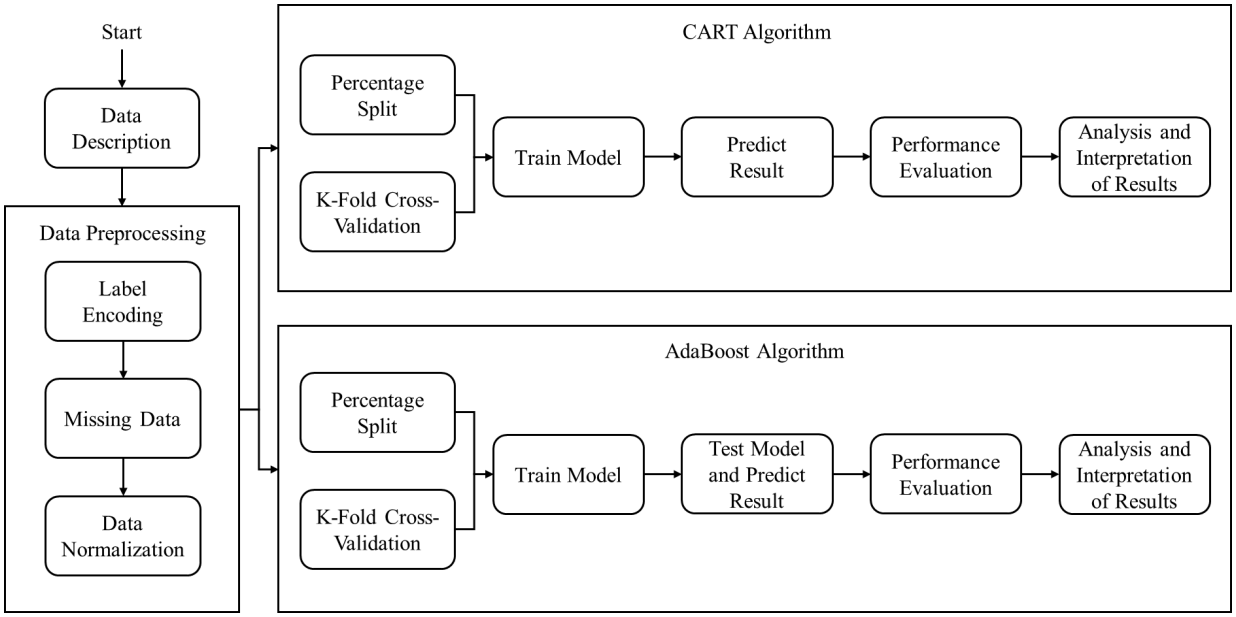
CART merupakan metode *nonparametric regression* yang membagi data berdasarkan variabel atau atribut sehingga membentuk pohon keputusan [4]. CART terdiri dari dua jenis pohon keputusan, yaitu *classification tree* dan *regression tree*. *Classification tree* digunakan untuk memprediksi variabel kategorik sedangkan *regression tree* digunakan untuk memprediksi variabel numerik [5]. CART mampu memberikan hasil yang akurat dan lebih mudah diinterpretasikan karena strukturnya yang berbentuk pohon keputusan memungkinkan pelacakan setiap keputusan secara menyeluruh dalam proses klasifikasi atau regresi [6]. Diwate *et al* [7] melakukan klasifikasi demensia dengan nilai akurasi dan *recall* dibawah 85%. CART kurang mampu menghasilkan pohon keputusan yang stabil karena sensitif terhadap data baru [8]. Algoritma yang mampu stabil terhadap data baru adalah algoritma *Adaptive Boosting* (AdaBoost).

AdaBoost merupakan metode *ensemble* yang mengkombinasikan beberapa klasifikasi tunggal (*single classifier*) didalam satu model sehingga membentuk model klasifikasi yang kuat (*strong learner*) dengan mengambil model lemah (*weak learner*) [9]. AdaBoost mampu menutupi kekurangan algoritma klasifikasi tunggal (*single classifier*) seperti CART karena menggabungkan beberapa model lemah (*weak learner*) menjadi satu model kuat (*strong learner*) [10], [11]. Dhakal *et al* [12] melakukan klasifikasi demensia menggunakan algoritma AdaBoost dengan nilai akurasi, presisi, *recall*, dan *f1-score* dibawah 90%.

Penelitian sebelumnya banyak berfokus pada klasifikasi citra otak menggunakan CNN, sedangkan penelitian ini mengeksplorasi pendekatan tabular data berbasis kognitif dengan CART dan AdaBoost. Keterbaruan penelitian ini terletak pada analisis perbandingan kinerja algoritma CART dan AdaBoost pada klasifikasi demensia berdasarkan fitur kognitif dan pemanfaatan data tabular kognitif untuk klasifikasi demensia. Penelitian ini bertujuan membandingkan algoritma CART dan AdaBoost untuk melihat metode yang paling efektif digunakan pada klasifikasi demensia. Tingkat keberhasilan metode yang diusulkan dalam penelitian ini diukur dengan menghitung hasil kinerja akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC.

2. METODE PENELITIAN

Metodologi penelitian menjelaskan tahapan-tahapan yang dilakukan dalam penelitian ini, mulai dari deskripsi data hingga analisis dan interpretasi hasil. Ilustrasi dari alur metodologi penelitian dapat dilihat pada Gambar 1.



Gambar 1. Alur Metodologi Penelitian

Berdasarkan Gambar 1 mengilustrasikan proses alur metodologi penelitian. Tahapan awal dari alur kerangka kerja penelitian adalah tahapan deskripsi data (*data description*). Tahapan prapemrosesan data (*data preprocessing*) dilakukan dengan mengubah kategori menjadi bentuk numerik (*label encoding*), penanganan data hilang (*missing data*), dan normalisasi data (*data normalization*). Tahapan selanjutnya data dibagi menggunakan dua pendekatan, yaitu *percentage split* dan *k-fold cross-validation*. Pembagian data dilakukan pada algoritma CART dan AdaBoost. Tahapan selanjutnya adalah pelatihan model (*train model*) menggunakan data pelatihan. Pengujian model (*test model*) dilakukan untuk menghasilkan prediksi hasil (*predict result*) pada data pengujian. Tahapan evaluasi kinerja (*performance evaluation*) dilakukan untuk mengukur performa model menggunakan metrik evaluasi kinerja yang

E-ISSN 2722-7162
P-ISSN 2089-5615

ditetapkan. Tahapan terakhir adalah analisis dan interpretation hasil (*analysis and interpretation of result*) yang dilakukan untuk menafsirkan performa dan kesesuaian model terhadap tujuan penelitian.

2.1. Deskripsi Data

Data yang digunakan pada penelitian ini berasal dari *Open Access Series of Imaging Studies* (OASIS) yang diperoleh dari *Kaggle* [13]. Dataset OASIS merupakan dataset yang berisi 150 data pasien demensia. Deskripsi data dapat dilihat pada Tabel 1.

Tabel 1. Deskripsi Data

Atribut	Keterangan	Tipe Data	Atribut	Jumlah Data	Nilai
Subject ID	Identitas Pasien (*Subject Identification*)	Kategorik	Identitas	373	OAS2_ID
MRI ID	Identifikasi Pemeriksaan MRI (*MRI Exam Identification*)	Kategorik	Identitas	373	OAS2_ID_MRID
Group	Kelas Prediksi	Kategorik	Target/Prediksi	373	*Nondemented* *Demented* *Converted*
Visit	Banyak Pemeriksaan	Kategorik	Fitur	373	1 - 5
MR Delay	Waktu Penundaan MRI	Numerik	Fitur	373	0 - 2639
M/F	Jenis Kelamin	Kategorik	Fitur	373	M = Laki-laki F = Perempuan
Hand	Tangan	Kategorik	Konstan	373	R = Tangan Kanan
Age	Usia	Numerik	Fitur	373	60 - 98
Educ	Tingkat Pendidikan (Tahun)	Numerik	Fitur	373	6 - 23
SES	Status Sosial-Ekonomi (*Socioeconomic Status*)	Kategorik	Fitur	354	1 = Tinggi 2 = Menengah ke Atas 3 = Menengah 4 = Menengah ke Bawah 5 = Rendah
MMSE	Uji Status Mental Mini (*Mini Mental State Examination*)	Numerik	Fitur	371	4 - 30
CDR	Tingkat Keparahan Demensia (*Clinical Dementia Rating*)	Kategorik	Fitur	373	0 = Tidak 0.5 = Sangat lemah 1 = Lemah 2 = Sedang
eTIV	Perkiraan Volume Rongga Kepala (*estimated Total Intracranial Volume*)	Numerik	Fitur	373	1106 - 2004
nWBV	Volume Otak Total yang Dinormalisasi (*normalize Whole Brain Volume*)	Numerik	Fitur	373	0.64 – 0.84
ASF	Faktor Penyesuaian Ukuran Otak (*Atlas Scaling Factor*)	Numerik	Fitur	373	0.88 – 1.59

Berdasarkan Tabel 1, dataset OASIS memiliki 15 atribut, yaitu *Subject ID*, *MRI ID*, *Group*, *Visit*, *MR Delay*, *M/F(Gender)*, *Hand*, *Age*, *Educ*, *SES*, *MMSE*, *CDR*, *eTIV*, *nWBF*, dan *ASF*. Dataset ini terdiri dari 8 atribut kategorik dan 7 atribut numerik. Atribut kategorik terdiri dari *Subject ID*, *MRI ID*, *Group*, *Visit*, *M/F*, *Hand*, *SES*, dan *CDR*. Atribut numerik terdiri dari *MR Delay*, *Age*, *EDUC*, *MMSE*, *eTIV*, *nWBV*, *ASF*. Dataset ini terdiri dari 11 atribut fitur, 2 atribut identitas, 1 atribut konstan, dan 1 atribut target/prediksi, yaitu *Group*. Atribut *Group* memiliki tiga kelas prediksi, yaitu *nondemented*, *demented*, dan *converted*. Dataset ini memiliki 373 data yang terbagi menjadi 190 data *nondemented*, 146 data *demented* dan 37 data *converted*.

E-ISSN 2722-7162
P-ISSN 2089-5615

2.2. *Data Preprocessing*

Data preprocessing merupakan proses mengubah data menjadi format yang sesuai, sehingga dapat diproses secara akurat menggunakan model. *Data preprocessing* dilakukan melalui beberapa langkah, yaitu *label encoding*, *missing data*, dan normalisasi data [14]. Atribut *Subject ID*, *MRI ID*, dan *Hand* dihapus dari data.

2.2.1. *Label Encoding*

Label encoding merupakan proses mengubah data kategori menjadi bentuk numerik [15]. Atribut *Group*, *M/F*, dan *SES* diproses menggunakan *label encoding* sehingga setiap atribut kategorik diubah menjadi bentuk numerik.

2.2.2. *Missing Data*

Missing data merupakan kondisi di mana terdapat nilai yang hilang atau tidak tercatat pada suatu data [16]. Data yang digunakan pada penelitian ini memiliki dua atribut yang mempunyai data hilang di dalamnya, yaitu *SES* dan *MMSE*. Atribut *MMSE* terdapat 2 data hilang dan Atribut *SES* terdapat 19 data hilang. Atribut *MMSE* dilakukan imputasi dengan menggunakan median, yaitu nilai tengah dari seluruh data kolom *MMSE* menjadi pegganti nilai pada data hilang. Atribut *SES* merupakan data kategorik, sehingga data kosong diisi dengan modus, yaitu data yang memiliki frekuensi terbanyak menjadi pengganti nilai pada data hilang.

2.2.3. Normalisasi Data

Normalisasi data merupakan proses menyesuaikan *output* ke skala yang diinginkan [17]. Normalisasi data dilakukan untuk mengecilkan rentang nilai yang memiliki perbedaan rentang nilai yang besar dibandingkan dengan atribut-atribut lainnya, sehingga proses klasifikasi lebih optimal [18]. Normalisasi data pada penelitian ini menggunakan teknik *min-max normalization*. *min-max normalization* merupakan teknik normalisasi data yang melakukan transformasi linier pada data asli dengan memetakan nilai ke rentang baru berdasarkan nilai *maksimum* dan *minimum* atribut [19]. Proses perhitungan *min-max normalization* dapat dilihat pada Persamaan (1) [20].

$$x_{norm} = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (1)$$

Dimana x_{norm} merupakan normalisasi, x_i merupakan *input* data ke-i, i merupakan data ke-i, x_{min} merupakan data dengan nilai terendah, dan x_{max} merupakan data dengan nilai tertinggi.

Penelitian ini melakukan normalisasi data terhadap 10 atribut, yaitu *Visit*, *MR Delay*, *Age*, *Educ*, *SES*, *MMSE*, *CDR*, *eTIV*, *nWBF*, dan *ASF*. Hasil *data preprocessing* dapat dilihat pada Tabel 2.

Tabel 2. Hasil *Data Preprocessing*

Atribut	Keterangan	Tipe Data	Atribut	Jumlah Data	Nilai
Visit	Banyak Pemeriksaan	Kategorik	Fitur	373	0 - 1
MR Delay	Waktu Penundaan MRI	Numerik	Fitur	373	0 - 1
Age	Usia	Numerik	Fitur	373	0 - 1
Educ	Tingkat Pendidikan (Tahun)	Numerik	Fitur	373	0 - 1
SES	Status Sosial-Ekonomi (*SocioEconomic Status*)	Kategorik	Fitur	373	0 = Tinggi 0.25 = Menengah ke Atas 0.5 = Menengah 0.75 = Menengah ke Bawah 1 = Rendah
MMSE	Uji Status Mental Mini (*Mini Mental State Examination*)	Numerik	Fitur	373	0 - 1
CDR	Tingkat Keparahan Demensia (*Clinical Dementia Rating*)	Kategorik	Fitur	373	0 = Tidak 0.25 = Sangat lemah 0.5 = Lemah 1 = Sedang

 E-ISSN 2722-7162
P-ISSN 2089-5615

eTIV	Perkiraan Volume Rongga Kepala (*estimated Total Intracranial Volume*)	Numerik	Fitur	373	0 - 1
nWBV	Volume Otak Total yang Dinormalisasi (*normalize Whole Brain Volume*)	Numerik	Fitur	373	0 - 1
ASF	Faktor Penyesuaian Ukuran Otak (*Atlas Scaling Factor*)	Numerik	Fitur	373	0 - 1

Berdasarkan Tabel 2, *data preprocessing* dilakukan pada atribut fitur yang tidak mengalami penghapusan atribut pada data. Hasil *data preprocessing* membuat jumlah data dan rentang nilai menjadi sama. Jumlah data pada semua atribut menjadi 373 data. Rentang nilai pada semua atribut menjadi antara 0 sampai 1.

2.3. *Percentage Split*

Percentage split merupakan proses pembagian dataset menjadi dua bagian, yaitu data pelatihan (*training data*) dan data pengujian (*testing data*) [21]. *Percentage split* bertujuan agar model dilatih pada sebagian data dan diuji pada data yang belum pernah dilihat sebelumnya.

2.4. *K-Fold Cross-validation*

K-fold cross-validation merupakan teknik validasi yang membagi data menjadi beberapa kelompok data pelatihan (*training data*) dan data pengujian (*testing data*) dengan jumlah data hampir sama [22]. *K-fold cross-validation* membagi satu kelompok data menjadi data pelatihan dan kelompok data lainnya menjadi data pengujian yang dilakukan berulang pada setiap kelompok data [23].

2.5. *ADASYN*

ADASYN (*Adaptive Synthetic Sampling*) adalah metode *oversampling* untuk menangani data tidak seimbang (*imbalanced dataset*) [24]. ADASYN digunakan untuk mengatasi ketidakseimbangan data dengan membuat data tambahan pada kelas yang jumlahnya sedikit, sehingga model dapat belajar lebih baik [25].

2.6. Algoritma CART

Algoritma CART menghasilkan pohon keputusan berdasarkan atribut. Setiap atribut memiliki nilai yang menghasilkan kemungkinan penyebab demensia dan menjadi dasar dalam membuat pohon keputusan. Menurut J. Sun *et al* [26], langkah-langkah pengerjaan algoritma CART sebagai berikut.

1. Menyiapkan dataset.
2. Menentukan *gini index* untuk seluruh atribut menggunakan Persamaan (2).

$$Gini(S) = 1 - \sum_{j=1}^{k} \left(p_j^{\,2}\right) \quad (2)$$

   Dimana S merupakan jumlah data, p_j merupakan peluang klasifikasi objek pada kelas ke-j, j merupakan indeks kelas ke-j, k merupakan jumlah kelas, dan $Gini(S)$ merupakan ukuran keberagaman kelas dari suatu partisi S.
3. Menentukan nilai *reduction in gini impurity* dengan menggunakan Persamaan (3).

$$GiniGain(A,S) = Gini(S) - \sum_{j=1}^{k} \left(\frac{|S_j|}{|S|}\right) x\, Gini(S_j) \quad (3)$$

   Dimana A merupakan atribut, S merupakan jumlah data, $|S_j|$ merupakan data sampel ke-j, $|S|$ merupakan jumlah sampel data, $Gini(S)$ merupakan ukuran keberagaman kelas dari suatu partisi S, $Gini(S_j)$ merupakan ukuran keberagaman kelas dari suatu partisi S ke-j, dan $GiniGain(A,S)$ merupakan ukuran penurunan keberagaman kelas dari suatu partisi S terhadap atribut A.
4. Membuat akar dan cabang *reduction in gini impurity* tertinggi.
5. Mengulangi langkah hingga semua variabel terpartisi.

2.7. Algoritma AdaBoost

Algoritma AdaBoost terdiri dari beberapa langkah, diawali dengan membangun model, membuat prediksi, menentukan bobot tertinggi pada klasifikasi yang salah, membangun model, dan membuat model akhir dari rata-rata setiap iterasi. Menurut Mienye *and* Y. Sun [27], langkah-langkah pengerjaan algoritma AdaBoost sebagai berikut.

1. Menentukan bobot pada tiap data menggunakan Persamaan (4).

E-ISSN 2722-7162
P-ISSN 2089-5615

$$w_i^{(t)} = \frac{1}{N}; i = 1, 2, ... N \quad (4)$$

Dimana t merupakan indeks *time step* (iterasi) ke-t, i merupakan indeks data ke-i, $w_i^{(t)}$ merupakan nilai *weight* (bobot) data ke-i (iterasi) ke-t, dan N merupakan banyak data.

2. Melatih *weak learner* $h_t(x)$.
3. Menentukan *error rate* menggunakan Persamaan (5).

$$\varepsilon_t = \frac{\sum_{i=1}^{N} w_i^{(t)} 1(h_t(x_i) \neq y_i)}{\sum_{i=1}^{N} w_i^{(t)}} \quad (5)$$

Dimana t merupakan indeks *time step* (iterasi) ke-t, ε_t merupakan *error rate time step* (iterasi) ke-t, N merupakan banyak data, $w_i^{(t)}$ merupakan nilai *weight* (bobot) data ke-i (iterasi) ke-t, dan x_i merupakan data ke-i, $h_t(x_i)$ merupakan merupakan model lemah (*weak learner*) ke-t data ke-i, dan y_i merupakan kelas data ke-i.

4. Menentukan bobot suara menggunakan Persamaan (6).

$$\alpha_t = \frac{1}{2} \ln ln \left(\frac{1-\varepsilon_t}{\varepsilon_t} \right) \quad (6)$$

Dimana t merupakan indeks *time step* (iterasi) ke-t, α_t merupakan nilai bobot suara *time step* (iterasi) ke-t, dan ε_t merupakan nilai *error rate time step* (iterasi) ke-t.

5. Melakukan pembaruan untuk bobot dengan pengamatan yang salah menggunakan Persamaan (7).

$$w_i^{(t+1)} = w_i^{(t)} e^{(\alpha_t y_i h_t(x))} \quad (7)$$

Dimana t merupakan indeks *time step* (iterasi) ke-t, i merupakan indeks data ke-i, $w_i^{(t+1)}$ merupakan nilai *weight* (bobot) data ke-i (iterasi) ke-t+1, $w_i^{(t)}$ merupakan nilai *weight* (bobot) data ke-i (iterasi) ke-t, α_t merupakan nilai bobot suara *time step* (iterasi) ke-t, y_i merupakan kelas data ke-i, dan $h_t(x)$ merupakan model lemah (*weak learner*) ke-t.

6. Melakukan prediksi pada model AdaBoost (*strong learner*) menggunakan Persamaan (8).

$$H(x) = sign(\sum_{t=1}^{T} \alpha_t h_t(x)) \quad (8)$$

Dimana $H(x)$ merupakan model AdaBoost (*strong learner*), t merupakan indeks *time step* (iterasi) ke-t, T merupakan jumlah *time step* (iterasi), α_t merupakan nilai bobot suara *time step* (iterasi) ke-t, dan $h_t(x)$ merupakan model lemah (*weak learner*) ke-t.

7. Melakukan pembaruan untuk semua bobot baru dengan menormalisasi nilainya agar jumlahnya menjadi 1. Normalisasi dilakukan dengan membagi semua bobot baru dengan jumlah semua bobot baru. Prediksi final dilakukan dengan menggabungkan semua model lemah (*weak learner*) menjadi (*strong learner)*.

2.8. Evaluasi Kinerja

Evaluasi kinerja merupakan kriteria spesifik yang digunakan untuk mengukur tingkat keberhasilan suatu pendekatan tertentu menggunakan *confusion matrix* [28]. *Confusion matrix* merupakan matriks berukuran 2×2 yang menampilkan hasil prediksi dari metode dengan hasil sebenarnya seperti pada Table 3 [29].

Tabel 3. *Confusion Matrix*

Kelas		**Nilai Aktual**	
		Positif	**Negatif**
Nilai Prediksi	Positif	*True Positive* (TP)	*False Positive* (FP)
	Negatif	*False Negative* (FN)	*True Negative* (TN)

Berdasarkan Tabel 3, *confusion matrix* terdiri dari label aktual pada baris dan label prediksi pada kolom. *Confusion matrix* mendefinisikan empat parameter yang menjadi dasar untuk menghitung metrik evaluasi kinerja, yaitu *True Positif* (*TP*), *False Positif* (*FP*), *False Negatif* (*FN*) , dan *True Negatif* (*TN*) [30]. *TP* merupakan prediksi benar dari hasil aktual benar. *FP* merupakan prediksi salah dari hasil aktual benar. *FN* merupakan prediksi salah dari hasil aktual salah. *TN* merupakan prediksi benar dari hasil aktual salah [31]. Ukuran evaluasi kinerja model dalam melakukan klasifikasi pada *confusion matrix*, yaitu akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC.

E-ISSN 2722-7162
P-ISSN 2089-5615

Akurasi merupakan jumlah dari data prediksi positif dan negatif yang benar dibagi dengan semua prediksi [32]. Proses perhitungan akurasi dapat dilihat pada Persamaan (10) [23].

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \quad (10)$$

Presisi merupakan proporsi prediksi positif yang benar terhadap semua prediksi positif [33]. Proses perhitungan presisi dapat dilihat pada Persamaan (11) [34].

$$Presisi = \frac{TP}{TP+FP} \quad (11)$$

Recall merupakan proporsi prediksi positif yang benar terhadap semua aktual positif [35]. Proses perhitungan *recall* dapat dilihat pada Persamaan (12) [36].

$$Recall = \frac{TP}{TP+FN} \quad (12)$$

F1-Score merupakan *geometric mean* antara presisi dan *recall* [32]. Proses perhitungan *f1-score* dapat dilihat pada Persamaan (13) [23].

$$F1 - Score = 2 \times \frac{Recall \times Presisi}{Recall + Presisi} \quad (13)$$

ROC-AUC (*Receiver Operating Characteristic-Area Under the Curve*) merupakan metrik evaluasi yang mengukur kemampuan model dalam membedakan setiap kelas [37].

3. HASIL DAN DISKUSI

3.1. Implementasi Algoritma CART

Algoritma CART bertujuan membentuk pohon keputusan yang mampu mengelompokkan atribut penyebab utama serta atribut pendukung yang berkontribusi terhadap terjadinya suatu kejadian. Pohon keputusan digunakan untuk mengidentifikasi faktor-faktor berpengaruh terhadap demensia. Tahapan awal implementasi algoritma CART adalah menyiapkan dataset. Dataset demensia memiliki 373 data yang terbagi menjadi 190 data *nondemented*, 146 data *demented*, dan 37 data *converted*. Tahapan selanjutnya melakukan pembagian data menggunakan metode *percentage split* dan *k-fold cross-validation*. *Percentage split* pada algoritma CART membagi data menjadi dua bagian dengan 70% data pelatihan dan 30% data pengujian. Data pelatihan memiliki 261 data yang terbagi menjadi 133 data *nondemented*, 102 data *demented*, dan 26 data *converted*. Data pelatihan mengalami ketidakseimbangan kelas, di mana jumlah sampel pada kelas *demented* dan *converted* lebih sedikit dibandingkan kelas *demented*. ADASYN digunakan untuk menyeimbangkan data pada setiap kelas. ADASYN meningkatkan jumlah data pelatihan pada algoritma CART menjadi 399 data yang terbagi menjadi 133 data *nondemented*, 137 data *demented*, dan 129 data *converted*. Parameter algoritma CART yang digunakan pada *percentage split* adalah *criterion='gini'*, *max_depth=None*, dan *min_samples_split=3*. Perhitungan algoritma CART pada *percentage split* diawali dengan menentukan nilai *gini index*. Hasil perhitungan *gini index* keseluruhan atribut sebesar 0.304 yang digunakan untuk menentukan nilai *reduction in gini impurity*. Hasil perhitungan *reduction in gini impurity* keseluruhan atribut sebesar 0.151 dengan atribut *CDR* merupakan penyebab utama demensia. Atribut *CDR* menjadi akar pohon keputusan, sedangkan atribut lainnya menjadi cabang yang menjelaskan faktor-faktor pendukung dalam klasifikasi demensia. Aturan *linguistik* yang diperoleh dari algoritma CART menggunakan *percentage split* pada klasifikasi demensia sebagai berikut.

1. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES <= 0.195) AND (ASF <= 0.264) THEN NONDEMENTED
2. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES <= 0.195) AND (ASF > 0.264) AND (EDUC <= 0.704) AND (Age <= 0.739) AND (MMSE <= 0.962) AND (Age <= 0.355) THEN CONVERTED
3. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES <= 0.195) AND (ASF > 0.264) AND (EDUC <= 0.704) AND (Age <= 0.739) AND (MMSE <= 0.962) AND (Age > 0.355) THEN NONDEMENTED
4. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES <= 0.195) AND (ASF > 0.264) AND (EDUC <= 0.704) AND (Age <= 0.739) AND (MMSE > 0.962) THEN CONVERTED
5. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES <= 0.195) AND (ASF > 0.264) AND (EDUC <= 0.704) AND (Age > 0.739) THEN NONDEMENTED
6. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES <= 0.195) AND (ASF > 0.264) AND (EDUC > 0.704) AND (Age <= 0.487) THEN NONDEMENTED
7. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES <= 0.195) AND (ASF > 0.264) AND (EDUC > 0.704) AND (Age > 0.487) AND (Age <= 0.750) THEN CONVERTED
8. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES <= 0.195) AND (ASF > 0.264) AND (EDUC > 0.704) AND (Age > 0.487) AND (Age > 0.750) THEN NONDEMENTED

E-ISSN 2722-7162
P-ISSN 2089-5615

9. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay <= 0.003) AND (MMSE <= 0.973) THEN NONDEMENTED
10. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay <= 0.003) AND (MMSE > 0.973) AND (SES <= 0.218) AND (nWBV <= 0.391) AND (nWBV <= 0.311) THEN NONDEMENTED
11. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay <= 0.003) AND (MMSE > 0.973) AND (SES <= 0.218) AND (nWBV <= 0.391) AND (nWBV > 0.311) THEN CONVERTED
12. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay <= 0.003) AND (MMSE > 0.973) AND (SES <= 0.218) AND (nWBV > 0.391) THEN NONDEMENTED
13. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay <= 0.003) AND (MMSE > 0.973) AND (SES > 0.218) AND (Age <= 0.228) AND (Age <= 0.105) THEN NONDEMENTED
14. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay <= 0.003) AND (MMSE > 0.973) AND (SES > 0.218) AND (Age <= 0.228) AND (Age > 0.105) THEN CONVERTED
15. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay <= 0.003) AND (MMSE > 0.973) AND (SES > 0.218) AND (Age > 0.228) AND (SES <= 0.363) THEN CONVERTED
16. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay <= 0.003) AND (MMSE > 0.973) AND (SES > 0.218) AND (Age > 0.228) AND (SES > 0.363) AND (ASF <= 0.495) AND (EDUC <= 0.471) THEN CONVERTED
17. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay <= 0.003) AND (MMSE > 0.973) AND (SES > 0.218) AND (Age > 0.228) AND (SES > 0.363) AND (ASF <= 0.495) AND (EDUC > 0.471) THEN NONDEMENTED
18. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay <= 0.003) AND (MMSE > 0.973) AND (SES > 0.218) AND (Age > 0.228) AND (SES > 0.363) AND (ASF > 0.495) THEN NONDEMENTED
19. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit <= 0.229) AND (SES > 0.195) AND (MR Delay > 0.003) THEN CONVERTED
20. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit > 0.229) AND (MMSE <= 0.846) THEN CONVERTED
21. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit > 0.229) AND (MMSE > 0.846) AND (MR Delay <= 0.145) THEN CONVERTED
22. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit > 0.229) AND (MMSE > 0.846) AND (MR Delay > 0.145) AND (Age <= 0.697) THEN NONDEMENTED
23. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit > 0.229) AND (MMSE > 0.846) AND (MR Delay > 0.145) AND (Age > 0.697) AND (Age <= 0.750) AND (SES <= 0.300) THEN CONVERTED
24. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit > 0.229) AND (MMSE > 0.846) AND (MR Delay > 0.145) AND (Age > 0.697) AND (Age <= 0.750) AND (SES > 0.300) THEN NONDEMENTED
25. IF (CDR <= 0.249) AND (CDR <= 0.005) AND (Visit > 0.229) AND (MMSE > 0.846) AND (MR Delay > 0.145) AND (Age > 0.697) AND (Age > 0.750) THEN NONDEMENTED
26. IF (CDR <= 0.249) AND (CDR > 0.005) THEN CONVERTED
27. IF (CDR > 0.249) AND (MR Delay <= 0.334) AND (MR Delay <= 0.257) AND (ASF <= 0.131) AND (ASF <= 0.089) THEN DEMENTED
28. IF (CDR > 0.249) AND (MR Delay <= 0.334) AND (MR Delay <= 0.257) AND (ASF <= 0.131) AND (ASF > 0.089) THEN NONDEMENTED
29. IF (CDR > 0.249) AND (MR Delay <= 0.334) AND (MR Delay <= 0.257) AND (ASF > 0.131) AND (Age <= 0.145) AND (ASF <= 0.498) THEN DEMENTED
30. IF (CDR > 0.249) AND (MR Delay <= 0.334) AND (MR Delay <= 0.257) AND (ASF > 0.131) AND (Age <= 0.145) AND (ASF > 0.498) THEN CONVERTED
31. IF (CDR > 0.249) AND (MR Delay <= 0.334) AND (MR Delay <= 0.257) AND (ASF > 0.131) AND (Age > 0.145) THEN DEMENTED
32. IF (CDR > 0.249) AND (MR Delay <= 0.334) AND (MR Delay > 0.257) AND (Age <= 0.578) AND (MMSE <= 0.993) THEN DEMENTED
33. IF (CDR > 0.249) AND (MR Delay <= 0.334) AND (MR Delay > 0.257) AND (Age <= 0.578) AND (MMSE > 0.993) THEN CONVERTED

E-ISSN 2722-7162
P-ISSN 2089-5615

34. IF (CDR > 0.249) AND (MR Delay <= 0.334) AND (MR Delay > 0.257) AND (Age > 0.578) AND (M/F <= 0.500) THEN CONVERTED
35. IF (CDR > 0.249) AND (MR Delay <= 0.334) AND (MR Delay > 0.257) AND (Age > 0.578) AND (M/F > 0.500) AND (EDUC <= 0.706) THEN DEMENTED
36. IF (CDR > 0.249) AND (MR Delay <= 0.334) AND (MR Delay > 0.257) AND (Age > 0.578) AND (M/F > 0.500) AND (EDUC > 0.706) THEN CONVERTED
37. IF (CDR > 0.249) AND (MR Delay > 0.334) AND (SES <= 0.500) AND (ASF <= 0.238) THEN DEMENTED
38. IF (CDR > 0.249) AND (MR Delay > 0.334) AND (SES <= 0.500) AND (ASF > 0.238) AND (Age <= 0.303) THEN DEMENTED
39. IF (CDR > 0.249) AND (MR Delay > 0.334) AND (SES <= 0.500) AND (ASF > 0.238) AND (Age > 0.303) AND (MMSE <= 0.814) THEN CONVERTED
40. IF (CDR > 0.249) AND (MR Delay > 0.334) AND (SES <= 0.500) AND (ASF > 0.238) AND (Age > 0.303) AND (MMSE > 0.814) THEN CONVERTED
41. IF (CDR > 0.249) AND (MR Delay > 0.334) AND (SES > 0.500) AND (MR Delay <= 0.404) THEN NONDEMENTED
42. IF (CDR > 0.249) AND (MR Delay > 0.334) AND (SES > 0.500) AND (MR Delay > 0.404) THEN DEMENTED

K-fold cross-validation mengelompokkan data menjadi beberapa kelompok secara acak dan validasi dilakukan sebanyak 10 kali. *K-fold cross-validation* pada algoritma CART menggunakan 1 kelompok data menjadi data pengujian dan 9 kelompok data lainnya menjadi data pengujian yang dilakukan berulang pada setiap kelompok data. Data pelatihan memiliki 335 data yang terbagi menjadi 171 data *nondemented*, 131 data *demented*, dan 33 data *converted*. Data pelatihan mengalami ketidakseimbangan kelas, di mana jumlah sampel pada kelas *demented* dan *converted* lebih sedikit dibandingkan kelas *demented*. ADASYN digunakan untuk menyeimbangkan data pada setiap kelas. ADASYN meningkatkan jumlah data pelatihan pada algoritma CART menjadi 508 data yang terbagi menjadi 171 data *nondemented*, 168 data *demented*, dan 169 data *converted*. Parameter algoritma CART yang digunakan pada *k-fold cross-validation* adalah *criterion='gini'*, *max_depth=5*, dan *min_samples_split=25*. Perhitungan algoritma CART pada *k-fold cross-validation* diawali dengan menentukan nilai *gini index*. Hasil perhitungan *gini index* keseluruhan atribut sebesar 0.201 yang digunakan untuk menentukan nilai *reduction in gini impurity*. Hasil perhitungan *reduction in gini impurity* keseluruhan atribut sebesar 0.058 dengan atribut *CDR* merupakan penyebab utama demensia. Atribut *CDR* menjadi akar pohon keputusan, sedangkan atribut lainnya menjadi cabang yang menjelaskan faktor-faktor pendukung dalam klasifikasi demensia. Aturan *linguistik* yang diperoleh dari algoritma CART menggunakan *k-fold cross-validation* pada klasifikasi demensia sebagai berikut.

1. IF (CDR <= 0.003) AND (Visit <= 0.237) AND (Visit <= 0.002) AND (SES <= 0.179) AND (SES <= 0.003) THEN NONDEMENTED
2. IF (CDR <= 0.003) AND (Visit <= 0.237) AND (Visit <= 0.002) AND (SES <= 0.179) AND (SES > 0.003) THEN CONVERTED
3. IF (CDR <= 0.003) AND (Visit <= 0.237) AND (Visit <= 0.002) AND (SES > 0.179) AND (MMSE <= 0.963) THEN NONDEMENTED
4. IF (CDR <= 0.003) AND (Visit <= 0.237) AND (Visit <= 0.002) AND (SES > 0.179) AND (MMSE > 0.963) THEN NONDEMENTED
5. IF (CDR <= 0.003) AND (Visit <= 0.237) AND (Visit > 0.002) THEN CONVERTED
6. IF (CDR <= 0.003) AND (Visit > 0.237) AND (MMSE <= 0.876) THEN CONVERTED
7. IF (CDR <= 0.003) AND (Visit > 0.237) AND (MMSE > 0.876) AND (Age <= 0.697) AND (eTIV <= 0.225) THEN NONDEMENTED
8. IF (CDR <= 0.003) AND (Visit > 0.237) AND (MMSE > 0.876) AND (Age <= 0.697) AND (eTIV > 0.225) THEN NONDEMENTED
9. IF (CDR <= 0.003) AND (Visit > 0.237) AND (MMSE > 0.876) AND (Age > 0.697) THEN NONDEMENTED
10. IF (CDR > 0.003) AND (MR Delay <= 0.272) AND (CDR <= 0.248) THEN CONVERTED
11. IF (CDR > 0.003) AND (MR Delay <= 0.272) AND (CDR > 0.248) AND (MR Delay <= 0.257) AND (Visit <= 0.375) THEN DEMENTED
12. IF (CDR > 0.003) AND (MR Delay <= 0.272) AND (CDR > 0.248) AND (MR Delay <= 0.257) AND (Visit > 0.375) THEN DEMENTED
13. IF (CDR > 0.003) AND (MR Delay <= 0.272) AND (CDR > 0.248) AND (MR Delay > 0.257) THEN DEMENTED

E-ISSN 2722-7162
P-ISSN 2089-5615

14. IF (CDR > 0.003) AND (MR Delay > 0.272) AND (SES <= 0.542) AND (ASF <= 0.242) THEN DEMENTED
15. IF (CDR > 0.003) AND (MR Delay > 0.272) AND (SES <= 0.542) AND (ASF > 0.242) AND (Age <= 0.307) THEN DEMENTED
16. IF (CDR > 0.003) AND (MR Delay > 0.272) AND (SES <= 0.542) AND (ASF > 0.242) AND (Age > 0.307) THEN CONVERTED
17. IF (CDR > 0.003) AND (MR Delay > 0.272) AND (SES > 0.542) AND (nWBV <= 0.693) THEN DEMENTED
18. IF (CDR > 0.003) AND (MR Delay > 0.272) AND (SES > 0.542) AND (nWBV > 0.693) THEN CONVERTED

Pengujian data pada algoritma CART menggunakan *percentage split* dan *k-fold cross-validation* memperoleh klasifikasi demensia berdasarkan atribut paling berpengaruh pada aturan *linguistik*. Hasil pengujian pada algoritma CART ditampilkan dalam bentuk *confusion matrix* yang digunakan untuk menentukan nilai akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC. *Confusion matrix* algoritma CART menggunakan *percentage split* dan *k-fold cross-validation* dapat dilihat pada Tabel 4.

Tabel 4. *Confusion Matrix* pada Algoritma CART menggunakan *Percentage Split* dan *K-Fold Cross-validation*

Kelas		**Nilai Aktual**		
		Converted	***Demented***	***Nondemented***
Nilai Prediksi dengan *Percentage Split*	*Converted*	3	2	6
	Demented	3	41	0
	Nondemented	4	0	53
Nilai Prediksi dengan *K-Fold Cross-validation*	*Converted*	16	6	15
	Demented	7	139	0
	Nondemented	18	2	170

Berdasarkan Tabel 4, hasil *confusion matrix* menunjukkan performa algoritma CART menggunakan *percentage split* dan *k-fold cross-validation* dalam mengklasifikasikan tiga kelas, yaitu *converted*, *demented*, dan *nondemented*. Algoritma CART menggunakan *percentage split* memprediksi 112 data. Algoritma CART menggunakan *percentage split* memprediksi 97 data benar yang terbagi menjadi 3 data *converted*, 41 data *demented*, dan 53 data *nondemented*. Algoritma CART menggunakan *percentage split* memprediksi 15 data salah yang terbagi menjadi 7 data *converted* yang diprediksi sebagai 3 data *demented* dan 4 data *nondemented*, 2 data demented yang diprediksi sebagai *converted*, serta 6 data *nondemented* yang diprediksi sebagai *converted*. Algoritma CART menggunakan *k-fold cross-validation* memprediksi 373 data. Algoritma CART menggunakan *k-fold cross-validation* memprediksi 325 data benar yang terbagi menjadi 16 data *converted*, 139 data *demented*, dan 170 data *nondemented*. Algoritma CART menggunakan *k-fold cross-validation* memprediksi 48 data salah yang terbagi menjadi 25 data *converted* yang diprediksi sebagai 7 data *demented* dan 18 data *nondemented*, 8 data demented yang diprediksi sebagai 6 data *converted* dan 2 data *nondemented*, serta 15 data *nondemented* yang diprediksi sebagai *converted*.

3.2. Implementasi Algoritma AdaBoost

Implementasi algoritma AdaBoost diawali dengan menyiapkan dataset. Dataset demensia memiliki 373 data yang terbagi menjadi 190 data *nondemented*, 146 data *demented*, dan 37 data *converted*. Tahapan selanjutnya melakukan pembagian data menggunakan metode *percentage split* dan *k-fold cross-validation*. *Percentage split* pada algoritma AdaBoost membagi data menjadi dua bagian dengan 70% data pelatihan dan 30% data pengujian. Data pelatihan memiliki 261 data yang terbagi menjadi 133 data *nondemented*, 102 data *demented*, dan 26 data *converted*. Data pelatihan mengalami ketidakseimbangan kelas, di mana jumlah sampel pada kelas *demented* dan *converted* lebih sedikit dibandingkan kelas *demented*. ADASYN digunakan untuk menyeimbangkan data pada setiap kelas. ADASYN meningkatkan jumlah data pelatihan pada algoritma AdaBoost menjadi 399 data yang terbagi menjadi 133 data *nondemented*, 137 data *demented*, dan 129 data *converted*. Parameter AdaBoost yang digunakan pada *percentage split* adalah *n_estimators=200* dan *learning_rate=0.5*. *K-fold cross-validation* mengelompokkan data menjadi beberapa kelompok secara acak dan validasi dilakukan sebanyak 10 kali. *K-fold cross-validation* pada algoritma AdaBoost menggunakan 1 kelompok data menjadi data pengujian dan 9 kelompok data lainnya menjadi data pengujian yang dilakukan berulang pada setiap kelompok data. Data pelatihan memiliki 335 data yang terbagi menjadi 171 data *nondemented*, 131 data *demented*, dan 33 data *converted*. Data pelatihan mengalami ketidakseimbangan kelas, di mana jumlah sampel pada kelas *demented* dan *converted* lebih sedikit dibandingkan kelas *demented*. ADASYN digunakan untuk menyeimbangkan data pada setiap kelas. ADASYN meningkatkan

E-ISSN 2722-7162
P-ISSN 2089-5615

jumlah data pelatihan pada algoritma AdaBoost menjadi 508 data yang terbagi menjadi 171 data *nondemented*, 168 data *demented*, dan 169 data *converted*. Parameter AdaBoost yang digunakan pada *k-fold cross-validation* adalah *n_estimators=100* dan *learning_rate=0.5*. Pengujian data pada algoritma AdaBoost menggunakan *percentage split* dan *k-fold cross-validation* memperoleh klasifikasi demensia berdasarkan atribut paling berpengaruh pada aturan *linguistik*. Hasil pengujian pada algoritma AdaBoost ditampilkan dalam bentuk *confusion matrix* yang digunakan untuk menentukan nilai akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC. *Confusion matrix* algoritma AdaBoost menggunakan *percentage split* dan *k-fold cross-validation* dapat dilihat pada Tabel 5.

Tabel 5. *Confusion Matrix* pada Algoritma AdaBoost menggunakan *Percentage Split* dan *K-Fold Cross-validation*

Kelas		**Nilai Aktual**		
		Converted	***Demented***	***Nondemented***
Nilai Prediksi dengan *Percentage split*	*Converted*	5	2	4
	Demented	3	41	0
	Nondemented	2	0	55
Nilai Prediksi dengan *K-Fold Cross-validation*	*Converted*	17	8	12
	Demented	4	142	0
	Nondemented	2	2	186

Berdasarkan Tabel 5, hasil *confusion matrix* menunjukkan performa algoritma AdaBoost menggunakan *percentage split* dan *k-fold cross-validation* dalam mengklasifikasikan tiga kelas, yaitu *converted*, *demented*, dan *nondemented*. Algoritma AdaBoost menggunakan *percentage split* memprediksi 112 data. Algoritma AdaBoost menggunakan *percentage split* memprediksi 101 data benar yang terbagi menjadi 5 data *converted*, 41 data *demented*, dan 55 data *nondemented*. Algoritma AdaBoost menggunakan *percentage split* memprediksi 11 data salah yang terbagi menjadi 5 data *converted* yang diprediksi sebagai 3 data *demented* dan 2 data *nondemented*, 2 data demented yang diprediksi sebagai *converted*, serta 4 data *nondemented* yang diprediksi sebagai *converted*. Algoritma AdaBoost menggunakan *k-fold cross-validation* memprediksi 373 data. Algoritma AdaBoost menggunakan *k-fold cross-validation* memprediksi 345 data benar yang terbagi menjadi 17 data *converted*, 142 data *demented*, dan 186 data *nondemented*. Algoritma AdaBoost menggunakan *k-fold cross-validation* memprediksi 28 data salah yang terbagi menjadi 6 data *converted* yang diprediksi sebagai 4 data *demented* dan 2 data *nondemented*, 10 data demented yang diprediksi sebagai 8 data *converted* dan 2 data *nondemented*, serta 12 data *nondemented* yang diprediksi sebagai *converted*.

3.3. Perbandingan Kedua Algoritma

Perbandingan kedua algoritma dilakukan untuk mengevaluasi kinerja CART dan AdaBoost pada klasifikasi demensia. Evaluasi kinerja dilakukan untuk mengukur hasil pengujian data berdasarkan nilai akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC yang diperoleh dari penerapan algoritma CART dan AdaBoost menggunakan *percentage split* dan *k-fold cross-validation*.

3.3.1. Perbandingan Hasil *Percentage Split*

Perbandingan hasil kinerja algoritma CART dan AdaBoost menggunakan *percentage split* dilakukan untuk menilai performa model berdasarkan metrik akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC. Hasil pengujian menggunakan *percentage split* bertujuan untuk menilai algoritma yang paling efektif digunakan pada klasifikasi demensia. Perbandingan hasil evaluasi kinerja pada algoritma CART dan AdaBoost menggunakan *percentage split* dapat dilihat pada Tabel 6.

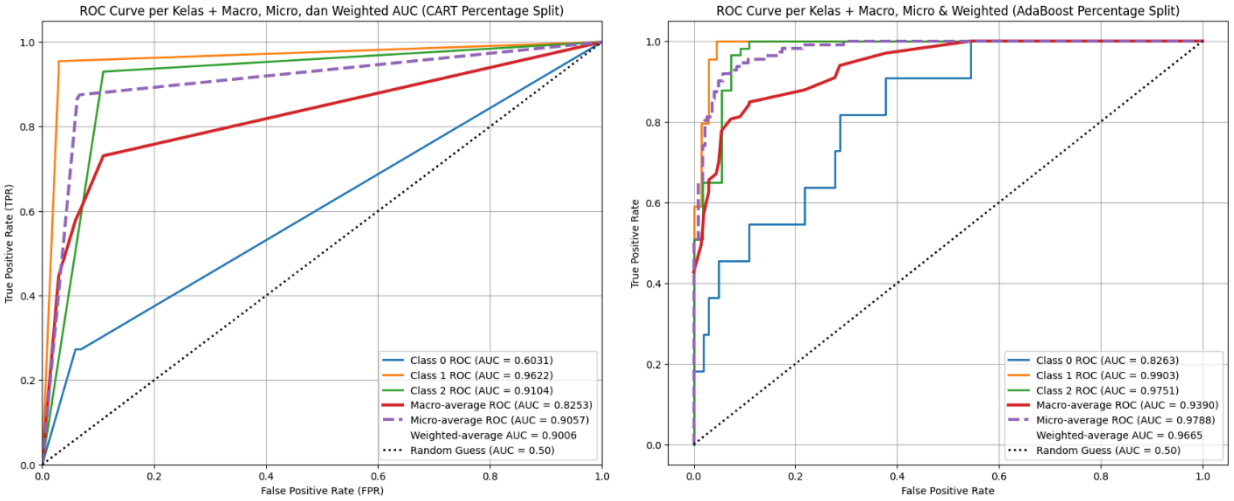
Tabel 6. Perbandingan Hasil Evaluasi Kinerja pada Algoritma CART dan AdaBoost menggunakan *Percentage Split*

Algoritma	**Akurasi**	**Presisi**	***Recall***	***F1-Score***	**ROC-AUC**
CART	86.61%	86.12%	86.61%	86.34%	90.06%
AdaBoost	90.18%	89.81%	90.17%	89.96%	96.65%

Berdasarkan Tabel 6, hasil evaluasi kinerja pada kedua algoritma menggunakan *percentage split* menunjukkan algoritma AdaBoost memiliki nilai tertinggi untuk akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC masing-masing sebesar 90.18%, 89.81%, 90.17%, 89.96%, dan 96.65%. AdaBoost menunjukkan performa yang lebih baik dari CART menggunakan *percentage split* dengan selisih nilai evaluasi diantara kedua algoritma 3% hingga 6%. Hasil perbandingan *percentage split* menunjukkan bahwa AdaBoost memberikan prediksi yang lebih baik dari CART.

E-ISSN 2722-7162
P-ISSN 2089-5615

Evaluasi kinerja lainnya adalah mengukur *Receiver Operating Characteristic* (ROC) dan *Area Under the Curve* (AUC). Perbandingan grafik ROC-AUC pada Algoritma CART dan AdaBoost menggunakan *percentage split* dapat dilihat pada Gambar 2.



Gambar 2. Perbandingan Grafik ROC-AUC pada Algoritma CART dan AdaBoost menggunakan *Percentage Split*

Berdasarkan Gambar 2, perbandingan grafik ROC-AUC pada kedua algoritma menggunakan *percentage split* menunjukkan algoritma AdaBoost mencapai hasil terbaik AUC pada *weighted-average*, *micro-average*, *macro-average*, kelas 0, kelas 1, kelas 2 masing-masing sebesar 96.65%, 97.88%, 93.90%, 82.63%, 99.03%, dan 97.51%. AdaBoost menunjukkan performa yang lebih baik dari CART menggunakan *percentage split* dengan selisih nilai setiap AUC diantara kedua algoritma 2% hingga 22%. Hasil ini menunjukkan bahwa algoritma AdaBoost mampu memperbaiki kelemahan CART dalam menghadapi data menyimpang dan meningkatkan stabilitas prediksi melalui *ensemble weighting*.

3.3.2. Perbandingan Hasil *K-Fold Cross-validation*

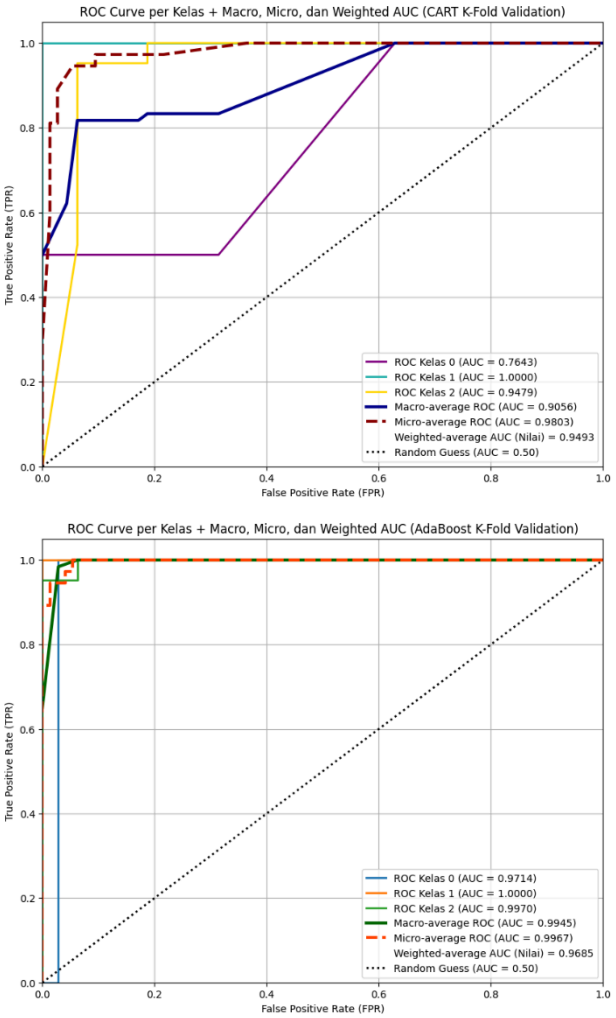
Perbandingan hasil kinerja algoritma CART dan AdaBoost menggunakan *k-fold cross-validation* dilakukan untuk menilai performa model berdasarkan metrik akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC. Hasil pengujian menggunakan *k-fold cross-validation* bertujuan untuk menilai algoritma yang paling efektif digunakan pada klasifikasi demensia. Perbandingan hasil evaluasi kinerja pada algoritma CART dan AdaBoost menggunakan *k-fold cross-validation* dapat dilihat pada Tabel 7.

Tabel 7. Perbandingan Hasil Evaluasi Kinerja pada Algoritma CART dan AdaBoost menggunakan *K-Fold Cross-validation*

Algoritma	Akurasi	Presisi	*Recall*	*F1-Score*	ROC-AUC
CART	87.15%	88.20%	87.15%	87.18%	94.93%
AdaBoost	92.52%	92.11%	92.52%	91.46%	96.85%

Berdasarkan Tabel 7, hasil evaluasi kinerja pada kedua algoritma menggunakan *k-fold cross-validation* menunjukkan algoritma AdaBoost memiliki nilai tertinggi untuk akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC masing-masing sebesar 92.52%, 92.11%, 92.52%, 91.46%, dan 96.85%. AdaBoost menunjukkan performa yang lebih baik dari CART menggunakan *k-fold cross-validation* dengan selisih nilai evaluasi diantara kedua algoritma 1% hingga 5%. Hasil perbandingan *k-fold cross-validation* menunjukkan bahwa AdaBoost memberikan prediksi yang lebih baik dari CART. Perbandingan grafik ROC-AUC pada Algoritma CART dan AdaBoost menggunakan *k-fold cross-validation* dapat dilihat pada Gambar 3.

E-ISSN 2722-7162
P-ISSN 2089-5615



Gambar 3. Perbandingan Grafik ROC-AUC pada Algoritma CART dan AdaBoost menggunakan *K-Fold Cross-Validation*

Berdasarkan Gambar 3, perbandingan grafik ROC-AUC pada kedua algoritma menggunakan *k-fold cross-validation* menunjukkan algoritma AdaBoost mencapai hasil terbaik AUC pada *weighted-average*, *micro-average*, *macro-average*, kelas 0, kelas 1, kelas 2 masing-masing sebesar 96.85%, 99.67%, 99.45%, 99.70%, 100%, dan 97.14%. AdaBoost menunjukkan performa yang lebih baik dari CART menggunakan *k-fold cross-validation* dengan selisih nilai setiap AUC diantara kedua algoritma 0% hingga 20%. Hasil ini menunjukkan bahwa algoritma AdaBoost mampu memperbaiki kelemahan CART dalam menghadapi data menyimpang dan meningkatkan stabilitas prediksi melalui *ensemble weighting*.

3.3.3. Analisis dan Interpretasi Hasil

Analisis dan interpretasi hasil dilakukan perbandingan evaluasi kinerja model pada klasifikasi demensia dengan penelitian terdahulu. Perbandingan hasil evaluasi kinerja pada metode yang diusulkan dengan penelitian terdahulu dapat dilihat pada Tabel 8.

Tabel 8. Perbandingan Hasil Evaluasi Kinerja pada Metode yang Diusulkan dengan Penelitian Terdahulu

Metode	Hasil (%)				
	Akurasi	Presisi	*Recall*	*F1-Score*	ROC-AUC
SVM (Neelaveni *and* Devasana, 2020) [38]	85	-	-	-	-
Random Forest (Vuddanti *et al*., 2024) [39]	90.6	-	-	-	-
SVM-*Fine-Tuning* (Antor *et al*., 2024) [40]	92	-	91.89	-	91.99
XGBoost (Kavitha *et al*., 2022) [41]	85.92	85	83	85	-

E-ISSN 2722-7162
P-ISSN 2089-5615

AdaBoost-ADASYN-*K-Fold Cross-validation* (**Penelitian yang diusulkan**)	**92.52**	**92.11**	**92.52**	**91.46**	**96.85**

Berdasarkan Tabel 8, perbandingan hasil kinerja dari metode yang diusulkan dengan beberapa metode lain menggunakan dataset OASIS pada klasifikasi demensia. Penelitian (Neelaveni *and* Devasana, 2020; Vuddanti *et al*., 2024) [38], [39] hanya menghitung nilai evalusi performa akurasi. Penelitian (Kavitha *et al*., 2022) [41] tidak menghitung nilai evaluasi performa ROC-AUC. Penelitian (Antor *et al*., 2024) [40] menghasilkan nilai akurasi dan *recall* paling baik dibandingkan penelitian terdahulu lainnya, namun tidak mengukur presisi dan *f1-score*. Hasil akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC pada metode yang diusulkan dalam penelitian ini menunjukkan kinerja yang lebih unggul dibandingkan metode yang digunakan pada penelitian lain, yaitu diatas 90%. Hasil akurasi menunjukkan kinerja algoritma AdaBoost yang sangat baik dalam memprediksi seluruh demensia dengan benar. Hasil presisi menunjukkan kinerja algoritma AdaBoost yang sangat baik dalam memprediksi demensia dengan tingkat kesalahan yang rendah. Hasil *recall* menunjukkan kinerja algoritma AdaBoost yang sangat baik dalam memprediksi sebagian besar demensia dengan benar. Hasil *f1-score* mengindikasikan bahwa algoritma AdaBoost memiliki presisi dan *recall* yang seimbang. Hasil ROC-AUC menunjukkan kinerja algoritma AdaBoost yang sangat baik dalam membedakan tiga kelas demensia, yaitu *nondemented*, *demented*, dan *converted*.

4. KESIMPULAN

Penelitian ini membandingkan algoritma CART dan AdaBoost pada klasifikasi demensia. Hasil evaluasi kinerja pada algoritma AdaBoost menggunakan ADASYN dan *k-fold cross-validation* memiliki nilai tertinggi untuk akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC. Hasil kinerja algoritma AdaBoost berdasarkan pada akurasi, presisi, *recall*, *f1-score*, dan ROC-AUC semuanya di atas 90%. Secara keseluruhan pengukuran menunjukkan performa AdaBoost pada klasifikasi demensia lebih unggul dibandingkan metode yang digunakan pada penelitian lain. Hasil penelitian menunjukkan keunggulan penggunaan *ensemble learning* dalam deteksi dini demensia berbasis data kesehatan. Penelitian ini dapat dikembangkan lebih lanjut dengan memperluas dataset, menambahkan fitur MRI, dan menguji algoritma berbasis *deep learning* untuk peningkatan performa.

REFERENSI

[1] WHO, *A Blueprint for Dementia Research*. Geneva, 2022. [Online]. Available: https://www.who.int/publications/i/item/9789240058248
[2] WHO, "Dementia in Refugees and Migrants: Epidemiology, Public Health Implications and Global Responses," WHO, Geneva, 2025.
[3] D. D. Shankar, A. S. Azhakath, N. Khalil, S. J, M. T, and S. K, "Data mining for cyber biosecurity risk management – A comprehensive review," *Comput. Secur.*, vol. 137, p. 103627, Feb. 2024, doi: 10.1016/j.cose.2023.103627.
[4] R. Shi, X. Leng, Y. Wu, S. Zhu, X. Cai, and X. Lu, "Machine learning regression algorithms to predict short-term efficacy after anti-VEGF treatment in diabetic macular edema based on real-world data," *Sci. Rep.*, vol. 13, no. 1, p. 18746, Oct. 2023, doi: 10.1038/s41598-023-46021-2.
[5] M. Khandelwal, D. J. Armaghani, R. S. Faradonbeh, M. Yellishetty, M. Z. A. Majid, and M. Monjezi, "Classification and Regression Tree technique in estimating peak particle velocity caused by blasting," *Eng. Comput.*, vol. 33, no. 1, pp. 45–53, Jan. 2017, doi: 10.1007/s00366-016-0455-0.
[6] F. Van Der Steen, F. Vink, and H. Kaya, "Privacy Constrained Fairness Estimation for Decision Trees," *Appl. Intell.*, vol. 55, no. 308, pp. 1–27, 2025, doi: 10.1007/s10489-024-05953-6.
[7] R. B. Diwate, R. Ghosh, R. Jha, I. Sagar, and S. K. Singh, "Dementia prediction using OASIS data for alzheimer's research," in *2021 International Conference on Artificial Intelligence and Machine Vision (AIMV)*, 2021, pp. 1–7. doi: 10.1109/AIMV53313.2021.9670900.
[8] A. Abedinia and V. Seydi, "Building Semi-Supervised Decision Trees with Semi-CART Algorithm," *Int. J. Mach. Learn. Cybern.*, vol. 15, no. 10, pp. 4493–4510, 2024, doi: 10.1007/s13042-024-02161-z.
[9] A. Subasi, "Machine learning techniques," in *Practical Machine Learning for Data Analysis Using Python*, A. B. T.-P. M. L. for D. A. U. P. Subasi, Ed., Elsevier, 2020, pp. 91–202. doi: 10.1016/B978-0-12-821379-7.00003-5.
[10] B. Jean-Marc and O. Lafitte, "Combining weak classifiers: a logical analysis," in *2021 23rd International Symposium on Symbolic and Numeric Algorithms for Scientific Computing (SYNASC)*, IEEE, Dec. 2021, pp. 178–181. doi: 10.1109/SYNASC54541.2021.00038.
[11] M. M. Høgsgaard, K. G. Larsen, and M. Ritzert, "AdaBoost is Not an Optimal Weak to Strong Learner," *ICML'23 Proc. 40th Int. Conf. Mach. Learn.*, vol. 532, pp. 13118–13140, 2023, doi: 10.5555/3618408.3618940.

E-ISSN 2722-7162
P-ISSN 2089-5615

[12] S. Dhakal, S. Azam, K. M. Hasib, A. Karim, M. Jonkman, and A. S. M. F. Al Haque, "Dementia prediction using machine learning," *Procedia Comput. Sci.*, vol. 219, pp. 1297–1308, 2023, doi: 10.1016/j.procs.2023.01.414.

[13] J. Boysen, "MRI and Alzheimers," Kaggle. Accessed: Nov. 16, 2024. [Online]. Available: https://www.kaggle.com/datasets/jboysen/mri-and-alzheimers?select=oasis_longitudinal.csv)

[14] L. A. Kumar, "Predicting Parkinson's: Analyzing Patterns with Data and Analytics," in *Predictive Analytics using MATLAB® for Biomedical Applications*, L. A. B. T.-P. A. using M. for B. A. Kumar, Ed., Academic Press, 2025, pp. 153–185. doi: https://doi.org/10.1016/B978-0-443-29888-2.00005-2.

[15] M. Alamuri, B. Surampudi, and A. Negi, "Multi Dimensional Deep Encoding for Categorical Feature Space," *ICCPR '24 Proc. 2024 13th Int. Conf. Comput. Pattern Recognit.*, pp. 120–128, 2025, doi: 10.1145/3704323.3704378.

[16] Y. Liu, Y. Cao, Z. Zeng, B. Yu, and M. Cai, "CAMI: A Missing Value Imputation Method Based on Causal Discovery and Self-attention," in *Advances and Trends in Artificial Intelligence. Theory and Applications*, H. Fujita, Y. Watanobe, M. Ali, and Y. Wang, Eds., Singapore: Springer Nature Singapore, 2025, pp. 342–353.

[17] S. Tahvili and L. Hatvani, "Benefits, Results, and Challenges of Artificial Intelligence," in *Uncertainty, Computational Techniques, and Decision Intelligence*, S. Tahvili and L. B. T.-A. I. M. for O. of the S. T. P. Hatvani, Eds., Academic Press, 2022, pp. 161–172. doi: https://doi.org/10.1016/B978-0-32-391913-5.00017-8.

[18] C. Alex, M. Herzberg, and L. Mader, "Normalized Data may have Limitations in Determining the Reliability of MVC Measurements," *Sci. Rep.*, vol. 15, pp. 1–10, 2025.

[19] A. Das, "Linear Regression for Balancing Feature Relevance and Redundancy with Optimization," *CODS-COMAD '24 Proc. 8th Int. Conf. Data Sci. Manag. Data (12th ACM IKDD CODS 30th COMAD)*, pp. 314–316, 2025, doi: 10.1145/3703323.3703700.

[20] A. T. Keleko, B. Kamsu-Foguem, R. H. Ngouna, and A. Tongne, "Health condition monitoring of a complex hydraulic system using Deep Neural Network and DeepSHAP explainable XAI," *Adv. Eng. Softw.*, vol. 175, p. 103339, Jan. 2023, doi: 10.1016/j.advengsoft.2022.103339.

[21] A. A. Shujaaddeen, F. Mutaher Ba -Alwi, A. T. Zahary, and A. Sultan Alhegami, "A model for measuring the effect of splitting data method on the efficiency of machine learning models: A comparative study," in *2024 4th International Conference on Emerging Smart Technologies and Applications (eSmarTA)*, IEEE, Aug. 2024, pp. 1–13. doi: 10.1109/eSmarTA62850.2024.10639022.

[22] V. H. Kamble and M. P. Dale, "Machine learning approach for longitudinal face recognition of children," in *Machine Learning for Biometrics*, P. P. Sarangi, M. Panda, S. Mishra, B. S. P. Mishra, and B. B. T.-M. L. for B. Majhi, Eds., Elsevier, 2022, pp. 1–27. doi: 10.1016/B978-0-323-85209-8.00011-0.

[23] O. Rainio, J. Teuho, and R. Klén, "Evaluation Metrics and Statistical Tests for Machine Learning," *Sci. Rep.*, vol. 14, no. 1, pp. 1–14, 2024, doi: 10.1038/s41598-024-56706-x.

[24] M. U. Safder, S. S. Naveed, K. Khurshid, and A. Salman, "Optimizing Imbalanced Learning with Genetic Algorithm," *Sci. Rep.*, vol. 15, pp. 1–30, 2025.

[25] M. Kannan, D. Umamaheswari, B. Manimekala, I. P. S. Mary, and P. M. Savitha, "An Enhancement of Machine Learning Model Performance in Disease Prediction with Synthetic Data Generation," *Sci. Rep.*, vol. 15, pp. 1–21, 2025.

[26] J. Sun, N. Jiang, G. Sun, and W. Huang, "Analysis of CART algorithms in data mining," in *2023 2nd International Conference on Machine Learning, Cloud Computing and Intelligent Mining (MLCCIM)*, IEEE, Jul. 2023, pp. 548–553. doi: 10.1109/MLCCIM60412.2023.00088.

[27] I. D. Mienye and Y. Sun, "A survey of Ensemble Learning: Concepts, algorithms, applications, and prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.

[28] H. A. Çetin, E. Do, and E. Tüzün, "Science of Computer Programming A Review of Code Reviewer Recommendation Studies : Challenges and Future Directions," *Sci. Comput. Program.*, vol. 208, pp. 1–21, 2021, doi: 10.1016/j.scico.2021.102652.

[29] I. M. De Diego, A. R. Redondo, R. R. Fernández, J. Navarro, and J. M. Moguerza, "General performance score for classification problems," *Appl. Intell.*, vol. 52, no. 10, pp. 12049–12063, 2022, doi: 10.1007/s10489-021-03041-7.

[30] J. Han, J. Pei, and H. Tong, "Classification: Basic Concepts and Methods," in *Data Mining (Fourth edition)*, J. Han, J. Pei, and H. B. T.-D. M. (Fourth E. Tong, Eds., Morgan Kaufmann, 2024, pp. 239–305. doi: https://doi.org/10.1016/B978-0-12-811760-6.00016-3.

[31] A. Pandey and D. K. Vishwakarma, "Progress, achievements, and challenges in multimodal sentiment analysis using deep learning: A survey," *Appl. Soft Comput.*, vol. 152, p. 111206, Feb. 2024, doi: 10.1016/j.asoc.2023.111206.

[32] J. S. Aguilar-Ruiz and M. Michalak, "Classification performance assessment for imbalanced multiclass data,"

E-ISSN 2722-7162
P-ISSN 2089-5615

Sci. Rep., vol. 14, no. 1, pp. 1–10, 2024, doi: 10.1038/s41598-024-61365-z.

[33] A. Kulkarni, D. Chong, and F. A. Batarseh, "Foundations of data imbalance and solutions for a data democracy," in *Data democracy*, F. A. Batarseh and R. B. T.-D. D. Yang, Eds., Academic Press, 2020, pp. 83–106. doi: 10.1016/B978-0-12-818366-3.00005-8.

[34] A. Subasi, "Introduction," in *Practical machine learning for data analysis using python*, A. B. T.-P. M. L. for D. A. U. P. Subasi, Ed., Academic Press, 2020, pp. 1–26. doi: 10.1016/B978-0-12-821379-7.00001-1.

[35] M. Deprez and E. C. Robinson, "Classification," in *Machine learning for biomedical applications*, M. Deprez and E. C. B. T.-M. L. for B. A. Robinson, Eds., Academic Press, 2024, pp. 87–104. doi: 10.1016/B978-0-12-822904-0.00009-1.

[36] P. Khakhar and R. K. Dubey, "The integrity of machine learning algorithms against software defect prediction," in *Artificial intelligence and machine learning for EDGE computing*, R. Pandey, S. K. Khatri, N. kumar Singh, and P. B. T.-A. I. and M. L. for E. C. Verma, Eds., Academic Press, 2022, pp. 65–74. doi: 10.1016/B978-0-12-824054-0.00027-7.

[37] E. Richardson, R. Trevizani, J. A. Greenbaum, and H. Carter, "Article The Receiver Operating Characteristic Curve Accurately Assesses Imbalanced Datasets The Receiver Operating Characteristic Curve Accurately Assesses Imbalanced Datasets," *Patterns*, vol. 5, no. 6, pp. 1–11, 2024, doi: 10.1016/j.patter.2024.100994.

[38] J. Neelaveni and M. S. G. Devasana, "Alzheimer Disease Prediction using Machine Learning Algorithms," *2020 6th Int. Conf. Adv. Comput. Commun. Syst.*, pp. 101–104, 2020, doi: 10.1109/ICACCS48705.2020.9074248.

[39] S. Vuddanti, N. Yasmin, L. Dishasri, N. Somanath, and Y. Prasanth, "Predictive Diagnosis of Alzheimer's Disease Using Machine Learning," in *2024 2nd International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, 2024, pp. 928–934. doi: 10.1109/ICSCSS60660.2024.10625639.

[40] M. B. Antor *et al.*, "A Comparative Analysis of Machine Learning Algorithms to Predict Alzheimer's Disease," *J. Healthc. Eng.*, vol. 2021, pp. 1–12, 2024, doi: 10.1155/2021/9917919.

[41] C. Kavitha, V. Mani, S. R. Srividhya, and O. I. Khalaf, "Early-Stage Alzheimer's Disease Prediction Using Machine Learning Models," *Front. Public Heal.*, vol. 10, pp. 1–13, 2022, doi: 10.3389/fpubh.2022.853294.