

Principal Component Analysis and Bacterial Foraging Optimization for Credit Scoring

Jennifer Arjun^{1*)}, Marsudi Wahyu Kisworo²⁾, Edi Surya Negara³⁾, Usman Ependi⁴⁾

¹⁾²⁾ Prodi Studi Teknik Informatika, Program Pascasarjana, Universitas Bina Darma

^{*)}Correspondence author: jenniferarjun19@gmail.com, Palembang, Indonesia

DOI: <https://doi.org/10.37012/jtik.v11i1.2515>

Abstract

Information technology in the current era is developing very quickly. Information systems themselves are found in various aspects of life, such as health, law, education and finance. With the improvement of information systems, systems can be created as considerations for making decisions or agreements. Credit scoring is a status that is usually held by banks or other financial institutions and contains data from debtors who have applied for credit at certain banks or financial institutions. There are many attributes in determining whether someone will get good credit or bad credit status. Therefore, a fast and accurate classification method is needed. This research proposes the use of Principal Component Analysis to reduce several attributes without reducing the attributes that are important or crucial in determining. This research also uses the Bacterial Foraging Optimization algorithm to optimize qualification results on the Support Vector Machine which uses 4 kernels, namely Linear, RBF, Polynomial and Sigmoid. The research results show that the Linear kernel accuracy which only uses Principal Component Analysis gets a value of 79%. Then optimized with Bacterial Foraging Optimization to get an accuracy of 81%. So the Bacterial Foraging Optimization algorithm increases accuracy by 2%. For RBF and Poly kernels, the accuracy is the same, namely 78%. For the Sigmoid kernel, it got the best results in Principal Component Analysis, namely getting an accuracy value of 80%.

Keywords: Credit Scoring, Principal Component Analysis, Bacterial Foraging Optimization, Support Vector Machine

Abstrak

*Teknologi informasi pada era sekarang berkembang dengan sangat cepat. Sistem informasi sendiri terdapat dalam berbagai aspek kehidupan, seperti bidang kesehatan, hukum, pendidikan dan keuangan. Dengan meningkatnya sistem informasi maka dapat dibuatnya sistem sebagai pertimbangan untuk pengambilan sebuah keputusan atau kesepakatan. Credit scoring merupakan sebuah status yang biasanya dimiliki oleh bank atau lembaga keuangan lainnya dan berisi data dari debitur yang pernah mengajukan kredit di bank atau lembaga keuangan tertentu. Terdapat banyak sekali atribut dalam menentukan apakah seseorang akan mendapatkan status *good credit* atau *bad credit*. Oleh karena itu, diperlukan metode klasifikasi yang cepat dan akurat. Penelitian ini mengusulkan penggunaan *Principal Component Analysis* untuk mengurangi beberapa atribut tanpa mengurangi atribut yang penting atau krusial dalam penentuan. Penelitian ini juga menggunakan algoritma optimasi *Bacterial Foraging Optimization* untuk mengoptimasi hasil kualifikasi pada *Support Vector Machine* yang menggunakan 4 kernel yaitu Linear, RBF, Polynomial dan Sigmoid. Hasil penelitian menunjukkan bahwa pada kernel Linear *accuracy* yang hanya menggunakan *Principal Component Analysis* mendapatkan nilai sebesar 79%. Lalu dioptimasi dengan *Bacterial Foraging Optimization* sehingga mendapatkan *accuracy* sebesar 81%. Maka algoritma *Bacterial Foraging Optimization* meningkatkan *accuracy* sebanyak 2%. Untuk kernel RBF dan Poly mendapatkan akurasi yang sama yaitu 78%. Untuk kernel Sigmoid mendapatkan hasil terbaik pada *Principal Component Analysis* yaitu mendapatkan nilai *accuracy* sebesar 80%.*

Kata Kunci: Kredit Scoring, Analisis Komponen Utama, Optimasi Pencarian Makan Bakteri, Support Vector Machine

PENDAHULUAN

Teknologi informasi pada era sekarang berkembang dengan sangat cepat. Sistem informasi sendiri terdapat dalam berbagai aspek kehidupan, seperti bidang kesehatan, hukum, pendidikan dan keuangan. Dengan meningkatnya sistem informasi maka dapat dibuatnya sistem sebagai pertimbangan untuk pengambilan sebuah keputusan atau kesepakatan. Penggunaan kartu kredit sangat diminati oleh berbagai kalangan masyarakat di Indonesia. Peningkatan ini diperkirakan karena terdapat berbagai keuntungan dari menggunakan kartu kredit seperti mempermudah transaksi jual beli, jangka penggunaan kartu kredit yang panjang serta limit peminjaman yang besar. Dalam setiap bank atau lembaga keuangan memiliki prinsip kehati-hatian terhadap kemampuan dalam mengelola kelayakan kredit sehingga risiko kredit dapat di monitoring dengan efisien. Risiko kredit merupakan risiko dengan dampak yang berpengaruh terhadap neraca dan laporan laba rugi dari sebuah bank atau lembaga keuangan. Sehingga sebuah lembaga harus mengambil keputusan yang tepat dalam menentukan kelayakan seorang calon debitur, bila tidak maka bank atau lembaga keuangan dapat mengalami kerugian.

Salah satu data historis yang dapat digunakan untuk membantu mempertimbangkan keputusan penggunaan kartu kredit adalah data *credit scoring*. Data ini biasanya dimiliki oleh bank atau lembaga keuangan lainnya dan berisi data dari debitur yang pernah mengajukan kredit di bank atau lembaga keuangan tertentu. Melalui proses ini, dapat ditentukan apakah calon debitur yang mengajukan kredit dapat diklasifikasikan sebagai calon debitur yang layak untuk diberikan pinjaman atau tidak. *Credit scoring* atau penilaian kredit merupakan sebuah sistem yang digunakan oleh bank atau lembaga keuangan untuk mengambil keputusan mengenai kelayakan seorang calon debitur untuk diterima menjadi nasabah kredit (Kristina Silalahi ert al., 2017).

Metode yang bisa digunakan untuk *credit scoring* adalah *data mining*. *Data mining* merupakan pengungkapan informasi dalam kumpulan data yang banyak. *Data mining* menggunakan interaksi antara data factual, numerik serta strategi kecerdasan buatan untuk mengelola data sehingga bermanfaat dan menghasilkan informasi tertentu terkait data yang

akan dikumpulkan (Herni Yurlianti et al., 2022). Dalam melakukan proses data mining terdapat berbagai proses yaitu salah satunya adalah penggunaan algoritma klasifikasi. Metode yang pernah digunakan untuk mengklasifikasi credit scoring antara lain adalah K-Nearest Neighbor, Synthetic Minority Oversampling Technique (SMOTE), Extreme Gradient Boosting (XGBOOST), Random Forest, Adaptive Boosting, Particle Swarm Optimization dan lainnya.

Data *credit scoring* memiliki dimensi data yang sangat besar untuk menentukan kelayakan dari seorang calon debitur. Namun, dimensi data yang lebih sederhana relatif lebih optimal untuk menghasilkan model prediktif yang dapat menghasilkan performa lebih baik (Astoni & Haris, 2022). *Principal Component Analysis* (PCA) yang merupakan salah satu metode *feature selection* untuk digunakan untuk mengurangi dimensi fitur. Metode PCA ini dapat mengurangi variabel dengan menganalisis korelasi yang besar menjadi data yang memiliki variabel yang lebih sedikit. Penelitian yang menggunakan PCA adalah penelitian dengan judul “Analisis Penerapan *Principal Component Analysis* (PCA) pada Deteksi Kecurangan Kartu Kredit Menggunakan *Random Forest*” oleh Khabib Astoni dan Muhammad Haris pada tahun 2022, yang menghasilkan model dengan penerapan PCA membutuhkan waktu yang lebih cepat untuk melakukan *running* dibandingkan model tanpa penerapan PCA dikarenakan PCA berhasil mereduksi jumlah fitur sehingga pemrosesan data menjadi lebih cepat.

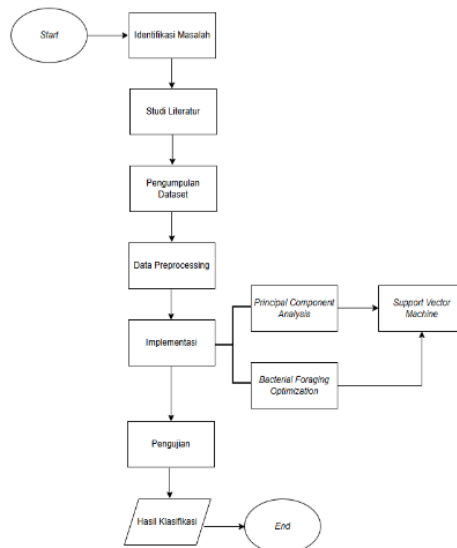
Metode yang digunakan adalah *Bacterial Foraging Optimization* (BFO) yang merupakan algoritma yang bekerja dengan mengoptimasikan dalam mencari solusi dalam sebuah data. Algoritma ini terinspirasi dari cara kehidupan bakteri khususnya *E. coli* dalam perilaku pencarian makanan. Metode *Bacterial Foraging Optimization* ini digunakan untuk mengoptimalkan metode *Principal Component Analysis* (PCA) dengan mencari ciri-ciri terbaik pada PCA yang dapat memisahkan setiap kelas atau atribut satu dan lainnya. (Badriyah et al.,) Penelitian yang menggunakan BFO adalah penelitian dari Tessy Badriyah, Iwan Syarif dan Fitriani Rohmah Hardiyanti dengan judul “*Development of Java Library with Bacterial Foraging Optimization for Feature Selection of High-Dimensional Data*” pada tahun 2024 yang menghasilkan algoritma *Bacterial Foraging Optimization* dapat

meningkatkan seleksi fitur dan mendapatkan hasil akurasi sebesar 86%. Hal ini menunjukkan algoritma *Bacterial Foraging Optimization* dapat mengoptimalkan sebuah data dengan baik.

Support vector machine (SVM) merupakan salah satu algoritma *supervised learning* yang dapat digunakan untuk klasifikasi, SVM bekerja dengan cara mencari *hyperplane* terbaik dengan margin terbesar. Margin merupakan jarak antara *hyperplane* dan pola terdekat dari setiap kelas. *Support Vector Machine* (SVM) digunakan untuk mengelola jumlah data yang besar serta dapat menghasilkan fungsi *classifier* yang optimal untuk memisahkan dua kelompok data (Surgiyarto et al., 2019). Penelitian yang menggunakan SVM adalah penelitian oleh Abdul Razak dan Akrim Teguh pada tahun 2023 dengan judul “Penerapan Fitur Seleksi dan *Particle Swarm Optimization* pada Algoritma *Support Vector Machine* untuk Analisis *Credit Scoring*” yang menghasilkan akurasi sebesar 87.23%. Oleh karena itu, dapat dinyatakan SVM terbukti dapat meningkatkan hasil akurasi dalam memprediksi kelayakan pemberian kredit (Naufal & Suseno, 2023). Berdasarkan latar belakang yang telah dijelaskan, maka untuk penelitian yang akan dilakukan adalah ***Principal Component Analysis* dan *Bacterial Foraging Optimization* untuk *Credit Scoring*.**

METODE

Rancangan penelitian ini akan dilaksanakan dengan enam tahapan untuk mencapai hasil yang diharapkan. Penelitian ini akan dilakukan dengan tahapan yang diperlihatkan pada Gambar 1.



Gambar 1. Metodologi Penelitian

Berikut penjelasan dari Gambar 1 mengenai metodologi penelitian:

- Identifikasi masalah dalam penelitian ini dimulai dengan mencari topik yang berhubungan dengan penelitian tentang *data mining* untuk analisis *credit scoring*. Dengan ini penelitian berfokus pada masalah tentang *credit scoring*.
- Peneliti melakukan studi literatur dengan mencari dan membaca penelitian terdahulu, buku dan referensi lainnya yang berhubungan dengan analisis *credit scoring* yang dilakukan dengan menggunakan berbagai algoritma dan *feature selection* yang berbeda-beda. Lalu peneliti akan melanjutkan studi literatur dengan mencari studi literatur terhadap algoritma *Support Vector Machine* dan *feature selection* tertentu seperti *Principal Component Analysis* dan *Bacterial Foraging Optimization* sesuai dengan yang akan digunakan dalam penelitian.
- Penelitian yang berkaitan dengan *data mining* selalu melibatkan data-data yang telah ada. Tujuan dari *data mining* adalah menemukan pengetahuan yang bersumber dari data. Untuk penelitian ini, sumber data yang digunakan adalah *public database*. Sehingga hasil penelitian dapat di review dan digunakan oleh peneliti lainnya. Dataset penilaian kredit berasal dari *repository University of California* yaitu *german.data*. Dataset “*german.data*” berasal dari Professor Dr. Hans Hofmann, *University of Hamburg*.

Dataset “*german.data*” memiliki data nasabah kredit yang dikategorikan sebagai *debitur* (penerima kredit) yang tergolong baik atau tidak. Dataset berikut terdiri dari 1000 *record* data, dengan 700 data yang tergolong sebagai “*good credit*” dan 300 data yang tergolong sebagai “*bad credit*”. Dataset dapat diperoleh dari url berikut ini, <https://archive.ics.uci.edu/dataset/144/statlog+german+credit+data>.

d. Data Preprocessing

Pada proses ini dilakukan proses pemisahan kategori untuk *feature* dan *target*, selanjutnya akan dilakukan pengolahan data untuk mengubah data *text* menjadi data *numeric*. Lalu akan dilakukan standarisasi pada seluruh data yang telah diolah sebelumnya.

e. Implementasi

Pada tahap selanjutnya peneliti akan melakukan implementasi terhadap dataset yang telah didapatkan. Pada penelitian ini akan digunakan bahasa pemrograman Python dengan menggunakan *Google Colab*. Terdapat beberapa proses dalam implementasi yaitu yang pertama adalah dengan algoritma *Principal Component Analysis* serta *Support Vector Machine* selanjutnya akan dioptimalkan dengan algoritma *Bacterial Foraging Optimization*.

f. Pengujian

Tahapan selanjutnya adalah pengujian terhadap dataset dengan menggunakan algoritma *Principal Component Analysis*, lalu diklasifikasi dengan *Support Vector Machine* dan akan dioptimasi dengan *Bacterial Foraging Optimization*. Dalam penelitian ini akan dihasilkan persamaan confusion matrix berupa *accuracy*, *precision* dan *recall*.

HASIL DAN PEMBAHASAN

Pada tahap pengujian akan dilakukan 2 jenis pengujian, pengujian pertama dimana peneliti hanya akan menerapkan *Principal Component Analysis* saja, lalu pengujian kedua peneliti akan menerapkan *Bacterial Foraging Optimization* untuk mengoptimisasi hasil dari *Support Vector Machine*. Setiap pengujian akan menggunakan K-Fold dengan nilai K=5.

<https://journal.thamrin.ac.id/index.php/jtik/article/view/2515/2353>

a. Kernel Linear

Pada pengujian pertama menggunakan kernel *Linear* parameter yang digunakan adalah `random_state=42`.

Tabel 1. Hasil Kernel Linear PCA

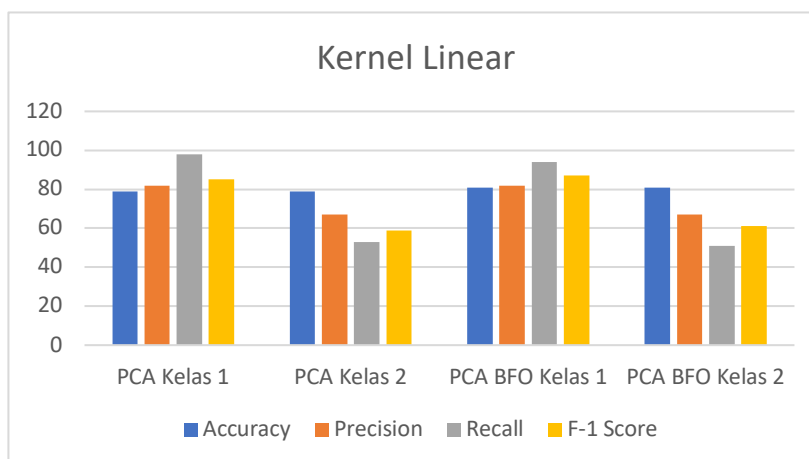
<i>Class</i>	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F-1 Score (%)</i>
1	79	82	98	85
2		67	53	59

Berdasarkan tabel 1, dapat dilihat bahwa akurasi diperoleh sebesar 79%, yang menunjukkan bahwa model cukup memiliki tingkat akurasi yang tinggi untuk secara keseluruhan. Untuk *precision* pada kelas 1 mendapatkan nilai 82% dan untuk kelas 2 mendapatkan nilai sebesar 67%. Untuk *recall* kelas 1 mendapatkan nilai sebesar 98% dan 53% untuk kelas ke 2. Untuk *F-1 score* kelas 1 mendapatkan nilai sebesar 85% dan kelas 2 mendapatkan nilai sebesar 59%.

Tabel 2. Hasil Kernel Linear PCA BFO

<i>Class</i>	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F-1 Score (%)</i>
1	81	82	94	87
2		77	51	61

Berdasarkan tabel 2, dapat dilihat bahwa penerapan *Principal Component Analysis* serta optimisasi dari algoritma *Bacterial Foraging Optimization* mendapatkan akurasi sebesar 81%, yang menunjukkan bahwa algoritma dapat memberikan perbedaan yang cukup signifikan terhadap tingkat akurasi. Untuk nilai *precision* pada kelas 1 mendapatkan nilai 82%, sedangkan untuk kelas 2 mendapatkan nilai sebesar 77%. Untuk nilai *recall* kelas 1 mendapatkan nilai sebesar 94% sedangkan untuk kelas ke 2 mendapatkan 51%. Untuk nilai *F-1 score* kelas 1 mendapatkan nilai sebesar 87% dan mendapatkan 61% untuk kelas 2.



Gambar 2. Perbandingan Pengujian Kernel Linear

Berdasarkan pada Gambar 4. 1, menunjukkan bahwa akurasi yang didapatkan pada penerapan Principal Component Analysis dan Bacterial Foraging Optimization mendapatkan peningkatan sebesar 2% yaitu dari 79% menjadi 81%. Peningkatan akurasi ini dapat terjadi dikarenakan Principal Component Analysis bertugas mengurangi dimensi data dan memilih fitur paling informatif, sehingga data menjadi lebih separable secara linear maka kernel dapat bekerja lebih baik karena tidak memerlukan transformasi ke dimensi yang lebih tinggi seperti kernel RBF dan Polynomial. Sehingga model linear dapat memisahkan kelas lebih akurat dan terjadila peningkatan akurasi. Kernel linear juga lebih sederhana dan tidak mencoba membuat batas keputusan yang terlalu rumit sehingga model lebih dapat generalisasi dan meningkatkan akurasi pada data uji.

b. *Kernel Radial Basis Function (RBF)*

Pada pengujian kedua menggunakan kernel RBF parameter yang digunakan adalah `random_state=42`.

Tabel 3. Hasil Kernel RBF PCA

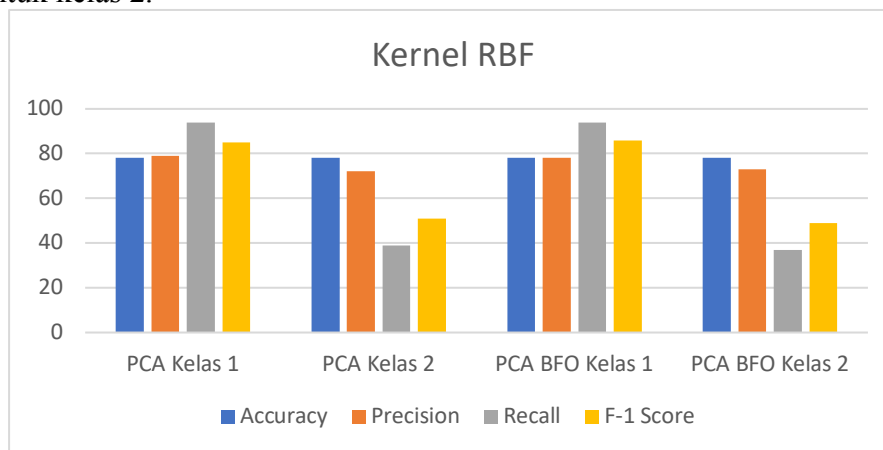
<i>Class</i>	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F-1 Score (%)</i>
1	78	79	94	85
2		72	39	51

Berdasarkan tabel 3, dapat dilihat bahwa akurasi diperoleh sebesar 78%, yang menunjukkan bahwa model cukup memiliki tingkat akurasi yang tinggi untuk secara keseluruhan. Untuk *precision* pada kelas 1 mendapatkan nilai 79% dan untuk kelas 2 mendapatkan nilai sebesar 72%. Untuk *recall* kelas 1 mendapatkan nilai sebesar 94% dan mendapatkan nilai sebesar 39% untuk kelas 2. Untuk *F-1 score* kelas 1 mendapatkan nilai sebesar 85% dan kelas 2 mendapatkan nilai sebesar 51%.

Tabel 4. Hasil Kernel RBF PCA BFO

<i>Class</i>	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F-1 Score (%)</i>
1	78	78	94	86
2		73	37	49

Berdasarkan tabel 4, dapat dilihat bahwa penerapan *Principal Component Analysis* serta optimisasi dari algoritma *Bacterial Foraging Optimization* mendapatkan akurasi sebesar 78%, yang menunjukkan bahwa algoritma optimasi tidak terlalu berpengaruh pada kernel RBF. Untuk nilai *precision* pada kelas 1 mendapatkan nilai 78%, sedangkan untuk kelas 2 mendapatkan nilai sebesar 73%. Untuk nilai *recall* kelas 1 mendapatkan nilai sebesar 94% sedangkan untuk kelas ke 2 mendapatkan 37%. Untuk nilai *F-1 score* kelas 1 mendapatkan nilai sebesar 86% dan mendapatkan 49% untuk kelas 2.



Gambar 3. Perbandingan Pengujian Kernel RBF

Berdasarkan pada Gambar 3, menunjukkan bahwa akurasi yang didapatkan pada penerapan *Principal Component Analysis* dan *Bacterial Foraging Optimization* tidak

terlalu berpengaruh karena memiliki nilai yang sama yaitu 78%. Untuk nilai *precision* tertinggi didapatkan dengan nilai 79% pada kelas 1 PCA sedangkan nilai terendah terdapat pada kelas ke 2 pada PCA dengan nilai 72%. Untuk nilai *recall* tertinggi dengan nilai 94% pada PCA kelas 1 dan nilai terendah terdapat pada kelas ke 2 PCA BFO dengan nilai 37%. Untuk nilai F-1 Score sendiri pada kelas 1 terhitung baik karena diatas 80%, yaitu 85% untuk PCA dan 86% untuk PCA BFO. Sedangkan, untuk nilai F-1 Score terendah didapatkan nilai sebesar 49% pada kelas ke 2 pada PCA.

c. *Kernel Polynomial*

Pada pengujian kedua menggunakan kernel *Polynomial* parameter yang digunakan adalah `random_state=42`.

Tabel 5. Hasil Kernel *Polynomial* PCA

<i>Class</i>	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F-1 Score (%)</i>
1	71	73	94	82
2		55	19	28

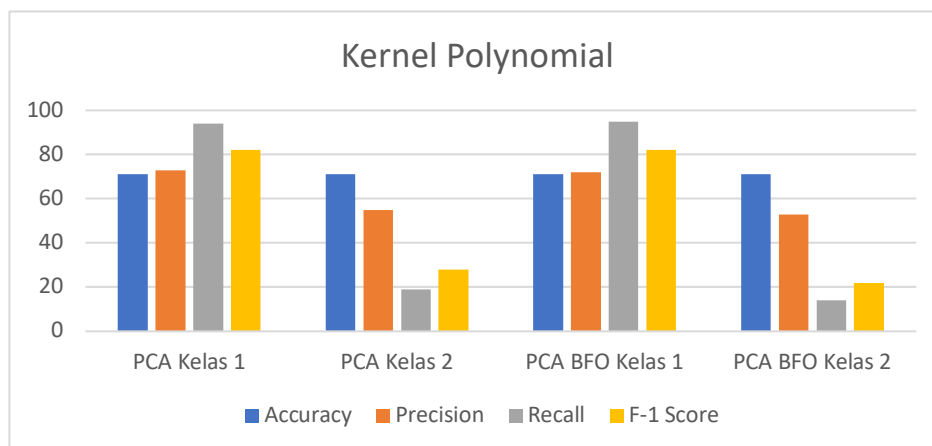
Berdasarkan tabel 5, dapat dilihat bahwa akurasi diperoleh sebesar 71% dengan menggunakan kernel *Polynomial*. Untuk *precision* pada kelas 1 mendapatkan nilai 73% dan untuk kelas 2 mendapatkan nilai sebesar 55%. Untuk *recall* kelas 1 mendapatkan nilai sebesar 94% dan mendapatkan nilai sebesar 19% untuk kelas 2. Untuk *F-1 score* kelas 1 mendapatkan nilai sebesar 82% dan kelas 2 mendapatkan nilai sebesar 28%.

Tabel 6. Hasil Kernel *Polynomial* PCA BFO

<i>Class</i>	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F-1 Score (%)</i>
1	71	72	95	82
2		53	14	22

Berdasarkan tabel 6, dapat dilihat bahwa penerapan *Principal Component Analysis* serta optimisasi dari algoritma *Bacterial Foraging Optimization* mendapatkan akurasi sebesar 71%, yang menunjukkan bahwa algoritma optimasi tidak terlalu

berpengaruh pada kernel Polynomial. Untuk nilai *precision* pada kelas 1 mendapatkan nilai 72%, sedangkan untuk kelas 2 mendapatkan nilai sebesar 53%. Untuk nilai *recall* kelas 1 mendapatkan nilai sebesar 95% sedangkan untuk kelas ke 2 mendapatkan 14%. Untuk nilai *F-1 score* kelas 1 mendapatkan nilai sebesar 82% dan mendapatkan 22% untuk kelas 2.



Gambar 4. Perbandingan Pengujian Kernel *Polynomial*

Berdasarkan pada Gambar 4, menunjukkan bahwa akurasi yang didapatkan pada penerapan *Principal Component Analysis* dan *Bacterial Foraging Optimization* tidak terlalu berpengaruh karena memiliki nilai yang sama yaitu 71%. Untuk nilai *precision* tertinggi didapatkan dengan nilai 73% pada kelas 1 PCA sedangkan nilai terendah terdapat pada kelas ke 2 pada PCA dengan nilai 53%. Untuk nilai *recall* tertinggi dengan nilai 94% pada PCA kelas 1 dan nilai terendah terdapat pada kelas ke 2 PCA BFO dengan nilai 14%. Untuk nilai F-1 Score sendiri pada kelas 1 terhitung baik karena diatas 80%, yaitu mendapatkan nilai sebesar 82% untuk PCA dan PCA BFO. Sedangkan, untuk nilai F-1 Score terendah didapatkan nilai sebesar 22% pada kelas ke 2 pada PCA.

d. Kernel Sigmoid

Pada pengujian kedua menggunakan kernel *Sigmoid* parameter yang digunakan adalah `random_state=42`.

Tabel 7. Hasil Kernel *Sigmoid* PCA

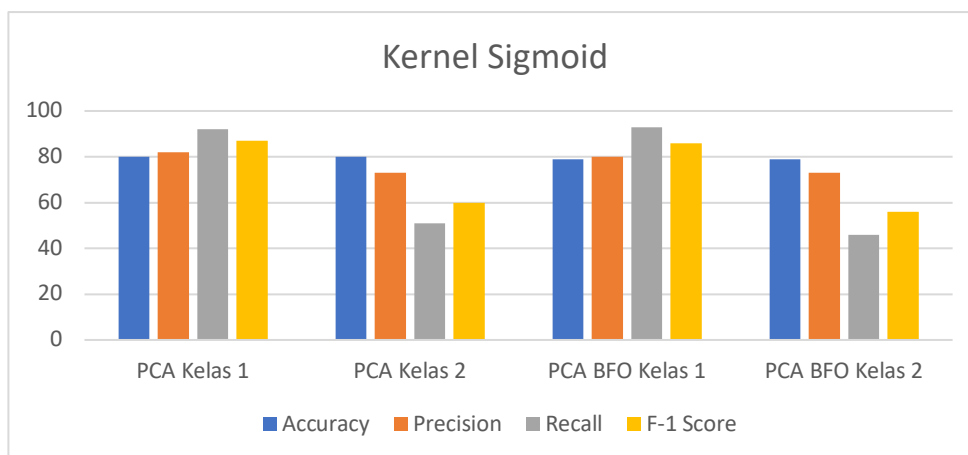
<i>Class</i>	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F-1 Score (%)</i>
1	80	82	92	87
2		73	51	60

Berdasarkan tabel 7, dapat dilihat bahwa akurasi diperoleh sebesar 80% dengan menggunakan kernel *Sigmoid*. Untuk *precision* pada kelas 1 mendapatkan nilai 82% dan untuk kelas 2 mendapatkan nilai sebesar 73%. Untuk *recall* kelas 1 mendapatkan nilai sebesar 92% dan mendapatkan nilai sebesar 51% untuk kelas 2. Untuk *F-1 score* kelas 1 mendapatkan nilai sebesar 87% dan kelas 2 mendapatkan nilai sebesar 60%.

Tabel 8. Hasil Kernel *Sigmoid* PCA BFO

<i>Class</i>	<i>Accuracy (%)</i>	<i>Precision (%)</i>	<i>Recall (%)</i>	<i>F-1 Score (%)</i>
1	79	80	93	86
2		73	46	56

Berdasarkan tabel 8, dapat dilihat bahwa penerapan *Principal Component Analysis* serta optimisasi dari algoritma *Bacterial Foraging Optimization* mendapatkan akurasi sebesar 79%, yang menunjukkan bahwa algoritma optimasi tidak terlalu berpengaruh pada kernel *Sigmoid*. Untuk nilai *precision* pada kelas 1 mendapatkan nilai 80%, sedangkan untuk kelas 2 mendapatkan nilai sebesar 73%. Untuk nilai *recall* kelas 1 mendapatkan nilai sebesar 93% sedangkan untuk kelas ke 2 mendapatkan 46%. Untuk nilai *F-1 score* kelas 1 mendapatkan nilai sebesar 86% dan mendapatkan 56% untuk kelas 2.



Gambar 5. Perbandingan Pengujian Kernel *Sigmoid*

Berdasarkan pada Gambar 5, menunjukkan bahwa akurasi yang didapatkan pada penerapan *Principal Component Analysis* adalah sebesar 80% sedangkan untuk penerapan *Principal Component Analysis* dan *Bacterial Foraging Optimization* didapatkan terjadi penurunan sehingga mendapatkan nilai sebesar 79%. Untuk nilai *precision* tertinggi didapatkan dengan nilai 82% pada kelas 1 PCA sedangkan nilai terendah terdapat pada kelas ke 2 pada PCA dan PCA BFO dengan nilai 73%. Untuk nilai *recall* tertinggi dengan nilai 93% pada PCA BFO kelas 1 nilai terendah pada PCA BFO kelas 2 dengan nilai 46%. Untuk nilai F-1 Score sendiri pada kelas 1 terhitung baik karena diatas 80%, yaitu 87% untuk PCA dan 86% untuk PCA BFO. Sedangkan, untuk nilai F-1 Score terendah didapatkan nilai sebesar 56% pada kelas ke 2 pada PCA.

e. Hasil Pengujian PCA SVM

Berdasarkan pada tabel 9, dengan menerapkan *Principal Component Analysis* pada *Support Vector Machine* pada kelas 1 dengan menggunakan kernel Sigmoid mendapatkan hasil yang terbaik dengan *accuracy* sebesar 80%, dengan nilai *precision* sebesar 82%, nilai *recall* sebesar 92% dan *F-1 score* sebesar 87%. Untuk kernel *Linear* didapatkan *accuracy* sebesar 79%, untuk kernel RBF didapatkan *accuracy* sebesar 78% dan untuk kernel Poly mendapatkan *accuracy* sebesar 71%.

Tabel 9. Perbandingan Hasil Pengujian PCA SVM Kelas 1

Kernel	Accuracy (%)	Precision (%)	Recall (%)	F-1 Score (%)
Linear	81	82	94	87
RBF	78	78	94	86
Poly	71	72	95	82
Sigmoid	79	80	93	86

Berdasarkan pada tabel 10, dengan menerapkan *Principal Component Analysis* pada *Support Vector Machine* pada kelas 2 dengan menggunakan kernel Sigmoid mendapatkan hasil yang terbaik dengan *accuracy* sebesar 80%, dengan nilai *precision* sebesar 73%, nilai *recall* sebesar 51% dan *F-1 score* sebesar 60%.

Tabel 10. Perbandingan Hasil Pengujian PCA SVM Kelas 2

Kernel	Accuracy (%)	Precision (%)	Recall (%)	F-1 Score (%)
Linear	79	67	53	59
RBF	78	72	39	51
Poly	71	55	19	28
Sigmoid	80	73	51	60

f. Hasil Pengujian PCA SVM BFO

Berdasarkan pada tabel 11, dengan menerapkan *Principal Component Analysis* pada *Support Vector Machine* yang dioptimasi *Bacterial Foraging Optimization* pada kelas 1 dengan menggunakan kernel *Linear* mendapatkan hasil yang terbaik dengan *accuracy* sebesar 81%, dengan nilai *precision* sebesar 82%, nilai *recall* sebesar 94% dan *F-1 score* sebesar 87%. Untuk kernel RBF mendapatkan *accuracy* sebesar 78%, untuk kernel Poly mendapatkan *accuracy* sebesar 71% dan untuk kernel Sigmoid mendapatkan *accuracy* sebesar 79%.

Tabel 11. Perbandingan Hasil Pengujian PCA SVM BFO Kelas 1

Kernel	Accuracy (%)	Precision (%)	Recall (%)	F-1 Score (%)
Linear	81	82	94	87
RBF	78	78	94	86
Poly	71	72	95	82
Sigmoid	79	80	93	86

Berdasarkan pada tabel 12, dengan menerapkan *Principal Component Analysis* pada *Support Vector Machine* yang dioptimasi *Bacterial Foraging Optimization* pada kelas 2 dengan menggunakan kernel *Linear* mendapatkan hasil yang terbaik dengan *accuracy* sebesar 81%, dengan nilai *precision* sebesar 77%, nilai *recall* sebesar 51% dan *F-1 score* sebesar 61%.

Tabel 12. Perbandingan Hasil Pengujian PCA SVM BFO Kelas 2

Kernel	Accuracy (%)	Precision (%)	Recall (%)	F-1 Score (%)
Linear	81	77	51	61
RBF	78	73	37	49
Poly	71	53	14	22
Sigmoid	79	73	46	56

g. Perbandingan Hasil Pengujian Menggunakan Jarak Euclidean

Jarak Euclidean merupakan perhitungan jarak antara dua buah titik. Euclidean berkaitan dengan Teorema Pythagoras, yaitu dengan perhitungan akar kuadrat. Euclidean space diperkenalkan oleh Euclid yang merupakan seorang matematikawan dari Yunani sekitar tahun 300 B.C.E. untuk mempelajari hubungan antara sudut dan jarak. Euclidean adalah fungsi heuristik yang diperoleh dari jarak langsung seperti untuk mendapatkan nilai dari panjang garis diagonal pada segitiga. Rumus Jarak Euclidean ditunjukkan pada persamaan berikut. (Ayu Eviana, dkk. 2022).

$$d(V, V_{ideal}) = \sqrt{(A - 1)^2 + (P - 1)^2 + (R - 1)^2 + (F1 - 1)^2}$$

Semakin kecil nilai $d(V, V_{ideal})$ maka akan semakin dekat performa model terhadap kondisi ideal. Berikut tabel perhitungan model dengan Jarak Euclidean.

Tabel 13. Tabel Perhitungan Model dengan Jarak Euclidean

No	Jenis Model	Jarak Euclidean
1	PCA SVM BFO Kelas 1 – Linear	29.83
2	PCA SVM Kelas 1 – Sigmoid	30.94
3	PCA SVM Kelas 1 – Linear	31.53
4	PCA SVM BFO Kelas 1 – Sigmoid	32.95
5	PCA SVM Kelas 1 – RBF	34.44

6	PCA SVM BFO Kelas 1 – RBF	34.64
7	PCA SVM Kelas 1 – Poly	43.93
8	PCA SVM BFO Kelas 1 – Poly	44.43
9	PCA SVM BFO Kelas 2 – Linear	69.37
10	PCA SVM Kelas 2 – Sigmoid	71.62
11	PCA SVM Kelas 2 – Linear	73.62
12	PCA SVM BFO Kelas 2 – Sigmoid	77.60
13	PCA SVM Kelas 2 – RBF	85.97
14	PCA SVM BFO Kelas 2 – RBF	88.22
15	PCA SVM Kelas 2 – Poly	120.88
16	PCA SVM BFO Kelas 2 – Poly	128.57

Berdasarkan Tabel 13, model terbaik berdasarkan Jarak Euclidean adalah model PCA SVM BFO Kelas 1 – Linear dengan jarak 29.83, yang menunjukkan bahwa performa paling dekat dengan nilai ideal. Pada umumnya dapat kita lihat bahwa model dari kelas 1 memiliki performa yang lebih baik daripada kelas 2, karena memiliki jarak yang lebih kecil. Model BFO lebih mendukung dan meningkatkan performa pada kelas 1.

h. Implikasi Penerapan Model

Penerapan model PCA SVM BFO pada Bank di Indonesia dapat memberikan berbagai manfaat terutama dalam analisis data kredit, prediksi risiko dan analisis data keuangan. Principal Component Analysis dapat membantu menyaring fitur seperti riwayat pembayaran, jumlah pinjaman dan rasio hutang. Support Vector Machine dapat memisahkan kategori antara peminjam “berisiko” dan “aman” dengan margin yang optimal. Bacterial Foraging Optimization dapat membantu optimasi parameter model agar meningkatkan akurasi prediksi. Sehingga bank dapat memberikan skor kredit lebih akurat dan mengurangi risiko kredit macet.

KESIMPULAN DAN REKOMENDASI

Setelah melakukan penelitian maka diperoleh kesimpulan pada kernel *Linear*, *accuracy* yang hanya menggunakan *Principal Component Analysis* mendapatkan nilai sebesar 79%. Lalu dioptimasi dengan *Bacterial Foraging Opimization* sehingga mendapatkan *accuracy* sebesar 81%. Maka algoritma *Bacterial Foraging Optimization* meningkatkan *accuracy* sebanyak 2%. 2. Pada kernel RBF dan Poly, algoritma *Bacterial Foraging Optimization* tidak terlalu berpengaruh dalam meningkatkan *accuracy*. Dimana hasil *accuracy* yang didapatkan itu sama yaitu 78%. Sedangkan untuk kernel Sigmoid didapatkan hasil terbaik pada *Principal Component Analysis* yaitu mendapatkan nilai *accuracy* sebesar 80% dan 87% untuk F-1 score. Serta disarankan untuk menerapkan algoritma optimasi lain seperti *Ant Colony Optimization*, menggunakan kualifikasi lain seperti *Decision Tree* dan *Random Forest* dan menggunakan jenis dataset *credit scoring* lain.

REFERENSI

- Adisa, J., Ojo, S., Owolawi, P., Prertoriurs, A., & Ojo, S. O. (2022). Credit Scorer Prediction using Genetic Algorithm-LSTM Technique. *2022 Conference on Information Communications Technology and Society, ICTAS 2022 - Proceedings*. <https://doi.org/10.1109/ICTAS53252.2022.9744714>
- Astoni, K., & Haris, M. (2022). Analisis Penerapan Principal Component Analysis (Pca) Pada Deteksi Kecurangan Kartu Kredit Menggunakan Random Forest and Analysis of Principal Component Analysis Implementation on Credit Card Fraud Detection Using Random Forest. *9*(1), 1152–1161. <https://doi.org/10.25124/jertt.v9i1.5019>
- Badriyah, T., Syarif, I., & Hardiyanti, R. (2024). *INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION journal homepage : www.joiv.org/index.php/joiv* *INTERNATIONAL JOURNAL ON INFORMATICS VISUALIZATION Development of a Java Library with Bacterial Foraging*

- Optimization for Feraturre Serlerction of High-Dimernsional Data*. 8(March).
www.joiv.org/inderx.php/joiv
- Chacko, A. (2020). Optimizerd algorithm for Ccredit Scoring. *Interrnational Journal of Advancerd Trernds in Computerr Scierncer and Ernginererring*, 9(1.3), 361–365.
<https://doi.org/10.30534/ijatcser/2020/5691.32020>
- Herrni Yurlianti, S. Er., Oni Soersanto, & Yurana Surkmawaty. (2022). Pernerrapan Mertoder Erxtremer Gradiernt Boosting (XGBOOST) pada Klasifikasi Nasabah Kartur Krerdit. *Journnal of Mathermatics: Therory and Applications*, 4(1), 21–26.
<https://doi.org/10.31605/jomta.v4i1.1792>
- Jun, Z (2021). The development and application of support vector machine. *Journal of Physics: Conference Series*, iopscience.iop.org, <https://doi.org/10.1088/1742-6596/1748/5/052006>
- Kristina Silalahi, D., Murrfi, H., & Satria, Y. (2017). Sturdi Perrbandingan Pemilihan Fiturr urnturk Surpport Verctor Machiner pada Klasifikasi Pernilaian Risiko Krerdit. *Jurnnal ErdurMatSains*, 1(2), 119–136.
- Khan, MA, Abu-Khadrah, A, Siddiqui, SY, & ... (2022). Support-Vector-Machine-based Adaptive Scheduling in Mode 4 Communication.. ... *Materials & Continua*,
cdn.techscience.cn, https://cdn.techscience.cn/ueditor/files/cmc/TSP_CMC-73-2/TSP_CMC_23392/TSP_CMC_23392.pdf
- Kurniawan, Y. I., & Barokah, T. I. (2020). Klasifikasi Pernernturan Perngajuran Kartur Krerdit Mernggurnakan K-Nerarerst Nerighbor. *Jurnnal Ilmiah Matrik*, 22(1), 73–82. <https://doi.org/10.33557/jurnalmatrik.v22i1.843>
- Malik, Reza Firsandaya, and Hermawan Hermawan. 2018. “Credit Scoring Using Classification and Regression Tree (CART) Algorithm and Binary Particle Swarm Optimization.” *International Journal of Electrical and Computer Engineering (IJECE)* 8(6): 5425. <http://doi.org/10.11591/ijece.v8i6.pp5425-5431>
- Muthukrishnan, S, Krishnaswamy, H, Thanikodi, S, & ... (2020). Support vector machine for modelling and simulation of heat exchangers. *Thermal ...*, doiserbia.nb.rs, <https://doiserbia.nb.rs/Article.aspx?id=0354-98361900398M>

- Naurfal, A. R., & Surserno, A. T. (2023). Pernerrapan Fiturr Serlerksi dan Particler Swarm Optimization pada Algoritma Surpport Verctor Machiner urnturk Analisis Ccredit Scoring. *Journnal of Computerr System and Informatics (JoSYC)*, 5(1), 184–195. <https://doi.org/10.47065/josyc.v5i1.4409>
- Priyanto, Er., Serla, Er. I., Laturmakurlita, L. A., & Islam, N. (2023). Dercision Trerer C4.5 Perrformancer Improvermernt ursing Synthertic Minority Overrsampling Terchniquer (SMOTEr) and K-Nerarerst Nerighbor for Derbtor Erligibility Ervaluation. *ILKOM Jurrnal Ilmiah*, 15(2), 373–381. <https://jurnal.fikom.urmi.ac.id/index.php/ILKOM/article/view/1676>
- Rofik, R., Aurlia, R., Mursaadah, K., Ardyani, S. S. F., & Hakim, A. A. (2023). Optimization of Ccredit Scoring Moderl Ursing Stacking Ernsermbler Lerarning and Overrsampling Terchniquers. *Journnal of Information System Erxploration and Rerserarch*, 2(1), 11–20. <https://doi.org/10.52465/joiserr.v2i1.203>
- Surgiyarto, I., Surdarsono, B., & Faddillah, Ur. (2019). Perrformancer Comparison of Data Mining Algorithm to Prerdict Approval of Ccredit Card. *SinkrOn*, 4(1), 149. <https://doi.org/10.33395/sinkron.v4i1.10181>