



Model Artificial Neural Networks (ANN) untuk Prediksi COVID-19 di Indonesia

Alfi Bella Kurniati¹, Wuryatmo A. Sidik², Jajang^{3*} 

^{1,2,3}Jurusan Matematika, Universitas Jenderal Soedirman, Purwokerto, Indonesia

ARTICLE INFO

Article history:

Received January 28, 2022

Accepted August 13, 2023

Available online October 25, 2023

Kata Kunci:

Artificial Neural Network, Backpropagation, Covid-19, Particle Swarm Optimization

Keywords:

Artificial Neural Network, Backpropagation, Covid-19, Particle Swarm Optimization



This is an open access article under the [CC BY-SA](https://creativecommons.org/licenses/by-sa/4.0/) license.

Copyright © 2023 by Author. Published by Universitas Pendidikan Ganesha.

ABSTRAK

Peningkatan jumlah kasus Covid-19 hampir terjadi di seluruh dunia penting dikaji karena pola persebarannya yang masif. Metode analisis yang digunakan adalah ANN. Penelitian ini mencakup dua hal, yaitu model ANN algoritma *Backpropagation Neural Network* (BPNN) dan kombinasi algoritma *Backpropagation Neural Network* dan *Particle Swarm Optimization* (BPNN-PSO). Masing-masing model ANN algoritma BPNN dan BPNN-PSO dilakukan 18 kali pengujian. Hasil pengujian menunjukkan kombinasi parameter terbaik ANN dengan algoritma BPNN diperoleh MAPE sebesar 16.29% dengan arsitektur 14-5-1 dengan *learning rate* 0.3. Kombinasi parameter ANN terbaik algoritma BPNN-PSO MAPE sebesar 15.45% pada arsitektur 14-10-1 dengan *learning rate* 0,2. Berdasarkan algoritma BPNN dan BPNN-PSO, model ANN terbaik adalah model ANN dengan algoritma BPNN-PSO. Dengan kata lain, nilai MAPE yang cenderung lebih besar dibandingkan MAPE dari hasil algoritma BPNN-PSO. Hal ini mengindikasikan bahwa kombinasi BPNN-PSO berhasil untuk meningkatkan akurasi hasil prediksi. Implementasi algoritma Particle Swarm Optimization (PSO) yang berperan sebagai penentu nilai bobot dan bias awal dalam proses pelatihan metode *Backpropagation Neural Network* (BPNN) dapat meningkatkan performa model ANN untuk memprediksi jumlah kumulatif kasus positif Covid-19 di Indonesia, dengan peningkatan akurasi yang signifikan. Sehingga penggunaan model ANN dengan algoritma BPNN-PSO dapat diterapkan dalam situasi ini untuk memprediksi jumlah kumulatif kasus positif Covid-19 di Indonesia.

ABSTRACT

The increase in the number of Covid-19 cases occurring almost all over the world is important to study because of its massive distribution pattern. The analysis method used is ANN. This research covers two things, namely the ANN model with the *Backpropagation Neural Network* (BPNN) algorithm and the combination of the BPNN algorithm and *Particle Swarm Optimization* (BPNN-PSO). Each ANN model with the BPNN and BPNN-PSO algorithms was tested 18 times. The test results show that the best combination of ANN parameters with the BPNN algorithm obtained a MAPE of 16.29% with a 14-5-1 architecture with a learning rate of 0.3. The best ANN parameter combination for the BPNN-PSO MAPE algorithm is 15.45% on a 14-10-1 architecture with a learning rate of 0.2. Based on the BPNN and BPNN-PSO algorithms, the best ANN model is the ANN model with the BPNN-PSO algorithm. In other words, the MAPE value tends to be greater than the MAPE from the results of the BPNN-PSO algorithm. This indicates that the BPNN-PSO combination is successful in increasing the accuracy of prediction results. Implementation of the PSO algorithm which acts as a determinant of initial weight and bias values in the BPNN method training process can improve the performance of the ANN model for predicting the cumulative number of positive Covid-19 cases in Indonesia, with a significant increase in accuracy. So, the use of the ANN model with the BPNN-PSO algorithm can be applied in this situation to predict the cumulative number of positive Covid-19 cases in Indonesia.

1. PENDAHULUAN

Pada bulan Desember 2019 di Wuhan, Provinsi Hubei, China ditemukan kemunculan penyakit baru oleh badan kesehatan dunia (*World Health Organization*, WHO) yang dinamakan dengan *Coronavirus Disease* (Covid-19). Covid-19 merupakan sebuah penyakit yang disebabkan oleh *Severe Acute Respiratory Syndrome Coronavirus 2* (SARS-CoV-2). Covid-19 memiliki tingkat penularan yang relatif cepat dan ditetapkan sebagai pandemi oleh WHO pada 11 Maret 2020 karena sudah menyebar secara luas di China dan lebih dari 190 negara. Masa inkubasi Covid-19 waktu antara terjadinya infeksi dan lama gejala yang ditimbulkan 1-14 hari (Amalia et al., 2020; Susilo et al., 2020).

Indonesia menjadi salah satu negara yang terdampak oleh penyebaran Covid-19. Peningkatan kasus Covid-19 harian di berbagai wilayah di Indonesia perlu diidentifikasi, dikarakterisasi dan pemodelan. Beberapa penelitian yang terkait dengan pemodelan telah dilakukan melalui pendekatan vector

*Corresponding author.

E-mail addresses: jajang@unsoed.ac.id (Jajang)

autoregressive, simulasi numerik model COVID-19 di Wuhan, China dengan model compartment (Meimela et al., 2021; Ndaïrou et al., 2020). Hasil penelitian dengan metode kombinasi kecerdasan buatan dan model kompartemen klasik, dan hasilnya memberikan prediksi penyebaran penyakit yang lebih baik dan masih banyak lagi. Sementara itu beberapa peneliti lain menggunakan *Artificial Neural Networks* (ANN) untuk prediksi jumlah kasus Covid-19 (Ahmad & Asad, 2020; Cahyaningsih, 2020; Fernandes et al., 2021; Ghanim et al., 2022; Hamadneh et al., 2021; Jena et al., 2021; Mollalo et al., 2020; Niazkar & Niazkar, 2020; Ojokoh et al., 2022; Torrealba-Rodriguez et al., 2020). Para peneliti tersebut menggunakan model ANN standar. Disisi lain, model ANN juga digunakan untuk prediksi pada data-data runtun waktu (*time series*) (Tealab, 2018; Wang et al., 2018).

Beberapa peneliti juga telah menggunakan kombinasi ANN dengan metode lain, seperti *Artificial Neural Network-iIntegrated by Grey Wolf Optimizer* (ANN-GWO), ANN dengan *Genetic Algorithm*, dan model ANN metode *Backpropagation Neural Network* (BPNN) dengan Fuzzy Tsukamoto (Ardabili et al., 2020; Arianto & Noviyanti, 2020; Mohamad et al., 2017). Pada algoritma BPNN terdapat tahapan inialisasi bobot dan bias awal biasanya ditentukan secara *random*, sementara pemilihan bobot dan bias awal sangat mempengaruhi performa ANN. Untuk meningkatkan kinerja BPNN dapat dipilih penentuan inisiasi bobot awal, diantaranya dengan metode optimasi seperti *Particle Swarm Optimization* (PSO) dan *Genetic Algorithm* (GA). Berdasarkan hasil penelitian, nilai MAPE dengan BPNN-PSO sebesar 0,11% sedangkan dengan GA-BPNN sebesar 1,13%. Penelitian diperoleh nilai akurasi model BPNN-PSO mencapai 91,5% yang secara signifikan lebih tinggi dibandingkan dengan model BPNN standar yaitu sebesar 45,2% (Huang & Wu, 2017; Liu, 2021; Suhendra & Wardoyo, 2015).

Penelitian terkait dengan Covid-19 yang melibatkan model ANN dan kajian hotspot wilayah-wilayah terdampak juga telah dikaji untuk memberikan wawasan yang berguna bagi pembuat keputusan kesehatan masyarakat mengenai pengaruh faktor risiko potensial yang terkait dengan kejadian COVID-19 (Guo, X. et al., 2020; Mollalo et al., 2020; Verianto & Oetomo, 2020). Adapun kajian yang terkait dengan metode optimasi PSO adalah dalam penentuan optimalisasi *Travelling Salesman Problem* (TSP) dan dalam algoritma BPNN pada prediksi nilai rupiah terhadap dolar.

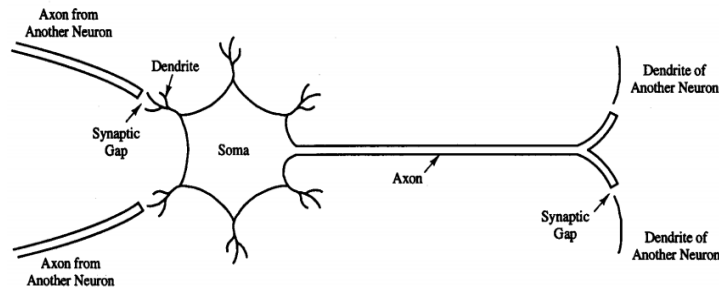
Urgensi dari penelitian ini adalah bahwa pola persebaran penyakit seperti Covid memerlukan penanganan cepat dengan mendeteksi dini pola persebaran datanya. Pola persebaran data dapat dikenali salah satunya dengan model statistika yang relevan. Pola sebaran data dan sifat data yang dependen waktu yang berbeda, perlu penanganan model yang pas. Disini yang peneliti gunakan adalah model ANN. Dalam model ANN pada kasus data ini, diasumsikan bahwa jumlah kasus saat waktu tertentu dipengaruhi oleh waktu-waktu sebelumnya. Waktu-waktu sebelumnya inilah yang dijadikan sebagai variabel bebas. Dari sisi perkembangan metode ANN, penelitian ini dimotivasi oleh beberapa hal yang telah dijelaskan sebelumnya, dimana algoritma BPNN dikombinasikan dengan algoritma PSO. Penelitian ini bertujuan untuk membangun model ANN metode BPNN-PSO dengan jumlah unit *hidden* dan *learning rate* yang optimal sehingga diharapkan dapat hasil prediksi yang baik.

2. METODE

Data penelitian ini menggunakan data sekunder yang diperoleh dalam rentang waktu 2 Maret 2020 –31 Agustus 2021. Dataset yang telah dikumpulkan yaitu sebanyak 548 data harian terkonfirmasi kasus positif Covid-19 di Indonesia. Desain penelitian yang digunakan merupakan penelitian kuantitatif, secara spesifiknya adalah ANN yang termasuk ke dalam *soft modeling* (Ghanim et al., 2022; Yan & Au, 2019). Tahapan penelitian mencakup beberapa tahap, (1) rancangan variable penelitian, yakni mendesain variable bebas dan variable terikat. Jumlah kasus 14 hari sebelumnya dijadikan sebagai variable bebas untuk hasil-h (sekarang) yang dijadikan sebagai variable tak bebas. Disini mengambil asumsi bahwa masa inkubasinya adalah 14 hari. Setelah membentuk variabel bebas dan variabel tak bebas, (2) melakukan kajian model ANN dan *Particle Swarm Optimization* (PSO), (3) Normalisasi dan transformasi, (4) aplikasi model ANN BPNN standar dan kombinasi ANN BPNN +PSO pada data Covid-19, dan (5) mengukur kebaikan model dengan kriteria *Mean Absolute Percentage Error* (MAPE). Berikut adalah uraian dengan model ANN metode BPNN, metode PSO, transformasi dan evaluasi model melalui MAPE.

Artificial Neural Network (ANN)

Artificial Neural Network (ANN) atau Jaringan Saraf Tiruan (JST) adalah sebuah sistem pemrosesan informasi yang arsitekturnya dimodelkan dengan meniru cara kerja jaringan saraf manusia (Kusmaryanto, 2014; Solikhun et al., 2017). Jaringan ini dapat menyimpan, menggunakan, dan memperoleh pengetahuan yang di dapat dari proses pembelajaran. Gambar 1 menunjukkan sebuah jaringan saraf biologis manusia yang terdiri atas 3 bagian dasar, yaitu *dendrit*, *soma* dan *akson*.



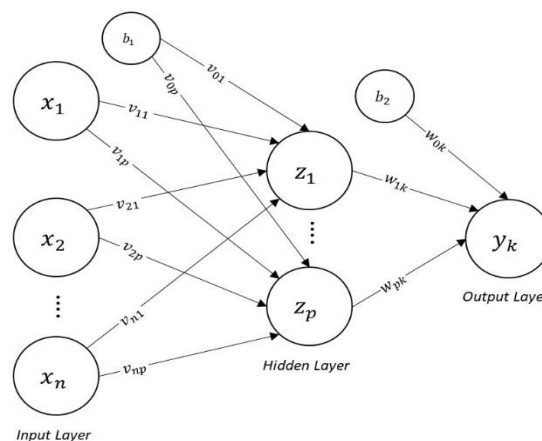
Gambar 1. Jaringan Saraf Biologis

Dendrit berfungsi sebagai penerima informasi dari *neuron* lain dan dibawa ke *soma*. *Soma* adalah bagian utama untuk menjumlahkan informasi yang diterima dan akan dikirimkan ke *neuron* lain yang disebut dengan *akson*. Informasi hasil pemrosesan akan dikirimkan pada *neuron* lain dimana antar *neuron* dihubungkan dengan sel sinapsis. Penyusun ANN terdiri dari lapisan *input* (*Input layer*), lapisan *hidden*, dan lapisan *output*. Lapisan *input* merupakan kumpulan dari *neuron-neuron input* yang berfungsi sebagai masukan atau *input* bagi ANN. Lapisan *hidden* merupakan kumpulan dari *neuron-neuron hidden* atau tersembunyi dimana nilai *output* yang dihasilkan tidak dapat diamati secara langsung. Lapisan *output* merupakan kumpulan dari *neuron-neuron output* yang menjadi keluaran ANN (Meutia & Himayani, 2021; Pradana et al., 2022).

Berdasarkan karakteristiknya, ANN terdiri dari arsitektur jaringan, metode pembelajaran dan fungsi aktivasi. Arsitektur ANN terdiri dari Jaringan Lapisan Tunggal (*Single Layer Network*), Jaringan Lapisan Jamak (*Multilayer Network*), dan Jaringan Kompetitif. Jaringan lapisan tunggal hanya terdapat *input layer* dan *output layer*. Jaringan lapisan jamak memiliki 3 *layer*, yaitu *input layer*, *hidden layer*, dan *output layer*. Jaringan kompetitif memiliki hubungan umpan balik atau *feedback loop* yaitu ada *neuron output* yang memberikan sinyal pada *neuron input*. Berdasarkan metode pembelajarannya, ANN terbagi menjadi 2 metode yaitu *supervised learning* dan *unsupervised learning*. Dalam metode *supervised learning* terdapat *input* dan target *output* yang sudah diketahui sebelumnya. *Unsupervised Learning* yaitu metode pembelajaran yang menggunakan sekumpulan data yang tidak memerlukan target *output* nya.

Backpropagation Neural Network (BPNN)

Backpropagation Neural Network (BPNN) termasuk dalam metode pembelajaran *supervised learning*. BPNN memiliki 3 *layer*, yaitu *input layer*, *hidden layer*, dan *output layer*. Gambar 2 merupakan gambar BPNN dengan n *neuron input layer* disimbolkan dengan x_i ($i = 1,2,3, \dots, n$), p *neuron hidden layer* yang disimbolkan dengan z_j ($j = 1,2,3, \dots, p$), dan m *neuron output layer* yang disimbolkan dengan y_k ($k = 1,2,3, \dots, m$). Setiap *neuron* pada *layer* yang berlainan dihubungkan dengan sebuah pembobot.



Gambar 2. Arsitektur *Backpropagation Neural Network*

Bobot penghubung antara *neuron* pada *input layer* dengan *neuron* pada *hidden layer* disimbolkan dengan v , dan bobot antara *neuron* pada *hidden layer* dengan *neuron* pada *output layer* disimbolkan dengan w . Fungsi aktivasi pada BPNN memiliki syarat yaitu kontinu, terdiferensiasi, dan monoton. Penentuan jumlah unit *neuron* pada *hidden layer* merupakan aspek yang paling menantang dalam ANN. Untuk

menentukan unit *neuron* pada *hidden layer* dengan menggunakan informasi nilai eigen, selanjutnya ada yang menggunakan Heurix. Namun pada berbagai terapannya, proses penentuan jumlah *hidden layer* dan jumlah unit *neuron* pada setiap *hidden layer* masih membingungkan (Cai et al., 2019; Dwiyanto et al., 2019; Thomas et al., 2015). Proses estimasi pembobot melalui metode pembelajaran BPNN baik untuk prediksi atau klasifikasi dilakukan melalui dua proses, yaitu proses pelatihan (*training*) dan proses pengujian (*testing*). Proses pelatihan terdiri dari tiga tahap, yaitu yaitu propagasi maju (*feedforward*), propagasi mundur (*backpropagation*), dan perubahan bobot. Sedangkan pada proses pengujian hanya menggunakan tahap propagasi maju (*feedforward*).

Tahap Propagasi Maju

Pada tahap propagasi maju setiap *neuron* pada *input layer* ($x_i, i = 1,2,3, \dots, n$) menerima sinyal x_i dan meneruskannya ke semua *neuron* yang ada pada *hidden layer* melalui bobot penghubungnya v_{ij} (bobot penghubung *neuron input* ke- i dengan *neuron hidden* ke- j), $i = 1,2, \dots, n, j = 1,2,3, \dots, p$. Setelah sinyal diterima, berikutnya setiap *neuron* pada *hidden layer* ($z_j, j = 1,2,3, \dots, p$) menjumlahkan sinyal-sinyal *input* yang sudah berboboti melalui persamaan (1).

$$z_{net_j} = v_{0j} + \sum_{i=1}^n v_{ij}x_i, j = 1,2, \dots, p \tag{1}$$

Dengan v_{0j} adalah bias pada *neuron hidden* ke- j . Setelah mendapatkan nilai z_{net_j} berikutnya adalah mengaktifasi z_{net_j} dengan fungsi aktivasi untuk mendapatkan z_j melalui persamaan (2).

$$z_j = f(z_{net_j}) \tag{2}$$

Tahap berikutnya setelah mendapatkan z_j adalah mengirimkannya ke semua *neuron* pada *output layer* melalui bobot penghubungnya yaitu w_{jk} . Bobot w_{jk} adalah bobot penghubung *neuron hidden* ke- j dengan *neuron output* ke- $k, k = 1,2,3, \dots, m$ melalui persamaan (3).

$$y_{net_k} = w_{0k} + \sum_{j=1}^p w_{jk}z_j \tag{3}$$

Dengan w_{0k} adalah bias pada *neuron output* ke- $k, k = 1,2, \dots, m$. Setelah mendapatkan y_{net_k} , berikutnya adalah mengaktifasikannya dengan fungsi aktivasi sehingga akan menghasilkan sinyal *output* melalui $y_k = f(y_{net_k})$.

Tahap Propagasi Mundur

Pada tahap propagasi mundur, setiap pembobot yang menghubungkan antar neuran pada *layer yang berbeda* diperbaharui. Proses perubahan bobot pada propagasi mundur merupakan proses untuk memperbarui bobot yang ada dengan menggunakan *delta rule*. Perubahan pembobot terdiri dari perubahan bobot yang menghubungkan antar *neuron* pada *layer hidden* dan *output* dan antar *neuron* pada *layer input* dan *hidden*. Perbaharuan pembobot *delta rule* antar *neuron* antara *layer hidden* dan *output* adalah (Cynthia & Ismanto, 2017; Rahakbauw, 2014):

$$\Delta w_{jk} = \alpha \delta_k z_j \text{ dan } \Delta w_{0k} = \alpha \delta_k, \tag{4}$$

Dengan $\delta_k = (t_k - y_k) f'(y_{net_k})$, $f'(\cdot)$ adalah turunan pertama fungsi aktivasi pada *neuron output*, t_k dan y_k berturut-turut adalah target dan *output* k, dan y_{net_k} diperoleh dari persamaan (3). Sedangkan *delta rule* untuk memperbaharui pembobot antar *neuron* antara *layer input* dan *hidden* dilakukan melalui cara.

$$\Delta v_{ij} = \alpha \delta_j x_i \text{ dan } \Delta v_{0j} = \alpha \delta_j, \tag{5}$$

Dengan $\delta_j = (\sum_k \delta_k w_{jk}) f'(z_{net_j})$, $f'(\cdot)$ adalah turunan pertama fungsi aktivasi pada *neuron output*. Perubahan pembobot ini selanjutnya digunakan untuk mengupdate pembobot antar *layer*. Masing-masing bobot baru yang menghubungkan *neuron-neuron* pada *layer hidden* dengan *output* dan *layer input* dengan *hidden* adalah.

$$w_{jk}(\text{baru}) = w_{jk}(\text{lama}) + \Delta w_{jk}, j = 0,1,2, \dots, p \tag{6}$$

$$v_{ij}(\text{baru}) = v_{ij}(\text{lama}) + \Delta v_{ij}, \quad i = 0,1,2, \dots, n \tag{7}$$

Particle Swarm Optimization (PSO)

Particle Swarm Optimization (PSO) adalah salah satu metode optimasi yang diperkenalkan oleh Eberhart dan Kennedy pada tahun 1995. Misalkan X_{ad} dan V_{ad} ($a = 1,2, \dots, N$; $d = 1,2, \dots, D$) adalah posisi dan kecepatan partikel ke- a dimensi ke- d , N adalah jumlah *swarm*, dan D adalah dimensi partikel. Untuk memperbarui posisi dan kecepatan partikel digunakan persamaan berikut (Marini & Walczak, 2015; Yang et al., 2017).

$$V_{ad}^{u+1} = V_{ad}^u + c_1 r_1 (P_{ad}^u - X_{ad}^u) + c_2 r_2 (G_{ad}^u - X_{ad}^u) \tag{8}$$

$$x_{ad}^{u+1} = x_{ad}^u + V_{ad}^{u+1} \tag{9}$$

Dengan V_{ad}^{u+1} menggambarkan kecepatan partikel ke- a dimensi ke- d pada iterasi ke- $u + 1$, V_{ad}^u dan X_{ad}^u menggambarkan kecepatan dan posisi partikel ke- a dimensi ke- d pada iterasi ke- u . P_{ad}^u adalah posisi terbaik yang diperoleh partikel ke- a dimensi ke- d pada iterasi ke- u dan G_{ad}^u adalah posisi terbaik partikel pada P_{ad}^u iterasi ke- u . c_1 dan c_2 adalah konstanta kecepatan positif, r_1 dan r_2 adalah bilangan acak yaitu $[0,1]$, sedangkan u mewakili iterasi saat ini dari proses optimasi. Untuk mengontrol kemampuan eksplorasi algoritma PSO, dapat dituliskan (Marini & Walczak, 2015; Yang et al., 2017).

$$V_{ad}^{u+1} = \omega V_{ad}^u + c_1 r_1 (P_{ad}^u - X_{ad}^u) + c_2 r_2 (G_{ad}^u - X_{ad}^u) \tag{10}$$

Bobot inersia yang mencapai keseimbangan:

$$\omega = \omega_{max} - \frac{(\omega_{max} - \omega_{min})}{u_{max}} \times u \tag{11}$$

Dengan u_{max} adalah jumlah iterasi, u adalah iterasi saat ini, ω_{max} dan ω_{min} berturut-turut menunjukkan bobot inersia awal dan akhir dalam rentang 0.9 hingga 0.3.

Normalisasi dan Denormalisasi

Normalisasi data adalah proses transformasi data sehingga data memiliki rentang yang sama. Salah satu metode yang digunakan untuk normalisasi data adalah metode *Min-Max Normalization* yang dirumuskan dengan:

$$X_{norm} = \frac{X' - \min(X')}{\max(X') - \min(X')} \tag{13}$$

dengan X_{norm} adalah data ternormalisasi, X' data actual, $\min(X')$ adalah nilai minimal data actual, dan $\max(X')$ nilai maksimum data actual. Untuk keperluan prediksi, data yang sebelumnya telah dinormalisasi dikembalikan lagi ke data asal. Sedangkan untuk keperluan evaluasi model digunakan *mean absolute presentage error (MAPE)*. MAPE merupakan salah satu kriteria dalam menentukan kebaikan dari model. Rumus perhitungan MAPE. Untuk ukuran data sebesar n , nilai prediksi y_k dan nilai target t_k , $k=1,2, \dots, n$, maka nilai MAPE dirumuskan dengan

$$MAPE = \frac{1}{n} \sum_{t=1}^n \left| \frac{t_k - y_k}{t_k} \right| \times 100\% \tag{14}$$

3. HASIL DAN PEMBAHASAN

Hasil

Pengumpulan Data

Data yang digunakan pada penelitian ini merupakan data sekunder yang diperoleh dalam rentang waktu 2 Maret 2020 -31 Agustus 2021. Dataset yang telah dikumpulkan yaitu sebanyak 548 data harian terkonfirmasi kasus positif Covid-19 di Indonesia dan dapat dilihat pada Tabel 1.

Tabel 1. Data Jumlah Terkonfirmasi Positif Covid-19 di Indonesia

| No. | Hari | Terkonfirmasi |
|-----|------------|---------------|
| 1 | 02/03/2020 | 2 |
| 2 | 03/03/2020 | 0 |
| 3 | 04/03/2020 | 0 |
| 4 | 05/03/2020 | 0 |
| 5 | 06/03/2020 | 2 |
| ⋮ | ⋮ | ⋮ |
| 545 | 28/08/2021 | 10050 |
| 546 | 29/08/2021 | 7427 |
| 547 | 30/08/2021 | 5436 |
| 548 | 31/08/2021 | 10534 |

Normalisasi Data

Pada tahap ini akan dilakukan proses normalisasi data berdasarkan persamaan (18). Hasil normalisasi dapat dilihat pada Tabel 2.

Tabel 2. Data Normalisasi

| No. | Hari | Terkonfirmasi |
|-----|------------|---------------|
| 1 | 02/03/2020 | 0,0000 |
| 2 | 03/03/2020 | 0,0000 |
| 3 | 04/03/2020 | 0,0000 |
| 4 | 05/03/2020 | 0,0000 |
| 5 | 06/03/2020 | 0,0000 |
| ⋮ | ⋮ | ⋮ |
| 545 | 28/08/2021 | 0,1771 |
| 546 | 29/08/2021 | 0,1309 |
| 547 | 30/08/2021 | 0,0958 |
| 548 | 31/08/2021 | 0,1856 |

Perancangan Dataset

Setelah melakukan pengumpulan data dan normalisasi data, selanjutnya melakukan perancangan dataset berdasarkan variabel yang sudah ditentukan pada Tabel 3 dan hasil perancangan dataset prediksi dapat dilihat pada Tabel 4. Pengaturan variabel bebas dilakukan dengan cara sebagai berikut. Variabel x_1 adalah variabel jumlah kasus positif Covid-19 pada hari $h-14$. Variabel x_2 adalah variabel jumlah kasus positif Covid-19 pada hari $h-13$ demikian seterusnya. Variabel tak bebas, adalah jumlah kasus positif Covid-19 hari ke- h .

Tabel 3. Penjelasan Variabel

| Variabel Input | Keterangan |
|-----------------|---|
| x_1 | Data jumlah kasus positif Covid-19 pada hari ke- $h - 14$ |
| x_2 | Data jumlah kasus positif Covid-19 pada hari ke- $h - 13$ |
| x_3 | Data jumlah kasus positif Covid-19 pada hari ke- $h - 12$ |
| x_4 | Data jumlah kasus positif Covid-19 pada hari ke- $h - 11$ |
| x_5 | Data jumlah kasus positif Covid-19 pada hari ke- $h - 10$ |
| x_6 | Data jumlah kasus positif Covid-19 pada hari ke- $h - 9$ |
| x_7 | Data jumlah kasus positif Covid-19 pada hari ke- $h - 8$ |
| x_8 | Data jumlah kasus positif Covid-19 pada hari ke- $h - 7$ |
| x_9 | Data jumlah kasus positif Covid-19 pada hari ke- $h - 6$ |
| x_{10} | Data jumlah kasus positif Covid-19 pada hari ke- $h - 5$ |
| x_{11} | Data jumlah kasus positif Covid-19 pada hari ke- $h - 4$ |
| x_{12} | Data jumlah kasus positif Covid-19 pada hari ke- $h - 3$ |
| x_{13} | Data jumlah kasus positif Covid-19 pada hari ke- $h - 2$ |
| x_{14} | Data jumlah kasus positif Covid-19 pada hari ke- $h - 1$ |
| Variabel Output | Keterangan |
| t | Data jumlah kasus positif Covid-19 pada hari ke- h |

Tabel 4. Dataset Hasil Rancangan

| Pola ke- | x_1 | x_2 | x_3 | ... | x_{12} | x_{13} | x_{14} | t |
|----------|--------|--------|--------|-----|----------|----------|----------|--------|
| 1 | 0.0000 | 0.0000 | 0.0000 | ... | 0.0006 | 0.0005 | 0.0004 | 0.0003 |
| 2 | 0.0000 | 0.0000 | 0.0000 | ... | 0.0005 | 0.0004 | 0.0003 | 0.0007 |
| 3 | 0.0000 | 0.0000 | 0.0000 | ... | 0.0004 | 0.0003 | 0.0007 | 0.0010 |
| 4 | 0.0000 | 0.0000 | 0.0000 | ... | 0.0003 | 0.0007 | 0.0010 | 0.0014 |
| 5 | 0.0000 | 0.0000 | 0.0000 | ... | 0.0007 | 0.0010 | 0.0014 | 0.0011 |
| 6 | 0,0000 | 0,0000 | 0,0002 | ... | 0,0010 | 0,0014 | 0,0011 | 0,0014 |
| 7 | 0,0000 | 0,0002 | 0,0001 | ... | 0,0014 | 0,0011 | 0,0014 | 0,0011 |
| 8 | 0,0002 | 0,0001 | 0,0001 | ... | 0,0011 | 0,0014 | 0,0011 | 0,0011 |
| ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ | ⋮ |
| 528 | 0,5396 | 0,4353 | 0,5425 | ... | 0,2186 | 0,1692 | 0,3366 | 0,3290 |
| 529 | 0,4353 | 0,5425 | 0,5039 | ... | 0,1692 | 0,3366 | 0,3290 | 0,2977 |
| 530 | 0,5425 | 0,5039 | 0,3667 | ... | 0,3366 | 0,3290 | 0,2977 | 0,2223 |
| 531 | 0,5039 | 0,3667 | 0,3063 | ... | 0,3290 | 0,2977 | 0,2223 | 0,1771 |
| 532 | 0,3667 | 0,3063 | 0,3654 | ... | 0,2977 | 0,2223 | 0,1771 | 0,1309 |
| 533 | 0,3063 | 0,3654 | 0,2778 | ... | 0,2223 | 0,1771 | 0,1309 | 0,0958 |
| 534 | 0,3654 | 0,2778 | 0,3886 | ... | 0,1771 | 0,1309 | 0,0958 | 0,1856 |

Pembagian Dataset

Untuk proses prediksi, dataset yang telah dirancang dibagi menjadi dua, yaitu data pelatihan (*training*) dan data pengujian (*testing*). Data pelatihan digunakan untuk pembentukan model, sedangkan data pengujian digunakan untuk prediksi sebagai pengujian dari model yang terbentuk. Pada penelitian ini dataset akan dibagi menjadi 90% dari dataset prediksi yaitu 481 pola data sebagai data pelatihan dan 10% dari dataset prediksi yaitu 53 pola data sebagai data pengujian.

Perancangan Arsitektur Model ANN

Pada penelitian ini arsitektur model ANN yang digunakan adalah arsitektur jaringan *multilayer network* dengan 3 *layer*. Arsitektur terdiri dari satu *input layer*, satu *hidden layer*, dan satu *output layer*. Jumlah *neuron* pada *input layer* terdiri dari 14 *neuron* yang berisi data jumlah kasus positif Covid-19 pada hari ke- $h - 1$ hingga hari ke- $h - 14$. Pada *hidden layer*, jumlah *neuron* ditentukan dengan proses *trial and error* dengan variasi 5, 10, 15, 20, 25, dan 30. Sedangkan untuk *output layer* terdiri dari 1 *neuron* yaitu jumlah kasus positif Covid-19 pada hari ke- h . Penentuan *learning rate* dilakukan dengan proses *trial and error*, pada penelitian ini dilakukan pengujian *learning rate* dengan variasi 0,1; 0,2; dan 0,3.

Tabel 5. Hasil Akurasi Pengujian Menggunakan Algoritma BPNN

| Pengujian ke- | Hidden Neuron | Learning Rate | MAPE Pengujian |
|---------------|---------------|---------------|----------------|
| 1 | 5 | 0.1 | 30.13% |
| 2 | 5 | 0.2 | 18.55% |
| 3 | 5 | 0.3 | 16.29% |
| 4 | 10 | 0.1 | 25.60% |
| 5 | 10 | 0.2 | 16.71% |
| 6 | 10 | 0.3 | 17.17% |
| 7 | 15 | 0.1 | 20.81% |
| 8 | 15 | 0.2 | 20.25% |
| 9 | 15 | 0.3 | 19.30% |
| 10 | 20 | 0.1 | 29.37% |
| 11 | 20 | 0.2 | 23.17% |
| 12 | 20 | 0.3 | 22.97% |
| 13 | 25 | 0.1 | 32.70% |
| 14 | 25 | 0.2 | 34.68% |
| 15 | 25 | 0.3 | 35.95% |
| 16 | 30 | 0.1 | 42.38% |
| 17 | 30 | 0.2 | 47.09% |
| 18 | 30 | 0.3 | 47.19% |

Pemilihan model prediksi yang optimal yaitu berdasarkan pengujian dengan nilai MAPE terendah. Pada **Tabel 5** nilai MAPE terendah diperoleh pada pengujian ke-3 dengan kombinasi parameter *learning*

rate 0.3 dan jumlah *heuron* pada *hidden layer* sebanyak 5 *neuron* atau arsitektur 14-5-1 yaitu sebesar 16.29%.

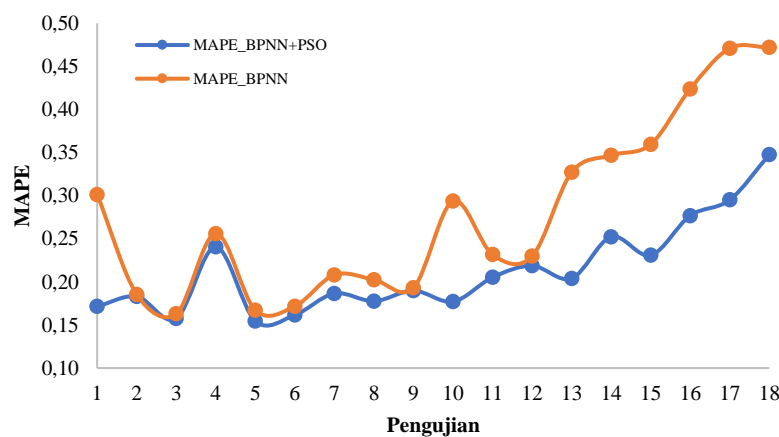
Pengujian Model Prediksi dengan BPNN-PSO

Parameter yang dibutuhkan pada pengujian model prediksi dengan BPNN-PSO yaitu parameter ANN dan parameter PSO. Parameter ANN telah dijelaskan sebelumnya, dan parameter PSO yang digunakan yaitu jumlah *swarm* sebanyak 40, $\omega_{max} = 0.9$, $\omega_{min} = 0.3$, $c1 = 1.5$, $c2 = 2$, dan jumlah iterasi sebanyak 50. Partikel pada penelitian ini adalah jumlah bobot dan bias untuk BPNN, sehingga dimensi *partikel* ditentukan dengan menggunakan rumus berikut ini (Huang dan Wu, 2017): Nilai MAPE untuk ANN dengan variasi banyaknya *hidden neuron* (5 sampai 30) dan variasi berbagai *learning rate* (0,1 sampai 0,3) disajikan pada Tabel 6.

Tabel 6. Hasil Akurasi Pengujian Menggunakan Algoritma BPNN-PSO

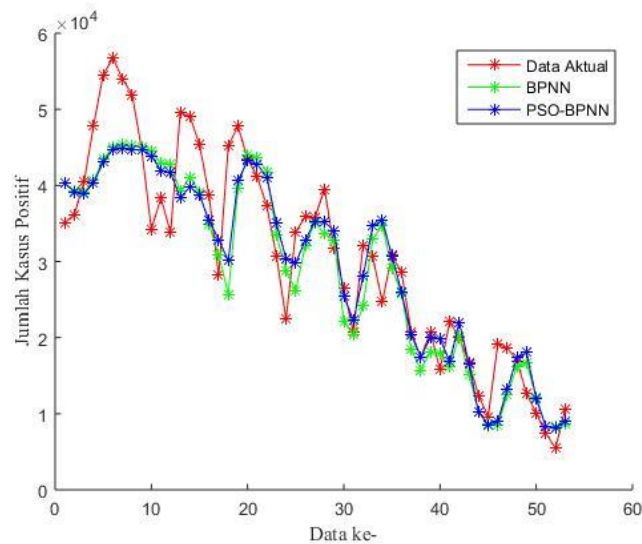
| Pengujian ke- | Hidden Neuron | Learning Rate | MAPE Pengujian |
|---------------|---------------|---------------|----------------|
| 1 | 5 | 0.1 | 17.14% |
| 2 | 5 | 0.2 | 18.30% |
| 3 | 5 | 0.3 | 15.75% |
| 4 | 10 | 0.1 | 24.06% |
| 5 | 10 | 0.2 | 15.45% |
| 6 | 10 | 0.3 | 16.16% |
| 7 | 15 | 0.1 | 18.65% |
| 8 | 15 | 0.2 | 17.76% |
| 9 | 15 | 0.3 | 18.98% |
| 10 | 20 | 0.1 | 17.71% |
| 11 | 20 | 0.2 | 20.52% |
| 12 | 20 | 0.3 | 21.86% |
| 13 | 25 | 0.1 | 20.40% |
| 14 | 25 | 0.2 | 25.22% |
| 15 | 25 | 0.3 | 23.09% |
| 16 | 30 | 0.1 | 27.68% |
| 17 | 30 | 0.2 | 29.51% |
| 18 | 30 | 0.3 | 34.76% |

Berdasarkan hasil pada Tabel 6, nilai MAPE cenderung mengecil ketika banyaknya *hidden neuron* dari 5 sampai 10, namun nilai MAPE cenderung membesar ketika *hidden unit* membesar untuk *hidden unit* yang lebih dari 10. Hasil prediksi model ANN standar dan ANN algoritma BPNN-PSO pada data pengujian 1 sampai 18 dapat dilihat pada Gambar 3.



Gambar 3. Grafik Perbandingan MAPE Algoritma BPNN dan BPNN-PSO

Plot data perbandingan hasil prediksi algoritma BPNN dan algoritma BPNN-PSO dengan data aktual pada arsitektur terbaik disetiap algoritmanya yang dapat dilihat pada Gambar 4.



Gambar 4. Perbandingan Data Aktual dengan Hasil Prediksi Algoritma BPNN dan Algoritma BPNN-PSO

Pencarian pola nilai-nilai aktual (warna merah), nilai prediksi BPNN (warna hijau) dan BPNN-PSO (warna Biru) memberikan akurasi berbeda. Hasil MAPE sebesar 15,45% tersermin dari perbedaan pencarian antara aktual pada algoritma BPNN-PSO.

Pembahasan

Aplikasi model ANN dari penelitian sebelumnya penggunaan *Particle Swarm Optimization* (PSO) untuk memilih bobot awal cukup efektif dalam meningkatkan akurasi model metode BPNN. Dalam penelitian ini dikaji pula bagaimana performa PSO dalam meningkatkan akurasi pada model ANN. Pengujian model ANN metode BPNN standar dengan menggunakan kombinasi parameter *learning rate* dan jumlah *neuron* pada *hidden layer* memberikan hasil nilai MAPE yang bervariasi (Umair, 2010; Yan & Au, 2019). Jumlah *neuron* pada *hidden layer* dan nilai *learning rate* memiliki pengaruh yang signifikan pada akurasi model. Nilai MAPE pada pengujian ke-1 hingga ke-3, dengan menggunakan 5 *hidden neuron* dan *learning rate* 0.1 hingga 0.3 secara berturut-turut adalah 30,13%, 18,55%, dan 16,29%. Berdasarkan pengamatan tersebut, ketika *learning rate* yang digunakan semakin tinggi, maka nilai MAPE cenderung menurun dan menandakan bahwa penggunaan *learning rate* yang lebih tinggi dapat membantu model untuk mencapai hasil prediksi yang lebih akurat. Pada pengujian ke-4 hingga ke-6, menunjukkan bahwa 10 *hidden neuron* dengan *learning rate* 0,2 menghasilkan nilai MAPE terendah dari semua pengujian yaitu sebesar 16,71%. Hal ini menunjukkan bahwa jumlah *hidden neuron* dan *learning rate* ini memberikan akurasi prediksi yang baik. Pengujian selanjutnya menggunakan 15 *hidden neuron* dengan variasi *learning rate*, diperoleh nilai MAPE terendah sebesar 19,30% pada penggunaan *learning rate* 0,3. Sehingga penggunaan jumlah *hidden neuron* yang lebih banyak dan *learning rate* yang lebih tinggi memberikan performa prediksi yang lebih baik. Hal tersebut juga berbanding lurus dengan penggunaan jumlah *hidden neuron* sebanyak 20. Namun, ketika pengujian dilakukan dengan 25 dan 30 *hidden neuron*, beberapa konfigurasi menghasilkan nilai MAPE yang cukup tinggi, khususnya pada *learning rate* yang rendah yaitu mencapai 47,09% pada pengujian ke-17. Ini menunjukkan bahwa kombinasi jumlah *hidden neuron* yang lebih sedikit dengan *learning rate* yang rendah cenderung menyebabkan *overfitting*, dimana model terlalu rumit untuk data pelatihan dan kurang generalisasi untuk data uji. Namun secara keseluruhan, penggunaan *learning rate* yang lebih tinggi pada jumlah *hidden neuron* yang lebih sedikit cenderung menghasilkan hasil prediksi yang lebih akurat, tetapi pemilihan kombinasi jumlah *hidden neuron* dengan *learning rate* yang tepat sangat penting dalam membangun model BPNN yang akurat (Grzenda et al., 2023; Liu, 2021).

Pengujian model ANN metode BPNN yang dikombinasikan dengan PSO juga menggunakan variasi parameter yang sama seperti pengujian sebelumnya. PSO digunakan untuk memperoleh bobot dan bias awal pada proses pelatihan BPNN. Penggunaan PSO untuk mengoptimalkan parameter dalam BPNN membantu meningkatkan performa keseluruhan. Berdasarkan hasil dari metode ini, nilai MAPE yang diperoleh lebih rendah dibandingkan dengan pengujian metode BPNN standar yaitu berkisar 15,45% hingga 34,76%. Peningkatan jumlah *hidden neuron* dalam ANN cenderung meningkatkan akurasi prediksi. Tidak selalu peningkatan *hidden neuron* akan menghasilkan akurasi yang lebih baik. Hal tersebut dapat diamati pada pengujian ke-4 hingga ke-6, meskipun jumlah *hidden neuron* ditingkatkan menjadi 10, akurasi prediksi justru menurun dibandingkan dengan pengujian yang menggunakan 5 *hidden neuron*. Selanjutnya,

learning rate juga mempengaruhi akurasi prediksi secara signifikan. Terlihat pada pengujian ke-2, ketika *learning rate* ditingkatkan menjadi 0,2, terjadi penurunan akurasi dibandingkan dengan pengujian yang menggunakan *learning rate* 0,1 pada *hidden neuron* yang sama. Dalam beberapa kasus, peningkatan jumlah *hidden neuron* atau *learning rate* dapat menyebabkan *overfitting*. Pengujian dengan 30 *hidden neuron* dan *learning rate* 0.3 menunjukkan akurasi prediksi yang paling tinggi yaitu sebesar 34.76%. Sebaliknya, pengujian dengan 10 *hidden neuron* dan *learning rate* 0.2 menunjukkan akurasi prediksi terbaik yaitu sebesar 15,45%. Dalam beberapa kasus, kombinasi yang tampak sederhana dapat memberikan hasil yang lebih baik daripada pengaturan yang lebih kompleks. Karena itu, eksplorasi dan penyesuaian parameter secara tepat masih diperlukan untuk mencapai hasil akurasi yang optimal dalam penggunaan BPNN-PSO (Cai et al., 2019; Liu, 2021).

Pada algoritma BPNN-PSO, menunjukkan bahwa performa akurasi secara umum lebih baik dibandingkan dengan BPNN standar. Penggunaan Particle Swarm Optimization sebagai penentu nilai bobot dan bias awal pada proses pelatihan dapat membantu BPNN untuk mencapai konvergensi yang lebih baik dan mendapatkan parameter yang optimal sehingga berdampak terhadap peningkatan hasil akurasi model (Huang & Wu, 2017; Ridla, 2018). Namun, tidak ada kombinasi jumlah *hidden neuron* dan *learning rate* yang secara konsisten memberikan hasil yang lebih baik untuk kedua metode. Oleh karena itu, dalam memilih konfigurasi yang optimal, perlu dilakukan pengujian lebih lanjut dengan variasi parameter lainnya atau mungkin menggunakan metode lain yang lebih sesuai dengan kasus yang sedang diteliti.

4. SIMPULAN

Hasil algoritma BPNN untuk memprediksi jumlah kumulatif kasus positif Covid-19 di Indonesia memberikan nilai MAPE yang cenderung lebih besar dibandingkan MAPE dari hasil algoritma BPNN-PSO. Hal ini mengindikasikan bahwa kombinasi BPNN-PSO berhasil untuk meningkatkan akurasi hasil prediksi. Implementasi algoritma *Particle Swarm Optimization* (PSO) yang berperan sebagai penentu nilai bobot dan bias awal dalam proses pelatihan metode *Backpropagation Neural Network* (BPNN) dapat meningkatkan performa model ANN untuk memprediksi jumlah kumulatif kasus positif Covid-19 di Indonesia, dengan peningkatan akurasi yang signifikan. Hal tersebut dapat dibuktikan melalui perbandingan akurasi menggunakan nilai MAPE. Pada setiap pengujian, metode BPNN-PSO berhasil mencapai nilai MAPE yang lebih rendah daripada metode BPNN standar. Sehingga penggunaan model ANN dengan algoritma BPNN-PSO dapat diterapkan dalam situasi ini untuk memprediksi jumlah kumulatif kasus positif Covid-19 di Indonesia.

5. DAFTAR PUSTAKA

- Ahmad, I., & Asad, S. M. (2020). Predictions of coronavirus COVID-19 distinct cases in Pakistan through an artificial neural network. *Epidemiology & Infection*, 148. <https://doi.org/10.1017/S0950268820002174>.
- Amalia, L., Irwan, I., & Hiola, F. (2020). Analisis gejala klinis dan peningkatan kekebalan tubuh untuk mencegah penyakit covid 19. *Journal of Health Sciences and Research*, 2(2). <https://doi.org/10.35971/jjhsr.v2i2.6134>.
- Ardabili, S., Mosavi, A., Band, S. S., & Varkonyi-Koczy, A. R. (2020). Coronavirus disease (COVID-19) global prediction using hybrid artificial intelligence method of ANN trained with Grey Wolf optimizer. *IEEE 3rd International Conference and Workshop in Óbuda on Electrical and Power Engineering (CANDO-EPE)*, 251–254.
- Arianto, F. S. D., & Noviyanti, P. (2020). Prediksi kasus COVID-19 di Indonesia menggunakan metode backpropagation dan fuzzy Tsukamoto. (*JurTI*) *Jurnal Teknologi Informasi*, 4(1), 120–127. <https://doi.org/10.36294/jurti.v4i1.1265>.
- Cahyaningsih, A. (2020). Model Prediksi Jumlah Kumulatif Kasus COVID-19 di Indonesia Menggunakan Deep Learning. *Journal of Informatics Information System Software Engineering and Applications (INISTA)*, 3(1), 76–83. <https://doi.org/10.20895/inista.v3i1.177>.
- Cai, G.-W., Fang, Z., & Chen, Y.-F. (2019). Estimating the number of hidden nodes of the single-hidden-layer feedforward neural networks. *15th International Conference on Computational Intelligence and Security (CIS)*, 172–176. <https://doi.org/10.1109/CIS.2019.00044>.
- Cynthia, E. P., & Ismanto, E. (2017). Jaringan syaraf tiruan algoritma backpropagation dalam memprediksi ketersediaan komoditi pangan provinsi riau. *Rabit: Jurnal Teknologi Dan Sistem Informasi Univrab*, 2(2), 83–98. <https://doi.org/10.36341/rabit.v2i2.152>.
- Dwiyanto, F. A., Pakpahan, H. S., Islamiyah, I., & Setyadi, H. J. (2019). Pengenalan Karakter Tulisan Menggunakan Metode Backpropagation Neural Network. *Sains, Aplikasi, Komputasi Dan Teknologi*

- Informasi*, 1(2), 14–22. <https://doi.org/10.30872/jsakti.v1i2.2601>.
- Fernandes, N., Costa, D., Costa, D., Keating, J., & Arantes, J. (2021). Predicting COVID-19 vaccination intention: The determinants of vaccine hesitancy. *Vaccines*, 9(10), 1161. <https://doi.org/10.3390/vaccines9101161>.
- Ghanim, M. S., Muley, D., & Kharbeche, M. (2022). ANN-Based traffic volume prediction models in response to COVID-19 imposed measures. *Sustainable Cities and Society*, 8(1), 103830. <https://doi.org/10.1016/j.scs.2022.103830>.
- Grzenda, M., Kaźmierczak, S., Luckner, M., Borowik, G., & Mańdziuk, J. (2023). Evaluation of machine learning methods for impostor detection in web applications. *Expert Systems with Applications*, 231, 120736. <https://doi.org/10.1016/j.eswa.2023.120736>.
- Guo, X., Ji, M., Zhao, Z., Wen, D., & Zhang, W. (2020). Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization (PSO) algorithm. *Ocean Engineering*, 21(6), 107693. <https://doi.org/10.1016/j.oceaneng.2020.107693>.
- Hamadneh, N. N., Khan, W. A., Ashraf, W., Atawneh, S. H., Khan, I., & Hamadneh, B. N. (2021). Artificial neural networks for prediction of covid-19 in Saudi Arabia. *Computers, Materials, & Continua*, 2787–2796. <https://doi.org/10.32604/cmc.2021.013228>.
- Huang, D., & Wu, Z. (2017). Forecasting outpatient visits using empirical mode decomposition coupled with back-propagation artificial neural networks optimized by particle swarm optimization. *PLOS ONE*, 12(2), e0172539. <https://doi.org/10.1371/journal.pone.0172539>.
- Jena, P. R., Majhi, R., Kalli, R., Managi, S., & Majhi, B. (2021). Impact of COVID-19 on GDP of major economies: Application of the artificial neural network forecaster. *Economic Analysis and Policy*, 69, 324–339. <https://doi.org/10.1016/j.eap.2020.12.013>.
- Kusmaryanto, S. (2014). Jaringan Saraf Tiruan Backpropagation untuk Pengenalan Wajah Metode Ekstraksi Fitur Berbasis Histogram. *Jurnal EECCIS (Electrics, Electronics, Communications, Controls, Informatics, Systems)*, 8(2), 193–198. <https://doi.org/10.21776/jeccis.v8i2.283>.
- Li, J., Luo, H., Deng, G., Chang, J., Qiu, X., Liu, C., & Qin, B. (2021). Multidimensional Evaluation of All-Cause Mortality Risk and Survival Analysis for Hospitalized Patients with COVID-19. *International Journal of Medical Sciences*, 18(14), 3140. <https://doi.org/10.7150%2Fijms.58889>.
- Liu, L. (2021). Research on Digital Economy of Intelligent Emergency Risk Avoidance in Sudden Financial Disasters Based on PSO-BPNN Algorithm. *Computational Intelligence and Neuroscience*, 2021, 10. <https://doi.org/10.1155/2021/7708422>.
- Marini, F., & Walczak, B. (2015). Particle swarm optimization (PSO). A tutorial. *Chemometrics and Intelligent Laboratory Systems*, 149, 153–165. <https://doi.org/10.1016/j.chemolab.2015.08.020>.
- Meimela, A., Lestari, S. S. S., Mahdy, I. F., Toharudin, T., & Ruchjana, B. N. (2021). Modeling of covid-19 in Indonesia using vector autoregressive integrated moving average. *Journal of Physics: Conference Series*, 1722(1), 012079. <https://doi.org/10.1088/1742-6596/1722/1/012079>.
- Meutia, S. M. S., & Himayani, R. (2021). Central and Peripheral Nervous System. *Medical Profession Journal of Lampung*, 11(3), 306–311. <https://doi.org/10.53089/medula.v11i3.189>.
- Mohamad, E. T., Faradonbeh, R. S., Armaghani, D. J., Monjezi, M., & Majid, M. Z. A. (2017). An optimized ANN model based on genetic algorithm for predicting ripping production. *Neural Computing and Applications*, 28(1), 393–406. <https://doi.org/10.1007/s00521-016-2359-8>.
- Mollalo, A., Rivera, K. M., & Vahedi, B. (2020). Artificial neural network modeling of novel coronavirus (COVID-19) incidence rates across the continental United States. *International Journal of Environmental Research and Public Health*, 17(12), 4204. <https://doi.org/10.3390/ijerph17124204>.
- Ndaïrou, F., Area, I. J., N. J., & Torres, D. F. M. (2020). Mathematical modeling of COVID-19 transmission dynamics with a case study of Wuhan. *Chaos, Solitons & Fractals*, 135, 109846. <https://doi.org/10.1016/j.chaos.2020.109846>.
- Niazkar, H. R., & Niazkar, M. (2020). Application of artificial neural networks to predict the COVID-19 outbreak. *Global Health Research and Policy*, 5(1), 1–11. <https://doi.org/10.1186/s41256-020-00175-y>.
- Ojokoh, B. A., Sarumi, O. A., Salako, K. V., Gabriel, A. J., Taiwo, A. E., Johnson, O. V., Adegun, I. P., & Babalola, O. T. (2022). Modeling and predicting the spread of COVID-19. *Data Science for COVID-19*, 299–317. <https://doi.org/10.1016/B978-0-323-90769-9.00039-6>.
- Pradana, D., Alghifari, M. L., Juna, M. F., & Palaguna, D. (2022). Klasifikasi Penyakit Jantung Menggunakan Metode Artificial Neural Network. *Indonesian Journal of Data and Science*, 3(2), 55–60. <https://doi.org/10.56705/ijodas.v3i2.35>.
- Rahakbauw, D. L. (2014). Analisis Jaringan Saraf Tiruan Backpropagation Terhadap Peramalan Nilai Tukar Mata Uang Rupiah Dan Dolar. *BAREKENG: Jurnal Ilmu Matematika Dan Terapan*, 8(2), 27–32.

- <https://doi.org/10.30598/barekengvol8iss2pp27-32>.
- Ridla, M. A. (2018). Particle Swarm Optimization Sebagai Penentu Nilai Bobot Pada Artificial Neural Network Berbasis Backpropagation Untuk Prediksi Tingkat Penjualan Minyak Pelumas Pertamina. *Jurnal Ilmiah Informatika*, 3(1), 183–192. <https://doi.org/10.35316/jimi.v3i1.473>.
- Solikhun, S., Safii, M., & Trisno, A. (2017). Jaringan Saraf Tiruan Untuk Memprediksi Tingkat Pemahaman Sisiwa Terhadap Matapelajaran Dengan Menggunakan Algoritma Backpropagation. *J-SAKTI (Jurnal Sains Komputer Dan Informatika)*, 1(1), 24–36. <https://doi.org/10.30645/j-sakti.v1i1.26>.
- Suhendra, C. D., & Wardoyo, R. (2015). Penentuan arsitektur jaringan syaraf tiruan backpropagation (bobot awal dan bias awal) menggunakan algoritma genetika. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 9(1), 77–88. <https://doi.org/10.22146/ijccs.6642>.
- Susilo, A., Rumende, C. M., Pitoyo, C. W., Santoso, W. D., Yulianti, M., Herikurniawan, H., Sinto, R., Singh, G., Nainggolan, L., & Nelwan, E. J. (2020). Coronavirus disease 2019: Tinjauan literatur terkini. *Jurnal Penyakit Dalam Indonesia*, 7(1), 45–67. <https://doi.org/10.7454/jpdi.v7i1.415>.
- Tealab, A. (2018). Time series forecasting using artificial neural networks methodologies: A systematic review. *Future Computing and Informatics Journal*, 3(2), 334–340. <https://doi.org/10.1016/j.fcij.2018.10.003>.
- Thomas, A. J., Petridis, M., Walters, S. D., Gheytaasi, S. M., & Morgan, R. E. (2015). On predicting the optimal number of hidden nodes. *International Conference on Computational Science and Computational Intelligence (CSCI)*, 565–570. <https://doi.org/10.1109/CSCI.2015.33>.
- Torrealba-Rodríguez, O., Conde-Gutiérrez, R. A., & Hernández-Javier, A. L. (2020). Modeling and prediction of COVID-19 in Mexico applying mathematical and computational models. *Chaos, Solitons & Fractals*, 138, 109946. <https://doi.org/10.1016/j.chaos.2020.109946>.
- Umair, S. M. (2010). Automation of Irrigation System Using ANN based Controller. *International Journal*, 02, 45–51.
- Verianto, E., & Oetomo, B. S. D. (2020). Artificial Neural Network Model with PSO as a Learning Method to Predict Movement of the Rupiah Exchange Rate against the US Dollar. *IJAIT (International Journal of Applied Information Technology)*, 4(02), 81–92. <https://doi.org/10.25124/ijait.v4i02.3381>.
- Wang, L., Wang, Z., Qu, H., & Liu, S. (2018). Optimal Forecast Combination Based on Neural Networks for Time Series Forecasting. *Applied Soft Computing*, 66, 1–17. <https://doi.org/10.1016/j.asoc.2018.02.004>.
- Yan, N., & Au, O. T.-S. (2019). Online learning behavior analysis based on machine learning. *Asian Association of Open Universities Journal*, 14(2), 97–106. <https://doi.org/10.1108/aaouj-08-2019-0029>.
- Yang, X., Xu, G., Li, Q., Guo, Y., & Zhang, M. (2017). Authorship attribution of source code by using back propagation neural network based on particle swarm optimization. *PloS One*, 12(11), e0187204. <https://doi.org/10.1371/journal.pone.0187204>.