# Genetic Algorithms and Rough Fuzzy Neural Network-based Hybrid Approach for Short-term Load Forecasting

Li Feng, Ziyan Liu

*Abstract*-- **This article describes a way of designing a hybrid system for short-term load forecasting, integrating rough sets theory with fuzzy neural networks using a multi-objective genetic algorithm. The multi-objective genetic algorithm is used to automatically learn the knowledge of historical data and find the best factors that are relevant to electric loads. The concept of entropy is introduced to describe the uncertainty of decision rules with dependency factors, and the crude domain knowledge expressed by decision rules is applied to design the structure and weights of the neural network. Simulation results demonstrate that the rough fuzzy neural network has better precision and convergence than the traditional fuzzy neural network for its simple, transparent network structure and effective inputs.**

*Index Terms*-- **Entropy, fuzzy neural networks, genetic algorithms, rough sets theory, short-term load forecasting**

## I. Introduction

Short-term electric load forecasting plays a key role in power system operation, and represents great potential uses for electric utilities. Its main objective is to extrapolate past load behavior while taking into account the effects of influencing factors. Since the electric load is a function of weather variables and human social activities, the variety of the short-term load is very complex, sometimes it changes evenly, sometimes linearly, and sometimes randomly. In early time regression models [1], time series and expert systems are generally used. Now artificial neural network have induced special interest for its ability in mapping complex non-liner relationships, which is responsible for the growing number of its application to the short-term load forecasting [2]-[6]. Several electric utilities over the world have been applying neural network for load forecasting in an experimental or operational basis [2], [7], [8]. But those algorithms above need a great deal of statistic information and transcendent knowledge, and it is complicated to calculate and train in intricate instances. However there are many factors that influence the precision of load forecasting directly or indirectly, and it is hard to determine the uncertainty between load and various factors. Therefore, the forecasting process has become even more complex, and more accurate forecasts are then needed. Data mining can solve the uncertainty that arises from inexact, noisy, or incomplete information. It provides an effective way for us to solve the difficulties.

Rough sets theory as a most typical algorithm of data mining has been applied in expert systems, decision support systems, and machine learning. In machine learning rough set is used to extract decision rules from operation data [9]; in neural network, rough sets is used in knowledge discovery, data pre-processing [10]-[12] and modeling knowledge-based neural networks [13], [14]. In order to improve the performance of the load forecasting, a novel model integrated with rough sets and fuzzy neural network for short-term load forecasting is presented in this article. In the model fuzzy sets help in handling linguistic input information and ambiguity in output decision, while rough sets extracts the relevant domain knowledge to the load, and then the network structure and initial weights are auto-adjusted by the knowledge encoded in the neural network. The simulation results show that the prediction accuracy is improved by applying the method on a real power system.

The article is organized as follows: Section II recalls elementary concepts of rough sets. A multi-objective genetic algorithm for attribute reduction is proposed in section III. The principle of the proposed rough fuzzy neural network is described in section IV. The simulation results and conclusions are presented in section V and VI respectively.

## II. Rough sets Theory

Rough sets theory, presented by Z.Pawlak in 1982, comes to be a new mathematic tool to manage inexact or incomplete information [15], [16].

A data set is represented as a quadruple $S=\langle U,A,V,f\rangle$, where $U=\{x_1, x_2 \dots x_n\}$ is a non-empty finite set of objects called the universe, and $A$ is a non-empty finite set of attributes, $A=C\cup D$ and $C\cap D=\varnothing$. $V=\bigcup_{a\in A}V_a$ is a set of attribute values of attribute *a*. $f:U\times A\rightarrow V$, where

Li Feng is with the Dispatch and Transaction Center, Chongqing Electrical Power Corporation, Chongqing 400014, P.R.China. (e-mail: beth002@sina.com ).

Ziyan Liu is with College of Computer Science and Communication Technology, Guizhou University (Cai Jiaguan Campus), Guiyang, Guizhou 550003, P.R.China. (e-mail: leizy@sina.com ).

$f(x_j, a) \in V_a$ for all $x_j \in V_a$ and $a \in A$.

### A. Indiscernibility Relation

Given an information system $S = \langle U, C \cup \{d\} \rangle$, $B \subseteq C$, and $x$, $y$ are members of $U$, there is an associated equivalence relation IND ($B$):

$$\mathrm{IND}(B) = \{(x, y) \in U \times U : a(x) = a(y) \ \ \forall a \in B\} \qquad (1)$$

IND ($B$) is also called an indiscernibility relation.

### B. Set Approximation

An equivalence relation induces a partitioning of the universe. Let $B \subseteq C$ and $X \subseteq U$. The $B$-lower and $B$-upper approximations of $X$, denoted $\underline{\mathrm{apr}}_B(X)$ and $\overline{\mathrm{apr}}_B(X)$ respectively, where $\underline{\mathrm{apr}}_B(X) = \cup\{Y \in U/B : Y \subseteq X\}$ and $\overline{\mathrm{apr}}_B(X) = \cup\{Y \in U/B : Y \cap X \neq \varnothing\}$. $\underline{\mathrm{apr}}_B(X)$ is also called the $B$-positive region of $X$ denoted as $\mathrm{POS}_B(X)$.

### C. Reducts and Core

A reduct of $S$, written as RED ($A$), is a minimal set of attributes $B \subseteq A$ such that

$$\mathrm{IND}(B) = \mathrm{IND}(A) \text{ and } \forall b \in B \ \mathrm{IND}(B - \{b\}) \neq \mathrm{IND}(A) \qquad (2)$$

A reduct preserves the partitioning of the universe, and hence the ability to perform classifications as the whole attribute set $A$ does, and it is usually not unique.

The core of $A$ is defined as $\mathrm{Core}(A) = \cap \mathrm{RED}(A)$.

### D. Discernibility Matrix and Discernibility Function

The discernibility matrix M($S$) is a symmetric $n\times n$ matrix with entries $c_{i,j}$ as given below.

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j)\} \qquad (3)$$

A discernibilty function $f_S$ for $S$ is defined as:

$$f_S = \wedge\{\vee(c_{ij}) :\ 1 \leq j \leq i \leq n, c_{ij} \neq \varnothing\} \qquad (4)$$

where $\vee(c_{ij})$ is the disjunction of all members of $c_{ij}$. The set of all prime implicants of $f_S$ determines the set of all reducts of $S$ [16].

Here we come to the notion of a relative reduct.

Relative reducts can be computed by using a $D$-discernibility matrix [denoted as $\mathrm{M}_D(S)$], the $ij$th component of which has the form

$$c_{ij} = \{a \in A : a(x_i) \neq a(x_j) \text{ and } (x_i, x_j) \notin \mathrm{IND}(D)\} \qquad (5)$$

for $i, j = 1,\ldots,n$.

The relative discernibility function $f_D$ is constructed in an analogous way as $f_S$ from the discernibility matrix of $S$.

### E. Dependency Factor and Significance of Attribute

For $C, D \subseteq A$, the dependency factor of $C$ on $D$ is given by

$$\gamma(C, D) = |\mathrm{POS}_C(D)| / |U| \qquad (6)$$

where |•| denotes cardinality of the set, and $\mathrm{POS}_C(D) = \underset{X \in U/D}{\cup} \underline{\mathrm{apr}}_C(X)$.

Significance of an attribute $a$ in $S$ can be defined as:

$$\mathrm{SGF}(a, A, D) = \frac{\gamma(A, D) - \gamma(A - \{a\}, D)}{\gamma(A, D)} \qquad (7)$$

### F. Decision Rules

Let $S$=$\langle U, A \cup \{d\} \rangle$ and $V = \bigcup\{V_a \mid a \in A\} \bigcup V_d$. A decision rule is any expression of the form:

$$\varphi \Rightarrow d{=}v \qquad (8)$$

where $\varphi$ and $d$=$v$ are referred to the predecessor and the successor of decision rule $\varphi \Rightarrow d$=$v$. $\varphi$=$\vee(\wedge(a{=}v))$, where $a \in B \subseteq A$ and $v \in V_a$. $a$=$v$ is an atomic formulae over $B$ and $V$. $\varphi$ is the disjunctive normal form (d.n.f.) of $f_S$.

## III. Attribute Reduction Algorithm

Statistic information and transcendent knowledge are needed in most of the algorithms for forecasting, but they are absent and imperfect in some cases. Moreover, Short-term electric load changes in a complicated way along with time, which cannot be described by an accurate mathematic model. And in short-term load forecasting there exist a great number of uncertain factors that are nonlinear to electric load. Some of them are correlated and some are independent. If all these factors are used as inputs of a neural network directly, it will not only result in a complicated structure, but long leaning time and inaccurate prediction.

In this paper an attribute reduction algorithm is applied to eliminate the redundant factors in our forecasting task. It is an NP-hard problem to find the minimal reduct [9]. Fortunately there exist good heuristics [17], [18] based on genetic algorithms that compute many reducts sufficiently in acceptable time.

### A. Attribute Reduction Based on Genetic Algorithm

Electric loads are normalized to the same range of values between 0 and 1 by $LD_{scale}(t) = (LD_{actual}(t) - LD_{\min}) / (LD_{\max} - LD_{\min})$, and then classified into $s$ clusters, where $LD_{max}$ and $LD_{min}$ denote the maximum and minimum value of the vector $LD_{scale}$ respectively, $t$=1,…,24. The weather factors fuzzified by triangular functions according to their characters, and are mapped into five-dimensional feature space of Very Low ($VL$), Low ($L$), Normal ($NM$), High ($H$) and Very High ($VH$). With the normalized historical loads used as the decision attribute $d$, and the fuzzified weather factors such as wind speed, maximum temperature, minimum temperature, humidity and rainfall treated as the conditional attributes set $C$, a decision table is gained.

Modify (3) to directly handle a real-valued attribute table consisting of fuzzy membership values. We define the individual of discernibility matrix as below.

$$c_{ij} = \{a \in A : |a(x_i) - a(x_j)| > Th\} \quad i, j{=}1\ldots n \qquad (9)$$

where $Th$ ($0.5 \leq Th \leq 1$) is a threshold depended on the inherent shape of the membership function.

The adaptive $Th$ of membership degree is illustrated in Fig.1. Let $a$, $b$ correspond to two membership functions with $b$ being steeper than $a$. It is observed that $r_1 > r_2$. So choosing an appropriate threshold enables the dicernibility matrix to contain the representative clusters present in a class.

While synthesizing the minimal decision rules, usually a reduct with minimal number of attributes is considered. But it is not sufficient to deduce exact decision rules by the minimum reduct, because the classification quality based on minimum reduct should be poor on unseen data for the noise or other peculiarities of the data set. Several strategies have been implemented to find the minimal decision rules with good performance, such as boundary region thinning [19], preserving up to a given threshold the positive region [20], entropy [21].

In this article we define a new criterion as the fitness function, and a multi-objective genetic algorithm is applied to choose the best reduct then.

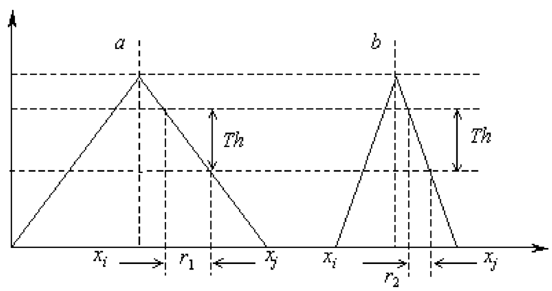


Fig. 1. Illustration of adaptive *Th* of Membership degree

Since the discernibility matrix M($S$) contained all the discerning information of objects, and a reduct can substitute the whole conditional attribute set without degrading the discrimination of decision rules, hence, the best reduct should cover the discernibility matrix as much as possible.

Furthermore, a reduct should have high approximation to the set $C$, and if any element of the reduct was absent, it is unable to make decisions with certainty. We use SGF($B$,$C$,$D$) to express how exactly the set $B$ approximates the set $C$.

$$\mathrm{SGF}(B,C,D)=\frac{\gamma(C,D)-\gamma(B,D)}{\gamma(C,D)}=1-\frac{\gamma(B,D)}{\gamma(C,D)} \quad (10)$$

where $B$ is a subset of $C$. If $B$ is a reduct of $C$, then $\gamma(B,D)=\gamma(C,D)$.

For the ideas discussed above, we define the fitness function with three items as given in (11).

$$F=\alpha_1/L(R)+\alpha_2\sum_{i=1}^{kl}s_i\Big/kl-\alpha_3\,\mathrm{SGF}(R,C,D) \quad (11)$$

where $L(R)$ is the numbers of attributes in the reduct. When there exists an intersection between the reduct and a certain individual of discernibility matrix, $s_i$ is set to 1, otherwise 0, and $kl$ is the number of the individuals in the matrix; SGF($R$,$C$,$D$) is the approximation of $R$ to $C$; $\alpha_1 \ldots \alpha_3$ are non-negative random weights less than 1. In order to gain simple decision rules with more knowledge, we evaluate $\alpha_1$ and $\alpha_2$ a larger number and $\alpha_3$ a smaller one.

The process of optimum searching is to maximize the former two items and minimize the last one. It is obvious a multi-objective problem, and fits to be solved with multi-objective genetic algorithm.

### *B. The process of Attribute Reduction*

Since the single element in the items of discernibility matrix is an element of core, and must be a member of a reduct. In the paper the single element is treated as a good gene to construct a reduct, and is included in an individual through the whole genetic process. Other attributes in a reduct are extracted from the items of more than one element.

All the items including the elements in core are removed from the matrix. Only comparing the individual with the remaining items in the matrix, whose number is small, can reduce the computational time of fitness. And a better searching ability of genetic algorithm is gained then.

The attribute reduction process has five steps.

Step 1. Generate discernibility matrix, calculate the core and remove the items including the core.

Step 2. Create random populations of $n$ individuals. Each individual contains all the attributes in core. All the individuals are represented as chromosome, each chromosome is a string of bits, 0 or 1, where 1 represents the attribute on that bit is included in the reduct, 0 otherwise.

Step 3. Create next generation with a larger fitness by using selection, crossover and mutation.

Step 4. Evaluate the fitness of each individual according to the given fitness function in (11).

Step 5. If a predefined number of generations is achieved, then return the best individual in current population and stop, else go to step 3.

For finding a simple reduct with a few attributes, we define two parameters in the mutation where a larger probability is assigned to the mutation from 1 to 0 than from 0 to 1. It can efficiently decrease the number of attributes in a reduct.

## IV. Rough Fuzzy Neural Network

### *A. Rough sets and Fuzzy Set*

Rough sets and fuzzy set are two important techniques for the problem of incomplete and uncertain information. They both can express the heuristic rules of knowledge base effectively, and have advantages and successful applications in many areas, such as pattern identification, machine learning, decision analyzing and knowledge discovering.

However rough sets and fuzzy set are different in some points. 1) They use two different concepts, the dependency relation and fuzzy, to manage incomplete data set. 2) They calculate the approximation by two different way the mathematic equations and the statistic techniques. 3) The ability of knowledge expression and inference of fuzzy set relies on the priori knowledge that is always redundant. Whereas rough sets have the advantage of eliminating unnecessary knowledge by attribute reduction, it is usable to classify and simplify information. 4) The calculation of rough sets is based on the fundamental concepts such as set approximation, positive region. It would lead to biases by the boundary problems. While fuzzy neural networks have good robustness and error-adaptability. So combining rough sets and fuzzy sets allows obtaining rough approximations of fuzzy sets as well as approximations of sets by means of fuzzy similarity relations [9].

*B. Rough Fuzzy Neural Network*

Now, rough sets applied in neural networks are used for decision support and knowledge discovering [10], [12]. Reference [11] attempts to use rough sets as a preprocessing tool for neural networks, [13], [14] attempts to construct and refine a neural network by using the knowledge from rough sets calculations. A knowledge-based neural network are proposed in [22] that has considered the domain knowledge during the network designing.

In this paper, combining with rough sets and fuzzy neural networks, a novel knowledge-based rough fuzzy neural network (RFNN) with three layers for load forecasting is proposed. The best reduct found in the attribute reduction algorithm is applied to generate decision rules that are implemented on the structure and initial weights designing of a RFNN.

We define two types of neurons in a rough fuzzy neural network, as depicted in Fig.2. One is the classical neuron used to manage the historical load; the other is the rough neuron used to manage the blurry, inexact discrete vectors, such as weather, types of days, random effects. In the following part, we mainly discuss the computation of rough neurons, for details of classical neurons one may refer to [4], [5].

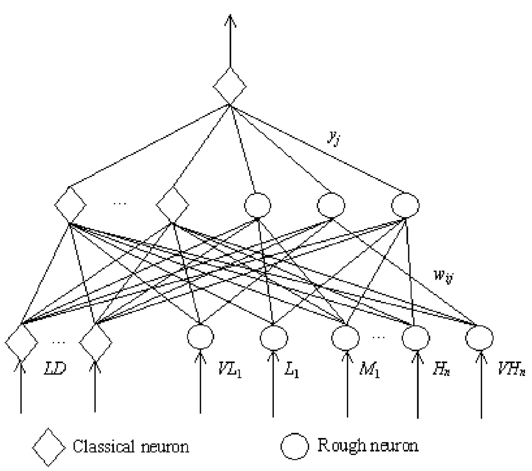


Fig. 2. Construction of Rough Fuzzy Neural Networks

In Fig.2 any input vector for weather factors is described in terms of some combination of membership values in the linguistic sets, i.e. very low (*VL*), low (*L*), Normal (*NM*), high (*H*) and very high (*VH*). *LD* denotes the inputs of electric loads, $y_j$ is the state of the *j*th neuron, $w_{ij}$ is a weight of the connection from the *i*th to the *j*th neuron.

There are 24 rough fuzzy neural networks implemented separately on hourly load forecasting in this article. By splitting the task into some subtasks and training the sub-networks parallelly, the method has the advantage of simplifying the task and super-linear speedup in training, and a sub-network has a more meaningful and clearer neural representation simultaneously.

*1) Structure of Rough Fuzzy Neural Network*

For $S=\langle U, C\cup\{d\}\rangle$, there exists $U\,|\,\mathrm{IND}(C)=\{X_1,\cdots,X_t\}$ and $U\,|\,\mathrm{IND}(d)=\{Y_1,\cdots,Y_s\}$. The statistical dependency relation between $X_i$ and $Y_j$ is measured by the mutual information in information theory, and it is defined as:

$$I(X_i;Y_j)=H(X_i)-H(X_i\,|\,Y_j) \tag{12}$$

where $i=1,\ldots,t$; $j=1,\ldots,s$.

Here, we defined a function *Gain* to measure the significance of the objects in $X_i$:

$$Gain(Y_j;x_l)=I(Y_j;X_i)-I(Y_j;X_i-\{x_l\}) \tag{13}$$

where $i=1,\ldots,t$; $j=1,\ldots,s$; $l=1,\ldots,|X_i|$. Those objects with maximal value of *Gain* are selected as the representative points, and then *s* objects are chose to form an $s\times n_R$ decision table based on the reduct, *s* is the number of classes of the decision attribute; $n_R$ is the number of attributes in the best reduct.

This $s\times n_R$-dimensional decision table is only used in the process of the structure designing and initial weights calculating. During the training phase, the network learns from the original $n\times n_R$ training set.

For each object $x_i$ in *U*, we consider the discernibility function $f_D^{x_i}$ that is defined as

$$f_D^{x_i}=\wedge\{\vee(c_{ij}):\ 1\le j\le i\le n, j\ne i, c_{ij}\ne\varnothing\}$$

$f_D^{x_i}$ is a conjunctive normal form (c.n.f.) of the predecessor of rule $r_i$: $\varphi_i \Rightarrow d_i=v_i$, where $d_i$ corresponds to the object $x_i$.

In RFNN, the number of the nodes of rough neurons in the hidden layer equals the number of items connected with conjuncts ( $\wedge$ ). Only those input attributes that appear in a disjunct ( $\vee$ ) are connected to the appropriate hidden node. A single attribute involving no disjuncts is directly connected to the appropriate output node via a hidden node. The number of nodes for classical neurons is specified while training.

Each conjunct is modeled at the output layer by joining the corresponding hidden nodes. Every neural network has one node in the output layer corresponding to one class (one decision rule). The output of the neural network is the forecasting load *LD* (*dt*,*t*) of day *dt* at time *t*.

*2) Connection Weights of Rough Fuzzy Neural Network*

Knowledge is treated as a partition of the universe in rough sets theory. It has granularity and is rough. The higher the granularity is, the smaller $\gamma$ becomes, and so many inconsistent rules will be generated, while the lower the granularity is, the higher $\gamma$ becomes, and the randomness of the rules increases. So the uncertainty of decision rules includes the inconsistency and the randomness. In this article consideration is given to both the inconsistency and the randomness of decision rules while calculating connection weights.

In information theory entropy is used to manage the uncertainty and the granularity of decision rules [23]. For $Q\subseteq C$ in $S=\langle U, C\cup\{d\}\rangle$, $\mathrm{IND}(Q)$is treated as a random variable on *U*, and $U\,|\,\mathrm{IND}(Q)=\{X_1,\cdots,X_t\}$. The entropy of variable IND($Q$) is defined as: $H(Q)=\sum_{i=1}^{t}p(X_i)\ln\frac{1}{p(X_i)}$, and the conditional entropy is $H(D\mid Q)=-\sum_{i=1}^{t}\sum_{j=1}^{s}p(X_i)p(Y_j\mid X_i)\ln p(Y_j\mid X_i)$, where $p(X_i)=|X_i|/|U|$, $p(Y_j\mid X_i)=|Y_j\cap X_i|/|Y_j|$. They are the probability and conditional probability of $X_i$.

According to the definitions of entropy and the dependency factor $\gamma$, the connection weights for rough neurons are specified.

The weight between a hidden node ($i$) and an output node ($k$) is $w_{kj}^{1}=\left(\gamma_\alpha+h^*\right)/numc+\varepsilon$, where $\gamma_\alpha$ is the dependency factor, $\gamma_\alpha=|\mathrm{POS}_R(D)|/|U|$, $R\subseteq C$; $h^*=h(R)+h(D\mid R)$, $h^*$ scales the uncertainty of the decision rule. $h(R)$ describes the complexity of the partition $U$/IND($R$); $h(D|R)$ manages the uncertainty; *numc* (*numc*≥1) is the number of conjuncts in a rule, and $\varepsilon$ is a small random number.

The weight between a fuzzy attribute $a_i$ and a hidden node $j$ is $w_{ja_i}^{0}=\gamma_\beta/numd+\varepsilon$, where $\gamma_\beta$ is the initial weight between the hidden nodes and the output nodes. *numd* (*numd*≥1) is the number of attributes connected by disjuncts.

The sign of the weight is set to positive (or negative) if the corresponding entry of M ($S$) in row $k$ and column $a_i$ is larger than $Th$ (or less than $Th$). All the other possible connection weights in the neural network are specified as a random number between 0 and 1. During training, the weights are updated by backpropagating errors, such that the contribution of uncertain vectors is automatically reduced.

## V. Simulation Results

In this section, we evaluate the performance of the proposed method by applying it to a real power system to forecast workday loads. The historical data including the load and weather data in 2001 are used as the training sets, and the data from Jan. to June in 2002 are selected randomly as test sets.

In a decision table, weather factors including wind speed ($Ws$), maximum and minimum temperature ($T_{max}$ and $T_{min}$), mean humidity ($Hu$) and rainfall ($Rf$) on day $dt$, $dt$-1, $dt$-2, $dt$-7 and $dt$-14 are chosen as the conditional attributes; loads of day $dt$ is treated as the decision condition, $dt$ denotes the day on which we want to have load forecasting. The decision table has 125 conditional attributes and one decision attribute, and the threshold $Th$=0.5, the classes $s$=7.

After using the attribute reduction algorithm, the total number of the attributes decreases from 125 to 32, which are just correlated directly to the forecasting target. The selected attributes are enumerated in Table I. The subscribe 1, 2, 3, 4, 5 in an item represent the linguistic sets $VL$, $L$, $NM$, $H$ and $VH$ respectively, such that $T_{max2}$ denotes $T_{max}=L$.

When the forecasted weather information is available, the conditions on each hour are classified into one of the 7 classes by the fuzzy rules, and then the corresponding ANN for that class is selected automatically by the system, and the necessary information of that hour is sent to the ANN to forecast the load.

In order to testify the effectiveness of the proposed methods in the paper, we compare the proposed rough neural network (RFNN) model based on attributes reduction with the traditional fuzzy neural network (FNN) and regression model for short-term load forecasting.

TABLE I
The Correlative Attributes Selected by Attribute Reduction

| Day | Correlative Attributes |
|---|---|
| $dt$ | $T_{max2}$, $T_{max3}$, $T_{min1}$, $T_{min2}$, $T_{min3}$, $Hu_3$, $Hu_4$ |
| $dt$-1 | $T_{max2}$, $T_{max3}$, $T_{max4}$, $T_{min1}$, $T_{min2}$, $Rf_4$, $Rf_5$, $Ws_3$, $Ws_4$ |
| $dt$-2 | $T_{max4}$, $T_{max5}$, $Rf_3$, $Rf_4$, $Rf_5$, $Hu_1$, $Hu_2$ |
| $dt$-7 | $Hu_3$, $Hu_4$, $Rf_2$, $Rf_4$, $Ws_1$ |
| $dt$-14 | $T_{max2}$, $T_{max3}$, $T_{min1}$, $T_{min2}$ |

Every rough fuzzy neural network corresponds to one class. Let the historical loads of $dt$-1, $dt$-2, $dt$-7 and $dt$-14 at time $t$-1, $t$, $t$+1 be the input vector of conventional neurons. Weather factors of $dt$, $dt$-1, $dt$-2, $dt$-7 and $dt$-14 corresponding to the best reduct are treated as the inputs of rough neurons. All the outputs of the 24 sub-networks compose the final forecasting load of day $dt$.

In Table II the prediction error of the proposed method compared with that of regression model and FNN are presented, where the numbers represent the mean errors of MAPE. It is good to mention that our method has gained better performances. The fairly good performance of the maximum prediction error of FNN is 3.877%, however RFNN has yielded a better performance of prediction error lower than 3.240%. The best precision of RFNN is 1.871 in April, and better than 1.903 and 1.899 of FNN and regression model respectively. Although regression model has an advantage of fast and accurate forecast ability, it has a bad performance for weather-sensitive load. The forecast errors in Feb. and June of regression model are 4.361 and 3.141 that are evidently lower than FNN and RFNN. The forecasted load curves of FNN and RFNN on April 16, 2002 compared with actual load curve are illustrated in Fig.3. From this figure, we can observe clearly that the load curve of RFNN is closer to the actual loads than traditional FNN, so good prediction accuracy is obtained by RFNN.

After the input vectors, the links and the weights of neural network are preprocessed by rough sets, unnecessary information are eliminated, the inherent relations in information are dug out at the same time, and various information resources are utilized sufficiently. The accuracy of rule reasoning is heightened and the accuracy of load forecasting is improved.

Furthermore, with the attribute reduction and network simplification in RFNN the links of the network becomes sparser than the full connection of the traditional neural network, the structure of the neural network becomes simpler and clearer. As we can see from the results, a new transparent and compact model for load forecasting is gained. And it reaches the needed precision of load forecasting, and outperforms the conventional method.

TABLE II
COMPARISON FOR RFNN, FNN AND REGRESSION MODEL BY MAPE

| Month | Jan. | Feb. | Mar. | Apr. | May | Jun. |
|---|---|---|---|---|---|---|
| Regression | 2.488 | 4.361 | 1.622 | 1.899 | 1.752 | 3.141 |
| FNN | 1.334 | 3.877 | 1.477 | 1.903 | 1.846 | 2.781 |
| RFNN | 1.214 | 3.240 | 1.458 | 1.871 | 1.617 | 2.133 |

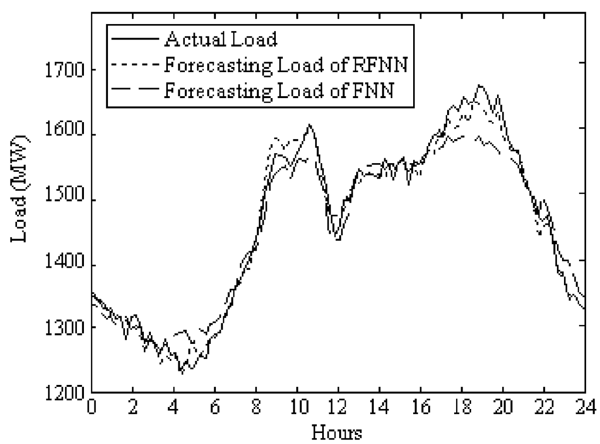


Fig. 3. Comparison between actual and forecasting load curves of FNN and RFNN

## VI. CONCLUSION

A new hybrid knowledge-based model RFNN integrating with rough sets theory and fuzzy neural networks has been presented for short-term load forecasting in this paper. It has been tested and compared with conventional FNN and regression model in a real power system. The results show that the proposed rough fuzzy neural network for load forecasting provides a more accurate and effective forecasting, and this paper shows that the proposed method is promising for load forecasting in power systems.

## VII. REFERENCES

[1] A.D.Papalexopoulos and T.C.Hesterberg. A regression-based approach to short term system load forecasting, IEEE Transaction on Power Systems, 5(4), 1990, 1535-1547.

[2] A.Bakirtzis, V. Petridis, S. Klartzis, M. Alexiadis, and A. Maissis. A neural network short-term load forecsting model for the greek power system, IEEE Trans. On Power Systems, 11(2), 1996, 858-863.

[3] O. Mohammed, D. Park, R. Merchant, T. Dinh, C. Tong, A. Azeem, J. Farah, and C. Drake. Practical experiences with an adaptive neural network short-term load forecasting system, IEEE Trans. On Power Systems, 10(1), 1995, 254-265.

[4] S.E.Papadakis, J.B.Theocharis. A novel approach to short-term load forecasting using fuzzy neural networks, IEEE Transactions on Power Systems, 13(2), 1998, 480-493.

[5] James W. Taylor and Roberto Buizza. Neural network load forecasting with weather ensemble predictions, IEEE Transactions on Power Systems, 17(3), 2002, 626-633.

[6] Ansarimer, P., Barghinia, I., Habibi, H., et, al. Short term load forecasting for Iran national power system using artificial neural network and fuzzy expert system, Proc. on Powercon, Kuming China, 2002, 1082-1085.

[7] H. Mori. State-of-the-art overview on artificial neural networks in power systems, In M. El-Sharkawi and D. Niebur, editors, A Turorial Course on Artificial Neural Networks with Applications to Power Systems, chapter 6, pages 51-70. IEEE Power Engineering Society, 1996.

[8] A. Khotanzad, R. Hwang, A. Abaye, and D. Maratukulam. An adaptive modular artificial neural hourly load forecaster and its implementation at electric utilities, IEEE Trans. on Power Systems, 10(3), 1995, 1716-1722.

[9] R.Slowinski, Ed. Intelligent Decision Support-Handbook of applications and advances of the rough sets theory (Dordrecht, Netherlands: Kluwer, 1992).

[10] A.Czyzewski and A. Kaczmarek, Speech recognition systems based on rough sets and neural networks, Proc. 3rd Wkshp. Rough Sets and Soft Computing, San Jose, CA, 1994, 97-100.

[11] Qiu-Dan Li, Zhong-Xian Chi, Wen-Bing Shi. Application of rough set theory and artificial neural network for load forecasting, Proc. the First International Conference on Machine Learning and Cybernetics, Beijing, China, 2002, 1148-1152.

[12] Lambert-Torres, G. Application of rough sets in power system control center data mining, Pro. IEEE Power Engineering Society Winter Meeting, New York, NY, 2002, 627 –631.

[13] Mohua Banerjee, Sushmita Mitra, and Sanka.K.Pal. Rough Fuzzy MLP: knowledge encoding and classification, IEEE Transactions on neural networks, 9(6), 1998, 1203-1217.

[14] Sankar K. Pal, Sushmita Mitra. Rough Fuzzy MLP: Modular Evolution, Rule Generation and Evaluation, IEEE Transactions on Knowledge and Data Engineering, 15(1), 2003, 14–25.

[15] Z. Pawlak. Rough Sets: Theoretical Aspects of Reasoning about Data (Norwell, Mass: Kluwer, 1991).

[16] Jan Komorowski, Zdzislaw Pawlak, Lech Polkowski, Andrzej Skowron, "Rough Sets: A Tutorial," [Online]. pp. 7-12. Available: http://folli.loria.fr/cds/1999/library/pdf/skowron.pdf

[17] J. Wróblewski. Genetic algorithms in decomposition and classification problems, Rough Sets in Knowledge Discovery 2: Applications, Case Studies and Software Systems. (Polkowski and Skowron (Eds.), Heidelberg: Physica-Verlag, 1998)

[18] J. Wróblewski. Finding minimal reducts using genetic algorithms, Proc. Second Annual Joint Conference on Information Sciences, Wrightsville Beach, NC, 1995, 186-189.

[19] A. Skowron. Synthesis of adaptive decision systems from experimental data, Proc. the Fifth Scandinavian Conference on Artificial Intelligence and Applications, Trondheim, Norway, 1995, 220-238.

[20] W.Ziarko. Variable precision rough set model, Computer and System Sciences, 46, 1993, 39-59.

[21] D. Ślęzak. Approximate reducts in decision tables, Proc the Sixth Conference on Information Processing and Management of Uncertainty in Knowledge-Base Systems, Granada, Spain, 1996, 1159-1164.

[22] G. G. Towell and J. W. Shavlik. Knowledge-based artificial neural network, Artificial Intelligence, 70(1), 1994, 119-165.

[23] Düntsch I, Gediga G. Uncertainty measures of rough set prediction, Artificial Intelligence, 106(1), 1998, 109-137.

## VIII. BIOGRAPHIES

**Li Feng** received her B.S. and M.S. degrees in electrical engineering from Guizhou University of technology, China, in 1998 and 2001 respectively, and her Ph.D. degree in electrical engineering from Zhejiang University, China, in 2005. She is now with the Dispatch and Transaction Center, Chongqing Electrical Power Corporation. Her current research interests are data mining, electrical market analysis, intelligent control in power system.

**Ziyan Liu** received her B.S. and M.S. degrees in electrical engineering from Guizhou University of technology, China, in 1997 and 2000 respectively. She is now with College of Computer Science and Communication Technology, Guizhou University. Her current research interests are communication engine, intelligence control, microcomputer control and data mining.