

Perbandingan Kinerja KNN dan Random Forest dengan Optimasi Hiperparameter PSO untuk Klasifikasi Mpox.

Lima Hartimar Rambe, Rika Rosnelly, Wanayumini

Program Studi Magister Ilmu Komputer, Fakultas Teknik Dan Ilmu Komputer
Universitas Potensi Utama Medan

limahartimarrambe12345@gmail.com; rikarosnelly@potensi-utama.ac.id;
wanayumini@potensi@gmail.com

Abstrak

Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja algoritma K-Nearest Neighbors (KNN) dan Random Forest (RF) dalam mengklasifikasikan Monkeypox dengan menerapkan teknik optimasi Particle Swarm Optimization (PSO). PSO digunakan untuk memperoleh parameter optimal pada masing-masing algoritma sehingga mampu meningkatkan akurasi dan efisiensi model klasifikasi. Dataset yang digunakan adalah data sekunder yang diperoleh dari Kaggle yang dipublikasikan oleh M.Ahmed pada tahun 2023, data observasi yang digunakan sebanyak 10.000 dengan 11 fitur gejala klinis yang merepresentasikan kondisi kesehatan terkait infeksi Mpox. Evaluasi performa model dilakukan sebelum dan sesudah proses optimasi menggunakan metrik akurasi, presisi, recall, F1-score, dan sensitivitas. Hasil penelitian menunjukkan penerapan PSO mampu meningkatkan kinerja kedua algoritma dibandingkan dengan model tanpa optimasi. Akurasi KNN meningkat dari 78% menjadi 80%, sedangkan Random Forest meningkat dari 65% menjadi 69%. Hasil tersebut menunjukkan bahwa PSO dapat membantu memperoleh konfigurasi parameter yang lebih optimal pada kedua algoritma. Penelitian ini memberikan kontribusi dalam menyediakan perbandingan empiris kinerja KNN dan Random Forest berbasis optimasi PSO serta dapat menjadi referensi dalam pengembangan sistem klasifikasi berbasis machine learning untuk mendukung identifikasi awal Mpox..

Kata Kunci : Pembelajaran Mesin, K-Nearest Neighbors, Random Forest, Particle Swarm Optimization, Klasifikasi Cacar Monyet.

I. PENDAHULUAN

Mpox merupakan penyakit infeksi yang dapat menimbulkan gejala seperti demam, nyeri otot, pembengkakan kelenjar getah bening, serta lesi atau ruam pada kulit. Skrining penting dilakukan untuk membantu mengidentifikasi individu yang memiliki karakteristik gejala Mpox dan menentukan kebutuhan pemeriksaan lebih lanjut. Namun, beberapa gejalanya memiliki kemiripan dengan penyakit atau kondisi lain sehingga klasifikasi hanya berdasarkan pengamatan gejala dapat menjadi sulit. Oleh karena itu, diperlukan metode klasifikasi yang mampu menganalisis berbagai fitur gejala secara bersamaan untuk membantu membedakan observasi positif dan negatif Mpox.

K-Nearest Neighbors (KNN) merupakan salah satu algoritma pembelajaran mesin yang paling populer karena sederhana, mudah diimplementasikan, dan memiliki kemampuan klasifikasi yang baik. Algoritma ini melakukan klasifikasi berdasarkan kedekatan jarak antara data baru dengan sejumlah tetangga terdekat (*nearest neighbors*) pada data pelatihan. Kelas dari data baru ditentukan berdasarkan mayoritas kelas dari tetangga terdekat tersebut. Oleh karena itu, pemilihan nilai K sangat memengaruhi tingkat akurasi model. [1]

Selain itu, KNN memiliki tingkat fleksibilitas yang tinggi karena dapat diterapkan pada berbagai jenis data selama tersedia ukuran kemiripan

(*similarity measure*) yang sesuai. Hal tersebut memungkinkan KNN digunakan pada berbagai bidang, seperti klasifikasi teks, citra, maupun data deret waktu (*time series*). Kelebihan lainnya adalah hasil klasifikasi dapat dijelaskan secara langsung berdasarkan data tetangga yang digunakan dalam proses pengambilan keputusan. [2]

Sementara itu, Random Forest merupakan salah satu algoritma pembelajaran mesin yang banyak digunakan untuk menyelesaikan permasalahan klasifikasi maupun regresi. Algoritma ini memiliki kemampuan yang baik dalam menangani data numerik, kategorikal, maupun data yang mengandung nilai hilang (*missing values*). Random Forest dibangun dari kumpulan pohon keputusan (*decision tree*) yang bekerja secara bersama-sama sehingga menghasilkan prediksi yang lebih stabil dan memiliki kemampuan generalisasi yang lebih baik dibandingkan satu pohon keputusan tunggal. [3]

Beberapa penelitian sebelumnya telah menggunakan algoritma K-Nearest Neighbors dan Random Forest untuk klasifikasi penyakit. Hamdan et al. (2024) melakukan penelitian menggunakan dataset cacar monyet dengan membandingkan performa berbagai algoritma pembelajaran mesin, termasuk KNN dan Random Forest. Dataset dibagi menjadi data pelatihan dan data pengujian, kemudian dievaluasi menggunakan metrik akurasi, presisi, recall, F1-score, dan sensitivitas. Hasil penelitian menunjukkan bahwa algoritma KNN memiliki performa yang lebih baik

dengan nilai evaluasi yang lebih seimbang dibandingkan Random Forest, yang memperoleh nilai paling rendah pada hampir seluruh metrik evaluasi.

Di sisi lain, Particle Swarm Optimization (PSO) merupakan metode optimasi yang terinspirasi dari perilaku sosial sekumpulan hewan, seperti burung, ikan, dan lebah, ketika mencari makanan. Algoritma ini diperkenalkan pada tahun 1995 dan banyak digunakan untuk menyelesaikan berbagai permasalahan optimasi, termasuk optimasi fungsi, pelatihan jaringan saraf tiruan, serta pengendalian sistem fuzzy.

Penelitian oleh Hadjaidji et al. (2025) [4] mengevaluasi dan membandingkan beberapa algoritma pembelajaran mesin dalam mendeteksi penyakit Parkinson. Penelitian tersebut menerapkan PSO untuk mengoptimalkan bobot fitur sehingga diperoleh performa klasifikasi yang lebih baik. Algoritma yang dibandingkan meliputi K-Nearest Neighbors (KNN), Support Vector Machine (SVM), Gradient Boosting (GB), dan Naïve Bayes. Dataset yang digunakan terdiri atas 197 data suara dengan 23 fitur akustik, sedangkan tahap pra-proses meliputi normalisasi menggunakan Z-Score, penyeimbangan data menggunakan SMOTE (Synthetic Minority Over-sampling Technique), serta evaluasi menggunakan k-fold cross validation. [5]

Berdasarkan penelitian-penelitian sebelumnya, terlihat bahwa KNN memiliki performa yang baik dalam klasifikasi penyakit, sedangkan PSO mampu meningkatkan kinerja berbagai algoritma pembelajaran mesin melalui proses optimasi parameter, namun belum ada penelitian sebelumnya yang mengkombinasikan PSO dengan algoritma KNN dan Random Forest pada proses pencarian hyperparameter terbaik di parameter yang digunakan. Oleh karena itu, penelitian ini bertujuan untuk membandingkan performa algoritma K-Nearest Neighbors dan Random Forest, baik sebelum maupun sesudah dilakukan optimasi menggunakan Particle Swarm Optimization (PSO) berdasarkan hyperparameter terbaik dari algoritma yang digunakan. [6]

Dalam penelitian ini, kedua algoritma hanya dapat memproses data numerik sehingga dilakukan tahap pra-proses berupa encoding untuk mengubah data kategorikal menjadi numerik, pembersihan data (cleaning) untuk menghilangkan data yang tidak valid, serta pembagian data menjadi data pelatihan dan data pengujian. Seluruh tahapan tersebut dilakukan agar data siap digunakan dalam proses pembangunan model klasifikasi.

Selanjutnya, performa masing-masing model dibandingkan menggunakan metrik akurasi, presisi, recall, F1-score, dan sensitivitas. Melalui evaluasi tersebut diharapkan dapat diketahui model klasifikasi yang memiliki performa terbaik setelah proses optimasi.

Pendukung skrining awal penyakit cacar monyet sangat penting mengingat penyakit ini telah menyebar ke berbagai negara dan memiliki

gejala yang mirip dengan beberapa penyakit lain, seperti demam, sakit kepala, dan nyeri otot. Oleh karena itu, penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem deteksi penyakit cacar monyet berbasis pembelajaran mesin yang lebih akurat dan efektif.

II. TINJAUAN PUSTAKA

2.1 Penelitian Terkait

Penelitian yang dilakukan oleh Hamdan et al. (2024) membandingkan performa beberapa algoritma machine learning dalam klasifikasi penyakit Monkeypox, termasuk K-Nearest Neighbors dan Random Forest. Hasil penelitian menunjukkan bahwa algoritma KNN menghasilkan nilai akurasi, recall, dan F1-score yang lebih baik dibandingkan Random Forest sehingga dinilai lebih efektif dalam mendeteksi kasus Monkeypox.

Penelitian lain oleh Hadjaidji et al. (2025) [4] menerapkan Particle Swarm Optimization (PSO) untuk mengoptimalkan parameter beberapa algoritma machine learning pada deteksi penyakit Parkinson. Hasil penelitian menunjukkan bahwa optimasi menggunakan PSO mampu meningkatkan performa model melalui penentuan bobot fitur dan parameter yang lebih optimal.

Berdasarkan penelitian-penelitian tersebut, dapat disimpulkan bahwa algoritma KNN memiliki performa yang baik pada klasifikasi data medis, sedangkan PSO terbukti efektif dalam meningkatkan kinerja berbagai algoritma pembelajaran mesin. Oleh karena itu, penelitian ini mengombinasikan algoritma KNN dan Random Forest dengan metode PSO untuk mengevaluasi peningkatan performa klasifikasi pada dataset Monkeypox.

2.2 Machine Learning

Machine Learning merupakan cabang dari kecerdasan buatan yang memungkinkan komputer mempelajari pola dari data sehingga dapat melakukan prediksi maupun klasifikasi secara otomatis. Dalam bidang kesehatan, machine learning telah banyak dimanfaatkan untuk membantu proses diagnosis penyakit karena mampu mengolah data dalam jumlah besar dengan tingkat akurasi yang tinggi [7]

2.3 Algoritma K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) merupakan algoritma klasifikasi yang menentukan kelas suatu data berdasarkan kedekatan jarak terhadap sejumlah tetangga terdekat pada data pelatihan. Nilai K menunjukkan jumlah tetangga yang digunakan dalam proses klasifikasi. Pemilihan nilai K sangat memengaruhi performa model karena menentukan keseimbangan antara sensitivitas dan kemampuan generalisasi model. Selain itu, algoritma ini dapat menggunakan berbagai ukuran jarak, seperti Euclidean Distance

maupun Manhattan Distance, sesuai karakteristik data yang digunakan. [8]

2.4 Algoritma Random Forest

Random Forest merupakan algoritma berbasis ensemble learning yang dibangun dari sekumpulan pohon keputusan (*decision tree*). Setiap pohon dibentuk menggunakan sampel data dan subset fitur yang dipilih secara acak, kemudian hasil prediksi diperoleh melalui mekanisme *majority voting*. Pendekatan tersebut membuat Random Forest memiliki kemampuan yang baik dalam menangani data numerik maupun kategorikal serta mampu mengurangi risiko *overfitting* pada proses klasifikasi. [9]

2.5 Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) merupakan algoritma optimasi yang diperkenalkan oleh Kennedy dan Eberhart pada tahun 1995. Algoritma ini terinspirasi dari perilaku sosial kawanan burung atau ikan dalam mencari sumber makanan. Dalam proses optimasi, setiap partikel mewakili solusi kandidat yang bergerak menuju posisi terbaik berdasarkan pengalaman individu (*personal best*) dan pengalaman kelompok (*global best*). Pada penelitian ini, PSO digunakan untuk menentukan parameter optimal pada algoritma KNN dan Random Forest sehingga mampu meningkatkan performa klasifikasi. [10]

Salah satu keunggulan PSO adalah algoritma ini memiliki struktur yang sederhana, mudah diimplementasikan, serta hanya memerlukan sedikit parameter untuk proses optimasi dibandingkan metode optimasi lainnya. Selain itu, PSO memiliki kemampuan konvergensi yang baik, tidak memerlukan informasi turunan (*derivative*) dari fungsi objektif, serta mampu diterapkan pada berbagai jenis permasalahan optimasi dengan ruang pencarian yang kompleks. Oleh karena itu, PSO banyak dimanfaatkan sebagai metode optimasi parameter pada berbagai algoritma pembelajaran mesin untuk meningkatkan performa model klasifikasi maupun regresi.

2.6 Monkeypox

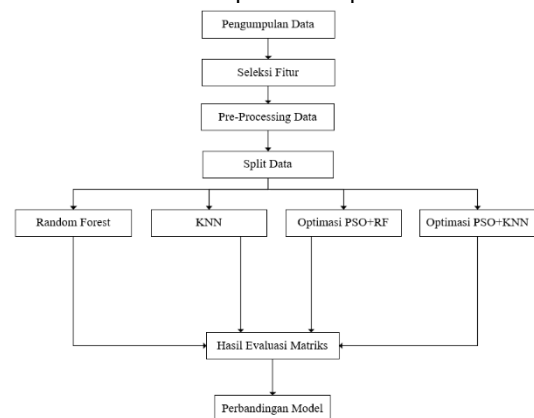
Monkeypox merupakan penyakit infeksi yang disebabkan oleh virus Monkeypox dari genus *Orthopoxvirus*. Penyakit ini ditandai dengan gejala seperti demam, sakit kepala, nyeri otot, pembengkakan kelenjar getah bening, serta munculnya lesi atau ruam pada kulit. Karena memiliki gejala yang hampir sama dengan beberapa penyakit infeksi lainnya, diperlukan metode diagnosis yang memiliki tingkat akurasi tinggi agar penanganan pasien dapat dilakukan secara tepat. [11]

III. METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Tahapan penelitian merupakan rangkaian proses yang dilakukan secara sistematis dalam pelaksanaan penelitian untuk mencapai tujuan yang telah ditetapkan. Pada penelitian ini, tahapan dimulai dari proses pengumpulan data yang digunakan sebagai bahan analisis, kemudian dilanjutkan dengan tahap *preprocessing data* untuk membersihkan dan mempersiapkan data sebelum dilakukan proses pengolahan lebih lanjut. Setelah data diproses, dilakukan pembagian data menjadi data pelatihan (*training*) dan data pengujian (*testing*) yang digunakan untuk membangun serta menguji model klasifikasi.

Selanjutnya, dilakukan penerapan metode klasifikasi yaitu K-Nearest Neighbor (KNN) dan Random Forest sebelum dan sesudah di optimasi menggunakan PSO (*Particle Swarm Optimization*) untuk pengklasifikasian dataset cacar monyet untuk memprediksi kasus positif maupun kasus negative. Hasil dari masing-masing metode kemudian dievaluasi menggunakan metrik evaluasi kinerja seperti *accuracy*, *precision*, *recall*, dan F1-score. Tahap terakhir adalah melakukan analisis perbandingan terhadap hasil kinerja kedua algoritma untuk mengetahui metode yang memiliki performa terbaik dalam mengklasifikasikan cacar monyet baik hasil positif maupun hasil negative, kemudian ditarik kesimpulan dari hasil penelitian yang telah dilakukan. Adapun tahapan penelitian secara keseluruhan dapat dilihat pada **Gambar 1**.



Gambar 1. Tahapan Penelitian

3.2 Pengumpulan Data

Pada penelitian ini, data yang digunakan merupakan dataset cacar monyet yang tersedia secara publik dan dipublikasikan oleh M.Ahmed pada tahun 2023 di Kaggle melalui situs web [kaggle.com/datasets? search=data+](https://www.kaggle.com/datasets/search=data+). Dataset tersebut berisi data gejala pasien secara nyata. Data terdiri dari 11 fitur seperti (*Patient_ID*, *Systemic Illness*, *Rectal Pain*, *Sore Throat*, *Penile Oedema*, *Oral Lesions*, *Solitary Lesion*, *Swollen Tonsils*, *HIV Infection*, *Sexually Transmitted Infection*, *MonkeyPox*) yang berjumlah sebanyak 10000 data. Dataset yang digunakan terdiri dari beberapa Fitur

MonkeyPox merupakan label target klasifikasi cacar monyet, positif menunjukkan pasien tersebut terkena cacar, sedangkan negatif menunjukkan tidak terkena cacar. Berikut ini ditampilkan *table* dataset cacar monyet pada Tabel 1.

Tabel 1. Fitur Dataset

Atribut	Tipe Data	Keterangan
Patient_ID	String	ID Pasien
Systemic Illness	Kategorik	Gejala
Rectal Pain	Boolean	Gejala
Sore Throat	Boolean	Gejala
Penile Oedema	Boolean	Gejala
Oral Lesions	Boolean	Gejala
Solitary Lesion	Boolean	Gejala
Swollen Tonsils	Boolean	Gejala
HIV Infection	Boolean	Gejala
Sexually Transmitted Infection	Boolean	Gejala
MonkeyPox	Object	Kelas Target

Dataset ini memiliki 11 fitur diantaranya ada 9 fitur gejala klinis dengan tipe data Boolean, 1 fitur ID dengan tipe data String, dan 1 fitur object yang jadi penentu hasilklasifikasi positif atau negative.

3.3 Preprocessing

Pada tahapan ini dilakukan *pre-processing* data yang bertujuan untuk mengoptimalkan model dalam menganalisis objek. Tahapan *pre-processing* pada dataset cacar monyet ini dilakukan 4 tahapan yaitu pembersihan data (*data cleaning*), pengecekan *missing value* dan pengecekan fitur duplikat, pelabelan (*encoding*) untuk mengubah data numerik jadi angka biner, seleksi fitur menemukan fitur mana yg korelasinya lebih berpengaruh ke kelas target, split data untuk membagi data latih dan data uji, kemudian dilakukan resampling pada data latih untuk menanganikan jumlah kelas yang tidak seimbang.

3.3.1 Pembersihan Data (*Data Cleaning*)

Setelah proses *preprocessing* data selesai Pada tahap pembersihan data (*data cleaning*), setelah dilakukan pengecekan *missing value* dan pengecekan data duplikat. Dilakukan seleksi fitur untuk mengidentifikasi hubungan atau korelasi antara fitur gejala dengan kelas target, sehingga diharapkan dapat meningkatkan kinerja model klasifikasi. Proses pelabelan (*encoding*) dilakukan dalam tiga tahap sesuai dengan perbedaan tipe data. Data bertipe kategorikal diubah menjadi data numerik menggunakan teknik *label encoding*. Untuk data gejala bertipe *boolean* digunakan teknik *one-hot encoding*. Sementara itu, kelas target juga diubah menjadi data numerik menggunakan teknik *label encoding*.

3.4 Split Data

Pada tahap ini Data dibagi menjadi 2 bagian data *training* data *testing*. Data *training* digunakan untuk membangun model klasifikasi. Pembagian data dibuat dengan ukuran percobaan yang berbeda, dimana data *training* sebesar 80% dan *testing* 20%. Data *training* digunakan sebagai pembentuk pola atau model, sedangkan data *testing* digunakan sebagai pengujian model yang telah dibangun. Model yang digunakan dalam pengujian data ini adalah klasifikasi algoritma KNN dan *Random forest* menggunakan teknik optimasi *particle swarm* (PSO).

3.5 Pelabelan (*Encoding*)

Proses *encoding* dilakukan dengan mengubah tipe data kategorikal menjadi data numerik menggunakan teknik *label encoding*. Pada proses pelabelan, dataset dengan tipe data *objeck* perlu dikonversi ke numerik, karena algoritma machine learning yang digunakan seperti KNN dan *random forest* tidak bisa memproses teks secara langsung. Berdasarkan 11 fitur yang tersedia pada dataset cacar monyet ini, terdapat 2 kolom dengan tipe data objek dan 9 fitur dengan tipe data boolean. Fitur *systemic illness* dan *monkeypox* yang awalnya berupa teks ("Yes"/"No"), telah diubah menjadi angka, 0 = Tidak, 1 = Ya. Fitur *Monkeypox* yang awalnya berupa teks ("Positive"/"Negative") telah di-encode menjadi, 1 = Positive, 0 = Negative. Fitur gejala lainnya dengan tipe data Boolean diubah ke dalam bentuk numerik dengan teknik *one hot encoding*.

3.6 Seleksi Fitur

Digunakan pendekatan uji *chi-squared* (Chi-Square Test) untuk menilai tingkat relevansi setiap fitur terhadap variabel target MonkeyPox. Seluruh fitur kategorikal yang telah dikodekan ke dalam bentuk biner dianalisis menggunakan uji Chi-Squared guna mengukur adanya hubungan atau ketergantungan antara masing-masing fitur dengan kelas target. Berikut langkah-langkah perhitungan manualnya.

1. Buat Tabel Kontingensi (Observed Frequencies): Ini adalah tabel silang yang menampilkan frekuensi gabungan dari setiap kategori dari dua variabel dari fitur ke kelas target.
2. Hitung Expected Frequencies: Ini adalah frekuensi yang diharapkan jika tidak ada hubungan antara kedua variabel (yaitu, jika mereka independen). Rumusnya Adalah pada persamaan 3.1 sebagai berikut:
- 3.

$$E_{ij} = \frac{(R_i \times C_j)}{N} \quad (3.1)$$

Keterangan:

- E_{ij} = frekuensi harapan pada baris

ke-i dan kolom ke-j

- R_i = total frekuensi pada baris ke-i
- C_j = total frekuensi pada kolom ke-j
- N = total seluruh data (grand total)

- Hitung Statistik Chi-squared: menggunakan rumus chi pada persamaan 3.2 sebagai berikut.

$$\chi^2 = \sum_i = 1r \sum_j = 1c E_{ij} (O_{ij} - E_{ij})^2$$

O_{ij} merupakan frekuensi observasi dan E_{ij} merupakan frekuensi harapan yang diperoleh dari hasil perkalian total baris dan total kolom dibagi dengan jumlah keseluruhan data. Tentukan Derajat Kebebasan (Degrees of Freedom - df): df = (jumlah baris - 1) (jumlah kolom - 1)

- Hitung P-value: Dengan menggunakan nilai Chi-squared yang dihitung dan derajat kebebasan, kita dapat mencari p-value dari tabel distribusi Chi-squared atau menggunakan perangkat lunak statistik. P-value ini menunjukkan probabilitas mengamati statistik Chi-squared setinggi atau lebih tinggi dari yang kita hitung, jika hipotesis nol (tidak ada hubungan) adalah benar.

3.8 Resampling/ SMOTE

Untuk mengatasi ketidakseimbangan kelas digunakan teknik **SMOTE (Synthetic Minority Over-sampling Technique)**. SMOTE bertujuan untuk menyeimbangkan jumlah sampel antara kelas **positif** dan **negatif**, sehingga model tidak bias terhadap kelas mayoritas. SMOTE diterapkan pada **data pelatihan (X_train, y_train)** dengan cara mensintesis sampel baru pada kelas minoritas melalui interpolasi antar data yang mirip. Setelah proses ini, distribusi kelas pada data pelatihan menjadi seimbang (50% : 50%).

IV. PENGUMPULAN DATA DAN HASIL

4.1 Pengumpulan Data

Pada tahap ini dilakukan proses pengumpulan dan pemuatan dataset yang digunakan dalam penelitian. Dataset yang digunakan dalam penelitian ini merupakan data sekunder yang diperoleh dari platform *kaggle*. Dataset berisi gejala klinis pasien cacar monyet (*monkeypox*) yang di klasifikasi 2 kategori yaitu positif dan negatif. Dataset terdiri dari 10.000 data dengan 11 fitur gejala klinis seperti: *Patient_ID*, *Systemic Illness*, *Rectal Pain*, *Sore Throat*, *Penile Oedema*, *Oral Lesions*, *Solitary Lesion*, *Swollen Tonsils*, *HIV Infection*, *Sexually Transmitted Infection*, *MonkeyPox*.

4.2 Hasil Preprocessing

Sebelum data melakukan proses modelling, ada beberapa teknik pre-processing data yang dilakukan untuk mengoptimalkan hasil modelling, tahapan pertama berawal dari proses pembersihan data, selanjutnya dilakukan teknik encoding pada 3 tipe data yang berbeda, untuk tipe data string dan object (target) dilakukan label encoding, dan untuk tipe data kategorik dilakukan *one-hot encoding*.

Dari hasil encoding ini mengubah bentuk numerik dengan representasi 0 untuk kondisi “tidak” dan 1 untuk kondisi “ya. Fitur fitur kategorikal seperti *Systemic Illness* telah menambahkan beberapa kolom baru pada tabel paling kanan.

4.3 Seleksi Fitur

Menggunakan pendekatan *chi-squared test* untuk menilai relevansi setiap fitur terhadap variabel target *MonkeyPox*. Fitur kategorikal yang telah dikodekan dalam bentuk biner, diterapkan Uji Chi-kuadrat untuk mengukur korelasi/hubungan fitur dengan kelas target. Langkah awal metode ini dengan membangun tabel kontingensi antara fitur dengan variabel target. Kemudian, dilakukan pengujian *Chi-Squared* untuk memperoleh nilai statistik uji dan p-value dari setiap fitur. Hasil pengujian tersebut kemudian diurutkan berdasarkan nilai p-value untuk memudahkan analisis tingkat signifikansi masing-masing fitur. Nilai $p < 0.05$ yang sudah ditetapkan sebagai ambang batas dianggap memiliki hubungan yang signifikan dengan variabel target dan dinyatakan relevan. Sebaliknya, fitur yang memiliki nilai p-value lebih besar dari 0,05 dianggap kurang relevan dan dihapus dari dataset. Untuk melihat hubungan semua fitur ke kelas target, maka ditampilkan hasil analisis seleksi fitur pada Tabel 2.

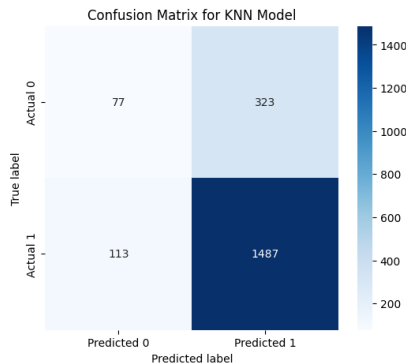
Tabel 2. Hasil Seleksi Fitur

MonkeyPox	
MonkeyPox	1.000000
Systemic_Illness_Muscle Aches and Pain	0.199703
HIV Infection	0.119110
Systemic_Illness_Fever	0.098724
Rectal Pain	0.098636
Systemic_Illness_Swollen Lymph Nodes	0.091971
Sexually Transmitted Infection	0.091227
Sore Throat	0.071119
Oral Lesions	0.039803
Penile Oedema	0.038008
Solitary Lesion	0.016902
Swollen Tonsils	0.010401

Dari Tabel 2 ini terlihat masing-masing fitur memiliki nilai yang berbeda beda, dari 11 fitur yang di cek korelasinya, 4 fitur (*Swollen Tonsils*, *Penile Oedema*, *Oral Lesions*, *Sore Throat*) memiliki nilai P dibawah 0.05 maka dilakukan penghapusan fitur secara manual berdasarkan nilai p yang sudah ditentukan.

4.4 Pemodelan

Pada model *k-nearest neighbors* (KNN) Model dibangun menggunakan parameter *default* digunakan dalam penelitian klasifikasi berbasis jarak. Parameter utama yang digunakan adalah $k = 5$ jumlah tetangga terdekat yang digunakan untuk menentukan kelas suatu sampel. Jarak (p) = 2 menunjukkan bahwa model memakai jarak Euclidean, yaitu metrik yang paling umum digunakan pada KNN karena mampu mengukur kedekatan antar data secara geometris dalam ruang multidimensi. Untuk melihat hasil yang lebih detail berikut divisualisasikan dalam bentuk gambar pada Gambar 2. KNN



Gambar 2. Model KNN

Confusion matrix memberikan visualisasi detail mengenai distribusi prediksi model dimana:

- True Negative (TN): 77 kasus
- False Positive (FP): 323 kasus
- False Negative (FN): 113 kasus
- True Positive (TP): 1487 kasus

Hasil evaluasi KNN memberikan akurasi sebesar 0.78(78%). Nilai presisi 0.82(82)% yang mengindikasikan bahwa sebagian besar prediksi positif yang dihasilkan model adalah benar, sehingga jumlah *false positive* relatif rendah. Sementara itu, recall 0.93(93)% menunjukkan bahwa model mampu mendeteksi hampir seluruh kasus positif pada dataset, sehingga risiko terlewatnya kasus *monkeypox* sangat kecil. Kombinasi presisi dan *recall* menghasilkan *F1-score* sebesar 0.87(87%), yang menegaskan bahwa model memiliki keseimbangan performa yang baik dalam mendeteksi dan memverifikasi kasus positif. Berikut ini contoh perhitungan manual untuk mencari nilai confusion matrix.

$$\text{Accuracy} = \frac{1487 + 77}{1487 + 77 + 323 + 113} = \frac{1564}{2000} = 0.782$$

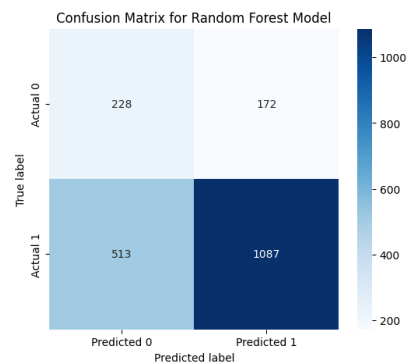
$$\text{Precision} = \frac{1487}{1487 + 323} = \frac{1810}{1487} = 0,821$$

$$\text{Recall} = \frac{1487}{1487 + 113} = \frac{1487}{1600} = 0,929$$

$$F1 = \frac{1,527}{1,7509} = 0,872$$

Hasil perhitungan manual akurasi sebesar 0.782 ini menunjukkan hasil yang hampir sama dengan hasil program pada python, terjadi perbedaan sedikit dikarenakan adanya pembulatan dan hanya memakai 2 angka dibelakang koma di setiap matrix.

Pada tahap klasifikasi *random forest* dibangun menggunakan beberapa parameter seperti *n_estimators* (100), *min_samples_split*(2), *min_samples_leaf*(1), dan *max_depth*(none), dan *Random_state* (42) untuk memastikan proses pelatihan bersifat *reproducible*. Setelah proses *pre-processing* data selesai kemudian model *random forest* di uji menggunakan parameter yang sudah ditentukan sebelumnya. Untuk melihat hasil yang lebih detail berikut divisualisasikan dalam bentuk gambar pada Gambar 3. Model Random Forest



Gambar 3. Model Random Forest

Confusion matrix memberikan visualisasi detail mengenai distribusi prediksi model dimana:

- Confusion matrix:
 - True Negatives (TN): 228
 - False Positives (FP): 172
 - False Negatives (FN): 513
 - True Positives (TP): 1087

Hasil evaluasi model *Random Forest* memperoleh akurasi sebesar 0.65(65%) yang berarti model mampu mengklasifikasikan sebagian besar data dengan benar namun masih terdapat ruang perbaikan. Presisi sebesar 0.86(86%) menunjukkan bahwa sebagian besar prediksi positif yang dibuat oleh model adalah benar, sehingga *false positive* relatif rendah.

4.5 Particle Swarm Optimization

Particle Swarm Optimization (PSO) digunakan untuk mencari parameter terbaik dari algoritma KNN dan random forest.

Pada proses optimasi algoritma *k-nearest neighbors* (KNN) terdapat tiga parameter utama yang dioptimalkan *n_neighbors*: 18, *p*: 1, *weights*: *uniform*. Nilai $k = 18$ akan mempertimbangkan 18 tetangga terdekat dari titik data tersebut. Setelah melalui proses iterasi dengan jumlah iterasi 40 Hasil optimasi menunjukkan bahwa nilai $k = 18$

memberikan performa terbaik, yang berarti model mencapai keseimbangan antara variansi dan bias. Dengan kata lain, nilai $k = 18$ memberikan representasi yang cukup stabil bagi model dalam menentukan kelas dari setiap sampel baru. Setelah hasil optimal dari parameter KNN diperoleh kemudian model diuji kembali menggunakan parameter optimal dengan akurasi 0.80(80%), presisi 0.81(81%), recall 0.98(98%), dan f1-score 0.89(89%).

Pada algoritma Random forest PSO digunakan untuk mengoptimalkan empat parameter utama *random forest*, yaitu *n_estimators* 145, *max_depth* 4, *min_samples_split* 4, dan *min_samples_leaf* 2. *N_estimators* membantu model lebih kuat dan mampu menangani variasi data dengan lebih baik serta mengurangi risiko *overfitting*. Proses optimasi dijalankan selama 40 iterasi, di mana setiap partikel dalam swarm membawa kombinasi nilai keempat parameter tersebut sebagai posisinya di ruang pencarian. Pada setiap iterasi, PSO mengevaluasi performa model yang dihasilkan menggunakan fungsi objektif. Kemudian posisi partikel diperbarui berdasarkan pengalaman terbaik individu (pBest) dan pengalaman terbaik dari seluruh swarm (gBest), sehingga partikel secara bertahap bergerak menuju solusi parameter yang paling optimal. Nilai matrix yang diperoleh menunjukkan kenaikan di beberapa matrix dengan nilai akurasi 0.69(69%), presisi 0.86(86%), recall 0.73(0.73), dan f1 score 0.79(79%).

V. PEMBAHASAN

5.1 Dataset

dataset yang digunakan dalam penelitian. Dataset yang digunakan dalam penelitian ini merupakan data sekunder secara nyata yang diperoleh dari platform *kaggle*, yang dipublikasikan oleh M.Ahmed pada tahun 2023. Dataset berisi gejala klinis pasien cacar monyet (*monkeypox*) yang terdiri dari 10.000 data dengan 11 fitur, 9 fitur gejala klinis seperti: *Patient_ID*, *Systemic Illness*, *Rectal Pain*, *Sore Throat*, *Penile Oedema*, *Oral Lesions*, *Solitary Lesion*, *Swollen Tonsils*, *HIV Infection*, *Sexually Transmitted Infection*, *MonkeyPox*, 1 fitur ID sebagai identitas dan 1 fitur *Monkeypox* sebagai kelas target klasifikasi. Fitur-fitur tersebut dipilih karena relevan dalam proses identifikasi awal infeksi *Monkeypox*.

5.2 Preprocessing

Tahap pembersihan data dilakukan untuk memastikan bahwa dataset yang digunakan memiliki kualitas dan konsistensi yang baik sebelum masuk ke tahap pemodelan. Pemeriksaan dilakukan terhadap seluruh 11 dengan fokus pada keberadaan *missing value* dan data duplikat. Berdasarkan hasil pengecekan, tidak ditemukan adanya nilai yang hilang pada seluruh atribut, sehingga proses imputasi atau penghapusan data

tidak diperlukan. Kondisi ini menunjukkan bahwa dataset memiliki kelengkapan data yang baik dan siap diproses ke tahap berikutnya. Selain itu, ketiadaan data duplikat memastikan bahwa tidak ada redundansi yang dapat memengaruhi hasil analisis.

5.3 Encoding

Proses *encoding* dilakukan untuk mengubah data non-numerik menjadi format numerik agar dapat diproses oleh algoritma *machine learning*. Fitur bertipe *boolean*, yaitu *Rectal Pain*, *Sore Throat*, *Penile Oedema*, *Oral Lesions*, *Solitary Lesion*, *Swollen Tonsils*, *HIV Infection*, dan *Sexually Transmitted Infection*, dikonversi menjadi nilai biner 0 (tidak) dan 1 (ya). Selanjutnya, fitur kategorikal *Systemic Illness* ditransformasikan menggunakan One-Hot Encoding, sedangkan variabel target *MonkeyPox* diencode menggunakan Label Encoding dengan memetakan kelas *negative* menjadi 0 dan *positive* menjadi 1.

5.4 Seleksi Fitur

Seleksi fitur dilakukan untuk menentukan fitur mana yang paling relevan terhadap variabel target dan mengurangi fitur yang kurang informatif. Metode *chi-squared test* digunakan karena sesuai untuk fitur kategorikal hasil *encoding*. Fitur-fitur yang memiliki p-value < 0.05 dinyatakan signifikan terhadap kelas *MonkeyPox* dan dipertahankan untuk tahap pemodelan. Pendekatan ini memastikan bahwa fitur yang dipilih benar-benar memiliki kontribusi terhadap proses klasifikasi, sehingga meningkatkan efisiensi dan akurasi model.

5.5 Split Data

Dataset yang sudah dibagi kedalam 2 bagian yaitu untuk pelatihan (*training*) dan pengujian (*testing*) dengan jumlah yang sebelumnya data kategori positif sebanyak 6.400 dan negatif 1.600 untuk data *training* dilakukan teknik SMOTE pada data untuk menyeimbangkan kelas agar performa *machine learning* lebih baik.

5.6 Resampling

Dataset awal memiliki ketidakseimbangan kelas yang signifikan, di mana kelas positif jauh lebih besar dibandingkan kelas negatif. Untuk mengatasi masalah ini, digunakan teknik SMOTE (*Synthetic Minority Over-sampling Technique*) pada data pelatihan. SMOTE menambah jumlah sampel kelas minoritas melalui proses interpolasi antar sampel yang mirip, sehingga menghasilkan data sintetik yang representatif. Setelah proses resampling, jumlah data antar kelas menjadi seimbang sehingga model tidak bias terhadap kelas mayoritas dan dapat mempelajari pola dari kedua kelas secara optimal.

5.5 Pemodelan

Model KNN menunjukkan akurasi yang cukup tinggi secara keseluruhan yaitu 78%. Namun, jika kita melihat lebih detail pada *classification report* dan *confusion matrix*, ada perbedaan kinerja yang signifikan antara prediksi kelas 0 dan kelas 1. Model sangat baik dalam mengidentifikasi kasus positif (kelas 1), dengan *recall* yang tinggi (0.93) dan *precision* yang tinggi (0.82). Ini berarti sebagian besar kasus positif berhasil terdeteksi, dan ketika model memprediksi positif, sebagian besar memang benar positif. Sebaliknya, kinerja model untuk kelas 0 (negatif) jauh lebih rendah, terutama pada *recall* (0.19) dan *precision* (0.41). Ini menunjukkan bahwa model kesulitan dalam mengidentifikasi kasus negatif yang sebenarnya (banyak False Positives, yaitu memprediksi positif padahal negatif) dan juga tidak terlalu akurat ketika memprediksi negatif.

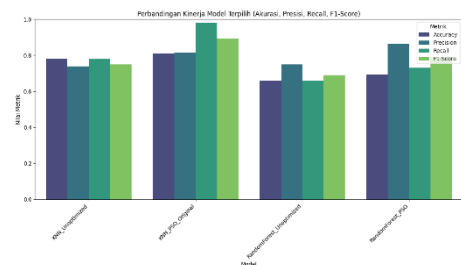
Model Random Forest awal menunjukkan akurasi keseluruhan sebesar 65.75%. Mirip dengan KNN, ada perbedaan kinerja antara kelas 0 dan kelas 1. Untuk kelas 0 (negatif), model memiliki *recall* yang lebih baik (0.57). Namun dari *matrix precision* untuk kelas 0 masih relatif rendah (0.31), menunjukkan bahwa banyak prediksi negatif ternyata salah (ada False Negatives yang signifikan). Untuk kelas 1 (positif), model menunjukkan *precision* yang tinggi (0.86), artinya ketika model memprediksi positif, sebagian besar memang benar. Namun, *recall* untuk kelas 1 (0.68) masih rendah yang berarti ada lebih banyak kasus positif yang tidak terdeteksi (False Negatives).

Setelah optimasi *hyperparameter* menggunakan PSO, model KNN menunjukkan peningkatan kinerja yang signifikan. Akurasi model meningkat menjadi 80%. Salah satu peningkatan paling mencolok adalah pada metrik *Recall* untuk kelas positif (kelas 1), yang mencapai 0.98. Ini berarti model KNN-PSO sangat efektif dalam mendeteksi hampir semua kasus positif yang sebenarnya (hanya 29 kasus positif yang terlewatkan dan salah diprediksi sebagai negatif, yaitu False Negatives). Namun, melihat *Confusion Matrix*, meskipun model sangat baik dalam mendeteksi kelas positif, model ini masih menghasilkan False Positives (FP) yang cukup tinggi (354 kasus). Jumlah *True Negatives* (TN) yang relatif rendah (46) juga menegaskan bahwa model masih lemah dalam mengidentifikasi kasus negatif dengan benar. Untuk model Random Forest menunjukkan peningkatan kinerja yang moderat dibandingkan dengan model Random Forest tanpa optimasi. Akurasi keseluruhan meningkat menjadi 69%. Untuk kelas positif (kelas 1), model menunjukkan *Precision* yang baik (0.86), yang berarti ketika model memprediksi suatu kasus sebagai positif, sebagian besar prediksi tersebut akurat. *Recall* untuk kelas positif juga meningkat menjadi 0.73, mengindikasikan bahwa model mampu mendeteksi sekitar 73% dari semua kasus positif yang sebenarnya. F1-Score sebesar 0.79

menunjukkan keseimbangan yang lebih baik antara *precision* dan *recall* untuk kelas mayoritas. Namun, melihat *Confusion Matrix*, meskipun terdapat peningkatan dalam deteksi kasus positif (TP), jumlah *false negatives* (FN) masih cukup tinggi (430). Ini berarti ada sejumlah besar kasus positif yang tidak terdeteksi oleh model dan salah diklasifikasikan sebagai negatif. Di sisi lain, False Positives (FP) relatif lebih rendah (183) dibandingkan dengan False Negatives.

5.6 Perbandingan Model

Dari empat model pengujian yang dicoba, terdapat 1 model yang memiliki nilai yang paling tinggi diantara 3 model lainnya, yaitu optimasi KNN menggunakan PSO. PSO berhasil menaikkan kinerja dari algoritma KNN dan Random forest, namun hasil akurasi tertinggi diperoleh oleh KNN. Untuk analisis hasil perbandingannya bisa dilihat pada Gambar 4. diagram sebagai berikut.



Gambar 4. Perbandingan Model

Berdasarkan hasil diagram di atas, secara perbandingan keseluruhan, KNN menunjukkan kinerja yang paling baik dari sisi akurasi, namun memiliki nilai yang berbeda-beda setiap matrix. Dari sisi akurasi, model KNN dengan optimasi PSO menunjukkan performa terbaik dengan nilai 80%, diikuti oleh KNN 78%, *random forest* + PSO 69%, dan *random forest* 65%. Hasil ini menunjukkan bahwa algoritma KNN secara umum lebih sesuai dengan karakteristik dataset dibandingkan *random forest*, dan bahwa optimasi PSO mampu meningkatkan kinerja kedua model, meskipun dampaknya lebih signifikan pada KNN. Model KNN dengan optimasi PSO merupakan model terbaik, dengan akurasi tertinggi (80%), *recall* kelas positif paling tinggi (93%), dan F1-score tertinggi (82%), yang menandakan kemampuan sangat baik dalam mendeteksi kasus positif dan meminimalkan *false negative*. KNN default juga menunjukkan performa yang kuat dengan *recall* tinggi (0.96), namun masih kalah dibandingkan KNN + PSO.

VI. KESIMPULAN DAN SARAN

6.1 Kesimpulan

Setelah melakukan implementasi, pengujian, dan analisis terhadap algoritma *k-nearest neighbors* (KNN) dan *random forest* baik sebelum maupun sesudah dioptimasi dengan

Particle Swarm Optimization (PSO), dapat ditarik beberapa kesimpulan sebagai berikut:

1. Model *k-nearest neighbors* (KNN) tanpa optimasi memperoleh akurasi sebesar 78%, presisi 82%, recall 93%, dan F1-score 87%. Nilai recall yang tinggi menunjukkan bahwa model sangat baik dalam mendeteksi kasus positif MonkeyPox, sehingga jumlah false negative relatif rendah. Namun, kemampuan model dalam mengenali kelas negatif masih terbatas.
2. Model *random forest* tanpa optimasi menghasilkan akurasi sebesar 65%, presisi 86%, recall 68%, dan F1-score 76%. Meskipun memiliki presisi yang tinggi, nilai recall yang lebih rendah dibandingkan KNN menunjukkan bahwa model ini masih kurang optimal dalam mendeteksi seluruh kasus positif.
3. KNN-PSO merupakan model terbaik berdasarkan keseluruhan metrik evaluasi, dengan akurasi 80%, recall 98%, dan F1-score 89%. Tingginya nilai *recall* menunjukkan bahwa model mampu meminimalkan kesalahan dalam mengidentifikasi kasus positif Monkeypox.
4. Kontribusi ilmiah penelitian ini terletak pada penyediaan bukti empiris mengenai pengaruh optimasi PSO terhadap kinerja KNN dan RF pada klasifikasi Monkeypox serta menunjukkan bahwa pemilihan algoritma dan optimasi parameter memberikan pengaruh terhadap kemampuan model dalam mendeteksi kasus positif. Hasil penelitian menunjukkan bahwa kombinasi KNN-PSO lebih efektif dibandingkan RF-PSO pada dataset yang digunakan, sehingga dapat menjadi referensi bagi penelitian selanjutnya dalam pengembangan model klasifikasi berbasis *machine learning* untuk identifikasi awal Monkeypox.

6.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat diberikan untuk pengembangan penelitian selanjutnya maupun penerapan hasil penelitian dalam bidang klasifikasi yaitu sebagai berikut:

1. Penelitian selanjutnya disarankan untuk menggunakan dataset yang lebih beragam atau berasal dari sumber klinis yang berbeda agar model yang dihasilkan memiliki kemampuan generalisasi yang lebih baik.
2. Dapat dilakukan eksplorasi algoritma lain, seperti *support vector machine* (SVM), *gradient boosting*, atau model berbasis *deep learning* untuk dibandingkan dengan KNN dan *random forest*.
3. Optimasi parameter dapat dikembangkan lebih lanjut dengan mengombinasikan PSO dengan metode optimasi lain.

4. Model yang dihasilkan dapat dikembangkan lebih lanjut menjadi sistem pendukung keputusan (*decision support system*) berbasis aplikasi atau web untuk membantu tenaga medis dalam melakukan deteksi awal penyakit MonkeyPox.

DAFTAR PUSTAKA

- [1]. J. Homepage, S. R. Cholil, T. Handayani, R. Prathivi, and T. Ardianita, 2021, *IJCIT (Indonesian Journal on Computer and Information Technology) Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa*.
- [2]. P. Cunningham and S. J. Delany, 2021, *K-Nearest Neighbour Classifiers-A Tutorial*, Jul. 31, *Association for Computing Machinery*. doi: 10.1145/3459665.
- [3]. A. Hamdan and D. Ekmekci, 2024, *Design of Monkeypox Disease Diagnosis Model Using Classical Machine Learning Algorithm*, *Journal of Soft Computing and Artificial Intelligence*, Jun. doi: 10.55195/jscai.1461849.
- [4]. E. Hadjaidji, M. C. A. Korba, and K. Khelil, "Improving detection of Parkinson's disease with acoustic feature optimization using particle swarm optimization and machine learning," *Mach. Learn. Sci. Technol.*, vol. 6, no. 1, Mar. 2025, doi: 10.1088/2632-2153/adadc3.
- [5]. A. Ariani, K. Kunci-Penyakit, and G. Kronis, *Klasifikasi Penyakit Ginjal Kronis menggunakan K-Nearest Neighbor*, vol. 5, no. 1. 2019. [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Chronic_Kidney_Dise
- [6]. Haidar, M. Field, J. Sykes, M. Carolan, and L. Holloway, 2021, *PSPSO: A package for parameters selection using particle swarm optimization*, *SoftwareX*, vol. 15, Jul. 2021, doi: 10.1016/j.softx.2021.100706.
- [7]. A. Jain, D. Somwanshi, K. Joshi, and S. S. Bhatt, "A Review: Data Mining Classification Techniques," in *Proceedings of 3rd International Conference on Intelligent Engineering and Management, ICIEM 2022*, Institute of Electrical and Electronics Engineers Inc., 2022, pp. 636–642. doi: 10.1109/ICIEM54221.2022.9853036.
- [8]. J. Homepage, S. R. Cholil, T. Handayani, Prathivi, and T. Ardianita, 2021, *IJCIT (Indonesian Journal on Computer and Information Technology) Implementasi Algoritma Klasifikasi K-Nearest Neighbor (KNN) Untuk Klasifikasi Seleksi Penerima Beasiswa*.
- [9]. A. Hamdan and D. Ekmekci, "Design of Monkeypox Disease Diagnosis Model Using Classical Machine Learning Algorithm,"

Journal of Soft Computing and Artificial Intelligence, Jun. 2024, doi: 10.55195/jscai.1461849.

- [10]. D. Freitas, L. G. Lopes, and F. Morgado-Dias, 2020, *Particle Swarm Optimisation: A historical review up to the current developments*, Mar. 01, 2020, MDPI AG. doi: 10.3390/E22030362.
- [11]. X. Wang and W. Lun, 2023, *Skin Manifestation of Human Monkeypox*, Feb. 01, MDPI. doi: 10.3390/jcm12030914.