 ISSN:2302-8149
e-ISSN:2540-9719

Komparasi k-Nearest Neighbor dan Support Vector Machine dengan Optimasi Binary Dragonfly Algorithm

Comparison of k-Nearest Neighbor and Support Vector Machine using Binary Dragonfly Algorithm Optimization

[1]Andi Nugroho*, [2]Muhammad Imam Khomeini, [3]Rifan Heraldi
[1,2,3]Sistem Informasi, Fakultas Ilmu Komputer, Universitas Mercu Buana
Jalan Raya, RT.4/RW.1, Meruya Sel., Kec. Kembangan, Jakarta, Daerah Khusus Ibukota Jakarta 11650
*e-mail: andi.nugroho@mercubuana.ac.id

(***received*:** 12 Juni 2023, ***revised*:** 29 Juli 2023, ***accepted*:** 19 September 2023)

Abstrak

BDA merupakan adaptasi dari Dragonfly Algorithm (DA) yang bertujuan untuk mengoptimalkan komputasi permasalahan single-objecivet, discrete, dan multi-objective. Penggabungan algoritma optimasi BDA dengan algoritma klasifikasi KNN dan SVM bertujuan untuk meningkatkan kinerja model prediksi. Penelitian ini bertujuan membandingkan dan menguji keakuratan akurasi dari algoritma KNN dan SVM terhadap dataset diabetes yang digunakan pada penelitian untuk mengetahui algoritma terbaik dalam memprediksi diabetes. Penelitian ini menggunakan algoritma optimasi BDA untuk menyeleksi fitur-fitur terbaik pada dataset, kemudian algoritma klasifikasi KNN dan SVM, dalam mengklasifikasikan data, memprediksi, dan membandingkan keakuratan akurasi dari kedua algoritma terhadap dataset diabetes. Data rekam medis dari pengidap diabetes diolah menggunakan algoritma KNN dan SVM, yang kemudian akan menghasilkan tingkat akurasi yang dapat digunakan dalam memprediksi diabetes. Penelitian terdahulu, telah melakukan perbandingan antara algoritma-algoritma klasifikasi dalam memprediksi penyakit diabetes. Pada penelitian terdahulu diatas belum ada yang menggabungkan BDA dengan algoritma klasifikasi, karena BDA sendiri merupakan suatu metode yang relatif baru dan belum banyak diteliti, sehingga peneliti menggunakan algoritma optimasi ini. Hasil dari penelitian yang dilakukan mendapatkan hasil akurasi tertinggi pada algoritma BDA + KNN dengan nilai Presisi 96,10%, Recall 79,36%, F-1 Score 86,93% dan Akurasi 85,55%.

Kata kunci: Diabetes, Machine Learning, Binary Dragonfly Algorithm (BDA), K-Nearest Neighbor (KNN), Support Vector Machine (SVM).

Abstract

BDA is an adaptation of Dragonfly Algorithm (DA) that optimizes computation for single-objective, discrete, and multi-objective problems. Combining BDA optimization algorithm with KNN and SVM classification algorithms aims to improve the performance of the prediction model. This research compares and tests accuracy of KNN and SVM algorithms on the diabetes dataset used in research to find out the best algorithm in predicting diabetes. This research uses the BDA optimization algorithm to select the best features in the dataset, then the KNN and SVM classification algorithms, in classifying data, predicting, and comparing the accuracy of the accuracy of the two algorithms on the diabetes dataset. Medical record data from people with diabetes is processed using the KNN and SVM algorithms, which will then produce an accuracy level that can be used in predicting diabetes. Previous research has conducted a comparison between classification algorithms in predicting diabetes. In the previous research above, no one has combined BDA with classification algorithms, because BDA itself is a relatively new method and has not been widely studied, so researchers use this optimization algorithm. The results of the research conducted obtained the highest accuracy results in the BDA + KNN algorithm with a Precision value of 96.10%, Recall 79.36%, F-1 Score 86.93% and Accuracy 85.55%.

Keywords: *Diabetes, Machine Learning, Binary Dragonfly Algorithm (BDA), K-Nearest Neighbor (KNN), Support Vector Machine (SVM).*

 ISSN:2302-8149
e-ISSN:2540-9719

1 Pendahuluan

Perkembangan teknologi saat ini telah mengantarkan era baru pada bidang kedokteran melalui evaluasi penyakit berbasis data dengan menggabungkan machine learning dan ilmu biomedis [1]. Dengan bantuan machine learning dan algoritma relatifnya, masalah dan hambatan signifikan dalam pendeteksian diabetes dapat lebih mudah diatasi, namun memberikan hasil yang akurat [2]. Maka dari itu, machine learning kini digunakan dalam proses mengidentifikasi, dan mendiagnosis penyakit untuk meminimalisir risiko kematian, serta meningkatkan status kesehatan pasien, karena machine learning berkontribusi pada keputusan tertentu.

Salah satu metode pada machine learning adalah supervised learning yang berfungsi dalam mengidentifikasi label input untuk membuat prediksi dan klasifikasi [3]. K-Nearest Neighbor (KNN) merupakan salah satu algoritma supervised learning yang bertugas dalam mengklasifikasikan data berdasarkan data latih dari tetangga terdekat, dimana k adalah jumlah tetangga terdekat [4]. Algoritma ini menghubungkan pencarian dan penetapan nilai rata-rata dari titik data yang teridentifikasi ke k titik data terdekat dalam training data pada titik data yang nilai targetnya tidak dapat dicapai [5]. Selain itu Support Vector Machine (SVM) adalah salah satu algoritma terbaik yang digunakan untuk menyelesaikan masalah klasifikasi dan regresi [6]. Pada algoritma ini, titik data dipetakan dari ruang data ke dalam ruang fitur berdimensi tinggi menggunakan fungsi kernel [7]. Binary Dragonfly Algorithm (BDA) merupakan salah satu teknik optimasi metaheuristik dan dikategorikan sebagai salah satu jenis algoritma evolusioner [8]. Mekanisme eksplorasi dan eksploitasi Dragonfly Algorithm (DA) dimodelkan dengan interaksi capung dalam menghindari musuh dan mencari sumber makanan [9]. DA sendiri dianggap sebagai algoritma optimasi yang memiliki kinerja tinggi, dan dapat mengungguli algoritma optimasi terkenal lainnya karena kesederhanaan dan efisiensinya [10].

Penelitian ini bertujuan untuk membandingkan dua algoritma machine learning populer, yaitu k-Nearest Neighbor (k-NN) dan Support Vector Machine (SVM), dengan menggunakan algoritma Binary Dragonfly Algorithm (BDA) sebagai algoritma optimasi dalam pemilihan fitur. Dataset yang digunakan dalam penelitian ini adalah data diabetes yang bersumber dari National Institute of Diabetes and Digestive and Kidney Diseases. Hasil dari penelitian ini dapat mengetahui dan membandingkan nilai Accuracy, Recall, Precision, dan F1_Score dari dua algoritma machine learning tersebut dalam melakukan klasifikasi pada dataset yang digunakan.

2 Tinjauan Literatur

Rakesh, Sanjay, Kusuma, dan Sampath [11] pada tahun 2019 melakukan penelitian dengan membandingkan dua algoritma yaitu dan Naïve Bayes Classifiers untuk melakukan analisis prediksi penyakit diabetes. Hasil dari penelitian ini yaitu efisiensi prediksi algoritma Naïve Bayes dengan tingkat akurasi 62,5% dan efisiensi prediksi Support Vector Machine demgam tingkat akurasi 82%.

Penelitian yang dilakukan Rahul dan Sajal [12] mengenai perbandingan algoritma machine learning dalam mendeteksi pola dan faktor risiko pada dataset diabetes menggunakan Support Vector Machine (SVM), K-Nearest Neighbor (KNN), Naïve Bayes (NB), Decision Tree (DT), Random Forest (RF), dan Logistic Regression (LR). Penelitian yang dilakukan memperoleh hasil akurasi pada algoritma KNN dan NB sebesar 75%, untuk algoritma DT mendapatkan nilai akurasi sebesar 82,7%, lalu algoritma SVM memperoleh nilai akurasi sebesar 74%, pada algoritma LR mencapai nilai akurasi sebesar 76%, dan algoritma RF mencapai akurasi tertinggi sebesar 84%.

Pada penelitian yang dilakukan oleh Naik, Kuppili, dan Reddy [13] pemilihan fitur hybrid baru yang menggunakan Fisher Score (FS) sebagai pendekatan filter dan pendekatan wrapper yang menggunakan Binary Dragonfly Algorithm (BDA), dan Radial Basis Function Neural Network (RBFNN). Hasil pengujian menunjukkan bahwa metode yang digunakan dalam penelitian ini memiliki kinerja lebih baik dibandingkan menggunakan metode Fisher Score secara terpisah dalam mengidentifikasi fitur terpenting dari kumpulan dataset microarray. Hal ini menunjukkan potensi penggunaan metode hybrid feature selection dengan BDA dan RBFNN dalam meningkatkan pemilihan fitur pada dataset mikroarray.

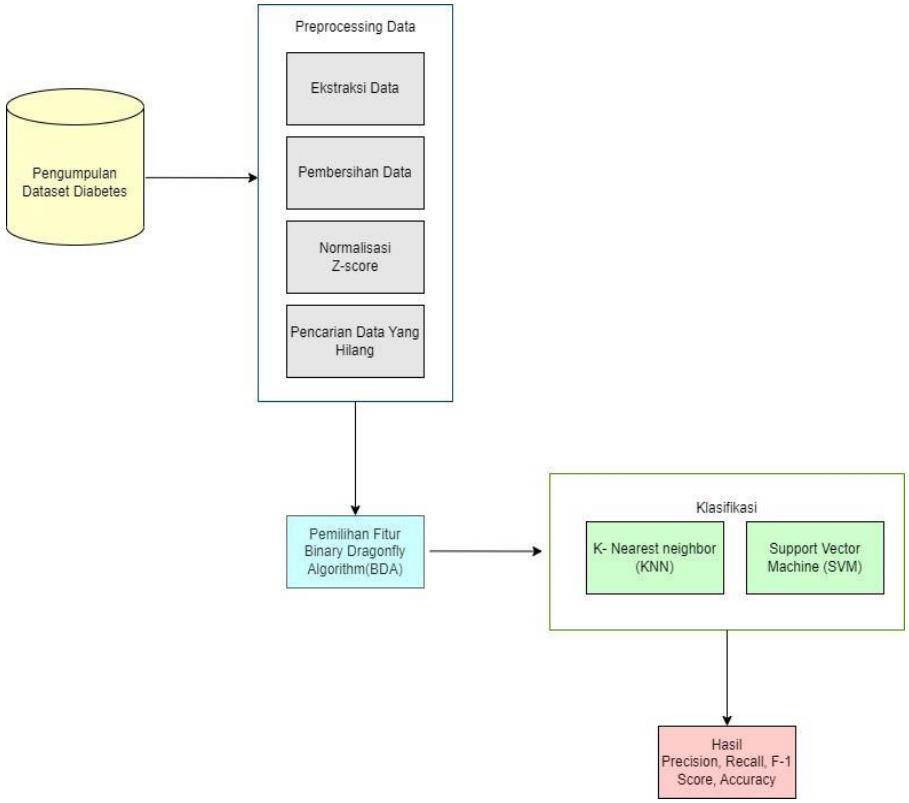
Berdasarkan penelitian terdahulu diatas pemilihan algoritma memiliki peran penting dalam mengklasifikasi dan menyeleksi fitur yang ada pada dataset, dengan pemilihan algoritma yang tepat akan menghasilkan prediksi yang tepat. Pada penelitian ini akan menggunakan algoritma klasifikasi KNN dan SVM, serta algoritma opstimasi BDA, peneliti menggunakan algoritma optimasi BDA sendiri

 ISSN:2302-8149
e-ISSN:2540-9719

karena metode ini merupakan pendekatan yang relatif baru dan belum banyak diteliti, sehingga peneliti menggunakan algoritma optimasi ini.

3 Metode Penelitian

Pada metode penelitian ini didapatkan sumber dataset rekam medis pengidap diabetes dengan melakukan optimasi nilai KNN dan SVM menggunakan algoritma BDA untuk mendapatkan hasil nilai yang akurat. Dalam penelitian ini akan dilakukan beberapa langkah-langkah atau tahapan penelitian seperti yang digambarkan pada Gambar 1.



Gambar 1. Diagram Alir Penelitian.

Gambar 1 merupakan diagram alir penelitian yang merupakan tahapan-tahapan yang akan digunalkan pada penelitian, berikut merupakan penjelasan dari tahap-tahap yang akan digunakan :

a. Pengumpulan Dataset
Pada tahap ini, dilakukan pengumpulan dataset untuk melakukan penelitian, dataset diambil dari situs kaggle https://www.kaggle.com/datasets/houcembenmansour/predict-diabetes-based-on-diagnostic-measures. Data yang diteliti merupakan data yang didapat dari *National Institute of Diabetes and Digestive and Kidney Diseases* terhadap pemeriksaan terhadap 390 baris dengan 16 kolom yaitu patient_number, cholesterol, glucose, hdl_chol, chol_hdl_ratio, age, gender, height, weight, bmi, systolic_bp, diastolic_bp, waist, hip, waist_hip_ratio, diabetes.

b. Pre-Processing Data
Pada tahap ini dilakukan preprocessing data untuk mengolah data mentah yang didapatkan pada situs Kaggle, agar menjadi data yang siap diolah untuk mempermudah proses selanjutnya. Berikut tahapan *pre-processing* yang dilakukan :

1. Data Extraction : Dalam tahap ini dilakukan pengambilan data dari dataset yang berisi informasi mengenai rekam medis pengidap diabetes, sumber data tersebut diperoleh dari situs Kaggle, data ini merupakan hasil diagnosa terhadap pasien dari National Institute of Diabetes and Digestive and Kidney Diseases.

 ISSN:2302-8149
e-ISSN:2540-9719

2. Data cleaning : Pada tahap ini juga dilakukan konversi huruf menjadi angka, agar dataset dapat dinormalisasi dengan menggunakan metode z-score normalization yang akan digunakan pada tahap selanjutnya.
3. Z-Score Normalization : Metode normalisasi yang digunakan pada penelitian ini menggunakan metode normalisasi Z-Score. Normalisasi z-score digunakan dengan cara mengambil nilai matriks dan mengurangkan setiap nilai dari rata-rata, lalu membaginya dengan standar deviasi.
4. Missing values : Missing values pada penelitian yang dilakukan digunakan untuk mencari nilai yang hilang pada data, yang menyebabkan terjadinya data yang bias, sehingga akan mempengaruhi kekuatan pada nilai akurasi pada klasifikasi algoritma.

c. Binary Dragonfly Algorithm

Binary Dragonfly Algorithm (BDA) merupakan adaptasi dari Dragonfly Algorithm (DA), dimana DA merupakan algoritma yang bertujuan untuk mengoptimalkan komputasi permasalahan single-objecivet, discrete, dan multi-objective [14]. *Dragonfly algorithm* merupakan metode metaheuristik yang diusulkan dengan mengulangi perilaku capung dalam bentuk matematis [15]. Menurut Reynolds, pengoperasian dari segerombolan capung, mengikuti tiga konsep dasar yaitu Separation, Alignment, dan Cohesion. Setiap perilaku tersebut dimodelkan secara matematis yang sesuai. Separation dihitung secara matematis pada persamaan berikut [16]:

$$S_i = -\sum_{j=1}^{n} X - X_j \quad (1)$$

Pada Eq.1 X mempresentasikan lokasi saat ini, X_j merupakan lokasi individu tetangga dari j^{th} dan n adalah jumlah individu tetangga.
Alignment dihitung dengan persamaan yang diberikan :

$$A_i = \sum_{j=1}^{n} V_j \quad (2)$$

Pada Eq.2 V_j mempresentasikan vektor kecepatan lokasi j^{th}.
Cohesion dihitung secara matematis pada persamaan dibawah :

$$C_i = \frac{\sum_{j=1}^{n} X_j}{n} - X \quad (3)$$

Eq.3 menjelaskan bahwa nilai Ci adalah kohesi untuk individu ith, n adalah ukuran lingkungan, Xj adalah posisi capung tetangga jth, dan X adalah individu capung saat ini.
Ketertarikan pada sumber makanan, serta gangguan dari luar, dan musuh dihitung dengan persamaan di bawah ini :

$$F_i = F^{+} - X \quad (4)$$

Eq.4 menjelaskan dimana, F^{+} merupakan lokasi sumber makanan, X merupakan lokasi makanan saat ini.

$$E_i = E^{-} + X \quad (5)$$

Eq. 5 menjelaskan bahwa E^{-}menunjukkan lokasi musuh, X adalah posisi anggota saat ini.
Untuk menemukan solusi yang optimal pada masalah optimisasi yang diberikan, DA mendeskripsikan vektor posisi dan vektor langkah Separation, Alignment, untuk setiap agen pencarian pada swarm. Vektor ini digunakan untuk memperbarui posisi agen pencarian dalam ruang pencarian pada tugas pengoptimalan yang diberikan. Langkah pada vektor serupa dengan vektor kecepatan particle swarm optimization (PSO), algoritma BDA dibangun berdasarkan algoritma PSO. Vektor langkah yang berkaitan dengan arah perjalanan capung diformulasikan sebagai berikut [17]:

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \quad (6)$$

 ISSN:2302-8149
e-ISSN:2540-9719

Pada Eq. 6 nilai *s*, *a*, *c*, *f*, dan *e* masing-masing dikenal sebagai faktor bobot untuk *separation* (S_i), *alignment* (A_i), *cohesion* (C_i), *food* (F_i) *dan enemy* (E_i) dari individu i^{th} pada agen pencari masing-masing, *w* mengacu pada bobot inersia dan *t* mengacu pada perhitungan iterasi saat ini. Vektor langkah (ΔX) yang diperoleh digunakan untuk memperkirakan vektor posisi agen pencari *X* sebagai berikut :

$$X_{t+1} = X_t + \Delta X_{t+1} \quad (7)$$

Pada Eq.7 nilai *t* menunjukkan iterasi saat ini
Model dasar pengoptimal Dragonfly diusulkan untuk masalah dalam ruang pencarian berkelanjutan. Seekor capung dapat memperbarui posisinya dengan menambahkan vektor langkah ke vektor posisinya. Pemilihan fitur merupakan masalah optimasi biner, sehingga strategi pembaharuan ditunjukkan pada *equation* (11) Tidak mungkin untuk ruang pencarian biner. BDA menggunakan fungsi transfer berikut untuk mengubah nilai vektor langkah menjadi angka dalam rentang nilai [0,1] [18] :

$$T(\Delta X_{t+1}) = \frac{|\Delta X|}{\left|\sqrt{(\Delta X^2)+1}\right|} \quad (8)$$

Pada Eq. 8 Fungsi transfer digunakan untuk memilih probabilitas memperbarui posisi capung dalam swarm, kemudian persamaan berikut digunakan untuk memperbarui posisi capung (agen pencarian) :

$$X_{t+1} = \begin{cases} \neg X_t, r < T(\Delta X_{t+1}) \\ X_t, r \geq T(\Delta X_{t+1}) \end{cases} \quad (9)$$

Nilai *r* pada Eq, 9 adalah angka dalam rentang nilai [0,1].

d. K-Nearest Neighbor
K-Nearest Neighbor (KNN) merupakan salah satu algoritma supervised learning, dimana hasil sampel yang akan diuji berfungsi untuk diklasifikasikan berdasarkan mayoritas nilai k tetangga terdekat, k merupakan hyperparameter yang ditentukan oleh pengguna [19]. Banyak fungsi metrik jarak yang digunakan dalam KNN dan algoritma *machine learning* lainnya. Pada umumnya KNN menggunakan metode perhitungan jarak dengan Euclidean distance yang berfungsi untuk menguji ukuran sebagai bentuk jarak kedekatan antara dua objek [20]. Tetangga terdekat pada KNN didefinisikan dalam rentang Euclidean antara dua objek $X = (x_1, x_2, \ldots, x_n)$ dan $Y = (y_1, y_2, \ldots, y_n)$ [21], berikut persamaan dari beberapa metrik jarak [22].

1. Metrik jarak Euclidean ditunjukkan pada persamaan berikut :

$$(x, y) = (\sum_{i=1}^{n}(x_i - y_i)^2)^{1/2} \quad (10)$$

Pada Eq. 10 nilai x_i dan y_i adalah nilai fitur dari *x* dan *y* pada dimensi i_{th}, dan *n* adalah jumlah dimensi dalam ruang fitur.

2. Metrik Jarak Minkowski ditunjukkan pada persamaan berikut :

$$D\ (x, y) = (\sum_{i=1}^{n}(x_i - y_i)^p)^{1/p} \quad (11)$$

Pada Eq. 11 menjelaskan nilai x_i dan y_i adalah nilai fitur dari data x dan y pada dimensi i_th, dan n adalah jumlah dimensi dalam ruang fitur. Nilai p adalah parameter yang dapat disesuaikan untuk mengubah sifat metrik jarak.

3. Metrik jarak Chebyshev ditunjukkan pada persamaan berikut :

 ISSN:2302-8149
e-ISSN:2540-9719

$$D\ (x, y) = max_i|x_i - y_i| \quad (12)$$

Pada Eq. 12 nilai yang terdapat pada *x* dan *y* merupakan dua titik data yang dicari untuk menemukan perbedaan absolut masimum antara nilai fitur yang sesuai pada setiap dimensi i_{th}.

4. Metrik jarak Manhattan ditunjukkan pada persamaan berikut :

$$D\ (x, y) = \sum_{i=1}^{n}|x_i - y_i| \quad (13)$$

Pada Eq.13 nilai *x* dan y adalah dua titik data yang dibandingkan. Nilai *n* adalah jumlah dimensi. $x_i\ dan\ y_i$ adalah nilai-nilai atribut dari dua titik data pada masing-masing dimensi.

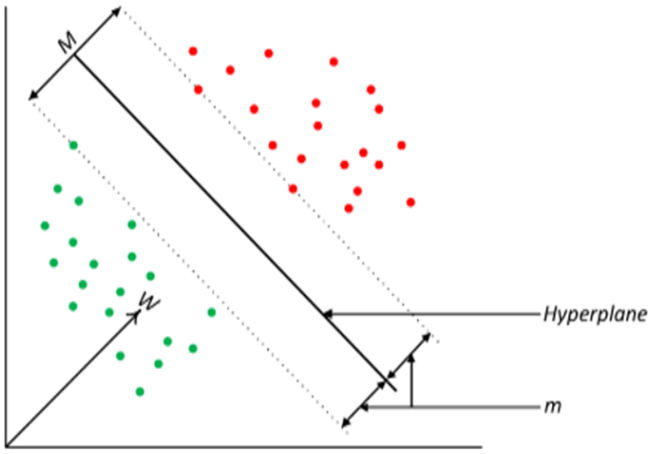
5. Metrik jarak Hamming ditunjukkan pada persamaan berikut :

$$D\ (x, y) = \sum_{i=1}^{n} 1_{x_i \neq y_i} \quad (14)$$

Pada Eq.14 nilai *x* dan *y* merupakan dua vektor biner dengan panjang *n*. $x_i\ dan\ y_i$ adalah nilai dari masing-masing elemen pada posisi i_{th} dari vektor *x* dan *y*. Nilai *1{xi ≠ yi}* adalah fungsi indikator yang bernilai 1 jika x_i tidak sama dengan y_i, dan 0 jika x_i sama dengan y_i.

e. Support Vector Machine

Support Vector Machine (SVM) adalah salah satu algoritma *machine learning* yang berfungsi menganalisis data untuk klasifikasi dan merupakan metode supervised learning yang mengurutkan data ke dalam kelas [23]. SVM mencoba untuk memecahkan masalah klasifikasi dengan membentuk hyperplane yang memaksimalkan margin dengan membagi data ke dalam kelas-kelas [24]. SVM menggambarkan hyperplane yang secara linier memisahkan dua kelas yang ada pada data, yaitu kelas positif dan kelas negatif [25]. Dalam SVM linier, dibuat dua batas margin untuk disebar sejajar dengan dua kelas data, sehingga margin tersebut digunakan untuk menentukan kelas dari setiap data. Kelas baru ditentukan tergantung pada sisi mana dari hyperplane itu berada, seperti Gambar 2.



Gambar 2. Ilustrasi Support Vector Machine

Gambar 2. menggambarkan cara kerja SVM linier, dengan cara membuat dua batas margin untuk disebar sejajar dengan dua kelas data, sehingga margin tersebut digunakan untuk menentukan kelas dari setiap data. Kelas baru pada SVM linier ditentukan tergantung pada sisi mana dari *hyperplane* itu berada, seperti gambar diatas.

Tujuan SVM adalah untuk menggambarkan hyperplane *h(x)* seperti persamaan berikut [26]:

$$h(x) = x^T w + b = 0 \quad (15)$$

 ISSN:2302-8149
e-ISSN:2540-9719

Eq. 15 menggambarkan hyperplane *h(x)* sehingga menghasilkan aturan keputusan klasifikasi *D(x)* yang memaksimalkan margin $M(=2m)$ seperti yang digambarkan pada Eq. 16.

$$D(x) = sign(x^T w + b) \quad (16)$$

Untuk menemukan hyperplane seperti Eq.16, SVM melibatkan pengoptimalan *M* sebagai persamaan pada Eq.17

$$\max_{w,b} M = \min_{w,b} \frac{1}{2}\|w\|^2 \quad (17)$$

Subjek yang terdapat pada $y_i(x^T w + b) \geq 1$, di mana *b* merupakan konstanta, *d* merupakan dimensi data, *w* merupakan vektor dengan panjang yang tidak diketahui dengan dimensi *d* merujuk dari titik asal dan normal ke margin, dan *m* adalah ditampilkan dengan persamaan $\frac{1}{\|w\|}$.

W yang diperoleh dari optimisasi Eq.17 memiliki bentuk yang ditunjukkan pada Eq.18, di mana αi adalah bukan nol untuk kelas i di mana batasan pada $y_i(x^T w + b) \geq 1$ terpenuhi.

$$w = \sum_{i=1}^{N} a_i y_i x_i \quad (18)$$

Menggunakan Eq. 18, *b* dapat ditentukan dari Eq. 15, dan mengikuti aturan dari Eq.16, aturan keputusan untuk sampel baru dari kelas *u* yang tidak diketahui dapat dinyatakan sebagai Eq. 19 :

$$(u) = sign\left[u^T\left(\sum_{i=1}^{N} a_i y_i x_i\right) + b\right] \quad (19)$$

Dimana Eq. 19 α_i merupakan *Lagrangian multipliers* yang diperoleh dari optimisasi Eq. 17.
Pada SVM non-linier menggunakan kernel non-linier, variabel input dapat dipetakan ke ruang dimensi yang lebih tinggi dengan menggunakan hubungan linier. Berikut kernel non-linear pada SVM [27]:
Kernel polinomial :

$$k(x_i, x_j) = (1 + x_i^T\ x_j)^P \quad (20)$$

Di mana Eq. 20 menjelaskan bahwa *k* merupakan simbol kernel, dan *p* merupakan polynomial
Kernel gaussian **:**

$$k(x_i, x_j) = \exp(-\frac{1}{2\sigma^2}\|x_i - x_j\|^2) \quad (21)$$

Pada Eq.21 σ^2merupakan parameter varians
Kernel tangen hiperbolik :

$$k(x_i, x_j) = tanh\ (\beta_0 - \beta_1\ x_i^T\ x_j) \quad (22)$$

Eq. 22 menggabarkan bahwa β_0 dan β_1 amerupakan parameter bebas

f. Confusion Metrix
Confusion matrix adalah tabel yang digunakan untuk menggambarkan kinerja model klasifikasi pada sebuah dataset uji dengan nilai sebenarnya yang diketahui [28]. *Confusion matrix* sanggup memberikan lebih banyak informasi tentang kinerja model, dengan memberikan informasi tentang kelas yang diklasifikasikan menggunakan nilai benar dan salah sehingga dapat mengidentifikasi kesalahan. *Confusion matrix* terdiri dari empat nilai yaitu *True Positive* (TP)

 ISSN:2302-8149
e-ISSN:2540-9719

merupakan prediksi positif yang benar, *False Positive* (FP) merupakan prediksi positif yang salah, *True Negative* (TN) merupakan prediksi negatif yang benar, *False Negative* (FN) merupakan prediksi negatif yang salah.
Nilai-nilai ini diterapkan untuk menghitung yang berikut ini [29]:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (23)$$

$$Recall = \frac{TP}{TP+FN} \quad (24)$$

$$Precision = \frac{TP}{TP+FP} \quad (25)$$

$$F1\_Score = \frac{2*Precision*Recall}{Precision+Recall} \quad (26)$$

Accuracy pada Eq.23 digunakan untuk mengenguji sampel yang diprediksi dengan benar dari seluruh kumpulan data. *Recall* adalah rasio sampel yang diprediksi dengan benar pada sampel positif yang sebenarnya seperti yang terdapat pada Eq. 24. *Precision* adalah rasio sampel yang diprediksi dengan benar pada sampel positif yang diprediksi seperti yang digambarkan pada Eq. 25. Terkadang *Recall* dan *precision* sulit untuk ditukar, hasil *F1_Score* pada Eq. 26 dapat digunakan sebagai metrik keseluruhan yang menggabungkan *Recall* dan *precision.*

4 Hasil dan Pembahasan

Data yang digunakan pada penelitian ini, merupakan data yang diambil dari situs kaggel dengan jumlah record 390 data. Dimana data tersebut adalah data penderita diabetes dengan 16 atribut yaitu *patient_number, cholesterol, glucose, hdl_chol, chol_hdl_ratio, age, gender, height, weight, bmi, systolic_bp, diastolic_bp, waist, hip, waist_hip_ratio, diabetes.*

Data ini merupakan hasil diagnosa terhadap pasien dari *National Institute of Diabetes and Digestive and Kidney Diseases.* Pada tabel 1 berisi informasi mengenai variabel prediktor medis dan satu variabel target dari pengidap diabetes.

Tabel 1. Dataset Diabetes

No	Atribut	Number	Keterangan atribut
1	patient_number	1-390	Nomor pasien
2	cholesterol	78-443	Jumlah kolesterol dalam tubuh
3	glucose	48-385	Jumlah glukosa dalam tubuh
4	hdl_chol	12-120	high-density lipoprotein merupakan kolesterol baik
5	chol_hdl_ratio	2-193	Perbandingan antar kolesterol dalam tubuh
6	Age	19-92	Umur
7	Gender	1-Male, 2-Female	Jenis kelamin
8	Height	52-76	Tinggi badan
9	Weight	99-325	Berat badan
10	BMI	16-558	Indeks massa tubuh
11	systolic_bp	90-250	Tekanan darah sistolik
12	diastolic_bp	48-124	Tekanan darah diastolik
13	waist	26-56	Lingkar pinggang
14	hip	30-64	Lingkar pinggul
15	waist_hip_ratio	1-114	Rasio lingkar pinggang dan lingkar pinggul
16	diabetes	0-No diabetes, 1-diabetes	Penyakit yang menyebabkan metabolisme karbohidrat menjadi tidak normal, dan meningkatkan gula darah

Tabel 1. merupakan tabel yang digunakan dalam penelitian ini, dengan atribut, nomor, dan informasi yang ada pada dataset.

 ISSN:2302-8149
e-ISSN:2540-9719

Pada tahap proses cleaning data, dilakukan dengan mengubah informasi pada bagian gender dan diabetes, dari huruf menjadi angka, dan menghapus coloumn yang berisi informasi mengenai patient_number, seperti yang seperti yang terdapat pada Tabel 2.

Tabel 2. Dataset Awal

Patient_number	cholesterol	glucose	hdl_chol	chol_hdl_ratio	…	Diabetes
1	193	77	49	3.9	…	No diabetes
2	146	79	41	3.6	…	No diabetes
3	217	75	54	4	…	No diabetes
4	226	97	70	3.2	…	No diabetes
5	164	91	67	2.4	…	No diabetes
…	…	…	…	…	…	…
386	227	105	44	5.2	…	No diabetes
387	226	279	52	4.3	…	Diabetes
388	301	90	118	2.6	…	No diabetes
389	232	184	114	2	…	Diabetes
390	165	94	69	2.4	…	No diabetes

Tabel 2. merupakan dataset awal dari variabel prediktor medis pengidap diabetes sebelum dilakukan cleaning data.

Tabel 3. Dataset Setelah Dilakukan Cleaning Data

Cholesterol	Glucose	Hdl_chol	Chol_hdl_ratio	…	Diabetes
193	77	49	3.9	…	0
146	79	41	3.6	…	0
217	75	54	4	…	0
226	97	70	3.2	…	0
164	91	67	2.4	…	0
Cholesterol	**Glucose**	**Hdl_chol**	**Chol_hdl_ratio**	**…**	**Diabetes**
…	…	…	…	…	…
227	105	44	5.2	…	0
226	279	52	4.3	…	1
301	90	118	2.6	…	0
232	184	114	2	…	1
165	94	69	2.4	…	0

Tabel 3. merupakan tabel setelah dilakukan cleaning data, dalam dataset ini dilakukan konversi dari huruf menjadi angka pada atribut *gender* dan *diabetes*, pada atribut *gender* diubah menjadi angka 1 untuk *male* dan angka 2 untuk *female*, lalu atribut *diabetes* diubah menjadi angka 0 untuk *no diabetes* dan 1 untuk *diabetes.* Setelah itu dalam dataset ini terdapat atribut yang dihapus yaitu *patient_number*, pada dataset awal terdapat 16 kolom kemudian setelah dilakukan cleaning data terdapat 15 kolom.

Dalam tahap pada tabel 4. dilakukan normalisasi terhadap dataset diabetes menggunakan metode *z-score normalization*. Normalisasi dilakukan pada 15 kolom, yaitu *cholesterol, glucose, hdl_chol, chol_hdl_ratio, age, gender, height, weight, bmi, systolic_bp, diastolic_bp, waist, hip, waist_hip_ratio,* dan *diabetes* sebagai class.

Tabel 4. Z-Score Normalization

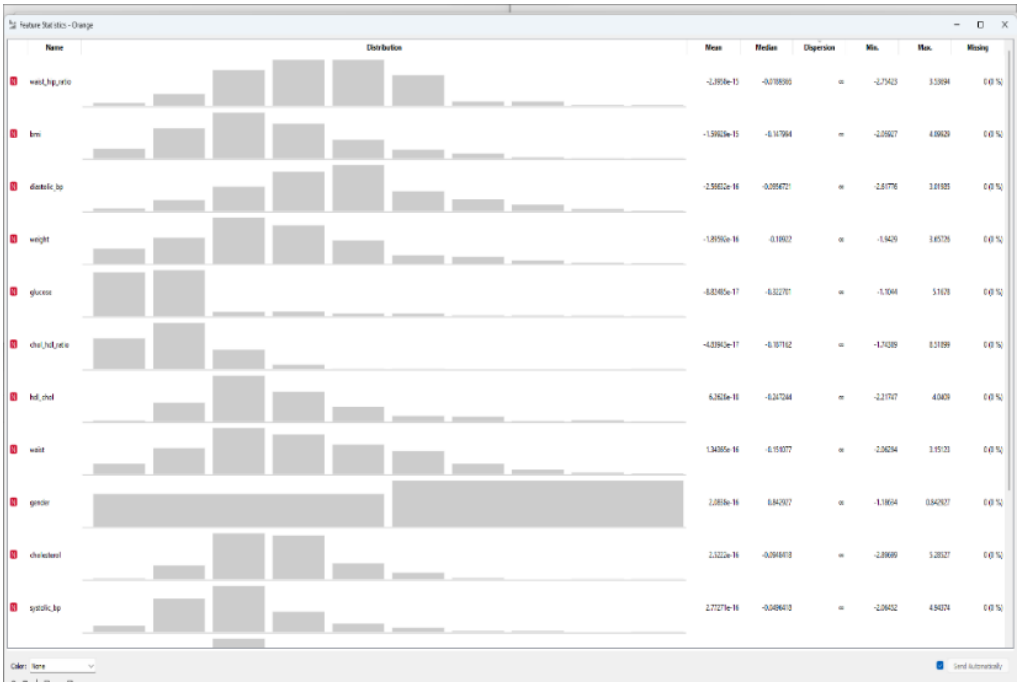
Cholesterol	Glucose	Hdl_chol	Chol_hdl_ratio	…	Class
-0.32	-0.56	-0.07	-0.36	…	0
-1.37	-0.53	-0.54	-0.53	…	0
0.22	-0.60	0.22	-0.30	…	0
Cholesterol	**Glucose**	**Hdl_chol**	**Chol_hdl_ratio**	**…**	**Class**

 ISSN:2302-8149
e-ISSN:2540-9719

0.42	-0.19	1.14	-0.76	…	0
-0.97	-0.30	0.97	-1.22	…	0
…	…	…	…	…	…
0.44	-0.04	-0.36	0.39	…	0
0.42	3.19	0.10	-0.13	…	1
2.10	-0.32	3.92	-1.11	…	0
0.56	1.43	3.69	-1.46	…	1
-0.95	-0.25	1.09	-1.22	…	0

Tabel 4. merupakan hasil dataset setelah dilakukan normalisasi menggunakan normalisasi Z-score.

Setelah dilakukannya cleansing data dan normalisasi, akan dilakukan pencarian *missing values* menggunakan software Orange. Untuk melihat *missing values* dalam dataset yang akan diteliti menggunakan *Feature Statistics* pada software orange seperti yang digambarkan pada Gambar 3.



Gambar 3. Feature Statistics Menggunakan Orange.

Setelah dilakukan identifikasi missing values pada dataset yang digunakan seperti yang ditunjukkan pada Gambar 3, tidak ditemukannya missing values pada dataset yang digunakan.

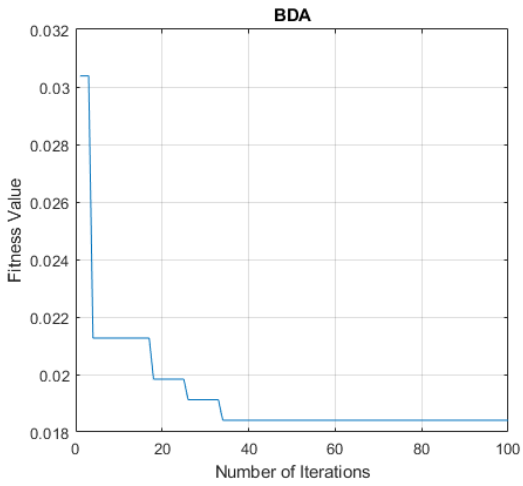
Dalam tahap ini dilakukan seleksi fitur dengan menggunakan model *Binary Dragonfly Algorithm* (BDA). Hasil yang ingin dicapai adalah mendapatkan hasil seleksi fitur optimal yang akan diolah menggunakan algoritma klasifikasi, lalu hasil seleksi fitur optimal yang didapat dari algoritma BDA akan diuji menggunakan algoritma klasifikasi KNN dan SVM. Tabel 5 dibawah ini merupakan parameter yang digunakan pada *Binary Dragonfly Algorithm*.

Tabel 5. Binary Dragonfly Algorithm Parameters

Parameter	BDA
Number of Dragonflies	24
Maximum Iterations	100

Pada tabel 5 Nilai N adalah jumlah capung yang digunakan dalam pencarian, nilai parameter adalah nilai yang telah diuji sebelumnya, dan maksimum iterasi adalah proses pengulangan maksimum dalam algoritma BDA. Dari pengaturan parameter tersebut, didapatkan hasil dari proses seleksi fitur BDA seperti yang ditunjukkan pada gambar 4.

 ISSN:2302-8149
e-ISSN:2540-9719



Gambar 4. Feature Selection Iteration BDA

Gambar 4. Merupakan hasil dari feature selection iteration pada algoritma BDA

Setelah proses pada algoritma BDA diilakukan, fitur yang digunakan dalam algoritma klasifikasi ditunjukkan pada tabel 6.

Tabel 6. Feature Selection Menggunakan Binary Dragonfly Algorithm

No	Atribut	Information
1	cholesterol	
2	glucose	✓
3	hdl_chol	✓
4	chol_hdl_ratio	✓
5	Age	✓
6	Gender	✓
7	Height	
8	Weight	
9	BMI	✓
10	systolic_bp	
11	diastolic_bp	
12	waist	✓
3	hip	✓
14	waist_hip_ratio	
15	class	✓

Pada Tabel 6 terdapat hasil seleksi fitur menggunakan algoritma BDA dari dataset diabetes yang dimasukkan untuk proses seleksi fitur berjumlah 15 atribut dan didapatkan hasil menjadi 9 atribut.

Pada tahap ini setiap data akan diuji dengan menggunakan *cross-validation* untuk mengevaluasi seberapa baik metode KNN dan SVM dapat melakukan generalisasi pada data yang belum pernah ditemui sebelumnya. *Cross-validation* membantu dalam menemukan parameter optimal untuk model. Dalam setiap iterasi pada *cross-validation*, model dapat diuji dengan berbagai konfigurasi parameter untuk mengevaluasi performanya. Setelah dilakukan klasifikasi pada algoritma KNN dan SVM pada dataset menggunakan Matlab software. Maka didapatkan setting parameter terbaik pada dataset yang digunakan seperti yang ditunjukkan pada Tabel 7.

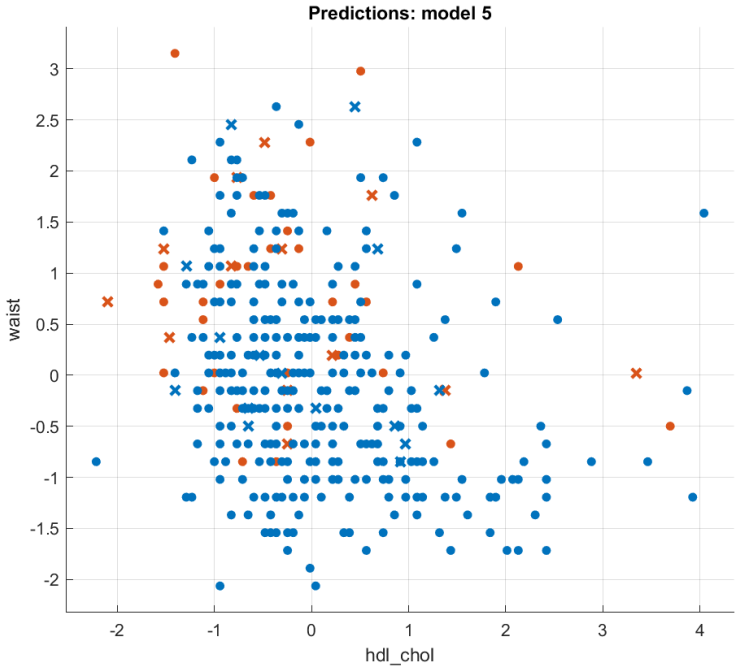
ISSN:2302-8149
e-ISSN:2540-9719

Tabel 7. Setting Parameter Algoritma

Parameter	KNN	SVM
Preset	Fine KNN	Fine Gaussian SVM
Cross-validation	5	5
Distance metrics	Consine	
Number of neighbor	4	
Distance weight	Equal	
Kernel function		Cubic
Kernel scale		2.77
Box constraint level		2
Multiclass method		One-VS-One
Standardize data	TRUE	TRUE

Tabel 7 merupakan parameter terbaik yang telah didapatkan pada algoritma KNN dan SVM, dari beberapa setting parameter yang telah diuji pada penelitian ini.

Proses pengembangan dataset dimulai dengan memasukkan dataset, scatter plot, dan nilai variabel x dan y kedalam software matlab, yang dapat ditunjukkan pada scatter plot yang terdapat pada Gambar 5.



Gambar 5. Scatter Plot

Pada Gambar 5. yang merupakan scatter plot, variabel x dapat dilambangkan dengan hdl_chol, dan untuk variabel y dapat dilambangkan dengan waist. Tahap pengujian data kemudian dilakukan dengan menggunakan software Matlab, selanjutnya setelah dilakukan klasifikasi, didapatkan hasil dari nilai kinerja algoritma seperti nilai True Positive (TP), True Negative (TN), False Positive (FP) dan False Negative (FN) yang terdapat pada Confusion Matrix. Nilai-nilai tersebut akan digunakan untuk mendefinisikan kinerja algoritma klasifikasi dalam menentukan nilai accuracy, recall, precision, dan F1-Score. Hasil dari nilai-nilai tersebut akan digunakan untuk membandingkan algoritma KNN dan SVM dalam menentukan algoritma terbaik yang akan digunakan dalam penelitian ini.

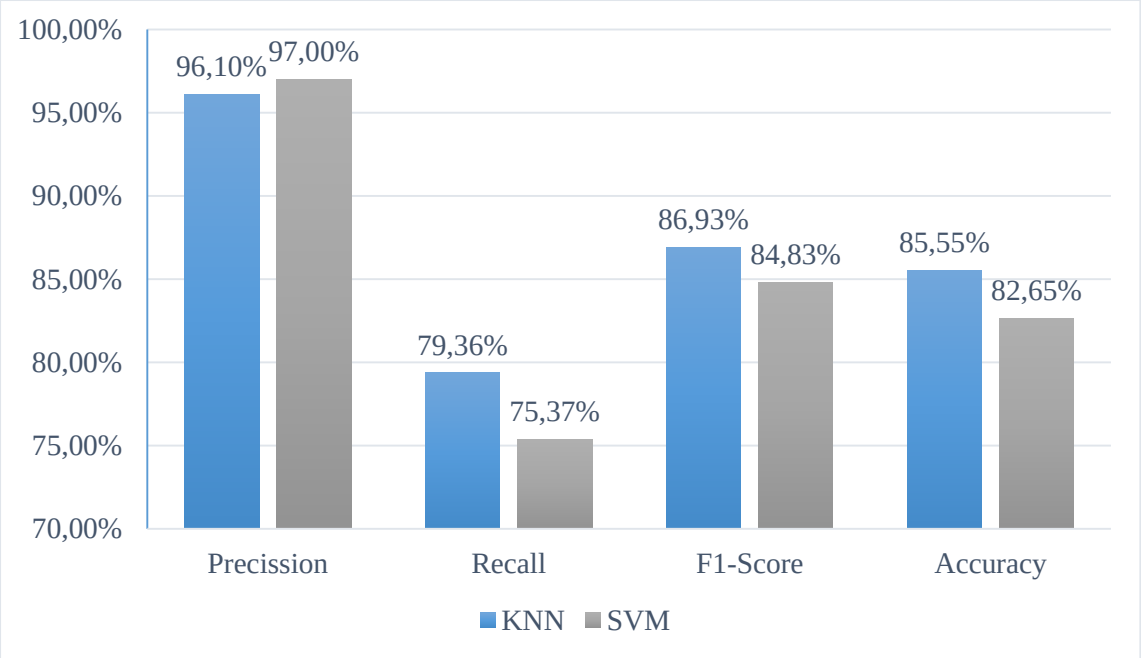
 ISSN:2302-8149
e-ISSN:2540-9719

Pada tahap ini akan dilakukan perbandingan algoritma KNN dan SVM untuk menentukan algoritma terbaik pada penelitian ini dengan menggunakan hasil dari nilai accuracy, recall, precision, dan F1-Score pada confusion matrix. Berikut ini adalah hasil dari perhitungan confusion matrix, dari hasil tertinggi K-NN dan SVM.

Tabel 8. Perbandingan KNN dan SVM

Algorithm	Precission	Recall	F1-Score	Accuracy
BDA + KNN	96.10%	79.36%	86.93%	85,55%
BDA + SVM	97,00%	75,37%	84,83%	82,65%

Pada Tabel 8 didapatkan nilai akurasi tertinggi dengan menggunakan algoritma BDA + KNN dibandingkan dengan menggunakan algoritma BDA + SVM. BDA + KNN mendapatkan nilai yaitu Precission 96,10%, Recall 79,36%, F-1 Score 86,93%, dan Accuracy 85,55%, Sementara untuk algoritma BDA dan SVM mendapatkan nilai Precission 97,00%, Recall 75,37%, F-1 Score 84,83%, dan Accuracy 82,65%.



Gambar 6. Diagram Perbandingan KNN dan SVM

Gambar 6 merupakan diagram perbandingan hasil dari pengujian algoritma BDA+KNN dan BDA+SVM. Dari diagram diatas terlihat bahwa algoritma BDA+KNN memiliki tingkat akurasi yang lebih baik dibandingkan dengan BDA+SVM, maka dari itu pada penelitian ini BDA+KNN memiliki nilai akurasi yang lebih baik dan lebih akurat dibandingkan BDA+SVM.

5 Kesimpulan

Berdasarkan hasil penelitian dan pengujian yang dilakukan pada penerapan algoritma optimasi Binary Dragonfly Algorthm (BDA), algoritma klasifikasi *K-Nearest Neighbor* (KNN) dan *Support Vector Machine* (SVM), didapatkan hasil bahwa algoritma BDA + KNN memberikan performa yang lebih baik dibandingkan dengan algoritma BDA + SVM pada dataset Diabetes yang digunakan dalam penelitian ini. Hasil pengujian menunjukkan bahwa BDA + KNN memiliki nilai Precission sebesar 96,10%, Recall sebesar 79,36%, F-1 Score sebesar 86,93%, dan Akurasi sebesar 85,55%. Sementara itu, BDA + SVM memiliki nilai Precission sebesar 97,00%, Recall sebesar 75,37%, F-1 Score sebesar84,83%, dan Akurasi sebesar 82,65%. Penelitian selanjutnya disarankan menggunakan dataset

yang berbeda seperti penyakit gagal jantung, kanker payudara, dan covid-19, sehingga mendapatkan hasil kesimpulan yang lebih akurat dalam melakukan perbandingan dari kedua algoritma tersebut.

Referensi

[1] R. Rafique, S. M. R. Islam, and J. U. Kazi, "Machine learning in the prediction of cancer therapy," *Comput. Struct. Biotechnol. J.*, vol. 19, pp. 4003–4017, 2021, doi: 10.1016/j.csbj.2021.07.003.

[2] A. C. Lyngdoh, N. A. Choudhury, and S. Moulik, "Diabetes Disease Prediction Using Machine Learning Algorithms," *Proc. - 2020 IEEE EMBS Conf. Biomed. Eng. Sci. IECBES 2020*, pp. 517–521, 2021, doi: 10.1109/IECBES48179.2021.9398759.

[3] M. E. Febrian, F. X. Ferdinan, G. P. Sendani, K. M. Suryanigrum, and R. Yunanda, "Diabetes prediction using supervised machine learning," *Procedia Comput. Sci.*, vol. 216, no. 2022, pp. 21–30, 2023, doi: 10.1016/j.procs.2022.12.107.

[4] A. Salam, Sri Suryani Prasetiyowati, and Yuliant Sibaroni, "Prediction Vulnerability Level of Dengue Fever Using KNN and Random Forest," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 3, pp. 531–536, 2020, doi: 10.29207/resti.v4i3.1926.

[5] M. T. Viega, "Heart Disease Prediction System using Data Mining Classification Techniques: Naïve Bayes, KNN, and Decision Tree," *Int. J. Adv. Trends Comput. Sci. Eng.*, vol. 9, no. 3, pp. 3028–3035, 2020, doi: 10.30534/ijatcse/2020/82932020.

[6] A. G. M. Sari, A. M. Putri, Z. Rustam, and J. Pandelaki, "Preprocessing Unbalanced Data using Support Vector Machine with Method K-Nearest Neighbors for Cerebral Infarction Classification," *J. Phys. Conf. Ser.*, vol. 1752, no. 1, 2021, doi: 10.1088/1742-6596/1752/1/012037.

[7] K. Devendran, S. K. Thangarasu, P. Keerthika, R. Manjula Devi, and B. K. Ponnarasee, "Effective prediction on music therapy using hybrid SVM-ANN approach," *ITM Web Conf.*, vol. 37, p. 01014, 2021, doi: 10.1051/itmconf/20213701014.

[8] L. Abuomar and K. Al-Aubidy, "Cooperative search and rescue with swarm of robots using binary dragonfly algoritlnn," *2018 15th Int. Multi-Conference Syst. Signals Devices, SSD 2018*, pp. 653–659, 2018, doi: 10.1109/SSD.2018.8570410.

[9] J. Too and S. Mirjalili, "A Hyper Learning Binary Dragonfly Algorithm for Feature Selection: A COVID-19 Case Study," *Knowledge-Based Syst.*, vol. 212, p. 106553, 2021, doi: 10.1016/j.knosys.2020.106553.

[10] X. Cui, Y. Li, J. Fan, T. Wang, and Y. Zheng, "A Hybrid Improved Dragonfly Algorithm for Feature Selection," *IEEE Access*, vol. 8, pp. 155619–155629, 2020, doi: 10.1109/ACCESS.2020.3012838.

[11] R. S. Raj, D. S. Sanjay, M. Kusuma, and S. Sampath, "Comparison of Support Vector Machine and Naïve Bayes Classifiers for Predicting Diabetes," *1st Int. Conf. Adv. Technol. Intell. Control. Environ. Comput. Commun. Eng. ICATIECE 2019*, pp. 41–45, 2019, doi: 10.1109/ICATIECE45860.2019.9063792.

[12] R. Katarya and S. Jain, "Comparison of different machine learning models for diabetes detection," *Proc. 2020 IEEE Int. Conf. Adv. Dev. Electr. Electron. Eng. ICADEE 2020*, no. Icadee, pp. 0–4, 2020, doi: 10.1109/ICADEE51157.2020.9368899.

[13] A. Naik, V. Kuppili, and D. Reddy Edla, "Binary dragonfly algorithm and fisher score based hybrid feature selection adopting a novel fitness function applied to microarray data," *Proc. - 2019 Int. Conf. Appl. Mach. Learn. ICAML 2019*, no. 1, pp. 40–43, 2019, doi: 10.1109/ICAML48257.2019.00015.

[14] A. Nugroho, H. L. H. S. Warnars, S. M. Isa, and W. Budiharto, "Comparison of Binary Particle Swarm Optimization And Binary Dragonfly Algorithm for Choosing the Feature Selection," *Proc. - Int. Conf. Informatics Comput. Sci.*, vol. 2021-Novem, no. October 2022, pp. 24–28, 2021, doi: 10.1109/ICICoS53627.2021.9651779.

[15] C. D. Patel, P. Bhayani, and B. Wankawala, "Multi-objective optimal PMU placement using binary dragonfly algorithm," *Proc. 2021 1st Int. Conf. Adv. Electr. Comput. Commun. Sustain. Technol. ICAECT 2021*, 2021, doi: 10.1109/ICAECT49130.2021.9392420.

[16] S. Behera, N. B. Dev Choudhury, D. Tripathy, R. Panda, and S. K. Bhagat, "Performance

 ISSN:2302-8149
e-ISSN:2540-9719

Improvement for Optimum Positioning of PMUs using Binary Dragonfly Algorithm and Loss Minimization of Transmission Network," *Proc. 2020 Int. Conf. Renew. Energy Integr. into Smart Grids A Multidiscip. Approach to Technol. Model. Simulation, ICREISG 2020*, pp. 254–257, 2020, doi: 10.1109/ICREISG49226.2020.9174442.

[17] H. Chantar, M. Tubishat, M. Essgaer, and S. Mirjalili, "Hybrid Binary Dragonfly Algorithm with Simulated Annealing for Feature Selection," *SN Comput. Sci.*, vol. 2, no. 4, pp. 1–11, 2021, doi: 10.1007/s42979-021-00687-5.

[18] M. H. Kakueinejad, A. Heydari, M. Askari, and F. Keynia, "Optimal planning for the development of power system in respect to distributed generations based on the binary dragonfly algorithm," *Appl. Sci.*, vol. 10, no. 14, 2020, doi: 10.3390/app10144795.

[19] S. Karan, E. N. Meese, Y. Yang, H. G. Yeh, C. G. Lowe, and W. Zhang, "Classification of Shark Behaviors using K-Nearest Neighbors," *2019 IEEE Green Energy Smart Syst. Conf. IGESSC 2019*, pp. 1–6, 2019, doi: 10.1109/IGESSC47875.2019.9042395.

[20] R. Puspadini, H. Mawengkang, and S. Efendi, "Feature Selection on K-Nearest Neighbor Algorithm Using Similarity Measure," *Mecn. 2020 - Int. Conf. Mech. Electron. Comput. Ind. Technol.*, pp. 226–231, 2020, doi: 10.1109/MECnIT48290.2020.9166612.

[21] B. Priambodo *et al.*, "Predicting GDP of Indonesia Using K-Nearest Neighbour Regression," *J. Phys. Conf. Ser.*, vol. 1339, no. 1, 2019, doi: 10.1088/1742-6596/1339/1/012040.

[22] Vibha and A. P. Singh, "Analysis of Variants of KNN Algorithm based on Preprocessing Techniques," *Proc. - IEEE 2018 Int. Conf. Adv. Comput. Commun. Control Networking, ICACCCN 2018*, pp. 186–191, 2018, doi: 10.1109/ICACCCN.2018.8748429.

[23] D. A. Ragab, M. Sharkas, S. Marshall, and J. Ren, "Breast cancer detection using deep convolutional neural networks and support vector machines," *PeerJ*, vol. 2019, no. 1, pp. 1–23, 2019, doi: 10.7717/peerj.6201.

[24] C. Aroef, Y. Rivan, and Z. Rustam, "Comparing random forest and support vector machines for breast cancer classification," *Telkomnika (Telecommunication Comput. Electron. Control.*, vol. 18, no. 2, pp. 815–821, 2020, doi: 10.12928/TELKOMNIKA.V18I2.14785.

[25] D. Tanouz, R. R. Subramanian, D. Eswar, G. V. P. Reddy, A. R. Kumar, and C. H. V. N. M. Praneeth, "Credit card fraud detection using machine learning," *Proc. - 5th Int. Conf. Intell. Comput. Control Syst. ICICCS 2021*, no. Iciccs, pp. 967–972, 2021, doi: 10.1109/ICICCS51141.2021.9432308.

[26] C. S. Eke, E. Jammeh, X. Li, C. Carroll, S. Pearson, and E. Ifeachor, "Early Detection of Alzheimer's Disease with Blood Plasma Proteins Using Support Vector Machines," *IEEE J. Biomed. Heal. Informatics*, vol. 25, no. 1, pp. 218–226, 2021, doi: 10.1109/JBHI.2020.2984355.

[27] R. Arian, A. Hariri, A. Mehridehnavi, A. Fassihi, and F. Ghasemi, "Protein kinase inhibitors' classification using K-Nearest neighbor algorithm.," *Comput. Biol. Chem.*, vol. 86, no. March, p. 107269, Jun. 2020, doi: 10.1016/j.compbiolchem.2020.107269.

[28] S. Ravikumar and P. Saraf, "Prediction of stock prices using machine learning (regression, classification) Algorithms," *2020 Int. Conf. Emerg. Technol. INCET 2020*, pp. 1–5, 2020, doi: 10.1109/INCET49848.2020.9154061.

[29] Z. Faraji, "A Review of Machine Learning Applications for Credit Card Fraud Detection with A Case study," *SEISENSE J. Manag.*, vol. 5, no. 1, pp. 49–59, 2022, doi: 10.33215/sjom.v5i1.770.