

PENERAPAN NEURAL NETWORK BERBASIS BACKWARD ELIMINATION UNTUK PREDIKSI SUSU SAPI PERAH

Yunita

AMIK BSI Jakarta

Jl.RS. Fatmawati No.24 Jakarta Selatan

yunita.ynt@bsi.ac.id

Abstract – Predicted milk production of dairy cows is one way that can be used to determine the condition of food supplies. Although there has been research on the condition of food, research begins with a data processing milk production of dairy cows to get a time series of data of milk production of dairy cows. The method used is to compare methods of neural networks with methods of artificial neural network-based backward elimination with the input data is the data on the previous year. Several experiments were conducted to obtain the optimal architecture and generate accurate predictions. The results show the method of artificial neural network-based backward elimination produces smaller RMSE 0272 compared to only using the neural network only.

Keywords: Prediction, milk of dairy cow, Neural Network, Backward Elimination

I. PENDAHULUAN

Akurasi prediksi hasil susu dari sapi perah adalah usaha inheren yang sulit. sapi perah sangat tunduk terhadap sejumlah kondisi stokastik (cuaca, kondisi kesehatan hewan dan sebagainya) yang mempengaruhi tingkat produksi susu selama musim menyusui. Kemampuan memprediksi susu sapi perah perhari, minggu dan perbulan dengan hasil yang akurat sangat menguntungkan praktik manajemen pertanian (Murphy dkk, 2012). Selain itu, hal ini sangat membantu dalam mengalokasikan sumber daya seperti pakan, baik untuk individu dan sapi untuk kawanan (Khan dkk, 2005).

Dalam hal memprediksi produksi susu sapi dalam suatu wilayah telah diselesaikan dengan beberapa metode seperti Neural Network (NN) (Murphy dkk, 2012) melakukan prediksi produksi susu sapi perah perhari di wilayah Amerika Utara dengan data yang diambil pada tahun 2010. Autoregressive Integrated Moving Average (ARIMA) (Ahmed dkk, 2011) melakukan peramalan dengan menggunakan ARIMA yang dilatih dan diuji menggunakan data rentet waktu selama sepuluh tahun (1990 s/d 2010) tahun di Pakistan.

Studi ini menunjukkan bahwa Model NN dapat lebih efisien dalam perhitungan cepat dan mampu menangani data yang tidak stabil yang khas dalam kasus data untuk peramalan produksi susu sapi perah. Tetapi NN memiliki kekurangan jika digunakan untuk prediksi jangka menengah (Murphy dkk, 2012).

Neural Network mempunyai kelebihan yaitu jaringan syaraf mampu menyelesaikan problem nonlinear, mempunyai toleransi yang cukup tinggi terhadap data yang mengandung noise dan mampu menangkap hubungan yang sangat kompleks antara variabel-variabel prediktor dan outputnya, namun *Neural Network* juga mempunyai kekurangan yaitu adanya overgeneralisasi, dimana penggunaan jumlah data yang banyak, *Neural Network* melakukan dengan sangat baik untuk data pelatihan, tetapi gagal untuk melakukan dengan baik untuk data validasi. Masalah ini mungkin disebabkan oleh jumlah neuron yang berlebihan, durasi pelatihan yang berlebihan, atau alasan lain yang serupa. Generalisasi yang berlebihan terjadi ketika *Neural Network*, dengan algoritma pelatihan, mulai meniru fungsi yang sangat kompleks di tempat yang sederhana (Shuka, Tiwari & Kala, 2010). *Neural Network* adalah jaringan prosesor sederhana dengan banyak unit, masing-masing mungkin memiliki sejumlah kecil memori lokal. Unit dihubungkan oleh saluran komunikasi yang selalu berupa data angka, dikodekan oleh berbagai sarana. Unit hanya beroperasi hanya pada data lokal dan pada input yang mereka terima melalui koneksi (Pinkus, 1999).

Feature Selection, adalah masalah yang berkaitan erat dengan pengurangan dimensi. Tujuan *Feature Selection* adalah untuk mengidentifikasi fitur dalam kumpulan data yang sama pentingnya, dan membuang semua fitur lain seperti informasi yang tidak relevan dan berlebihan. Karena *feature selection* mengurangi dimensi dari data, sehingga

memungkinkan operasi yang lebih efektif & algoritma data *mining* yang cepat (yaitu data algoritma *mining* dapat dioperasikan lebih cepat dan lebih efektif dengan menggunakan *feature selection*) (Maimon & Rokach, 2010). *Feature selection* adalah salah satu faktor yang paling penting yang dapat mempengaruhi tingkat akurasi klasifikasi (Yuanning Liu, Gang Wang, Huiling Chen, Hao Dong, Xiaodong Zhu & Sujing Wang, 2011). Penelitian menggunakan *feature selection* yang pernah dilakukan oleh Yuanning Liu, Gang Wang, Huiling Chen, Hao Dong, Xiaodong Zhu & Sujing Wang tahun 2011, dimana hasil penilitan menunjukkan bahwa penggunaan metode *Improved feature selection* (IFS) menunjukkan hasil yang signifikan lebih baik daripada tiga metode lain yang digunakan dalam penelitian tersebut (*GA, SVM, PSO*) dalam hal akurasi prediksi dengan *subset feature* yang lebih kecil.

Berdasarkan uraian diatas, maka penulis menggunakan model *Neural Network* berbasis *feature selection*. Model ini akan digunakan untuk memprediksi atau meramalkan produksi susu sapi perah dalam satu bulan berikutnya.

Susu merupakan suatu makanan atau minuman bergizi yang banyak mengandung mineral dan protein. Kebutuhan akan protein dan kalsium per hari akan dapat dipenuhi 25-44% hanya dengan mengkonsumsi susu 2 gelas sehari (Ali Khomsan, 2002). Konsumsi susu penduduk Indonesia masih rendah (7 liter/kap/thn) sedangkan di negara ASEAN mencapai 21 liter/kap/thn.

Setiap peristiwa atau kejadian yang ada di kehidupan sehari – hari pasti akan berubah-ubah. Sebagai contoh, di bidang ekonomi terdapat rata – rata angka penjualan produk yang tidak menetap, indeks harga saham yang selalu berubah dan lain sebagainya. Peramalan diperlukan untuk memperkirakan apa yang akan terjadi sehingga dapat dilakukan tindakan yang tepat terhadap situasi yang terjadi. Dalam peramalan, data yang digunakan merupakan runtun waktu, yaitu rentetan kronologi dari pengamatan – pengamatan yang dilakukan (Hanke, 1998).

Pada GAPOKTAN Cikadu Sukabumi, prediksi produksi susu sapi saat ini masih dilakukan dengan melihat rentet waktu. Dimana peramalan terhadap produksi susu perbulan melihat dari data produksi susu perbulan dalam satu atau dua tahun terakhir yang akan dijadikan target produksi bulan berikutnya.

II. METODOLOGI PENELITIAN

2.1. Neural Network

Neural Network adalah jaringan prosesor sederhana dengan banyak unit, masing-masing mungkin memiliki sejumlah kecil memori lokal. Unit dihubungkan oleh saluran komunikasi yang selalu berupa data angka, dikodekan oleh berbagai sarana. Unit hanya beroperasi hanya pada data lokal dan pada input yang mereka terima melalui koneksi (Pinkus, 1999).

Neural Network (NN), selanjutnya disebut *NN*, awalnya mendapat inspirasi dari sistem jaringan syaraf makhluk hidup. *NN* hadir sebagai alternatif pendekatan konvensional yang biasanya kurang fleksibel terhadap perubahan struktur masalah. *NN* dapat mengatasi beberapa persoalan tanpa mengadakan perubahan drastis terhadap modelnya (Santosa, 2007).

Menurut (Santosa,2007) ada beberapa karakteristik kemampuan otak manusia:

1. Mengingat
2. Menghitung
3. Mengeneralisasi
4. Adaptasi
5. Konsumsi energi yang rendah

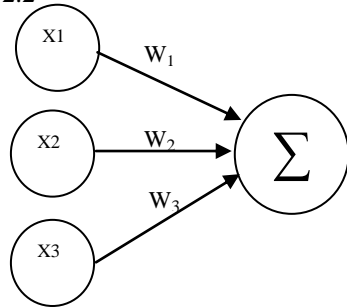
NN berusaha meniru struktur atau arsitektur dan cara kerja otak manusia sehingga mampu menggantikan beberapa pekerjaan manusia. Menurut (Jong Jek Siang, 2009) *NN* dibentuk sebagai generalisasi model matematika dari jaringan syaraf biologi, dengan asumsi bahwa:

- a) Pemrosesan informasi terjadi pada banyak elemen sederhana (neuron),
- b) Sinyal dikirimkan diantara neuro-neuron melalui penghubung-penghubung,
- c) Penghubung antar neuron memiliki bobot yang akan memperkuat atau memperlemah sinyal,
- d) Untuk menentukan output, setiap neuron menggunakan fungsi aktivasi (biasanya bukan fungsi linier) yang dikenakan pada jumlahan input yang diterima. Besarnya output ini selanjutnya dibandingkan dengan suatu batas ambang.

NN ditentukan oleh 3 hal:

- a) Pola hubungan antar neuron(disebut arsitektur jaringan),
- b) Metode untuk menentukan bobot penghubung (disebut metode *training/learning/algoritma*),
- c) Fungsi aktivasi.

Sebagai contoh, perhatikan neuron Y pada gambar 2.2



Sumber : (Jong Jek Siang, 2009)

Gambar 2.1
Contoh Arsitektur Jaringan

Y menerima input dari neuron X_1 , X_2 dan X_3 dengan bobot hubungan masing-masing adalah W_1 , W_2 dan W_3 . Ketiga impuls neuron yang ada dijumlahkan

$$\text{Net} = X_1W_1 + X_2W_2 + X_3W_3$$

Besarnya impuls yang diterima oleh Y mengikuti fungsi aktivasi $y = f(\text{net})$. Apabila nilai fungsi aktivasi cukup kuat, maka sinyal akan diteruskan. Nilai fungsi aktivasi (keluaran model jaringan) juga dapat dipakai sebagai dasar untuk menambah bobot.

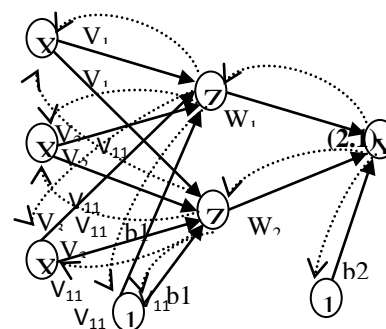
2.1.1. Arsitektur Jaringan Syaraf

Menurut (Jong Jek Siang, 2009) pada dasarnya ada 3 macam arsitektur jaringan syaraf, yaitu:

- Jaringan dengan lapisan tunggal (*single layer net*)
yaitu jaringan dengan lapisan dengan bobot-bobot terhubung. Ciri-ciri dari arsitektur jaringan syaraf dengan lapisan tunggal hanya terdiri satu lapisan input dan satu lapisan output, tanpa lapisan tersembunyi.
- Jaringan syaraf dengan banyak lapisan (*multi layer net*)
Jaringan dengan banyak lapisan memiliki satu atau lebih lapisan yang terletak diantara lapisan input dan lapisan output (memiliki satu atau lebih lapisan tersembunyi).
- Jaringan syaraf dengan lapisan kompetitif (*competitive layer net*)
Arsitektur ini memiliki bentuk yang berbeda, dimana antar neuron dapat saling dihubungkan.

2.1.2. Algoritma Pembelajaran Backpropagation

Backpropagation merupakan algoritma pembelajaran yang terawasi dan *backpropagation* merupakan metode pembelajaran jaringan syaraf tiruan yang paling umum digunakan dan bekerja melalui proses secara iteratif dengan menggunakan sekumpulan contoh data (data *training*), membandingkan nilai prediksi dari jaringan dengan setiap contoh data. Dalam setiap proses, bobot relasi dalam jaringan dimodifikasi untuk meminimalkan nilai *Mean Square Error (MSE)* antara nilai prediksi dari jaringan syaraf tersebut dilakukan dalam arah mundur, dari output layer hingga layer pertama dari *hidden layer* sehingga metode ini disebut *backpropagation* (Jong Jek Siang, 2009; kusrini dan luthfi, 2009).



Sumber : (Kusumadewi & Hartati, 2010)

Gambar 2.2
Arsitektur Jaringan Backpropagation

Pada gambar 2.3 terdiri atas 3 unit (neuron) pada lapisan input yaitu X_1 , X_2 dan X_3 ; 1 lapisan tersembunyi dengan 2 neuron, yaitu Z_1 dan Z_2 ; serta 1 unit lapisan output, yaitu Y. Bobot yang menghubungkan X_1 , X_2 dan X_3 dengan neuron pertama pada lapisan tersembunyi, adalah V_{11} , V_{21} dan V_{31} (V_{ij} : bobot yang menghubungkan neuron input ke-i ke neuron ke-j pada lapisan tersembunyi). b_{11} dan b_{12} adalah bobot bias menuju ke neuron pertama dan kedua pada lapisan tersembunyi. Bobot yang menghubungkan Z_1 dan Z_2 dengan neuron pada lapisan output adalah W_1 dan W_2 . Bobot bias b_2 menghubungkan lapisan tersembunyi dengan lapisan output. Fungsi aktivasi yang digunakan, antara lapisan input dan lapisan tersembunyi, dan antara lapisan tersembunyi dengan lapisan output adalah fungsi aktivasi sigmoid biner.

Dalam *backpropagation*, fungsi aktivasi yang dipakai harus memenuhi beberapa syarat yaitu: kontinu, terdiferensial dengan mudah dan merupakan fungsi tidak turun, salah satu fungsi yang memenuhi ketiga syarat tersebut sehingga sering dipakai adalah

fungsi sigmoid biner yang memiliki range (0,1)

$$f'(x) = f(x)(1 - f(x))$$

$$f(x) = \frac{1}{1 + e^{-x}} \quad \text{dengan turunan}$$

fungsi lainya yaitu fungsi sigmoid bipolar dengan range (-1,1)

$$f'(x) = (f(x)(1 - f(x)))/2$$

$$f(x) = \frac{2}{1 + e^{-x}} - 1 \quad \text{dengan turunan}$$

Fungsi sigmoid memiliki nilai maksimum = 1, maka untuk pola yang targetnya >1, pola masukan dan keluaran harus terlebih dahulu ditranformasi sehingga semua polanya memiliki range yang sama seperti fungsi sigmoid yang dipakai. Alternatif lain adalah menggunakan fungsi aktivasi sigmoid hanya pada layar yang bukan layar keluaran. Pada layar keluaran, fungsi aktivasi yang dipakai adalah fungsi identitas:

$$f(x) = x$$

Pelatihan *backpropagation* meliputi tiga fase, yaitu:

1. Fase pertama adalah fase maju dimana pola masukan dihitung maju mulai dari layar masukan hingga layar keluaran menggunakan fungsi aktivasi yang ditentukan.
2. Fase kedua adalah fase mundur, dimana selisih antara keluaran jaringan dengan target yang diinginkan merupakan kesalahan yang terjadi. Kesalahan tersebut dipropagasikan mundur, dimulai dari garis yang berhubungan langsung dengan unit-unit diayar keluaran.
3. Fase ketiga adalah modifikasi bobot untuk menurunkan kesalahan yang terjadi. Ketiga fase diatas diulang-ulang terus hingga kondisi penghentian dipenuhi (jumlah iterasi atau kesalahan).

Langkah pembelajaran dalam algoritma *backpropagation* adalah sebagai berikut (Myatt, 2010):

1. Inisialisasi bobot jaringan secara acak (biasanya antara -0.1 sampai 1.0)
2. Untuk setiap data pada data *training*, hitung input untuk simpul berdasarkan nilai input dan bobot jaringan saat itu, menggunakan rumus:

$$Input_j = \sum_{i=1}^n O_i w_{ij} + \theta_j$$

Keterangan:

O_i = Output simpul i dari layer sebelumnya

w_{ij} = bobot relasi dari simpul i pada layer sebelumnya ke simpul j

θ_j = bias (sebagai pembatas)

3. Berdasarkan input dari langkah dua, selanjutnya membangkitkan output untuk simpul menggunakan fungsi aktivasi sigmoid:

$$Output = \frac{1}{1 + e^{-Input}}$$

4. Hitung nilai *Error* antara nilai yang diprediksi dengan nilai yang sesungguhnya menggunakan rumus:

$$Error_j = Output_j \cdot (1 - Output_j) \cdot (Target_j - Output_j)$$

Keterangan:

$Output_j$ = Output aktual dari simpul j

$Target_j$ = Nilai target yang sudah diketahui pada data *training*

5. Setelah nilai *Error* dihitung, selanjutnya dibalik ke layer sebelumnya (*backpropagated*). Untuk menghitung nilai *Error* pada *hidden layer*, menggunakan rumus:

$$Error_j = Output_j(1 - Output_j) \sum_{k=1}^n Error_k w_{jk}$$

Keterangan:

$Output_j$ = Output aktual dari simpul j

$Error_k$ = error simpul k

w_{jk} = Bobot relasi dari simpul j ke simpul k pada layer berikutnya

6. Nilai *Error* yang dihasilkan dari langkah sebelumnya digunakan untuk memperbaiki bobot relasi menggunakan rumus:

$$w_{ij} = w_{ij} + l \cdot Error_j \cdot Output_i$$

Keterangan:

w_{ij} = bobot relasi dari unit i pada layer sebelumnya ke unit j

l = *learning rate* (konstanta, nilainya antara 0 sampai dengan 1)

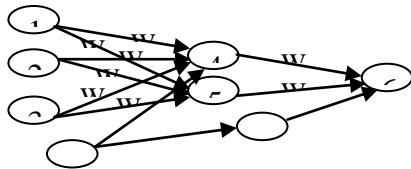
$Error_j$ = Error pada output layer simpul j

$Output_i$ = Output dari simpul i

2.1.3. Studi Kasus Penerapan Algoritma Backpropagation

Berikut contoh kasus penyelesaian masalah menggunakan neural network dengan metode *backpropagation* (Han & Kamber, 2007):

Sebagai contoh sebuah jaringan syaraf tiruan digambarkan sebagai berikut:



Gambar 2.3
Arsitektur Contoh neural network multilayer feed-forward

Inisialisasikan nilai awal dan nilai bias. Misalkan nilai *learning rate* (*l*) adalah 0.9 dan inisialisasi nilai awal dan nilai bias, pada training data pertama $X=(1,0,1)$, yang kelas label nya adalah 1.

Hitung nilai input, output dan *error* untuk setiap *node*.

Rumus perhitungan input:

$$\begin{aligned} \text{Input 4} &= (X_1 \cdot W_{11}) + (X_2 \cdot W_{21}) + (X_3 \cdot W_{31}) + b1 \\ &= (1 \cdot 0.2) + (0 \cdot 0.4) + (1 \cdot -0.5) + - \\ &0.4 \\ &= 0.2 + 0 + (-0.5) + (-0.4) \\ &= -0.7 \end{aligned}$$

Tabel 2.2.
Inout nilai bias, bobot dan bias

X_1	X_2	X_3	W_{11}	W_{12}	W_{21}	W_{22}	W_{31}	W_{32}	W_{41}	W_{51}	$b1$	$b2$	$b3$
1	0	1	0.2	-0.3	0.1	0.4	-0.5	0.2	-0.3	-0.2	-0.4	0.2	0.1

$$\text{Output} = \frac{1}{1 + e^{-\text{Input}}}$$

$$\begin{aligned} \text{Output 4} &= 1/(1 + e^{0.7}) \\ &= 1/(1 + 2.71828^{0.7}) \\ &= 0.332 \end{aligned}$$

$$\begin{aligned} \text{Input 5} &= (X_1 \cdot W_{12}) + (X_2 \cdot W_{22}) + (X_3 \cdot W_{32}) + b2 \\ &= (1 \cdot (-0.3)) + (0 \cdot 0.1) + (1 \cdot 0.2) + \\ &0.2 \\ &= (-0.3) + 0 + 0.2 + 0.2 \\ &= 0.1 \end{aligned}$$

$$\begin{aligned} \text{Output 5} &= 1/(1 + e^{0.1}) \\ &= 1/(1 + 2.71828^{0.1}) \\ &= 0.525 \end{aligned}$$

$$\begin{aligned} \text{Input 6} &= (\text{output 4} \cdot W_{41}) + (\text{output 5} \cdot W_{51}) + b3 \\ &= (0.332 \cdot -0.3) + (0.525 \cdot -0.2) + \\ &0.1 \\ &= -0.105 \end{aligned}$$

$$\begin{aligned} \text{Output 6} &= 1/(1 + e^{-0.105}) \\ &= 1/(1 + 2.71828^{-0.105}) \\ &= 0.474 \end{aligned}$$

Rumus perhitungan *error*:

$$\text{Error}_j = \text{Output}_j(1 - \text{Output}_j) \sum_{k=1}^n \text{Error}_k w_{jk}$$

$$\begin{aligned} \text{Error 6} &= \text{output6} \cdot (1 - \text{output6}) \cdot \\ &(\text{outputtarget} - \text{output6}) \\ &= 0.474 \cdot (1 - 0.474) \cdot (1 - 0.474) \\ &= 0.474 \cdot 0.526 \cdot 0.526 \\ &= 0.1311 \end{aligned}$$

$$\begin{aligned} \text{Error5} &= \text{output5} \cdot (1 - \text{output5}) \cdot \text{error6} \cdot \\ W_{51} \\ &= 0.525 \cdot (1 - 0.525) \cdot 0.1311 \cdot -0.2 \\ &= 0.525 \cdot 0.475 \cdot 0.1311 \cdot -0.2 \\ &= -0.0065 \end{aligned}$$

$$\begin{aligned} \text{Error4} &= \text{output4} \cdot (1 - \text{output4}) \cdot \text{error6} \cdot \\ W_{41} \\ &= 0.332 \cdot (1 - 0.332) \cdot 0.1311 \cdot -0.3 \\ &= 0.332 \cdot 0.668 \cdot 0.1311 \cdot -0.3 \\ &= -0.0087 \end{aligned}$$

Hitung nilai bobot dan nilai bias terbaru untuk setiap relasi antara hidden layer dan output layer. Untuk perhitungan nilai bobot dan bias baru menggunakan *learning rate* 0.9. Tabel 2.2 adalah tabel nilai bias dan bobot terbaru, perhitungan nilai bias dan bobot terbaru dengan menggunakan rumus:

$$w_{ij} = w_{ij} + l \cdot \text{Error}_j \cdot \text{Output}_i$$

Tabel 2.3.
Perhitungan untuk bias dan bobot baru

Bobot atau bias	Nilai baru
W_{41}	$-0.3 + (0.9)(0.1311)(0.332) = -0.261$
W_{51}	$-0.2 + (0.9)(0.1311)(0.525) = -0.138$
W_{11}	$0.2 + (0.9)(-0.0087)(1) = 0.192$
W_{12}	$-0.3 + (0.9)(-0.0065)(1) = -0.306$
W_{21}	$0.4 + (0.9)(-0.0087)(0) = 0.4$
W_{22}	$0.1 + (0.9)(-0.0065)(0) = 0.1$
W_{31}	$-0.5 + (0.9)(-0.0087)(1) = -0.508$
W_{32}	$0.2 + (0.9)(-0.0065)(1) = 0.194$
$b1$	$0.1 + (0.9)(0.1311) =$

	0.218	
b2	$0.2 + (0.9)(-0.0065)$	=
	0.194	
b3	$-0.4 + (0.9)(-0.0087)$	=
	-0.408	

2.2. Feature Selection

Feature Selection adalah masalah berkaitan erat dengan pengurangan dimensi. Tujuan *Feature Selection* adalah untuk mengidentifikasi fitur dalam kumpulan data yang sama pentingnya, dan membuang semua fitur lain seperti informasi yang tidak relevan dan berlebihan. Karena *Feature Selection* mengurangi dimensi dari data, sehingga memungkinkan operasi yang lebih efektif & algoritma data mining yang cepat (yaitu data algoritma Mining dapat dioperasikan lebih cepat dan lebih efektif dengan menggunakan *Feature Selection*) (Maimon & Rokach, 2010).

Tujuan seleksi fitur adalah untuk pengurangan fitur, untuk menghilangkan dari dataset subset dari variabel yang tidak dianggap relevan untuk tujuan dari kegiatan data mining dan fitur metode seleksi dapat diklasifikasikan ke dalam tiga kategori utama (Vercellis, 2009):

1. Metode filter

Metode Filter adalah memilih atribut yang relevan sebelum pindah ke tahap pembelajaran berikutnya, atribut yang dianggap paling penting yang dipilih untuk pembelajar, sedangkan sisanya dikecualikan.

2. Metode wrapper

Metode *wrapper* menilai sekelompok variabel dengan menggunakan klasifikasi yang sama atau algoritma regresi digunakan untuk memprediksi nilai dari variabel target.

3. Metode embedded

Untuk metode *embedded*, proses seleksi atribut terletak di dalam algoritma pembelajaran, sehingga pemilihan set optimal atribut secara langsung dibuat selama fase generasi model.

Terdapat dua *feature selection* yaitu *backward elimination* dan *forward selection*

2.2.1. Forward selection

Berikut langkah-langkah dalam *forward selection*

1. Buat populasi awal dengan individu n dimana n adalah jumlah contoh masukan set atribut. Setiap individu akan menggunakan salah satu fitur yang persis.

2. Mengevaluasi set atribut dan memilih hanya yang terbaik k (*best k*).
3. Untuk setiap set atribut k lakukan: Jika ada j atribut yang tidak terpakai, membuat salinan j dari himpunan atribut dan menambahkan tepat satu dari atribut sebelumnya tidak terpakai ke set atribut.
4. Selama kinerja membaik dalam iterasi p , kemudian ulangi langkah ke-2.

2.2.2. Backward Elimination

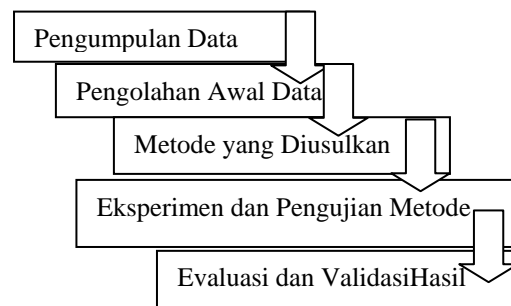
Berikut langkah-langkah dalam *backward elimination*

1. Mulailah dengan semua prediktor dalam model
2. Hapus prediktor dengan p -value tertinggi atau lebih besar dari target
3. Perbaiki model dan ulangi langkah ke-2
4. Hentikan ketika semua jika p -value kurang dari target

2.3. Perancangan Penelitian

2.3.1. Metode Penelitian

Penelitian ini adalah penelitian eksperimen, data yang digunakan adalah data produksi susu sapi perah dari tahun 2008 s/d 2010 dengan runtun waktu perbulan, dengan metode penelitian sebagai berikut :



Gambar 2.4
Metode Penelitian

1. Pengumpulan Data

Pada bagian ini dijelaskan tentang bagaimana dan darimana data dalam penelitian ini didapatkan, meliputi data sekunder dan data primer. Data sekunder berisi tentang sumber pemerolehan data untuk keperluan penelitian, sedangkan data primer berisi tentang data yang dihasilkan dari penelitian.

2. Pengolahan Awal Data

Pada bagian ini dijelaskan tentang tahap awal *data mining*. Pengolahan awal data meliputi proses *input* data ke format yang dibutuhkan, pengelompokan dan penentuan atribut data.

3. Metode yang Diusulkan

Pada bagian ini dijelaskan tentang metode yang diusulkan untuk digunakan pada prediksi cuaca jangka pendek. Penjelasan meliputi pengaturan dan pemilihan nilai dari parameter-parameter dan arsitektur melalui uji coba.

4. Eksperimen dan Pengujian Metode

Pada bagian ini dijelaskan tentang langkah-langkah eksperimen meliputi cara pemilihan arsitektur yang tepat dari model atau metode yang diusulkan sehingga didapatkan hasil yang dapat membuktikan bahwa metode yang digunakan adalah tepat.

5. Evaluasi dan Validasi Hasil

Pada bagian ini dijelaskan tentang evaluasi dan validasi hasil penerapan metode pada penelitian yang dilakukan. Penjelasan mengenai hal ini akan dipaparkan pada bab IV.

2.4. Metode Pengumpulan Data

1. Data sekunder yaitu data susu sapi perah perbulan dimulai dari tahun 2008 s/d 2010 dengan runtun waktu 1 bulan yang didapat dari GAPOKTAN Cikadu Sukabumi.
2. Data Primer yaitu data yang dihasilkan dari pengolahan data dengan menggunakan Rapid Miner.

2.5. Metode Pengolahan Awal Data

Data yang didapatkan dari GAPOKTAN Cikadu Sukabumi adalah data produksi susu sapi perah dengan runtun waktu perbulan, data yang digunakan yaitu data selama tahun 2008 s/d 2010, dimana setiap tahunnya terdapat 12 data jika dalam waktu 3 tahun maka didapat data sebanyak 36 data. Yang memuat deret perbulan yang dimulai dari bulan januari 2008 s/d desember 2010.

III. PEMBAHASAN

3.1. Neural Network

3.1.1. Perhitungan dengan menggunakan Algoritma Backpropagation

Diketahui record pertama pada tanggal 31/01/2010)

Tabel 3.1 Data uji (Record 1 pada tanggal 31 Januari 2010)

Tanggal	A	B	C	D	E	F	G	H	I	J	K
31/01/2012	1	0,7	0,1	0,7	0,6	0,3	0,3	0,7	0,9	0,6	0,9

Diketahui perhitungan pada pengujiannya, menggunakan learning rate 0,2 dan momentum 0,2 dan nilai bobot terakhir:

Tabel 3.1 Nilai bobot akhir untuk hidden layer

Hidden Layer (Sigmoid)											
Node	A	B	C	D	E	F	G	H	I	J	K
1	-1.047	1.317	0.885	-0.630	0.933	-1.050	0.732	-0.543	0.237	-0.607	-0.103

Nilai bobot akhir untuk output layer

Node	Output(Regression (Linear))
1	2.109
Threshold	-0.189

perhitungan pada pengujiannya, menggunakan learning rate 0,2 dan momentum 0,2 sebagai berikut:

Hitung nilai input, output untuk setiap node.

Rumus perhitungan input:

Berdasarkan gambar 4.1 maka , perhitungan input yaitu pada node 1, 2, 3 dan node target

$$\text{Input } 1 = (x1 * x_{11}) + (x2 * x_{21}) + (x3 * x_{31}) + (x4 * x_{41}) + (x5 * x_{51}) + (x6 * x_{61}) + (x7 * x_{71}) + (x8 * x_{81}) + (x9 * x_{91}) + (x10 * x_{101}) + (x11 * x_{111}) + (x12 * x_{121}) + bias1$$

$$\text{Input } 1 = (1 * (-1.047)) + (0.7 * 1.317) + (0.1 * 0.885) + (0.7 * (-0.630)) + (0.6 * 0.933) + (0.5 * (-1.050)) + (0.3 * 0.732) + (0.7 * -0.543) + (0.9 * 0.237) + (0.6 * -0.607) + (0.9 * -0.103) + (0.8 * 0.129) + (-4.372) = -1.0221$$

$$\text{Output } 1 = \frac{1}{1 + e^{-\text{input}}}$$

$$\text{Output } 1 = \frac{1}{1 + e^{(-\text{input})}}$$

$$\text{Output } 1 = \frac{1}{1 + 2,71828^{(-1.0221)}}$$

$$\text{Output } 1 = 0,3677$$

$$\text{Input node target} = (\text{output } 1.01) + 0_bias$$

$$\text{Input node target} = (0,3677 * 2.109) + 0.189$$

$$\text{Input node target} = 0,9$$

3.1.2. Perhitungan dengan menggunakan Algoritma Backpropagation dengan Rapid Miner

Eksperimen pertama dilakukan untuk mendapatkan nilai training cycles, yaitu dilakukan uji coba dengan memasukan nilai dengan range 100 sampai dengan 1000, nilai 0.1 untuk learning rate dan nilai 0.2 untuk momentum. Berikut hasil eksperimen terhadap training cycles:

Tabel 3.2 eksperimen untuk

menentukan nilai training cycles

Training	Learning	Momentum	RMSE
----------	----------	----------	------

Cycles	Rate		
100	0.2	0.2	0.499
200	0.2	0.2	0.514
300	0.2	0.2	0.516
400	0.2	0.2	0.518
500	0.2	0.2	0.518
600	0.2	0.2	0.519
700	0.2	0.2	0.519
800	0.2	0.2	0.519
900	0.2	0.2	0.520
1000	0.2	0.2	0.520

Hasil penelitian menunjukkan bahwa menambah nilai *training cycles* hanya membawa perubahan sedikit terhadap nilai *root mean square error* (RMSE), berdasarkan dari hasil training cycle diatas training cycle dengan nilai 100 memiliki nilai RMSE terendah. menambah nilai *training cycles* hanya memperlambat jalannya eksekusi. Dikarenakan hal tersebut maka, nilai *training cycles* yang dipilih yaitu *training cycles* dengan nilai 100. Nilai ini selanjutnya akan digunakan untuk menentukan nilai *learning rate*.

Untuk menentukan nilai *learning rate* dilakukan eksperimen dengan memasukan nilai dari 0.1 sampai dengan 0.5, nilai *training cycles* yang digunakan yaitu nilai *training cycles* yang didapat dari eksperimen sebelumnya yaitu 100 dan 0.2 untuk nilai momentum.

Tabel 3.3 eksperimen untuk menentukan nilai *learning rate*

Training Cycles	Learning Rate	Momentum	RMSE
100	0.1	0.2	0.467
100	0.2	0.2	0.499
100	0.3	0.2	0.531
100	0.4	0.2	0.573
100	0.5	0.2	0.693

Tabel 3.4 eksperimen untuk menentukan nilai momentum

Training Cycles	Learning Rate	Momentum	RMSE
100	0.1	0.0	0.459
100	0.1	0.1	0.459
100	0.1	0.2	0.467
100	0.1	0.3	0.479
100	0.1	0.4	0.490

Tabel 3.5 eksperimen untuk menentukan *Hidden Layer Sizes* pada *Hidden Layer 2*

Dari hasil eksperimen untuk *hidden layer* RMSE yang dihasilkan dengan nilai terkecil

Training Cycles	Learning rate	Momentum	Sizes Hidden 1	Sizes Hidden 2	RMSE
100	0.1	0.1	1	1	0.313
100	0.1	0.1	1	2	0.302
100	0.1	0.1	1	3	0.300
100	0.1	0.1	1	4	0.337
100	0.1	0.1	1	5	0.340
100	0.1	0.1	1	6	0.335
100	0.1	0.1	1	7	0.336
100	0.1	0.1	1	8	0.336
100	0.1	0.1	1	9	0.357
100	0.1	0.1	1	10	0.351

yaitu dengan hidden layer pertama dengan size 1 dan hidden layer kedua dengan size 3.

Berdasarkan eksperimen yang telah dilakukan, didapatkan hasil arsitektur jaringan yang dianggap terbaik yaitu dengan nilai RMSE yang paling rendah, sebagai berikut:

Tabel 3.6 Parameter *Neural Network* hasil eksperimen

Training cycles	Learning rate	Momentum	Jumlah hidden layer	Size hidden layer1	Size hidden layer2	RMSE
100	0.1	0.1	2	1	3	0.300

Tabel 3.7 Nilai bobot akhir untuk *output layer*

Node	Output(Regression (Linear))
1	0.295
2	0.289
3	0.265
Threshold	0.379

Penjabaran hasil perhitungan akhir *backpropagation* fungsi aktivasi untuk simpul pada *hidden layer* dan table 4.8 menunjukkan bobot akhir hasil eksperimen, langkah selanjutnya adalah melakukan pengujian. Proses pengujian dapat dilakukan, baik pada data yang dilatih maupun data input lain (Kusumadewi & Hartati, 2010).

3.1.3. *Neural Network* Berbasis *Backward Elimination*

3.1.3.1. Parameter *Neural Network* berbasis *Backward Elimination*

Eksperimen pertama dilakukan untuk mendapatkan nilai *training cycles*, yaitu dilakukan uji coba dengan memasukan nilai dengan range 100 sampai dengan 1000, nilai 0.3 untuk *learning rate* dan nilai 0.2 untuk *momentum*. Berikut hasil eksperimen terhadap *training cycles*:

Tabel 3.8 eksperimen untuk menentukan nilai *training cycles*

Trainin	Learnin	Momentu	RMS
---------	---------	---------	-----

<i>g Cycles</i>	<i>g Rate</i>	<i>m</i>	<i>E</i>
100	0.2	0.2	0.369
200	0.2	0.2	0.448
300	0.2	0.2	0.389
400	0.2	0.2	0.402
500	0.2	0.2	0.507

Hasil eksperimen menunjukkan bahwa nilai *training cycles* dengan *RMSE* terendah terdapat pada *training cycles* dengan nilai 100 dengan *RMSE* 0.369

Tabel 3.9 eksperimen untuk menentukan nilai *learning rate*

<i>Trainin g Cycles</i>	<i>Learnin g Rate</i>	<i>Momentu m</i>	<i>RMS E</i>
100	0.1	0.2	0.347
100	0.2	0.2	0.448
100	0.3	0.2	0.391
100	0.4	0.2	0.351
100	0.5	0.2	0.432

Dari hasil eksperimen diatas didapat bahwa nilai *learning rate* dengan *RMSE* terendah yaitu pada *learning rate* dengan nilai 0.1 dengan *RMSE* 0.347.

Tabel 3.10 eksperimen untuk menentukan nilai *momentum*

<i>Trainin g Cycles</i>	<i>Learnin g Rate</i>	<i>Momentu m</i>	<i>RMS E</i>
100	0.1	0.0	0.353
100	0.1	0.1	0.349
100	0.1	0.2	0.347
100	0.1	0.3	0.344
100	0.1	0.4	0.347
100	0.1	0.5	0.343

Berdasarkan hasil eksperimen diatas maka didapat nilai *momentum* 0.5 dengan *RMSE* terendah yaitu 0.343.

3.1.3.2. Menentukan Jumlah *Hidden Layer* dan *Hidden Layer Sizes*

Pada jumlah *hidden layer* sebanyak 1, dilakukan percobaan pada *hidden layer sizes* dengan range 1 sampai dengan 5 sizes. Berikut hasil eksperimen dengan 1 *hidden layer* :

Tabel 4.11 Eksperimen untuk menentukan *Hidden Layer Sizes* pada *Hidden Layer 1*

<i>Training Cycles</i>	<i>Learning Rate</i>	<i>Momentum</i>	<i>Sizes</i>	<i>RMSE</i>
100	0.1	0.5	1	0.333
100	0.1	0.5	2	0.302
100	0.1	0.5	3	0.351
100	0.1	0.5	4	0.378
100	0.1	0.5	5	0.292
100	0.1	0.5	6	0.319

Hasil eksperimen menunjukkan 1 *hidden layer* dengan *sizes* 5 yang menghasilkan *Root Mean Square Error (RMSE)* yang terkecil yaitu, dengan nilai *RMSE* 0.292.

Tabel 4.12 Eksperimen untuk menentukan *Hidden Layer Sizes* pada *Hidden Layer 2*

<i>Training Cycles</i>	<i>Learning Rate</i>	<i>Momentum</i>	<i>Hidden 1 Sizes</i>	<i>Hidden 2 Size</i>	<i>RMSE</i>
100	0.1	0.5	5	1	0.294
100	0.1	0.5	5	2	0.298
100	0.1	0.5	5	3	0.272
100	0.1	0.5	5	4	0.297
100	0.1	0.5	5	5	0.335
100	0.1	0.5	5	6	0.319

Hasil eksperimen menunjukkan 2 *hidden layer* dengan *sizes* 5 pada *hidden layer 1* dan *size* 3 pada *hidden layer 2* yang menghasilkan *Root Mean Square Error (RMSE)* yang terkecil yaitu, dengan nilai *RMSE* 0.272.

Dari hasil eksperimen diatas didapat arsitektur jaringan yang dianggap terbaik yaitu dengan nilai *RMSE* yang paling rendah, sebagai berikut:

Training cycles 100, Learning rate 0.1, Momentum 0.5, jumlah *hidden layer 2*, *size hidden layer1* 5, *size hidden layer2* 3 dengan *RMSE* 0.272

Tabel 3.13 Komparasi Root Mean Square Error (RMSE)

	<i>Neural Network</i>	<i>Neural Network berbasis Backward Selection</i>
RMSE	0.300	0.272

4. KESIMPULAN

4.1. Kesimpulan

Dalam penelitian ini dilakukan pengujian model dengan menggunakan *neural network* dan *neural network* berbasis *backward elimination* dengan menggunakan data produksi susu sapi perah perbulan. Model yang dihasilkan diuji untuk mendapatkan nilai *RMSE* terendah sehingga didapat hasil pengujian dengan *Neural Network* yaitu dengan *RMSE* 0.300 dan hasil pengujian dengan menggunakan *Neural Network* berbasis *Backward Elimination* didapatkan *RMSE* terendah yaitu 0.272.

Dengan demikian dari hasil pengujian model diatas dapat disimpulkan bahwa *neural network* berbasis *Backward Elimination* memberikan pemecahan untuk prediksi produksi susu sapi perah jangka pendek lebih akurat jika dibandingkan dengan *Neural Network* saja.

4.2. Saran

Dari hasil pengujian yang telah dilakukan dan hasil kesimpulan yang diberikan maka ada saran atau usul yang di berikan antara lain:

1. Dalam Penelitian ini dilakukan menggunakan metode *neural network* dan *neural network* berbasis *Backward Elimination*. Pada *Backward Elimination* mencoba mengurangi beberapa atribut dan mencobakan kembali dengan *neural network*.
2. Menggunakan parameter yang lebih bervariasi, sehingga akan menghasilkan prediksi yang lebih baik lagi.
3. Melakukan simulasi prediksi dengan jumlah data yang lebih banyak, seperti mengganti data bulanan menjadi data perharisehingga analisa akan bertambah optimal dan akurat.

Hasil penelitian ini diharapkan bisa digunakan untuk prediksi produksi susu sapi perah jangka pendek guna meningkatkan akurasi dalam prediksi produksi susu sapi perah jangka pendek.

DAFTAR PUSTAKA

- Aldrian, E., & Djamil, Y. S. (2008). Application of Multivariate ANFIS For Daily Rainfall Prediction : Influences of Training Data Size. *MAKARA*, 12(April 2008), 7–14.
- Beniwal, S., & Arora, J. (2012). Classification and Feature Selection Techniques in Data Mining. *International Journal of Engineering Research & Technology (IJERT)*, 1–6.
- Chris Fraley, Adrian Raftery, Tilmann Gneiting, McLean Slaughter & Veronica Berrocal. Probabilistic Weather Forecasting in R. *The R Journal*. 56-62.
- Felkin, M. (2007). between N -ary and Binary Problems. *Between N-ary and Binary Problems*, 1–25.
- Han & Kamber. (2006). *Data Mining: Concept and Technique* (2nd ed). United State America. Heaton. (2008). *Introduction to Neural Network With Java* (2nd ed). USA. Heaton Research, Inc.
- Hong, X., Harris, C., Brown, M., & Chen, S. (2002). *Backward Elimination Methods for Associative Memory Network Pruning*. *Computers and Technology*, (Reed 1993).
- Hung, N. Q., Babel, M. S., Weesakul, S., & Tripathi, N. K. (2009). An artificial neural network model for rainfall forecasting in Bangkok , Thailand. *Hydrology and Earth System Sciences*, 1413–1425.
- Jong Jek Siang. (2009). *Jaringan Syaraf Tiruan & Pemrogramannya Menggunakan MATLAB*. Yogyakarta : Andi Yogyakarta.
- Kohavi, R., & John, H. (1997). *Artificial Intelligence Wrappers for feature subset selection*. *elsevier*, 273–324.
- Kusumadewi, Sri & Hartati, Sri. (2010). *Neuro-Fuzzy Integrasi Sistem Fuzzy & Jaringan Syaraf*. (2nd ed). Yogyakarta: Graha Ilmu.
- Lin, S.-W., Chen, S.-C., Wu, W.-J., & Chen, C.-H. (2009). Parameter determination and feature selection for back-propagation network by particle swarm optimization. *Knowledge and Information Systems*, 21(2), 249–266.
- Liu, H., Member, S., Yu, L., & Member, S. (2005). Algorithms for Classification and Clustering. *IEEE Transaction on Knowledge and Data Engineering*, 17(April 2005), 491–502. Program Pascasarjana Magister Ilmu Komputer STMIK Nusa Mandiri
- Pallavi, V. P., Vaithyanathan, V., & Ph, D. (2013). Combined Artificial Neural Network and Genetic Algorithm for Cloud Classification. *International Journal of*

- Engineering Research & Technology (IJET), (May), 787–794.
- Radhika, Y., & Shashi, M. (2009). Atmospheric Temperature Prediction using Support Vector Machines. *International Journal Of Computer Theory and Engineering*, 1(April), 55–58. doi:10.7763/IJCTE.2009.V1.9
- Selection, O. F., & Elimination, B. (2004). Orthogonal Forward Selection and Backward Elimination for Feature Subset Selection. *IEEE Transactions on System Man and Cybernetics*, 1(February), 629–634.
- Solaimani, K. (2009). Rainfall-runoff Prediction Based on Artificial Neural Network (A Case Study: Jarahi Watershed). *IDOSI Publication*, 5(6), 856– 865.
- Vercellis, C. (2009). *Business Intelligence: Data Mining and Optimization for Decision Making*. Wiley.
- Winarso.(2002). *Pemikiran dan Praktek Perencanaan dalam Era Tranformasi di Indonesia*. Bandung: Departemen Teknik Planologi ITB.
- Zhu, Z., Ong, Y., & Dash, M. (n.d.). Wrapper-Filter Feature Selection Algorithm Using A Memetic Framework. *International Journal of Engineering Research & Technology (IJET)*, 1–19.