

# Algoritma Swarm Intelligence untuk Data Berdimensi Tinggi pada Machine Learning: Review

Joan Angelina Widians\*<sup>1</sup>, Ade Fiqri Tjiko<sup>2</sup>

<sup>1,2</sup> Informatika, Fakultas Teknik, Universitas Mulawarman, Samarinda  
e-mail: \*<sup>1</sup>angelwidians@unmul.ac.id

## Abstrak

Data berdimensi tinggi berpengaruh pada learning model, ruang pencarian dan waktu komputasi, serta akurasi informasi. Banyaknya karakteristik dan dimensi tinggi, serta klasifikasi pola memerlukan pemilihan fitur atau feature selection (FS). FS dapat menghapus fitur-fitur yang berlebihan dan tidak relevan saat memilih subset fitur-fitur terkait. Swarm Intelligence (SI) banyak digunakan untuk mengatasi permasalahan data berdimensi tinggi. Secara konsep, SI dapat berarti sebagai kecerdasan kolektif yang dihasilkan dari tingkah laku kawan-an agen yang terinspirasi dari alam. Makalah ini mengenai tinjauan literatur yang komprehensif tentang algoritma SI terkhusus Ant Colony Optimization, Particle Swarm Optimization, dan Grey Wolf Optimizer. Selain itu, pemaparan analisis framework SI terpadu dan investigasi berbagai pendekatan algoritma SI dalam FS. Analisis lebih lanjut menggambarkan terdapat gabungan beberapa algoritma SI di berbagai domain riset. Teknik hybrid SI ini dapat diterapkan dalam FS untuk menemukan subset fitur dengan ukuran lebih kecil dan meningkatkan performa klasifikasi dibandingkan penggunaan algoritma FS biasa. Dengan makalah ini kami menganalisis research-gap, memberikan gambaran komprehensif tentang penerapan SI dalam FS, serta mengusulkan diagram hibrid algoritma SI untuk permasalahan teks FS. Penelitian mendatang diharapkan dapat melakukan penggabungan maupun modifikasi berbagai algoritma SI menjadi suatu algoritma yang dapat meningkatkan performa sekaligus menurunkan kompleksitas waktu komputasi pada data mining dan teks mining, sehingga penggunaan algoritma SI semakin berkembang dan memberikan solusi yang efektif dan efisien di era big data.

**Kata kunci**—Ant Colony Optimization, Feature Selection, Grey Wolf Optimizer, Particle Swarm Optimization, Swarm Intelligence.

## 1. PENDAHULUAN

Saat ini, para peneliti di bidang Machine Learning (ML) dan domain serupa semakin menyadari pentingnya pengurangan dimensi data yang dianalisis. Data berdimensi tinggi tersebut tidak hanya mempengaruhi learning model, meningkatkan ruang pencarian dan waktu komputasi, tetapi juga dapat berpengaruh pada akurasi informasi. Selain itu, data berdimensi tinggi, fitur yang banyak, akan menuntut pembuatan model ML yang sesuai namun seringkali hampir tidak membuahkan hasil. Untuk mengatasi masalah tersebut, dapat digunakan dua pendekatan yaitu ekstraksi fitur atau (*feature extraction*), yang terdiri dari pembuatan ruang fitur baru dengan dimensi rendah. Pendekatan kedua adalah pemilihan fitur (*feature selection*) yang berfokus pada penghapusan fitur-fitur yang tidak relevan dan berlebihan dari kumpulan fitur asli. Swarm Intelligence (SI) merupakan teknik komputasi modern. Pada perkembangan awalnya SI digunakan untuk optimisasi, namun beberapa tahun terakhir SI banyak digunakan untuk *data mining*, *text mining*, dan *big data* [1]. Feature Selection (FS) semakin banyak ditangani dengan algoritma SI, dan SI telah terbukti sebagai teknik yang dapat menyelesaikan masalah komputasi

dan menemukan subset fitur yang optimal.

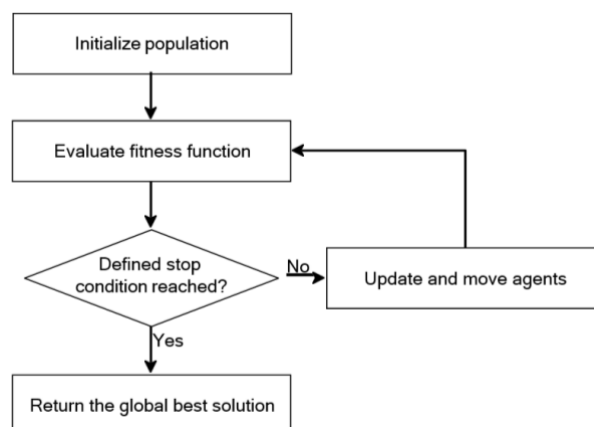
Secara istilah, *swarm* adalah kelompok, kumpulan, kawanan, koloni, kerumunan, rombongan, atau segala hal yang bersifat banyak, kolektif, dan massif. SI dapat diartikan sebagai kecerdasan kolektif atau kecerdasan kelompok [2]. Particle Swarm Optimization (PSO), dan Ant Colony Optimization (ACO) merupakan algoritma optimasi yang sering digunakan untuk menyelesaikan masalah FS. Selain itu, terdapat algoritma Grey Wolf Optimizer (GWO) yang juga dapat mengatasi problem FS. ACO diperkenalkan pada awal 1990-an. Kebiasaan mencari makan pada *ant* biologis, terutama bagaimana mereka menggunakan feromon untuk berkomunikasi dalam koloni mereka secara informal, menjadi inspirasi utama bagi ACO[2]. Sedangkan PSO merupakan algoritma terinspirasi dari tingkah laku sosial kawanan burung yang terbang (*bird flocking*) atau gerombolan ikan yang berenang (*fish schooling*) [3]. Algoritma GWO berdasarkan perilaku perburuan mangsa secara alami dari kawanan serigala abu-abu (*grey wolf*). Perilaku perburuan mangsa ini yang diadopsi ke dalam algoritma optimisasi. GWO berperan dalam eksploitasi untuk mendapatkan solusi global secara cepat atau konvergensi yang cepat[4].

Tujuan dari paper ini antara lain yaitu tinjauan literatur yang komprehensif tentang algoritma kecerdasan *swarm*. Selain itu, kami juga mengusulkan diagram hybrid SI untuk FS. Kami juga menyajikan gambaran komprehensif tentang framework SI, algoritma SI untuk FS di dalam *text mining*.

## 2. METODE PENELITIAN

### 2.1. Swarm Intelligence

Swarm tidak memiliki manajemen luar, namun masing-masing agen dalam kelompok mengendalikan perilakunya secara mandiri. Keunggulan kecerdasan *swarm* ini terlihat dalam perubahan lingkungan di mana agen merespons dengan baik terhadap perubahan tersebut, memodifikasi perilaku mereka, dan beradaptasi secara mandiri [5]. SI termasuk *nature-inspired meta-heuristics*, yaitu framework algoritma yang terinspirasi dari alam dan memiliki kemampuan teknik yang lebih untuk mengoptimalkan pada saat pencarian atau penelusuran solusi pada berbagai macam permasalahan [6]. Framework dari SI terdiri dari inisiasi populasi, mendefinisikan kondisi berhenti, evaluasi fungsi kebugaran, update dan move agen, serta kembali ke solusi global terbaik. Gambar 1 menunjukkan kerangka dari SI [7].



Gambar 1. Framework Swarm Intelligence

### 2.2. Ant Colony Optimization

Algoritma Ant Colony Optimization (ACO) diperkenalkan oleh Marco Dorigo pada tahun 1991 dalam thesis PhD-nya yang kemudian dipublikasikan dengan nama Ant System[8]. Algoritma ini terinspirasi oleh tingkah laku koloni semut (*ant*). Semut menggunakan isyarat kimiawi untuk berkomunikasi ketika berjalan dari sumber makanan ke sarang dan sebaliknya, ant meletakkan suatu zat pheromone di sepanjang jalur yang mereka lalui. Ketika hormon tersebut disekresikan sebagai isyarat oleh seekor ant, maka ant yang lain bisa mengenalinya. Ketika mencari makanan, pada awalnya semut-semut berkeliling di daerah sekitar sarangnya secara acak. Dalam perjalanan, mereka meninggalkan pheromone sebagai pemandu untuk menemukan sumber makanan. Jumlah pheromone yang ditinggalkan bergantung pada jumlah makanan yang ditemukan. Semakin banyak makanan yang didapat, semakin banyak pula pheromone yang ditinggalkan [9].

Dalam FS, saat membangun solusi, sebelum melakukan traversal pada graf, *artificial ant* harus diletakkan dalam kondisi memori kosong yang dilakukan dengan mengatur semua array. Kemudian setiap *ant* harus diberi simpul atau fitur awal secara acak. Mereka membangun tur lengkap dengan menerapkan aturan pilihan pada setiap langkah konstruksi berdasarkan informasi heuristik dan jalur pheromone[9]. Aturan transisi probabilistik adalah membangun dua parameter yaitu informasi heuristik dan tingkat pheromone, yang dinotasikan dalam Persamaan (1) [10].

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_i(t)]^A [\eta_i]^B}{\sum_{j \in J^k} [\tau_j(t)]^A [\eta_j]^B}, & \text{if } i \in J^k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Diketahui bahwa:

- $p_{ij}^k(t)$  : probabilitas ant k memilih fitur i
- $J^k$  : jumlah fitur yang belum dikunjungi oleh ant k
- $\eta_i$  : nilai heuristic pada fitur i
- $\eta_j$  : nilai heuristic pada fitur-j
- $\tau_i(t)$  : nilai pheromone pada fitur i
- $\tau_j(t)$  : nilai pheromone pada fitur-j
- A : pembobotan pheromone
- B : pembobotan heuristic

### 2.3. Particle Swarm Optimization

Particle Swarm Optimization (PSO) dikenalkan oleh James Kennedy dan Russ Eberhart pada tahun 1995[11]. PSO terinspirasi oleh tingkah laku sosial kawanannya burung yang terbang berduyun-duyun (*bird flocking*) atau gerombolan ikan yang berenang berkelompok (*fish schooling*). Ribuan ikan sanggup bergerak cepat tanpa saling berbenturan. Ratusan bahkan ribuan burung bangau mampu terbang membentuk formasi tertentu tanpa saling bertabrakan, padahal jarak antaranya sangat dekat. Burung dan ikan memiliki kecerdasan yang luar biasa dalam mengatur kecepatan terbang atau berenang sehingga jarak tetap terjaga[1].

PSO menggunakan suatu partikel yang tersusun atas tiga buah vektor dan dua nilai fitness. Tiga vektor tersebut yaitu X, P dan V. Vektor X adalah posisi partikel saat ini dalam ruang pencarian. P ialah posisi solusi terbaik yang ditemukan partikel tersebut. V adalah arah terbang partikel. Dua nilai fitness yaitu *x-fitness* dan *p-fitness* [12]. Sebagaimana ciri khas SI, algoritma PSO dapat menemukan solusi terbaik atau optimal. Algoritma PSO dimulai dengan sekumpulan partikel (calon solusi) yang dibangkitkan secara acak. Kemudian kualitas setiap partikel dievaluasi menggunakan fungsi fitness. Selanjutnya, partikel-partikel akan terbang mengikuti partikel terbaik. Pada setiap generasi, setiap partikel diperbarui mengikuti dua nilai terbaik. Yang

pertama merupakan nilai fitness terbaik yang dicapai oleh satu partikel saat ini. Nilai ini dilambangkan sebagai  $p$ . Sementara itu, nilai fitness terbaik kedua merupakan nilai fitness terbaik yang dicapai oleh semua partikel dalam topologi ketetanggaan. Indeks  $g$  digunakan untuk menunjukkan partikel dengan fitness terbaik tersebut. Jika menggunakan topologi ring, cara ini disebut PSO versi global. Sedangkan jika menggunakan topologi star, maka ini disebut sebagai PSO versi lokal. Jika telah diperoleh dua nilai terbaik, suatu partikel  $i$  pada posisi  $X_i$ , memperbarui *vector velocity* dan kemudian memperbarui posisi menggunakan Persamaan (2) dan (3) [1].

$$v_{id} = v_{id} + \varphi_1 \times r \times (p_{id} - x_{id}) + \varphi_2 \times r \times (p_{gd} - x_{id}) \quad (2)$$

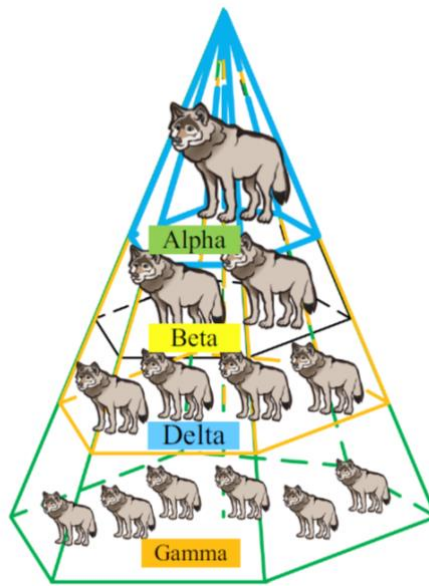
$$x_{id} = x_{id} + v_{id} \quad (3)$$

Diketahui  $i$  adalah partikel ke- $i$ ,  $d$  adalah dimensi ke- $d$ ,  $\varphi_1$  adalah laju belajar (*learning rates*) untuk komponen *cognition* (kecerdasan individu),  $\varphi_2$  adalah laju belajar untuk komponen *social* (hubungan sosial antar-individu),  $p$  adalah vektor nilai fitness terbaik yang dihasilkan sejauh ini,  $g$  adalah indeks partikel dengan fitness terbaik di dalam topologi ketetanggaan, dan  $r$  adalah bilangan acak dalam interval  $[0,1]$  [13].

#### 2.4. Grey Wolf Optimizer

Algoritma *Grey Wolf Optimizer* (GWO) diusulkan oleh Mirjalili dkk di tahun 2014 [4]. Sesuai dengan namanya, GWO terinspirasi oleh perilaku, khususnya dalam teknik berburu dan hierarki sosial dari serigala abu-abu. *Grey wolf* atau serigala abu-abu (*Canis lupus*) termasuk dalam keluarga *Canidae* [14]. *Grey wolf* dianggap sebagai predator puncak yang artinya bahwa mereka berada di puncak rantai makanan. *Grey wolf* kebanyakan lebih suka hidup dalam satu kawanan dengan jumlah rata-rata 5 hingga 12 ekor. Hirarki sosial *grey wolf* dibagi menjadi empat kelompok, alfa, beta, delta, dan omega, seperti terlihat pada Gambar 2. Serigala alpha ( $\alpha$ ) adalah para pemimpin dan pengambil keputusan. Serigala alfa disebut serigala dominan karena perintahnya harus diikuti oleh kelompok. *Alpha wolf* tidak harus anggota terkuat dari kawanan tetapi yang terbaik dalam hal mengelola kawanan. Hal ini menunjukkan bahwa organisasi dan disiplin suatu kawanan lebih penting daripada kekuatannya [15].

Tingkatan kedua adalah serigala Beta ( $\beta$ ) yang merupakan asisten alpha dalam memimpin grup. Tingkatan selanjutnya ialah serigala Delta ( $\delta$ ). Serigala tingkat ini mematuhi perintah dari alfa dan beta. Penjaga, pemburu, penatua dan pengasuh termasuk dalam serigala delta. Serigala yang tersisa disebut omega ( $\omega$ ) yang mengikuti serigala lain dalam gerakan mereka. Kecerdasan, kepemimpinan, dan kecenderungan berburu serigala abu-abu di alam liar menjadi sumber inspirasi utama algoritma GWO [10].



Gambar 2. Struktur hirarki Grey wolf

GWO dimulai dengan populasi acak awal dan dimodifikasi selama iterasi. *Grey wolf* mengepung (mengelilingi atau mengitari) mangsa selama perburuan. *Wolf* menyelesaikan perburuan dengan menyerang mangsanya ketika berhenti bergerak. Hal ini sama dengan eksploitasi solusi dalam masalah optimisasi. Persamaan (4) digunakan untuk memodelkan perilaku *grey wolf* secara matematis [16].

$$\begin{aligned} \vec{D} &= |\vec{C} \cdot \vec{X}_p(t) - \vec{X}(t)| \\ \vec{X}(t + 1) &= \vec{X}_p(t) - \vec{A} \cdot \vec{D} \end{aligned} \tag{4}$$

Untuk membangun model pendekatan mangsa secara matematis dengan mengurangi nilai  $\vec{a}$ . Dimana nilai komponen-komponen  $\vec{a}$  menurun secara linier dari 2 menjadi 0 selama iterasi. Sedangkan  $\vec{r}_1$  dan  $\vec{r}_2$  adalah vector acak dalam interval [0, 1]. Model matematisnya menggunakan Persamaan (5). Diketahui bahwa  $\vec{A}$ ,  $\vec{C}$  adalah koefisien vektor *wolf*. Sedangkan  $\vec{r}_1$ ,  $\vec{r}_2$  merupakan vektor acak [17].

$$\begin{aligned} \vec{A} &= 2\vec{a} \cdot \vec{r}_1 - \vec{a} \\ \vec{C} &= 2 \cdot \vec{r}_2 \end{aligned} \tag{5}$$

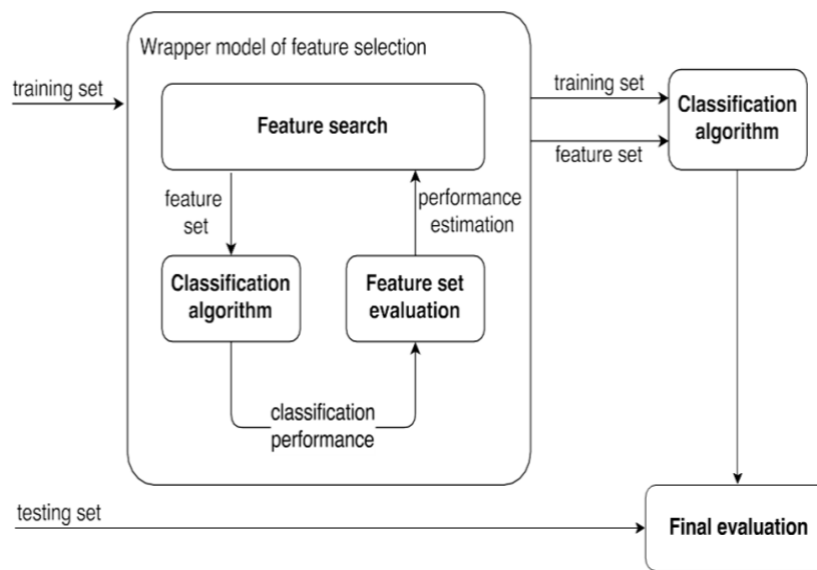
Untuk menstimulasikan secara matematis perilaku perburuan serigala, diasumsikan bahwa alpha (solusi kandidat terbaik), beta, dan delta memiliki pengetahuan yang lebih baik tentang lokasi potensial mangsa. Dengan demikian, tiga solusi terbaik yang diperoleh harus disimpan dan agen pencarian lain termasuk omega harus memperbarui posisi agen pencarian terbaik [18].

### 3. HASIL DAN PEMBAHASAN

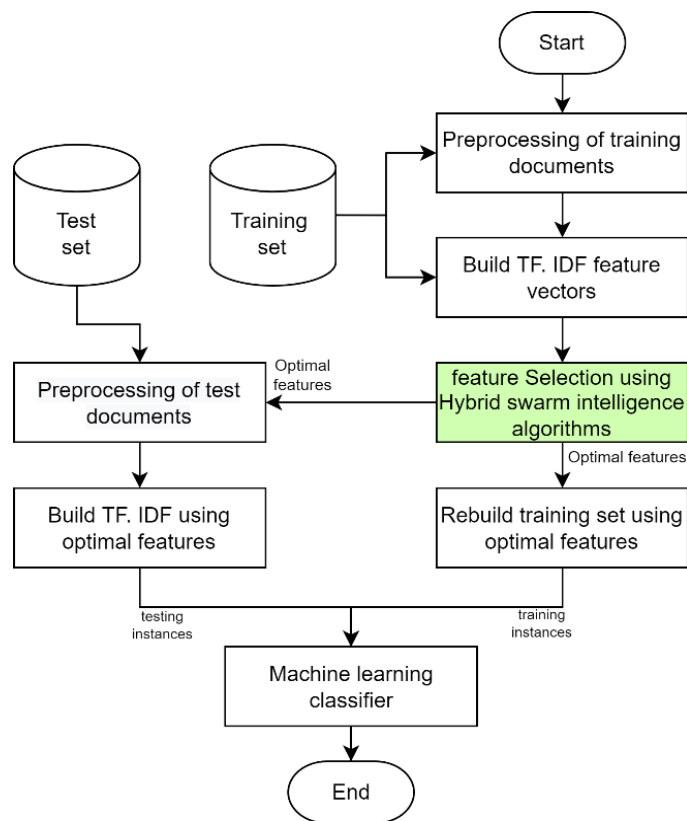
Secara umum, teknik FS yang sering digunakan dalam mengatasi problem tersebut. FS bekerja dengan menghilangkan fitur-fitur yang tidak relevan sehingga dapat mengoptimalkan kinerja ML [19]. Jenis pendekatan dalam FS adalah filter, wrapper, embedded [19]. Teknik filter menganalisis dan memilih subkumpulan fitur berdasarkan kualitas data yang tersedia, bukan

berdasarkan algoritme. Sedangkan, pendekatan FS wrapper bahwa suatu algoritma (seperti algoritma klasifikasi) dan metrik evaluasi menilai subset fitur (misalnya akurasi klasifikasi). Pendekatan embedded merupakan bagian dari proses FS secara otomatis tertanam dalam suatu algoritma [20]. Pendekatan SI adalah yang paling efektif untuk mengoptimalkan proses seleksi fitur yaitu model FS wrapper. Untuk mengevaluasi kualitas subset fitur tertentu, menerapkan teknik ML, yang secara komputasi sangat mahal. Meningkatkan proses FS dalam model wrapper; algoritma SI berupaya mengidentifikasi subset mana dari semua fitur yang tersedia yang berpasangan dengan algoritma ML yang mempunyai performa prediksi tertinggi [21]. Gambar 3 menampilkan struktur umum model wrapper untuk FS. Teknik Wrapper mengurangi jumlah ruang pencarian yang tersedia untuk menemukan fitur. Algoritma SI untuk FS telah dikembangkan dan banyak penelitian yang menggabungkan dua atau lebih SI untuk FS. Tabel 1 menyajikan beragam penerapan hybrid SI dalam mengatasi problematika FS.

Kami mengusulkan diagram hybrid SI untuk FS ditunjukkan pada Gambar 4. Tahapan-tahapan dalam teks FS antara lain adalah preprocessing, pemilihan fitur, pembuatan model klasifikasi, serta tugas evaluasi. Pada preprocessing, menghilangkan kata-kata seperti item leksikal dan kata-kata yang tidak umum—sering ditemukan dalam dokumen teks namun kurang informasi yang membantu membedakan kelas teks. Sekelompok kata pilihan yang sangat informatif disimpan di memori untuk mewakili teks sebagai vektor fitur. Proses ekstraksi fitur dilakukan melalui komputasi n-gram Term Frekuensi-Invers Dokumen Frekuensi (TF-IDF) untuk mendapatkan vektor fitur.



Gambar 3. Struktur pendekatan Wrapper dalam FS



Gambar 4. Blok diagram hybrid SI untuk seleksi fitur

#### 4. KESIMPULAN

Teknik pemilihan fitur atau *Feature Selection* (FS) diklasifikasikan menjadi tiga jenis yaitu strategi filter, wrapper, embedded. Penelitian ini mengkaji strategi wrapper berbasis FS kontemporer dengan menggunakan kombinasi dua algoritma *Swarm Intelligence* (SI). Kami menganalisis berbagai pendekatan berbasis *swarm* yang memerlukan hibridisasi dengan ACO atau PSO ataupun GWO dengan algoritma SI lainnya. Teknik FS dengan algoritma SI bertujuan untuk menemukan subset fitur yang relevan dan meningkatkan performa klasifikasi lebih bagus.

Diharapkan penelitian di masa mendatang, penerapan SI untuk berbagai bidang, tidak hanya bidang komputer namun diimplementasikan pada berbagai sektor seperti pertanian, perbankan, kesehatan dan sebagainya. Dengan demikian, penggunaan algoritma SI semakin berkembang dan memberikan solusi yang efektif dan efisien di era *Big Data*.

Tabel 1. Penerapan Hibrid SI dalam FS

Penulis	Metode	Prinsip Penggabungan	Domain
[2]	ACO-PSO	ACO-PSO untuk mengurangi jumlah fitur yang tak relevan	Klasifikasi
[22]	GWO-CS	Serigala Alfa, Serigala Beta, dan Serigala Delta adalah tiga solusi teratas yang telah diperbarui GWO menggunakan fungsi pencarian global algoritma Chi Square (CS).	Klasifikasi
[23]	ACO-ABC	Kombinasi ACO dan Artificial Bee Colony (ABC)	Klasifikasi
[12]	PSO-GWO	Memasukkan PSO ke dalam GWO untuk menetapkan label kelas dan menghasilkan berbagai cluster dokumen teks.	Klasterisasi teks
[15]	BGWO-PSO	Hibrid Binary GWO-PSO untuk mencari subset fitur terbaik.	Klasifikasi
[18]	GWO-PSO	Mencari fitur terbaik menggunakan kemampuan eksploitasi dan eksplorasi GWO dan PSO.	Klasifikasi
[16]	GWO-GOA	Hibridisasi GWO dan Grasshopper Optimization Algorithm (GOA) dalam FS, dengan cara terlebih dahulu memilih lokal optima dari dokumen teks, kemudian memilih global optima terbaik dari local optimum.	Klasterisasi teks
[24]	BGWO-BPSO	Menggabungkan Binary PSO dan Binary GWO untuk meningkatkan kemampuan pencarian global ketika memecahkan masalah FS dalam kumpulan data berdimensi tinggi.	Klasifikasi

### UCAPAN TERIMA KASIH

Penulis mengucapkan terimakasih kepada Fakultas Teknik, Universitas Mulawarman yang telah mendukung penelitian ini.

### DAFTAR PUSTAKA

- [1] S. Suyanto, “Swarm Intelligence Komputasi Modern untuk Optimasi dan Big Data Mining,” *Inform. Bandung*, 2017.
- [2] K. Menghour and L. Souici-Meslati, “Hybrid ACO-PSO based approaches for feature selection,” *Int J Intell Eng Syst*, vol. 9, no. 3, pp. 65–79, 2016.
- [3] N. Chopra, G. Kumar, and S. Mehta, “Hybrid GWO-PSO algorithm for solving convex economic load dispatch problem,” *Int J Res Adv Technol*, vol. 4, no. 6, pp. 37–41, 2016.
- [4] S. Mirjalili, S. M. Mirjalili, and A. Lewis, “Grey wolf optimizer,” *Adv. Eng. Softw.*, vol. 69, pp. 46–61, 2014.
- [5] E. Indramaya and S. Suyanto, “Comparative Study of Recent Swarm Algorithms for Continuous Optimization,” *Procedia Comput. Sci.*, vol. 179, no. 2019, pp. 685–695, 2021, doi: 10.1016/j.procs.2021.01.056.



- [6] I. Cholissodin and E. Riyandani, "Swarm Intelligence," *Malang Fak. Ilmu Komput. Univ. Brawijaya*, 2016.
- [7] L. Brezočnik, I. Fister, and V. Podgorelec, "Swarm intelligence algorithms for feature selection: A review," *Appl. Sci.*, vol. 8, no. 9, 2018, doi: 10.3390/app8091521.
- [8] S. Mirjalili, "Ant colony optimisation," *Stud. Comput. Intell.*, vol. 780, no. November, pp. 33–42, 2019, doi: 10.1007/978-3-319-93025-1\_3.
- [9] J. A. Widians, R. Wardoyo, and S. Hartati, "Feature selection based on chi-square and ant colony optimization for multi-label classification," *Int. J. Electr. Comput. Eng.*, vol. 14, no. 3, p. 3303, 2024, doi: 10.11591/ijece.v14i3.pp3303-3312.
- [10] J. A. Widians, R. Wardoyo, and S. Hartati, "A Study on Text Feature Selection Using Ant Colony and Grey Wolf Optimization," in *2022 Seventh International Conference on Informatics and Computing (ICIC)*, 2022, pp. 1–7.
- [11] N. Singh and S. B. Singh, "Hybrid algorithm of particle swarm optimization and grey wolf optimizer for improving convergence performance," *J. Appl. Math.*, vol. 2017, 2017.
- [12] C. Vidyadhari, N. Sandhya, and P. Premchand, "Particle grey wolf optimizer (pgwo) algorithm and semantic word processing for automatic text clustering," *Int. J. Uncertainty, Fuzziness Knowledge-Based Syst.*, vol. 27, no. 02, pp. 201–223, 2019.
- [13] C. Shen and K. Zhang, "Two-stage improved Grey Wolf optimization algorithm for feature selection on high-dimensional classification," *Complex Intell. Syst.*, pp. 1–21, 2021.
- [14] H. Faris, I. Aljarah, M. A. Al-Betar, and S. Mirjalili, "Grey wolf optimizer: a review of recent variants and applications," *Neural Comput. Appl.*, vol. 30, pp. 413–435, 2018.
- [15] Q. Al-Tashi, S. J. A. Kadir, H. M. Rais, S. Mirjalili, and H. Alhussian, "Binary optimization using hybrid grey wolf optimization for feature selection," *Ieee Access*, vol. 7, pp. 39496–39508, 2019.
- [16] R. Purushothaman, S. P. Rajagopalan, and G. Dhandapani, "Hybridizing Gray Wolf Optimization (GWO) with Grasshopper Optimization Algorithm (GOA) for text feature selection and clustering," *Appl. Soft Comput.*, vol. 96, p. 106651, 2020.
- [17] M. A. M. Shaheen, H. M. Hasanien, and A. Alkuhayli, "A novel hybrid GWO-PSO optimization technique for optimal reactive power dispatch problem solution," *Ain Shams Eng. J.*, vol. 12, no. 1, pp. 621–630, 2021.
- [18] E.-S. El-Kenawy and M. Eid, "Hybrid gray wolf and particle swarm optimization for feature selection," *Int. J. Innov. Comput. Inf. Control*, vol. 16, no. 3, pp. 831–844, 2020.
- [19] J. Žižka, F. Dařena, and A. Svoboda, *Text mining with machine learning: principles and techniques*. CRC Press, 2019.
- [20] X. Zhou *et al.*, "A survey on text classification and its applications," *Web Intell.*, vol. 18, no. 3, pp. 205–216, 2020, doi: 10.3233/WEB-200442.

- [21] O. M. Alyasiri, Y.-N. Cheah, and A. K. Abasi, "Hybrid filter-wrapper text feature selection technique for text classification," in *2021 International Conference on Communication & Information Technology (ICICT)*, 2021, pp. 80–86.
- [22] H. Xu, X. Liu, and J. Su, "An improved grey wolf optimizer algorithm integrated with cuckoo search," in *2017 9th IEEE international conference on intelligent data acquisition and advanced computing systems: technology and applications (IDAACS)*, 2017, vol. 1, pp. 490–493.
- [23] P. Shunmugapriya and S. Kanmani, "A hybrid algorithm using ant and bee colony optimization for feature selection and classification (AC-ABC Hybrid)," *Swarm Evol. Comput.*, vol. 36, pp. 27–36, 2017.
- [24] J. Yousef, A. Youssef, and A. Keshk, "A Hybrid Swarm Intelligence Based Feature Selection Algorithm for High Dimensional Datasets," *IJCI. Int. J. Comput. Inf.*, vol. 0, no. 0, pp. 0–0, 2021, doi: 10.21608/ijci.2021.62499.1040.