2.1 INTRODUCTION

The data in medical domain are categorized into two types. The first type of a data which is collected by medical experts for a special research purpose is called prospective data. It has the following characteristics: the number of records is small; the number of attributes for each record is large, in comparison to the number of records; and the number of attributes with missing values is very few. The analysis of these data is called prospective analysis in epidemiology, because data collection is triggered by the generated hypothesis. Statistical analysis is usually applied to these datasets.

The second type of data which is retrieved from hospital information systems automatically. These data are collected without any specific research purpose. Usually, these data only include laboratory tests, although researchers in medical informatics are discussing how to store medical image, and physical examinations as electronic patient records. These data have the following characteristics; the number of records is huge; there are
the large numbers of attributes for each record (more than several hundred); many missing values can be observed; and many temporal sub-records can be stored for each record. This type of data is called retrospective data. The analysis of these data is called retrospective analysis in epidemiology, because data is analyzed after data collection. These data will lose any good features which prospective data holds and even statistical techniques do not perform well. This type of data is very similar to business databases. Concerning prospective data, data will be prepared with a hypothesis generated by medical experts very carefully. The quality of data is very high, and any data analysis technique will be applicable and useful. Only the problem with prospective data is that the number of measurements is very large, compared to the number of records. Data reduction or rule induction will be useful to detect the important attributes for analysis. On the other hand, as for retrospective data, there are many difficult issues for data analysis.

2.1.1 RENAL SYSTEM

The renal system consists of all the organs involved in formation and release of urine. It includes the kidneys, ureters, bladder and urethra. Initially, it is without specific symptoms and can only be detected as an increase in serum creatine. As the kidney function decreases, renal failure
becomes a serious medical condition affecting the kidneys. When persons suffer from renal failure, it means that their kidneys are not functioning properly or are no longer working. Renal failure can be a progressive disease or a temporary one depending on the cause and available treatment options\[62\].

The kidneys are glands that are located in the abdominal region just above the pelvis on either side of the spine. When functioning normally, the kidneys separate and filter excess water and waste from the blood stream. The kidneys are responsible for producing urine, which is used to flush away the toxins. The kidneys maintain a healthy balance of fluids and electrolytes, or salt compounds, in the body. In renal failure the kidneys undergo cellular death and are unable to filter wastes, or produce urine and maintain fluid balances. This dysfunction causes a build-up of toxins in the body which can affect the blood, brain and heart, as well as cause other complications. Renal failure is very serious and even fatal if left untreated \[62\].

The quantity and complexity of data acquired, time-stamped and stored in clinical databases by automated medical devices is rapidly and continuously increasing. As a result, it becomes more and more important to provide clinicians with easy-to-use interactive tools to analyze huge amounts of this data. These tools would serve different purposes, such as supporting
clinical decision making, evaluating the quality of the provided care, and carrying out medical research[48]. The specific clinical context is in the area of hemodialysis, where clinicians have to deal with huge amounts of data automatically acquired during the hemodialysis treatment of patients suffering from renal failure.

The implementation of the neural network in a training process is shown in the flow-chart given in Figure 2.1. The flow-chart shows training of ANN and implementation of ANN for renal data mining. During training of ANN, the network learns the training patterns by a weight updating algorithm. The training of ANN is stopped when a desired performance index of the network is reached. The weights obtained at this stage are considered as final weights. During implementation of ANN for renal data mining, the data collected from the patients are transformed with the final weights obtained during the training of ANN. Based on the outputs of the network, mining classification and diagnosis is done.

2.2 COLLECTION OF DATA

2.2.1 EXPERIMENTAL SET-UP

Renal data such as person's Age in terms of Years, Sex, Edema, Oliguria, Ultrafiltration Pressure, Polyuric, Nocturia, Urgency, Hydroxyethyl
Starch, Hypertension, Diabetics, Family History, Amino glycoside Usage, Obesity, Hemoglobin, Cholesterol and Creatinine have been collected from renal patients. The collected data are given in Table 2.1 at Appendix-C. A total of 17 parameters about renal organ have been collected.

2.2.2 DATA PREPARATION

The data include information on the dialysis prescription, data electronically collected during each dialysis treatment, laboratory tests, pharmacy records, patient diagnosis and demographic data. Before each session, the patient was weighed and her/his blood pressure (systolic and diastolic) registered while sitting (supine pressure), and when possible, while standing. The weight and blood pressure measurements are repeated at the end of the session. The levels of sodium, bicarbonate, potassium, calcium, and glucose in the dialysis solution are recorded [49].

Total time for the dialysis session, blood flow rate, total volume of blood processed, dialysis flow rate, and the overall average pressures at the arterial and venial side of the blood pump were another set of collected values. A set of measurements was collected by the dialysis machine every twenty minutes or on request. This set includes systemic blood pressure, pulse, blood flow rate, arterial and venial blood pump pressures, trans-
membrane pressure, and the rate of ultra filtration. To reduce data noise, averages were computed over the fifteen readings taken by the machine during the dialysis session[49].

The demographic and outcomes data set contains the patient’s date of birth, gender, and race; the date(s) of death, kidney transplant, and transfer into or out of the dialysis center. The final portion contains the diagnosis codes for the primary and secondary diagnoses. Differences between each patient’s average post and Pre-systemic blood pressures were calculated for all four combinations of systolic and diastolic pressures and supine and standing positions. The pulse pressures (determined by the difference between the systolic and diastolic blood pressures) were calculated for both pre and post conditions for both supine and standing positions. Differences between the supine and standing pressures were also calculated for both systolic and diastolic blood pressure and for the pre and post dialysis conditions. Some new features were added to the data set by using the concept of data transformation. Averages were computed for each patient for all variables to form a single representative record (aggregate data set). Initial data mining focused on a selected group of long-term dialysis patients with at least fifteen or more views[49].
Fig. 2.1 (A) Flow-chart for Training the ANN using Renal Data

Fig. 2.1 (B) Flow-chart for Implementation of Renal Data Mining
2.3 NORMALIZATION OF THE PATTERNS

The patterns are normalized so that the values of the features are in the range of 0 to 1, and the computational complexity is reduced. The normalization of the patterns is done by:

\[ x_i = \frac{x_i}{x_{\max}} \] ....(2.3.1)

where

- \( x_i \) is the value of a feature, and
- \( x_{\max} \) is the maximum value of the feature [55].

2.3.1 SELECTION OF PATTERNS FOR TRAINING

The number of classes, which are based on the classification range of the outputs, are decided. If only one output is considered, the range of classification is simple.

If more than one output is considered, a combination criterion has to be considered. The total number of patterns are decided for each class. Out of these patterns, the number of patterns to be used for training the network is decided. The remaining patterns are used for testing the classification performance of the network. The patterns selected for training the network should be such that they represent the entire population of the data. The selection of patterns is done by:
\[ E_i^2 = \frac{\sum_{j=1}^{nf} (x_{ij} - \bar{x}_j)^2}{\sigma_i^2} \]  

...(2.3.2)

where

\( E_i^2 \) is the maximum variance of a pattern,

\( nf \) is the number of features, and

\[ \sigma_i^2 = \frac{\sum_{j=1}^{nf} (x_{ij} - \bar{x}_j)^2}{L} \]  

...(2.3.3)

\( \bar{x}_j \) is the mean for each feature, and

\( L \) is the number of patterns

where the value of \( E_i^2 \) is found for each pattern. Patterns with maximum \( E_i^2 \) are chosen from each class for training the network[55].

2.4 TRAINING STRATEGIES FOR THE NETWORK

For the network to learn the patterns, different weight updating algorithms have been developed. They are called supervised methods and unsupervised methods. Since both the inputs and outputs are considered for renal data, supervised learning technique has been used. The present work involves modification of existing weight updation algorithm, combination of
classical method with neural network, method of training the network for more number of patterns, and training the network properly for more than two classifications. The performance of the different methods developed and trained has been compared with the performance of BPA, since BPA is a well known algorithm. Besides, exclusive-or (XOR) problem has been considered as the standard problem, to know the influence of various learning parameters on the training methods developed.

The number of patterns in each class, which are based upon the desired range of outputs in each class, has been taken into consideration. To start with, only two classes are considered for the purpose of renal data classification. The classification performance of the network for the test patterns with two classifications and four classifications will be analyzed and presented. When more than two classifications are to be considered, the training procedure of the network has to be improved. The number of classes, the number of patterns used for training and testing in each class, and the range of classification output are shown in Table 2.2.
Table 2.2 Number of patterns and classification range in each class

<table>
<thead>
<tr>
<th>Class</th>
<th>No. of patterns in each class</th>
<th>No. of patterns used for training ANN</th>
<th>No. of patterns used for testing ANN</th>
<th>Class</th>
</tr>
</thead>
<tbody>
<tr>
<td>C 1</td>
<td>54</td>
<td>25</td>
<td>28</td>
<td>1</td>
</tr>
<tr>
<td>C 2</td>
<td>116</td>
<td>50</td>
<td>66</td>
<td>2</td>
</tr>
<tr>
<td>C 3</td>
<td>62</td>
<td>30</td>
<td>32</td>
<td>3</td>
</tr>
<tr>
<td>C 4</td>
<td>769</td>
<td>500</td>
<td>269</td>
<td>4</td>
</tr>
<tr>
<td>Total</td>
<td>1000</td>
<td>605</td>
<td>395</td>
<td></td>
</tr>
</tbody>
</table>

The network functions on a supervised learning strategy. The inputs of a pattern are presented. The output of the network obtained in the output layer is compared with the desired output of the pattern. The difference between the calculated output of the network and the desired output is called the Mean Squared Error (MSE). The MSE of the network for the pattern presented is minimized. This error is propagated backwards, such that the weights connecting the different layers are updated. By this process, the MSE of the network for the pattern presented is minimized. This procedure has to be adopted for all the training patterns and the MSE of each pattern is summed up. After presenting the last training pattern, the network is considered to have learnt all the training patterns through iterations, but the
MSE is large. To minimize MSE, the network has to be presented with all the training patterns many times. There is no guarantee that the network will reach the global minimum; instead, it will reach one of the local minima. The MSE may increase, which means divergence rather than convergence. Sometimes, there may be oscillation between convergence and divergence.

The training of the network can be stopped either by considering MSE or by considering classification performance as the criterion. When classification performance is considered as the criterion, test patterns are presented at the end of each iteration. Once the desired performance is obtained, training of the network is stopped. When MSE is considered as the criterion, one may not know the exact MSE, to which the network has to be trained. If the network is trained till it reaches a very low MSE, over-fitting of the network occurs. Over-fitting represents the loss of generality of the network. That is, the network classifies only the patterns, which are used during training, and not the test patterns.

2.5 BACK-PROPAGATION ALGORITHM (BPA)

The BPA uses the steepest-descent method to reach a global minimum. The flow-chart for the BPA is given in Figure 2.2. The number of layers and number of nodes in the hidden layers are decided. The connections between
nodes are initialized with random weights. A pattern from the training set is presented in the input layer of the network and the error at the output layer is calculated. The error is propagated backwards towards the input layer and the weights are updated. This procedure is repeated for all the training patterns. This forms one iteration. At the end of each iteration, test patterns are presented to ANN, and the classification performance of ANN is evaluated. Further training of ANN is continued till the desired classification performance is reached[65]. The algorithm for BPA is given in Appendix A.
Fig. 2.2 Flow-chart for the Back Propagation Algorithm (BPA)
2.5.1 RESULTS OF THE NETWORK TRAINED BY USING BPA FOR THE EXCLUSIVE-OR (XOR) PROBLEM

The XOR problem is given in Table 2.3. In this table, the output has been modified from 0 to 0.1, and from 1 to 0.9. Because, the sigmoid function used to train the network will never reach either 0 or 1, due to the presence of the exponential function. In this chapter and in the subsequent chapters, MSE is fixed as 0.01 when ever XOR problem is used to train the network. There is no sanctity in considering MSE as 0.01; instead, one can consider some other value. The initialization of the weights and the thresholds are in the range of 0.25 to 0.47. The iterations required by the network, which are trained by using BPA for different number of nodes in the hidden layer to reach MSE of 0.01, are shown in Figure 2.3.

<table>
<thead>
<tr>
<th>Pattern no.</th>
<th>Inputs</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Original</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Fig. 2.3 Effect of number of nodes in the hidden layer for the network trained by using BPA

The network is trained with two hidden layers. The total number of nodes used in both the hidden layers is 14. The different combinations of number of nodes in the first hidden layer and in the second hidden layer are given in Table 2.4. The convergence rates of the network with two hidden layers and the convergence rates of the network with one hidden layer are shown in Figure 2.4. When there is only one hidden layer and the number of hidden nodes is 6, it requires 1015 iterations for the network to reach MSE of 0.01. When there are two hidden layers with 5 nodes in the first hidden layer and 8 nodes in the second hidden layer, it requires 6880 iterations for the
network to reach MSE of 0.01. Since it requires more number of iterations for the network with more than one hidden layer, it is sufficient to have only one hidden layer.

Table 2.4 Number of nodes in each hidden layer

<table>
<thead>
<tr>
<th>No. of nodes in layer 1</th>
<th>No. of nodes in layer 2</th>
<th>No. of iterations to reach MSE of 0.01</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>10</td>
<td>7222</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>8663</td>
</tr>
<tr>
<td>6</td>
<td>8</td>
<td>6880</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>10010</td>
</tr>
<tr>
<td>8</td>
<td>6</td>
<td>10322</td>
</tr>
<tr>
<td>9</td>
<td>5</td>
<td>11277</td>
</tr>
<tr>
<td>10</td>
<td>4</td>
<td>11695</td>
</tr>
</tbody>
</table>
The learning factor ($\eta$) is supposed to guide the convergence rates of the network to the desired MSE with less number of iterations. It so happens that sometimes $\eta$ will make the network to converge to the desired MSE after an increased number of iterations. For 6 nodes in the hidden layer, it requires 985 iterations for the network to reach MSE of 0.01 when $\eta$ is 1.0 and 2404 iterations for the network to reach MSE of 0.01 when $\eta$ is 0.05. The convergence rates of the network for various number of nodes in the hidden layer for different values of $\eta$ is shown in Figure 2.5.
The network is trained with threshold ($\theta$) and without $\theta$. When $\theta$ is used, updation of $\theta$ is done similar to weight updation. The parameter $\theta$ is used in all the layers except in the input layer. For 6 nodes in the hidden layer, it requires 985 iterations for the network to reach MSE of 0.01 without $\theta$, and 1722 iterations for the network to reach MSE of 0.01 with $\theta$. The convergence rates of the network with $\theta$ and without $\theta$ are shown in Figure 2.6. From the graph, it can be seen, that network trained without $\theta$ converges faster than the network trained with $\theta$. 

Fig. 2.5 Effect of $\eta$ in the network trained by using BPA
To achieve faster convergence of the network an accelerating factor is used which is a parameter called momentum factor ($\alpha$). The network is trained with $\alpha$ and without $\alpha$. The value of $\alpha$ is from 0 to 1. For 6 nodes in the hidden layer, it requires 985 iterations for the network to reach MSE of 0.01 without $\alpha$, and 380 iterations for the network to reach MSE of 0.01 with $\alpha$. The value of $\alpha$ used is 0.8. For other values of $\alpha$, the network requires very large number of iterations to reach MSE of 0.01. The convergence rates of the network trained with $\alpha$ and without $\alpha$ are shown on Figure 2.7. The network trained with $\alpha$ requires less number of iterations to reach the desired MSE.
2.5.2 RESULTS OF THE NETWORK TRAINED BY USING BPA FOR THE RENAL DATA

The patterns given in Table 2.1 At APPENDIX-C are normalized using equation (2.3.1). The number of patterns used for training the network are 605. The training patterns are selected out of 1000 patterns by using the equation 2.3.2. The number of test patterns used are 395. The network is trained by using BPA. Five nodes are used in the hidden layer. The training conditions used are $\eta$ as 1 and $\alpha$ as 0.8. A minimum classification
performance of 90% is obtained in 1341 iterations, when the MSE is 0.0043. The classification performance and the MSE curves for the network is shown in Figure 2.8. For some iterations, the classification performance curves oscillate. After 1340\textsuperscript{th} iteration, the classification performance curve is steady without any further changes.

Fig. 2.8 MSE and classification performance of the network trained by using BPA for the Renal data

2.6 EXTENDED KALMAN FILTER ALGORITHM (EKF)

The algorithm uses a modified form of the BPA to minimize the MSE between the desired outputs and the outputs of the network with respect to
the inner products to the non-linear function. But, in the conventional BPA, the MSE is minimized with respect to the weights. The EKF algorithm is a state estimation method for a nonlinear system, and it can be used as a parameter estimation method, by augmenting the state with an unknown parameter. A multi-layered network is a non-linear system with layered structure, and its learning algorithm is regarded as parameter estimation for such a system. The EKF based learning algorithm gives approximately the minimum variance estimates of the weights. The convergence of the EKF is faster than that of BPA. Error signals, which are generated by EKF, are used to estimate the input of the non-linearity. These estimated inputs, along with the input vectors to the respective nodes, are used to produce an updated set of weights, through a system of linear equations at each node[55]. These systems of linear equations are solved by using a Kalman filter at each layer.

The flow-chart for the EKF weight update algorithm is given in Figure 2.9. The number of layers and the number of nodes in the hidden layers are decided. The weights of the network are initialized. The parameter $\lambda$ is initialized to a very small value. The initial trace of the inverse matrix $T$ and the accelerating parameter $T_{\text{max}}$ are initialized to 20. The matrix $Q$ represents the error co-variance. The parameter $\lambda$ is calculated by using $T_{\text{max}}$. A training pattern is presented to the input layer of the network, and the error of the network is calculated in the output layer. The error $\delta$ for each
node in the output layer is calculated. The inverse matrices Q for the nodes in the output layer are updated. The $\sigma$ for the nodes in the immediate hidden layer is calculated. The Q and $\lambda$ for the nodes in the hidden layer are updated. The above procedure is repeated for all the training patterns and for all the iterations, until the classification performance of the network for the test patterns reaches the desired value. The algorithm for the weight updation of the network, by using EKF, is given in Appendix-B.
Fig. 2.9 Flow-chart for the Extended Kalman Filter algorithm (EKF)
2.6.1 RESULTS OF THE NETWORK TRAINED BY USING EKF FOR THE XOR PROBLEM

The network is trained for different combinations of values for $T_{\text{max}}$ and initial trace of $Q$. For a given $T_{\text{max}}$ of 2, the value of the initial trace of $Q$ is increased from 1 to 7. When the initial trace of $Q$ is high, convergence of the network is increased from 2 onwards, the convergence of the networks is fast. Similarly when the value for $T_{\text{max}}$ is still faster. For 8 nodes in the hidden layer, it requires 564 iterations for the network when the initial value for the trace of $Q$ is 1, and 158 iterations when the initial value for the trace of $Q$ is 7, to reach MSE of 0.01. The convergence rates of the network for different combinations of values for $T_{\text{max}}$ and initial trace of $Q$ are shown in Figure 2.10.

The network is trained with $\alpha$ and without $\alpha$. The training conditions used are: $\alpha$ as 0.9, $T_{\text{max}}$ as 2 and $Q$ as 7. The nodes used in the hidden layer are from 8 onwards. The network trained with nodes less than 8 did not converge to MSE of 0.01. For 8 nodes in the hidden layer, it requires 43 iterations for the network to reach MSE of 0.01 with $\alpha$ as 0.8 and 119 iterations without $\alpha$, to reach MSE as 0.01. The convergence rates of the network with $\alpha$ and without $\alpha$ are shown in Figure 2.11.
Fig. 2.10 Effect of $T_{\text{max}}$ and initial value for the trace of $Q$ in the network trained by using EKF for the XOR problem

Fig. 2.11 Effect of $\alpha$ in the network trained by using EKF
2.6.2 RESULTS OF THE NETWORK TRAINED BY USING EKF FOR THE RENAL DATA

The network is trained by using EKF weight update algorithm to learn the renal data. The training conditions used are: $\alpha$ as 0.5, initial value for the trace of $Q$ as 20 and initial value of $T_{\text{max}}$ as 20. The number of nodes in the hidden layer is 5. A maximum classification performance of 87% is obtained in 6 iterations. The MSE is 0.0394. The classification performance and the MSE curves are shown in Figure 2.12. The classification performance increases up to the 3rd iteration, and then decreases followed by an increase.

Fig. 2.12 Classification performance of the network trained by using EKF for the Renal patterns
The analysis of the network's performance was carried out, by presenting all the test patterns at the end of each iteration during training. Further training was stopped, when the performance of the network reached maximum classification of the test patterns. The convergence rates of the network trained by using EKF is faster than that in BPA. To compare EKF with BPA, it requires 9 iterations (Figure 2.12) for EKF and 18 iterations for BPA (Figure 2.8), in order to achieve a classification performance of 86.74% of the test patterns. From the above conclusions, it can be seen that the EKF weight updating algorithm can be preferred instead of BPA, to train the network, when the number of iteration are taken as the criteria to reach the desired MSE.