

PENENTUAN BOBOT MODEL NEURAL NETWORK UNTUK DATA TIME SERIES MENGGUNAKAN ALGORITMA GENETIKA

BUDI WARSITO¹, SUBANAR², ABDURAKHMAN³, WIDODO⁴

¹Jurusan Statistika Fakultas Sains dan Matematika UNDIP, Mahasiswa S3 Jurusan Matematika FMIPA UGM, email : budi_wrst@yahoo.com

²Jurusan Matematika FMIPA UGM, email : subanar@yahoo.com

³Jurusan Matematika FMIPA UGM, email : rachmanstat@ugm.ac.id

⁴Jurusan Matematika FMIPA UGM, email : widodo_math@yahoo.com

abstract

Pada beberapa tahun terakhir studi literatur yang berkaitan dengan pemodelan Neural Network telah berkembang dengan pesat. Namun penentuan bobot untuk mencapai solusi optimum pada model Neural Network dengan metode optimasi standar menyisakan beragam permasalahan diantaranya adalah ketidakstabilan hasil estimasi yang berkaitan dengan inisialisasi bobot secara random. Pada penelitian ini dibahas pendekatan yang berbeda dalam mengestimasi bobot model Neural Network yaitu dengan menggunakan algoritma genetika. Algoritma ini secara praktis sangat efektif dalam optimasi fungsi dan efisien dalam pencarian pada ruang yang kompleks untuk mendapatkan nilai optimum global. Metode ini diterapkan untuk aplikasi pada peramalan data time series finansial. Hasil penelitian menunjukkan bahwa optimasi dengan algoritma genetika dan metode *hybrid* menghasilkan variansi error yang lebih kecil dari metode Levenberg-Marquardt.

Kata Kunci : Neural Network, algoritma genetika, Levenberg-Marquardt, time series

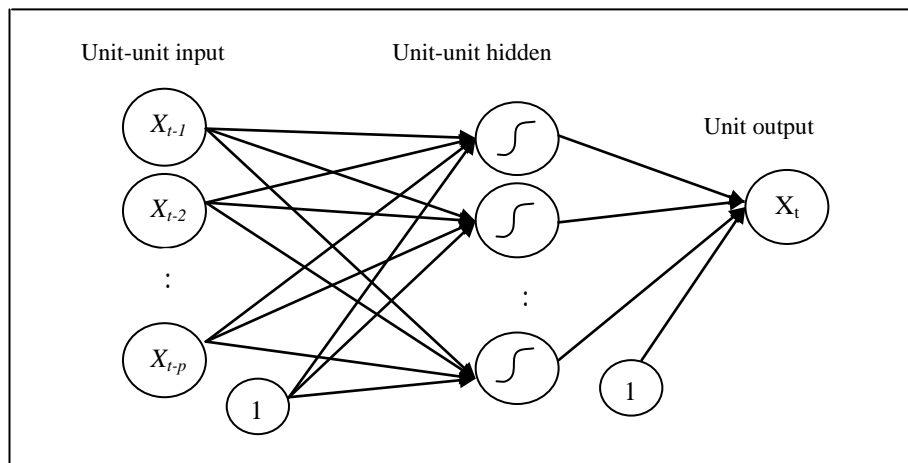
1. Pendahuluan

Pada perkembangan pemodelan statistika non parametrik, salah satu model yang banyak dikembangkan adalah Neural Network (NN). Model ini mempunyai fungsi aktivasi atau semacam fungsi transfer nonlinear yang memungkinkan untuk melakukan berbagai analisis data dengan output yang diinginkan, termasuk analisis data time series. Algoritma model juga memungkinkan untuk memilih berbagai metode optimasi untuk mendapatkan output dengan kesalahan yang kecil. Konstruksi arsitektur/model NN juga sangat fleksible yang menjadikan model ini terbuka untuk dikaji lebih lanjut. Fleksibilitas yang dimiliki diantaranya adalah pemilihan input model, fungsi aktivasi yang digunakan serta pemilihan metode optimasi pada pelatihan jaringan untuk mendapatkan bobot optimal. Hasil-hasil penelitian yang telah dilakukan menunjukkan bahwa pada penerapan model NN untuk data time series, resolusi model yang meliputi ketepatan prediksi maupun waktu yang diperlukan sangat dipengaruhi oleh pemilihan bobot awal [1]. Bobot-bobot dalam model NN biasanya diinisialisasi secara random dengan nilai-nilai random yang kecil [2]. Jika inisialisasi bobot random diperoleh nilai yang jauh dari solusi yang baik, atau dekat dengan nilai optimum lokal yang kurang baik, proses training mungkin akan memerlukan waktu yang cukup lama atau mungkin akan terjebak dalam optimum lokal.

Ketepatan inialisasi akan dapat membuat bobot-bobot tersebut mempunyai kemungkinan yang lebih besar untuk mencapai solusi yang baik dengan waktu training yang singkat. Pertanyaan yang kemudian muncul adalah apakah nilai optimal yang diperoleh benar-benar mencapai optimum global atau hanya optimum lokal. Beberapa metode optimasi yang digunakan seringkali tidak mampu memberikan hasil optimal global. Selain itu metode iterasi yang digunakan sangat sulit untuk menentukan nilai awal yang dapat mencapai konvergensi dan diperlukan *trial and error* [3]. Beberapa penelitian kemudian mengembangkan metode optimasi alternatif dengan algoritma pencarian global, salah satu diantaranya adalah algoritma genetika yang dikembangkan pertama kali oleh Holland (1975) dan kemudian dilanjutkan oleh Goldberg (1989) serta beberapa peneliti yang lain. Keunggulan dari algoritma genetika diantaranya adalah berproses sangat baik untuk optimisasi global khususnya jika fungsi objektifnya diskontinu atau mempunyai beberapa minimum lokal. Keunggulan ini menjadikan algoritma genetika dapat digunakan sebagai teknik optimasi pada model NN untuk mendapatkan bobot optimal. Beberapa penelitian yang telah menggunakan Algoritma Genetika untuk optimasi model NN diantaranya adalah [4], [5], [6], [7], [8], [9], [10] dan [11].

2. Neural Network untuk Data Time Series

Model Neural Network yang akan diamati secara khusus dalam penelitian ini adalah model Feed Forward Neural Network (FFNN) dengan penerapan pada data time series. Arsitektur jaringan FFNN untuk peramalan data time series yang dikembangkan adalah konfigurasi unit input X_{t-1} sampai X_{t-p} dan satu unit konstan (bias), sebuah *hidden layer* yang terdiri dari N neuron dan 1 unit output. Arsitektur tersebut disajikan pada gambar 1.



Gambar 1 Arsitektur FFNN untuk peramalan data time series dengan satu hidden layer yang terdiri dari N neuron dan variabel input nilai pada lag $X_{t-1}, X_{t-2}, \dots, X_{t-p}$

Model dari arsitektur pada gambar 1 tersebut dapat ditulis dalam bentuk persamaan berikut :

$$X_t = \psi_0 \left\{ w_{bo} + \sum_{n=1}^N w_{no} \psi_n \left(w_{bn} + \sum_{i=1}^p w_{in} X_{t-i} \right) \right\} \quad (1)$$

dimana w_{bn} adalah bobot antara unit bias dengan neuron dan w_{bo} adalah bobot antara unit konstan dengan output sedangkan w_{in} dan w_{no} masing-masing menyatakan bobot koneksi

input dengan neuron dan antara neuron dengan output. Kedua fungsi ψ_n dan ψ_o masing-masing fungsi aktivasi yang digunakan pada neuron dan output. Banyaknya bobot yang diestimasi dari model (1) adalah $\{N \times (p + 2)\} + 1$.

3. Algoritma Genetika

Algoritma genetika merupakan suatu metode penyelesaian yang tergolong heuristik. Proses evolusi pada algoritma genetika meliputi proses seleksi, perkawinan silang (*crossover*) dan mutasi terhadap sekelompok individu dalam satu populasi untuk menghasilkan keturunan yang lebih baik berdasarkan seleksi alam. Secara prinsip, algoritma ini mempunyai beberapa karakteristik diantaranya bekerja dengan himpunan pengkodean parameter, bukan himpunan parameter itu sendiri; mencari dari suatu populasi titik-titik, bukan hanya dari satu titik saja; menggunakan informasi fungsi sasaran (*fitness function*), tidak menggunakan derivatif serta menggunakan operasi random dengan aturan perubahan probabilistik, bukan operasi dengan aturan derivatif tertentu dalam setiap iterasi. Algoritma genetik bekerja dengan mengkodekan kandidat solusi suatu masalah dalam bentuk kromosom yang masing-masing mempunyai panjang sama dan terdiri dari elemen-elemen berupa simbol-simbol dari himpunan terpilih, misalkan alphabet biner. Setiap kromosom x berkorespondensi dengan fungsi fitness $f(x)$. Terpilihnya panjang kromosom, alphabet, dan pengkodean yang merupakan pemetaan dari himpunan Ω (himpunan dengan semesta pembicaraan tertentu) atas himpunan kromosom-kromosom disebut sebagai pola representasi (*representation scheme*) untuk suatu permasalahan [12]. Tahap selanjutnya adalah menginisialisasi populasi awal $P(0)$ yang diawali dengan merepresentasikan solusi ke dalam kromosom,

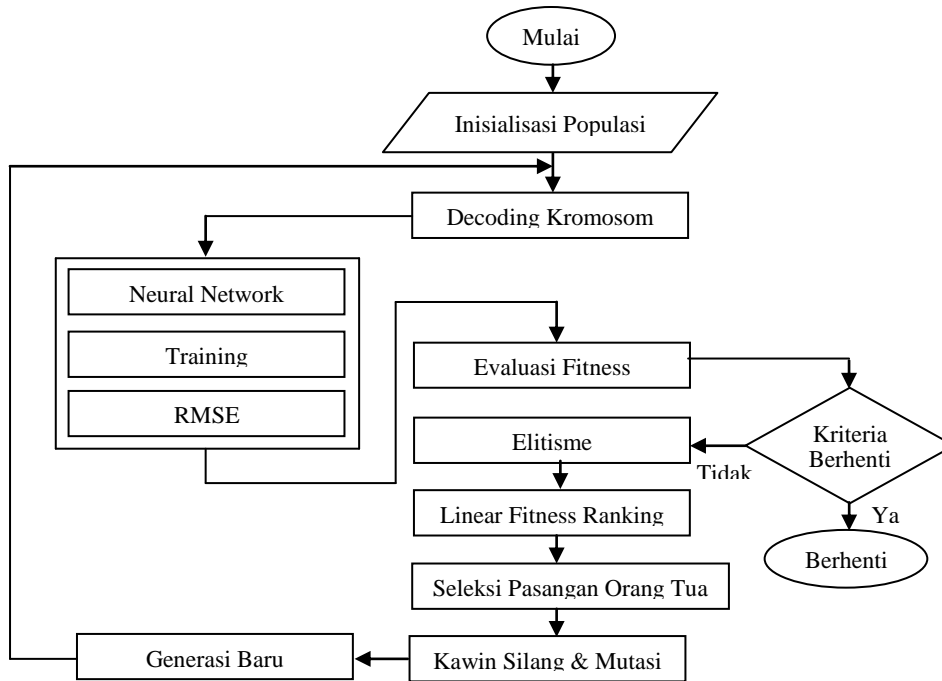
Tingkatan pertama dalam algoritma genetika digunakan operasi seleksi yaitu dengan membentuk himpunan mating pool $M(k)$ yang banyak elemennya sama dengan banyaknya elemen pada $P(k)$. Setiap titik $m^{(k)} \in M^{(k)}$ diambil dari titik-titik $x^{(k)} \in P(k)$. Tingkatan kedua berupa evolusi yaitu digunakan operasi perkawinan silang (*crossover*) dan mutasi. Operasi perkawinan silang mengambil sepasang kromosom sebagai orang tua yang menghasilkan sepasang kromosom keturunan. Pasangan orang tua dipilih dari $M(k)$ secara random dengan probabilitas yang dinyatakan dengan p_c . Setelah perkawinan silang selesai dilanjutkan dengan proses mutasi dengan mengubah nilai dari satu atau beberapa gen dalam satu kromosom. Proses mutasi terjadi secara random dengan probabilitas yang sangat kecil yaitu $p_m \ll 1$. Setelah proses seleksi, perkawinan silang dan mutasi dilakukan terhadap populasi dalam satu generasi, tahap selanjutnya adalah memilih kromosom yang dapat bertahan hidup untuk generasi selanjutnya. Suatu strategi bantuan yang baik untuk digunakan adalah elitisme, yaitu dengan menyimpan beberapa kromosom yang baik dalam populasi sehingga tetap terpelihara pada generasi berikutnya. Strategi lain yang berguna adalah Linear Fitness Ranking (LFR), yaitu melakukan penskalaan nilai fitness yang diperoleh dari evaluasi individu. LFR dilakukan untuk menghindari kecenderungan konvergen pada solusi optimal lokal dengan diperoleh nilai fitness baru yang memiliki variansi yang lebih besar. Langkah-langkah algoritma genetika dapat disederhanakan menjadi sebagai berikut [12] :

- (i) Bentuk populasi awal (set : $k = 0 \rightarrow P(0)$)
- (ii) Evaluasi $P(k)$
- (iii) Jika kriteria berhenti terpenuhi maka berhenti.
- (iv) Seleksi $M(k)$ dari $P(k)$
- (v) Susun (evolusi) $M(k)$ ke bentuk $P(k + 1)$
- (vi) Kembali ke langkah 2 (set : $k = k + 1$)

4. Metodologi Penelitian

Misalkan dipunyai data time series X_t dan akan dibangun model FFNN sebagaimana pada persamaan (1) dengan input lag 1 sampai dengan lag p ($X_{t-1}, X_{t-2}, \dots, X_{t-p}$), ψ_n dan ψ_o

masing-masing adalah fungsi logistik sigmoid $f(x) = 1/(1 + \exp(-x))$ dan fungsi linear $y = x$, dengan jumlah neuron N . Selanjutnya akan dicari estimasi terhadap bobot jaringan $\mathbf{w} = (w_{bn}, w_{bo}, w_{in}$ dan $w_{no})$ yaitu $\hat{\mathbf{w}} = (\hat{w}_{bn}, \hat{w}_{bo}, \hat{w}_{in}$ dan $\hat{w}_{no})$ menggunakan algoritma genetika yang meminimumkan nilai RMSE. Beberapa kriteria yang diatur dalam penggunaan algoritma genetika diantaranya adalah metode seleksi, teknik kawin silang, probabilitas kawin silang dan mutasi, serta jumlah generasi. Garis besar pemecahan masalah tersebut digambarkan pada diagram alir berikut.



Gambar 2 Diagram alir garis besar pemecahan masalah

5. Hasil Simulasi

Dipunyai data time series harian nilai tukar (kurs jual) US Dolar terhadap Rupiah dari tanggal 5 Januari 2009 sampai 1 Juni 2012 sebanyak 841 data dengan 800 data untuk training dan 41 sebagai data testing. Dibangun model FFNN sebagaimana pada persamaan (1) dengan fungsi aktivasi ψ_n dan ψ_o , masing-masing adalah fungsi logistik sigmoid dan fungsi linear dan jumlah neuron dipilih sebanyak 4. Pemilihan input model menggunakan bantuan fungsi autokorelasi dan autokorelasi parsial dan yang terpilih adalah lag 1. Kemudian dilakukan overfitting dengan penambahan lag sebagai perbandingan sehingga terpilih tiga macam input yaitu lag 1 saja (X_{t-1}), lag 1 dan lag 2 (X_{t-1}, X_{t-2}) serta lag 1, lag 2 dan lag 3 ($X_{t-1}, X_{t-2}, X_{t-3}$). Selanjutnya dilakukan pengolahan data menggunakan Matlab 7.6.0 (R2008a) dengan metode optimasi algoritma genetika dan Levenberg-Marquardt sebagai perbandingan masing-masing dengan sepuluh kali pengulangan. Dilakukan juga percobaan dengan model *hybrid* yaitu menggunakan algoritma genetika untuk mendapatkan bobot awal dan pencarian bobot akhir dilanjutkan dengan metode Levenberg-Marquardt. Pada teknik pencarian menggunakan algoritma genetika dipilih ukuran populasi 20, panjang gen 10, probabilitas kawin silang $p_c = 0.7$, probabilitas mutasi $p_m = 0.1$, metode seleksi roulette wheel serta jumlah generasi 25000 sedangkan pada metode Levenberg-Marquardt dipilih jumlah epoch sebanyak 10000.

Pemilihan jumlah iterasi (generasi pada algoritma genetika dan epoch pada metode Levenberg-Marquardt) berbeda karena perbedaan lama proses dari kedua algoritma. Pada model hybrid jumlah generasi untuk mendapatkan bobot awal adalah 1000 dan jumlah epoch adalah 10000. Hasil simulasi disajikan pada tabel 1.

Tabel 1 Hasil simulasi perbandingan algoritma genetika, Levenberg-Marquardt dan metode hybrid pada model FFNN untuk data Kurs Dolar terhadap Rupiah.

Metode Optimasi	Lag Input	Nilai RMSE Data Training				Nilai RMSE Data Testing			
		Mean	Var	Min	Max	Mean	Var	Min	Max
AG	1	52.8614	0.0013	52.8274	52.9181	47.3652	0.0249	47.1374	47.5280
	1,2	52.4330	0.0279	52.1776	52.5848	48.2200	0.0171	48.0701	48.4098
	1,2,3	52.2784	0.1731	51.9826	53.0002	48.7607	0.0288	48.5574	49.0275
LM	1	52.3799	0.0120	52.2716	52.6829	46.5195	0.2363	46.2975	47.6868
	1,2	50.9414	0.1675	50.3296	51.4465	49.0290	0.4853	47.4283	49.9098
	1,2,3	49.9323	0.1754	49.1963	50.4167	51.3593	1.2770	49.4831	53.3238
HYBRID AG-LM	1	52.0134	0.1756	51.7386	52.7157	47.1104	0.0198	46.8538	47.3168
	1,2	50.7612	0.1593	50.3767	51.6181	48.5782	0.5967	46.8762	49.0836
	1,2,3	50.2010	0.0020	50.1374	50.2497	48.9908	0.4778	48.4430	50.8330

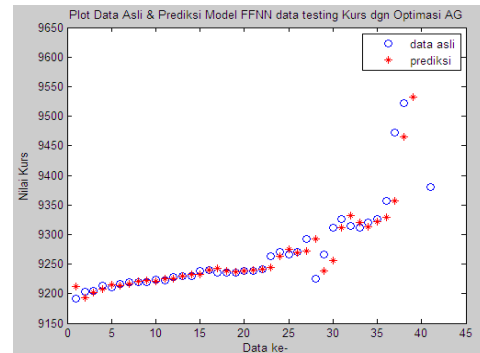
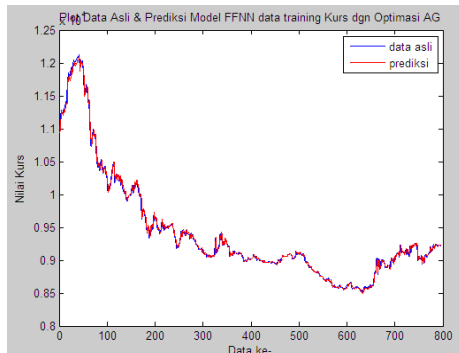
Berdasarkan hasil perhitungan pada tabel 1 nampak bahwa secara umum ketiga metode tidak menghasilkan perbedaan error yang signifikan. Nilai RMSE data training pada algoritma genetika lebih tinggi dari metode Levenberg-Marquardt tetapi pada data testing lebih rendah. Metode hybrid menghasilkan nilai RMSE data training yang hampir sama dengan metode Levenberg-Marquardt dan relatif lebih baik untuk data testing. Berdasarkan hasil tersebut maka model terbaik yang dipilih adalah model yang menggunakan lag satu saja sebagai input. Hasil lain yang menarik untuk diamati adalah variansi dari nilai RMSE dengan algoritma genetika lebih kecil dari hasil perhitungan menggunakan metode Levenberg-Marquardt baik untuk data training maupun data testing. Hal ini menunjukkan bahwa algoritma genetika mampu menghasilkan nilai prediksi yang lebih stabil dibanding metode Levenberg-Marquardt. Begitu pula dengan metode hybrid yang menghasilkan variansi error yang lebih kecil dibanding hasil dari metode Levenberg-Marquardt. Pada tabel 2 berikut disajikan statistik dari nilai RMSE untuk gabungan ketiga macam input pada masing-masing metode optimasi.

Tabel 2 Statistik dari nilai RMSE pada masing-masing metode optimasi

Metode Optimasi	Nilai RMSE Data Training				Nilai RMSE Data Testing			
	mean	var	min	max	mean	var	min	max
AG	52.5242	0.0674	52.3292	52.8344	48.1153	0.0236	47.9216	48.3218
LM	51.0845	0.1183	50.5992	51.5154	48.9693	0.6662	47.7363	50.3068
Hybrid	50.9919	0.1123	50.7509	51.5278	48.2265	0.3648	47.3910	49.0778

Dari tabel 2 nampak bahwa algoritma genetika memberikan nilai variansi paling kecil yang berarti merupakan metode paling baik dari sisi kestabilan hasil estimasi, sedangkan metode *hybrid* menghasilkan nilai variansi yang lebih kecil dari metode Levenberg-Marquardt terutama pada data testing yang berarti Levenberg-Marquardt

merupakan metode yang paling tidak stabil dari ketiga teknik optimasi. Pada gambar 3a dan 3b berikut disajikan plot data asli dan prediksi untuk data training maupun testing pada salah satu proses menggunakan algoritma genetika.



Gambar 3a Plot data asli dan prediksi model FFNN data training dengan optimasi algoritma genetika

Gambar 3a Plot data asli dan prediksi model FFNN data testing dengan optimasi algoritma genetika

Hasil plot pada gambar 3a dan 3b menunjukkan hasil prediksi yang secara visual relatif baik dimana data asli dan prediksinya saling berdekatan. Beberapa kali eksperimen yang dilakukan menghasilkan pola visual prediksi data training dan data testing yang relatif tidak berubah. Berikut disajikan hasil peramalan untuk 12 tahap ke depan dari masing-masing metode.

Tabel 3 Hasil peramalan 12 tahap ke depan dari ketiga metode yang digunakan

No	Metode LM			Metode Hybrid			AG		
	Lag Input			Lag Input			Lag Input		
	1	1,2	1,2,3	1	1,2	1,2,3	1	1,2	1,2,3
1	9.374,2	9.448,8	9.403,8	9.378,8	9.470,2	9.724,0	9.374,9	9.586,3	9.624,1
2	9.368,6	9.432,7	9.440,5	9.377,6	9.388,5	9.600,9	9.369,8	9.392,5	9.566,2
3	9.363,2	9.426,1	9.424,3	9.376,3	9.453,3	9.545,2	9.364,8	9.563,5	9.405,7
4	9.358,1	9.419,3	9.421,9	9.375,1	9.391,6	9.695,9	9.359,8	9.402,1	9.620,0
5	9.353,1	9.412,5	9.415,9	9.373,9	9.442,4	9.624,2	9.354,8	9.543,9	9.529,8
6	9.348,3	9.405,7	9.410,6	9.372,6	9.392,9	9.566,9	9.349,9	9.409,4	9.429,8
7	9.343,8	9.398,9	9.405,3	9.371,3	9.434,2	9.670,0	9.345,0	9.526,9	9.608,8
8	9.339,3	9.392,2	9.400,1	9.370,0	9.393,2	9.634,0	9.340,2	9.414,6	9.502,4
9	9.335,1	9.385,5	9.395,1	9.368,7	9.427,5	9.566,5	9.335,4	9.512,0	9.450,8
10	9.331,0	9.378,8	9.390,1	9.367,4	9.392,9	9.649,0	9.330,6	9.418,2	9.593,2
11	9.327,0	9.372,2	9.385,2	9.366,0	9.421,7	9.636,4	9.325,9	9.498,8	9.482,7
12	9.323,2	9.365,6	9.380,3	9.364,7	9.392,2	9.560,5	9.321,2	9.420,3	9.467,8

Hasil peramalan dari ketiga metode optimasi memberikan nilai yang hampir sama terutama pada model dengan input lag 1, masing-masing dengan kecenderungan menurun sedangkan pada input sampai lag 2 dan lag 3 nilai peramalan lebih variatif tetapi tetap mempunyai kecenderungan menurun.

6. Penutup

Hasil simulasi model FFNN pada data kurs USD terhadap rupiah menunjukkan kekuatan algoritma genetika lebih pada kestabilan hasil estimasi sementara nilai errornya belum lebih baik dari metode Levenberg-Marquardt. Kelemahan utama dari algoritma genetika terletak pada lamanya proses iterasi sehingga dalam satu proses memerlukan banyak sekali generasi untuk mencapai konvergen. Metode hybrid perlu dikembangkan dengan memanfaatkan kestabilan hasil dari algoritma

genetika dengan jumlah generasi yang tidak terlalu banyak untuk mendapatkan nilai awal dari metode standar seperti Levenberg-Marquardt. Perlu dikembangkan pula metode hybrid yang lain dengan menggabungkan algoritma genetika dengan metode standar lain disertai dengan penentuan arsitektur model FFNN yang optimal untuk mendapatkan nilai error yang lebih kecil.

Daftar Pustaka

- [1] Banakar, A. and Azeem, M.F., 2006, Generalized Wavelet Neural Network Model and its Application in Time Series Prediction, *International Joint Conference on Neural Networks, Vancouver, BC, Canada, July 16-21*
- [2] Gholizadeh, M.H., and Darand, M., 2009, Forecasting Precipitation with Artificial Neural Networks (Case Study : Tehran), *Journal of Applied Sciences* 9 (9) 1786-1790
- [3] Sit, C.W., 2005, Application of Artificial Neural Network – Genetic Algorithm in Inferential Estimation and Control of a Distillation Column, *thesis of Master of Engineering (Chemical)*, Faculty of Chemical and Natural Resources Engineering, Universiti Teknologi Malaysia
- [4] Makvandi, P., Jassbi, J., and Khanmohammadi, S., 2005, Application of Genetic Algorithm and Neural Network in Forecasting with Good Data, *Proceedings of the 6th WSEAS Int. Conf. on NEURAL NETWORKS*, Lisbon, Portugal, June 16-18 (pp56-61)
- [5] Dondeti, S., Kannan, K., and Manavalan, R., 2005, Genetic Algorithm Optimized Neural Networks Ensemble for Estimation of Mefenamic Acid and Paracetamol in Tablets, *Acta Chim. Slov.* 52, 440–449
- [6] Giovanis, E., 2010, Application of Feed-Forward Neural Networks Autoregressive Models with Genetic Algorithm in Gross Domestic Product Prediction, *World Academy of Science, Engineering and Technology* 64
- [7] Jose, T.J., and Mythili, P., 2009, Neural Network and Genetic Algorithm Based Hybrid Model for Content Based Mammogram Image Retrieval, *Journal of Applied Sciences* 9(19): 3531-3538, ISSN 1812-5654, Asian Network for Scientific Information
- [8] Neruda, R., Srejber, J., Neruda, M., and Pascenko, P., 2010, Combining neural networks and genetic algorithms for hydrological flow forecasting, *Geophysical Research Abstracts*, Vol. 12, EGU2010-9099, EGU General Assembly 2010
- [9] Qureshi, I.M., Cheema, T.A., Naveed, A., and Jalil, A., 2003, Genetic Algorithms Based Artificial Neural Networks for Blur Identification and Restoration of Degraded Images, *Pakistan Journal of Information and Technology* 2(1): 21-24
- [10] Suchar, R., Schalling, B., Ciocoiu, I., Brezulianu, A., and Bonciu, C., 2000, Noise Analysis and Modelling with Neural Networks and Genetic Algorithms, *Seoul 2000 FISITA World Automotive Congress F2000H184*, June 12-15, Seoul
- [11] Tselentis, G.A and Vladutu, L., 2010, An Attempt to Model the Relationship Between Mmi Attenuation and Engineering Ground-Motion Parameters Using Artificial Neural Networks and Genetic Algorithms, *Nat. Hazards Earth Syst. Sci.*, 10, 2527–2537, Received: 23 July 2010 – Accepted: 13 October 2010 – Published: 7 December 2010
- [12] Widodo dan Widiyantoro, 2003, Analisis Algoritma Genetik dan Aplikasinya dalam Penyelesaian Masalah Optimisasi Fungsi Tanpa Kendala, *Conference on Statistical and Mathematical Sciences of Islamic Society in South East Asia Region*, Universitas Islam Bandung Tanggal 25-26 April 2003

LAMPIRAN LISTING PROGRAM

1. Program NN dengan metode optimasi Levenberg Marquardt

```
function nnloop1lag(tes, looping, periode)
dt=xlsread('kursdolarbaru.xlsx'); dat=dt/10000;
data=dat(1:length(dat)-(tes+1));
input1=data(1:length(data)-1); target=data(2:length(data));
data_test=dat(length(dat)-tes:length(dat));
input_test=data_test(1:length(data_test)-1);
target_test=data_test(2:length(data_test));
rmse_training=0; rmse_testing=0;
min_rmse_train=99999; max_rmse_train=0;
min_rmse_test=99999; max_rmse_test=0;
j=looping;
for i=1:j
net=newff([min(input1)max(input1)],[4 1],{'logsig' 'purelin'},'trainlm');
net.trainParam.epochs=10000;
net = train(net,input1,target);
Y = sim(net,input1); e=10000*(Y-target);
mse_train=mse(e);rmse_train=sqrt(mse_train);
rmsee_train=rmse_train;
if rmsee_train<min_rmse_train
min_rmse_train=rmsee_train;
nil_i=i;
end
if rmsee_train>max_rmse_train
max_rmse_train=rmsee_train;
nil_ii=i;
end
rmse_training=rmse_training+rmse_train;
rata_train=rmse_training/j;
rmse_train_step(i)=rmse_train;
test=sim(net,input_test);
e_test=10000*(target_test-test);
mse_test=mse(e_test);
rmse_test=sqrt(mse_test);
rmsee_test=rmse_test;
if rmsee_test<min_rmse_test
min_rmse_test=rmsee_test;
nil_m=i;
end
if rmsee_test>max_rmse_test
max_rmse_test=rmsee_test;
nil_mm=i;
end
rmse_testing=rmse_testing+rmse_test;
rata_test=rmse_testing/j;
rmse_test_step(i)=rmse_test;
end
mean_rmse_training=mean(rmse_train_step)
var_rmse_training=var(rmse_train_step)
minimum_rmse_train=min_rmse_train
maksimum_rmse_train=max_rmse_train
mean_rmse_testing=mean(rmse_test_step)
var_rmse_testing=var(rmse_test_step)
minimum_rmse_test=min_rmse_test
maksimum_rmse_test=max_rmse_test
data_test=data_test';
for k=1: periode;
input_fore2(k)=data_test(length(data_test));
input_forecast=input_fore2(k);
ramal(k)=sim(net,input_forecast);
data_test=[data_test;ramal(k)];
end
peramalan=10000*ramal'
x=1:length(target); targett=10000*target; YY=10000*Y;
figure(2)
plot(x,targett,'r-',x,YY,'b-')
title('Plot data asli & prediksi data training model NN')
legend('data asli','prediksi')
x1=1:length(target_test);targett_test=10000*target_test;testt=10000*test;
figure(3)
plot(x1,targett_test,'ro',x1,testt,'b*')
title('Plot data asli & prediksi data testing model NN')
legend('data asli','prediksi')
```


2. Program NN untuk time series dengan optimasi Algoritma Genetika

```

function gann4hiddenllag(test,looping,periode)
data_asli=xlsread('kursdolarbaru.xlsx');
data=data_asli/10000; n=length(data); y=data(2:n-test,1);
x1=data(1:n-test-1,1); yy=data(n-test+1:n,1); x11=data(n-test:n-1,1);
a=3*ones(13);batas=a(1,:);
for i=1:looping
    OPTIONS=gaoptimset('CreationFcn',@gacreationlinearfeasible,
    'SelectionFcn',@selectionroulette,'CrossoverFraction',0.7,'PlotFcns',
    @gaplotbestf,'Generations',25000,'CrossoverFcn',@crossoverheuristic,
    'TolFun',1e-20,'Display','final');
    [bt,FVAL,EXITFLAG,OUTPUT,POPULATION]=ga(@(b) fungsinn4lag1(y,x1,b),13,
    [],[],[],[],-batas,batas,[],OPTIONS);
    bobot_akhir=bt;
    wb1=bt(1); wb2=bt(2); wb3=bt(3); wb4=bt(4); w11=bt(5); w21=bt(6);
    w31=bt(7); w41=bt(8); v=bt(9); v11=bt(10); v12=bt(11); v13=bt(12);
    v14=bt(13);
    f_in1=wb1+w11.*x1; f_in2=wb2+w21.*x1;
    f_in3=wb3+w31.*x1; f_in4=wb4+w41.*x1;
    f_out1=1./(1+exp(-f_in1)); f_out2=1./(1+exp(-f_in2));
    f_out3=1./(1+exp(-f_in3)); f_out4=1./(1+exp(-f_in4));
    y_out=v+v11.*f_out1+v12.*f_out2+v13.*f_out3+v14.*f_out4;
    ytopi=10000.*y_out; y_asli=10000*y;
    sse = sum((y_asli-ytopi).^2); err=y_asli-ytopi;
    ms_e=mse(err); rms_e=sqrt(ms_e); rmse_training(i)=rms_e;
    mape_training(i)=100.*(sum(abs(y_asli-ytopi)./y_asli))/length(y_asli);
    f_in1_test=wb1+w11.*x11; f_in2_test=wb2+w21.*x11;
    f_in3_test=wb3+w31.*x11; f_in4_test=wb4+w41.*x11;
    f_out1_test=1./(1+exp(-f_in1_test)); f_out2_test=1./(1+exp(-f_in2_test));
    f_out3_test=1./(1+exp(-f_in3_test)); f_out4_test=1./(1+exp(-f_in4_test));
    y_out_test=v+v11.*f_out1_test+v12.*f_out2_test+v13.*f_out3_test+v14.*f_out4_test;
    y_test=10000.*y_out_test; yy_asli=10000*yy;
    sse_test = sum((yy_asli-y_test).^2); err_test=yy_asli-y_test;
    mse_test=mse(err_test); rmse_test=sqrt(mse_test);
    mape_test(i)=100.*(sum(abs(yy_asli-y_test)./yy_asli))/length(yy_asli);
    rmse_testing(i)=rmse_test;
end
rmse_training_lengkap=rmse_training
rmse_testing_lengkap=rmse_testing
rata_rmse_train=mean(rmse_training)
varian_rmse_train=var(rmse_training)
minimum_rmse_train=min(rmse_training)
maksimum_rmse_train=max(rmse_training)
rata_rmse_test=mean(rmse_testing)
varian_rmse_test=var(rmse_testing)
minimum_rmse_test=min(rmse_testing)
maksimum_rmse_test=max(rmse_testing)
%Membuat plot
x=1:length(y); xx=1:length(yy);
figure(3)
plot(x,y_asli,'b-',x,ytopi,'r-')
legend('data asli','prediksi'); xlabel('Data ke-'); ylabel('Nilai Kurs')
title('Plot Data Asli & Prediksi FFNN data train dgn Optimasi AG')
figure(4)
plot(xx,yy_asli,'bo',xx,y_test,'r*')
legend('data asli','prediksi'); xlabel('Data ke-'); ylabel('Nilai Kurs')
title('Plot Data Asli & Prediksi FFNN data testing dgn Optimasi AG')
for k=1:periode;
    input_fore(k)=yy(length(yy));
    f_in1_fore(k)=wb1+w11.*input_fore(k);
    f_in2_fore(k)=wb2+w21.*input_fore(k);
    f_in3_fore(k)=wb3+w31.*input_fore(k);
    f_in4_fore(k)=wb4+w41.*input_fore(k);
    f_out1_fore(k)=1./(1+exp(-f_in1_fore(k)));
    f_out2_fore(k)=1./(1+exp(-f_in2_fore(k)));
    f_out3_fore(k)=1./(1+exp(-f_in3_fore(k)));
    f_out4_fore(k)=1./(1+exp(-f_in4_fore(k)));
    ramal(k)=v+v11.*f_out1_fore(k)+v12.*f_out2_fore(k)+v13.*f_out3_fore(k)
    +v14.*f_out4_fore(k); yy=[yy;ramal(k)];
end
peramalan=10000*ramal'

function fitnes = fungsinn4lag1(y,x1,b)
wb1=b(1); wb2=b(2); wb3=b(3); wb4=b(4); w11=b(5); w21=b(6); w31=b(7);
w41=b(8); v=b(9); v11=b(10); v12=b(11); v13=b(12); v14=b(13);
f_in1=wb1+w11.*x1; f_in2=wb2+w21.*x1; f_in3=wb3+w31.*x1; f_in4=wb4+w41.*x1;
f_out1=1./(1+exp(-f_in1)); f_out2=1./(1+exp(-f_in2));
f_out3=1./(1+exp(-f_in3)); f_out4=1./(1+exp(-f_in4));
y_out=v+v11.*f_out1+v12.*f_out2+v13.*f_out3+v14.*f_out4;
fitnes = sum((y-y_out).^2);

```

3. Program NN untuk time series dengan optimasi Hybrid AG-LM

```
function nn4hiddenawalga(test,looping,periode)
data_asli=xlsread('kursdolarbaru.xlsx');
dataasli=data_asli(:,1);
data=dataasli/10000;
n=length(data); y=data(2:n-test,1);
x1=data(1:n-test-1,1); yy=data(n-test+1:n,1); x11=data(n-test:n-1,1);
a=3*ones(13);batas=a(1,:);
for p=1:looping
    OPTIONS=gaoptimset('CreationFcn',@gacreationlinearfeasible,'SelectionFcn',
    @selectionroulette,'CrossoverFraction',0.7,'PlotFcns',@gaplotbestf,
    'Generations',1000,'CrossoverFcn',@crossoversinglepoint,'TolFun',1e-10,
    'Display','final');
    [bt,FVAL,EXITFLAG,OUTPUT,POPULATION] = ga(@(b) fungsinn4lag1(y,x1,b),13,
    [],[],[],[],-batas,batas,[],OPTIONS);
    bobot_akhir=bt;
    wb1=bt(1);wb2=bt(2); wb3=bt(3); wb4=bt(4); w11=bt(5); w21=bt(6);w31=bt(7);
    w41=bt(8); v=bt(9); v11=bt(10); v12=bt(11); v13=bt(12); v14=bt(13);
    f_in1=wb1+w11.*x1; f_in2=wb2+w21.*x1;
    f_in3=wb3+w31.*x1; f_in4=wb4+w41.*x1;
    f_out1=1./(1+exp(-f_in1)); f_out2=1./(1+exp(-f_in2));
    f_out3=1./(1+exp(-f_in3)); f_out4=1./(1+exp(-f_in4));
    y_out=v+v11.*f_out1+v12.*f_out2+v13.*f_out3+v14.*f_out4;
    ytopi=10000.*y_out;
    err=y-ytopi; ms_e=mse(err); rms_e=sqrt(ms_e);
    y_asli=10000*y;
    mape = 100.*(sum(abs(y_asli-ytopi)./y_asli))/length(y_asli);
    f_in1_test=wb1+w11.*x11; f_in2_test=wb2+w21.*x11;
    f_in3_test=wb3+w31.*x11; f_in4_test=wb4+w41.*x11;
    f_out1_test=1./(1+exp(-f_in1_test));
    f_out2_test=1./(1+exp(-f_in2_test));
    f_out3_test=1./(1+exp(-f_in3_test));
    f_out4_test=1./(1+exp(-f_in4_test));
    y_out_test = v + v11.*f_out1_test + v12.*f_out2_test + v13.*f_out3_test +
    v14.*f_out4_test;
    y_test=10000.*y_out_test; yy_asli=10000*yy;
    err_test=yy_asli-y_test; mse_test=mse(err_test);
    rmse_test=sqrt(mse_test);
    mape_test = 100*(sum(abs(yy_asli-y_test)./yy_asli))/length(yy_asli);
    input1=[x1]; target=y; input_test=[x11];
    net = newff([minmax(input1)],[4 1],{'logsig' 'purelin'},'trainlm');
    net.trainParam.epochs=10000;
    net.IW{1,1}=[w11 w21 w31 w41]';
    net.b{1,1}=[wb1 wb2 wb3 wb4]';
    net.LW{2,1}=[v11 v12 v13 v14];
    net.b{2,1}=v;
    net = train(net,input1,target);
    Y = sim(net,input1); e=10000.*(target-Y);
    mse_train=mse(e); rmse_train=sqrt(mse_train);
    ytest=sim(net,input_test); e_test=10000.*(yy-ytest);
    mse_test=mse(e_test); rmse_test=sqrt(mse_test);
    rmse_training(p)=rmse_train; rmse_testing(p)=rmse_test;
end
rata_rmse_train=mean(rmse_training)
varian_rmse_train=var(rmse_training)
minimum_rmse_train=min(rmse_training)
maksimum_rmse_train=max(rmse_training)
rata_rmse_test=mean(rmse_testing)
varian_rmse_test=var(rmse_testing)
minimum_rmse_test=min(rmse_testing)
maksimum_rmse_test=max(rmse_testing)
x=1:length(target); x1=1:length(yy);
figure(2)
plot(x,10000*target,'r-',x,10000*Y,'b-')
title('Plot data asli & prediksi data training model NN dg optimasi Hybrid')
legend('data asli','prediksi')
figure(3)
plot(x1,10000*yy,'ro',x1,10000*ytest,'b*')
title('Plot data asli & prediksi data testing model NN dg optimasi Hybrid ')
legend('data asli','prediksi')
for k=1:periode;
    input_fore(k)=yy(length(yy));
    input_forecast=[input_fore(k)];
    ramal(k)=sim(net,input_forecast);
    yy=[yy;ramal(k)];
end
peramalan=10000*ramal'
```