Comparison of Evolutionary Algorithm to Neuro-Fuzzy and Fuzzy Clustering for Electrical Load Time Series Forecasting

Hany Ferdinando, Felix Pasila, Dharma Gunawan and William

The Neuro-Fuzzy Network and Fuzzy Clustering system are known as good algorithm in time series forecasting application but the parameters need to be optimized. As the Evolutionary Algorithm (EA) grows, it is interesting to implement EA to optimize the parameters of Neuro-Fuzzy and Fuzzy Clustering. The time series data is the electrical load of East Java-Bali, Indonesia, in 2005-2007. The results of unoptimized parameters for both algorithms were not satisfied. Therefore, the EA is implemented to optimize their parameters. The EA used Real Code Genetic Algorithm. The parameters optimized with EA are mean and variance of the Gaussian MF with pc=0.6, pm=0.1 and 20 chromosomes per population. The experiments without optimization showed that MSE LTF for Fuzzy Clustering is 2.9x10^{-3} and Neuro-Fuzzy is 2.0x10^{-3}. MSE of long time forecasting (LTF) for optimized Fuzzy Clustering is 2.42x10^{-3} while for Neuro-Fuzzy Network is 1.9x10^{-3}. These results indicated that the EA has no effect for Neuro-Fuzzy because the difference is small. For Fuzzy Clustering, the result is interesting. But it is still not very satisfying. This project still needs improvements in order to get more satisfying results.

Keywords: Evolutionary Algorithm, Long Time Forecasting, Neuro-Fuzzy, Fuzzy Clustering.

1. INTRODUCTION

Forecasting data time series is interesting for it involves non-linear model. A lot of algorithms are designed to have better performance in forecasting time series data. Electrical load can be considered as time series data for the data is retrieved based on time. The electrical load can be used as tool in planning electrical system and maintain day to day operation of the power plant [1]. It can also be used to manage electrical energy system efficiently [2].

Parallel to [1] and [2], this research is intended to get such a decision tool to assist the management in planning and maintaining the operation of the power plant. Using such an intelligent system or soft computing as tools to do this is not new. [3] uses hybrid intelligent maintenance system to diagnosis fault and to forecast condition based on the signal processing method. It is applied to maintain turbo-generator set in a power plant. [4] uses Empirical Mode Decomposition, Fuzzy Feature Extraction and Support Vector Machines to diagnosis fault in power plant.

This paper elaborates two previous researches in electrical load time series data forecasting with Fuzzy Clustering with Gustafson-Kessel [5] and Neuro-Fuzzy [6]. Results from the previous researches are not satisfied due to unoptimized parameters. If the
parameter used in [5] and [6] can be optimized, then better result will be achieved. The Focus of this research is on the Long Time Forecasting (LTF) of East Java-Bali Electrical load in 2004-2007. [5] showed the MSE of LTF is $2.9 \times 10^{-3}$ while [6] gave $2.0 \times 10^{-3}$.

The Evolutionary Algorithm (EA) is one of optimization algorithm adapted from natural process. The parameters will evolve to certain convergence values through natural-like process. Some call EA as Genetic Algorithm (GA). Here, the unoptimized parameters will be optimized in order to get better performance. The goal is to get good parameters for Neuro-Fuzzy and Fuzzy Clustering so that the MSE is below 5% and compare their results. This is the goal of this research.

First, this paper shows the previous result of similar project with the same time series data. General approach and result of each project will be shown. Next, the authors discuss the implementation of EA for Neuro-Fuzzy and Fuzzy Clustering. The implemented algorithms are simulated to get the result. Discussions and conclusions at the end this paper summarize the whole result.

2. PREVIOUS RESEARCHES

Two previous similar researches for time series data forecasting with electrical load are presented here. For both used the same data, it is easy to compare their result. The data is retrieved every 30 minutes from September 2005 to June 2007. It was divided into two parts, i.e. data for training (September 2005 to December 2006) and data for testing (January to June 2007). All previous researches and the current research use the same data. This enables us to compare their performance under the same condition. Both [5] and [6] use normalized data before processing. This helps in limiting the variable size of the system. The data is arranged weekly so the number of data is 336. This data is placed in a matrix for the next calculation.

2.1 Fuzzy Clustering for Time Series Data Forecasting

Time series data forecasting with Fuzzy Clustering was implemented in [5]. The Fuzzy Clustering was enhanced with Gustafson-Kessel clustering method in order to get better performance compared to Fuzzy C-mean result [7]. The Fuzzy Clustering uses Takagi-Sugeno (TS) Fuzzy inference system with MISO (Multiple Input Single Output) structure. Here is the Fuzzy inference written in TS

If $X_1$ is $G_1$ AND … AND $X_n$ is $G_n$ THEN $Y = H_0 + H_1x_1 + \ldots + H_nx_n$ \hspace{1cm} (1)

Output of defuzzification can be calculated with

$$y_0 = \frac{\sum_{i=1}^{M} \beta^i \cdot y_{TS}^i}{\sum_{i=1}^{M} \beta^i} = \frac{(\beta^1 \cdot y_{TS}^1 + \beta^2 \cdot y_{TS}^2 + \ldots + \beta^M \cdot y_{TS}^M)}{(\beta^1 + \beta^2 + \ldots + \beta^M)}$$ \hspace{1cm} (2)

With

$$\beta^i = \prod_{i=1}^{t} \mu_{G_i}(x_i) = \mu_{G_i}(x_1) \times \mu_{G_i}(x_2) \times \ldots \times \mu_{G_i}(x_t)$$ \hspace{1cm} (3)

$t$ is the number of inputs.

So the output of TS is
\[ y_o = (\gamma_1 y_{TS} + \gamma_2 y_{TS} + \cdots + \gamma_M y_{TS}) \]  

with \( \gamma \) is normalized degree of fulfillment

\[ \gamma_s = \frac{\beta_s}{\sum_{i=1}^{M} \beta_s} = \frac{\beta_s}{\beta_1 + \beta_2 + \cdots + \beta_M} \]  

The membership function is Gaussian with equation

\[ G_i = e^{-\frac{(x_i - \mu)^2}{2\sigma^2}} \]  

The two important parameters in Gaussian Membership Function (GMF) are mean and variance with Takagi-Sugeno parameters for defuzzification process. These two parameters will be optimized.

System uses nine inputs, four clusters and the number of fuzziness exponent equal to two. The short term forecasting (SFT) shows that the system had to use 1 data per set, i.e. data arranged per day. The RMSE for STF is 0.0336.

For the long time forecasting, the 336 data per set is used, the data is arranged per week. The RMSE for LTF is 0.0534.

Compare to Fuzzy C-mean result, the Gustafson-Kessel Fuzzy clustering method shows better result. The RMSE for STF and LTF are 0.05 and 0.048 respectively.

2.2 Neuro-Fuzzy for Time Series Data Forecasting

Another implemented algorithm with the same time series data is Neuro-Fuzzy. The training algorithm used Levenberg-Marquardt Algorithm (LMA) [6]. The Fuzzy part uses Takagi-Sugeno type as part of multiple input multiple output for the feed forward neural network. The membership function uses Gaussian type. Fig. 1 shows the architecture of the Neuro-Fuzzy system.

Feedforward structure of MIMO Neuro-fuzzy in Fig. 1 uses Gaussian membership function as in (6), TS fuzzy rule type, product inference and weighted average defuzzification. Nodes at the first layer calculate degree of membership of fuzzy set numeric input. Outputs from block with ‘X’ (see fig. 1) correspond to the degree of fulfillment of fuzzy rule system. Outputs from block with ‘/’ together with summing point
at the center result normalized degree of fulfillment of respective fuzzy rule. Result of these blocks multiplied with TS rule consequent are used as inputs to the final summing points output.

Fuzzy Logic System (FLS) used in [6] is product inference rule and weighted average defuzzifier defined as

$$f_j(x_p) = \sum_{l=1}^{M} \frac{y^j \cdot z^l}{\sum_{l=1}^{M} z^l}$$  \hspace{1cm} (7)

$$y^j = w^j_0 + \sum_{l=1}^{n} w^j_l x_i, \text{ with } i = 1, 2, 3, \ldots, n$$  \hspace{1cm} (8)

$$z^l = \prod_{i=1}^{n} \exp\left\{-\frac{(x_i - c^l_i)^2}{\sigma^2_i}\right\}$$  \hspace{1cm} (9)

The architecture will be trained with Levenberg-Marquardt Algorithm (LMA). This algorithm speeds up ordinary Back Propagation Algorithm (BPA). To overcome the oscillation in LMA, Modified Error Index (MEI) and momentum are added [9]. The oscillation is controlled with parameter called Wildness Factor (WF). Best performance of this architecture is four inputs, one output with momentum is 0.005, WF is 1.001, five GMF and gamma is 0.0005. The RMSE for LTF is 0.0332.

The result of Neuro-Fuzzy algorithm is not satisfying. It is interesting to get better performance for its parameters are not optimized yet. When they are optimized, then it is possible to get better performance.

The main question raised in this research is which algorithm will have better performance after the optimization process. The EA will be implemented on Fuzzy Clustering with Gustafson-Kessel approach and Neuro-Fuzzy system. The authors did not choose Fuzzy C-mean clustering for its result is not good as with Gustafson-Kessel.

3 EVOLUTIONARY ALGORITHM

The need of good optimizer becomes more and more important nowadays. A lot of algorithms for optimization have been developed but they do not make the researchers satisfied with the results.

One of the best optimizer is natural process, so it is interesting to find an algorithm similar to natural process for optimization. This leaded us to Evolutionary Algorithm (EA). In the EA, we will find Genetic Algorithm (GA) and Evolutionary Computation (EC). Actually, they are the same so the authors will use EA to refer to GA or EC.

First, EA came with binary chromosomes and it is very simple and easy to understand. But not all problems can be solved with binary chromosomes. Then, there was an idea to deal with real number chromosome. With this new type of chromosome, the crossover and mutation must be modified.

The EA with real number chromosomes draws a lot of interest because its application is wide. Almost all problem can be approached with real number chromosomes.
4 IMPLEMENTATION

The modeling process from the previous research resulted mean and variance of the GMF. These two parameters will be optimized with EA.

For the means and variances are real number, the chromosomes must be in real as well. Using binary chromosome will make the implementation difficult. This gives certain consequences for crossover and mutation procedures. Crossover combines genetic code from two chromosomes, called parents, to produce two other chromosomes with better genetic combination than their parents. For the chromosomes are real number, the crossover uses heuristic crossover with equation

\[ S_{a}^{S+1} = S_{a}^{g} + r(S_{b}^{g} - S_{a}^{g}) \]
\[ S_{b}^{S+1} = S_{b}^{g} + r(S_{a}^{g} - S_{b}^{g}) \]

with \( r \) is the probability crossover procedures, called \( P_{c} \). The \( S_{a}^{g} \) and \( S_{b}^{g} \) are parents for crossover operation while \( S_{a}^{S+1} \) and \( S_{b}^{S+1} \) are offspring with new genetic code. \( P_{c} \) is chosen between 0.6 and 0.9. The mutation for real chromosomes uses multiple uniform mutations. System generates \( n \) random variable and \( n \in \{1, 2, 3, \ldots, L_{\text{chromosome}}\} \). \( P_{m} \) is chosen between 0.1 and 0.4. The selection procedure uses roulette wheel and elitist with fitness function is

\[ fit = \frac{1}{MSE_{\text{LTF}}} \]

4.1 Fuzzy Clustering with Gustafson-Kessel

The EA is implemented on the same architecture as in [5], i.e. 9 inputs, 1 output, 4 cluster and 336 data per set (data per week). This architecture gives the best performance, i.e. 2.9x10^{-3}.

![Fig. 2. Example of GMF for input](image-url)
of GMF. For there are 4 cluster, so are the GMFs, see Fig. 2. Each GMF has mean and variance. The range of mean for GMFs in the middle is mean of left and right one. The left-most GMF take lower limit 10% from the initial GMF. This applies also for the right-most GMF but for the upper limit.

This 10% value is determined based on the pre-experiment to evaluate the improvement of architecture performance. When this value goes over 10%, there is significant result, so the lower limit is determined 10% below the initial value of the most left GMF. This also applies to upper limit for the most right GMF. The range for variance is simpler. The value is reduced or increased up to 50% of initial one. Variance only influences whether the GMF will be narrow or wide and the value is small. The EA uses heuristic crossover and multiple uniform mutations. When the difference of current and previous fitness value is less than 1.0x10^{-4}, the iteration will be stopped.

4.2 Neuro-Fuzzy with LMA Training

The architecture of Neuro-Fuzzy for optimization is the same as in [6], i.e. 4 inputs, 1 output, 5 GMFs and 336 data per set. It gives RMSE fro LTF 2.0x10^{-3}.

There are two matrices for mean and variance of GMF respectively. The dimension is 4x5 to represent 4 inputs and 5 GMFs. As the EA need specific range to search the optimum values, the range of mean and variance uses the same technique as in Fuzzy Clustering with Gustafson-Kessel. The EA use the following parameters: heuristic crossover and multiple uniform mutations. As in the optimization for [5], this process will use the same stopping criteria.

5. SIMULATIONS

5.1 Fuzzy Clustering with Gustafson-Kessel

First, it is important whether the optimization will be done simultaneously or separately for means and variances. The experiment for both method with Pc and Pm are 0.8 and 0.2 respectively are shown in Fig. 3. These experiments use 20 as population size.

![Fig. 3. Comparison between optimized the parameters simultaneously (black) or separately (grey)](image)
Experiment for optimization the parameters simultaneously show that the best MSE is achieved in 408th generation. Compare to the other method, the best MSE is found in 1939th generation and there is no significant improvement until 2000th generation. From this result, the experiment will optimize the parameters simultaneously. The iteration for the next experiments will be stopped at 1000th generation based on Fig. 3.

The authors use variation for Pm and Pc in order to find global optimum of LTF to optimize mean and variance. The experiments are setting Pc constant and Pm changing in determined range. Both Pc and Pm are changed with 0.1 step. The population size is 20.

The simulation shows that the best combination for Pc and Pm is 0.6 and 0.1 respectively. The MSE LTF for this combination is 2.4594x10⁻³. Table 1 shows the summary of all experiments to find the best combination for Pc and Pm.

<table>
<thead>
<tr>
<th>Pc</th>
<th>Pm</th>
<th>MSE LTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.6</td>
<td>0.1</td>
<td>2.46x10⁻³</td>
</tr>
<tr>
<td>0.6</td>
<td>0.2</td>
<td>2.54x10⁻³</td>
</tr>
<tr>
<td>0.6</td>
<td>0.3</td>
<td>2.52x10⁻³</td>
</tr>
<tr>
<td>0.6</td>
<td>0.4</td>
<td>2.95x10⁻³</td>
</tr>
<tr>
<td>0.7</td>
<td>0.1</td>
<td>2.47x10⁻³</td>
</tr>
<tr>
<td>0.7</td>
<td>0.2</td>
<td>2.63x10⁻³</td>
</tr>
<tr>
<td>0.7</td>
<td>0.3</td>
<td>2.54x10⁻³</td>
</tr>
<tr>
<td>0.7</td>
<td>0.4</td>
<td>2.66x10⁻³</td>
</tr>
<tr>
<td>0.8</td>
<td>0.1</td>
<td>2.49x10⁻³</td>
</tr>
<tr>
<td>0.8</td>
<td>0.2</td>
<td>2.52x10⁻³</td>
</tr>
<tr>
<td>0.8</td>
<td>0.3</td>
<td>2.59x10⁻³</td>
</tr>
<tr>
<td>0.8</td>
<td>0.4</td>
<td>2.61x10⁻³</td>
</tr>
<tr>
<td>0.9</td>
<td>0.1</td>
<td>2.56x10⁻³</td>
</tr>
<tr>
<td>0.9</td>
<td>0.2</td>
<td>2.49x10⁻³</td>
</tr>
<tr>
<td>0.9</td>
<td>0.3</td>
<td>2.58x10⁻³</td>
</tr>
<tr>
<td>0.9</td>
<td>0.4</td>
<td>2.55x10⁻³</td>
</tr>
</tbody>
</table>

It means these parameters will be used for next experiments. Fig. 4 shows the simulation for Pc=0.6 with variation value for Pm.

Fig. 4 shows that the MSE is improved until 20th generation and then it slows down. All combinations have almost the same performance. It looks like the unoptimized parameters are close to the optimized one.

If result on fig. 4 is compared to the other combinations for Pc and Pm, this result is good. The other combinations give worse result compare to [5].

It is also interesting to evaluate the simulation for variation in population size. Summary of this experiment is shown in Table 2.

Table 2 shows that the best MSE LTF is the experiment with population size 200 but the iteration is time consuming. If we compare 20 to 200 chromosomes, it looks like that the difference is not significant. Considering that the 200 chromosomes in a population is time consuming, it is recommended to choose 20 instead of 200.
This optimization for Neuro-fuzzy uses simultaneously approach as well. The comparison between two approach is similar to the one of fuzzy clustering. The simultaneously approach shows better in number of generation but the MSE difference is not significant. The only difference is that the iteration will be stop at 500th generation.

The authors use variation for Pm and Pc in order to find global optimum of LTF to optimize mean and variance with 4 inputs and 1 output. The number of GMF is 5. The simulation uses 336 data per set, it means data per week. Table 3 shows the summary of this experiment.

The simulation shows that the best combination for Pm and Pc is 0.2 and 0.9 respectively. These are used to optimized mean and variance together with variation in the population size. Fig. 5 shows the MSE LTF for Pm=0.2 and Pc=0.9. The MSE LTF is 1.9x10^{-3}.

The population size is an interesting issue in EA, so the experiment with population size is summarized in Table 4.

Table 4 shows that the best result is 1.9x10^{-3}. Compare to the unoptimized parameters [6], this is not significant. It looks like the parameters are already optimized.

This experiment recommends choosing 120 as population size. The others have MSE around 1.95x10^{-3} and only this population size has MSE below it.
This research recommends optimizing both parameters simultaneously. Although the MSE of this approach is greater than the other, the number of generation reduced significantly. In this experiment, the time for iteration is little bit consuming. When it is
multiplied with certain number, the value becomes greater and it is not recommended waiting with no significant MSE improvement.

The goal of this research is to compare the performance of EA in Fuzzy Clustering and Neuro-Fuzzy optimization to forecast time series data. So, it is important to analyze their performance in parallel.

Table 5 shows the MSE of unoptimized and optimized parameters in parallel.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Unoptimized</th>
<th>Optimized</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy Clustering</td>
<td>2.9x10⁻³</td>
<td>2.42x10⁻³</td>
</tr>
<tr>
<td>Neuro-Fuzzy</td>
<td>2.0 x10⁻³</td>
<td>1.90x10⁻³</td>
</tr>
</tbody>
</table>

The optimization with EA for Fuzzy Clustering gives no significant result. The MSE is reduced but the order is too small. Based on fig. 4, the each Pc and Pm combination gives different performance in early iteration but later, their performance is almost the same when the Pm is one, two or three.

The result for Neuro-fuzzy is more interesting because the MSE is almost the same with unoptimized implementation. It looks like that the parameters are closed to the ideal one. This is verified from Fig. 5. The iterations converge to single value. The authors never know what the converged value for this. The iteration is stop before getting that number for it is very slow.

These results open for the next research in order to improve it using another optimization algorithm, e.g. ant colony, simulated annealing, etc. Using more complex EA is also recommended for the next research.

Another idea to enhance this research is to add more parameter for optimization. By doing this, the performance can be improved.

In the EA, it is important to find the best combination among Pm, Pc and population size. It is determined by the complexity of the system being optimized. Table 6 shows the comparison of them.

<table>
<thead>
<tr>
<th>Architecture</th>
<th>Pm</th>
<th>Pc</th>
<th>Pop. Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Fuzzy Clustering</td>
<td>0.6</td>
<td>0.1</td>
<td>20</td>
</tr>
<tr>
<td>Neuro-Fuzzy</td>
<td>0.9</td>
<td>0.2</td>
<td>120</td>
</tr>
</tbody>
</table>

Actually, the experiment for Fuzzy Clustering with population size 200 gives the best result. The problem is the iteration is time consuming and it is recommended to use 20 instead of 200.

It is different from the Neuro-Fuzzy. The recommended population size is 120, not 20. From table 2, it is shown that population size of 20, 60 and 200 gives the MSE around 1.95x10⁻³. When the population size is 120, the MSE is 1.9x10⁻³. The iteration is little bit time consuming but the result is good.

If we compare mean and variance matrices from both architectures, it is shown that
dimension for Fuzzy Clustering is bigger, i.e. 4x9 compare to 4x5. This research will not change it for they are from the previous experiments.

The validation experiments only use data from January 2007 to June 2007. The data for current date is not available due technical problem.

The combination of Pm and Pc for both Fuzzy Clustering and Neuro-Fuzzy could be improved using adaptive Pm-Pc. We can also apply the optimization for other parameters in both Fuzzy Clustering and Neuro-Fuzzy. This research only focuses on the mean and variance of GMF. There is also possibility to optimize the number of GMF.

References


