# A New Approach for Prediction by using Integrated Neural Networks

**Hazem M. El-Bakry**
Faculty of Computer Science & Information Systems,
Mansoura University, EGYPT
E-mail: helbakry20@yahoo.com

**Nikos Mastorakis**
Technical University of Sofia,
BULGARIA

Abstract- In this paper, a new neural model is presented. Fast Feedforwared Neural Networks (FFNNs) is integrated with modified recurrent neural networks for powerful estimation. The proposed new model is applied for prediction of power consumption. First, Modified Kohonen's Neural Networks (MKNNs) are used to facilitate the prediction process because they have the ability for clustering the input space into a number of classes. Therefore it is used for data classification to identify the categories which are essential for the prediction process. The unsupervised process performs the role of front-end data compression. For each category, the supervised learning algorithm LVQ is used for training FFNNs. The operation of FFNNs relies on performing cross correlation in the frequency domain between the input data and the weights of neural networks. Simulation results have shown that the presented integrated neural model is very powerful prediction.

Keywords- FFNNs, MKNNs, LVQ, Cross Correlation, Frequency Domain and Prediction

## I. Introduction

A prediction is a statement about the way things will happen in the future, often but not always based on experience or knowledge. While there is much overlap between *prediction* and *forecast*, a *prediction* may be a statement that some outcome is expected, while a *forecast* may cover a range of possible outcomes. Although guaranteed information about the information is in many cases impossible, prediction is necessary to allow plans to be made about possible developments; Howard H. Stevenson writes that prediction in business "... is at least two things: Important and hard [53]. Prediction is closely related to uncertainty. Reference class forecasting was developed to eliminate or reduce uncertainty in prediction [54].

Outside the rigorous context of science, prediction is often confused with informed guess or opinion. A prediction of this kind might be (inductively) valid if the predictor is a knowledgeable person in the field and is employing sound reasoning and accurate data. Large corporations invest heavily in this kind of activity to helpfocus attention on possible events, risks and business opportunities, using futurists. Such work brings together all available past and current data, as a basis to develop reasonable expectations about the future.Predictions have often been made, from antiquity until the present, by using paranormal or supernatural means such as prophecy or by observing omens. Methods including water divining, astrology, numerology, fortune telling, interpretation of dreams, and many other forms of divination, have been used for millennia to attempt to predict the future. These means of prediction have not been substantiated by controlled experiments, and are disputed by most, including scientists and skeptics.

In politics it is common to attempt to predict the outcome of elections via political forecasting techniques (or assess the popularity of politicians) through the use of opinion polls. Prediction games have been used by many corporations and governments to learn about the most likely outcome of future events. Predictions have often been made, from antiquity until the present, by using paranormal or supernatural means such as prophecy or by observing omens. Methods including water divining, astrology, numerology, fortune telling, interpretation of dreams, and many other forms of divination, have been used for millennia to attempt to predict the future. These means of prediction have not been substantiated by controlled experiments, and are disputed by most, including scientists and skeptics.

In science a prediction is a rigorous, often quantitative, statement, forecasting what will happen under specific conditions; for example, if an apple falls from a tree it will be attracted towards the center of the earth by gravity with a specified and constant acceleration. The scientific method is built on testing assertions that are logical consequences of scientific theories. This is done through repeatable experiments or observational studies. A scientific theory whose assertions are contradicted by observations and evidence will be rejected. Notions that make no *testable* predictions are usually considered not to be part of science (protoscience or nescience) until testable predictions can be made. New theories that generate many new predictions can more easily be supported or falsified (see predictive power). In some cases the probability of an outcome, rather than a specific outcome, can be predicted, for example in much of quantum physics.Mathematical equations and models, and computer models, are frequently used to describe the past and future behaviour of something. In microprocessors, branch prediction permits avoidance of pipeline emptying at branch instructions. In engineering, possible failure modes are predicted and avoided by correcting the mechanism causing the failure.Accurate prediction and forecasting are very difficult in some areas, such as software reliability, natural disasters, pandemics, demography, population dynamics and meteorology.

Established science makes useful predictions which are considered to be extremely reliable and accurate; for example, eclipses are routinely predicted. New theories make predictions which allow them to be falsified if the predictions are not borne out. For example in the early twentieth century the scientific consensus was that there was an absolute frame of reference, given the name *luminiferous ether*. The famous Michelson-Morley experiment ruled this out, falsifying the idea of an absolute frame and leaving the very counter-intuitive special theory of relativity as the only possibility.

Albert Einstein's theory of general relativity could not easily be tested as it did not produce any effects observable on a terrestrial scale. However, the theory predicted that large masses such as stars would bend light, in contradiction to accepted theory; this was observed in a 1919 eclipse.Mathematical models of stock market behaviour are also unreliable in predicting future behaviour. Consequently, stock investors may anticipate or predict a stock market boom, or fail to anticipate or predict a stock market crash. Some correlation has been seen between actual stock market movements and prediction data from large groups in surveys and prediction games.

An actuary uses actuarial science to assess and predict future business risk, such that the risk(s) can be mitigated. For example, in insurance an actuary would use a life table to predict (truly, estimate or compute) life expectancy. In literature, vision and prophecy are literary devices used to present a possible timeline of future events. They can be distinguished by vision referring to what an individual sees happen. The New Testament book of Revelation (Bible) thus uses vision as a literary device in this regard. It is also prophecy or prophetic literature when it is related by an individual in a sermon or other public forum. Fiction (especially fantasy, forecasting and science fiction) often features instances of prediction achieved by unconventional means.

- In fantasy literature, predictions are often obtained through magic or prophecy, sometimes referring back to old traditions. For example, in J. R. R. Tolkien's *The Lord of the Rings*, many of the characters possess an awareness of events extending into the future, sometimes as prophecies, sometimes as more-or-less vague 'feelings'. The character Galadriel, in addition, employs a water "mirror" to show images, sometimes of possible future events.
- In some of Philip K. Dick's stories, mutant humans called *precogs* can foresee the future (ranging from days to years). In the story called *The Golden Man*, an exceptional mutant can predict the future to an indefinite range (presumably up to his death), and thus becomes completely non-human, an animal that follows the predicted paths automatically. Precogs also play an essential role in another of Dick's stories, *The Minority Report*, which was turned into a film by Steven Spielberg in 2002.
- In the *Foundation* series by Isaac Asimov, a mathematician finds out that historical events (up to some detail) can be theoretically modelled using equations, and then spends years trying to put the theory in practice. The new science of psychohistory founded upon his success can simulate history and extrapolate the present into the future.
- In Frank Herbert's sequels to Dune, his characters are dealing with the repercussions of being able to see the possible futures and select amongst them. Herbert sees this as a trap of stagnation, and his characters follow a Golden Path out of the trap.
- In Ursula K. Le Guin's *The Left Hand of Darkness*, the humanoid inhabitants of planet Gethen have mastered the art of prophecy and routinely produce data on past, present or future events on request. In this story, this was a minor plot device.

FFNNs for detecting a certain code in one dimensional serial stream of sequential data were described in [10,27]. Compared with traditional feedforward neural networks (TFNNs), FFNNs based on cross correlation between the tested data and the input weights of neural networks in the frequency domain showed a significant reduction in the number of computation steps required for certain data detection [5-35]. Here, we make use of the theory of FFNNs implemented in the frequency domain to increase the speed of neural networks for prediction of power consumption. The idea of moving the testing process from the time domain to the frequency domain is applied to time delay neural networks. Theoretical and practical results show that the proposed FFNNs are faster than FFNNs.

This paper is organized as follows. Prediction of power consumption by using ANNs is discussed in section 2. The Theory of FFNNs is described in section 3. The proposed approach for prediction is presented in section 4. Finally conclusions are given.

## 2. Prediction of Power Consumption by using ANNs

Artificial neural network (ANNs) is a mathematical model, which can be set one or more layered and occurred from

many artificial neural cells. The wide usage of the ANN may be due to the three basic properties:
1. The ability of the ANNs as a parallel processing of the problems, for which if any of the neurons violate the constraints would not affect the overall output of the problem.
2. The ability of the ANNs to extrapolate from historical data to generate forecasts.
3. The successful application of the ANN to solve non-linear problems. The history and theory of the ANN have been described in a large number of published literatures and will not be covered in this paper except for a very brief overview of how neural networks operate.

The ANN computation can be divided into two phases: learning phase and testing phase. The learning phase forms an iterative updating of the synoptic weights based upon the error back propagation algorithm. Back propagation algorithm is generalized of least mean square learning rule, which is an approximation of steepest descent technique. To find the best approximation, multi-layer feed forward neural network architecture with back propagation learning rule is used. A schematic diagram of typical multi-layer feed-forward traditional neural network architecture. The number of neurons in the hidden layer is varied to give the network enough power to solve the forecasting problem. Each neuron computes a weighted sum of the individual inputs ($I_1, I_2, ..., Ij$) it receives and adding it with a bias (*b*) to form the net input (*x*). The bias is included in the neurons to allow the activation function to be offset from zero.

ANNs is a computing system that imitates intelligent behavior and processes information by its dynamic state response to external inputs. The neural network has two processes; learning and classification. There are two types of learning; supervised and unsupervised [1-5]. In the supervised learning process the network is supplied with a pair of patterns, an input pattern and a corresponding target output pattern. In the classification process, the user provides a pattern of system description parameters to the neural network and it returns an estimate of the output pattern. The unsupervised learning's used for clustering the input space into affinity number of classes represented by the neural network weights vectors. A principal challenge in neural computing is adjusting the set of internal parameters known as weights in order to encode an underlying relation assumed to exist amongst various data sets.

The power consumption (Gas, Electricity, Water) demand is influenced by many factors, such as weather, economic and social activities and different power consumption components (residential, industrial, commercial etc.). By analysis of only historical power consumption data, it is difficult to obtain accurate power consumption demand for prediction. The relation between power consumption demand and the independent variables is complex and it is not always possible to fit the power consumption curve using statistical models [6]. The daily power consumption patterns in the same geographical area have been repeated for the same day type in the same season so, the ANNs approach was proposed for power consumption prediction [7-13].

Sophisticated algorithms more or less rely on presumptions on influences like temperature, humidity, wind speed, cloudiness, and rain. Although most of these information might be taken and updated (automatically) from the statistics applied to recorded data, the underlying power consumption models inherited in power consumption prediction algorithms make traditional approaches inflexible against major changes in the consumer behavior. In order to overcome these problems, AI techniques have been applied to power consumption prediction. The ANNs approach has several key features that make it highly suitable for power consumption prediction [12]. For example,

- It does not require any presumed functional relationship between electric power consumption and other variables such as weather conditions.
- It provides a nonlinear mapping between weather variables and previous power consumption patterns, and electric power consumption with out the need for predetermined model.
- It is usually fault tolerant and robust.

ANNs proved to be capable of finding internal representations of interdependencies within raw data not explicitly given or even known by human experts. This typical characteristic together with the simplicity of building and training ANNs and their very short response time encouraged various groups of researchers to apply ANNs to the task of power consumption prediction. Most of the papers [6-13] present feasibility studies carried out at universities or research institutes often in cooperation with utilities. They mainly addresses peak power consumption prediction, total power consumption prediction, and hourly power consumption prediction with lead time from 1 to 48 hours. Typically ANNs map input data to predict the value of power consumption. Therefore, FFNNs are used. It has been proved that these networks are very efficient in many different applications [1-37]. The self-organizing feature map is used by [8] which clusters the prediction data to clusters with similar power consumption profiles. There is no paper describing an ANNs concept coping with the task of e.g. hourly power consumption prediction for all days of the year. In this paper, a generalized approach based on ANNs for prediction process is presented. An Information system that Implement the generalized approach is developed. MKNNs is used for data classification to identify the day classes/types which are essential for prediction processes.

Due to the dynamic nature of hourly power consumptions and differences in power consumption characteristics from region to region, it is required to analysis power consumption data for each region separately and to design suitable neural networks for prediction process for that region. Power consumption analysis is performed to identify the day type/classes of that region. In this work, it is required to collect the effective hourly data of the power system and save them in a dynamic database. Once the database is developed, their information can be update, delete, display, and insert. Also, from database we can identify the day types/classes of daily power consumption patterns using MKNNs.

ISBN: 978-960-474-270-7

## 3. Theory of FFNNs for Prediction

Computing the resulted output; for a certain pattern of information; in the incoming serial data, is a prediction problem. First neural networks are trained to predict the estimated variable and this is done in time domain. In pattern detection phase, each position in the incoming matrix is processed to predict the estimated variable by using neural networks. At each position in the input one dimensional matrix, each sub-matrix is multiplied by a window of weights, which has the same size as the sub-matrix. The outputs of neurons in the hidden layer are multiplied by the weights of the output layer. Thus, we may conclude that the whole problem is a cross correlation between the incoming serial data and the weights of neurons in the hidden layer. The convolution theorem in mathematical analysis says that a convolution of f with h is identical to the result of the following steps: let F and H be the results of the Fourier Transformation of f and h in the frequency domain. Multiply F and H* in the frequency domain point by point and then transform this product into the spatial domain via the inverse Fourier Transform. As a result, these cross correlations can be represented by a product in the frequency domain. Thus, by using cross correlation in the frequency domain, speed up in an order of magnitude can be achieved during the detection process [5-35]. Assume that the size of the attack code is 1xn. In attack detection phase, a sub matrix I of size 1xn (sliding window) is extracted from the tested matrix, which has a size of 1xN. Such sub matrix, which may be an attack code, is fed to the neural network. Let $W_i$ be the matrix of weights between the input sub-matrix and the hidden layer. This vector has a size of 1xn and can be represented as 1xn matrix. The output of hidden neurons h(i) can be calculated as follows [5]:

$$h_i = g\left(\sum_{k=1}^{n} W_i(k)I(k) + b_i\right) \qquad (1)$$

where g is the activation function and b(i) is the bias of each hidden neuron (i). Equation 1 represents the output of each hidden neuron for a particular sub-matrix I. It can be obtained to the whole input matrix Z as follows [5]:

$$h_i(u)=g\left(\sum_{k=-n/2}^{n/2} W_i(k)\ Z(u+k)+b_i\right) \qquad (2)$$

Eq.1 represents a cross correlation operation. Given any two functions f and d, their cross correlation can be obtained by [4]:

$$d(x)\otimes f(x)=\left(\sum_{n=-\infty}^{\infty} f(x+n)d(n)\right) \qquad (3)$$

Therefore, Eq. 2 may be written as follows [5]:

$$h_i = g\left(W_i \otimes Z + b_i\right) \qquad (4)$$

where $h_i$ is the output of the hidden neuron (i) and $h_i$ (u) is the activity of the hidden unit (i) when the sliding window is located at position (u) and (u) $\in$ [N-n+1].

Now, the above cross correlation can be expressed in terms of one dimensional Fast Fourier Transform as follows [5]:

$$W_i \otimes Z = F^{-1}\left(F(Z)\bullet F^*\left(W_i\right)\right) \qquad (5)$$

Hence, by evaluating this cross correlation, a speed up ratio can be obtained comparable to traditional neural networks. Also, the final output of the neural network can be evaluated as follows:

$$O(u) = g\left(\sum_{i=1}^{q} W_o(i)\, h_i(u) + b_o\right) \qquad (6)$$

where q is the number of neurons in the hidden layer. O(u) is the output of the neural network when the sliding window located at the position (u) in the input matrix Z. $W_o$ is the weight matrix between hidden and output layer.

The complexity of cross correlation in the frequency domain can be analyzed as follows:

1- For a tested matrix of 1xN elements, the 1D-FFT requires a number equal to $N\log_2 N$ of complex computation steps [13]. Also, the same number of complex computation steps is required for computing the 1D-FFT of the weight matrix at each neuron in the hidden layer.

2- At each neuron in the hidden layer, the inverse 1D-FFT is computed. Therefore, q backward and (1+q) forward transforms have to be computed. Therefore, for a given matrix under test, the total number of operations required to compute the 1D-FFT is $(2q+1)N\log_2 N$.

3- The number of computation steps required by FFNNs is complex and must be converted into a real version. It is known that, the one dimensional Fast Fourier Transform requires $(N/2)\log_2 N$ complex multiplications and $N\log_2 N$ complex additions [3]. Every complex multiplication is realized by six real floating point operations and every complex addition is implemented by two real floating point operations. Therefore, the total number of computation steps required to obtain the 1D-FFT of a 1xN matrix is:

$$\rho=6((N/2)\log_2 N) + 2(N\log_2 N) \qquad (7)$$

which may be simplified to:

$$\rho=5N\log_2 N \qquad (8)$$

4- Both the input and the weight matrices should be dot multiplied in the frequency domain. Thus, a number of complex computation steps equal to qN should be considered. This means 6qN real operations will be added to the number of computation steps required by FFNNs.

5- In order to perform cross correlation in the frequency domain, the weight matrix must be extended to have the same size as the input matrix. So, a number of zeros = (N-n) must be added to the weight matrix. This requires a total real number of computation steps = q(N-n) for all neurons. Moreover, after computing the FFT for the weight matrix, the conjugate of this matrix must be obtained. As a result, a real number of computation steps = qN should be added in order to obtain the conjugate of the weight matrix for all neurons. Also, a number of real computation steps equal to N is required to create butterflies complex numbers ($e^{-jk(2\Pi n/N)}$), where 0<K<L. These (N/2) complex numbers are multiplied by the elements of the input matrix or by previous complex numbers during the computation of FFT. To create a complex number requires two real floating point operations. Thus, the total number of computation steps required for FFNNs becomes:

$$\sigma=(2q+1)(5N\log_2 N)+6qN+q(N-n)+qN+N \quad (9)$$

which can be reformulated as:

$$\sigma=(2q+1)(5N\log_2 N)+q(8N-n)+N \quad (10)$$

6- Using sliding window of size 1xn for the same matrix of 1xN pixels, q(2n-1)(N-n+1) computation steps are required when using TFNNs for certain attack detection or processing (n) input data. The theoretical speed up factor η can be evaluated as follows:

$$\eta=\frac{q(2n-1)(N-n+1)}{(2q+1)(5N\log_2 N)+q(8N-n)+N} \quad (11)$$

Time delay neural networks accept serial input data with fixed size (n). Therefore, the number of input neurons equals to (n). Instead of treating (n) inputs, the proposed new approach is to collect all the incoming data together in a long vector (for example 100xn). Then the input data is tested by time delay neural networks as a single pattern with length L (L=100xn). Such a test is performed in the frequency domain as described before.

The theoretical speed up ratio for searching short successive (n) code in a long input vector (L) using time delay neural networks is listed in tables 1, 2, and 3. Also, the practical speed up ratio for manipulating matrices of different sizes (L) and different sized weight matrices (n) using a 2.7 GHz processor and MATLAB is shown in table 4.

An interesting point is that the memory capacity is reduced when using FFNNs. This is because the number of variables is reduced compared with TFNNs.

## 4. The Proposed Technique for Prediction of Power Consumption

From the analysis of the prediction methods [6-11], a generalized approach for prediction using ANNs is concluded. The steps of the generalized approach for prediction process using ANNs can be described as:-

***Phase 1: Data Collection***

Form a historical database that contains the data of the attributes of the prediction process (input-output) that cover enough period immediately preceding the current time.

***Phase 2: Data Classification***

- Classify the historical database into groups according to a certain criteria (day type, season, similarity,…)
- Reject the redundant and inconsistent records from the database.
- Identify the prediction parameters (period, class, …).

***Phase 3: Training and testing***

Identify the initial design of FFNNs that requires to determine the following parameters ( no of input, output, hidden neurons, no of hidden layers, activation function, learning rate and momentum rate, and no of iterations).

- Form training and test sets of patterns for a class or day type from the database.
- Normalize the training and test sets.
- Train the neural network using the suitable learning algorithm.
- Test the trained FFNNs
- Calculate the testing performance measures of the FFNNs ( absolute percentage error , average error and standard deviation)
- If the performance measures are not accepted then change the FFNNs design parameters and repeat phase 3.

***Phase 4: Saving and updating***

- Save the parameters of the trained FFNNs and its weights (Final ANNs design).
- Update the historical database with the current recorded values.

In this paper, Phase1 and 2 is developed. The output of phase 1 is the historical database that contains recorded hourly power consumptions, temperatures, humidity, atmospheric pressure. The database is used to form the training patterns for the MKNNs. Phase 2 is performed with MKNNs. The output of phase 2 is the classes of data patterns that corresponding to day types classification. Phase 3 and 4 are presented in the second part of this work. An extra-large supervised learning system and a hierarchical system [10 ] are shown in Figs. 1 and 2. Fig.1 illustrates a situation where in almost every cluster there are a few local regions and separate supervised learning procedures are necessary. In Fig.2 large numbers of training set patterns are then sent forward to a supervised

Table 1: The theoretical speed up ratio for prediction of power consumption (n=400).

| Length of serial data | Number of computation steps required for TFNNs | Number of computation steps required for FFNNs | Speed up ratio |
|---|---|---|---|
| 10000 | 2.3014e+008 | 4.2926e+007 | 5.3613 |
| 40000 | 0.9493e+009 | 1.9614e+008 | 4.8397 |
| 90000 | 2.1478e+009 | 4.7344e+008 | 4.5365 |
| 160000 | 3.8257e+009 | 8.8219e+008 | 4.3366 |
| 250000 | 5.9830e+009 | 1.4275e+009 | 4.1912 |
| 360000 | 8.6195e+009 | 2.1134e+009 | 4.0786 |
| 490000 | 1.1735e+010 | 2.9430e+009 | 3.9876 |
| 640000 | 1.5331e+010 | 3.9192e+009 | 3.9119 |

Table 2: The theoretical speed up ratio for prediction of power consumption (n =625).

| Length of serial data | Number of computation steps required for TFNNs | Number of computation steps required for FFNNs | Speed up ratio |
|---|---|---|---|
| 10000 | 3.5132e+008 | 4.2919e+007 | 8.1857 |
| 40000 | 1.4754e+009 | 1.9613e+008 | 7.5226 |
| 90000 | 3.3489e+009 | 4.7343e+008 | 7.0737 |
| 160000 | 0.5972e+010 | 8.8218e+008 | 6.7694 |
| 250000 | 0.9344e+010 | 1.4275e+009 | 6.5458 |
| 360000 | 1.3466e+010 | 2.1134e+009 | 6.3717 |
| 490000 | 1.8337e+010 | 2.9430e+009 | 6.2306 |
| 640000 | 2.3958e+010 | 3.9192e+009 | 6.1129 |

Table 3: The theoretical speed up ratio for prediction of power consumption (n =900).

| Length of serial data | Number of computation steps required for TFNNs | Number of computation steps required for FFNNs | Speed up ratio |
|---|---|---|---|
| 10000 | 4.9115e+008 | 4.2911e+007 | 11.4467 |
| 40000 | 2.1103e+009 | 1.9612e+008 | 10.7600 |
| 90000 | 4.8088e+009 | 4.7343e+008 | 10.1575 |
| 160000 | 0.8587e+010 | 8.8217e+008 | 9.7336 |
| 250000 | 1.3444e+010 | 1.4275e+009 | 9.4178 |
| 360000 | 1.9381e+010 | 2.1134e+009 | 9.1705 |
| 490000 | 2.6397e+010 | 2.9430e+009 | 8.9693 |
| 640000 | 3.4493e+010 | 3.9192e+009 | 8.8009 |

Table 4: Practical speed up ratio for prediction of power consumption.

| Length of serial data | Speed up ratio (n=400) | Speed up ratio (n=625) | Speed up ratio (n=900) |
|---|---|---|---|
| 10000 | 8.94 | 12.97 | 17.61 |
| 40000 | 8.60 | 12.56 | 17.22 |
| 90000 | 8.33 | 12.28 | 16.80 |
| 160000 | 8.07 | 12.07 | 16.53 |
| 250000 | 7.95 | 17.92 | 16.30 |
| 360000 | 7.79 | 11.62 | 16.14 |
| 490000 | 7.64 | 11.44 | 16.00 |
| 640000 | 7.04 | 11.27 | 15.89 |

ISBN: 978-960-474-270-7

training phase. In fact, an unsupervised process performs the role of front-end data compression. The unsupervised/supervised learning concept relieves the neural network of the burden of trying to associate many dissimilar patterns with the same output. If two groups, which are not alike, happen to end up in the same cluster, the message is that either pattern description is not adequate or vigilance factor is not adequately stringent. Using this concept, the supervised learning process is carried out on the cluster-wise data structure rather than on the entire data set.

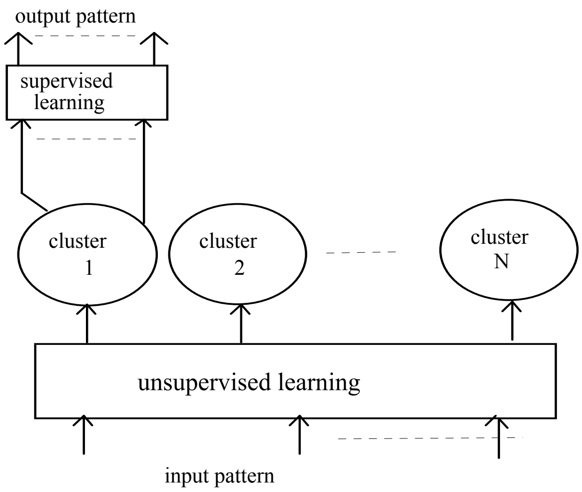
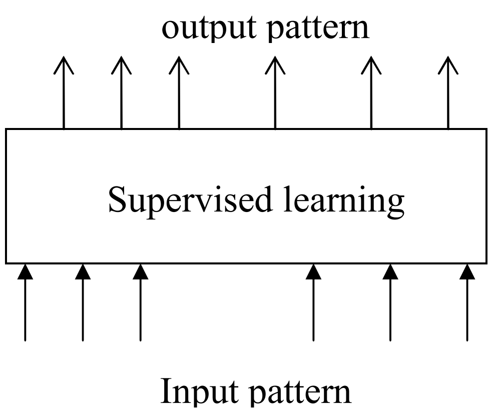



Fig. 1. Unsupervised/supervised concept-type1

Fig. 2. supervised concept-type




MKNNs consists of N input nodes and M output nodes arranged in a two dimensional grid as shown in Fig.3. The MKNNs has the following purposes [3,5]:
1) Clustering of the input space into a finite number of classes represented by the neural networks weight vectors
2) Topology preserving mapping of high dimensional input vectors into a lower dimensional surface represented by the location of the neuron on the grid.

MKNNs applies cross correlation in the frequency domain between input data and the weights of neural networks. In MKNNs, unsupervised learning is achieved in the feature map layer through competition. When an input pattern is presented to the neural network, it computes the activation value for each output node based on present connection weights. The input pattern is said to be mapped to the output node with maximum activation value. After enough input pattern vectors have been presented, input patterns with similar features will be mapped to the same output unit or to output units within a small neighborhood. This unit and its immediate neighbors on the grid are the only units permitted to learn in this pattern presentation by updating their weight vectors. During training, the weight adjustment is proportional to the difference between the input vector and weight vector.

In this paper, MKNNs maps the N inputs $x_1,x_2$ .... $x_N$ (N =24) on the M output node $y_1,y_2$ .... $y_M$. The output node are arranged on a two dimensional grid-like network. Because the clustering of input pattern vectors is self-organized in the learning process, and the ordering of the output node of an input pattern vector is based on that feature, this kind of neural network is called the self-organizing feature map by MKNNs.

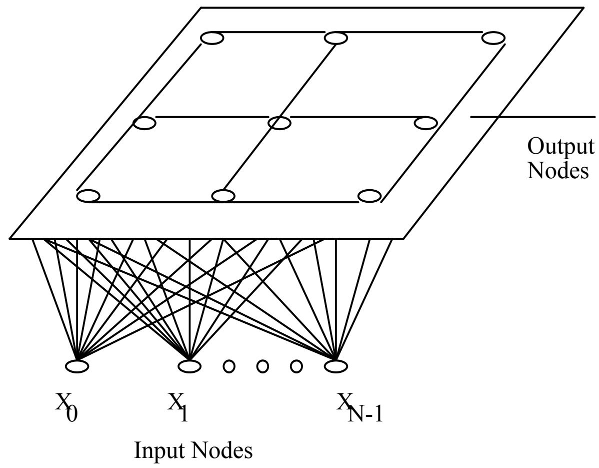



Fig.3 Modified Kohonen's Neural Networks.

Fig. 4 shows the flowchart of the algorithm to produce the self-organizing feature map that can be described as follows:-

1. Read the input patterns X(1), X(2), …X(P).
2. Select the initial values for the connection weights wij (i =1,….N, j= 1,…..M) and the radius of the neighborhood Nc.
3. The continuous-valued input patterns and the connection weights will give the output node j a continuous activation value which is given by
4. The node j* with maximum activation value is picked up. The connection weights for that node and all the nodes in the neighborhood defined by Nj are updated using the following equation

$$a_j = \sum_{i=1}^{N} w_{ij} x_i$$

5. Where, eta is the step size for the updating.
6. Repeat for each input pattern. When all P input patterns have been presented, then one iteration has been completed.

$$w_{i,j\,\text{updated}} = w_{ij} + \eta * (x_i - w_{ij})$$

7. Because there are ten iterations sharing one common radius of neighborhood and There are different values of Nc in the training process, then the iteration are repeated until Nc=0.

After training, any input vector stimulates only the neuron whose weight vector is the closest to the input vector in the input space. The weight vectors therefore represent certain averages of disjoint set or classes of input vectors. Further input vectors which are close in the input space stimulate neurons which are close to each other on the grid. Other neurons may not be stimulated by any input vector. Regrouping those neurons may not be stimulated by the same group of the input vectors leads to the concept of neuron clusters represented by output classes.
The characteristics of this unsupervised model [3,5] are:

- winner-take-all processing
- unsupervised learning
- lateral connected architecture used for topology preserving clustering and classification

Fig.5 shows the main components of the proposed prediction system.

This paper discussed the data classification using MKNNs for power system.
The elements of the new techniques are:-

1 –Historical DB that contains the recording hourly power consumptions, temperatures, wind speed, humidity, atmospheric pressure, from power system, through Data collection module.
2- Networks configuration which contains the parameters of MKNNs (number of input nodes, number of output nodes, iterations, number of patterns, and the Files for:- output results, Input Patterns, Input and Output weights, and the learning rate).
3- Kohonen's software that implements the self-organization Algorithms.
4- Self-organization user Interface As shown in Fig.6.
5- Output classes :- represent the day types/classes for the patterns of historical database.
6- Supervised software that implement the Linear Vector Quantization Learning Algorithm (LVQLA).
7- LVQ user interface that represent a graphic interface for supervised learning that calls the LVQLA.
8- FFNN(1) … FFNN(n):- represent feed-forward neural network for class 1, 2,… N respectively. These networks are used for prediction process for all day types.

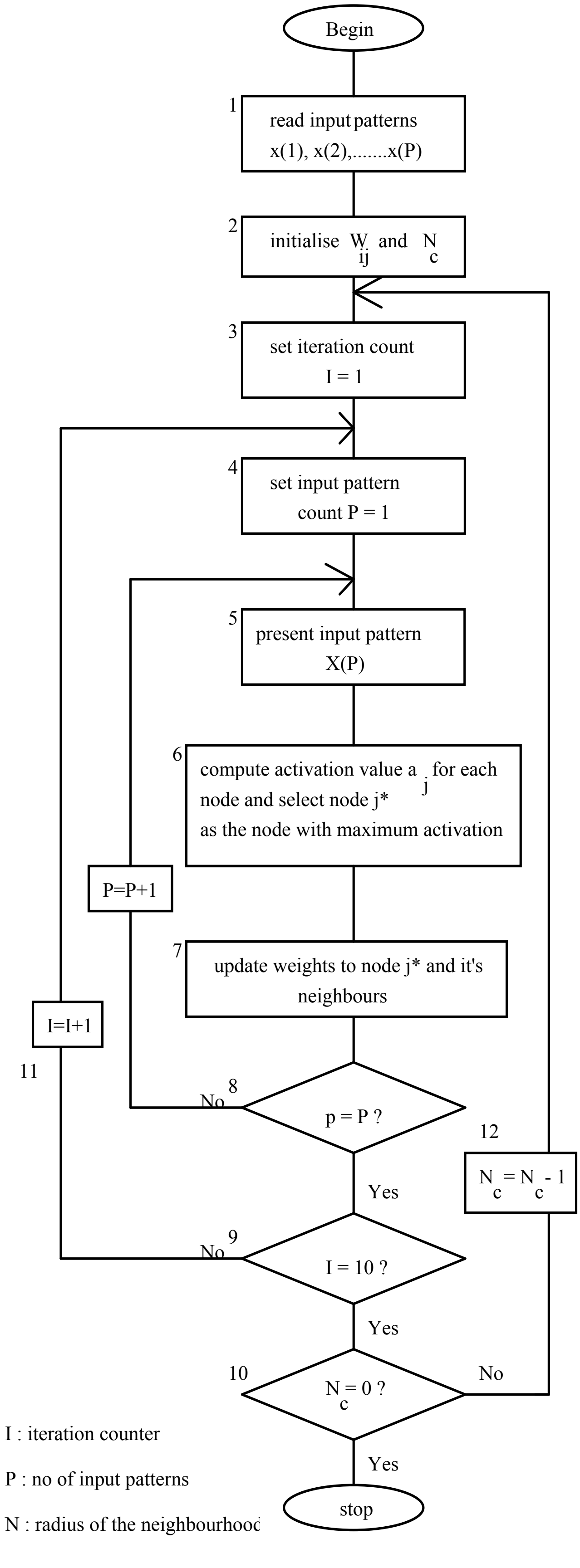



Fig.4.The algorithm of MKNNs.

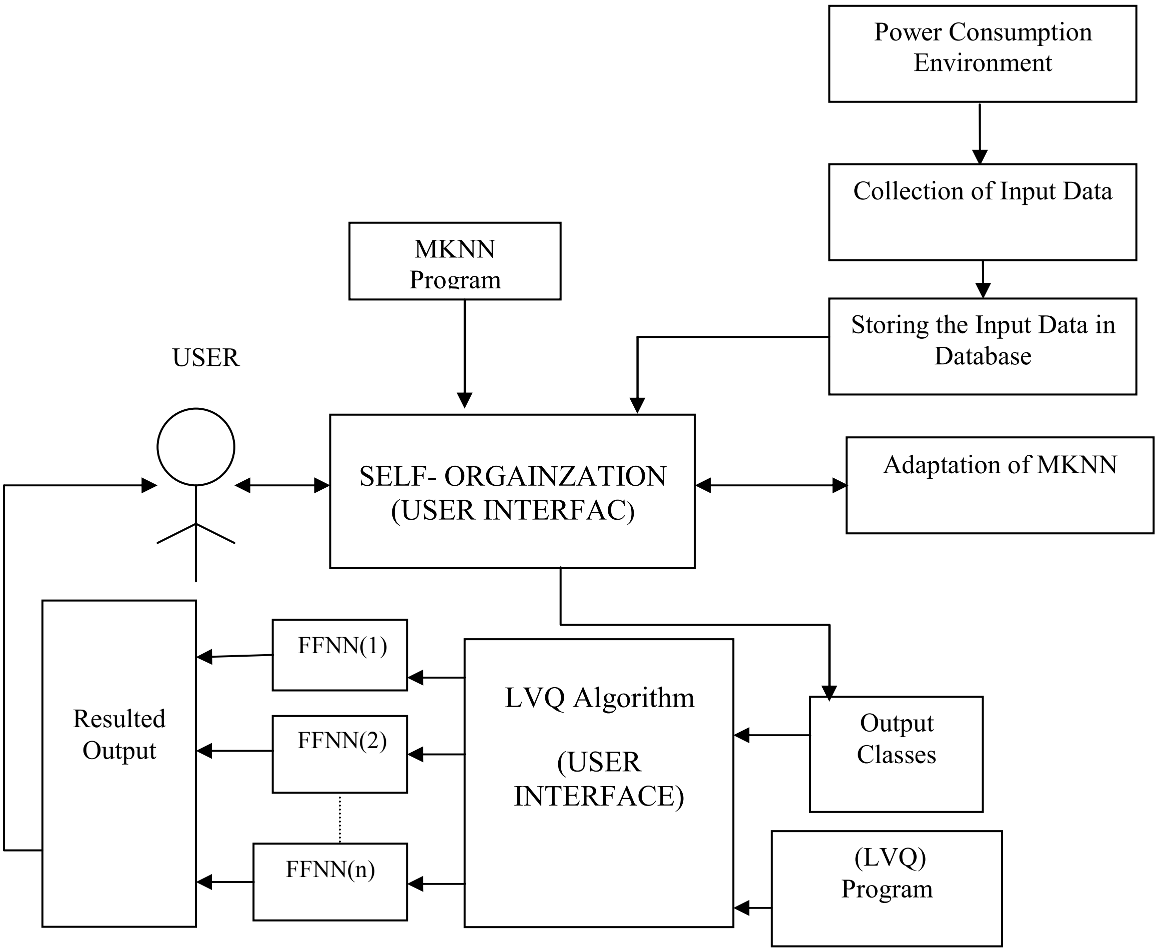



Fig.5 The structure of the prediction system

# VI. Conclusion

A New technique for prediction of power consumption has been presented. The proposed technique integrates the benefits of feedforward and feedback neural networks. It combines the unsupervised and supervised learning concept which classifies input patterns into classes and supervised learning and testing within each formed class. KNNs which uses self-organizing feature map has been modified to identify the day types which are essential to power consumption prediction processes. The inputs patterns of MKNNs are the 24 hourly power consumption patterns. Each input power consumption pattern is mapped to a node on the output plane according to its feature. Similar power consumption patterns are mapped to the same node or neighboring nodes. Examination of the power consumption patterns for each power system, has indicated that four week classes are working days, Thursday, Friday, and Saturday and special events. Each class should be separately used a neural network in the training process. Furthermore, it has been shown that FFNS are very effective in day type classification. The idea of these is to apply cross correlation in the frequency domain between the input data and the weights of neural networks. Moreover, It has been proven that the capability of MKNNs for identifying a new type of power consumption pattern (due to holiday, special event, bad data) before the expert operator is very high. Therefore, the proposed technique can be used as a valuable aid to the system operator in the prediction process.

# References

[1] Hazem M. El-Bakry, and Nikos Mastorakis, "A New Fast Forecasting Technique using High Speed Neural Networks," WSEAS Transactions on Signal Processing, Issue 10, vol. 4, October 2008, pp. 573-595.

[2] Hazem M. El-bakry, and Nikos Mastorakis, “Prediction of Market Price by using Fast Time Delay Neural Networks,” Proc. of 10th WSEAS Int. Conf. on Neural Networks (NN'10), Romania, June 13-15, 2010, pp. 230-237.

[3] Hazem M. El-bakry, and Wael A. Awad, “Fast Forecasting of Stock Market Prices by using New High Speed Time Delay Neural Networks,” Waset International Journal of Computer and Information Engineering, vol. 4, no.2., 2010, pp. 138-144.

[4] Hazem M. El-Bakry, and Nikos Mastorakis, " A New Fast Forecasting Technique using High Speed Neural Networks, " 8th WSEAS International Conference on SIGNAL, SPEECH AND IMAGE PROCESSING (SSIP '08), Santander, Cantabria, Spain, September 23-25, 2008, pp. 116-138.

[5] Hazem M. El-Bakry, "A Novel High Speed Neural Model for Fast Pattern Recognition," Soft Computing Journal, vol. 14, no. 6, 2010, pp. 647-666.

[6] Hazem M. El-Bakry, "New Faster Normalized Neural Networks for Sub-Matrix Detection using Cross Correlation in the Frequency Domain and Matrix Decomposition, " Applied Soft Computing journal, vol. 8, issue 2, March 2008, pp. 1131-1149.

[7] Hazem M. El-Bakry, "Face detection using fast neural networks and image decomposition," Neurocomputing Journal, vol. 48, 2002, pp. 1039-1046.

[8] Hazem M. El-Bakry, "Human Iris Detection Using Fast Cooperative Modular Neural Nets and Image Decomposition," Machine Graphics & Vision Journal (MG&V), vol. 11, no. 4, 2002, pp. 498-512.

[9] Hazem M. El-Bakry, "New Fast Principal Component Analysis For Real-Time Face Detection," Machine Graphics & Vision Journal (MG&V), vol. 18, no.4, 2009, pp. 405-426.

[10] Hazem M. El-Bakry, "A New Neural Design for Faster Pattern Detection Using Cross Correlation and Matrix Decomposition," Neural World journal, Neural World Journal, 2009, vol. 19, no. 2, pp. 131-164.

[11] Hazem M. El-Bakry, "An Efficient Algorithm for Pattern Detection using Combined Classifiers and Data Fusion," Information Fusion Journal, vol. 11, 2010, pp. 133-148.

[12] Hazem M. El-Bakry, "New Fast Time Delay Neural Networks Using Cross Correlation Performed in the Frequency Domain," Neurocomputing Journal, vol. 69, October 2006, pp. 2360-2363.

[13] Hazem M. El-Bakry, "A New High Speed Neural Model For Character Recognition Using Cross Correlation and Matrix Decomposition," International Journal of Signal Processing, vol.2, no.3, 2005, pp. 183-202.

[14] Hazem M. El-Bakry, "New High Speed Normalized Neural Networks for Fast Pattern Discovery on Web Pages," International Journal of Computer Science and Network Security, vol.6, No. 2A, February 2006, pp.142-152.

[15] Hazem M. El-Bakry "Fast Iris Detection for Personal Verification Using Modular Neural Networks," Lecture Notes in Computer Science, Springer, vol. 2206, October 2001, pp. 269-283.

[16] Hazem M. El-Bakry, and Qiangfu Zhao, "Fast Normalized Neural Processors For Pattern Detection Based on Cross Correlation Implemented in the Frequency Domain," Journal of Research and Practice in Information Technology, Vol. 38, No.2, May 2006, pp. 151-170.

[17] Hazem M. El-Bakry, and Qiangfu Zhao, "Fast Time Delay Neural Networks," International Journal of Neural Systems, vol. 15, no.6, December 2005, pp.445-455.

[18] Hazem M. El-Bakry, and Qiangfu Zhao, "Speeding-up Normalized Neural Networks For Face/Object Detection," Machine Graphics & Vision Journal (MG&V), vol. 14, No.1, 2005, pp. 29-59.

[19] Hazem M. El-Bakry, and Qiangfu Zhao, "A New Technique for Fast Pattern Recognition Using Normalized Neural Networks," WSEAS Transactions on Information Science and Applications, issue 11, vol. 2, November 2005, pp. 1816-1835.

[20] Hazem M. El-Bakry, and Qiangfu Zhao, "Fast Complex Valued Time Delay Neural Networks," International Journal of Computational Intelligence, vol.2, no.1, pp. 16-26, 2005.

[21] Hazem M. El-Bakry, and Qiangfu Zhao, "Fast Pattern Detection Using Neural Networks Realized in Frequency Domain," Enformatika Transactions on Engineering, Computing, and Technology, February 25-27, 2005, pp. 89-92.
[22] Hazem M. El-Bakry, and Qiangfu Zhao, "Face Detection Using Fast Neural Processors and Image Decomposition," International Journal of Computational Intelligence, vol.1, no.4, 2004, pp. 313-316.
[23] Hazem M. El-Bakry, and Ahmed Atwan, "Fast Business Networks for Real-Time Applications," International Journal of Universal Computer Sciences, issue 1, vol. 1, 2010, pp. 41-50.
[24] Hazem M. El-bakry, and Mohamed Hamada "Fast Time Delay Neural Networks for Detecting DNA Coding Regions," Springer, Lecture Notes on Artificial Intelligence (LNAI 5711), 2009, pp. 334-342.
[25] Hazem M. El-Bakry, and Nikos Mastorakis, "A New Fast Forecasting Technique using High Speed Neural Networks," WSEAS Transactions on Signal Processing, issue 10, vol. 4, October 2008, pp. 573-595.
[26] Hazem M. El-Bakry, "Automatic Human Face Recognition Using Modular Neural Networks," Machine Graphics & Vision Journal (MG&V), vol. 10, no. 1, 2001, pp. 47-73.
[27] Hazem M. El-Bakry, and H. Stoyan, "FNNs for Code Detection in Sequential Data Using Neural Networks for Communication Applications," Proc. of the First International Conference on Cybernetics and Information Technologies, Systems and Applications: CITSA 2004, 21-25.
[28] Hazem M. El-Bakry, and Nikos Mastorakis, "Fast Word Detection in a Speech Using New High Speed Time Delay Neural Networks," WSEAS Transactions on Information Science and Applications, issue 7, vol. 5, July 2009, pp. 261-270.
[29] Hazem M. El-Bakry, "Fast Virus Detection by using High Speed Time Delay Neural Networks," Journal of Computer Virology, vol.6, no.2, 2010, pp.115-122.
[30] Hazem M. El-Bakry, and Nikos Mastorakis, "Fast Image Matching on Web Pages," WSEAS Transactions on Signal Processing, issue 4, vol. 5, June 2009, pp. 157-166.
[31] Hazem M. El-Bakry, and Nikos Mastorakis, "Fast Information Retrieval from Web Pages," WSEAS Transactions on Information Science and Applications, issue 6, vol. 6, June 2009, pp. 1018-1036.
[32] Hazem M. El-Bakry, and Nikos Mastorakis, "Fast Detection of Specific Information in Voice Signal over Internet Protocol," WSEAS Transactions on Communications, issue 5, vol. 8, May 2009, pp. 483-494.
[33] Hazem M. El-Bakry, and Ahmed A. Mohammed, "Optimal Document Archiving and Fast Information Retrieval," The International Journal of Computer science, and Engineering, issue 2, vol. 3, July 2009, pp. 108-121.
[34] Hazem M. El-bakry, and Nikos Mastorakis "Fast Time Delay Neural Networks for Word Detection in a Video Conference," Proc. of European Computing and Computational Intelligence International Conference, Tbilisi, Georgia, June 26-28, 2009, pp. 120-129.
[35] Hazem M. El-bakry, and Nikos Mastorakis, "Fast Packet Detection by using High Speed Time Delay Neural Networks," Proc. of the 10th WSEAS Int. Conference on Multimedia Systems & Signal Processing, Hangzhou University, China, April 11-13, 2010, pp. 222-227.
[36] Hazem M. El-Bakry, and Ahmed Atwan, " Improving Quality of Business Networks for Information Systems," The International Journal of Computer, Information, and systems science, and Engineering, issue 3, vol. 3, July 2009, pp. 138-150
[37] Alaa M. Riad, Hazem M. El-bakry, and Nikos Mastorakis, "Fast Harmonic Current / Voltage Prediction by using High Speed Time Delay Neural Networks, " Proc. of WSEAS International Conference on Communication and Information, Athens, Greece, December 29-31, 2009, pp. 245-272.
[38] A. M. Riad, and A. K. Mahmoud, "Information System based-on Unsupervised/ Supervised Neural Networks for Daily Electric Load Forecasting process", Egyptian Computer Journal, ISSR, Cairo Univ., 2001.
[39] A. M. Riad, A. E. M. EL-Misiery, A. A. Soliman and A. K. Mahmoud, "A Hybrid Intelligent System for Forecasting Process", Egyptian Computer Journal, ISSR, Cairo Univ., 2002.
[40] "Neural Networks -An introduction", http://www.neusciences.com/nn_intro htm, 2000.
[41] "Neural Networks –Supervised learning ",http://www.neusciences. com/ nn_spvs2.htm, 2000.
[42] "Neural Networks –Unsupervised learning ",http://www.neusciences.com/nn_unsup.htm, 2000.
[43] " Artificial Neural Network Models",http://www-person.usyd.edu.au/~desm/afc-ann.html, 1999.
[44] D. W. Patterson, "Artificial neural networks: Theory and applications", Prentice Hall, 1996.
[45] K. Liu, S. Subbarayan, R. R. Shouults, M. T. Manry. C. Kwan, and F. L. Lewis, "Comparison of very short-term load forecasting techniques", IEEE Trans. PWS, Vol. 11, 1996, pp. 877-882.
[46] " Neurocomputing for energy forecasting", http://www.cordis.lu/esprit/scr /results /pages/ energy/energy2.htm, 1999.
[47] Y. Y.Hsu, and C. C. Yang, " Design of artificial neural networks for short-term load forecasting. Part1: Self-organization feature maps for day type identification", IEE Proc.-C, 1991, pp. 407-413.
[48] Y. Y. Hsu, and C. C. Yang, " Design of artificial neural networks for short-term load forecasting. Part2 : Multilayer feedforward networks for peak load and valley load forecasting", IEE Proc.-C, 1991, pp. 414-418.
[49] M. Djukanovic, B. Babic, D. J. Sobajic and Y. H. Pao " Unsupervised/ supervised learning concept for 24-hour load forecasting", IEE Proc.,Vol.140, 1993, pp. 311-318.
[50] A. G. Tejedor, M. J. Cosculluela, C. Bermejo and R. Montes " A neural system for short term load forecasting based on day-type classification" , ISAP-94 , France, 1994, Volume 1, pp. 35360.

[51] R. Satoh, E. Tanaka, and J. Hasegawa " Daily load forecasting using a neural network combined with regression analysis ", ISAP-94 , France, 1994, Volume 1, pp. 345- 352.
[52]"Forecast of Technology",http;//www,appliedfutures.com/reports/intelligent_agents,html 2000.
[53]Stevenson, Howard, ed. *Do lunch or be lunch*. Boston: Harvard Business School Press, 1998
[54] Flyvbjerg, B., 2008, "Curbing Optimism Bias and Strategic Misrepresentation in Planning: Reference Class Forecasting in Practice." European Planning Studies, vol. 16, no. 1, January, pp. 3-21.