CHAPTER 8

AN IMPROVED BI-MODEL NEURAL NETWORK CLASSIFICATION SYSTEM FOR IMPROVED CLASSIFICATION OF CARDIOTOCOGRAM DATA

8.1 INTRODUCTION

In the previous chapter, showed that an outlier based Bi-Model neural network classification system for improved classification of Cardiotocogram data can classify the CTG data better than most of the other methods. In this work, present the improved classification models which will consider outliers in the data and eliminate them from training phase of the classification process. This model is almost similar to the previous BM-NN Model. But here use the Eigen Vectors of the training data to reduce the dimension of the training data as well as testing data. The proposed idea also considerably improved the performance in classifying Normal, Suspicious and Pathologic CTG patterns.

8.2 PRINCIPAL COMPONENT ANALYSIS (PCA)

Principal component analysis (PCA) is a mathematical procedure that uses an orthogonal transformation to convert a set of observations of possibly correlated variables into a set of values of linearly uncorrelated variables called principal components. The number of principal components is
less than or equal to the number of original variables. This transformation is defined in such a way that the first principal component has the largest possible variance (that is, accounts for as much of the variability in the data as possible), and each succeeding component in turn has the highest variance possible under the constraint that it be orthogonal to (i.e., uncorrelated with) the preceding components. Principal components are guaranteed to be independent only if the data set is jointly normally distributed. PCA is sensitive to the relative scaling of the original variables.

PCA is the simplest of the true eigenvector-based multivariate analyses. Often, its operation can be thought of as revealing the internal structure of the data in a way that best explains the variance in the data. If a multivariate dataset is visualised as a set of coordinates in a high-dimensional data space (1 axis per variable), PCA can supply the user with a lower-dimensional picture, a "shadow" of this object when viewed from its (in some sense; see below) most informative viewpoint. This is done by using only the first few principal components so that the dimensionality of the transformed data is reduced.

PCA is closely related to factor analysis. Factor analysis typically incorporates more domain specific assumptions about the underlying structure and solves eigenvectors of a slightly different matrix.

PCA is also related to canonical correlation analysis (CCA). CCA defines coordinate systems that optimally describe the cross-covariance between two datasets while PCA defines a new orthogonal coordinate system that optimally describes variance in a single dataset.
8.3 THE IMPROVED BI-MODEL (IBM-NN) CLASSIFICATION SYSTEM

In this work, the records in the training data classified with class labels. The Eigen vectors of the training data is used to reduce the dimension of the training data as well as testing data. Used the reduced dimensionality of training data as well as testing data were trained and classified. Outliers or abnormal records in the training data are detecting during the first stage of training and testing of the Back Propagation Neural network (BPN). After detecting outliers, those outliers will be removed from the training data, and again the same network will be trained with the outlier removed data to improve the training performance of the neural network and all the outliers will be included in the classification process. So, in this work, are going to address some of the machine learning based hybrid datamining techniques for the better classification of CTG data.

Advantages of IBM-NN Classification Model

1. Since the reduced dimension data is only used for training and testing, there will be considerable improvement in performance in terms of accuracy of classification.

2. Further the low dimensional data will also reduce the overhead in training. So that, will get little improvement in performance than BM-NN model during practical implementations.
Figure 8.1 Proposed Improved Bi-Model NN Classification System

The Above block diagram shows the proposed IBM-NN system. This model is almost similar to the previous BM-NN Model. But here use the
Eigen Vectors of the training data to reduce the dimension of the training data as well as testing data.

**The Steps in Improved Bi-Model classification System**

**Procedure IBM-NN** {

1. Read training data \(D_L\) and targets \(C_L\) and test data \(D_T\)

2. \((\lambda_i , u_i) \leftarrow \text{PCA}(D_L)\)

   Where \(\lambda_i\) – Eigen Values and \(u_i\) - Eigen Vectors and \(i : 1 \ldots n\)

3. \(u_iD_L \rightarrow D_{LR}\) - the dimensionality reduced representation of \(D_L\)

4. Create Network N1 to learn \(D_{LR}\) and map it to the original output class \(C_L\)

5. Classify \(D_{LR}\) using the trained network N1.

6. Separate the Outliers \(O_L\) from \(D_L\) Based on the Log-Sigmoid output of the output layer of N1

7. Train Another Network N2 only using \(O_L\)

8. \(U_iD_T \rightarrow D_{TR}\) - the dimensionality reduced representation of \(D_T\)

9. Classify the \(D_{TR}\) using the trained network N1.

10. Separate the Outliers \(O_T\) from \(D_T\) Based on the Log-Sigmoid output of N1

11. Find Class labels \(C_O\) of the outliers \(O_T\) using the Trained Network N2

12. Separate the Predicted Class Labels \(C_N\) of non outliers of \(D_T\) from step-9 output

13. Combining \(C_O\) from step-11 and \(C_N\) from step-12 Gives the Predicted Class Labels \(C_T\) of \(D_T\)

}
8.4 IMPLEMENTATION AND EVALUATION

For implementing and evaluating the proposed improved Bi-Model neural network classification system, used Matlab 7. For evaluating the algorithms under consideration, used Cardiotocogram data from UCI Machine Learning Repository.

8.5 RESULTS AND DISCUSSION

The following table shows the Classification performance of the proposed Improved Bi-Model neural network based classification algorithm.

Table 8.1 Classification Performance of IBM-NN algorithm

<table>
<thead>
<tr>
<th>Class</th>
<th>Precision</th>
<th>Recall</th>
<th>F-Measure</th>
</tr>
</thead>
<tbody>
<tr>
<td>Normal</td>
<td>0.9499</td>
<td>0.9735</td>
<td>0.9615</td>
</tr>
<tr>
<td>Suspicious</td>
<td>0.7603</td>
<td>0.7242</td>
<td>0.7411</td>
</tr>
<tr>
<td>Pathological</td>
<td>0.9551</td>
<td>0.8030</td>
<td>0.8718</td>
</tr>
</tbody>
</table>

The following table show the performance of proposed and compared algorithms in terms of Rand index.

Table 8.2 Performance in terms of Rand Index

<table>
<thead>
<tr>
<th>Method</th>
<th>Rand Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPN</td>
<td>0.8433</td>
</tr>
<tr>
<td>BL-NN</td>
<td>0.8566</td>
</tr>
<tr>
<td>BM-NN</td>
<td>0.8915</td>
</tr>
<tr>
<td>IBM-NN</td>
<td>0.8992</td>
</tr>
</tbody>
</table>
The following table show the performance of proposed and compared algorithms in terms of time.

**Table 8.3 Performance in teams of Time**

<table>
<thead>
<tr>
<th>Method</th>
<th>Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>BPN</td>
<td>2.1671</td>
</tr>
<tr>
<td>BL-NN</td>
<td>4.2741</td>
</tr>
<tr>
<td>BM-NN</td>
<td>3.6513</td>
</tr>
<tr>
<td>IBM-NN</td>
<td>3.5050</td>
</tr>
</tbody>
</table>

The Analysis of Results

The following chart (Figure 8.2) shows the comparison of Rand index under four different methods. While comparison of performance in terms of Rand index the IBM-NN algorithm provided good performance than other methods.

**Figure 8.2 Comparison of performance in teams Rand Index**
The following chart (Figure 8.3) shows the performance of the algorithms in terms of CPU time. In terms of time, the performance of IBM-NN was good than the other two our proposed algorithms.

![Comparison of CPU Time](chart)

**Figure 8.3 Comparison of performance in terms CPU time**

The following chart (Figure 8.4) shows the Comparison of Precision under four different methods. The proposed IBM-NN based classifier provided good Precision in all the cases (Normal, Suspicious and pathological). Even though the performance of BPN in terms of Precision is good while classifying the Normal and Pathologic records, it was not good in identifying the suspicious cases. Particularly, the proposed method significantly improved the performance in the case of suspicious class.
Figure 8.4 Comparison of Performance in terms of Precision

The following chart (Figure 8.5) shows the Comparison of Recall under four different methods. The Proposed IBM-NN based classifier provided good Recall in all the cases. In terms of recall, BPN was not good in identifying the suspicious and pathological cases.

Figure 8.5 Comparison of Performance in terms of Recall
The following chart (Figure 8.6) shows the Comparison of F-Measure under four different methods. The proposed IBM-NN based classifier provided good recall in all the cases (Normal, Suspicious and pathological). Even though the performance of BPN in terms of recall is good while classifying the Normal and Pathologic records, it was not good in identifying the suspicious cases.

**Figure 8.6 Comparison of Performance in terms of F-Measure**

The following charts (Figure 8.7) show the performance of IBM-NN algorithm. In general, the algorithm gives good precision for normal records and poor performance in all other cases.
8.6 SUMMARY

The derived results obviously shows that the proposed two level training improved the classification performance of system. The IBM-NN approach provided good performance in all cases than compared other methods. The performance of standard neural network based classification model and our previous algorithm BL-NN and BM-NN were compared with proposed IBM-NN Method. According to the arrived results, the performance of the proposed supervised machine learning based classification approach provided significant performance than other compared methods.