A Radial Basis Function Network Based Classifier for Detection of Selected Tea Pests

Gouravmoy Banerjee  
Agro-Computing Research Laboratory  
Dept. of Computer Science, Ananda Chandra College, Jalpaiguri-735101, India

Uditendu Sarkar  
SCIENTIST-E, National Informatics Centre, Ministry of Electronics and Information Technology, Govt. of India, Collectorate Buildings, Jalpaiguri-735101, India

Indrajit Ghosh*  
Agro-Computing Research Laboratory  
Dept. of Computer Science, Ananda Chandra College, Jalpaiguri-735101, India

DOI: 10.23956/ijarcsse/V7I5/0152

Abstract- Tea is one of the most important cash crops of India. It is cultivated across a large area throughout the country. Being a perennial crop one of the major challenges for tea growers is mitigation of pest attacks. Proper identification of the active pests in the field is very tough and requires expertise. Moreover, human experts are not readily available on demand. In this paper we propose a radial basis function network based classifier for the identification of selected tea pests. The model so developed has an accuracy of about 99% in identifying the pests while testing with real field data.

Keywords: Radial Basis Function, Artificial Neural Networks, Tea Pest, Integrated Pest Management.

I. INTRODUCTION

Tea (Camellia Sinensis) is one of the major cash crops of India that contributes to about 11% of the total tea export of the world [1]. The amount of land under tea cultivation in India is about 563.98 Hectares and the net production of tea in the year 2013-14 was about 1208.78 million Kgs [2]. The Tea industry employs about 1.1 million workers out of which one half are women [3]. Tea industry is a pillar to the economy of our country.

Tea is a perennial plantation crop, so it is a favourable breeding ground of various insect pests throughout the year. The amount of crop loss due to pest attacks is heavy leading to loss of revenue and man hours. One of the major challenges in the production of tea is to mitigate the attacks of insect pests by adopting control measures. The primary task is proper identification of active insect pests which is a very tough some job requiring human expertise, experience and judgement. But, the availability of experts is scarce and often not available on demand for this wide spread industry. So in a large number of cases, active insect pests are not identified properly and inadequate control strategies are the ultimate result. As an alternative, a computer aided pest identification system can be developed. In fact, various systems have been proposed with their merits and limitations.

Several works using Radial Basis Function (RBF) have been reported in different fields such as survival chances of a burn patient [4], diagnosis of diabetes [5], Terrain Data Classification [6], Diagnosis of several diseases [7], Coffee Disease Recognition [8], Prediction of Palm oil production [9], Brain tumor classification [10], Construction cost estimation [11], Surface roughness prediction [12], Quality Assessment of poultry egg [13], Mild cognitive disorder and Alzheimer’s disease diagnosis [14], Solar radiation estimation [15]. But, to the best of our knowledge no such work has been done in the case of tea pest identification.

In this paper we have proposed an artificial neural network based solution that employs Radial Basis Function model for the identification of three major tea insect pests namely Aphids (Toxopteraaurantii Boyrde Fons.), Red Spider (Oligonychus Coffeeae) and Thrips (Taeniothripssetiventris Bagnall). The model have been trained and tested with real field data collected from the seven tea gardens of the districts of North Bengal and two tea gardens of the state of Assam.

This paper discusses the missing values and data pre-processing issues in insect pest identification in Section II. The details of the proposed model have been presented in Section III. The obtained results have been produced in Section IV. Finally, the outcome of the paper is concluded in Section V.

II. MISSING VALUES AND DATA PRE-PROCESSING

The relevant dataset collected from the tea gardens contains various signs and symptoms of attack as features related to the active insect pests, the identification by the experts and the final outcome. We have divided eleven major features into more refined attributes forming sub-categories for each, as presented in Table I.
This refinement allows us to understand the features with more intricate knowledge which in turn smooth the process of identification. Each of the refined feature was assigned a value 1 (one) for presence and 0 (zero) for absence while preparing the data for training the network.

Being a real field based experiment, such identification problems often suffer from missing data problems. Several methods of handling missing data values such as replacing the missing value with mean, mode etc., or using multiple imputation techniques have been suggested by Allison [16]. The refined features cannot be assigned a numerical magnitude to denote its value because they are linguistic in nature. These features have been observed from real field and have been assigned a 0 (zero) value if absolutely certain that the feature is absent and 1 (one) if the feature is present with absolute certainty. We can observe that the assigned numeric values inherently probabilistic in nature. So, we propose a frequency based probabilistic method to tackle the missing value problem for our current experiment. The methodology of the model is as follows:

The dataset obtained from the field have been grouped into n categories according to the different pest outcomes and frequency of occurrence ($\gamma$) of each major feature has been obtained as follows:

$$\gamma = \frac{O_f}{O_t}$$

Where $O_f$ is the number of observation present for a particular feature and $O_t$ is the total number of observations. The missing value has been obtained by calculating the probability of each missing observation ($P$) as follows:

$$P = \frac{\gamma}{n}$$

The probabilistic distribution attempts to capture the chances of occurrence for each of the sub-features by distributing the frequency of observation between each of them. As an example, if the frequency for site of damage feature is 7 (seven) then the value assigned to each of the sub-features i.e. Young leaf, Matured leaf, Bud & young leaf and Tender stem is 7/4=1.75.

### III. IDENTIFICATION OF PESTS BY RADIAL BASIS NETWORK

Radial Basis Function network creates a number of radial structures that are used to classify the input vector into its corresponding output class. The output class is evaluated depending the Euclidian distance of the input vector and the stored vector usually called the centre. The radial basis function network consists of three different layers; an input layer, a single hidden layer and an output layer. The weights between the input and the hidden layers are fixed to 1, the weights from the hidden to output layer tend to vary from one iteration to another during the learning phase. The Hidden layer
stores the radial centres along with the radius of each cluster (class). At first the input vector is used to calculate the Euclidian distance from the stored vector, this distance parameter $z_i$ is then fed to an activation function to generate the intermediate result. The final output of the network is obtained by obtaining the weighted sum from the hidden to output layers and a linear activation function.

Different types of activation functions can be used in Radial Basis Function networks such as Gaussian, Multiquadrics, Inverse Multiquadrics [17]. In our current experiment we have used the Gaussian Activation function because the other two functions require the centres to be different or non-singular whereas our dataset being a real field experimental data contains redundant information which can only be captured by Gaussian distribution. Similarly several learning mechanism for Radial Basis Function networks have been suggested such as Pseudo Inverse, Gradient Descent and Hybrid Leaning [18]. We have employed supervised Gradient Descent technique to train our network as the data was obtained from different tea gardens in a cumulative manner.

Using Gradient Descent rule or commonly known as Error Back-propagation rule we can adjust the free parameters like the weights (W) from the hidden to output layer, the radial spreads $\sigma$ and the radial centres $C_i$ in a supervised manner. The biggest advantage of this method is that it helps us to track the error in the network with each iteration or epoch and adjust the parameters accordingly.

In our current experiment we have used 31 input parameters each representing the refined features and 3 output classes corresponding to the three pests to be identified. Before training the network we have initialized the radial centres $C$ in the hidden layer by random distribution of the input vectors. The weights (W) between the hidden and the output layers were also set to small random numbers and the radial spreads $\sigma$ were set as follows [19]:

$$\sigma = \frac{\text{Max distance between two centres}}{\sqrt{\text{Number of centres}}}$$

In the training phase, the Euclidian distance between the input pattern $i$ and the radial centres are obtained as follows:

$$z_i = \sqrt{\sum (x - x_i)^2}$$

By using Gaussian activation function, the calculated activation output ($\phi_i$) of the hidden layer has been obtained as:

$$\phi_i = e^{-\frac{z_i^2}{2\sigma^2}}$$

The weighted sum ($\Sigma\phi W$) is calculated and then added with a bias (B) value to get the value of the final output.

$$Output = \Sigma\phi W + B$$

After getting the output, the error ($\varepsilon$) in the result has been obtained by using RMS rule.

For $i$-th pattern,

$$\varepsilon = \frac{1}{2} \sum_{i=1}^{n} (Target_i - Output_i)^2$$

The error ($\varepsilon$) is used to modify the weights $W$, centres $C$ and spreads $\sigma$ as follows [20]:

$$W_i(k + 1) = W_i(k) - \eta_1 \frac{\partial \varepsilon_k}{\partial W_i^k}$$

$$C_i(k + 1) = C_i(k) - \eta_2 \frac{\partial \varepsilon_k}{\partial C_i^k}$$
\[ \sigma_i(k+1) = \sigma_i(k) - \eta_3 \frac{\partial E_k}{\partial \sigma_i} \]

Where \( \eta_1 \), \( \eta_2 \) and \( \eta_3 \) are learning rates and \( i \) vary from 1 to \( q \), where \( q \) is the total number of input patterns used for training the network. For each pattern we repeat all the steps until and unless a termination condition is reached; for example, number of epochs <=40.

IV. EXPERIMENTAL RESULTS AND DISCUSSION

The entire experiment was coded using MATLAB R2008a software and the pre-processing was done by MS-Excel 2007 spreadsheet software. The experiments have been carried out using four different Radial Basis Function architecture configurations; consisting of 33, 32, 31 and 30 hidden units while the number of inputs and outputs remained the same. All of these models have been trained using Gradient descent algorithm to adjust the free parameters of the network using the procedure mentioned in the previous section. The termination condition for the learning phase was set to 40 epochs for all the models. Out of the total 228 cases we have used 182 cases to train the network and the rest 46 to test it in 80:20 ratios. The best value of the learning rates \( \eta_1 \), \( \eta_2 \) and \( \eta_3 \) were found at 0.1, 0.25 and 0.3 respectively by trial and error method. The performance of the model is presented in Table II.

<table>
<thead>
<tr>
<th>Configuration</th>
<th>( \eta_1 )</th>
<th>( \eta_2 )</th>
<th>( \eta_3 )</th>
<th>Epochs</th>
<th>Training error %</th>
<th>Testing error %</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-33-3</td>
<td>0.1</td>
<td>0.25</td>
<td>0.3</td>
<td>40</td>
<td>23.82</td>
<td>0.0314530</td>
</tr>
<tr>
<td>31-32-3</td>
<td>0.1</td>
<td>0.25</td>
<td>0.3</td>
<td>40</td>
<td>7.82</td>
<td>0.0710840</td>
</tr>
<tr>
<td>31-31-3</td>
<td>0.1</td>
<td>0.25</td>
<td>0.3</td>
<td>40</td>
<td>3.56</td>
<td>0.0037502*</td>
</tr>
<tr>
<td>31-30-3</td>
<td>0.1</td>
<td>0.25</td>
<td>0.3</td>
<td>40</td>
<td>7.32</td>
<td>0.0040082</td>
</tr>
</tbody>
</table>

It is clearly inferred from the above result that the model with 31 hidden units marked by ‘*’ performs better in detection of the pests selected for this study with an accuracy of 99.99% (100-0.0037502).

Some attempts have been made in the past for detection of tea pests using artificial intelligence techniques such as rule based expert systems [21], case based reasoning [22] and multilayer perceptrons [23]. This proposed model overcomes the drawbacks of rule based systems which are unable to handle uncertainty i.e. missing data problem. Case based reasoning approach offers a low accuracy rate as compared to this model. On the other hand the multilayer perceptrons suffers from complexity due to large number of training iterations. But, this model gives better accuracy rate with comparatively less number of iterations.

V. CONCLUSION

In our present experiment, we have developed a Radial Basis Function Network model for detection of three major tea pests. The model was trained using Gradient descent rule for adjusting the free parameters of the network. The model performs very well with only about 1% testing error in the present study. Our future intention is to extend it for detection of all major pests of tea and other cash crops of North Bengal region.

REFERENCES


