

AN EFFICIENT MODELLING AGRICULTURAL PRODUCTION USING ARTIFICIAL NEURAL NETWORK (ANN)

*¹Ms. Jeeva C, *² Mrs. Shoba S.A.,

*¹M.Phil Research Scholar, PG & Research Department of Computer Science & Information Technology , Arcot Sri Mahalakshmi Women's College, Vilapakkam, Vellore, TamilNadu, India.

*²Assistant Professor, HOD of PG & Research Department of Computer Science & Information Technology Arcot Sri Mahalakshmi Women's College, Vilapakkam, Vellore, TamilNadu, India.

Abstract -

Data mining is the process of extracting important and useful information from large sets of data. The goal of the data mining process is to extract knowledge from an existing data set and transform it into a unique human understandable format for some advance use. Data mining in agriculture is in relation to novel research field. Some efficient techniques can be developed and tailored for solving complex agricultural problems using data mining. Agriculture and allied activities constitute the single largest component of India's gross domestic product, contributing nearly 25% of the total and nearly 60% of Indian population depends on this life's precise profession. Due to vagaries of climate factors the agricultural productivities in India are continuously decreasing over a decade. The reasons for this were studied mostly using regression analysis. In this paper an attempt has been made to compile the research findings of different researchers who used agriculture data. This paper summarizes the application of data mining techniques such as k-means, bi clustering, k nearest neighbor, Neural Networks, Support Vector Machine and Naïve Bayes Classifier in the agriculture field.

Key Words: Support Vector Machine, Agriculture data. naive Bayes Classifier, artificial neural network,

I. INTRODUCTION

Agriculture is the mainstay of all economic activity. Self sufficiency in agricultural production gives a nation the pride of being able to feed its population on its own. Unfortunately, agriculture is the most vulnerable profession for farmers, since it is affected unforeseen factors like heavy rain, lack of rain, insect attacks [3], and inherent qualities of the soil and policies of the Government. Hence, prediction of agricultural production on an annual basis is tough job, but worthy of the effort since it will give the farmers a foresight of what they could expect in the next year.

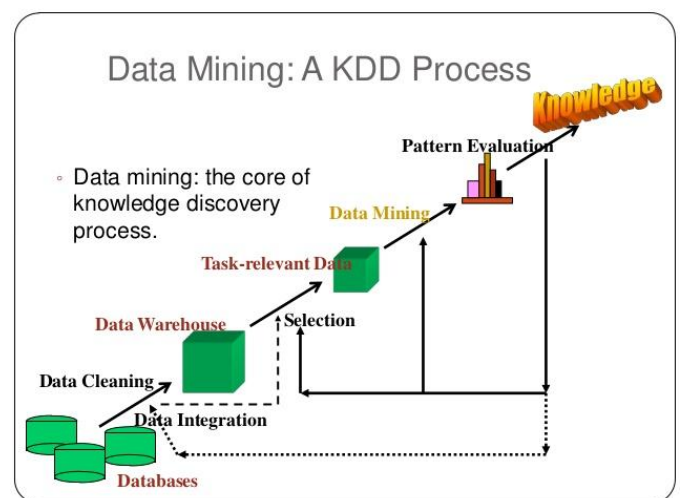


Fig : Data mining KDD Process

Artificial Neural Network (ANN) is a versatile modelling system which permits the accomplishment many complicated tasks with ease. ANN is the result of extensive research carried out in the field of Artificial Intelligence (AI) and machine learning problems around the world[2]. The tool easily adopts itself to the prediction of mathematical relations, recognition of particular patterns, error correction and noise removal in signals. Hence, the application of ANN for predicting agricultural production in Tamilnadu might enable estimation of the yield levels of crops as a function the annual rainfall levels in the previous years[6]. Data mining is a process of extracting/equating the meaningful data from the large database by using different tools and techniques. Data mining (sometimes called information or knowledge discovery) [3] is the process of evaluating data from different outlooks and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of diagnostic tools for analyzing data. It allows users to investigate data from many different dimensions or angles, categorize it, and summarize the relationships identified [4][5]. Technically, data mining is the process of finding correlations or forms among dozens of fields in

large relational databases. Data mining or knowledge discovery in databases (KDD) is an interdisciplinary field where we integrate techniques from different fields including data base systems, statistics, mathematics, high performance computing, artificial intelligence and machine learning.

Data mining in agriculture is a very recent research topic. It consists in the application of data mining techniques to agriculture. This data mining techniques used in agriculture for prediction of problem, disease detection, optimizing the pesticide and so on[7]. Recent technologies are nowadays able to provide a lot of information on agricultural-related activities, which can then be analyzed in order to find important information and to collect relevant information. This data mining techniques are used for disease detection, pattern recognition by using multiple application. Data mining is about to identify the similarities between searching the valuable business information from the large database systems such as finding linked products in gigabytes of store scanner data or the mining a mountain for a vein of valuable dataset. Both kind of processes required either shifting through an immense amount of material, or to perform the search intelligently so that exactly match will be performed. Data mining can be done on a database whose size and quality are sufficient [2][5]. The technology of data mining can generate new business opportunities by providing these capabilities:

1.1 Agriculture as the Backbone of Economy

Agriculture in India is an unorganized industry, where the rules of thumb govern many decisions rather than scientific insight. This is due to the difficulty in understanding the factors which affect agricultural production every year.

1.2 Objectives of the Study

Following are the objectives of the present study:

- 1) To study the relation between rainfall and crop yield patterns in Tamilnadu.
- 2) To generate an Artificial Neural Network (ANN) based model for predicting the annual rainfall and crop production levels of current year as a function of rainfall in three previous years.
- 3) To measure the error levels and fine tune the neural network for optimal performance.

1.3 Data for Artificial Neural Network

The data used for the generation of the Artificial Neural Network was obtained from AGRISTAT (2006) report. The data was used for training the network after normalizing it to the range of 0 to 1. Data mining techniques are able to get the benefits of automation on existing software and hardware platforms that can be implemented on new systems which can be upgraded and new products can be developed. When data mining tools

are defined on high performance parallel systems, they can be analyzed with massive databases in minutes. Faster processing is required in such system to derive the effective results from complex systems. High speed processing and accurate outcome from the system makes it possible for users to analyze large set of information. Larger databases, in turn, gives more improved predictions.

- Agricultural organizations store huge amounts of data in the form of crop databases. Trends in these databases can be identified using data mining practices, which sort and model the data in order to arrive at a conclusion. The data mining applications present the data in the form of data marts.[12]
- In the agricultural industry, however, the lack of standard vocabulary has hindered the process of data mining to a certain extent. This could lead to unnecessary problems, during the process of data mining. The increase in the use of standardized terms will reduce the percentage of errors in the data mining process. [12]

1.4 Introduction to Artificial Neural Network (ANN)

Artificial Neural Network is a replica of the biological neural network, which helps in the several fields like function approximation, pattern recognition (finger print identification, face recognition etc.) and error correction (removal of noise from incoming signal)[3]. ANN is organized in the form of neurons grouped into layers. Usually, there are two layers reserved for taking input data and sending the computed results. There may be some hidden layers, lying between the input layer and the output layer. Some of the essential terms related to ANN are presented here.

Data mining algorithms primarily based on statistical techniques and machine learning algorithms. During the design of models, algorithms are used. Not only algorithms are important in this context but also data itself is critical there are two forms of data analysis and study so that we can examine the things and gets the knowledge that can be used for extract models describing important classes or predict future data trends[9], [3]. These data analysis help us to provide a better understanding of large data these two forms are as follows:

- **Classification:** In Classification prophesies unconditional and calculation models predicts continuous valued functions. For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.
- **Prediction:** In prediction which is used to predict the data according to the given data. Following are the examples of cases where the data analysis

task is Prediction For example the marketing manager needs to predict how much a given customer will spend during a sale at his company. In this example we are bother to predict a numeric value. Therefore the data analysis task is example of numeric prediction

1.5 Java Object Oriented Neural Engine (JOONE)

Java Object Oriented Neural Engine (JOONE) is a free software library written in Java programming language for generating neural networks. The power of the framework lies in the fact that Java programming language inherently supports multi-threading, serialization, networking and embedded systems. These capabilities enable a neural network object created using JOONE library to run across several platforms, like mobile devices, desktop computers and main frames. Hence, JOONE provides a powerful Application Programming Interface for developing Object Oriented Neural Networks. The present study uses JOONE for generating feed forward neural network to predict agricultural production.

RELEATED WORK

1. In Year 2010, Li Zhanfeng performed a work, "An Unattended Detection Method of Soil Respiration". This method uses proposed utility line peak extraction method to extract the CO₂ concentration in the soil respiration chamber, and sets up a mathematical model of soil CO₂ concentration. The system equipment is independently developed and it is verified that the extraction method is feasible with the site automatically detection [1].
2. In Year 1994, Metin Akay performed a work, "Automated Noninvasive Detection of Coronary Artery Disease Using Wavelet-Based Neural Networks". This study examines the utility of neural networks for detecting coronary artery disease noninvasively by using clinical examination variables and extracting useful information from the diastolic heart sounds associated with coronary occlusions. It has been widely reported that coronary stenosis produce sounds due to the turbulent blood flow in these vessels [2].
3. In Year 1991, Xavier SIMON performed a work, "Plot Analysis for Crop Identification and Disease Detection". Most methods and algorithms use satellite images as a main or exclusive source of information. Author describes a crop identification method which uses geographic information to identify agricultural plots on a time succession of images, and analyses its history through several years of cultivation. The pixels of each plot are considered as a statistical

population and the distribution parameters of the distributions of their radiometric values are calculated for each wavelength channel. Author use these parameters to perform discriminant analyses bearing on the test-site plots and, at a later stage, on the surrounding plots. The results of this classification are the first step of a deductive method which allows a precise and early crop identification and further investigation of other agricultural parameters, such as damage caused by plant disease [3].

4. I.V.Bragin performed a work, "Passive icrowave Complex of Remote Sensing for the Detection of Objects in the Soil". The paper deals with some grounding in theory of a passive radiometric complex designed to detect from a helicopter metal and non-metal objects in the soil at the depth of several meters [4].

II. OVERVIEW OF JAVA OBJECT ORIENTED NEURAL ENGINE (JOONE)

2.1 Introduction

Java Object Oriented Neural Engine (JOONE) is a Java framework to build and run AI applications based on neural networks. JOONE applications can be built on a local machine, be trained on a distributed environment and run on a wide range of devices. JOONE consists of a modular architecture based on linkable components that can be extended to build new learning algorithms and neural networks architectures.

All the components have some basic specific features, like persistence, multithreading, serialization and parameterisation. These features guarantee scalability, reliability and expansibility, all mandatory features to reach the final goal to represent the future standard on the AI world. JOONE applications are built out of components. Components are pluggable, reusable, and persistent code modules.

A great need of the industrial market is to have the possibility to resolve business problems suitable with neural networks (or with AI applications in general). JOONE wants to represent the optimal solution to build applications to satisfy such needs (i.e. bank loan assessment, sales forecasting, etc.). Its characteristics are optimal to build custom applications driven from the user's needs, where it's important to have flexibility, scalability and portability. The core engine of JOONE is already suitable for small devices, having a small footprint and running on Personal Java environments.

2.2 Features OF JOONE

The JOONE library is built to take advantage of the multi-threaded nature of Java programming language. The networks built using JOONE library can work asynchronously if required, in order to provide accomplish the job of prediction.

The JOONE library permits the addition of layers of neurons, whose input / output are linked to preceding / succeeding layers using transmission objects called synapse. Each layer of neurons can be a linear layer, softmax layer, sigmoid layer, hyperbolic tangent layer, logarithmic layer etc.. The choice layer type and the number of neurons in a layer are to be exercised by the developer in such a way that the error produced by the network remains a minimum.

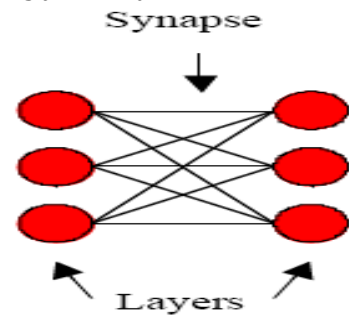


Fig. 2.1 Layers and Synapse

Fig. 2.1 shows the arrangement of layers and synapse. Fig. 2.2 shows the computational process occurring within a single neuron. The computation of the net input X_n can be done using the expression,

$$X_n = \sum_{i=1}^N x_i W_{ni} + bias \tag{2.1}$$

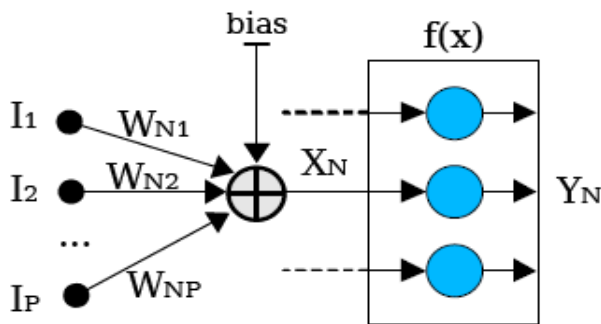


Fig. 2.2 Computation within a Neuron

When the input signal travels from the input layer to the output layer, the result is available from the network. During training process, the result is compared against desired result for the given input and the connection weights are adjusted to minimize the error in the result.

2.3 Types of Layers in JOONE Library

Some of the typical layer types available in the JOONE library are presented along with the mathematical function used for computing the output of the neurons belonging to the layer. This mathematical relation is known as the transfer function.

Linear Layer produces output in proportion to the input signal, based on the relation,

$$y = \beta \cdot x \quad \dots (2.2)$$

Biased Linear Layer adds a constant value called bias to the result of a normal linear layer. This layer does not have any scalar parameter like β in the linear layer. Hence, the transfer function of a biased linear layer is,

$$y = x + bias \quad \dots (2.3)$$

2.4 Essential Classes In Joone Library

Some of the essential classes in the JOONE library used for generating a neural network are introduced in this section. These classes are essential for programming neural network objects.

NeuralNetwork class is contained in the package org.joone.net. This class can be used for generating a neural network object. Layers of neurons can be added to this neural network object.

Monitor class controls the network and maintains many useful data related to the training and working states of the neural network. Monitor class is introduced into the neural network because there is the necessity to control all the parameters interested in the running and/or training process. For this purpose the engine is composed by several other components designed to provide the neural network with a series of services.

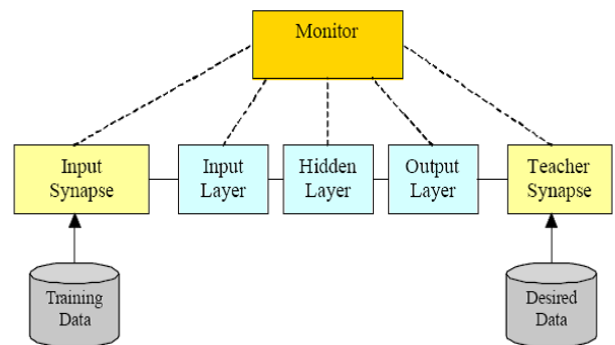


Fig. 2.3 Role of a Monitor Object

The Monitor provides services not only to the internal components of a neural network, but also to the external application that uses it. The Monitor object, in fact, provides any external application with a notification mechanism based on several events raised when a particular action is performed. For instance, an external application can be advised when the neural network starts or stop the training epochs, when it finishes a cycle or when the value of the global error (the RMSE) changes during the training phase.

Teaching Synapse is used for training the network to predict values very close to the desired values. This class is contained in the package

org.joone.engine.learning. This class is responsible for adjusting the weights of neurons during the training process. The class is active only when the learning property of the monitor class is set to true.

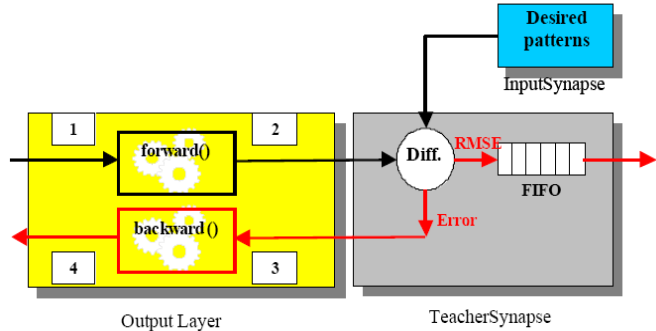


Fig. 2.4 Role of Teaching Synapse

2.5 Using JOONE Graphical User Interface

The JOONE library comes with a graphical user interface to enable visual development of neural networks. The neural networks created from this graphical user interface can be serialized and then accessed using other Java programs.

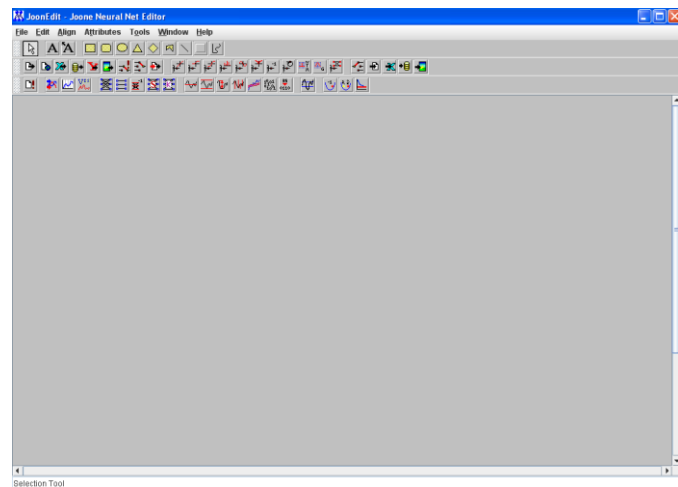


Fig. 2.5 Blank GUI Neural Workbench

2.6 Using the JOONE Engine For Coding

The JOONE package is distributed with a library called joone-engine.jar and lies in lib directory. This file should be added to the CLASSPATH environment variable before using the JOONE engine in Java programs.

JOONE based neural networks are developed and saved into a file using ObjectOutputStream class contained in java.io package. These network files can be later used by other programs to make predictions.

2.7 Measuring Network Performance

The performance of a JOONE based neural network can be measured against the both the training data and validation data. For measuring the performance

against training, getGlobalError method of Monitor class can be used.

Methodology	Applications
K-means	Forecasts of pollution in atmosphere Classifying soil in combination with GPS
k-nearest Neighbor	Simulating daily precipitations and other weather variable
Support Vector Machine	Analysis of different possible change of the weather scenario
Decision Tree Analysis	Prediction soil dept
Unsupervised Clustering	Generate cluster and determine any existence of pattern
WEKA Tool	Classification system for sorting and grading mushrooms.

Table : Data mining methodologies and its use in Agriculture domain

III. MODELLING AGRICULTURAL PRODUCTION

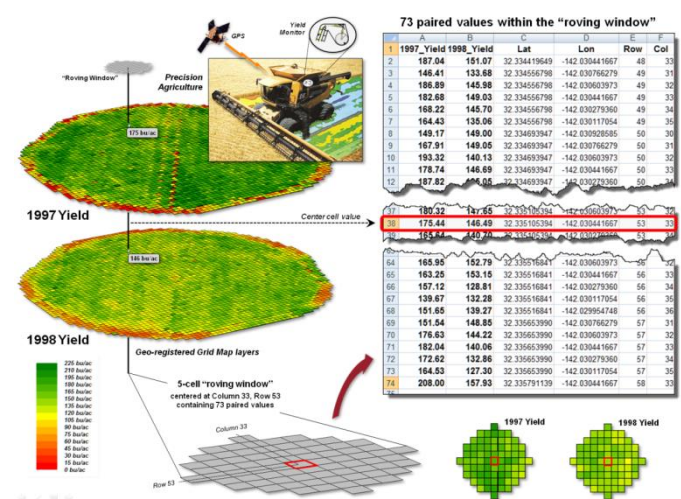


Fig : Precision Agriculture yield maps identify the yield volume harvested from each grid location throughout a field. These data can be extracted using a "roving window" to form a localized subset of paired values surrounding a focal location.

Each grid cell location identifies the paired yield volumes for the two years. The simplest comparison would be to generate a Difference map by simply subtracting them. The calculated difference at each location would tell you how different the yield is between the two years and where the greatest differences occur. But it doesn't go far enough to determine if the differences are "significantly different" within a statistical context.

An often used procedure for evaluating significant difference is the paired T-test that assesses whether the means of two groups are statistically different.

Traditionally, an agricultural scientist would sample several locations in the field and apply the T-test to the sampled data. But the yield maps in essence form continuous set of geo-registered sample plots covering the entire field. A T-test could be evaluated for the entire set of 3,289 paired yield values (or a sampled sub-set) for an overall statistical assessment of the difference. However, the following discussion suggests a different strategy enabling the T-test concept to be spatially evaluated to identify 1) a continuous map of localized T-statistic metrics and 2) a binary map the T-test results. Instead of a single scalar value determining whether to accept or reject the null hypothesis for an entire field, the spatially extended statistical procedure identifies where it can be accepted or rejected—valuable information for directing attention to specific areas.

The equation and solution steps are

$$T_{Statistic} = d_{Mean} / (d_{Stdev} / Sqrt(n))$$

Step 1. Calculate the difference ($d_i = y_i - x_i$) between the two values for each pair.

Step 2. Calculate the mean difference of the paired observations, d_{Avg} .

Step 3. Calculate the standard deviation of the differences, d_{Stdev} .

Step 4. Calculate the T-statistic by dividing the mean difference between the paired observations by the standard deviation of the difference divided by the square root of the number of paired value.

$$T_{Statistic} = d_{Avg} / (d_{Stdev} / Sqrt(n)).$$

3.1 Introduction

Agricultural production data is modelled using a customized Java class called AgriNet which extends Neural Network class of org.joone.net package. This AgriNet class is trained with the rainfall records for the last three years as input and the production of various agricultural crops as result. One network is trained for predicting one type of result and a total of thirty seven networks were generated for this study. The data for training the networks was obtained from AgriStat (2006) report compiled by the Directorate of agriculture for Tamilnadu. The data covers a time span of twenty five years from 1970 to 2004.

3.2 Input Parameters

Input parameters for the AgriNet class were the rainfall levels measured in milli meter for the last three years. Three year rainfall levels were chosen for input because rainfall in a single year did not show much correlation with the agricultural production of the same year. basic familiarity of the two fundamental data types supporting important for understanding map analysis procedures and capabilities (see author’s note).

3.3 Coding the Agrinet

The AgriNet class is coded as a feed forward network using the JOONE library with three layers. The input layer contained three neurons (corresponding to input count) and the output layer contained one neuron. Both input and output layers were linear layers. There was one hidden layer with three neurons and sigmoid transfer function. The schematic layout of the network is shown in Fig. 3.1. The source code for AgriNet is shown in Appendix 1.

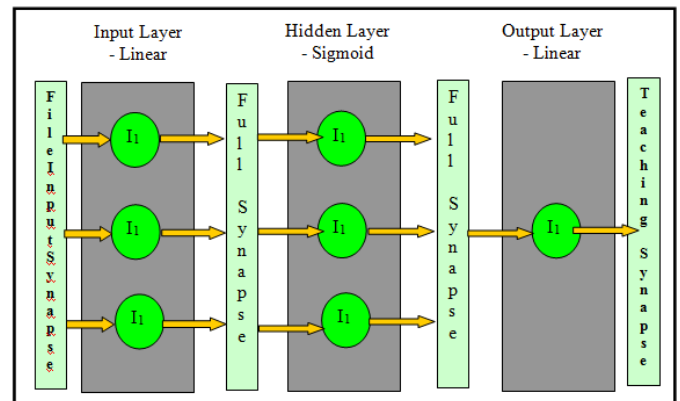


Fig. 3.1 Schematic Layout of AgriNet Class

3.4 Saving and Restoring Agrinet

Each of the networks was trained for predicting one of the 37 output parameters and saved in a file having .net extension. For example, rainfall network was saved into a file called Rainfall.net. The binary files were written using the WriteObject method of ObjectOutputStream class available in the java.io package.

The network objects saved into files were to be reloaded into memory in order make predictions. A graphical user interface bearing the title Agriculture in Tamilnadu was developed for getting input, loading the appropriate network into memory and making the predictions. The restoration was done using the readObject method of ObjectInputStream available in java.io package.

3.5 Normalizing Input and Denormalizing Output

The input data presented to the networks during training and testing were normalized to the range of 0 to 1. But, the input presented from the user need not be normalized since it poses an additional computational burden to the user. Hence, the AgriNet class was coded with a method called makePrediction which takes care to normalize the input data before presenting it to the network and denormalize the results after the predictions were obtained from the network.

IV. EVALUATION RESULT:

4.1 Graphical User Interface for Accessing Agrinet

A Graphical User Interface (GUI) was developed using the swing toolkit available in Java. The GUI was named Agriculture in Tamilnadu and it read rainfall levels for the three preceding years.

Predictions could be chosen by ticking the appropriately named check box buttons. Blank GUI is presented in Fig. 4.1.

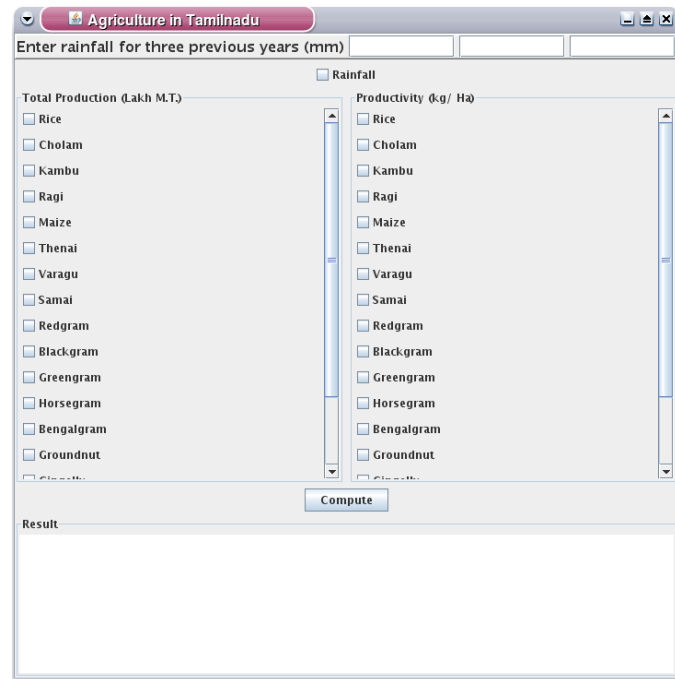


Fig. 4.1 Blank GUI for AgriNet

Figures 4.2 and 4.3 show typical predictions made from the GUI. The results are appropriately scaled by the AgriNet and are readily usable. Similarly, the input can also be presented without any scaling to the AgriNet, which takes care to normalize the data before using it for predictions. This project GUI is having five buttons as Weather, Image, Graph, Clear and Close. Weather button will show the dialog box showing the weather of that particular date. Image button will show the result in pictorial form. Graph button will show the output in the form of graph. Clear button will clear all the text box recordings of weather forecasting parameters shown in project GUI. Close button will exit from the application. The Pop up window displaying the output will vary with respect to the combination of weather parameters retrieved in the project GUI from the database on that particular date.

It also concludes that the Back propagation Algorithm can also be applied on the weather forecasting data. The results suggest that this Neural Network could be an important tool for weather forecasting. Neural Networks are capable of modeling a weather forecast system. Statistical indicators chosen are capable of extracting the trends, which can be considered as features

for developing the models. Statistical indicators except coefficients of skewness and kurtosis are found suitable to extract the hidden patterns present in whether data. The Neural Network signal processing approach for weather forecasting is capable of yielding good results and can be Considered as an alternative to traditional meteorological approaches. The Redistribution error rate of random forest is less than decision tree since random forest is ensemble of trees

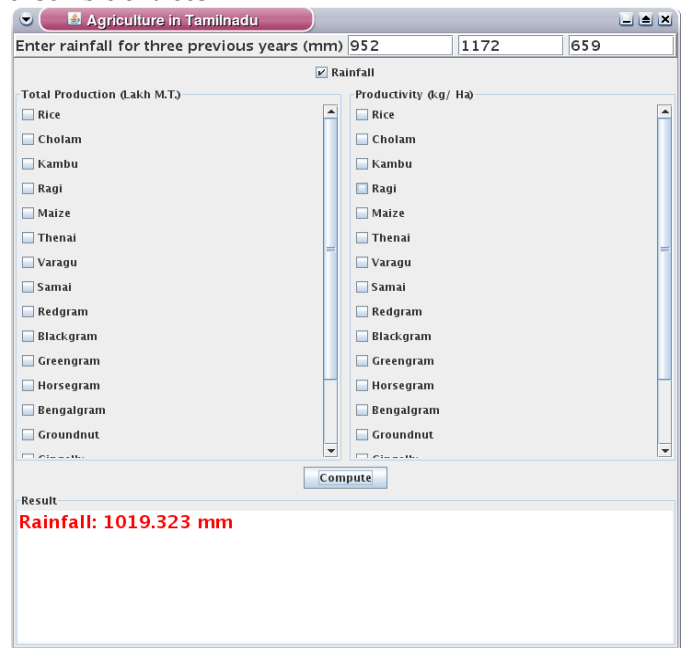


Fig. 4.2 Prediction of Rainfall Using the GUI

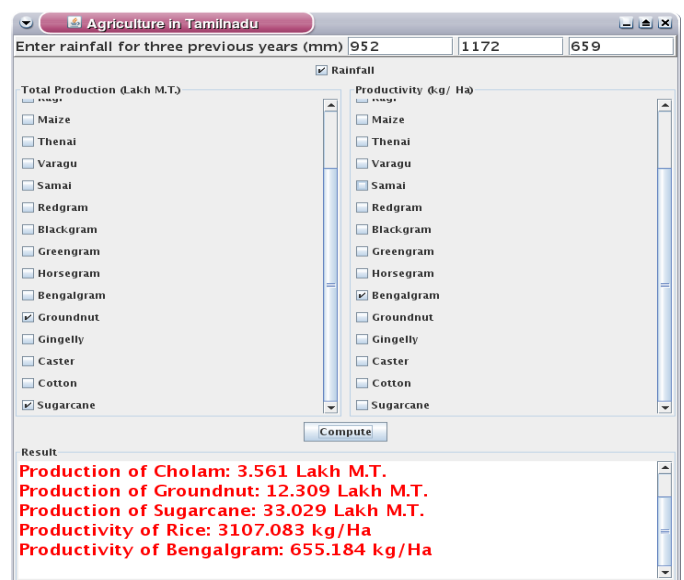


Fig. 4.3 Prediction of Crop Yield Using the GUI

CONCLUSION

Based on the JOONE based AgriNet developed for the present study, the following conclusions are drawn:

- i. The relationship between rainfall patterns and crop yield levels in Tamilnadu was established through this study.
- ii. The Java Object Oriented Neural Engine (JOONE) performs well for modelling neural networks suited for the prediction of annual agricultural yield.
- iii. The AgriNet objects could make predictions which were in very close agreement with the training data.
- iv. The Root Mean Squared Error (RMSE) for the training data was in the range of 0.05786002201052837 to 0.19946544757537438.
- v. The Root Mean Squared Error (RMSE) for the validation data was in the range of 0.057859982107289536 to 0.19946466001889254.
- vi. The Graphical User Interface (GUI) developed for the present study can help easy use of the AgriNet objects generated for predicting annual production and productivity of major crops in Tamilnadu.
- vii. The AgriNet could predict the expected annual rainfall for the current year based on the rainfall data for the three preceding years.
- viii. The AgriNet objects could be made to make predictions using different interfaces, like web based prediction models, since the objects are not tied to any specific front end.

Future studies might concentrate upon providing front ends which can be accessed from the network, so that anyone trying to predict the rainfall or crop yield levels might be able to use the AgriNet without any constraints.

REFERENCES:

1. 1---,(2006), Agristat 2006, Directorate of Agriculture, Chepauk, Chennai - 600 005, Tamilnadu, India, pp. 1-318.
2. Carpenter, W.C. and Barthelemy, J.F. (1994), Common Misconceptions about Neural Networks as Approximators, ASCE Jl. of Computing in Civil Engineering, 8(3), 345-358.
3. Chao, L.C. and Skibniewski, M.J. (1994), Estimating Construction Productivity: Neural Network Based Approach, ASCE Jl. of Computing in Civil Engineering, 8(2), 234-251.
4. Chen, H.M., Tsai, K.H., Qi, G.Z., Yang, J.C.S. and Amini, F. (1995), Neural Network for Structural Control, ASCE Jl. of Computing in Civil Engineering, 9(2), 168-176.
5. Coulibaly, P., Anctil, F. and Bobee, B. (2000), Daily Reservoir Inflow Forecasting using Artificial Neural Networks with Stopped Training Approach, Elsevier Jl. of Hydrology, 230, 244-257.
6. Demuth, H., Beale, M. and Hagan, M., (2006), Neural Network Toolbox for use with MATLAB®, The MathWorks, Inc., 3, Apple Hill Drive, Natick, MA 01760-2098, USA.
7. Elazouni, A.M., Nosair, I.A., Mohieldin, Y.A. and Mohamed, A.G. (1995), Estimating Resource Requirements at Conceptual Design Stage Using Neural Networks, ASCE Jl. of Computing in Civil Engineering, 11(4), 217-223.
8. Gangobadyay, S., Gautham, T.R. and Gupta, A.D. (1999), Subsurface Characterization using Artificial Neural Network and GIS, ASCE Jl. of Computing in Civil Engineering, 13(3), 153-161.
9. Hung, S.L. and Jan, J.C. (1999), MS_CMAC Neural Network Learning Model in Structural Engineering, ASCE Jl. of Computing in Civil Engineering, 13(1), 001-011.
10. Liong, S.Y., Lim, W-H. and Paudial, G.N. (2000), River Stage Forecasting Bangladesh: Neural Network Approach, ASCE Jl. of Computing in Civil Engineering, 14(1), 001-008.
11. Marcus, G. and Lounis, Z. (2005), Prediction of Onset of Corrosion in Bridge Decks using Neural Networks and Case Based Reasoning, Computer Aided Civil and Infrastructural Engineering, 20(2), 108-117.
12. Marrone, P. (2007), Java Object Oriented Neural Engine: The Complete Guide, <http://www.joone.org>, pp. 1-142.
13. Maru, S. and Nagpal, A.K. (2004), Neural Network for Creep and Shrinkage Deflections in Reinforced Concrete Frames, ASCE Jl. of Computing in Civil Engineering, 18(4), 350-359.
14. Oreta, A.W.C. and Kawashima, K. (2003), Neural Network Modeling of Confined Compressive Strength and Strain of Circular Concrete Columns. ASCE Jl. of Structural Engg., 129(4), 554-561.
15. Rao, M.M. and Dutta, T.K. (2006), Modal Seismic Control of Building Frames by Artificial Neural Network, ASCE Jl. of Computing in Civil Engineering, 20(1), 69-73.
16. Savin, D., Alkas, S. and Fazio, P. (1998), Calculating Weight Matrix of Neural Network for Resource Levelling, ASCE Jl. of Computing in Civil Engineering, 12(4), 241-248.
17. Shi, J.J. (2000), "Reducing Prediction Error by Transforming Input Data for Neural Networks", ASCE Jl. of Computing in Civil Engineering, 14(2), 109-116.