KESATRIA: Jurnal Penerapan Sistem Informasi (Komputer & Manajemen)
Terakreditasi Nomor 204/E/KPT/2022 | Vol. 4, No. 4, Oktober (2023), pp. 916-924

Prediksi Penyakit Stroke Menggunakan Metode Random Forest

Priyo Wahyu Setiyo Aji[1], Suprianto[2], Rohman Dijaya[3]
[1,2,3]Fakultas Sains dan Teknologi, Informatika, Universitas Muhammadiyah Sidoarjo, Sidoarjo, Indonesia
E-mail: [1]setiyoaji1010@gmail.com, [2]suprianto@umsida.ac.id, [3]rohman.dijaya@umsida.ac.id

Abstract

Stroke is a medical condition classified under cerebrovascular diseases, which involve disruptions in the blood vessels of the brain. In a stroke, there is a blockage or rupture of brain blood vessels, leading to an interruption in the blood supply to specific brain regions. This can result in brain damage and symptoms related to brain functions, such as speech impairments, movements, and other functions. Nowadays, technology is advancing rapidly, greatly benefiting the medical community. One example is the development of artificial intelligence-powered programs for stroke detection. In this research, data was sourced from Kaggle.com, and the researchers utilized the random forest machine learning method. Random Forest combines independent decision trees originating from the same distribution, where the final prediction outcome is determined through a voting process. This research involved several stages, including preprocessing, processing, and evaluation. The research yielded an accuracy of 99%.

Keywords: *Classification; Machine Learning; Random Forest; Stroke*

Abstract

Stroke adalah suatu kondisi medis yang termasuk dalam kategori penyakit cerebrovaskular, yaitu gangguan yang terjadi pada pembuluh darah di otak. Pada stroke, terjadi penyumbatan atau pecahnya pembuluh darah otak, yang mengakibatkan terganggunya pasokan darah ke bagian otak tertentu. Hal ini dapat menyebabkan kerusakan otak dan gejala yang berkaitan dengan fungsi otak, seperti gangguan bicara, gerakan, dan fungsi lainnya. Saat ini teknologi semakin berkembang. Komunitas medis sangat terbantu dengan adanya perkembangan teknologi. Salah satunya adalah program yang dapat digunakan untuk deteksi penyakit stroke dengan kecerdasan buatan. Pada penelitian ini, data yang digunakan berasal dari website Kaggle.com dan peneliti menggunakan salah satu metode machine learning yaitu random forest. Random Forest adalah gabungan pohon keputusan independen yang berasal dari distribusi yang sama, di mana hasil prediksi akhir diperoleh melalui proses voting. Beberapa tahapan dilakukan dalam penelitian ini diantaranya adalah tahapan preprocessing, processing dan evaluasi. Hasil dari penelitian ini yaitu akurasi sebesar 99%.

Keywords: *Klasifikasi; Machine Learning; Random Forest; Stroke*

1. Pendahuluan

WHO mendefinisikan stroke sebagai kerusakan fungsi otak secara tiba-tiba yang terjadi dalam waktu 24 jam atau lebih [1]. Strok merupakan penyakit cerebrovaskular atau cedera otak dimana pembuluh darah tersumbat dan membatasi suplai darah ke otak. Stroke merupakan penyakit neurologis terbanyak yang mengakibatkan masalah kesehatan serius dan dampaknya adalah penderita mengalami kecacatan hingga kematian [2]. Saat ini teknologi semakin berkembang. Komunitas medis sangat terbantu dengan perkembangan teknologi. Salah satunya adalah program yang dapat digunakan untuk mendeteksi

ISSN: 2720-992X

penyakit stroke menggunakan kecerdasan buatan. Adapun bidang kecerdasan yang dapat digunakan adalah machine learning.

Machine learning adalah salah satu cabang dari kecerdasan buatan yang dikembangkan agar mesin dapat belajar secara otomatis. Machine learning berfokus pada analisis data untuk menemukan hubungan antar input dan output yang diinginkan. Salah satu algoritma machine learning yang sangat terkenal adalah random forest. Random forest merupakan gabungan dari pohon klasifikasi yang bekerja secara independen dan berasal dari distribusi yang sama, yang kemudian menghasilkan hasil akhir melalui proses pemungutan voting [3]. Pada penelitian ini, peneliti membangun sebuah sistem yang dapat digunakan untuk memprediksi penyakit stroke berdasarkan data historis yang telah dikumpulkan sebelumnya. Adapun metode yang digunakan pada penelitian ini adalah metode random forest.

Penelitian mengenai penyakit stroke sebelumnya telah dilakukan dengan menggunakan berbagai data dan metode. Penelitian tersebut dilakukan oleh Firman Akbar, Hanif Wira Saputra, Adhitya Karel Maulaya, Muhammad Fikri Hidayat, Rahmaddeni dengan judul “Implementasi Algoritma Decision Tree C4.5 dan Support Vector Regression untuk Prediksi Penyakit Stroke” pada tahun 2022. Hasil dari laporan ini dianalisis dengan menggunakan algoritma data mining Decision Tree C4.5 dan Support Vector Regression. Tujuannya adalah untuk mengidentifikasi individu yang mungkin menderita stroke berdasarkan variabel yang telah ditemukan. Dari hasil analisis, terlihat bahwa algoritma Decision Tree C4.5 dengan rasio 70:30 menghasilkan tingkat kesalahan sebesar 0,235. Sementara itu, algoritma Support Vector Regression dengan perbandingan yang sama menghasilkan tingkat kesalahan sebesar 0,399. Penggunaan algoritma Decision Tree C4.5 juga memberikan hasil berupa grafik pohon keputusan yang menunjukkan jalur-jalur prediksi [4].

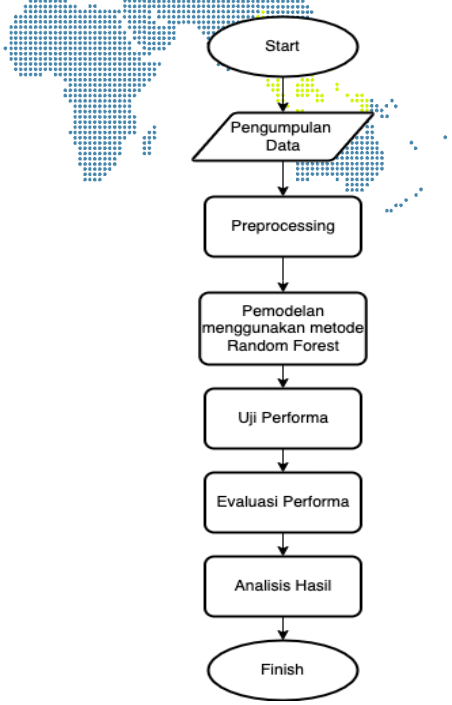
Pada tahun 2022, dilakukan penelitian oleh Ulfa Amelia, Jamaludin Indra, dan Anis Fitri Nur Masruriyah berjudul "Implementasi Algoritma Support Vector Machine (SVM) untuk Prediksi Penyakit Stroke dengan Atribut Berpengaruh". Penelitian ini menggunakan Algoritma Support Vector Machine (SVM) untuk mengklasifikasikan dataset dengan menerapkan metode Confusion Matrix guna memprediksi kemungkinan adanya penyakit stroke. Uji coba algoritma SVM dilakukan dengan kernel linier untuk mendapatkan hasil terbaik. Penelitian ini juga memanfaatkan algoritma Relief-f. Dataset yang digunakan terdiri dari 3426 baris dan 5 kolom. Hasil pengujian menunjukkan tingkat akurasi data sebesar 100% [5].

Berdasarkan penjelasan yang telah disampaikan, akan dilakukan penelitian tentang prediksi penyakit stroke dengan metode random forest. Data yang didapatkan berasal dari website Kaggle.com. harapan dari penelitian ini adalah dapat membantu kalangan medis untuk dengan mudah mendiagnosa seseorang terkena penyakit stroke. Karena semakin dini penyakit terdeteksi, semakin cepat pula penanganannya.

2. Metodologi Penelitian

2.1. Tahapan Penelitian

Garis besar penelitian mencakup rangkaian langkah yang akan diambil selama pelaksanaan penelitian ini, mulai dari permulaan hingga kesimpulan. Proses penelitian dapat diilustrasikan melalui diagram alir seperti yang ditunjukkan dalam ilustrasi di bawah ini.



Gambar 1. Tahapan Penelitian

2.2. Pengumpulan Data

Data yang digunakan terdiri dari 40910 record. Dataset ini terdiri dari 11 indeks stroke. Di bawah ini adalah detail indikator atau atribut kumpulan data yang digunakan.

Tabel 1. Rincisn Indikator

	sex	age	hypertension	heart_disease	ever_married	work_type	Residence_type	avg_glucose_level	bmi	smoking_status	stroke
0	1.0	63.0	0	1	1	4	1	228.69	36.6	1	1
1	1.0	42.0	0	1	1	4	0	105.92	32.5	0	1
2	0.0	61.0	0	0	1	4	1	171.23	34.4	1	1
3	1.0	41.0	1	0	1	3	0	174.12	24.0	0	1
4	1.0	85.0	0	0	1	4	1	186.21	29.0	1	1
...	...	...	...	...	...	...	...	...	...	...	...
40905	1.0	38.0	0	0	0	4	1	120.94	29.7	1	0
40906	0.0	53.0	0	0	1	4	0	77.66	40.8	0	0
40907	1.0	32.0	0	0	1	2	0	231.95	33.2	0	0
40908	1.0	42.0	0	0	1	3	0	216.38	34.5	0	0
40909	1.0	35.0	0	0	0	4	0	95.01	28.0	0	0

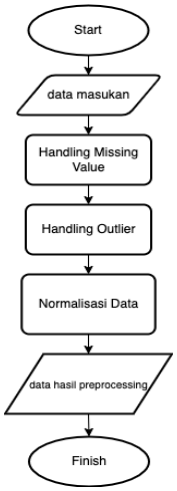
40910 rows × 11 columns

Tabel 2. Attribut Dataset

Attribut	Keterangan
Sex	Jenis Kelamin Pasien
Age	Umur Pasien
Hypertension	Pasien pernah menderita hipertensi (1) atau tidak (0)
Ever_married	pasien menikah (1) atau tidak (0)
Work_type	Tipe pekerjaan pasien
Residence_type	Area tempat tinggal pasien, perkotaan (1) atau pedesaan (0)
Avg_glucose_level	Kadar gula rata-rata pasien
BMI	Indeks Massa Tubuh
Smoking_status	Merokok (1) atau tidak pernah merokok (0)
Stroke	Apakah pasien mengalami stroke (1) atau tidak (0)

2.3. Preprocessing

Data awal perlu melalui tahap pra-pemrosesan sebelum dapat dimanfaatkan oleh sistem. Oleh karena itu, ada beberapa langkah pra-pemrosesan harus diterapkan guna mengubah beberapa data guna meningkatkan kualitasnya. Dalam konteks penelitian ini, proses pra-pemrosesan dilakukan untuk membersihkan data sebelum langkah pemodelan dapat dilakukan.



Gambar 2. Flowchart Preprocessing

2.3.1. Handling Missing Value

Pada tahapan awal *preprocessing*, data masukan dilakukan handling missing value terlebih dahulu. Missing value dapat terjadi karena kesalahan penginputan data atau memang data tersebut tidak ada. Karena algoritma machine learning tidak dapat memproses data yang terdapat *missing value*, maka sebelum dilakukan *modelling* harus dilakukan handling missing value terlebih dahulu. Peneliti menggunakan Teknik imputasi mean. Sehingga data yang terdapat missing value diisi dengan nilai rata-rata dari kolom tersebut [10].

2.3.2. Handling Outlier

Langkah selanjutnya melibatkan penanganan outlier menggunakan metode Z-score. Teknik Z-score digunakan untuk mengidentifikasi apakah suatu data memiliki nilai yang sangat ekstrem, yang dikenal sebagai outlier. Outlier dalam data merujuk pada nilai yang secara signifikan berbeda dari rata-rata. Prinsip umumnya adalah nilai Z-score kurang dari -3 atau lebih dari +3 menunjukkan adanya nilai yang sangat ekstrem dalam data. Oleh karena itu, data yang melebihi batas bawah atau batas atas ini akan dihapus [11].

2.3.3. Normalisasi Data

Normalisasi data merupakan salah satu teknik yang penting untuk dilakukan pada tahapan *preprocessing.* Hal ini dikarenakan seringkali sebuah data memiliki rentang nilai antar variable yang sangat jauh. Pada penelitian ini, menggunakan metode *minmax scaler* untuk melakukan normalisasi data. MinMax Scaler adalah metode pra-pemrosesan data dalam analisis data dan pembelajaran mesin yang digunakan untuk mengubah nilai-nilai fitur dalam dataset sehingga terdistribusi dalam rentang 0 hingga 1 [9].

2.4. *Processing*

Subbab ini menjelaskan tentang proses pengolahan data untuk mencapai hasil yang diharapkan seperti yang telah dijelaskan di atas.

2.4.1. Klasifikasi dengan *Random Forest*

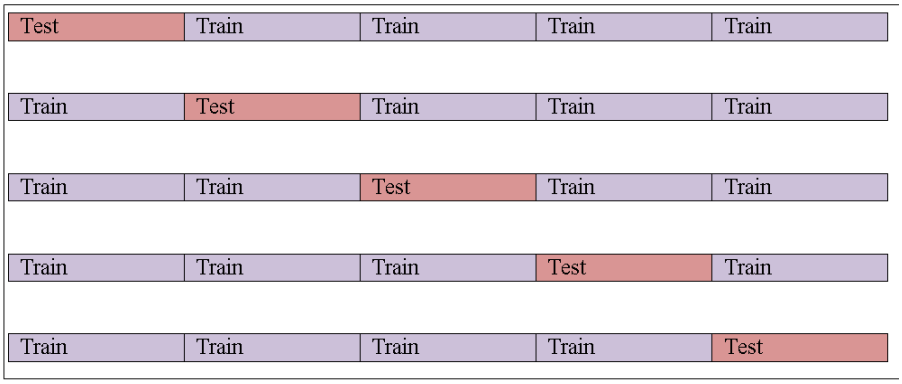
Langkah awal dalam proses klasifikasi menggunakan Random Forest adalah mengimpor data yang telah mengalami ekstraksi fitur ke dalam lingkungan Jupyter Notebook menggunakan pustaka pandas. Setelah data berhasil dimuat, langkah

selanjutnya adalah memisahkan data menjadi dua bagian, yaitu data X dan data y. Data X merujuk pada kolom yang berisi fitur-fitur, sementara data y mencakup kolom-kolom yang berisi target atau label [9].

Setelah langkah pembagian data X dan y, langkah berikutnya adalah membagi data X menjadi data pelatihan (train) dan data uji (test) menggunakan modul `train_test_split` dari pustaka scikit-learn. Pembagian dilakukan dengan perbandingan 90% untuk data pelatihan dan 10% untuk data uji. Untuk memperoleh parameter optimal dalam konteks ini, peneliti menerapkan Teknik Pencarian Acak dengan Validasi Silang (Randomized Search Cross Validation). Metode ini digunakan untuk mencari parameter terbaik untuk algoritma yang digunakan dalam analisis kasus. Pencarian Acak dengan Validasi Silang merupakan bagian dari pustaka scikit-learn yang bertujuan untuk melakukan validasi secara otomatis dan sistematis pada berbagai model dan setiap hyperparameter. Setelah proses Pencarian Acak dengan Validasi Silang selesai, akan dihasilkan model beserta skor uji (test score) dan skor pelatihan (train score) [1].

2.5. Tahap Evaluasi

Tahapan ini digunakan untuk mengukur performa dari model *machine learning* yang dibuat. terdapat tiga metrik evaluasi yang dapat digunakan yaitu *precision*, *recall* dan *confusion matrix*. Terdapat 4 istilah sebagai representasi hasil proses klasifikasi pada *confusion matrix*. Dengan adanya perhitungan dari *confusion matrix* maka dapat diperoleh *accuracy*, *sensitivity*, dan *specificity*. Seperti yang telah dijelaskan pada sub bab sebelumnya, data akan dibagi menjadi data train dan data test. Agar mendapatkan hasil terbaik, beberapa perbandingan data train dan data test yang ada pada table di atas nantinya akan dilakukan percobaan pada masing-masing perbandingan. Dalam penelitian ini, peneliti menggunakan Randomized Search Cross Validation dengan cross validation sebanyak 5 kali. Sehingga, dataset akan dibagi menjadi 5 bagian yang setara. Bagian pertama akan digunakan sebagai data uji, sementara bagian kedua hingga kelima akan menjadi data pelatihan. Sebaliknya, ketika bagian kedua menjadi data uji, bagian pertama dan bagian ketiga hingga kelima akan digunakan sebagai data pelatihan, dan seterusnya hingga bagian kelima menjadi data uji.



Gambar 3. Ilustrasi Cross validation

3. Hasil dan Pembahasan

3.1. Preprocessing

Dari sub bab yang telah dijelaskan sebelumnya, tahap ini dilakukan untuk mengolah data agar siap untuk dilakukan proses pemodelan. Adapun hasil dari preprocessing adalah sebagai berikut.

a) Handling Missing Value

berikut merupakan deskripsi data mentah yang masih terdapat missing value di dalamnya.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40910 entries, 0 to 40909
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   sex                40907 non-null  float64
 1   age                40910 non-null  float64
 2   hypertension       40910 non-null  int64
 3   heart_disease      40910 non-null  int64
 4   ever_married       40910 non-null  int64
 5   work_type          40910 non-null  int64
 6   Residence_type     40910 non-null  int64
 7   avg_glucose_level  40910 non-null  float64
 8   bmi                40910 non-null  float64
 9   smoking_status     40910 non-null  int64
 10  stroke             40910 non-null  int64
dtypes: float64(4), int64(7)
memory usage: 3.4 MB
```

Gambar 4. Deskripsi data mentah

Dari Gambar 4 dapat diketahui bahwa kolom “sex” terdapat missing value. Dari deskripsi tersebut menujukkan bahwa jumlah value pada kolom “sex” terdapat 40907 baris. Oleh karena itu dilakukan handling missing value agar baris pada kolom “sex” terisi seluruhnya.

```
df.info()
```

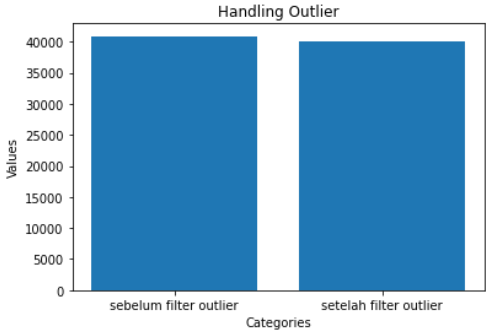
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40910 entries, 0 to 40909
Data columns (total 11 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   sex                40910 non-null  float64
 1   age                40910 non-null  float64
 2   hypertension       40910 non-null  int64
 3   heart_disease      40910 non-null  int64
 4   ever_married       40910 non-null  int64
 5   work_type          40910 non-null  int64
 6   Residence_type     40910 non-null  int64
 7   avg_glucose_level  40910 non-null  float64
 8   bmi                40910 non-null  float64
 9   smoking_status     40910 non-null  int64
 10  stroke             40910 non-null  int64
dtypes: float64(4), int64(7)
memory usage: 3.4 MB
```

Gambar 5. Deskripsi data setelah handling missing value

Gambar 5 merupakan data yang telah dilakukan handling missing value. Dapat dilihat dari Gambar tersebut bahwa baris pada kolom “sex” telah terisi seluruhnya.

b) Handling Outlier

Seperti yang telah dijelaskan pada sub bab sebelumnya bahwa pada penelitian ini untuk melakukan filter outlier peneliti mnggunakan Teknik zcore. Berikut merupakan hasil dari filter outlier menggunakan zscore.



Gambar 6. Hasil filter outlier

c) Normalisasi Data

Berikut merupakan hasil dari normalisasi data menggunakan minmax scaller

```
Original Data:
       sex   age  hypertension  heart_disease  ever_married  work_type  \
0      1.0  63.0           0.0            1.0           1.0        4.0
1      1.0  42.0           0.0            1.0           1.0        4.0
2      0.0  61.0           0.0            0.0           1.0        4.0
3      1.0  41.0           1.0            0.0           1.0        3.0
4      1.0  85.0           0.0            0.0           1.0        4.0
...    ...   ...           ...            ...           ...        ...
40905  1.0  38.0           0.0            0.0           0.0        4.0
40906  0.0  53.0           0.0            0.0           1.0        4.0
40907  1.0  32.0           0.0            0.0           1.0        2.0
40908  1.0  42.0           0.0            0.0           1.0        3.0
40909  1.0  35.0           0.0            0.0           0.0        4.0

       Residence_type  avg_glucose_level   bmi  smoking_status  stroke
0                 1.0             228.69  36.6             1.0     1.0
1                 0.0             105.92  32.5             0.0     1.0
2                 1.0             171.23  34.4             1.0     1.0
3                 0.0             174.12  24.0             0.0     1.0
4                 1.0             186.21  29.0             1.0     1.0
...               ...                ...   ...             ...     ...
40905             1.0             120.94  29.7             1.0     0.0
40906             0.0              77.66  40.8             0.0     0.0
40907             0.0             231.95  33.2             0.0     0.0
40908             0.0             216.38  34.5             0.0     0.0
40909             0.0              95.01  28.0             0.0     0.0

[40001 rows x 11 columns]

Normalized Data:
[[1.         0.64285714 0.         ... 0.63705584 1.         1.        ]
 [1.         0.45535714 0.         ... 0.53299492 0.         1.        ]
 [0.         0.625      0.         ... 0.58121827 1.         1.        ]
 ...
 [1.         0.36607143 0.         ... 0.55076142 0.         0.        ]
 [1.         0.45535714 0.         ... 0.58375635 0.         0.        ]
 [1.         0.39285714 0.         ... 0.41878173 0.         0.        ]]
```

Gambar 7. Hasil normalisasi data

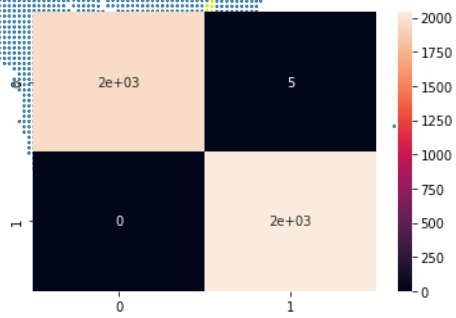
Dari Gambar 7 dapat dilihat bahwa seluruh data telah diubah menjadi range yang sama yaitu antara 0 hingga 1.

3.3. Klasifikasi dengan Random Forest

Setelah melalui beberapa tahap pra-pemrosesan, langkah berikutnya adalah melakukan klasifikasi menggunakan salah satu algoritma machine learning, yaitu Random Forest. Data akan dibagi menjadi dua bagian, yaitu data latih dan data uji, dengan proporsi 90% untuk data latih dan 10% untuk data uji. Setelah pembagian data, peneliti akan melanjutkan dengan melakukan penyetelan hiperparameter untuk mendapatkan konfigurasi terbaik. Teknik yang digunakan untuk penyetelan hiperparameter ini adalah randomized search cross validation, yang memungkinkan penyetelan parameter dilakukan dengan lebih efisien dalam hal waktu jika dibandingkan dengan grid search cross validation [2]. Proses tuning tersebut menghasilkan skor pelatihan sebesar 100% dan skor uji sebesar 99%.

3.4. Evaluasi

Setelah tahap klasifikasi selesai, evaluasi dilakukan untuk mengukur akurasi, presisi, recall, dan skor F1. Pada penelitian ini diperoleh skor akurasi sebesar 99% yang dapat dilihat pada Gambar 7. Untuk menampilkan confusion matrix yang terdapat pada Gambar 6, peneliti menggunakan pustaka scikit-learn dengan memanggil fungsi metrik. Di sisi lain, untuk menampilkan perhitungan lebih lanjut dari confusion matrix yang terdapat pada Gambar 7, Peneliti juga memanfaatkan pustaka scikit-learn dengan menggunakan fungsi classification_report.



Gambar 8. Confusion Matrix

	precision	recall	f1-score	support
0.0	1.00000	0.99643	0.99821	1961
1.0	0.99658	1.00000	0.99829	2040
accuracy			0.99825	4001
macro avg	0.99829	0.99822	0.99825	4001
weighted avg	0.99826	0.99825	0.99825	4001

Gambar 9. Classification Report

4. Kesimpulan

Penelitian ini berhasil melalui semua tahapan, mulai dari pra-pemrosesan hingga pemrosesan serta evaluasi. Hasil yang diperoleh dari penelitian ini menunjukkan tingkat akurasi sebesar 99%. Dari hasil ini, pada penelitian yang akan datang, diharapkan dapat ditemukan metode lain yang dapat digunakan sebagai pembanding antara algoritma Random Forest dengan algoritma machine learning lainnya. Hal ini akan membantu dalam memperluas pemahaman tentang performa algoritma dan memberikan wawasan lebih dalam dalam pemilihan model yang tepat.

Daftar Pustaka

[1] M. A. As Sarofi, I. Irhamah, and A. Mukarromah, "Identifikasi Genre Musik dengan Menggunakan Metode Random Forest," *Jurnal Sains dan Seni ITS*, vol. 9, no. 1, pp. 79–86, 2020, doi: 10.12962/j23373520.v9i1.51311.

[2] E. Agustin, A. Eviyanti, and N. Lutvi Azizah, "Deteksi Penyakit Epilepsi Melalui Sinyal EEG Menggunakan Metode DWT dan Extreme Gradient Boosting," vol. 7, no. 1, pp. 117–127, 2023, doi: 10.30865/mib.v7i1.5412.

[3] N. Permatasari, "Perbandingan Stroke Non Hemoragik dengan Gangguan Motorik Pasien Memiliki Faktor Resiko Diabetes Melitus dan Hipertensi," *Jurnal Ilmiah Kesehatan Sandi Husada*, vol. 11, no. 1, 2020, doi: 10.35816/jiskh.v11i1.273.

[4] J. Antares, "Artificial Neural Network Dalam Mengidentifikasi Penyakit Stroke Menggunakan Metode Backpropagation (Studi Kasus di Klinik Apotik Madya Padang)," *Djtechno: Jurnal Teknologi Informasi*, vol. 1, no. 1, 2021, doi: 10.46576/djtechno.v1i1.965.

[5] M. A. As Sarofi, I. Irhamah, and A. Mukarromah, "Identifikasi Genre Musik dengan Menggunakan Metode Random Forest," *Jurnal Sains dan Seni ITS*, vol. 9, no. 1, 2020, doi: 10.12962/j23373520.v9i1.51311.

[6] F. Akbar, H. Wira Saputra, A. Karel Maulaya, M. Fikri Hidayat, and Rahmaddeni, "Implementasi Algoritma Decision Tree C4.5 dan Support Vector Regression untuk Prediksi Penyakit Stroke," vol. 2, no. October, pp. 61–67, 2022.

[7] U. Amelia *et al.*, "Implementasi Algoritma Support Vector Machine (Svm) Untuk Prediksi Penyakit Stroke Dengan Atribut Berpengaruh," *... Student Journal for ...*, vol. III, pp. 254–259, 2022.

[8] M. N. Maskuri, K. Sukerti, and R. M. Herdian Bhakti, "Penerapan Algoritma K-Nearest Neighbor (KNN) untuk Memprediksi Penyakit Stroke Stroke Desease Predict Using KNN Algorithm," *Jurnal Ilmiah Intech : Information Technology Journal of UMUS*, vol. 4, no. 1, pp. 130–140, 2022.

[9] A. Byna and M. Basit, "Penerapan Metode Adaboost Untuk Mengoptimasi Prediksi Penyakit Stroke Dengan Algoritma Naïve Bayes," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 9, no. 3, pp. 407–411, 2020, doi: 10.32736/sisfokom.v9i3.1023.

[10] D. E. Cahyani, "Penerapan Machine Learning Untuk Prediksi Penyakit Stroke," *Jurnal Kajian Matematika dan Aplikasinya*, vol. 3, no. January, pp. 8–14, 2022, doi: 10.17977/um055v3i1p15-22.

[11] D. Prajarini, S. Tinggi, S. Rupa, D. Desain, and V. Indonesia, "Perbandingan Algoritma Klasifikasi Data Mining Untuk Prediksi Penyakit Kulit," *Informatics Journal*, vol. 1, no. 3, p. 137, 2016.

[12] R. S. Rohman, R. A. Saputra, and D. A. Firmansaha, "Komparasi Algoritma C4.5 Berbasis PSO Dan GA Untuk Diagnosa Penyakit Stroke," *CESS (Journal of Computer Engineering, System and Science)*, vol. 5, no. 1, p. 155, 2020, doi: 10.24114/cess.v5i1.15225.

[13] M. A. As Sarofi, I. Irhamah, and A. Mukarromah, "Identifikasi Genre Musik dengan Menggunakan Metode Random Forest," *Jurnal Sains dan Seni ITS*, vol. 9, no. 1, pp. 79–86, 2020, doi: 10.12962/j23373520.v9i1.51311.

[14] J. Xu, Y. Zhang, and D. Miao, "Three-way confusion matrix for classification: A measure driven view," *Inf Sci (N Y)*, vol. 507, 2020, doi: 10.1016/j.ins.2019.06.064.