

MODIFIKASI FRUIT FLY OPTIMIZATION ALGORITHM UNTUK OPTIMASI GENERAL REGRESSION NEURAL NETWORK PADA KASUS PREDIKSI TIME-SERIES

Ni Putu Novita Puspa Dewi¹, Ni Ketut Kertiasih², Ni Luh Dewi Sintiar³

^{1,3} Prodi Ilmu Komputer, Universitas Pendidikan Ganesha

² Prodi Teknologi Rekayasa Perangkat Lunak, Universitas Pendidikan Ganesha

email: novita.puspa.dewi@undiksha.ac.id¹

Abstrak

FOA merupakan algoritma optimasi *swarm intelligence* yang dikenal unggul dan cenderung sederhana untuk diimplementasikan, namun algoritma ini diketahui sulit untuk memecahkan masalah optimasi nonlinier berdimensi tinggi dan mudah jatuh ke dalam optimum lokal. Untuk mengatasi kekurangan ini, *immune algorithm* digunakan untuk menyeimbangkan kekurangan FOA guna meningkatkan efisiensi pencarian. Penelitian ini bertujuan untuk menggabungkan algoritma optimasi FOA dengan *immune algorithm* untuk digunakan mengoptimasikan model prediksi GRNN. Model *hybrid* dari GRNN dan FOA modifikasi (IAFOA) akan diuji-coba terhadap beberapa dataset *time-series* di berbagai domain. Performanya dibandingkan dengan model FOA *basic* untuk melihat dampak jelas dari proses modifikasi tersebut terhadap performa model GRNN dalam melakukan prediksi. Hasil pengujian menunjukkan bahwa eror prediksi RMSE dan MAE dari IAFOA berhasil mengungguli FOA pada 4 kasus *training* dan *testing* dan 1 kasus lebih unggul pada proses *training*. Berdasarkan pengujian yang dilakukan kepada 3 dataset (6 kasus), IAFOA menghasilkan rata-rata eror prediksi lebih kecil yaitu RMSE sebesar 35348.63 dan MAE 26699.02 dibandingkan FOA dengan rata-rata eror prediksi secara berturut-turut 35792.59 dan 26967.12.

Kata kunci: Algoritma optimasi, GRNN, FOA modifikasi, FOA dengan *immune algorithm*, IAFOA.

Abstract

FOA is a *swarm intelligence optimization algorithm* known to be superior and tends to be simple to implement. Unfortunately, this algorithm is known to be difficult to solve high-dimensional nonlinear optimization problems and easily falls into local optimum. To overcome this deficiency, an *immune algorithm* is used to balance the drawbacks of FOA to improve search efficiency. This study aims to combine the FOA optimization algorithm along with the *immune algorithm* in order to optimize the GRNN prediction model. The hybrid model of GRNN and modified FOA (IAFOA) will be tested against several *time-series datasets* in various domains. Its performance is compared with the basic FOA model to see the clear impact of the modification process on the performance of the GRNN model in making predictions. The test results show that the RMSE and MAE of IAFOA managed to outperform FOA in 4 cases of both *training* and *testing* and 1 case was superior in the *training* process. Based on tests conducted on 3 datasets (6 cases), IAFOA produced a smaller average prediction error, namely RMSE of 35348.63 and MAE 26699.02 compared to FOA with average prediction errors of 35792.59 and 26967.12 respectively.

Keywords : Optimization algorithm, GRNN, modified FOA, FOA with *immune algorithm*, IAFOA

Diterima Redaksi: 28-11-2022 | Selesai Revisi: 06-12-2022 | Diterbitkan Online: 27-12-2022

DOI: <https://doi.org/10.23887/janapati.v11i3.54521>

PENDAHULUAN

Algoritma optimasi berperan dalam menemukan solusi nilai x untuk mengoptimalkan sebuah fungsi $f(x)$ agar bernilai sekecil atau sebesar mungkin untuk suatu fungsi f yang diberikan. Algoritma optimasi banyak diaplikasi untuk mendukung pembelajaran mesin, salah satunya untuk

keperluan prediksi. Optimasi terhadap model prediksi perlu dilakukan untuk memperoleh model yang optimal.

Perkembangan keilmuan di bidang ilmu komputer yang didukung oleh teknologi memungkingkan banyak algoritma optimasi dikembangkan, salah satunya algoritma yang diadaptasi dari kecerdasan makhluk biologis.

Optimasi yang didasari oleh kecerdasan komputasional dan kecerdasan buatan yang tergolong sebagai *bio-inspired algorithms* terutama algoritma berbasis *swarm intelligence* sangat populer penggunaannya. *Swarm intelligence* yang merupakan algoritma metaheuristik yang diilhami oleh alam ini menjadi salah satu algoritma yang paling banyak digunakan untuk optimasi dan kecerdasan komputasi [1], efektif untuk memecahkan masalah optimasi yang kompleks [2].

Beberapa algoritma optimasi yang tergolong sebagai *swarm intelligence* adalah *Fruit Fly Optimization Algorithm* (FOA). FOA adalah kecerdasan kelompok yang diusulkan oleh Wen-Tsao Pan pada tahun 2011 [3]. Aplikasi FOA saat ini sebagian besar fokus pada prediksi/prakiraan dan pengambilan keputusan. Pada beberapa implementasinya, FOA terbukti dalam segi kalkulasi, kompleksitas, stabilitas, dan presisi mengungguli beberapa algoritma sejenis seperti *Ant Colony Optimization* (ACO), *Fish Swarm Algorithm* (FSA), *Locust Swarm Algorithm* (LSA), dan *Particle Swarm Optimization* (PSO) [2]. Jika dibandingkan dengan algoritma optimasi yang lain, FOA lebih sederhana dan mudah diprogram, selain itu memiliki parameter yang lebih sedikit dan efisiensi pencarian yang lebih baik.

Meskipun cukup unggul dibandingkan algoritma optimasi lain dan cenderung sederhana untuk diimplementasikan, FOA diketahui sulit untuk memecahkan masalah optimasi nonlinier berdimensi tinggi dan mudah untuk jatuh ke dalam *local optimum* [4]. Untuk mengatasi kekurangan dari FOA, *immune algorithm* digunakan untuk menyeimbangkan kekurangan FOA yang rentan jatuh ke dalam *local optimum* dan meningkatkan efisiensi pencarian.

Immune algorithm dikembangkan berdasarkan teori imunologi. Simulasi menunjukkan bahwa algoritma ini memiliki kinerja yang baik dalam menyelesaikan masalah optimasi kombinatorial [5]. Algoritma ini memiliki keunggulan berupa prinsip yang jelas, parameter yang lebih sedikit, dan pengoperasian yang sederhana. Penggunaan *immune algorithm* untuk melengkapi FOA dilakukan ketika jumlah langkah stasis evolusioner t lebih besar dari nilai ambang langkah stasis evolusioner T , maka *immune algorithm* akan diaktifkan untuk mengatasi cacat algoritma dasar lalat buah [4].

Metode yang diusulkan oleh Li and Han (2020) menunjukkan bahwa *immune algorithm*

memiliki kemampuan pencarian lokal yang kuat sehingga stabilitas algoritma optimasi yang distabilkan olehnya dapat meningkat dan efisiensinya dapat diakui [4]. Selain itu, penggunaan *immune algorithm* secara luas pada berbagai bidang membuktikan ketahanan memecahkan masalah *knapsack* 0-1 [6], [7]; memecahkan masalah *multi-distribution center location* dengan sifat NP-hard [5]; optimasi terhadap *fitness function* untuk parameter kontrol PID [8]; komposisi layanan *cloud* yang sadar *Quality of Service* [9]; dan optimasi struktur rangka [10].

FOA sebagai algoritma optimasi dapat dimodifikasi dengan algoritma lain yang dapat melengkapi kekurangannya, salah satunya *immune algorithm*. Modifikasi FOA ini akan diterapkan pada algoritma prediksi untuk kepentingan prakiraan terhadap suatu nilai di masa yang akan datang. Performa FOA modifikasi akan diujicoba implementasi terhadap model prediksi *time-series*. Model prediksi yang dimaksud adalah *General Regression Neural Networks* (GRNN). Dasar dari operasi GRNN adalah teori regresi non-linear (kernel) di mana estimasi nilai output yang diharapkan ditentukan oleh set input [11]. Keberadaan hubungan data dapat ditemukan lebih cepat tanpa memerlukan asumsi data [12]. Perkiraan *weight* dan jumlah unit dalam *pattern layer* dilakukan tanpa melalui banyak iterasi [13].

Beberapa penelitian terdahulu telah dilakukan untuk membuktikan bahwa GRNN adalah model yang cukup bagus untuk melakukan prediksi. GRNN telah menunjukkan kinerja yang baik pada prediksi saham [13], prediksi ketebalan lapisan *ic* pada saluran transmisi listrik [14], dan berhasil mengungguli metode statistik ARIMA untuk memprediksi IHSG (Indeks Harga Saham Gabungan) untuk beberapa periode berikutnya [12].

Nilai *smoothing parameter* pada GRNN perlu diatur untuk mencapai performa prediksi yang baik. Penentuan nilai ini yang dilakukan secara *hand tuning* akan memakan waktu cukup lama selain juga karena ruang pencarian yang cukup luas. Algoritma optimasi sangat membantu dalam menemukan nilai tersebut secara efektif [15]. GRNN yang dioptimasi dengan FOA menambah kemampuan pencarian global yang cukup baik untuk prediksi arus listrik terpakai pada penyulang [15].

Penelitian ini bertujuan untuk meningkatkan kinerja FOA *basic*, algoritma

optimasi modifikasi FOA dirancang untuk digabungkan dengan *immune algorithm* dan digunakan untuk mengoptimasi GRNN pada kasus prediksi *time-series*. *Immune algorithm* akan diutilisasi untuk memperbaiki kekurangan FOA yang rentan jatuh ke nilai optimum lokal pada proses pelatihan model prediksi GRNN. Model *hybrid* dari GRNN dan FOA modifikasi akan diuji-coba terhadap beberapa dataset *time-series* di berbagai domain, performanya akan dibandingkan dengan model FOA *basic* untuk melihat dampak jelas dari proses modifikasi tersebut terhadap performa model GRNN dalam melakukan prediksi. Performa model yang diukur dalam hal ini adalah kekuatan prediksi (error prediksi) yang dihitung dengan matrik performa RMSE (*Root Mean Square Error*) dan MAE (*Mean Absolute Error*).

PREDIKSI TIME-SERIES

Time series merupakan serangkaian pengamatan terhadap suatu peristiwa, kejadian, gejala atau perubahan yang disajikan dalam interval waktu yang tetap. *Time series* biasanya dimodelkan melalui proses stokastik $Y(t)$, yaitu sekuens (urutan) variabel acak [16]. Fitur utama dari kebanyakan *time-series* adalah adanya tren, variasi musiman, dan observasi yang berdekatan dalam waktu cenderung berkorelasi [17].

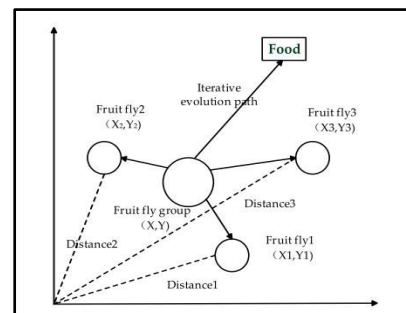
Prediksi *time-series* merupakan prediksi ilmiah berdasarkan data historis dalam interval waktu tertentu [18]. Prediksi ini melibatkan pembangunan model melalui analisis terhadap data historis dan menggunakan model ini untuk melakukan pengamatan dan mendukung pengambilan keputusan strategis di masa depan [19]. Dengan kata lain, prediksi *time-series* menggunakan data (observasi) pada waktu t dan berusaha untuk memperkirakan $Y(t+h)$, hanya menggunakan informasi yang tersedia pada waktu t . Prediksi *time-series* digunakan dalam banyak domain, antara lain: peramalan cuaca dan iklim, peramalan ekonomi, peramalan untuk bidang teknik, peramalan di bidang kesehatan, peramalan keuangan, peramalan ritel dan bisnis, peramalan studi lingkungan dan studi sosial, dan masih banyak lagi.

FRUIT FLY OPTIMIZATION

Fruit Fly Optimization Algorithm (FOA) merupakan algoritma *swarm intelligence* untuk optimasi yang mengadopsi perilaku lalat buah dalam menemukan makanan [3]. Algoritma ini berusaha untuk mencapai *global optimal* dari sebuah *objective function*. Analogi kerja dari

algoritma ini meniru cara kerja lalat buah yang bekerja dalam kelompok menggunakan kemampuan penginderaan dan persepsi dalam mencari makanan.

Terdapat dua langkah utama dalam proses pencarian makanan oleh lalat buah yang diimitasi oleh algoritma FOA [3]. Pertama, lalat buah menggunakan organ penciuman untuk merasakan bau (aroma buah) di udara kemudian terbang ke lokasi asal bau tersebut. Kedua, lalat buah mengadopsi pandangan tempat makanan dan posisi berkumpul lalat buah lainnya dan terbang ke arah itu. Proses pencarian makanan secara berulang dari kawanan lalat buah diilustrasikan pada Gambar 1.



Gambar 1. Proses iterasi dalam pencarian makanan oleh lalat buah [14]

Langkah kerja dari algoritma FOA adalah sebagai berikut:

1. Inialisasi ukuran populasi, iterasi gen, dan koordinat posisi X_0 dan Y_0 dari populasi lalat buah acak.
2. Menentukan arah terbang acak untuk individu lalat buah dan ukuran langkah untuk menemukan makanan dengan menggunakan kemampuan penginderaan mereka di mana $i = 1, 2, \dots, n - 1, n$, sehingga:

$$X_i = X_0 + \text{Random Value} \quad (1)$$

$$Y_i = Y_0 + \text{Random Value} \quad (2)$$

3. Karena lalat buah tidak dapat memperoleh posisi makanan, maka jarak $Dist_i$ antara individu dan asal lalat diperkirakan terlebih dahulu dan nilai penentuan konsentrasi bau S_i dihitung dengan:

$$Dist_i = \sqrt{X_i^2 + Y_i^2} \quad (3)$$

$$S_i = \frac{1}{Dist_i} \quad (4)$$

4. Nilai konsentrasi bau S_i diikutkan dalam fungsi *Fitness* untuk menentukan konsentrasi bau $Smell_i$ dari posisi individu lalat buah:

$$Smell_i = Fitness(S_i) \quad (5)$$

5. Individu dengan nilai konsentrasi bau terbaik dipilih di antara kumpulan lalat buah, termasuk konsentrasi dan koordinatnya. Nilai $Smell_i$ merupakan nilai yang menentukan seberapa baik konsentrasi bau pada setiap individu lalat buah. Nilai maksimal atau minimal dari $Smell_i$ ditentukan sesuai dengan keperluan optimasi. Untuk kasus prediksi, nilai $Smell_i$ maksimum mengindikasikan nilai eror terendah.

$$[best\ smell\ best\ index] = \min(Smell_i) \quad (6)$$

6. Jika nilai $bestSmell$ lebih kecil dari nilai $Smell_{best}$ saat ini, ganti nilai $Smell_{best}$ dengan $bestSmell$. Set lokasi baru dari kumpulan lalat buah ke koordinat dari individu lalat buah dengan nilai $bestSmell$:

$$Smell_{best} = bestSmell \quad (7)$$

$$X_0 = X(bestIndex) \quad (8)$$

$$Y_0 = Y(bestIndex) \quad (9)$$

7. Ulangi langkah 2-5 untuk tentukan apakah konsentrasi bau maksimum lebih tinggi dari generasi sebelumnya, lalu jalankan langkah (6) jika perlu. Hentikan proses iterasi apabila iterasi telah mencapai nilai maksimal iterasi.

IMMUNE ALGORITHM

Immune Algorithm (IA) merupakan algoritma optimasi bionik yang diperkenalkan pada tahun 1990 untuk memecahkan masalah [20]. Algoritma ini mensimulasikan sistem imun biologis untuk mengidentifikasi antigen (dalam hal ini fungsi objektif), diferensiasi sel, memori sistem kekebalan dan fungsi pengaturan diri [21]–[23]. Langkah kerja algoritma ini dijelaskan sebagai berikut [4]:

1. Mengenali antigen - input *objective function* dan kondisi pembatas sebagai antigen dari *immune algorithm*.
2. Menghasilkan antibodi awal – antibodi dihasilkan secara acak dalam ruang solusi.

3. Menghitung kompatibilitas (*fitness evaluation*). Berdasarkan fungsi *fitness evaluation*, afinitas antara antibody and antigen dan juga antar antibodi ditentukan. Kompatibilitas antara antigen dan antibodi A_v dihitung sebagai berikut:

$$A_v = \frac{1}{[1 + O_{Pt_v}]} \quad (10)$$

O_{Pt_v} merupakan degree kecocokan dari antigen dan antibodi, dan nilai dari A_v berada diantara 0 dan 1. Apabila nilai $O_{Pt_v} = 0$, $A_v = 1$ maka antibodi dianggap cocok dengan antigen, sehingga antibodi tersebut dianggap sebagai solusi optimal.

4. *Update memory units* – antibodi dengan afinitas terbesar terhadap antigen yang dihitung pada langkah 3 ditambahkan ke memory unit and menggantikan antibodi sebelumnya.
5. Menghitung nilai E_{xi} dari antibodi ke- i , nilai antibodi yang rendah akan ditekan melalui persamaan (11).

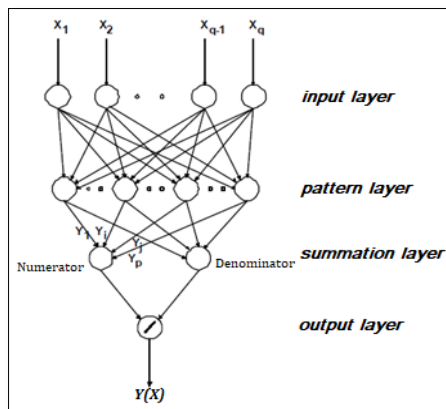
$$E_{xi} = \frac{A_i}{C_i} \quad (11)$$

Dimana A_i adalah afinitas antara antigen dan antibody ke- i , and C_i adalah jumlah antibodi ke- i ;

6. Membentuk antibodi baru – *Father Generation* menghasilkan antibodi generasi berikutnya melalui keturunan, mutasi, dan persilangan.
7. Apabila kondisi terpenuhi, maka iterasi dihentikan, jika tidak kembali ke langkah 3.

GENERAL REGRESSION NEURAL NETWORK

General Regression Neural Network (GRNN) merupakan salah satu varian dari *radial basis function networks* (RBFN). GRNN didasarkan pada jaringan regresi kernel. GRNN tidak memerlukan prosedur pelatihan yang berulang seperti pada jaringan progagasi balik (BPN). GRNN mendekati fungsi arbitrari antara vektor input dan vektor target.



Gambar 2. Arsitektur dari GRNN [24]

Arsitektur GRNN yang terdiri dari 4 layer seperti ditunjukkan pada Gambar 2, yaitu: *input layer*, *pattern layer*, *summation layer*, dan *output layer*. Notasi Y merupakan luaran (output) prediksi yang dihasilkan dari input X (vektor input). Vektor input dapat terdiri dari sejumlah q variabel prediktor. Input layer menerima vektor input X kemudian meneruskannya ke *pattern layer*. Transformasi nonlinear diterapkan pada *pattern layer* untuk menghitung *weight* setiap *training data point*. Persamaan (12) menunjukkan *Transfer function* dari neuron ke- i pada *pattern layer* P_i [14]:

$$P_i = \exp\left(\frac{-D_i^2}{2\sigma^2}\right), \text{ dimana } i = 1, 2, 3, \dots, n \quad (12)$$

Jarak D_i antara sampel pelatihan dan data poin yang menunjukkan nilai prediksi, digunakan untuk mengukur seberapa bagus setiap sampel pelatihan dapat merepresentasikan sampel prediksi dari X [24]. Dalam proses optimasi GRNN, hanya satu parameter *smoothing* yang harus disesuaikan satu kali melalui data, sehingga tidak ada prosedur iterasi yang dilakukan pada GRNN. Estimasi parameter tersebut dibatasi oleh nilai minimum dan maksimum data. Dalam hal ini, nilai σ adalah parameter *smoothing* yang dimaksud. Nilai dari D_i dihitung pada persamaan (13), dimana X merupakan input vector; X_i adalah sampel pembelajaran terhadap neuron ke- i .

$$D_i = (X - X_i)^T \cdot (X - X_i) \quad (13)$$

Neuron ke- i pada *pattern layer* menghitung output dari P_i (*weight* untuk sampel X_i). Hasilnya diterima oleh *summation layer*. Neuron pada *summation layer* bertindak sebagai numerator dan denominator. Fungsi objektif $Y(X)$ pada *output layer* merupakan pembagian antara numerator dan denominator [14] yang ditunjukkan pada persamaan (14).

Numerator menghitung *weighted sum* dari output untuk semua titik sampel pelatihan dan denominator menghitung penjumlahan dari semua *weight*.

$$Y(X) = \frac{\sum_i^n Y_i P_i}{\sum_i^n P_i} \quad (14)$$

METODOLOGI

Penelitian ini secara garis besar ingin mengimplementasikan proses optimasi FOA yang telah dimodifikasi pada GRNN untuk keperluan prediksi *time-series* pada beberapa domain. FOA merupakan algoritma optimasi yang akan dimodifikasi dengan *immune algorithm*. Algoritma modifikasi ini kemudian akan diimplementasikan pada GRNN untuk menemukan *parameter smoothing* optimal pada GRNN. Untuk mengetahui dampak dari modifikasi algoritma optimasi, maka performa model yang telah dioptimasi akan dinilai. GRNN dengan algoritma modifikasi akan diuji coba untuk melakukan prediksi pada beberapa dataset *time series*. Performa dari model dengan algoritma optimasi modifikasi akan dibandingkan dengan model dengan algoritma optimasi *basic*. Tahapan dari penelitian ditunjukkan pada Gambar 4.

Penelitian diawali dengan studi pustaka untuk menemukan beberapa penelitian terkait terutama yang berkaitan dengan *framework* penggabungan kedua algoritma untuk keperluan optimasi (FOA dan *immune algorithm*). Pengumpulan data dilakukan untuk mencari data yang digunakan sebagai *dataset* prediksi untuk menguji model. Pada tahap implementasi, model prediksi akan dibangun dengan menggunakan *dataset*. Terdapat beberapa sub-tahapan yang dijelaskan pada bagan dengan diagram putus-putus di sebelah kanan. Sub-tahapan implementasi terdiri dari *data preprocessing*, *data reframing*, *data split*, *build model*, dan *model optimization*. Proses implementasi akan menghasilkan model optimal dari proses optimasi. Model optimal ini kemudian akan diuji dengan dataset pengujian dan akan dibandingkan dengan performa model lain dengan algoritma optimasi *basic*. Analisis hasil dilakukan dengan membandingkan *metric* performa model untuk menemukan bagaimana dampak modifikasi terhadap algoritma optimasi. Hasil yang diperoleh dari penelitian akan menjadi dasar bagi penelitian lanjutan.

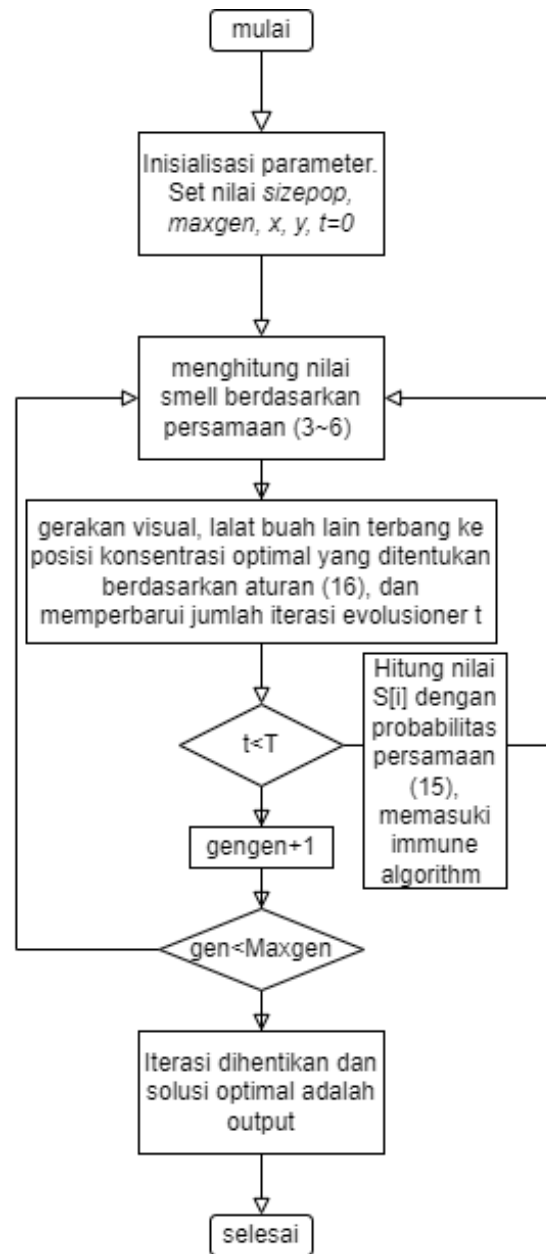
MODIFIKASI FOA (IAFOA)

Algoritma FOA digunakan untuk menginisiasi himpunan solusi yang layak. Kondisi pemicu untuk memasuki *Immune Algoritim* [25]

digunakan berdasarkan jumlah langkah stasis evolusioner t . Apabila nilai $t > T$ (ambang langkah stasis evolusioner), proses pencarian *Immune Algortihm* dipanggil, dan proses faktor imun dalam algoritma imun (sesuai dengan *drosophila* individu) mencari antigen (sesuai dengan sumber makanan dalam algoritma *drosophila*) memproduksi antibodi (solusi optimal) digunakan untuk memperluas ruang pencarian. Nilai $T = 6$ digunakan sebagai nilai ambang batas [4]. Jika kondisi pemicu terpenuhi, maka nilai probabilitas 0.25 [4] akan dihitung kembali berdasarkan probabilitas redistribusi faktor imun tetap pada persamaan 15 untuk optimasi ruang. Individu yang berbeda diberi probabilitas imun adaptif yang berbeda $S[i]$ sesuai dengan nilai *fitness* pada persamaan 15 dan 16. Proses optimasi ditunjukkan secara terstruktur pada Gambar 3.

$$S[i] = \frac{smellBest - best}{(smellBest - smellWorst) * 0.25} \quad (15)$$

$$\begin{cases} \text{if}(\text{bestSmell} < \text{Smellbest}) \\ \text{Smellbest} = \text{bestSmell1} \\ \text{else}(t = t + 1) \\ \text{Smellbest} = \text{bestSmell2} \\ \text{end} \end{cases} \quad (16)$$



Gambar 3. Proses optimasi dengan FOA modifikasi (IAFOA)

DATASET

Dataset yang digunakan merupakan data *time series* pada domain cuaca, iklim, dan energi. Dataset yang digunakan dalam eksperimen berasal dari Kaggle.com yakni: *Wind Speed Prediction Dataset* [26]; *Wind Energy in Germany* [27], dan *Daily Climate time series data* [28].

Wind Speed Prediction Dataset

Dataset berisi 6574 contoh respons rata-rata harian dari serangkaian 5 sensor variabel cuaca yang tertanam di stasiun meteorologi.

Perangkat itu terletak di lapangan di area yang sangat kosong, pada 21M. Data dicatat dari Januari 1961 hingga Desember 1978 (17 tahun). Dataset terdiri dari *Ground Truth* rata-rata curah hujan harian, suhu maksimum dan minimum, dan suhu minimum kotor.

Wind Energy in Germany

Dataset ini berisi pengukuran harian kapasitas angin, pembangkitan angin, dan suhu dari 01-01-2017 hingga 30-12-2019.

Daily Climate time series data

Dataset sepenuhnya didedikasikan untuk pengembang yang ingin melatih model tentang Prakiraan Cuaca untuk iklim India. Dataset ini diambil dari 1 Januari 2013 hingga 24 April 2017 di kota Delhi, India. Dataset terdiri dari 4 parameter yang terdiri dari rata-rata temperatur, kelembaban, kecepatan angin, tekanan rata-rata.

EXPERIMENTAL SETUP

Model dilatih dan diuji dengan menggunakan dataset seperti yang disebutkan pada metodologi. Pembagian dataset ditunjukkan pada Tabel 1. Model prediksi dilatih dengan menggunakan dataset *training* lalu diuji dengan dataset *testing*. FOA modifikasi akan disematkan pada GRNN untuk mengoptimalkan *smoothing parameter* (σ) pada GRNN. Input model adalah dataset *training* dan *testing* dalam bentuk *input sequence* sebagai prediktor untuk menentukan target prediksi.

Beberapa jenis *input sequence* dieksperimenkan dalam penelitian ini untuk melihat penggunaan *previous time-step* yang tepat. Nilai *previous time-step* untuk *reframing* data berkisar antara 1-10 hari. Ini berarti untuk memprediksi nilai pada hari tertentu digunakan prediktor berupa data 1 hingga 10 hari sebelumnya. Sedangkan untuk parameter pelatihan model prediksi seperti: *maxgen* dan *size pop* menggunakan nilai berturut-turut 30 dan

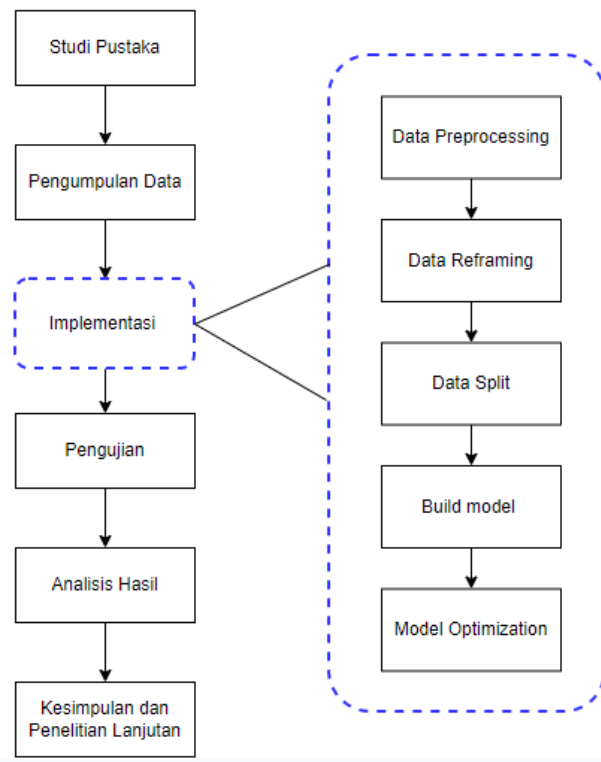
10. Kedua nilai parameter ini digunakan untuk ketiga dataset. Untuk evaluasi kinerja akhir pada dataset *testing*, model FOAGRNN dengan *smoothing parameter* optimal dan RMSE (*Root Mean Square Error*) terkecil akan dipilih untuk kemudian diuji dengan dataset *testing*.

HASIL DAN PEMBAHASAN

Pelatihan dan pengujian model prediksi dilakukan dengan menggunakan 3 jenis dataset (Tabel 1). Jenis prediksi yang dilakukan adalah prediksi univariat menggunakan 1 variabel sebagai prediktor sekaligus target. GRNN yang dioptimalkan dengan modifikasi FOA (IAFOA) akan diuji coba untuk melakukan prediksi pada ketiga dataset yakni: kecepatan angin (*wind speed*) pada dataset 1; energi angin (*wind energy*) pada dataset 2; temperatur (*daily temp*), kelembaban udara (*humidity*), tekanan (*mean pressure*), dan kecepatan angin (*wind speed*) pada dataset 3.

Dataset 1 dan dataset 2 menggunakan setting *previous time-step* berturut-turut sama dengan 9 dan 10. Dataset 3 untuk prediksi *daily temp* menggunakan setting *previous time-step* sama dengan 2. Untuk prediksi *wind speed* dan *humidity* berturut-turut menggunakan setting *previous time-step* sama dengan 7 dan 8. Sedangkan pada *mean pressure* menggunakan setting *previous time-step* sama dengan 3. Hasil percobaan pada dataset 1 untuk memprediksi *wind speed* menggunakan GRNN dan IAFOA dapat dilihat pada Gambar 5 untuk performa *training* dan Gambar 6 untuk performa model pada saat *testing*.

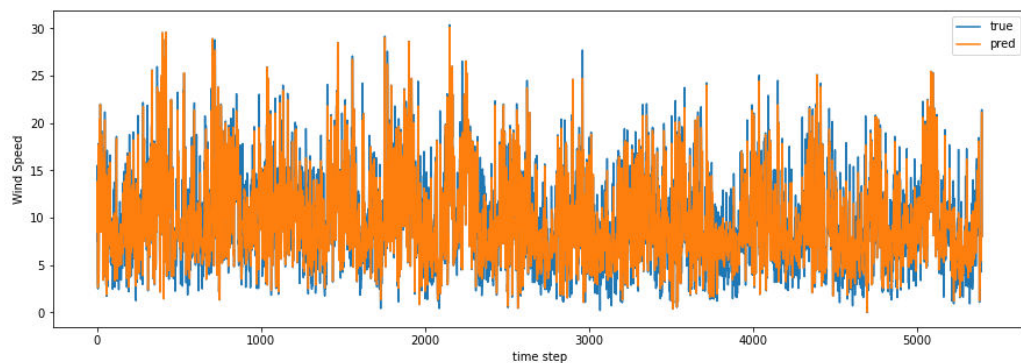
Performa FOA modifikasi (IAFOA) akan dibandingkan dengan FOA tanpa modifikasi. Ada 2 *evaluation metric* yang digunakan dalam evaluasi yaitu RMSE (*Root Mean Square Error*) dan MAE (*Mean Absolute Error*). Ketiga dataset disimulasikan menggunakan GRNN yang dioptimasi dengan IAFOA dan FOA. Hasil evaluasi dari perbandingan IAFOA dan FOA ditunjukkan pada Tabel 2.



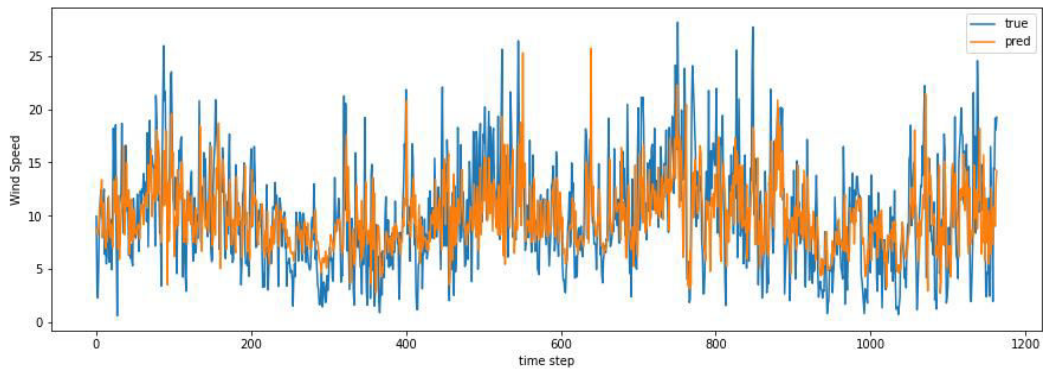
Gambar 4. *Framework* penelitian untuk model prediksi yang menggabungkan FOA Modifikasi dan GRNN

Tabel 1. Pembagian dataset untuk pelatihan dan pengujian model prediksi

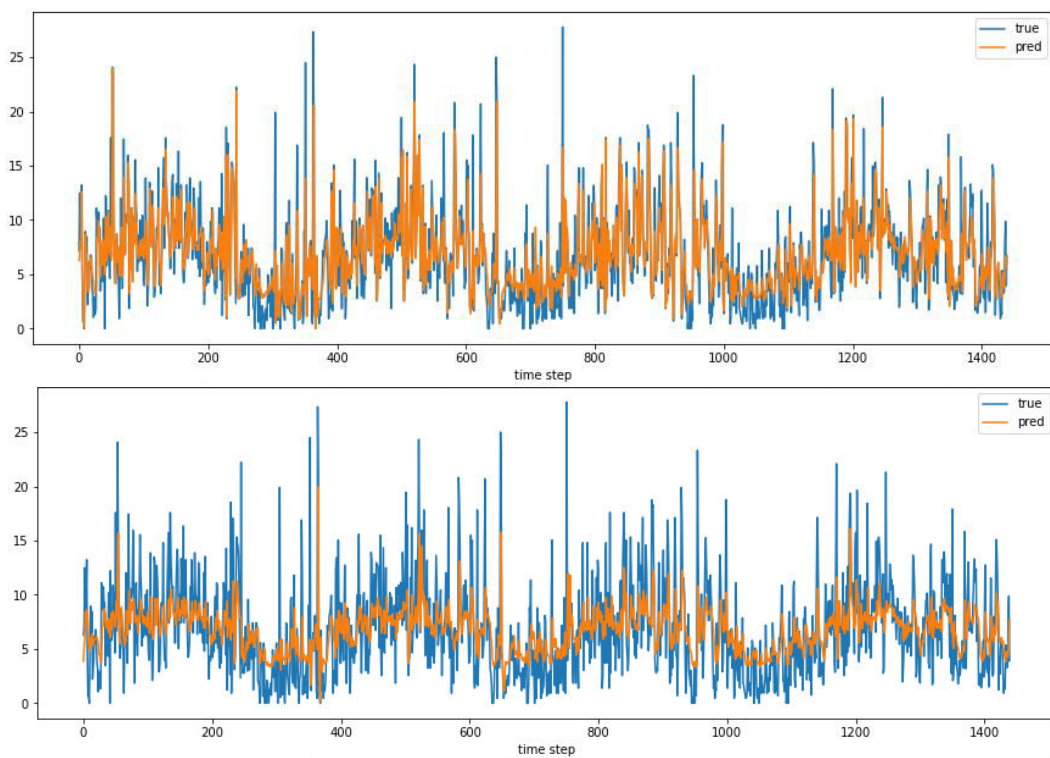
No	Dataset	Training	Testing
1	<i>Wind Speed Prediction Dataset</i>	15 tahun pertama (5400 baris)	2 tahun terakhir (1170 baris)
2	<i>Wind Energy in Germany</i>	4 tahun pertama (1440 baris)	4 bulan terakhir (134 baris)
3	<i>Daily Climate time series data</i>	2 tahun pertama (720 baris)	±1 tahun terakhir (373 baris)



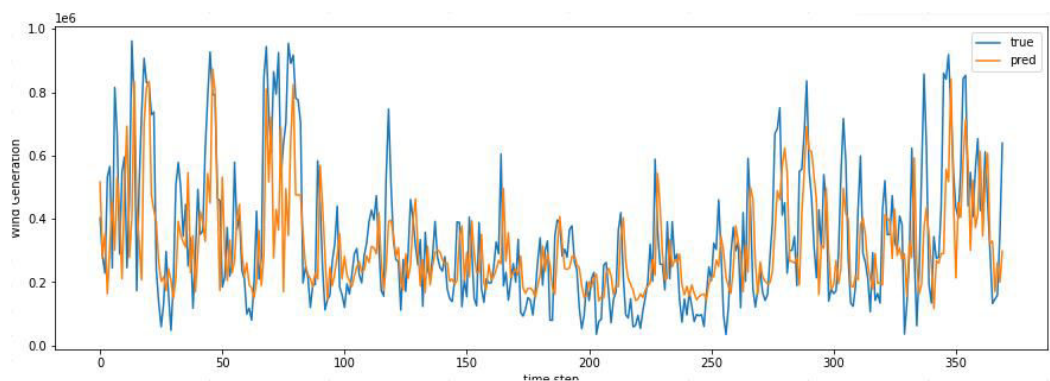
Gambar 5. Grafik perbandingan data prediksi dan aktual pada dataset *training* (dataset 1)

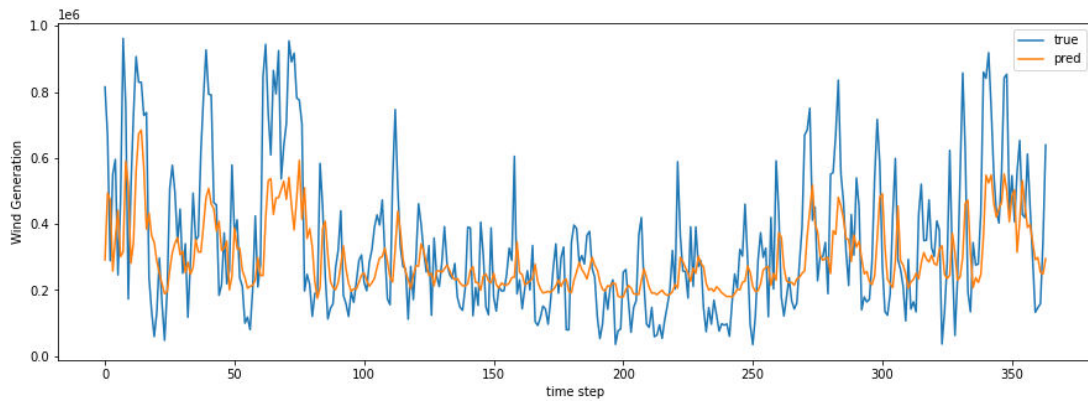


Gambar 6. Grafik perbandingan data prediksi dan aktual pada dataset *testing* (dataset 1)

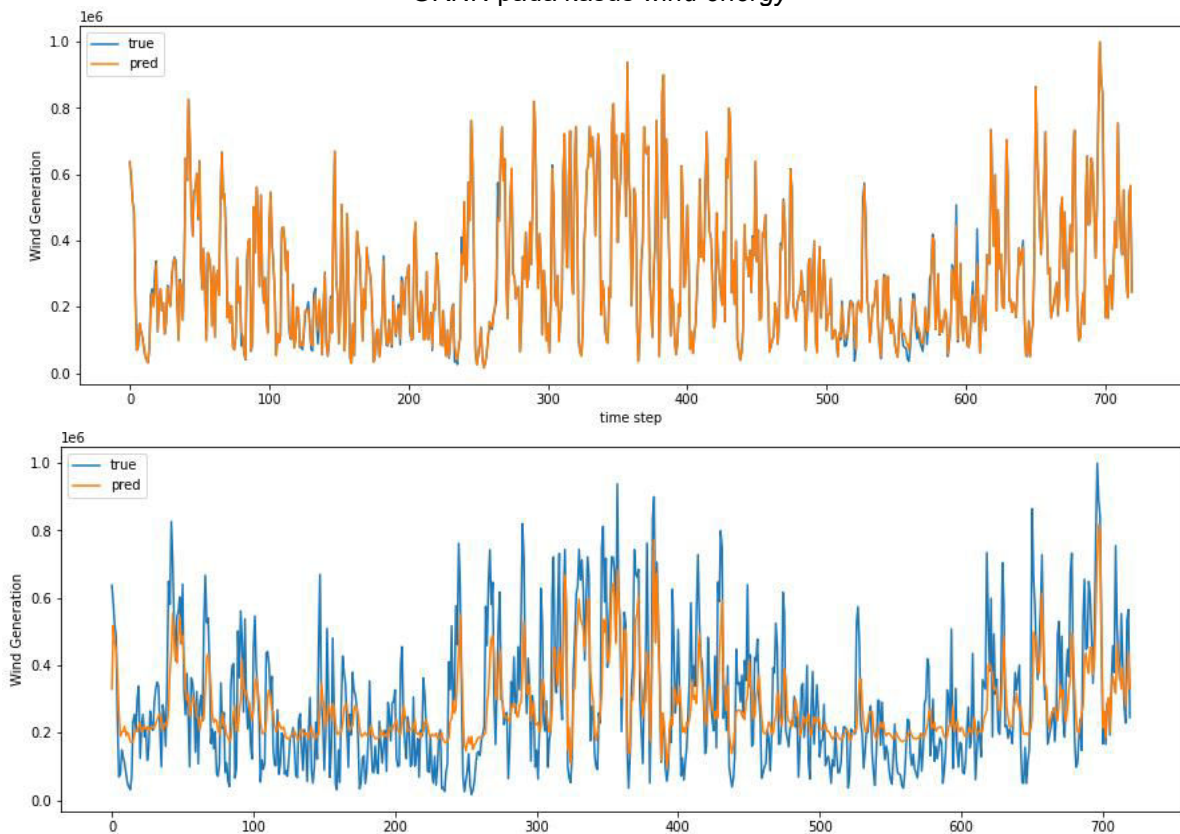


Gambar 7. Perbandingan performa model *training* IAFOA (atas) dan FOA (bawah) untuk optimasi GRNN pada kasus *wind speed* (dataset 3)

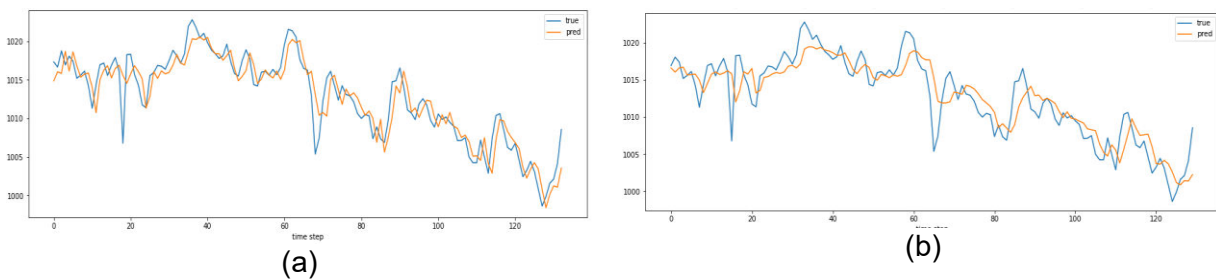


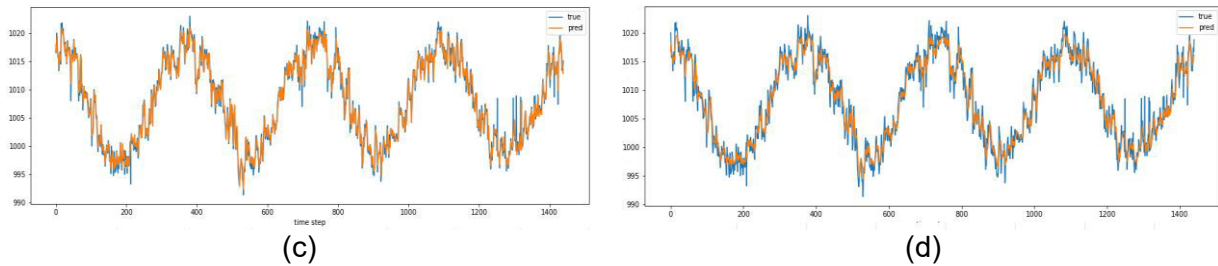


Gambar 8. Perbandingan performa model *testing* IAFOA (atas) dan FOA (bawah) untuk optimasi GRNN pada kasus *wind energy*

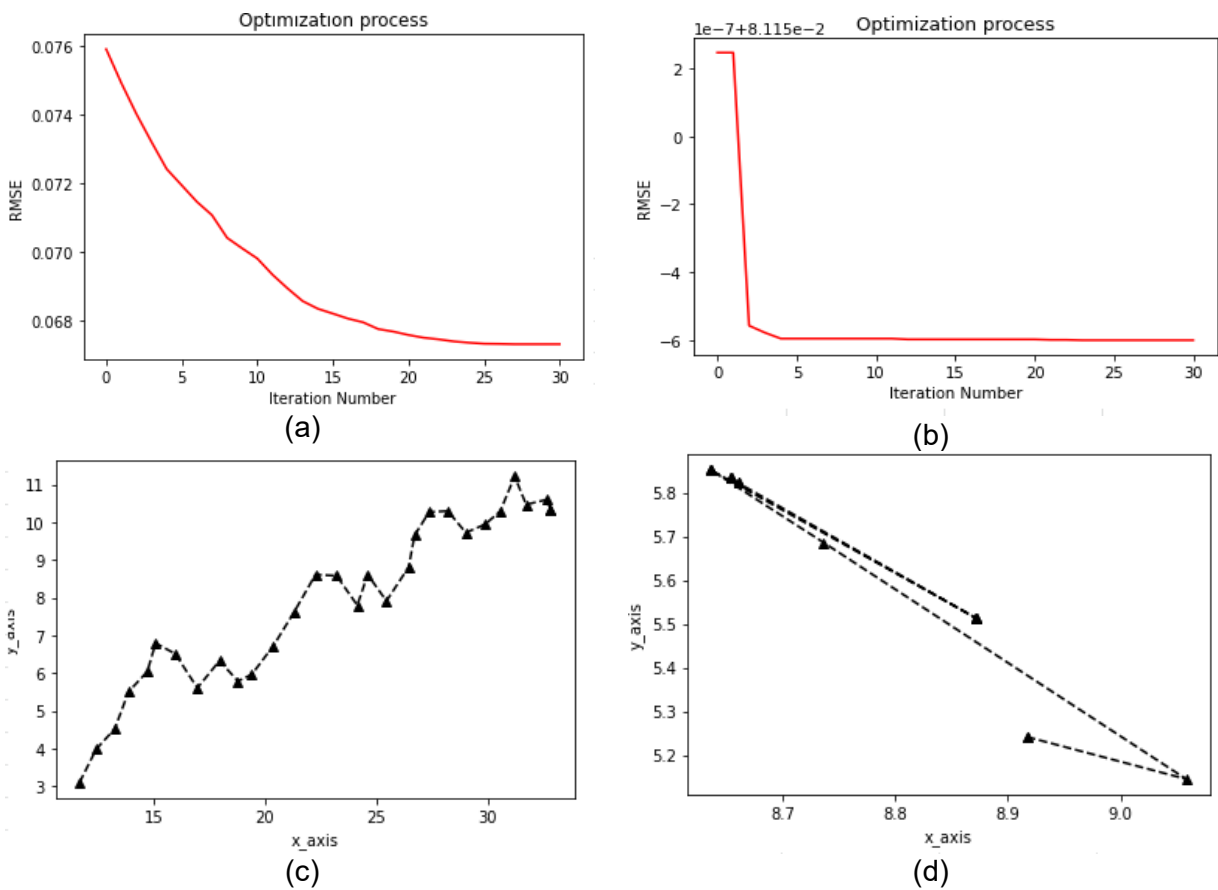


Gambar 9. Perbandingan performa model *training* IAFOA (atas) dan FOA (bawah) untuk optimasi GRNN pada kasus *wind energy*





Gambar 10. Perbandingan performa model *training* dan *testing* IAFOA (a dan c) dan FOA (b dan d) untuk optimasi GRNN pada kasus *mean pressure*



Gambar 11. Proses optimasi dan arah pergerakan alat buah IAFOA (a dan c) dan FOA (b dan d) pada kasus *mean pressure*

Tabel 2. Perbandingan performa IAFOA dan FOA pada prediksi *time-series*

DATASET	IAFOA-TEST		FOA-TEST		IAFOA-TRAIN		FOA-TRAIN	
	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE
1. Wind Speed Prediction Dataset	4.2982	3.4300	4.0585	3.2017	1.8043	1.1663	3.7242	2.9425
2. Wind Energy in Germany	176726.07	133481.97	178945.85	134822.53	11273.21	3521.98	131052.60	100732.04

3. Daily Climate
Time Series Data

Daily Temp	1.7116	1.3210	1.6882	1.3047	1.6885	1.3104	1.5905	1.2244
Humidity	7.7163	5.7329	7.9401	5.8585	5.6578	4.1622	6.3620	4.8058
Wind Speed	3.3505	2.6660	3.4022	2.6924	2.0624	1.3223	3.5259	2.6030
Mean Pressure	2.1282	1.6441	2.5664	1.9384	1.4114	1.0313	1.7947	1.3824
Average	35348.63	26699.02	35792.59	26967.12	2256.88	705.99	26213.56	20148.72

Berdasarkan hasil evaluasi performa model prediksi yg ditunjukkan pada Tabel 2, pada 4 kasus prediksi dapat diketahui bahwa modifikasi FOA (IAFOA) mampu meningkatkan performa prediksi dan menekan error lebih kecil dibandingkan FOA tanpa modifikasi. Meskipun, pada satu kasus (prediksi *daily temp*) FOA terbukti lebih unggul dengan selisih yang tidak begitu signifikan dibandingkan dengan IAFOA. Pada ujicoba pada kasus prediksi *Daily Temp*, diketahui bahwa FOA unggul dalam proses *training* dan *testing* model dengan selisih yang tidak begitu signifikan yakni < 0.5 . Selisih error prediksi model pada saat *training* berturut-turut untuk RMSE dan MAE mencapai 0.0234 dan 0.0163, sedangkan pada proses testing selisih keduanya berkisar 0.098 untuk RMSE dan MAE 0.086.

Pada kasus prediksi *wind speed prediction* (dataset 1) IAFOA lebih unggul pada proses *training* dibandingkan pada saat *testing*. Pada kasus ini, model prediksi FOA lebih unggul pada proses *testing* model namun selisih error prediksi tidak begitu signifikan berbeda dibandingkan IAFOA dengan selisih RMSE dan MAE ± 0.2 . IAFOA secara signifikan mengungguli FOA pada kasus prediksi *wind energy* (dataset 2), terlihat selisih error yang cukup jauh terutama pada performa model pada saat *training*.

Lebih lanjut dapat dilakukan pengamatan terhadap grafik perbandingan data prediksi dan aktual untuk IAFOA dan FOA. Terlihat bagaimana performa GRNN yg dioptimasi IAFOA dan FOA pada dataset uji ditunjukkan pada Gambar 7 – 10. IAFOA mampu memberikan nilai prediksi yang hampir mendekati nilai aktual pada kasus *wind speed* dan *mean pressure* pada dataset 3, begitu juga pada kasus *wind energy* pada dataset 2. Sedangkan performa FOA pada GRNN tidak begitu memperlihatkan performa yang cukup baik, terlihat dari nilai prediksi yang masih menjauhi nilai aktual. Pergerakan alat buah pada proses optimasi IAFOA bergerak pada arah yang teratur ditunjukkan pada Gambar 11, hal ini diikuti dengan penurunan nilai RMSE

yang perlahan hingga iterasi (*maxgen*) mendekati 30. Sedangkan pada FOA, pergerakan alat buah berbeda dengan yang terjadi pada IAFOA, proses optimasi terhenti setelah iterasi (*maxgen*) ke-5, artinya ruang pencarian solusi belum bisa diperluas lagi.

KESIMPULAN

Penelitian untuk menghasilkan algoritma optimasi FOA yang dimodifikasi dengan *immune algorithm* (IAFOA) telah dilakukan dan memperoleh hasil. IAFOA telah diuji bersama GRNN pada beberapa dataset untuk melihat performanya dalam melakukan prediksi *time-series*. Selain itu, untuk memastikan bahwa ada perubahan peningkatan performa dari FOA, IAFOA telah dibandingkan dengan FOA pada kasus dan dataset yang sama. Hasil evaluasi menunjukkan bahwa IAFOA mengungguli FOA. Dari ke-6 kasus yang disimulasikan, IAFOA unggul dalam *training* dan *testing* untuk 4 kasus, sedangkan 1 kasus lainnya unggul pada saat *training*. Terlihat fenomena bahwa FOA unggul pada 1 kasus untuk training dan testing, 1 kasus lagi pada saat testing. Selisih error dari model prediksi yang dioptimasi dari kedua algoritma ini tidak begitu jauh signifikan, kecuali pada kasus *wind energy* (dataset 2).

Masih terdapat ruang perbaikan untuk penyempurnaan IAFOA dan perlu ujicoba pada kasus lainnya. Berdasarkan ujicoba pada beberapa dataset, diketahui terjadi kemungkinan *overfitting* pada dataset 2. Maka dari itu, peningkatan kinerja IAFOA disadari perlu dilakukan terutama pada penentuan nilai $T = 6$ digunakan sebagai nilai ambang batas untuk memasuki fase *immune algorithm* pada FOA. Ujicoba optimasi IAFOA pada algoritma lain dan pada studi kasus (dataset) lain juga diperlukan untuk meyakinkan bahwa definisi IAFOA memberikan perubahan signifikan dalam kinerja optimasi.

UCAPAN TERIMAKASIH

Penulis mengucapkan terima kasih kepada LPPM Universitas Pendidikan Ganesha yang telah memberikan kesempatan dan

pendanaan kepada penulis untuk berpartisipasi dalam kegiatan penelitian dosen pemua dengan kontrak penelitian nomor: 646/UN48.16/LT/2022.

REFERENSI

- [1] X. S. Yang, "Swarm intelligence based algorithms: A critical analysis," *Evol. Intell.*, vol. 7, no. 1, pp. 17–28, 2014, doi: 10.1007/s12065-013-0102-2.
- [2] X. Guo, J. Zhang, W. Li, and Y. Zhang, "A fruit fly optimization algorithm with a traction mechanism and its applications," *Int. J. Distrib. Sens. Networks*, vol. 13, no. 11, 2017, doi: 10.1177/1550147717739831.
- [3] W. Pan, "Knowledge-Based Systems A new Fruit Fly Optimization Algorithm: Taking the financial distress model as an example," *Knowledge-Based Syst.*, vol. 26, pp. 69–74, 2012, doi: 10.1016/j.knsys.2011.07.001.
- [4] Y. Li and M. Han, "Improved fruit fly algorithm on structural optimization," *Brain Informatics*, vol. 7, no. 1, pp. 1–13, 2020, doi: 10.1186/s40708-020-0102-9.
- [5] Y. Liu, J. J. Sun, and X. Wang, "Research on Multi-Distribution Center Location Based on Fruit Fly – Immune Algorithm," *Appl. Mech. Mater.*, vol. 743, pp. 338–342, 2015, doi: 10.4028/www.scientific.net/amm.743.338.
- [6] D. Chen, S. Li, J. Wang, Y. Feng, and Y. Liu, "A multi-objective trajectory planning method based on the improved immune clonal selection algorithm," *Robot. Comput. Integr. Manuf.*, vol. 59, pp. 431–442, 2019, doi: <https://doi.org/10.1016/j.rcim.2019.04.016>.
- [7] Z. Yang, Y. Ding, K. Hao, and X. Cai, "An adaptive immune algorithm for service-oriented agricultural Internet of Things," *Neurocomputing*, vol. 344, pp. 3–12, 2019, doi: <https://doi.org/10.1016/j.neucom.2018.06.094>.
- [8] J. Song and H. Pan, "PID control parameters optimize based on an immune fruit fly optimization algorithm," *Proc. 28th Chinese Control Decis. Conf. CCDC 2016*, pp. 6383–6388, 2016, doi: 10.1109/CCDC.2016.7532147.
- [9] B. B. Savarala and P. R. Chella, "An improved fruit fly optimization algorithm for QoS aware cloud service composition," *Int. J. Intell. Eng. Syst.*, vol. 10, no. 5, pp. 105–114, 2017, doi: 10.22266/ijies2017.1031.12.
- [10] D. Corus, P. S. Oliveto, and D. Yazdani, "When hypermutations and ageing enable artificial immune systems to outperform evolutionary algorithms," *Theor. Comput. Sci.*, vol. 832, pp. 166–185, 2020, doi: <https://doi.org/10.1016/j.tcs.2019.03.002>.
- [11] M. T. Leung, A. Chen, and H. Daouk, "Forecasting Exchange Rates Using General Regression Neural networks," *Comput. Oper. Res.*, vol. 27, no. 11–12, pp. 1093–1110, 2000, doi: [https://doi.org/10.1016/S0305-0548\(99\)00144-6](https://doi.org/10.1016/S0305-0548(99)00144-6).
- [12] Adnyani and Subanar, "General Regression Nerve Network (GRNN) Forecasting Dollar Exchange Rate and Composite Stock Price Index (CSPI)," *Factor Exacta*, vol. 8, pp. 137–144, 2015.
- [13] R. E. Caraka, H. Yasin, and A. Prahutama, "Modeling of General Regression Neural Network (GRNN) on Data Return of Euro 50 Stock Price Index," *Gaussian*, vol. 4, pp. 181–192, 2015.
- [14] D. Niu, H. Wang, H. Chen, and Y. Liang, "The General Regression Neural Network Based on the Fruit Fly Optimization Algorithm and the Data Inconsistency Rate for Transmission Line Icing Prediction," *Energies*, vol. 10, no. 2066, 2017, doi: 10.3390/en10122066.
- [15] N. P. N. P. Dewi and R. A. Nugroho, "Optimasi General Regression Neural Network Dengan Fruit Fly Optimization Algorithm Untuk Prediksi Pemakaian Arus Listrik Pada Penyulang," *KOMPUTASI J. Ilm. Ilmu Komput. dan Mat.*, vol. 18, no. 1, pp. 1–12, 2021, doi: <https://doi.org/10.33751/komputasi.v18i1.2144>.
- [16] D. Burba, "An overview of time series forecasting models," 2019, [Online]. Available: <https://towardsdatascience.com/an-overview-of-time-series-forecasting-models-a2fa7a358fcb>.
- [17] P. S. P. Cowperrwait and A. V. Metcalfe, *Introduction Time Series with R*. Springer, 2009.
- [18] J. Brownlee, "What Is Time Series Forecasting?," 2020. <https://machinelearningmastery.com/time-series-forecasting/>.
- [19] Tableau, "Time Series Forecasting: Definition, Applications, and Examples," 2022. <https://www.tableau.com/learn/articles/ti>

- me-series-forecasting#:~:text=Time series forecasting occurs when,drive future strategic decision-making.
- [20] H. Bersini and F. J. Varela, "Hints for adaptive problem solving gleaned from immune networks," in *Parallel Problem Solving from Nature*, H.-P. Schwefel and R. Männer, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 1991, pp. 343–354.
- [21] A. Babalik, H. İşcan, İ. Babaoğlu, and M. Gündüz, "An improvement in fruit fly optimization algorithm by using sign parameters," *Soft Comput.*, vol. 22, no. 22, pp. 7587–7603, 2018, doi: 10.1007/s00500-017-2733-1.
- [22] T.-S. Du, X.-T. Ke, J.-G. Liao, and Y.-J. Shen, "DSLCO-FOA: Improved fruit fly optimization algorithm for application to structural engineering design optimization problems," *Appl. Math. Model.*, vol. 55, pp. 314–339, 2018, doi: <https://doi.org/10.1016/j.apm.2017.08.013>.
- [23] G. Azamirad and B. Arezoo, "Structural design of stamping die components using bi-directional evolutionary structural optimization method," *Int. J. Adv. Manuf. Technol.*, vol. 87, no. 1, pp. 969–979, 2016, doi: 10.1007/s00170-016-8344-7.
- [24] D. F. Specht, "General Regression Neural Network (GRNN)," in *General Regression Neural Network (GRNN)*, 1991, pp. 42–60.
- [25] B. Mohanty and P. Hota, "Comparative performance analysis of fruit fly optimization algorithm for multi-area multi-source automatic generation control under deregulated environment," *Gener Trans Distrib*, vol. 9, no. 14, pp. 1845–1855, 2015, doi: <https://doi.org/10.1049/iet-gtd.2015.0284>.
- [26] Fedesoriano, "Wind Speed Prediction Dataset," 2022. <https://www.kaggle.com/datasets/fedesoriano/wind-speed-prediction-dataset>.
- [27] Q. Al-Btoush, "Wind Energy in Germany," 2022. <https://www.kaggle.com/code/qusaybtoush1990/wind-energy-in-germany/data>.
- [28] Sumanthvrao, "Daily Climate time series data," 2022. <https://www.kaggle.com/datasets/sumanthvrao/daily-climate-time-series-data>.