



Implementasi Algoritma Neural Network Berbasis Particle Swarm Optimazation untuk Penentuan Kredit

Syaifur Rahmatullah

Prodi Teknik Informatika, STMIK Nusa Mandiri, Jakarta, Indonesia

Email: syaifur.syl@nusamandiri.ac.id

Email Penulis Korespondensi: syaifur.syl@nusamandiri.ac.id

Abstrak—Kredit macet merupakan salah satu resiko kredit yang dihadapi oleh pelaku industri keuangan dan perbankan. Kredit macet terjadi apabila dalam jangka panjang, lembaga keuangan atau perbankan tidak dapat menarik pinjaman kredit dalam waktu yang telah ditentukan. Kredit macet memiliki dampak yang buruk bagi penyedia kredit yaitu berupa resiko kerugian. Tentu saja hal ini tidak boleh dibiarkan berlarut-larut dan harus dicari jalan keluarnya. Namun untuk menjamin akurasi dalam penentuan kelayakan kredit diperlukan suatu algoritma yang akurat. Neural Network merupakan sebuah algoritma yang bisa memetakan dan mengklasifikasi inputan data serta dapat digunakan untuk proses prediksi dari beberapa variabel yang di inputkan ke dalam algoritma ini. Untuk dapat menambah akurasi dalam prediksi variabel – variabel penentu kelayakan kredit maka algoritma ini di optimasi dengan Particle Swarm Optimazation. Hasil penelitian berupa confusion matrix membuktikan akurasi Neural Network berbasis Particle Swarm Optimazation memiliki tingkat akurasi sebesar 96,67% dan hasil AUC sebesar 0.965.

Kata Kunci: Kredit, Neural Network, Particle Swarm

Abstract—Bad credit is one of the credit risks faced by financial and banking industry players. Bad credit occurs if in the long run, financial institutions or banks cannot withdraw credit loans within the allotted time. Bad credit has a bad impact for credit providers in the form of risk of loss. Of course this should not be allowed to drag on and must be resolved. However, to guarantee accuracy in determining creditworthiness, an accurate algorithm is needed. Neural Network is an algorithm that can map and classify data input and can be used for the prediction of several variables that are input into this algorithm. To be able to add accuracy in predicting the determinants of credit worthiness, this algorithm is optimized with Particle Swarm Optimazation. The results of the study in the form of confusion matrix prove that the accuracy of Neural Networks based on Particle Swarm Optimazation has an accuracy rate of 96.67% and AUC results of 0.965.

Keywords: Credit, Neural Network, Particle Swarm

1. PENDAHULUAN

Data dari Bank Indonesia, kegiatan terbesar dalam sebuah perbankan adalah perkreditan. Perkreditan pun tidak luput dari resiko. Resiko kredit merupakan salah satu resiko yang harus di ambil oleh bank dalam memberikan kredit kepada debitur, resiko ini timbul karena terjadi debitur yang tidak bisa memenuhi janjinya dalam masa kredit[1]. resiko kredit semakin besar maka rasio *Non Performing Loan* (npl) akan semakin besar[2], hal tersebut sangat berbahaya terhadap suatu bank karena akan mengurangi jumlah modal di bank tersebut[3]. Untuk meminimalisir resiko tersebut di perlukan metode atau algoritma yang dapat dijadikan pedoman untuk pengambilan keputusan pemberian kredit kepada debitur. Dalam penelitian sebelumnya telah di lakukan beberapa implementasi algoritma untuk meminimalisir resiko kredit diantaranya :

1. Penelitian yang dilakukan oleh adi sucipto dengan judul prediksi kredit macet melalui perilaku nasabah pada koperasi simpan pinjam dengan menggunakan metode algoritma klasifikasi c4.5, dalam penelitiannya beliau menggunakan algoritma c4.5 dengan mengevaluasi AUC, ROC dan T-Test dengan rapidminer dan menghasilkan tingkat akurasi sebesar 91,06%[4].
2. Penelitian yang dilakukan M. Husni Rifqo dan Ardi Wijaya dengan judul implementasi algoritma *naïve bayes* dalam penentuan pemberian kredit. Dalam penelitiannya mereka menggunakan beberapa dataset yang di ambil dari UCI dataset untuk di training menggunakan *naïve bayes* dan melakukan perbandingan terhadap *accuracy* dari setiap dataset untuk membuktikan kehadandalan algoritma *naïve bayes* dalam menganalisa penentuan kredit[5].
3. Penelitian yang dilakukan Henny Leidiyana dengan judul penerapan algoritma *k-nearest neighbor* untuk penentuan resiko kredit kepemilikan kendaraan bermotor. Dalam penelitiannya beliau menggunakan algoritma *k-nearest neighbor* serta *cross validation* dengan mengevaluasi AUC, ROC dan *Confusion Matrix*. Penelitian ini menghasilkan tingkat akurasi sebesar 81,46 % untuk proses penentuan pemberian kredit[6].
4. Penelitian yang di lakukan Achmad Rifai dan Rizky Aulianita dengan judul Komparasi Algoritma Klasifikasi C4.5 dan *Naïve Bayes* Berbasis *Particle Swarm Optimization* Untuk Penentuan Resiko Kredit. Dalam penelitiannya mereka melakukan komparasi 2 algoritma dengan di optimasi *particle swarm optimization*. Hasil dari penelitian ini adalah terjadi peningkatan akurasi setelah di lakukan optimasi oleh *particle swarm optimization* untuk penentuan kredit[7].

Dari ke empat penelitian diatas dapat dilihat ada beberapa algoritma yang bisa di gunakan untuk proses penentuan pemberian kredit. Algoritma – algoritma yang dipakai tentu menghasilkan nilai *accuracy* yang berbeda sesuai dengan atribut yang dipakai dalam proses penelitian.



Seleksi fitur merupakan salah satu cara untuk meningkatkan *accuracy* dari suatu algoritma. Teknik seleksi fitur digunakan untuk menyeleksi dan mengurangi fitur/atribut yang tidak relevan[8].

Dalam hal ini peneliti akan melakukan penerapan metode *neural network* yang akan di optimasi oleh metode seleksi fitur *particle swarm optimization* untuk mengetahui tingkat *accuracy* algoritma ini untuk proses penentuan pemberian kredit.

2. METODOLOGI PENELITIAN

2.1 Tahapan Penelitian

Tahapan penelitian yang digunakan dalam penelitian kali ini disusun dengan menggunakan algoritma Neural Network dan di optimasi dengan *particle swarm optimization* untuk mendapatkan hasil yang akurat, tahapannya adalah sebagai berikut :



Gambar 1. Tahapan penelitian

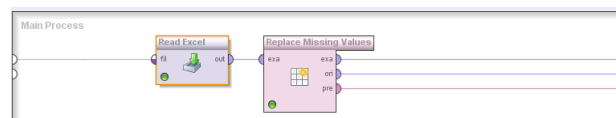
2.2 Instrumen Penelitian

Dalam penelitian kali ini penulis menggunakan data primer nasabah bank yang telah melakukan pengajuan kredit di Bank DKI. Data nasabah kredit diambil dari salah satu Bank milik pemerintah daerah yaitu Bank DKI Jakarta Cabang Bendungan Hilir pada tahun 2018 sampai 2019, data yang akan digunakan adalah data para nasabah kredit yang berjumlah 632 data debitur, dari data yang di kumpulkan selama setahun tersebut ditemukan 97 debitur yang mengalami kredit macet. Dari Analisa sementara data tersebut ditemukan beberapa atribut yang di jadikan variable dalam penelitian kali ini, diantaranya akun bank, lama pinjaman, jaminan, jumlah kredit, jumlah angsuran, umur, pekerjaan, masa kerja, gaji, persentase pemotongan gaji, jenis kelamin, status, pinjaman sebelumnya, response. Nilai dari atribut tersebut ada yang merupakan nilai kategorikal diantaranya akun bank, jaminan, pekerjaan, masa kerja, jenis kelamin, status, pinjaman sebelumnya.

2.3 Data Preparation

Untuk menghasilkan data yang berkualitas dari 632 data yang telah di dapatkan dilakukan beberapa Teknik preprocessing, yaitu [9]

- Data validation, untuk mengidentifikasi dan menghapus data yang ganjil (outlier/noise), data yang tidak konsisten, dan data yang tidak lengkap (missing value)



Gambar 2. Data validation menggunakan aplikasi rapidminer.

- Data integration & transformation, dilakukan untuk meningkatkan akurasi dan efisiensi, data yang berupa kategorikal di transformasi ke dalam bentuk angka sebagai berikut :

Tabel 1. Katergori Atribut

No	Atribut	Nilai	Keterangan
1	akun_bank	1	Tidak Memiliki akun bank
		2	Memiliki akun bank
2	lama_pinjam		dalam bulan
3	Jaminan	1	Surat - Surat Berharga
		2	Rumah
		3	Tanah
		4	Lain lain
4	Jumlah Kredit		Jumlah kredit
5	Angsuran		Jumlah angsuran
6	umur		umur dalam tahun
7	pekerjaan	1	Pensiunan
		2	PNS
		3	Karyawan Swasta
		4	Wiraswasta



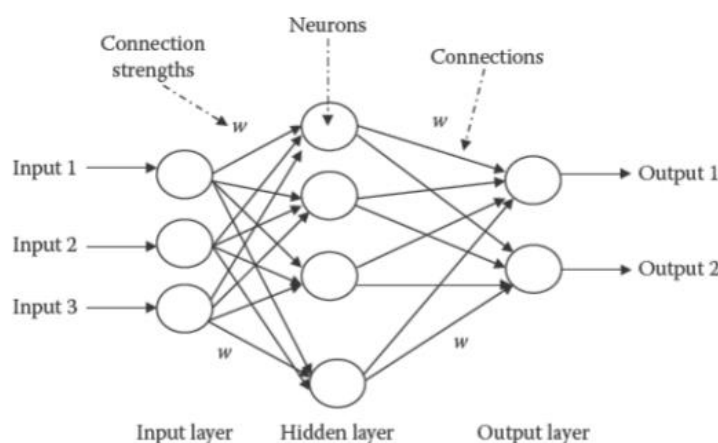
No	Atribut	Nilai	Keterangan
8	masa_kerja	5	guru
		1	< 1 tahun
		2	1 - 4 tahun
		3	4 - 7 tahun
		4	> 7 tahun
9	Gaji		Jumlah gaji
10	persentase _pemotongan_gaji	1	<10%
		2	10 - 20%
		3	20 - 30%
		4	30 - 40 %
		5	40 - 50 %
		6	> 50 %
11	Jenis_kelamin	1	Pria
		2	Wanita
12	Status	1	Single
		2	Menikah
		3	Berceraai
13	pinjaman sebelumnya	1	tidak ada pinjaman
		2	ada pinjaman (belum lunas)
		3	ada pinjaman (sudah lunas)
14	Response	1	Lancar
		2	Bermasalah

2.4 Cross Validation

Cross validation merupakan cara untuk menemukan parameter terbaik dari satu model dengan menguji besarnya error pada data test. Setiap kelas data set harus diwakili proporsi yang tepat antara data training dan data testing. Data dibagi secara acak pada masing-masing kelas dengan perbandingan yang sama. Seluruh proses training dan testing diulang berkali-kali dengan sampel yang berbeda untuk mengurangi bias yang disebabkan oleh sampel tertentu. Tingkat kesalahan pada iterasi yang berbeda akan dihitung rata-ratanya untuk menghasilkan error rate[10]

2.5 Algoritma Neural Network

Neural Network (NN) atau Jaringan Syaraf Tiruan (JST) adalah suatu usaha untuk meniru fungsi otak manusia. Otak manusia diyakini terdiri dari jutaan unit pengolahan kecil, yang disebut *neuron*, yang bekerja secara paralel. *Neuron* saling terhubung satu sama lain melalui koneksi *neuron*. setiap individu *neuron* mengambil *input* dari satu set *neuron*. Ini kemudian memproses *input* tersebut dan melewati *output* untuk satu set *neuron*. Keluaran dikumpulkan oleh *neuron* lain untuk diproses lebih lanjut. Para otak manusia adalah jaringan kompleks *neuron* dimana koneksi tetap melanggar dan membentuk. Banyak model mirip dengan otak manusia telah diusulkan[11]



Gambar 3. Arsitektur Artificial Neural Network[11]

Setiap node (neuron) dalam *neural network* merupakan sebuah unit pemrosesan. Tiap node memiliki beberapa masukan dan sebuah keluaran. Setiap node mengkombinasikan beberapa nilai masukan, melakukan kalkulasi, dan membangkitkan nilai keluaran (aktifasi). Dalam setiap node terdapat dua fungsi, yaitu fungsi untuk mengkombinasikan masukan dan fungsi aktifasi untuk menghitung keluaran. Terdapat beberapa metode untuk



mengkombinasikan masukan antara lain *weighted sum*, *mean*, *max*, logika OR, atau logika AND (Kusrini & Luthfi, 2009). Serta beberapa fungsi aktivasi yang dapat digunakan yaitu *heaviside* (threshold), *step activation*, *piecewise*, *linear*, *gaussian*, *sigmoid*, *hyperbolic tangent* [12]

Salah satu keuntungan menggunakan *neural network* adalah bahwa *neural network* cukup kuat sehubungan dengan data. Karena *neural network* berisi banyak node (*neuron* buatan) dengan bobot ditugaskan untuk setiap koneksi [11]. Algoritma *neural network* mempunyai karakteristik-karakteristik lainnya antara lain [12]

1. Masukan dapat berupa nilai diskrit atau *real* yang memiliki banyak dimensi.
2. Keluaran berupa vektor yang terdiri dari beberapa nilai diskrit atau *real*.
3. Dapat mengetahui permasalahan secara *black box*, dengan hanya mengetahui nilai masukan serta keluarannya saja.
4. Mampu menangani pembelajaran terhadap data yang memiliki derau (*noise*)
5. Bentuk dari fungsi target pembelajaran tidak diketahui karena hanya berupa bobot-bobot nilai masukan pada setiap *neuron*.
6. Karena harus mengubah banyak nilai bobot pada proses pembelajaran, maka waktu pembelajaran menjadi lama, sehingga tidak cocok untuk masalah-masalah yang memerlukan waktu cepat dalam pembelajaran.
7. *Neural network* hasil pembelajaran tiruan dapat dijalankan dengan tepat.

2.6 Particle Swarm Optimazation

Particle swarm optimization (PSO) dapat diasumsikan dengan sekelompok burung yang secara acak mencari makanan di suatu daerah. Hanya ada satu potong makanan di daerah yang dicari tersebut. Burung-burung tidak tahu di mana makanan tersebut. Tapi mereka tahu seberapa jauh makanan tersebut dan posisi rekan-rekan mereka. Jadi strategi terbaik untuk menemukan makanan adalah dengan mengikuti burung yang terdekat dari makanan [13]. Untuk menemukan solusi yang optimal, maka setiap partikel akan bergerak ke arah posisi yang terbaik sebelumnya (*pbest*) dan posisi terbaik secara global (*gbest*). Sebagai contoh, partikel ke-*i* dinyatakan sebagai : $x_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$ dalam ruang *d*-dimensi. Posisi terbaik sebelumnya dari partikel ke-*i* disimpan dan dinyatakan sebagai $pbest_i = (pbest_{i,1}, pbest_{i,2}, \dots, pbest_{i,d})$. Modifikasi kecepatan dan posisi tiap partikel dapat dihitung menggunakan kecepatan saat ini dan jarak $pbest_i$ ke $pbest_d$ seperti ditunjukkan oleh persamaan berikut [13]:

$$v_{i,m} = w \cdot v_{i,m} + c_1 * R * (pbest_{i,m} - x_{i,m}) + c_2 * R * (gbest_m - x_{i,m}) \quad (1)$$

$$x_{i,d} = x_{i,m} + v_{i,m} \quad (2)$$

Keterangan :

<i>n</i>	: jumlah partikel dalam kelompok
<i>d</i>	: dimensi
$v_{i,m}$	: kecepatan partikel ke- <i>i</i> pada iterasi ke- <i>i</i>
<i>w</i>	: faktor bobot inersia
<i>c1, c2</i>	: konstanta akselerasi (<i>learning rate</i>)
<i>R</i>	: bilangan random (0-1)
$x_{i,d}$	: posisi saat ini dari partikel ke- <i>i</i> pada iterasi ke- <i>i</i>
<i>pbest_i</i>	: posisi terbaik sebelumnya dari partikel ke- <i>i</i>
<i>gbest</i>	: partikel terbaik diantara semua partikel dalam satu kelompok atau populasi

Persamaan (1) menghitung kecepatan baru untuk tiap partikel (solusi potensial) berdasarkan pada kecepatan sebelumnya ($V_{i,m}$), lokasi partikel dimana nilai fitness terbaik telah dicapai (*pbest*), dan lokasi populasi global (*gbest* untuk versi global, *lbest* untuk versi local) atau local neighborhood pada algoritma versi local dimana nilai fitness terbaik telah dicapai.

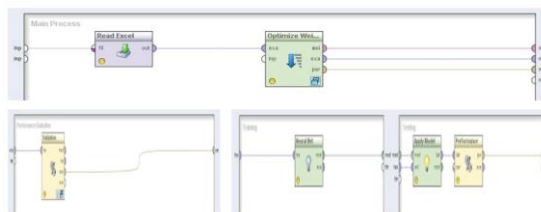
Persamaan (2) memperbaharui posisi tiap partikel pada ruang solusi. Dua bilangan acak *c1* dan *c2* dibangkitkan sendiri. Penggunaan berat inersia *w* telah memberikan performa yang meningkat pada sejumlah aplikasi.

3. HASIL DAN PEMBAHASAN

Pada Penelitian ini menggunakan Metode K-fold Cross Validation untuk membagi data Training dan Testing. Dan pada penelitian ini menggunakan 90 % dataset untuk metode pelatihan dan 10 % dataset akan digunakan untuk metode pengujian.

3.1 Algoritma Neural Network berbasis particle swarm optimization

Pada penelitian prediksi kelayakan pemberian kredit menggunakan algoritma *neural network* berbasis Particle Swarm Optimization. Particle Swarm Optimization memiliki pencarian kinerja yang lebih baik untuk memecahkan banyak masalah optimasi dengan lebih cepat dan tingkat konvergensi yang stabil.



Gambar 4. Desain model Algoritma *Neural Network* Berbasis *Particle Swarm Optimazation*

Berdasarkan hasil dari uji coba dengan menggunakan algoritma neural network maka data training tersebut akan diseleksi dengan attribute yang digunakan yaitu Akun_bank, lama_pinjam, jamina, jumlah_kredit, jumlah_angsuran, umur, pekerjaan, masa_kerja, gaji, persentase_pemotongan_gaji, jenis_kelamin, status, pinjaman_sebelumnya dan 1 atribut sebagai label yaitu response.

Tabel 2. Bobot Atribut *Neural Network* berbasis PSO

Attribute	Weight
Akun_bank	0.084
Lama_pinjam	0.318
jaminan	0.753
jumlah_kredit	0
jumlah_angsuran	0
umur	0.007
pekerjaan	0.463
masa_kerja	0.595
gaji	0.990
persentase_pemotongan_gaji	0.875
jenis_kelamin	0.121
status	0
pinjaman_sebelumnya	0

Dari hasil uji coba menggunakan algoritma *neural network* berbasis particle swarm optimization diperoleh beberapa atribut-atribut yang berpengaruh terhadap bobot atribut yaitu: akun_bank, lama_pinjam, jaminan, umur, pekerjaan, masa_kerja, gaji, persentase_pemotongan_gaji, jenis_kelamin. Berdasarkan hasil uji coba jumlah_kredit, jumlah_angsuran, status, pinjaman_sebelumnya tidak berpengaruh terhadap bobot attribute. Berdasarkan hasil model algoritma *neural network* berbasis particle swarm optimization dapat dilakukan uji coba pada nilai training cycles ditentukan dengan cara memasukkan nilai range dari 100 sampai dengan 1500 untuk training cycles, serta nilai 0.3 untuk learning rate dan 0.2 untuk momentum dengan menggunakan data training. Berikut ini adalah hasil dari percobaan yang telah dilakukan untuk penentuan nilai training cycles:

Tabel 3. Percobaan Penentuan Nilai *Training Cycles Neural Network* berbasis PSO

Training Cycles	Learning Rate	Momentum	NN		NN+PSO	
			Accuracy	AUC	Accuracy	AUC
100	0,3	0,2	89,55	0,938	91,62	0,942
200	0,3	0,2	89,7	0,938	92,41	0,945
300	0,3	0,2	90,97	0,942	92,56	0,939
400	0,3	0,2	90,97	0,941	93,36	0,945
500	0,3	0,2	92,23	0,941	93,99	0,946
600	0,3	0,2	92,71	0,942	93,99	0,946
700	0,3	0,2	92,77	0,945	95,57	0,956
800	0,3	0,2	93,03	0,945	95,25	0,956
900	0,3	0,2	93,18	0,943	94,93	0,951
1000	0,3	0,2	93,03	0,945	95,25	0,943
1100	0,3	0,2	93,18	0,945	95,41	0,955
1200	0,3	0,2	93,02	0,944	95,26	0,946
1300	0,3	0,2	93,65	0,943	95,73	0,972
1400	0,3	0,2	93,34	0,943	95,89	0,956
1500	0,3	0,2	93,65	0,945	95,89	0,961

Berdasarkan hasil uji coba pada nilai training cycles dipilih berdasarkan nilai akurasi dan nilai Area Under Curve (AUC) terbesar yang dihasilkan, dengan nilai 1500 pada *training Cycles*. Kemudian nilai tersebut akan



digunakan untuk percobaan selanjutnya yaitu untuk menentukan nilai *Learning Rate*. Nilai *Learning Rate* ditentukan dengan cara melakukan dengan uji coba memasukkan nilai dengan range 0.1 sampai dengan 0.9. Nilai *Training cycles* dipilih dari percobaan sebelumnya yaitu 1500, sedangkan 0.2 digunakan untuk nilai momentum. Berikut adalah hasil dari eksperimen yang telah dilakukan untuk menentukan nilai *Learning Rate*.

Tabel 4. Percobaan Penentuan nilai *Learning Rate* dengan *Neural Network* berbasis PSO

<i>Learning Rate</i>	<i>Training Cycles</i>	<i>Momentum</i>	NN		NN+PSO	
			<i>Accuracy</i>	AUC	<i>Accuracy</i>	AUC
0,1	1500	0,2	92,34	0,948	95,41	0,956
0,2	1500	0,2	92,65	0,947	96,20	0,957
0,3	1500	0,2	92,65	0,945	95,89	0,961
0,4	1500	0,2	93,29	0,949	95,58	0,938
0,5	1500	0,2	92,97	0,951	96,20	0,945
0,6	1500	0,2	92,5	0,943	96,04	0,95
0,7	1500	0,2	92,66	0,940	95,59	0,951
0,8	1500	0,2	93,44	0,957	95,41	0,955
0,9	1500	0,2	92,96	0,950	95,57	0,950

Berdasarkan hasil uji coba pada nilai *Learning Rate* dipilih berdasarkan nilai akurasi dan nilai *Area Under Curve* (AUC) terbesar yang dihasilkan, dengan nilai 0.5 pada *Learning Rate* dan 1500 pada *training Cycles*. Kemudian nilai tersebut akan digunakan untuk percobaan selanjutnya yaitu untuk menentukan nilai *Momentum*. Nilai *momentum* ditentukan dengan cara melakukan eksperimen dengan cara memasukkan nilai *training cycles* 1500 dan *learning rate* 0.5, dan memasukkan nilai *momentum* dengan range 0.1 sampai dengan 0.9. Dengan nilai *training cycles* 1500 dan *learning rate* 0.5 berdasarkan percobaan sebelumnya. Berikut adalah hasil dari percobaan yang telah dilakukan untuk penentuan nilai *momentum*:

Tabel 5. Percobaan Penentuan nilai *Momentum* dengan *Neural Network* berbasis PSO

<i>Momentum</i>	<i>Training Cycles</i>	<i>Learning Rate</i>	NN		NN + PSO		
			<i>Accuracy</i>	AUC	<i>Learning Rate</i>	<i>Accuracy</i>	AUC
0,0	1500	0,8	92,34	0,930	0,5	95,26	0,935
0,1	1500	0,8	92,81	0,947	0,5	96,20	0,953
0,2	1500	0,8	93,44	0,957	0,5	96,20	0,945
0,3	1500	0,8	93,6	0,959	0,5	95,42	0,953
0,4	1500	0,8	92,02	0,926	0,5	95,89	0,958
0,5	1500	0,8	91,71	0,937	0,5	95,57	0,952
0,6	1500	0,8	91,7	0,928	0,5	96,20	0,943
0,7	1500	0,8	89,66	0,896	0,5	94,94	0,951
0,8	1500	0,8	88,22	0,772	0,5	94,47	0,941
0,9	1500	0,8	92,34	0,795	0,5	91,62	0,871

Berdasarkan hasil uji coba diatas, maka untuk parameter *neural network* dipilih nilai 1500 untuk nilai *training cycles*, 0.5 untuk nilai *learning rate* dan 0.2 untuk nilai *gfmomentum*. Langkah berikutnya adalah menentukan jumlah *hidden layer* dan *neuron sizes* untuk mencari nilai akurasi dan AUC yang paling tinggi. Pada jumlah *hidden layer* sebanyak 1, dilakukan percobaan pada *neuron size* dengan range 1 sampai dengan 25.

Tabel 6. Penentuan *neuron size* NN - PSO

<i>Hidden Layer</i>	NN		NN + PSO	
	<i>Accuracy</i>	AUC	<i>Accuracy</i>	AUC
1	88,99	0,912	91,18	0,919
2	88,52	0,9	90,89	0,918
3	86,93	0,898	91,5	0,925
4	90,24	0,896	91,83	0,909
5	90,26	0,906	90,26	0,912
6	91,06	0,919	90,39	0,91
7	91,05	0,927	91,26	0,902
8	92,15	0,924	93,88	0,933
9	93,1	0,929	94,2	0,915
10	92	0,918	93,58	0,92
11	92,15	0,912	94,05	0,928
12	92,31	0,914	94,36	0,921



<i>Hidden Layer</i>	<i>NN Accuracy</i>	<i>NN AUC</i>	<i>NN + PSO Accuracy</i>	<i>NN + PSO AUC</i>
13	91,85	0,914	94,67	0,929
14	92,47	0,907	94,21	0,926
15	92,15	0,913	94,51	0,918
16	92,95	0,929	94,68	0,926
17	93,1	0,91	94,21	0,93
18	93,26	0,925	96,67	0,965
19	93,73	0,914	94,85	0,933
20	93,58	0,926	94,84	0,93
21	92,95	0,917	94,63	0,923
22	93,83	0,944	93,83	0,944
23	92,94	0,921	94,67	0,933
24	93,42	0,919	94,68	0,93
25	93,41	0,929	94,84	0,932

Berdasarkan tabel perbandingan tingkat kelancaran kredit antara model *Neural Network* dengan *Neural Network* berbasis PSO. Hasil terbaik pada percobaan diatas adalah dengan satu *hidden layer* dengan *size* 18 dengan accuracy yang dihasilkan sebesar 96,67% dan AUC-nya 0.965

3.2 Evaluasi hasil algoritma *Neural Network* berbasis *particle swarm optimization* dengan *Confusion* dan *Kurva ROC*

Akurasi *Neural Network* berbasis PSO 96,67 % . dimana Jumlah True Positive (TP) adalah 526 , Untuk False Negative (FN) adalah 12 , untuk False Positive (FP) adalah 9 dan Untuk True Negative (TN) adalah 85

☒ Table View ☐ Plot View			
accuracy: 96.67% +/- 2.51% (mikro: 96.68%)			
	true Lancar	true Bermasalah	class precision
pred. Lancar	526	12	97.77%
pred. Bermasalah	9	85	90.43%
class recall	98.32%	87.63%	

Gambar 4. Hasil Akurasi Algoritma *Neural Network* berbasis PSO

Dari hasil diatas dapat di hitung untuk mencari nilai *accuracy*, *sensitivity*, *specificity*, *ppv*, dan *npv* hasilnya pada persamaan dibawah ini:

$$Acc = \frac{tp+tn}{tp+tn+fp+fn}$$

$$acc = \frac{526+85}{526+85+9+12}$$

$$Sensitivity = \frac{tp}{tp+fn}$$

$$Sensitivity = \frac{526}{526+12}$$

$$Specificity = \frac{tn}{tn+fp}$$

$$Specificity = \frac{85}{85+9}$$

$$PPV = \frac{tp}{tp+fp}$$

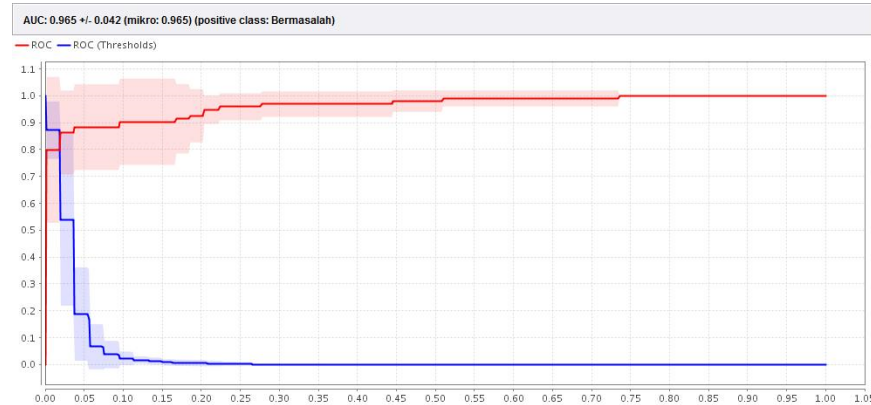
$$PPV = \frac{526}{526+9}$$

$$NPV = \frac{tn}{tn+fn}$$

$$NPV = \frac{85}{85+12}$$

Tabel 7. *Accuracy*, *Sensitivity*, *Specificity*, *ppv*, dan *npv* *Neural Network* berbasis PSO

	Nilai (%)
<i>Accuracy</i>	96,67
<i>Sensitivity</i>	97,76
<i>Specificity</i>	90,42
<i>PPV</i>	98,31
<i>NPV</i>	87,62



Gambar 5. Nilai AUC *Neural Network* berbasis PSO

Grafik ROC dengan nilai AUC (Area Under Curve) untuk algoritma *Support Vector Machine* sebesar 0.932 dengan tingkat diagnosa *Excellent Classification*

4. KESIMPULAN

Dari penelitian menggunakan algoritma *neural network* berbasis *particle swarm optimization* dengan menggunakan data nasabah perkreditasi di Bank DKI Jakarta, dapat diambil beberapa kesimpulan diantaranya :

1. Algoritma *Neural Network* menghasilkan nilai akurasi 93,83 % dan nilai AUC 0,944. Algoritma *Neural Network* berbasis PSO menghasilkan nilai akurasi 96,67 % dan nilai AUC 0,965, hal tersebut dapat membuktikan bahwa optimasi *particle swarm optimization* dapat meningkatkan akurasi dari algoritma *neural network*.
2. Dari hasil penelitian ini juga didapatkan bahwa parameter yang paling berpengaruh terhadap kelancaran seseorang nasabah dalam melaksanakan kredit adalah dari parameter gaji, karena memiliki bobot yang paling tinggi yaitu 0,990

Dengan hasil penelitian tersebut dapat dilanjutkan dengan membuat prototipe aplikasi sistem penunjang keputusan sebagai tahapan dari persetujuan dalam pengambilan kredit di Bank, untuk lebih meningkatkan akurasi dapat juga dilakukan penelitian lain dengan mengoptimasi *neural network* dengan algoritma optimasi lain seperti *genetic algorithm*, *chi square* dan lain sebagainya untuk mendapatkan algoritma yang paling akurat dalam penentuan kelayakan kredit.

REFERENCES

- [1] I. bankir Indonesia (IBI), *Mengelola Kredit Secara Sehat*, 1st ed. Jakarta: PT Gramedia Pustaka Utama, 2014.
- [2] I. Hariyani, *Restrukturisasi dan Penghapusan Kredit Macet*, Pertama. Jakarta: PT Elex Komputindo, 2010.
- [3] Y. Murdianingsih, "Klasifikasi Nasabah Baik Dan Bermasalah Menggunakan Metode Naive Bayes," *J. Inf.*, vol. 2015, no. November, pp. 349–356, 2015, doi: 10.1007/3-540-61794-9_66.
- [4] A. Sucipto, "Pada Koperasi Simpan Pinjam Dengan Menggunakan," *J. DISPROTEK*, vol. 6, no. 1, pp. 75–87, 2015.
- [5] M. H. Rifqo and A. Wijaya, "Implementasi Algoritma Naive Bayes Dalam Penentuan Pemberian Kredit," *Pseudocode*, vol. 4, no. 2, pp. 120–128, 2017, doi: 10.33369/pseudocode.4.2.120-128.
- [6] H. Leidiyana, "Penerapan Algoritma K-Nearest Neighbor Untuk Penentuan Resiko Kredit Kepemilikan Kendaraan Bermotor," *J. Penelit. Ilmu Komputer, Syst. Embed. Log.*, vol. 1, no. 1, pp. 65–76, 2013.
- [7] A. Rifai and R. Aulianita, "Komparasi Algoritma Klasifikasi C4.5 dan Naive Bayes Berbasis Particle Swarm Optimization Untuk Penentuan Resiko Kredit," *J. speed- sentra Penelit. engineering dan edukasi*, vol. 10, no. 2, 2018, [Online]. Available: speed.web.id.
- [8] B. Nurina Sari, "Implementasi Teknik Seleksi Fitur Information Gain Pada Algoritma Klasifikasi Machine Learning Untuk Prediksi Performa Akademik Siswa," *Semin. Nas. Teknol. Inf. dan Multimed.* 2016, pp. 55–60, 2016, [Online]. Available: <http://seminas.amikom.ac.id/document/pdf/1482.pdf>.
- [9] M. Refaat, *Data Preparation for Data Mining Using SAS*, 1st ed. San Fransisco: Elsavier, 2010.
- [10] F. F. Harryanto and S. Hansun, "Penerapan Algoritma C4.5 untuk Memprediksi Penerimaan Calon Pegawai Baru di PT WISE," *J. Tek. Inform. Dan Sist. Inf.*, vol. 3, no. 2, pp. 95–103, 2017, [Online]. Available: <http://jurnal.mdp.ac.id/index.php/jatisi/article/view/71>.
- [11] A. Shukla, R. Tiwari, and R. Kala, *Real Life Applications of Soft Computing*, 1st ed. Boca Raton: CRC Press, 2010.
- [12] F. Gorunescu, *Data Mining: Concepts, Models and Techniques*, 1st ed. India: Springer, 2011.
- [13] D. Palupi Rini, S. Mariyam Shamsuddin, and S. Sophiyati Yuhaziz, "Particle Swarm Optimization: Technique, System and Challenges," *Int. J. Comput. Appl.*, vol. 14, no. 1, pp. 19–27, 2011, doi: 10.5120/1810-2331.