CHAPTER 5
IMPLEMENTATION OF RADIAL BASIS FUNCTION AND
COMBINATIONS OF ANN FOR INTRUSION DETECTION

5.1 INTRODUCTION
This chapter presents the implementation of radial basis function for Intrusion Detection.

5.2 RADIAL BASIS FUNCTION
Radial Basis Function (RBF) uses the concept finding distance between the input pattern and the center. The summed distance is passed through an exponential activation function and a bias is used. This RBF is very much suitable for non-linear data, especially the non-linear distribution of the temperature in the root.

The concept of distance measure is used to associate the input and output pattern values. Radial Basis Functions are capable of producing approximations. The approximation is produced bye-passing an input point through a set of basic functions each of which contains one of the RBF centers. multiplying the result of each function by a coefficient and then summing them linearly. The data have been trained using radial basis function. The RBF learns 1000 patterns during training. The RBF learns the dataset quickly. It has an important advantage that it does not involve in repeated training.

The approximation is essentially stored in the coefficients and center of the RBF. These parameters are in no way unique since for each function is approximated many combinations of parameter values exist. RBF has the following mathematical representation

\[ F(x) = c_o + \sum_{i=0}^{N-1} c_i \Phi(|| x - R_i ||) \] (5.1)

where,

\[ c_o \] bias
\( c_i \)  vector containing the coefficients of the RBF  
\( R \)  vector containing the center of the RBF  
\( \Phi \)  basis function or activation function of the network  
\( F(x) \)  approximation produced as the output of the network  
\( N \)  total number of nodes in the input layer  
\( x \)  input vector  

The equation (5.2) shows the Euclidean distance for a vector ‘\( x \)’ containing \( n \) elements  
\[
\| x \| = \sqrt{\sum_{i=1}^{n} x_i^2} \quad (5.2)
\]

Each center \( R_i \) has the same dimension as the input vector ‘\( x \)’, which contains ‘\( n \)’ input values.

The centers are points within the input data space and are chosen so that they are representative of the input data. When a RBF calculates its approximation to some input data point, the distance between the input point and each center is calculated, in terms of the Euclidean distance. The distances are then passed through the basis function \( \Phi \).

The results of the basic functions are weighted with the coefficients \( c_i \) and these weighted results are then linearly summed to produce the overall RBF output. One of the most common choices for the basic function is that of the Gaussian  
\[
\Phi(x) = \exp\left(\frac{-x^2}{2\sigma}\right) \quad (5.3)
\]

where,
\( x \)  distance between the input vector and the center, and  
\( \sigma \)  scaling parameter.
Fig. 5.1 Flow chart of Radial Basis Function implementation

1. Read input pattern
2. Create centres
3. Calculate rb = exp(-x)
4. G = rbTrb
5. Determinant D = det(G)
6. Is D = 0?
7. If yes, find svd (d)
   G = U * W * VT
8. If no, B = inv(G)
9. E = B * GT
10. Final weight F = E * TARGET
### Table 5.1 The algorithm of Training Radial Basis Function

<table>
<thead>
<tr>
<th>Step 1: Radial Basis Function are applied</th>
</tr>
</thead>
<tbody>
<tr>
<td>No. of Inputs = 41</td>
</tr>
<tr>
<td>No. of Patterns = 1000</td>
</tr>
<tr>
<td>No. of Centres = 100</td>
</tr>
<tr>
<td>Calculated RBF as</td>
</tr>
<tr>
<td>RBF = exp (-X)</td>
</tr>
<tr>
<td>Matrix G = RBF is calculated.</td>
</tr>
<tr>
<td>Matrix A = G^T * G is calculated.</td>
</tr>
<tr>
<td>Matrix B = A^-1 is calculated.</td>
</tr>
<tr>
<td>Matrix E = B * G^T is calculated.</td>
</tr>
</tbody>
</table>

**Step 2:** Final Weight vector F = E * D is calculated.

**Step 3:** The Final Weight is stored in a file.

### Table 5.2 The algorithm of Testing Radial Basis Function

<table>
<thead>
<tr>
<th>Step 1: Features of a packet is read.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2: Final weights are read.</td>
</tr>
<tr>
<td>Calculated Numerals = F * E</td>
</tr>
</tbody>
</table>

**Step 3:** Presence / absence of intrusion is detected.

The data that have been given in Table 3.4 has been used for generation of the results. Figure 5.2 shows the arrangement of layers and nodes in the RBF. Inputs are presented in the input layer. The center patterns represent the input layer and hidden layer matrix. The distance between input patterns and the center patterns are found and passed as input for the hidden layer. The summed value is passed through an exponential activation function and the outputs are obtained in the output of hidden layer. These outputs are processed with target outputs to obtain the final weights, which will be used for testing.
Fig. 5.2 Training with Radial Basis Function
**Fig.5.3 Radial Basis Function output**

**Figure 5.3** shows intrusion detection in sample packets. The blue color dots indicate the presence / absence of intrusion.
5.3 BPA with RBF FOR INTRUSION DETECTION

**Step 1:** Features of a packet is presented in the input layer of BPA.

**Step 2:** Forward propagation of the features through the hidden layer and to the output layer.

**Step 3:** The error is input to the input layer of the RBF.

**Step 4:** The intrusion target is presented in the output layer of the RBF.

**Step 5:** The training process is continued till all the patterns are presented and the summation of error in the output layer of the BPA is less than a specified value.

![Fig.5.4 Topology of BPA with RBF](image-url)
5.4 BPA with ESNN RBF FOR INTRUSION DETECTION

![Diagram of BPA with ESNN](image)

**Fig.5.5 Topology of BPA with ESNN**

**Step 1:** Features of a packet is presented in the input layer of BPA.

**Step 2:** Forward propagation of the features through the hidden layer and to the output layer.

**Step 3:** The error is input to the input layer of the ESNN.

**Step 4:** The intrusion target is presented in the output layer of the ESNN.

**Step 5:** The training process is continued till all the patterns are presented and the summation of error in the output layer of the BPA is less than a specified value.

5.5 RBF with ESNN RBF FOR INTRUSION DETECTION

**Step 1:** Features of a packet is presented in the input layer of RBF.

**Step 2:** Distance among input pattern and features are found.

**Step 3:** RBF matrix is formed to obtain final weight matrix.

**Step 4:** Feature patterns are presented to the input layer of RBF.

**Step 5:** The features are processed with final weights of RBF to obtain value in the output layer of RBF.
**Step 6:** The output of RBF is input to ESNN and state vector is formed. Step 4 to step 6 is repeated for all the training patterns.

**Step 7:** Final weights are obtained for the ESNN.

**Step 8:** To detect presence of intrusion, features of a packet is presented into the input layer of RBF. The features are processed with final weights of RBF. The output of RBF is processed with final weights of ESNN. The output layer of ESNN gives the presence or absence of intrusion.

![Fig.5.6 Topology of RBF with ESNN](image)

### 5.6 SUMMARY

This chapter has discussed the implementation of Radial basis function, BPA with ESNN, RBF with ESNN applied to intrusion detection. Chapter 6 presents results and discussions.