ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF INFORMATICS

DEPARTMENT OF INFORMATION SCIENCE

KNOWLEDGE BASED REASONING FOR AGRICULTURAL CROP MANAGEMENT DECISIONS:

AN EXPERIMENT USING RULE BASED AND ARTIFICIAL NEURAL NETWORK APPROACH

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE

BY

TSEGAW KELELA

JANUARY, 2009

ADDIS ABABA UNIVERSITY
SCHOOL OF GRADUATE STUDIES
FACULTY OF INFORMATICS
DEPARTMENT OF INFORMATION SCIENCE

KNOWLEDGE BASED REASONING FOR AGRICULTURAL CROP MANAGEMENT DECISIONS: AN EXPERIMENT USING RULE BASED AND ARTIFICIAL NEURAL NETWORK APPROACH

A THESIS SUBMITTED TO THE SCHOOL OF GRADUATE STUDIES OF ADDIS ABABA UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENT FOR THE DEGREE OF MASTER OF SCIENCE IN INFORMATION SCIENCE

BY

TSEGAW KELELA

Name and Signature of Members of the Examining Board

------------------, Chair person, Examining Board ---------------------------

Dr Dejene Ejigu, Advisor ----------------------------

--------------------------------- --------------------- ---------------------------

Chair person, Faculty Signature Date

------------------------- ----------------- --------------------------

Chair person, Graduate Council Signature Date

Acknowledgment

My first and special gratitude goes to my advisor, Dr Dejene Ejigu who patiently supervised this work with professional guidance and constructive feedback in the course of conducting this research.

I would like to forward my heartfelt gratitude for all the staff members of the faculty of informatics, particularly the department of information science, for their friendly assistance and encouragement while facilitating the conditions necessary for conducting this research.

I am also greatly indebted to Ato Abebe Cheffo, Ato Feredegn Dagnew, Ato Abebaw Gizaw, and Ato Mesfin Tessera, for their kind cooperation and commitment in providing me the domain information on vegetable production during the knowledge acquisition phase of this research work.

In addition, I would like to express my sincere gratitude and appreciation to all the staff members of Ethiopian Institute of Agricultural research and Ministry of Agriculture libraries for their positive cooperation in allowing me to use the resources during the exploration of relevant literatures, working manuals, reports and brochures.

Finally, I would like to appreciate and thank all members of my family who gave me the encouragement and moral from start to completion of this research.

Table of Contents

List of Acronyms

AI	Artificial Intelligence
ANN	Artificial Neural Network
AEZ	Agro Ecological Zone
CAPS	Computer Assisted Problem Solving
DAP	Diamonium Phosphate
DA	Development Agent
GDP	Gross Domestic Product
IMS	Intelligent Maintenance System
Kg/ha	Kilogram per hectare of land
MoARD	Ministry of Agriculture and Rural Development
MSE	Mean Square Error
NN	Neural Network
NPK	Nitrogen, Phosphorous, Potassium
PASDEP	Plan for Accelerated and Sustainable Development to End Poverty

List of Figures

List of Tables

List of Appendices

Abstract

Studies conducted so far and annual reports frequently issued by the national bank of Ethiopia indicated that agriculture remained the main source of living for the greatest portion of Ethiopia's population. Despite the fact that government is giving attention to the sector, most farmers are still using their traditional knowledge to solve crop related problems. The very limited number of specialized experts in the area, coupled with lack of appropriate technologies, contributed to the present low productive land and low income status of Ethiopian farmers. Maximizing yield potential and quality need the presence of proper expert decisions at a field level. In situations where there is a shortage of high level domain experts, automating crop management decision making has paramount importance. Expert system technologies can be used for automating crop management decisions as they have been effectively applied to solve problems in other domain areas of similar nature.

Based on the information gathered at the start of this research, the problem of the country's agricultural human resource is more intense in the area of vegetable production. With this background, this research is conducted to develop an expert system model as an attempt to automate the reasoning strategy of human vegetable experts. There are a number of approaches to develop expert systems ranging from rule based methods that represent knowledge in the form of IF-THEN rules to systems that employ machine learning techniques.

The approach adopted in this research uses the combination of the rule based and neural network methods with an aim to exploit the best features of the two methods. The system is modeled to have hybrid architecture by integrating rule based and neural network modules as a component of one single system.

In the course of building the hybrid model, knowledge acquisition, data preprocessing, rule generation, knowledge representation and model integration tasks had been performed. In the rule based module of the hybrid model, knowledge of vegetable experts was represented as rules. To build the neural network module and perform the integration with the rule based module, the fast artificial neural network libraries written in the C language were used after compiling and importing them in the prolog environment. The neural network module is built to handle user requests that may go beyond the capability of the rule based module. The artificial neural network module was integrated with the rule based module to create the hybrid vegetable expert system.

To measure the effect on performance after integration, ten random queries of consultation requests were presented to both the rule based module and the hybrid system. The hybrid system responded to eight of them while the rule based module alone provided answers to only five of these questions. The performance gain observed in the hybrid system is due to the neural network module embedded in it. The result obtained in this work showed that integration of the two approaches into one system produced better result and it is encouraging to advance the system into fully functional vegetable expert system.

Chapter one

Introduction

1.1 Background

Agriculture is the backbone of the Ethiopian economy. It has been a source of livelihood for more than eighty percent of the country's population. Reports in the 2006/2007 fiscal year from the National Bank of Ethiopia indicate that Agriculture remains a major contributor to overall economic performance with 45.9 percent share in the total GDP(Gross Domestic Product) followed by services and industry sector (40.8 percent) share in GDP. Generally, overall economic growth in Ethiopia is highly associated with the performance of the agricultural sector [1].

As per the information obtained from the document in the five year development strategic plan for sustainable development to end poverty (PASDEP), the Ethiopian government has been giving significant focus and attention for agriculture and rural development. Among the major agricultural activities run by the government, crop production takes the highest share which consists of cereals, vegetable cops and fruits [2].

In order to maximize the yield potential of these crops, there should be a systematic and modern way of crop management practices in areas of plant breeding, weeding, seeding, fertilizer application and disease control. Smooth running of these activities in turn demands availability of highly qualified personnel at a grass root level. Although the number of agricultural experts is getting increasing as compared to the past, there is still a significant shortage of specialized experts to support the needy farmers. When development agents (DAs) encounter technical difficulties in particular production areas, they usually request help from weredas or zones. There are situations where the experts are unable to respond to requests due to distance limitations. This affects yield potential and sometimes results in total crop loss due to delayed response for situations that need urgent technical support to farmers.

To accelerate and expand industrial development and increase overall economic growth, it is essential to support and develop the agriculture sector which is crucial to ensure the provision of inputs for industries and hence, to fulfill food requirements. This can be achieved only when crop management decisions are assisted with expert knowledge and utilization of up-to-date technologies. There have been a number of research efforts made to automate decision making strategy of human experts in other domain areas; such as health and law.

This research is initiated with an aim to contribute to the effort in technology adoption and utilization for assisting vegetable crop management decisions; thereby reducing the time and effort required to provide expert advice to development agents. Development agents are lower qualified people, usually diploma holders, having only general knowledge with no technical specialization. These people are assigned to serve farmers at a community level.

Scarcity of high level experts in the area of vegetable production attracted the researcher to focus on representing and modeling domain knowledge of experts in the field so that the research output can serve as a landmark in realizing a system that can provide expertise advise in absence of specialized vegetable experts.

1.2 Statement of the Problem and Justification

In the past couple of decades, the number of vegetarians is increasing fast because, it has been realized that eating vegetables make people healthy and cheaper than eating animal proteins. To develop a sustainable strong farmer labor force, production and consumption of green leafy vegetables should be included in the agricultural development programs [3].

Crop management decisions at various situations demand high level expertise knowledge and practical experience about the particular crop. This in turn requires availability of qualified manpower at various levels; both at wereda and kebele levels to support farmers in the production of agricultural crops. However, the practical situations in the country dictate that there is a significant shortage of trained manpower in the sector [2].

As per the information obtained from vegetable experts, there are a lot of constraints that hamper the growth of crop production such as:

- Less or limited number of senior experts in the regions
- Low level of expertise knowledge and training /education held by development agents who work at a community level
- Prevalence of pest and crop disease which need Scientific and urgent decision at a field level.
- Poor communication between the farmers, research centers and agriculture bureaus
- Absence of infrastructure or communication medium to disseminate information to farmers

The problem is more intense in some specialized fields such as horticulture and vegetable crops. In some regional states, one vegetable expert serves at zonal levels [2]. We can see that there is a real problem of getting immediate access to knowledge and support to farmers and development agents. It is very difficult to have (at least in the short term) the required number of experts for each kebele or locality. The solution to the problem needs to involve a structural change in human capacity, input supply, technology adoption and provision of infrastructure. Besides, there should be a means of representing scientific and expertise knowledge in such a way that knowledge based reasoning or decision making can take place in the absence of human experts. This research is initiated to assist this effort by exploring the potential of expert system technologies for crop management decisions.

There are three fundamental issues that seem to challenge the applicability and effective utilization of the research output. These include Finance (affordability), access to electric power and knowledge of computer skills on the part of Development Agents (DAs). However, there is still a promising tendency in the environment to overcome these constraints and application of the research output is potentially feasible for the following reasons:

- The price of computer and electronics equipments is surprisingly decreasing every year. Thus, the trend is towards having access to low cost technology and there is a promising future for these resources to be affordable to everyone or organizations.

> *It should come as no surprise that computing costs have been falling steadily for the last several years [4].*

- The Ethiopian electric power corporation has a rural electrification program for the next five years. This will create a favorable condition to get access to electric power for computers and other electronic devices. In addition, solar technologies are now becoming common source of power [5].
- The system does not require users to have advanced knowledge of computers. It only requires basic knowledge of how to open, close and change files in the computer environment. Since Information technology is given as one of the school subjects starting from preparatory to university level, almost all DAs already have the opportunity to acquire this basic knowledge required to use the system. It is also possible to provide the training with very minor orientation.

To sum up, capturing the knowledge of vegetable experts and looking for the best possible way of modeling them is a research worthy problem. Besides, it is potentially feasible to implement the output of this research with the existing knowledge and progressively advancing infrastructure

1.3 Objective of the Study

The next two sections will describe the general and specific objective of the research:

General Objective

The general objective of this research is to explore the potential application of the expert system technology and develop a rule based – Artificial neural network hybrid model that can assist expert decision making in the production of vegetable crops.

Specific Objectives

The specific objective of the study includes the detailed sub goals that must be achieved in order to attain the general objective. These include:

- To study the foundations of expert systems with their application to agricultural crop management decision making.
- To prepare the knowledgebase for the study in consultation with senior vegetable experts
- To build the model for the hybrid vegetable expert system
- To develop the prototype and implement the system based on the developed model
- Evaluate the performance of the prototype system.
- Draw conclusions based on the findings of the research

1.4 Research Methods (approaches)

1.4.1 Data Sources

Relevant data for this research are collected from various sources in two different ways:

Primary Data Sources

Knowledge of the domain area is collected through interviewing vegetable experts. Efforts had been made to interview senior experts with diversified experiences. That is, experts working at different capacities were interviewed to acquire the necessary knowledge.

Interviews were conducted by forwarding open ended questions to the experts and their responses were structured for inclusion in the rule base of the system.

Secondary data sources

For the sake of getting a deeper insight about the characteristics of exotic and indigenous vegetables and to strengthen the information obtained from experts, literatures related to fertilizer application and disease control were reviewed.

1.4.2 Sampling /Data selection method

Even if there are a number of vegetable crop species and varieties, it was difficult to represent all the relevant knowledge for all species. Hence, only three vegetable crops (onion, tomato and potato) were selected based on their economic importance to farmers, level of technical difficulties on the part of farmers and development agents on these vegetables. The selection was done in consultation with vegetable experts.

1.4.3 Tools

For representation of the rules, visual prolog version 7.1 was selected and used. For training the neural network model, artificial neural network libraries were compiled and imported. In addition, Microsoft visual C++ 6.0 was used to compile the neural network libraries and create the library files that were imported for use in the visual prolog environment. The selection criteria of all of these software tools are their performance and convenience on the part of the researcher.

1.4.4 Prototype Development

The prototype system is developed using hybrid approach. By integrating the rule based approach with neural network, efforts have been made to exploit the strength of these two methods.

1.5 Scope and Limitation of the Study

The scope of this research is limited to developing a hybrid decision making model that uses a combination of rule based and machine learning technique in the production of vegetable crops. The research is bound to modeling decision making on fertilizer recommendation and disease control. It does not incorporate modeling decision making related to weeding, seeding and other aspects of vegetable crop management. Given certain parameters like soil type, PH, texture, altitude, previous cropping history and additional soil characteristics, it only provides expert advice on issues related to fertilizer application and disease control for vegetables.

The location of most of the agricultural research centers is too far from Addis Ababa. Since the allocated amount of research budget is very small, it was not possible to frequently visit and collect knowledge from researchers working at these centers. Thus, in addition to review of horticultural literatures, the domain knowledge was collected from experts working at few of these research centers, Ministry of Agriculture and Rural Development (MoARD), national soil laboratory and other international organizations.

1.6 Organization of the Thesis

The whole work of this research is compiled and presented in the rest of this document. Chapter two describes the scientific foundation and general principles in developing expert systems. It covers the different approaches of developing expert systems, the powerful features of neural networks, vegetable production in Ethiopia, and related researches which have been conducted in other domain areas. The basic issues related to fertilizer application and disease control on potato, onion and tomato are also presented in this chapter.

In chapter three, the knowledge acquisition process for this research is explained. This chapter mainly concentrates on showing how knowledge of experts and related concepts were acquired and modeled using decision trees. Chapter four is about knowledge representation. This chapter contains the discussion of the rules represented using prolog and the data feed to the artificial neural network. Chapter five presents the hybrid vegetable expert system model developed in this research. Chapter six is devoted to the discussion of implementation of the proposed model.

The major achievements gained in this research and the gaps that need to be filled by future work are presented in the seventh chapter. Here, important recommendations are proposed for future researchers for upgrading and or improving the performance of the system.

Chapter Two

Literature Review

2.1 Expert Systems

> *"Expert systems are computer programs which attempt to emulate the manner of problem solving and decision making process of a human expert with in a restricted context" [6].*

An expert system is a system which employs human expertise captured in a computer based information system to solve problems which usually require human expertise. An expert system either supports or automates decision making in an area of which experts perform better than non experts. It is also known as "Expert Computing Systems", or "Knowledge Based Systems" [7].

Knowledge-based expert systems, or simply expert systems, use human knowledge to solve problems that normally would require human intelligence. These expert systems represent the expertise knowledge as data or rules within the computer and these rules and data can be called upon when needed to solve the problems.

Experts spend a long time studying and practicing their skills and they do their jobs well. But the trouble with human experts is that they are scarce, they are not always reliable, they require payment, and in the long run, they die – taking much of their knowledge with them. Many people, therefore, are interested in the idea of encoding expert knowledge in computer programs [8].

2.1.1 History of Expert Systems

The Early Period

The 1950s and 1960s were the period when artificial intelligence was primarily concerned with the development of computer programs that could perform tasks that were considered to require a high degree of intelligence, e.g. games such as chess; theorem solving etc. A key development during this period was the idea of heuristics (guidelines for choosing among alternative actions) [7].

The Cognitive Period

This was the time period in which artificial intelligence was applied towards solving practical, real-world problems. The concept of the expert system arose in the 1970s when artificial intelligence (AI) researchers abandoned, or postponed the quest for generally intelligent machines and turned instead to the solution of narrowly focused real - world problems. Thus the expert systems in this period become one of the first examples of applied AI and spread out far beyond the confines of the research laboratories in which they were devised [8].

The Modern Period

Research in this period turned toward trying to clone human experts by capturing their experiential knowledge. These successes led to the idea of an expert system that had the basic structure in which rules could be entered, and the matching capability to make inferences based on the rules. The simplicity of this concept led to the rapid commercialization of expert systems. Thus, it was with the start of the early 1980s, that knowledge-based systems were applied to a wide variety of areas [7].

2.1.2 Approaches for Designing Expert Systems

Different approaches have been used for designing expert systems. In general, there are five different existing approaches for modeling expert systems [8].

Rule Based Approach

Expert system developed with this approach encodes its knowledge in the form of “If…Then “rules, alternatively referred to as production rules. A typical rule assumes the form “IF condition /premises, THEN, action/conclusion". The design of expert systems using rule based technique is particularly appealing for expert systems that involve dialogue between the user and the system.

Rules also have some disadvantages: Rules are brittle when presented with noisy data that contains unexpected values or incomplete data with missing values. Also data with nonlinear relationships may be difficult to express by rules. Rules do not scale up very well and inconsistencies and errors may occur. Furthermore, manual encoding of rules is time consuming and expensive .The main cause of brittleness within rule-based systems is the requirement that every possible combination of antecedents and their values must be explicitly provided. Otherwise the system will fail when presented with novel combinations of input data [8].

Although this approach is widely used, it fits well only in a domain where problems have a well defined model or structure. That is, the rule based approach is applicable for problems that are initially understood and easier to be formulated in to sets of rules.

For problems in complex domain areas, it is hard to model the significant part of the set of problems in advance since some of the problems may have never occurred before. For example, suppose various condition/ action rules pertaining to pest management of a particular vegetable is designed and implemented. Latter, a natural disaster such as flooding has occurred and the crop became water logged. As a result, new pest invasion may emerge whose controlling rule was not represented in the knowledge base. Since the rule based approach uses only the knowledgebase to perform an action by selecting the rule that is applicable to the particular situation, it is difficult to give a solution for such problems. This is because some problems require frequent updating of information and systematic association of patterns to make decisions about the problem.

Human experts use commonsense and creativity in decision making. They also tend to learn from previous experiences. However, rule based expert systems are not good at recognizing a solution when no answer exists or when the problem is outside their area of expertise.

The Frame Based Approach

This approach provides a more structured representation in the form of frames. A frame describes an object, consisting of slots containing default values, pointers to other frames, sets of rules, or procedures. Frames are linked to provide for inheritance and communicate by passing messages. However, the modularity of knowledge represented in frames cannot be defined clearly, and the representation lacks flexibility [8].

Object Oriented Approach

These use the fundamental concept of “object” which combines data structure and behavior in a single component. Each object is characterized by its states and actions, which are the attributes and values of the object respectively. Actions are then the operations that the object is capable of executing. Each object is also a member of a class. Libraries of class definitions serve as the basis for reusability of classes in object oriented approach.

Amongst the limitations of object oriented systems, the system may be slower in execution. Also, the message-passing mechanism in this approach does not include the provisions of how a message has to be handled by the receiving object from a sender object’s background and perspective. In addition, as human communication includes context and intention apart from content, merely sending a message content becomes a very restricted form of communication [8].

Logic Based Approach

In this method, expert systems use simple true /false logic to evaluate data. They are capable of performing at least some evaluation taking in to account real world uncertainties, using methods such as fuzzy logic which is an extension of Boolean logic dealing with concept of partial truth. They replace Boolean truth values with degrees of truth [8].

Hybrid Approach

This approach employ a decision analysis tool formed through the careful combination of an expert system and some other methodology [7].

2.1.3 The Architecture of Expert Systems

There are four components common to any expert system. This includes the knowledge base, the inference engine, the knowledge acquisition module and the explanatory interface. Yazdani[8] explained that the knowledge base and the inference engine are the most fundamental components in the design of expert systems. Figure 2.1 shows the architecture (general framework) of expert systems and the interaction between these components.

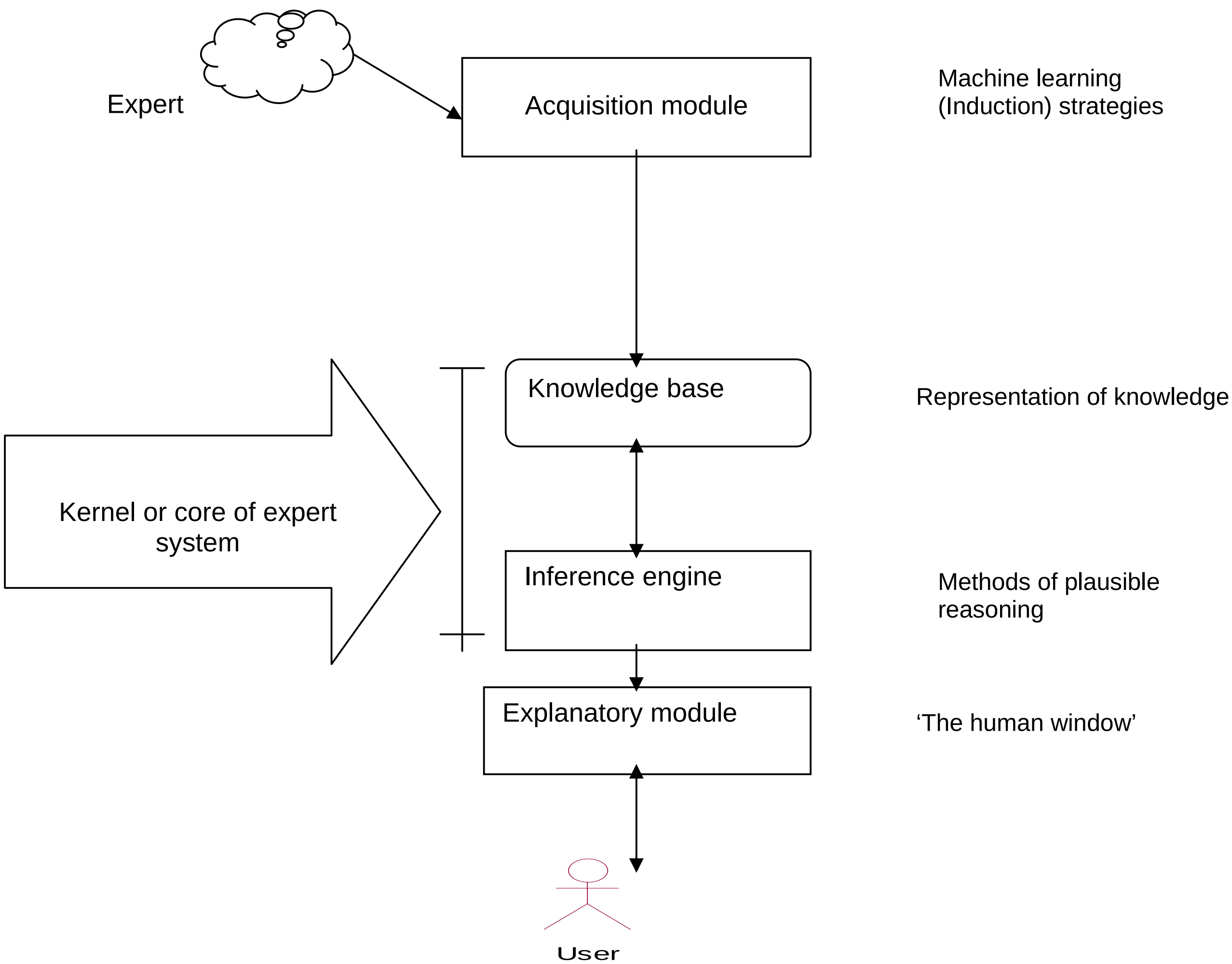

Fig 2.1 Architecture of expert systems

Knowledge Base

The two fundamental components of an expert system are the knowledge base and the inference engine. The knowledge base stores information about the subject domain; however, information in a knowledge base is not the passive collection of records and items that you find in a conventional database. Rather it contains symbolic representation of experts' rules of judgment and experience, in a form that enables the inference engine to perform logical deductions upon it [8].

Inference engine

Inference is the process of deploying evidence in order to arrive at new conclusions. An inference mechanism consists of search and reasoning methods that enable the system to find solutions and if necessary, provide justification for its answers. There are two fundamental reasoning strategies:

I. **Forward Chaining**

It is a reasoning strategy by working forward from the evidence (or symptoms) to the conclusions (Diagnosis). In a rule based system, it simply involves matching the IF conditions to the facts, possibly in a predetermined order.

II. Backward Chaining

This strategy works from hypothesis to evidence. The system picks a hypothesis and looks for data to support or refute it.

2.1.4 The Present Stage of Expert System Technology

From the very beginning, scientists and researchers of AI have been trying to produce systems that can behave like an intelligent being. In course of such developments, they realized the clue of human expertise in a certain field and tried to encode the knowledge and experience of human experts in computers that led to the idea of developing expert systems in areas of different domains [9].

The following are few of the remarkable achievements in the field of AI, particularly expert systems:

- April 17, 2007: **New system launched to help citizens self-serve**. To help local authorities reduce the burden of in-bound calls relating to application queries, Computer Assisted Problem Solving (CAPS) solutions is launching Public Access Expert System. The intelligent web-based system uses a flow-chart approach to advise citizens what course of action they should take, based upon their responses to a set of questions.
- March 23, 2007: '**Thinking' computer may copy doctors**. By Robin Turner. A "thinking computer" which can imitate doctors' decisions about treatment for patients is being developed by scientists at a Welsh university. The system will monitor patients' vital signs then evaluate and administer drugs, a job now done by specialist medics.
- October 14, 2006: **Detecting art fakes at a stroke**. "Spotting a forged painting usually takes an expert eye and hours of analysis". That could change with a computer program that analyses artwork for signs of an artist's unique style. The software, called **"Authentic",** help date paintings by a particular artist.
- July 2005: **AI in Control** - Artificial intelligence, expert systems, fuzzy logic, neural nets, and rules-based algorithms for factory control. The Intelligent Maintenance Systems (IMS) center has developed a toolbox of algorithms. Of particular interest is the Watchdog Agent. This agent, explains Lee, "can assess and predict" the process or equipment performance based on the inputs from the sensors mounted on it [9].

2.2 Artificial Neural Network

An artificial neural network is an information processing paradigm that is inspired by the way biological nervous system process information. It is composed of a large number of processing elements (neurons). Several neural network configurations are possible, such as feed forward, self-organizing and recurrent networks but they have common features. The most important is the use of weighted connections for information storage. The information learned by a neural network is held in the weights and biases as a result of exposure to the training set [10].

As indicated in Figure 2.2 below, a feed forward artificial neural network can have a number of input, hidden and one or more output neurons. The number of neurons in the hidden layer must be determined experimentally. Here is a diagram of an artificial neural network with three layers and ten neurons.

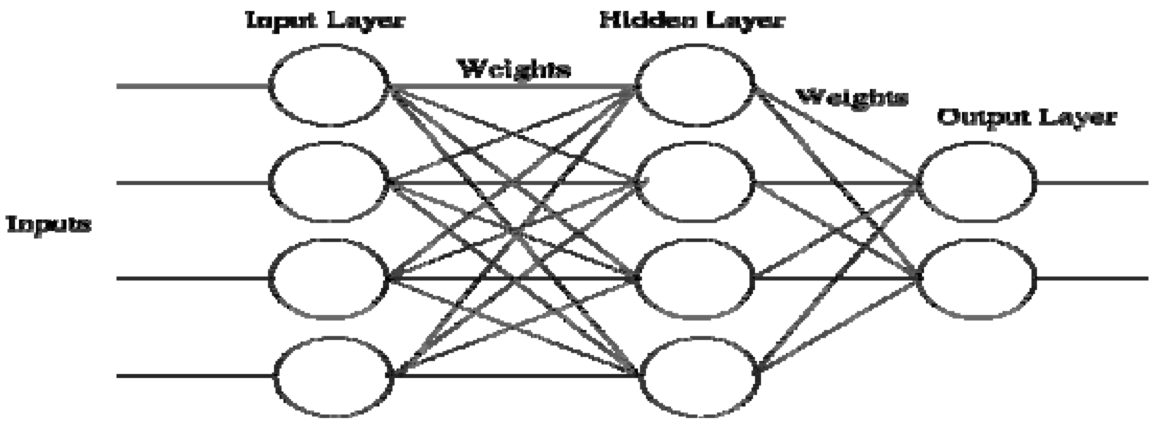

Fig 2.2 Multilayer Perceptron Neural Network

Input Layer —the input layer standardizes the input values and distributes them to each of the neurons in the hidden layer.

Hidden Layer — arriving at a neuron in the hidden layer, the value from each input neuron is multiplied by a weight, and the resulting weighted values are added together producing a combined value(After they are processed by activation function). The outputs from the hidden layer are distributed to the output layer.

Output Layer — arriving at a neuron in the output layer, the value from each hidden layer neuron is multiplied by a weight, and the resulting weighted values are added together producing a combined value. These values are the outputs of the network.

According to McGarry[10], Neural networks have the following features and advantages:

- Compact knowledge representation in the form of weight and threshold value matrices.
- Very fast and can operate with noisy or missing data and can generalize quite well to similar unseen data.
- They learn inductively from training data and their ability to process non-linear functionality is an important feature for managing real-world data.

However, neural networks also have their own limitations:

- They may need lengthy training times and may not necessarily converge to an acceptable solution.
- They have limited explanation facility which may prevent their use in certain applications.

2.3 Vegetable Production in Ethiopia

Ethiopia is well known for its diversity of indigenous food plants, including vegetables. The country is gifted with various agro climatic regions for growing an array of indigenous and exotic vegetables. Different types of leafy, root, bulb and fruit vegetables are grown in different parts of the country under rain fed and irrigated condition. The warm season vegetables such as tomato, onion and capsicum are grown in the lowland areas under irrigation. Whereas the highland areas offer favorable condition to grow cool season vegetables such as cabbage, garlic, shallot and carrot [11].

2.3.1 Importance of the Vegetable Sector

Vegetables are the major source of vitamins, minerals and proteins. They have significant and profound benefits for the people living in different parts of the country where people experience malnutrition due to dependence on cereals. In addition, these crops generate higher economic value than cereal crops. Hence, vegetables are used as income source of our farmers and contribute to the growth of the country's economy through providing diversified alternatives for export and local markets [12].

> *"The vegetable industry could benefit from tapping the potential of the indigenous species as this would improve local and national food security, augment farmers' incomes and help surmount some health problems associated with nutrient deficiency" [11].*

Vegetable crops are produced in large quantities and play an important role in the lives of many people in the Eastern part of Ethiopia. Until recently, vegetables were regarded as being second in importance to 'chat' as cash crops. Present trends, however, indicate that vegetables are becoming primary cash crops [13].

2.3.2 Production Agro Ecological Zone (AEZ)

There are five different agro ecological zones with adequate irrigation, water, favorable climatic conditions and high potential for the development of the sector.

Hot To Warm and Lowland Plains

Areas under this category cover 30 percent of the country which includes some parts of oromia, Afar and Gambella. These zones have low rainfall and various soil related constraints [12].

Tepid to Cool Semiarid and mid highlands

This agro ecological zone (AEZ) covers 28 percent of the country with low rainfall, dry weather and dominated by the rift valley lakes. Different warm season vegetables and irrigated vegetables are produced here [12].

Tepid to Cool Moist Midlands

Covers 11.2 percent and have high potential for biennial vegetables. Research centers established to serve these areas include Holeta, Alemaya and Sirinka [12].

Tepid to Cool Sub Humid Highlands

This AEZ cover 7.6 percent of the country .It encompasses mid highlands of southern Ethiopia and north harerge [12].

Tepid to cool sub moist mid highlands

In this area, there is a potential for biennial vegetable seed production. They are represented by Sheno, Mekelle and Debre Zeit. Covers 0.42 percent and includes Tigray, North east shoa and North wollo [12].

2.3.3 Major Vegetable Crops

Tomato

There is no definite time recorded regarding the introduction of cultivated tomato to Ethiopia. However, cherry type has been growing for long around big cities and in small gardens. Recently, the crop has expanded to commercial production for home use, export and processing industries.

The bulk of fresh market tomatoes are produced by small-scale farmer. Like in many other countries, it is also becoming important in Ethiopia in a variety of dishes. It is an important cash-generating crop to small-scale farmers and provides employment used as salad. It is also important source of vitamin A and C as well as minerals. Such diverse uses make the tomato an important vegetable in irrigated agriculture in the country [14].

In Ethiopia, altitudes between 700 and 2000 which is characterized as warm and dry day and cooler night are favorable for optimum growth and development of tomatoes. Experiences under humid or rainy growing conditions indicate that tomato is susceptible to disease. Tomato grows better at a constant day and night temperature. Fruit setting is poor when the temperature is either high or low. Extreme temperatures cause flower drops and poor fruit set. A temperature range between 21 and 27^0c day and 10 to 20^oc night is favorable for optimum production of tomato [14].

Chemical fertilizers, organic manure or compost provides nutrients for producing healthy and vigorous seedlings. The amount of fertilizer applied on seedbed depends on the fertility of the soil. Experience at Melkassa showed that on sandy loam soil, 200 g DAP at seeding and 100 g urea at thinning (at first true leaf stage) could be applied to enhance growth. Incorporating well decomposed manure is also good practice [14].

Onion

Onion is a cool season crop and adapts to a 13-24 degree centigrade temperature and tolerant to frost. It requires cool temperature during early stages of growth before bulbing where it requires 20-25 degree centigrade. Since onion roots are shallow, surface irrigation is essential for development of roots. Different varieties of red, brown, white and yellow types are produced for local and export markets. The red ones are currently the most preferred ones. The recently released cultivars develop flower stack and produce good quality seed in the rift valley region. Under field conditions, 150 kilogram per hectare (kg/ha) urea and 100 kg/ha dap will be applied for optimal growth of the crop [15].

Potato

Virtually the entire central highlands of Ethiopia have been classified as "most favorable" for potato cultivation, while a broad arc to the west and south of this area is deemed "favorable". Taken together, these areas include the entire administrative regions (formerly provinces) of Gondar (Begemendir), Gojam, Shewa, Wellega, Ilubabor, Kefa, Arsi, Gamogafa, and Sidamo, as well as large areas of Tigray, Wello, Hararge, and Bale [15].

2.4 Related research works

So far, a number of research works have been conducted in the field of expert systems. Despite the output of the researches have their limitations, important achievements have been obtained and produced significant contribution to the present stage of expert systems.

Thomas A. O'Callaghan [16] developed a hybrid legal expert system called SHYSTER-MYCIN that combines two existing expert systems (SHYSTER and MYCIN). He used portion of the Australian copy write law for experimentation in developing the hybrid system. SHYSTER is a case-based legal expert system while MYCIN is a rule-based reasoning medical expert system.

He tried to modify the MYCIN to report not just the conclusions, but also the reasons for reaching to the conclusions (that is, he provided an explanation facility in the system). Finally, he concluded that the approach taken in constructing the system is appropriate. That is, it is appropriate to use rule-based reasoning when dealing with statutes, and that it is appropriate to use case-based reasoning when dealing with cases.

In this system, it is observed that there is a need for creating rule management system for handling translation of the laws in to rules, since a great number of rules are required to represent only a small number of provisions in the law. If the whole articles related to the copy write law is to be represented in the system, the creation of this rule management system is desirable with which the rule base is weak. That is, when changing it, there is a greater likelihood for occurrence of error.

V. Van Hoof et al. [17] and Thomas Siessmeier et al. [18] conducted related researches on medical expert systems. Much of their work concentrates on applying the rule based techniques for representation of expert knowledge in these systems. They used expert system shells for their development.

Ermias[19] and Rediet[20], used the rule based approach to develop prototype expert systems. Both used knowledge pro version 3.0 expert system shell for representing rules. Ermias conducted the experiment for social security determination in Ethiopia, by taking sample records of civil servant workers. Given the input parameters such as experience, retirement reason and family condition, Ermias claims the system can determine the eligibility status of the applicant as to whether it is retirement pension, survivor's pension and so on.

Although he reported that encouraging result has been obtained, some limitations can be observed in his work. In the first place, since there are other complicated issues such as determination of eligibility for retirement and pension amount, it is not clear why he selected determination of retirement type as the major problem and as a problem which solution demands expert advice. Secondly, he had placed experience in the root of the decision tree giving more weight for experience. It would have been better to put age at the root of the decision tree since age has to be checked before any other parameters. In addition, the system was not designed to aid calculation of the pension amount.

Rediet's work was to develop a prototype expert system for HIV pretest counseling. The prototype system is developed with an aim to create an environment of freedom for users who seek pretest counseling service by avoiding third part intervention in the process (maintaining individual's privacy during the counseling process). By forwarding some determinant questions to the user and collecting important information, she claimed that the system can decide the level of risk and vulnerability to HIV thereby giving the relevant counseling service for the user. The major limitation that is observed in her work is the construction of the rules based on limited number of facts. That is the input parameters asked from the user are very few and the conclusion seems to be based on shallow premises.

Apart from their limitation to incorporate sufficient data in the rule base, both systems developed by Ermias and Rediet are unable to deal with unseen circumstances (new cases) which restricts its capacity to only deal with known facts and circumstances.

Seifu[21] used hybrid approach to develop legal expert system on criminal cases. He divided a given problem in two parts: one that can be solved by the rule based part and the other that can be solved by the neural network. He used the rule based part to represent rules of codes pertaining to sexual crime and associated Article Number with which the offender is convicted. On the other hand, he used the neural network to predict punishment in terms of the number of years of imprisonment the offender is expected to face. NEUROSOLUTIONS software had been used to train the neural network model in his prototype legal expert system.

Three major weaknesses can be observed in seifu's work. The first one is its inadequacy (very limited size) of data fed to train the neural network. Because of this, it is reported that the output of the network has a pronounced variation between result of the system's sentencing prediction and judge's actual sentencing decision.

The second problem is in relation to integration of the rule based and neural network systems modules. The two modules work separately instead of allowing user to access them in one interface. Besides, he didn't utilize the power of neural networks to handle sentencing decisions that are beyond the capacity of the rule based module. Since the neural network takes input parameters from the result of the rule based part, it will not be able to provide a solution to a problem that has no solution in the rule based module (it waits for part of a solution from the rule based module). If the rule based module fails, the whole system will fail to provide a solution to a given problem. Seifu's work would have been better if the system was designed to deal with cases that are not solved in the rule based part by utilizing the power of neural networks.

The second problem is attributed to inability of his neural network model to convert sentencing prediction into fees when the offender is allowed to compensate imprison punishments with money. He didn't consider the option of converting some kind of prison punishment in to punishment with money.

Although encouraging achievements have been obtained from these researches, there are still research worthy issues with respect to extending their capacity, performance and reliability of these systems.

Chapter Three

Knowledge Acquisition and Concept Modeling

3.1Introduction

The rules in the knowledge base are representations of expert reasoning strategies used to solve practical problems in the field. One of the primary tasks of a knowledge engineer in the development of an expert system is capturing the expert's knowledge for representation in the rule base. This knowledge capturing process can be performed in three different ways:

1) By interviewing individual experts

2) Through Workshops focusing on the problem domain and

3) Review of related documents.

Systems developed through this process must be able to liaise with people in order to gain knowledge and the people must be acknowledged domain experts as doctors, researchers, lawyers or investment analysts. Having acquired the information, the knowledge engineer is responsible for the self consistency of the data, and a number of specific tests have to be performed to ensure that the conclusions reached are sensible [16].

In this research, the knowledge acquisition process was performed through interviewing experts and review of related documents.

3.2Knowledge Acquisition

3.2.1 Knowledge Acquisition from Domain Experts

Human experts usually apply their knowledge and experience to solve problems in their workplace. They also use heuristic and subjective judgment in making decisions related to the particular problem. Since expert systems should embed the knowledge of real experts in their knowledge base to perform as human experts, putting all the important knowledge of experts into a form that can be easily used by a computer is a challenging task. The reason for this is attributed to human behavior; some knowledge is tacit (are inherent to the individual and difficult to explain), some people may not be interested to explain all what they know, and some people have a different saying of the same thing. All these and other factors may affect the knowledge acquisition process.

The knowledge acquisition process for this research was done taking into consideration of the above mentioned challenges; due attention was given to most important concepts, avoiding ambiguities and minimizing their effect on the overall acquisition process. In this regard, the following techniques were applied:

- **Interview and discussion with experts**: I have conducted in-depth interviews with senior experts by asking questions about the methods of fertilizer application, the problems they usually encounter, their experience in solving these problems, the kind of advices usually requested by farmers, the different vegetable disease and controlling mechanisms. See Annex I of this document for the sample questions.
- I observed that some experts are not good at communication and unable to clearly explain what they know. This will make the knowledge acquisition process incomplete. To fill this gap, experts were given white sheet of papers to write questions (problems) usually asked by farmers, development agents (DAs) related to their understanding about vegetable crop management decisions, fertilizer application and disease control. This has given the chance to experts who are not good at oral communication to articulate and explain what they know and experienced before. See Annex II of this document for the information collected with this method.

3.2.2 Knowledge Acquisition from Relevant Documents

In order to strengthen the knowledge acquired through the above methods, production manuals, research journals, leaflets (brochures) and relevant vegetable books have been carefully reviewed. These documents were obtained from various sources. I used the library of ministry of agriculture and rural development (MoARD), Ethiopian Institute of Agricultural Research (EIAR), formerly called EARO (Ethiopian agricultural research organization), website of EIAR (to download publications by Ethiopian researchers on different vegetables), and the documentation center of the research institutes.

3.3 Modeling the Concepts

Conceptual model provides an implementation-independent specification of knowledge in an application domain. Here, the basic concepts that reflect the major activities and decisions that are made to solve problems in the domain are modeled. The next sections illustrate the conceptual models for fertilizer application and disease control in vegetable crops.

3.3.1 Fertilizer Application for Vegetables

Determining the type and amount of fertilizer required for vegetables is one of the toughest tasks that need careful expert decision. First, the type of fertilizer needed must be decided and then the exact amount of fertilizer recommended must be computed. Applying fertilizer without careful expert recommendation can lead to irreversible disastrous consequences on the crop [22].

Vegetables are high nutrient consumers. The type and amount of fertilizer to be applied can be determined by soil/plant tissue analysis. Farmers could prepare compost or get well decomposed manure, which contribute to increased soil nutrient and water holding capacity of the soil and subsequently increased yield.

The following steps must be followed to determine the right type of fertilizer required for the given vegetable:

- Determine the level of Nitrogen, phosphorous, potassium (NPK) required for optimum growth of that particular vegetable.
- Determine the amount of NPK currently available in the soil.

- Determine the type of nutrient which is deficient in the soil.

Once the nutrient deficiency of the soil is known, it is easier to determine the type of fertilizer required to fill the gap between what is available and what is normally required. In addition, the following information is equally important in the whole process of fertilizer determination.

  - The environmental conditions in which the vegetable under consideration is grown (altitude, rainfall, soil color, texture, farming practice, previous crop, soil pH…).
  - Growth stage of the vegetable crop (Germination stage, seedling stage, maturity stages).
  - Method of sowing (lining, broad casting).
  - Method of application (side dressing, broad casting).

Despite there are many types of fertilizers available today, we can generally categorize fertilizers under two major groups as organic (Natural) and inorganic (chemical) fertilizers. Sources of Organic fertilizers can be compost, manure (cow dung) and animal and plant residues. On the other hand, the major chemical fertilizers commonly used in Ethiopia include DAP (to supply phosphorous requirement) and Urea (to supply Nitrogen requirement) [22].

For this research, only the concepts in relation to chemical fertilizers are modeled. There are a lot of factors that must be taken into consideration in the course of deciding the correct type of fertilizer for each vegetable. During the interview with vegetable experts, they pointed out that there is no as such significant difference between the varieties as far as their fertilizer requirement is concerned. Thus, for this research, “Variety” is not taken as a parameter to determine the fertilizer requirement of the given vegetable. Variety is to mean the different cultivars with in each vegetable category (for example, Adama Red is one variety of onion).

Table 3.1, 3.2, 3.3 summarize the relationship between the conditions and corresponding actions taken to decide the fertilizer recommendation to onion, tomato and onion vegetable crops respectively. Since there was limited access to obtain such data already prepared in this format, much of the time for this research was invested in collecting and organizing the knowledge from the experts. In all the cases, calculating the correct amount is done after deciding what type of fertilizer is required and the amount depends on time of application.

Fertilizer Application for Onion

Fertilizer requirement for onion depends on the fertility of the soil and the amount of organic matter present in the soil. When there are deficiencies, specific micronutrient should be applied after conducting soil analysis. Current nutrient availability depends on a lot of factors such as soil color, texture, temperature, pH, etc. Table 3.1 is presented to summarize these factors in determining fertilizer requirement of onion. Generally, when it is evident that there is a deficiency of nitrogen and phosphorous elements, applying 150 kg per hectare DAP (during planting) and 75 kg/ha of urea (after planting) is required [22].

Condition											**Action**
Previous crop	Soil color	Soil texture	Fixation	Temperature	Previous fertilizer	Time of application	Method of application	pH	Frequency of ploughing	Altitude	**Recommendation**
Legume	Red	sandy	high	Low	-	seedling	Side dress	Acidic	>=3	high	DAP
Legume	Black	clay	low	Medium	Urea	sowing	Broadcast	Neutral	<3	medium	DAP
Non-Legume	brown	loam	low	low	Dap	sowing	Broadcast	Alkaline	<3	High	Urea
Legume	Black	clay	low	Medium	Urea	sowing	Broadcast	Neutral	<3	low	NOT REQUIRED
Non-Legume	brown	loam	low	high	Dap	seedling	Broadcast	Alkaline	>=3	High	Urea
Non-Legume	red	sandy	low	medium	-	seedling	broadcast	acidic	<3	medium	BOTH

Table 3.1 Condition/ action rules for fertilizer recommendation of onion

In addition to providing the information, experts were also participated in the final organization and summarization of the collected knowledge with the form presented in this table. In this and all the remaining tables, the sample condition/actions are presented taking smaller portion of the knowledge collected.

Fertilizer application for tomato

Chemical fertilizers, organic manure or compost provides nutrients for producing healthy and vigorous seedlings. Experience at Melkassa showed that on sandy loam soil, 200 g DAP at seeding and 100 g urea at thinning (at first true leaf stage) could be applied to enhance growth. Besides, Incorporating well decomposed manure is also good practice [22]. Table 3.2 shows the various conditions where DAP, Urea and both fertilizers are applied to tomato crop.

Condition											Action
Previous crop	Soil color	Soil texture	Fixation	Temperature	Previous	Vegetable growth	Method of applicati	pH	Frequency of	Altitude	**Recommendation**
Legume	black	clay	high	Low	-	seedling	Side dress	Acidic	>=3	high	Both
Non-Legume	red	sandy	low	Medium	Urea	sowing	Broadcast	Neutral	>=3	medium	Urea
Legume	Black	loam	low	Low	Dap	sowing	Broadcast	Alkaline	<3	High	Not Required
Legume	brown	clay	low	Medium	Urea	sowing	Broadcast	Neutral	>=3	medium	BOTH
Non-Legume	red	loam	low	Low	Dap	sowing	Broadcast	Alkaline	<3	High	Urea
Non-Legume	red	sandy	low	Medium	-	seedling	broadcast	acidic	<3	medium	DAP
Legume	Black	sandy	low	Medium	-	seedling	broadcast	acidic	<3	medium	BOTH
Non-Legume	black	loam	low	High	DAP	sowing	broadcast	acidic	<3	medium	Urea

Table 3.2 Condition /action rules for fertilizer recommendation of tomato

Fertilizer application for potato

Similar to the fertilizer need of tomato and onion, optimum production of potato depends on sufficient availability of nitrogen, phosphorous and potassium in the soil. Depending on the characteristics of the soil, the amount of fertilizer applied to potato amounts to 65- 70 kg/hectare Urea and 195 Kg/hectare DAP [15]. Side application of DAP during planting is preferred as the best method of applying the fertilizer. Before applying any of these fertilizers, the environmental and soil conditions have to

be seen as presented in Table 3.3.

Condition											Action
Previous crop	Soil color	Soil texture	Fixation	Temperature	Previous fertilizer	Vegetable growth stage	Method of application	pH	Frequency of plaguing	Altitude	Recommendation
Legume	brown	sandy	low	Low	-	sowing	broadcast	Acidic	>=3	low	Urea
Legume	Black	clay	high	Medium	Urea	sowing	Broadcast	Neutral	>=3	medium	Urea
Non-Legume	brown	loam	Low	low	Dap	seedling	Broadcast	Alkaline	<3	High	Both
Non-Legume	brown	loam	High	low	Dap	seeding	Broadcast	acidic	<3	Low	DAP
Legume	black	Sandy	High	low	Urea	Sowing	Side dress	Alkaline	<3	Medium	Not required
Legume	Red	clay	low	high	-	seedling	Side dress	Acidic	>=3	low	DAP

Table 3.3 Condition/ action rules for fertilizer recommendation of potato

Once the set of condition/ actions are identified in the current decision strategy of experts, the next task is to generate the hierarchy of the concepts and model these concepts using appropriate modeling technique. I decided to use decision trees since this facilitates rule extraction and to accordingly arrange the order of questions that will be asked to the user in defining the rules in the rule based module.

According to J.R. Quinlan [24], a decision tree is a simple structure where non-terminal nodes represent tests on one or more attributes and terminal nodes reflect decision outcomes.

Decision trees are also used to generate the rules by following the path from the root to the leaves. One complete path from the root to the leaf gives one rule where the leaf nodes represent the outcome of the decision.

Based on the condition /action data presented in preceding section, I used the J48 decision tree algorithm to generate the decision trees presented in Fig 3.1, Fig 3.2 and Fig 3.3.

Decision tree for fertilizer recommendation of onion

There are a number of factors that need to be considered in deciding the type of fertilizer for production of onion. Soil color takes the highest priority as compared to other parameters. If the soil color is brown, the type of fertilizer previously applied to the soil determines the current application. Again, if no fertilizer was previously applied and the land was sowed with non legume crop in the past season, both DAP and Urea are recommended for this vegetable. The overall hierarchy of the concepts in deciding fertilizer type for onion is presented in Fig 3.1 of this document.

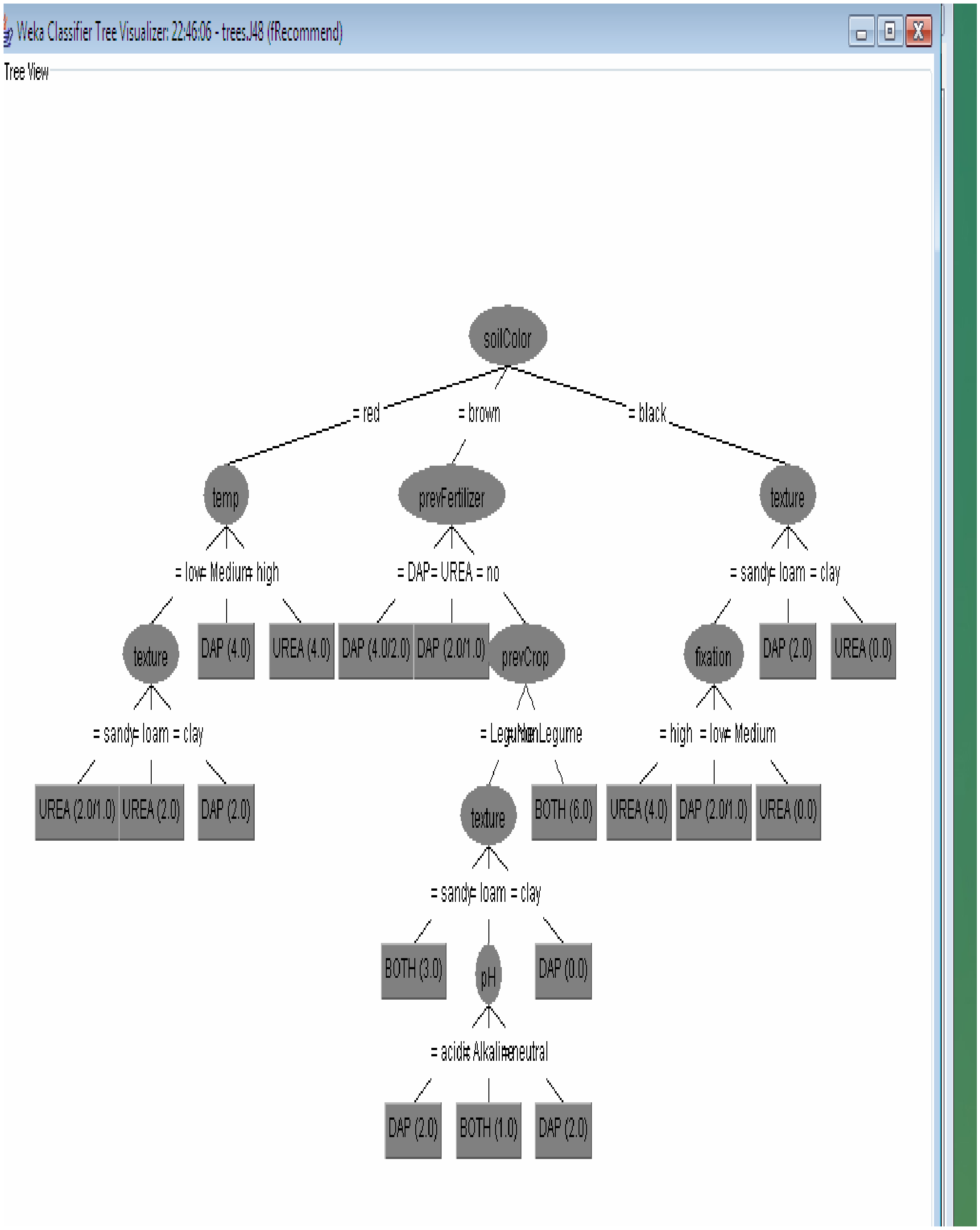

Fig 3.1 Decision tree for fertilizer recommendation of onion

Decision tree for fertilizer recommendation of tomato

When the soil pH is low, we should see the previous fertilizer applied on the soil. If the fertilizer was Urea, we recommend to apply DAP to decrease the acidity (provided that the temperature is medium and the soil color is brown). However, when the soil pH is high, again we should also look at the number of times the land has been ploughed. If the land was ploughed more than three times, DAP will be recommended on black soils. Fig 3.2 shows the decision tree for fertilizer recommendation of tomato.

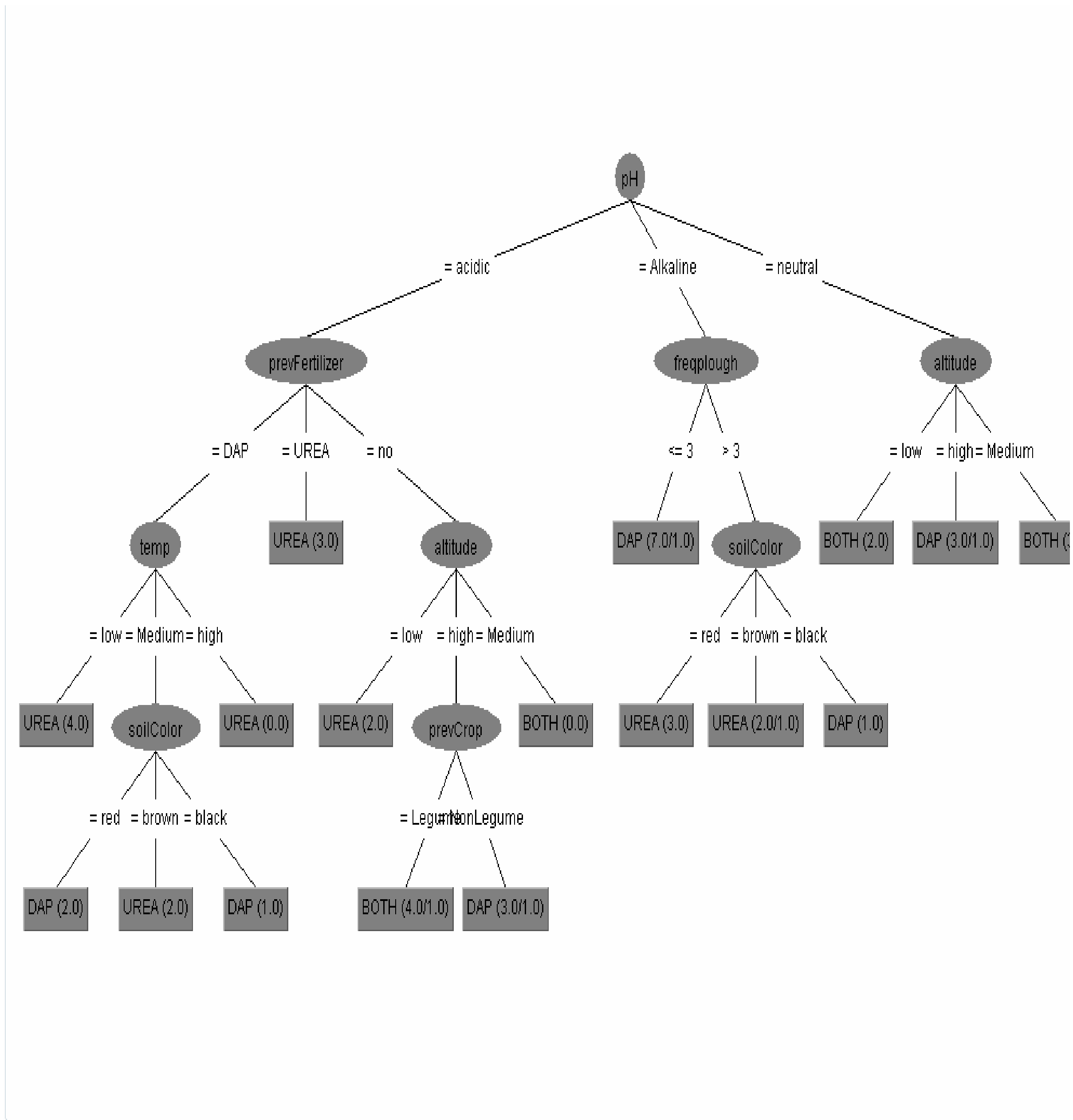

Fig 3.2 Decision tree for fertilizer recommendation of tomato

The decision tree in Fig 3.2 above shows how soil pH can affect the type of fertilizer to be applied to tomato. It is a measure of how acidic or alkaline a soil is. This hierarchy indicates that pH should be considered before any other parameter; A soil with low pH (below 7.0) is acidic, while a soil with high pH (above 7.0) is alkaline (basic) and pH =7 indicate neutral.

When the soil pH is low, we should see the previous fertilizer applied on the soil. If the fertilizer was Urea, we recommend to apply DAP to decrease the acidity (provided that the temperature is medium and the soil color is brown). However, when the soil pH is high, we should have to again look at the number of times the land has been ploughed. If the land was ploughed more than three times, DAP will be recommended on black soils.

Decision tree for fertilizer recommendation of potato

Similar to what is shown to onions, fertilizer requirement of potatoes is highly dependent on the color of the soil to which the potato plant is to be grown. The main difference lies in their dependence on attitude and temperature when the soil color is red and black. For potatoes, if the soil color is red, altitude has to be checked before looking at the type of fertilizer applied in the past season. However, fertilizer recommendation for onions is highly influenced with temperature than with altitudinal range with which the crop is to be grown. Refer figure 3.3 to see the diagrammatical representation of this decision flow in a tree.

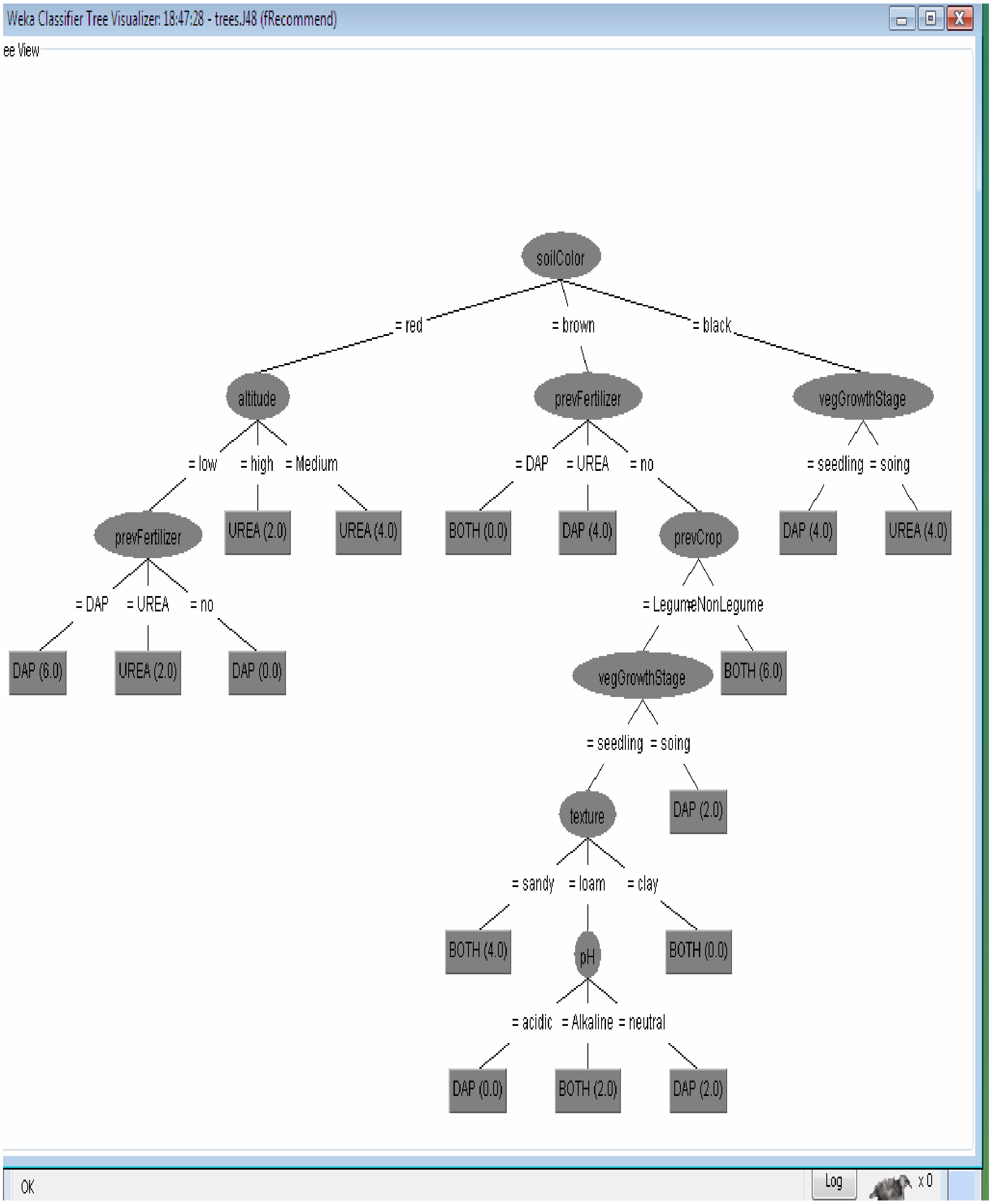

Fig 3.3 Decision tree for fertilizer recommendation of potato

3.3.2 Vegetable Disease and their control measures

Disease is one of the major constraints affecting vegetable plants at different growth stages and at post harvest. It can reduce yield and cause complete loss of the crop in the field.

Temperature and moisture especially high relative humidity, less sunshine, high night temperature increase incidence of disease.

Most small-scale farmers and DAs are not in a position to recognize disease and insect pest damage. This results in misusing of agricultural pesticides, which leads to hazardous effect to the farmer, consumer and to the environment [23].

For this research, disease symptoms and control measures have been modeled. With the same method used for modeling the concepts related to fertilizer recommendation. The next sections are used to present the explanations in relation to major vegetable disease and control measures.

Onion disease

Prevalence of disease Infestation is the greatest challenge for production of onions these days. According to Mohamed [23], the most common diseases of onion include purple blotch, mildew onion smut, white rot and stemphylium blight.

Purple blotch

Hot and humid climate with temperature ranging from 21-30°C and relative humidity (80-90%) favor the development of the disease. The symptoms occur on leaves and flower stalks as small, sunken, whitish flecks with purple colored centers. Spraying Mancozeb or Chlorothalonil or Iprodione after one month from transplanting at fortnightly interval reduces the disease incidence.

Downy mildew

The disease is worst in damp conditions and late planting of the crop, application of higher doses of fertilizers and numerous irrigation increased disease severity. Symptoms appear on the surface of leaves or flower stalk as violet growth of fungus, which later becomes pale greenish yellow and finally the leaves or seed stalks collapse. Spraying with Zineb , Karathane or Tridemorph gives good control of the disease .

Onion smut

The disease occurs in areas where temperature remains below 30°C. Since the fungus remains in soil, disease appears on the cotyledon of the young plant soon after it emerges. Smut appears as elongated dark, slightly thickened areas near the base of seedlings. The black lesions appear near the base of the scales on planting. The affected leaves bend downwards abnormally. On older plants, numerous raised blisters occur near the base of the leaves. The lesions on plant at all stages often expose a black powdery mass of spores. Seed bed treatment with Methyl Bromide (1 kg/25 m .) is effective in controlling the disease.

stemphylium blight.

This disease is very common on onion leaves and flower stalks. Infection occur on radial leaves of transplanted seedlings at 3-4 leaf stage during late March and early April. The symptoms appear as small yellowish to orange flecks or streaks in the middle of the leaves, which soon develop into elongated spindle shaped spots surrounded by pinkish margin. Spraying Mancozeb along with Monocrotophos sticker triton on appearance of disease controls the disease [23].

Tomato disease

The most common diseases in tomato production fields are damping off, late blight (Phytophtra infestant), early blight (Alternaria solani), powdery mildew (Leveillula taurica) and viruses.

According to Lemma [14] the following diseases, symptoms and their control measures are identified through several years of research conducted in the field.

Damping off:

It is the most common disease of tomato. It can be caused by various types of fungi like phythophtora sp and pythium sp. This disease attacks most vegetable crops at seeding stage. It is more pronounced at the early growth stage of the crop. It can be categorized as pre emergence (seed decay) and post emergence (seedling infection). Pre-emergence damping off affects seedlings during the early stage of germination but before emergence. In the emergence phase, the disease begins as dark colored, water soaked lesion on the root, which extends up the stem or above the soil level. Occurrence of damping off can be prevented by taking some preventive measures at the field level. Use of raised seed bed, watering using water can instead of flooding, burning of maize/ sorghum stacks on the bed before sowing can be used as important control measures .

Bacterial wilt

It is a serious disease caused by Ralstonia solanacearum bacteria. The bacteria multiply rapidly inside the water-conducting tissue of the plant, filling it with slime. This results in a rapid wilt of the plant, while the leaves stay green. If an infected stem is cut crosswise, it will look brown and tiny drops of yellowish ooze may be visible. To prevent the disease, remove all infected plant material and use only certified disease free plants.

Late blight

It is a highly destructive disease attacking most solanaceous crops like tomato. The pathogen attack all above ground parts of the tomato plant. Leaf lesion first appears at the edge of the leaf as water soaked spots which enlarge rapidly into brown and cover large areas of the leaf [14].

Potato disease

Potato is highly susceptible to disease pathogens. The major potato disease identified in most of the varieties includes late blight, bacterial wilt and viruses.

Late blight

Water soaked lesions on leaves and brown leaf spots signals the presence of the disease on the plant. The pathogen is at its most virulent in areas with cool, damp climates or where the soil has become over-irrigated. Disease-resistant varieties should be used when possible and farmers should keep abreast of news of outbreaks to select varieties and treatment.

Bacterial wilt

This is disease is caused by bacteria that multiply itself in the soil. When the plant is affected by this disease, it severely wilts characterized by vascular browning in stems and cut tubers. Infected tubers show soil crust on the eyes. To control the disease, rogue out infected plants along with the soil on the roots using plastic bags and bury in a pit. To control the disease, surface sterilize implements used in rouging diseased plants using either 70 % alcohol, 20 % bleach or burn on flame.

Viruses

The occurrence of viruses on potato is indicated by yellowish leaves and stunting growth. Leaf crinkling and mosaic appearance of diseased plants are also indicators of this disease. There is no chemical control available for these viruses. Using disease resistant varieties are recommended as good management practice [23].

Chapter Four

Knowledge Representation

4.1 Introduction

In the previous chapter, the discussion of how the knowledge of domain experts was captured for this work and the methods used to model the concepts had been presented. The different conditions and corresponding actions taken by experts were used to model knowledge hierarchy using decision trees. These models are not the final goal of this research, they are just concept models. The acquired knowledge has to be represented using suitable method for use by the inference engine of the development tool. Since I used prolog as the development tool, I decided to use Horn clause logic as a suitable method of representing the facts and rules which have been collected during knowledge acquisition. The next sections discuss the rules and data represented in the rule base and artificial neural network.

4.2 Representation of the knowledge in the rule base

The problem solving methods and reasoning strategies collected from vegetable experts have been converted into rules that can be activated to provide a solution to a given problem. The rules are generated from the concepts modeled in the decision tree and the condition / actions discussed in the previous chapter. Despite it is possible to generate a number of rules from the decision trees presented; only few of them selected from each vegetable will be discussed in this section. The rules are later represented by encoding the concepts according to the visual prolog language conventions.

Based on the environmental conditions and characteristics of the soil, the recommendation may be “DAP”,” UREA”, “Both” or “No need to apply fertilizer” on the land. The following rules are extracted from the decision tree generated in Fig 4.1, 4.2, and 4.3 of the previous chapter.

Rules for applying fertilizer to onion

Rule1:

If the soil is color is brown and

If no fertilizer was applied previously and

If the land was sowed with legume crop and

If the soil has loam textural property and

If the soil is acidic

Then

The fertilizer recommended is “DAP”

This same rule can be represented in visual prolog like this:

```
OnionDAP ():-

color(brown),

previousFertilizer(no),previousCrop(legume),texture(loam),pH(acidic)

.
```

Rule2

If the soil color is black and

If texture is sandy and

If fixation rate is high

Then

The fertilizer recommended is “Urea”

Prolog representation:

```
onionUrea():- color(black), texutre(sandy), fixation(high)
```

Rule3

If the soil color is brown and

If no fertilizer was applied previously and

If the land had sowed legume previously and

If the soil has alkaline loam textural property

Then

Both DAP and Urea should be applied on this soil

Prolog representation looks like this:

```
onionBoth():-color(brown),previousFertilizer(no),
previousCrop(Legume), texutre(loam), pH(alkaline).
```

Rules for recommending fertilizer for tomato

Rule4

If the soil is acidic with DAP fertilizer applied previously and

If the temperature is medium and

If the soil is red and

Then

DAP should be applied to this soil

Prolog representation:

```
tomatoDAP():-pH(acidic),previousFertilizer(DAP),
temperature(medium),color(red).
```

Rule5

If the soil is alkaline and

If the land is ploughed more than three times and

If the soil is brown

Then The fertilizer applied should be Urea

Prolog representation:

```
tomatoUrea():- pH(alkaline), freqPlough(more), color(brown).
```

Rule 6

If the soil is acidic and

If no fertilizer is applied previously and

If the altitude is high and

If the previous crop is Legume

Then

BOTH Dap and Urea should be applied to this soil

Prolog implementation:

```
tomatoBoth():-pH(acidic),     previousFertilizer(no),altitude(high),
prevCrop(Legume).
```

Rules for fertilizer recommendation of potato

Rule 7

```
If the soil is brown in color and
If no fertilizer is applied previously and
If the previous crop is Legume and
If the time of application is sowing
```

Then

```
The recommended fertilizer is DAP
```

Rule 8

```
If the soil color is red and
If altitude is low and
If no fertilizer is applied previously
```

Then

```
The recommended fertilizer is DAP
```

Rules for disease control module were generated by applying the same technique. To avoid redundant presentation of similar techniques used in this research, the researcher excluded this part and decided to report only the result obtained.

4.3 Representing Knowledge for the artificial neural network module

The rules in the previous section present only few of the possible rules that can be extracted from the decision trees. It is tedious to list all the possible rules extracted from these trees. The other problem of such method of representation is their weaknesses to handle new situations that are not explicitly defined in the rules. To improve the performance of the rule based techniques, the researcher tried to integrate the module that handles these rules with a neural network module trained with the data values initially used to build the decision trees.

4.3.1 Preprocessing the data for the neural network

Attribute reduction

Not all the attributes have the same influence in determining the type and amount of fertilizer to be applied. Some of the attributes are rarely required as for example, the method of application (either broadcast of side dress), others can be later asked after fertilizer determination, and some others are interrelated like temperature and altitude. Based on this, some attributes are eliminated from being used as input parameters in the neural network. After eliminating the less relevant attributes, the attributes “Previous Crop”, “Soil Color”, “Soil Texture”, “Soil pH”, “Fixation”, “Temperature” and “Frequency of plaughing” are selected to be used as input parameters.

If an attribute has ‘n’ possible values, it will have ‘n’ neurons to represent the values in the artificial neural network. Thus, each of the attributes selected are represented with the number of neurons equal to their possible values.

PreviousCrop(two neurons)

soilColor(three neurons)

soilTexture(three neurons)

pH(three neurons)

fixation(two neurons)

temperature(three neurons)

frequencyOfploughing(two neurons)

Totally, we have 18 neurons in the input layer

The output is the kind of fertilizer to be recommended. The “Recommendation column” contains four possible values. The value “Not required” is excluded from consideration because the observation shows that it rarely happens in the decision tree. Hence the number of output neurons are decided to be three.

Preparing the data for the ANN

After selecting the attributes, their values are tagged in to 1's and 0's. Table 4.1 is presented to demonstrate the preprocessing of the data for training the network model for fertilizer recommendation of onion.

Previous Crop		Soil Color			Soil Texture			Soil PH			Fixation		Temperature			Plaughing Frequency		Recommendation		
Legume)	Non Legume	Brown	Red	Black	sandy	loam	clay	acidic	neutral	alkaline	high	low	low	medium	high	<3	>=3	DAP	Urea	Both
1	0	0	1	0	1	0	0	1	0	0	1	0	1	0	0	0	1	0	0	1
1	0	1	0	0	0	1	0	0	1	0	0	1	0	1	0	1	0	1	0	0
0	1	1	0	0	0	0	1	0	0	1	0	1	0	0	1	0	1	1	0	0
1	0	0	1	0	1	0	0	0	1	0	1	0	0	1	0	1	0	1	0	0
1	0	0	1	0	0	0	1	1	0	0	1	0	0	1	0	0	1	0	1	0
0	1	0	0	1	0	1	0	1	0	0	0	1	1	0	0	0	1	0	0	1
0	1	1	0	0	1	0	0	0	0	1	1	0	0	1	0	1	0	1	0	0
1	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	1	0	1	0	0
0	1	0	1	0	1	0	0	1	0	0	0	1	0	1	0	0	1	0	1	0
1	0	1	0	0	0	1	0	0	0	1	1	0	1	0	0	1	0	1	0	0
0	1	0	1	0	1	0	0	0	1	0	1	0	0	1	0	0	1	1	0	0
0	1	0	1	0	0	1	0	1	0	0	1	0	0	1	0	1	0	1	0	0
0	1	1	0	0	0	1	0	0	0	1	0	1	0	0	1	0	1	0	0	1
1	0	0	1	0	1	0	0	0	0	1	1	0	0	0	1	0	1	1	0	0
0	1	0	1	0	0	0	1	1	0	0	0	1	0	1	0	1	0	0	0	1
0	1	1	0	0	0	1	0	0	0	1	0	1	1	0	0	0	1	0	1	0
1	0	0	1	0	0	0	1	0	0	1	1	0	0	0	1	1	0	1	0	0
0	1	1	0	0	1	0	0	0	1	0	0	1	0	1	0	0	1	0	0	1
1	0	0	1	0	0	0	1	0	1	0	1	0	1	0	0	1	0	1	0	0
0	1	1	0	0	1	0	0	1	0	0	1	0	0	1	0	0	1	1	0	0
1	0	1	0	0	0	1	0	0	0	1	0	1	0	1	0	0	1	0	0	1
0	1	1	0	0	1	0	0	1	0	0	1	0	1	0	0	1	0	1	0	0
1	0	0	0	1	1	0	0	0	1	0	1	0	0	1	0	0	1	0	1	0

Table 4.1 Preprocessed data of onion using Microsoft Excel spread sheet

Once the data preprocessing phase is completed like in Table 4.1 above, the next task is to present this data using the FANN data format. This can be done in a notepad or any text editor. I had taken the tagged data to a notepad and prepared the data according to the fast artificial neural network data format (see Table 4.2). The first line of the file contains three integer numbers to indicate the legend information. The first number specifies the number of data pairs; the second one indicates the number of inputs and the last integer represent the number of outputs. For this research, 40 data pairs are used to train the network for fertilizer recommendation of onion.

```
40 18 3
1 0 0 1 0 1 0 0 1 0 0 1 0 1 0 0 0 1
0 0 1
1 0 1 0 0 0 1 0 0 1 0 0 1 0 1 0 1 0
1 0 0
0 1 1 0 0 0 0 1 0 0 1 0 1 0 0 1 0 1
1 0 0
1 0 0 1 0 1 0 0 0 1 0 1 0 0 1 0 1 0
1 0 0
1 0 0 1 0 0 0 1 1 0 0 1 0 0 1 0 0 1
0 1 0
0 1 0 0 1 0 1 0 1 0 0 0 1 1 0 0 0 1
0 0 1
0 1 1 0 0 1 0 0 0 0 1 1 0 0 1 0 1 0
1 0 0
1 0 1 0 0 0 1 0 0 1 0 0 1 0 0 1 1 0
1 0 0
0 1 0 1 0 1 0 0 1 0 0 0 1 0 1 0 0 1
0 1 0
1 0 1 0 0 0 1 0 0 0 1 1 0 1 0 0 1 0
1 0 0
0 1 0 1 0 1 0 0 0 1 0 1 0 0 1 0 0 1
1 0 0
0 1 0 1 0 0 1 0 1 0 0 1 0 0 1 0 1 0
1 0 0
0 1 1 0 0 0 1 0 0 0 1 0 1 0 0 1 0 1
0 0 1
1 0 0 1 0 1 0 0 0 0 1 1 0 0 0 1 0 1
1 0 0
0 1 0 1 0 0 0 1 1 0 0 0 1 0 1 0 1 0
0 0 1
0 1 1 0 0 0 1 0 0 0 1 0 1 1 0 0 0 1
0 1 0
1 0 0 1 0 0 0 1 0 0 1 1 0 0 0 1 1 0
1 0 0
0 1 1 0 0 1 0 0 0 1 0 0 1 0 1 0 0 1
0 0 1
1 0 0 1 0 0 0 1 0 1 0 1 0 1 0 0 1 0
1 0 0
```

Table 4.2 The Fast Artificial Neural Network (FANN) Data Format

Chapter Five

Modeling the Hybrid Vegetable Expert System

5.1 Overview of the Research Effort

Most expert systems developed so far used individual approaches for developing expert systems. The commonest type of expert systems developed using these methods include the rule based expert systems in the legal and medical domain. Rule based expert systems are brittle in that they are unable to deal with problem situations where their solutions are not explicitly represented in advance in the knowledge base of the system. This will make the rule based systems inefficient to deal with unseen situations or new problems. To overcome this problem and enhance the performance of these systems, a **rule based + neural network** hybridization is performed in this research (with an aim to model a hybrid vegetable expert system). This chapter gives an explanation of how the system is modeled and the way it is expected to operate in giving consultation service to development agents.

The rule based reasoning module is prepared to represent the decision making strategy of experts (the rules) for fertilizer recommendation and disease control. On the other hand, the neural network reasoning module is used to handle decisions that are not possible to be made by the rule based module. Figure 6.1 shows the schematic representation of this idea and how the user interacts with the hybrid system through the user interface.

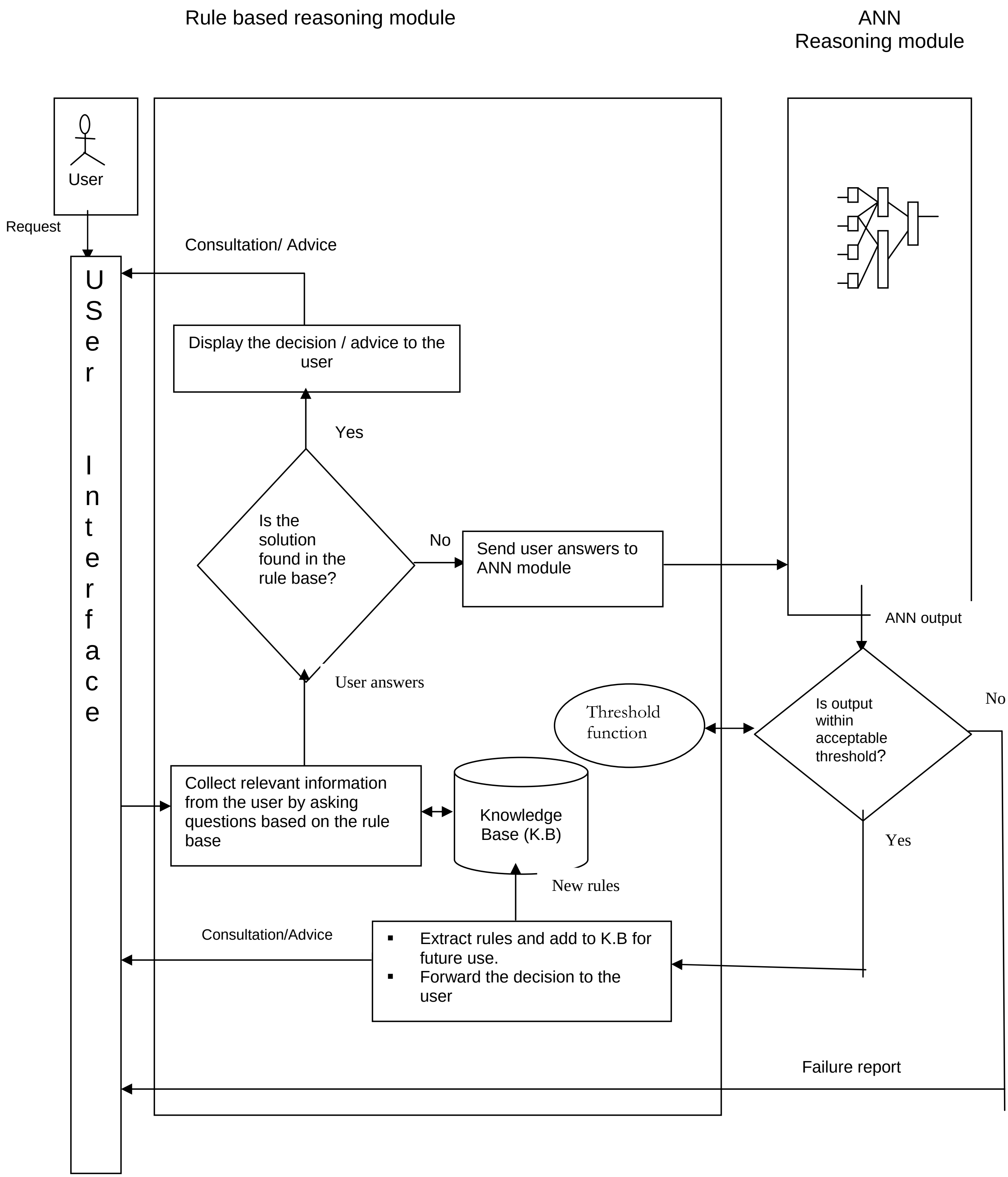

Fig 5.1 Model of the hybrid vegetable expert system

Expert decision making strategy can be automated by defining a set of condition action rules in the domain area. Rule based expert systems developed using this method are excellent in providing correct decision and explanation to the user. However, since these systems base their decision on the knowledge they already fed, their reliability is perfect provided that their knowledge base is rich and up- to -date. However, these systems have two major drawbacks: the first one is related to their requirement for manual edition of rules to make their knowledge up-to date. The second problem is attributed to their inability to deal with problems that are not identical to the facts in the rule base.

In addition, rule based systems can not generalize new problems and associate the case to the knowledge in the rule base. Human experts can make use of their intellect and subjectively judge the situation based on the knowledge gained through experience and training. However, ordinary computer programs do not have this kind of subjective judgment capability unless a special technique is embedded in such systems. Rule based systems require an exhaustive list of all the relevant knowledge and the rules in the domain. Capturing all the rules will not to be practical because of the very dynamic nature of knowledge and the difficulty to exhaustively declare all the knowledge of experts in advance.

To automate generalization and judgment based on experience of human, artificial neural networks are the best solution. Although they have the limitation of providing explanation to the user, they have excellent power of approximation if they are trained with similar cases and relevant data.

As it has been discussed in chapter one, the focus of this research is to model and implement a hybrid vegetable expert system and see how better results can be obtained by integrating rule based and neural network approaches.

5.2 Description of the Model

When DAs want to get consultation service, they enter a request to the system through the user interface. Then, the system will initiate the rule based module and prompts the user to answer some relevant questions to give the consultation. If the system finds a match in the rule base, it will provide the technical consultation through the user interface. Besides, explanation will be provided to the user as to how the system reached to that conclusion. However, when the problem situation does not have a matching solution with any one of the rules represented, the call to the rule based module will fail and hence the ANN module will be called to solve the problem. This will be done by transferring the input parameters entered by the user to the ANN.

The artificial neural network module returns the confidence for each of the output nodes. If the output passes an already set threshold value, explanations will be extracted and the decision/advice will be communicated to the user. New rules will also also be extracted from the ANN decision and added to the knowledge base.

Suppose a user started the system; he/she will be prompted to choose the desired area of consultation service such as:

- Fertilizer recommendation
- Disease control

Once the user selected the problem area, the system will start collecting information from the user by asking questions about the problem situation that he want to solve. After comparing the answers with the existing facts in the knowledge base (with the help of condition action rules in the rule based module), the system will take the next action in the following ways:

Chapter Six

Implementation of the Proposed Model

6.1 Implementation

This chapter contains two main sections. The implementation part covers the discussion of system in the rule based and neural network module. The evaluation part contains the result obtained when the trained network is tested with a test data.

6.1.1 Rule Based Module

The expert knowledge collected in chapter five are transformed in to formal language rule representation to automate the reasoning method of human vegetable experts. This module is developed using the visual prolog version 7.1 programming tool since it has an attractive feature of linking with external libraries which will be discussed later in this chapter.

In this research, the knowledge modeling process is performed for three vegetable crops (Onion, tomato and potato). In the process of deciding the correct type of fertilizer for these vegetables, experts commonly use similar decision parameters. Because of this, a system developed for one vegetable crop can be easily extended to any of the rest of the two. Hence, I decided to develop the implementation only on the data of onion vegetable crop for each consultation category.

Fertilizer Recommendation Module

This module is designed to decide which fertilizer (either DAP, Urea or a mixture of the two) should be applied on the land for optimal production of a given vegetable crop. The information needed to decide on this matter is many. Initially there were eleven parameters, which later reduced to seven after excluding those parameters that carry more or less similar concept as shown in the decision tree. The rules are manually extracted from the decision tree generated in chapter four. The next discussion focuses on the steps followed to develop the fertilizer recommendation rules for onion.

Initializing fact variables

The fact variables are in this case the variables that are used to store answers collected during the dialog with the user. These variables are initialized with empty values which are termed as “erroneous” in visual prolog.

```
prevCropLegume:real :=erroneous.
prevCropNonLegume:real :=erroneous
soilColorBrown:real:=erroneous.
soilColorRed:real:=erroneous.
soilColorBlack:real:=erroneous.
soilTextureSandy:real:=erroneous.
soilTextureClay:real:=erroneous.
soilTextureLoam:real:=erroneous.
soilpHAcidic:real:=erroneous.
soilpHAlkaline:real:=erroneous.
soilpHNeutral:real:=erroneous.
temperatureHigh:real:=erroneous..
temperatureLow:real:=erroneus.
temperatureMedium:real:=erroneous.
fixationHigh:real:=erroneous.
fixationLow:real:=erroneous.
```

Soon after the user started the system, the state of these variables will be changed to either 0 or 1 depending on the answer from the user.

The main predicate that call all the predicates containing the rules for fertilizer recommendation is the useMyRules() method.

```
useMyRules():- onion_DAP(),!.
               onion_Urea(),!.
               onion_both(),!.
```

The onion_Dap() predicate is used to answer the question of what is the criteria to recommend DAP fertilizer for onion vegetable. This predicate again calls for the rules that specify the value of the parameters required for DAP recommendation.

```
onion_DAP():-DAPRule1:()
             DAPRule2: ()
             DAPRule3: ()
```

DAPRule1 :(), DAPRule2 :() and DAPRule3 :() specify the set of information that must be asked and collected from the user in order to recommend DAP fertilizer for onion crop.

The order of the questions in the rules is arranged according to their sequence in the decision tree with the root node taking the first place in the list.

```
DAPRule1:():-
soilColorBrown:=0,soilColorRed:=0,soilColorBlack:=0,
soilColor=ask("Soil Color", "What is the soil type?",
["Brown","Red","Black"]),
if SoilColor=0 then
soilColorBrown:=1
elseif SoilColor=1 then
soilColorRed:=1
elseif SoilColor=2 then
soilColorBlack:=1
end if,
prevCropLegume:=0,prevCropNonLegume:=0,
PrevCrop=ask("Previous crop","Is the previous crop Legume?",
["Yes","No"]),
if PrevCrop=0 then
prevCropLegume:=1
elseif  PrevCrop=1 then
prevCropNonLegume:=1
end if,
soilTextureSandy:=0,soilTextureClay:=0,soilTextureLoam:=0,
Texture=ask("Soil Texture","What is the texture of the soil?",["Sand
y","Clay","Loam"]),
if Texture=0 then
soilTextureSandy:=1
elseif Texture=1 then
soilTextureClay:=1
```

```
else
soilTextureLoam:=1
end if,
PH=ask("Soil pH","What is the pH of the soil",
["Acidic","Alkaline","Neutral"]),
if PH=0 then
soilpHAcidic:=1
elseif PH=1 then
soilpHAlkaline:=1
else
soilpHNeutral:=1
end if,
soilColorBrown=1,prevCropLegume=1,soilTextureLoam=1,
soilpHAcidic=1,
note("Recommendation","The recommended fertilizer is DAP").
```

The goal is to find the situation where DAP is applied for onion. Rule1 says: “DAP is recommended to be applied if the vegetable is to be planted on brown soils having sandy texture and acidic pH previously sowed with legume crop in the past season. The above fragment of code in the rule based module collects and stores user answers from the dialog into an appropriate fact variable. The click action on the buttons in the dialog return integer values starting from 0 to ‘n’ where ‘n’ represents the number of buttons that appear in the dialog. The order of numbering is from top to bottom as they appear in the dialog window that pops up.

Every answer using the click action of the user is asserted to the corresponding fact variable already initialized with an empty value. For example, to ask the user whether the land was previously sowed with legume or not, the dialog will pop up with “Yes” and “No” buttons in it. In this case, the response will be stored in the variable “PrevCrop”. This variable will be checked for its values: 0 means the first button is clicked (which means “yes” in this case; 1 means user clicked the second button (which means “No” in this case). If the user clicks Yes in the window, the values of the fact variables prevCropLegume and prevCropNonLegume will be changed to 0 and 1 respectively. This will continue until the rule based module finds a matching recommendation in the rule base. If there is no rule for DAP that match the situation specified with the user, the system store the answers and go the next rule to check for Urea and it continues until all the rules are exhausted.

Explanation facility

Explanation facility is important to explain how the decision to recommend a given fertilizer is reached. In the hybrid vegetable expert system, this facility is attached to provide an explanation for each decision. This is done in a separate window that pops up after clicking the “How” button in a dialog. For instance when the predicate DapRule1 () succeeds for onion, the system will provide an explanation to the user through a message window. One of the problems in using artificial neural networks is their limitation to provide explanation to the user. In this research, effort has been made to overcome this limitation by introducing a predicate that extract explanation from the decision made by the neural network. Once the decision is made by the neural network, those attributes that contribute for the decision will be traced and used for constructing the explanation.

6.1.2 The Artificial Neural Network Module

The *`useMyRules()`* predicate discussed in the previous section either succeeds or fails. When it succeeds, it provides the recommendation as either DAP, Urea or Both fertilizers. It succeeds only when it finds rules that are written to handle conditions that are exactly similar to the situation provided by the user. However, this may not always be the case. There are situations where the useMyRules() predicate fails due to absence of a match in the rule base. In this case, another predicate named “useMyANN()” is called. This predicate is prepared to call the artificial neural network module developed in this research.

The module is developed by making use of the fast artificial neural network libraries written by steffen Nissen(creator and maintainer of the FANN library). The source codes for the libraries are written with the C language. To access these functions in the prolog environment, I compiled them using Microsoft visual C++ and then imported in to the program file that contain the rule based module of the system. This task realized my aim of getting access to both modules in one single interface. The next paragraphs present the methods and the steps used to train, develop and use this ANN module.

Setting the parameters of the network

Once the preprocessing and tagging of the data is finalized as discussed in chapter five, the next task is creating the neural network object and setting the parameters used for training the network. The following paragraphs present the different parameters used to model the network for fertilizer recommendation of onion. The parameters are set based on the principles and theories implemented in the FANN library by steffen Nissen[25].

Step 1: Creating the network

The artificial neural network object is created from the imported class library using the following statement.

```
ANN = fann::new().
```

Step 2: setting the number of neurons in the input, hidden and out layers:

The number of input and output neurons is determined by looking at the nature of the dataset to be used for training. To decide the fertilizer type for onion, the values of seven attributes (soil color, previous crop, soil texture, fixation rate, pH, temperature and frequency of plaughing) are taken in to account as decision parameters. Each of these attribute has two to three possible values making a total of eighteen possible values for each decision cases. Because of this, the number of input neurons is set to be eighteen. Similarly, there are three possible values for recommendation column as DAP, Urea and Both. This made the number of output neurons to be set to three. The remaining is setting the hidden neurons. There is no easy way of determining the number of neurons in the hidden layer. In this research, this is determined to be nine through experimentation by looking at the change in accuracy of the model (by changing the number to higher or lower values).

ANN:create_standard([18,9,3])

This line of code builds a three layered network with 18 neurons in the input, 9 neurons in the hidden and 3 neurons in the output layers.

Step 3 : Setting the activation steepness

ANN:set_activation_steepness_hidden(1.0)

ANN:set_activation_steepness_output(1.0)

The steepness of an activation function indicates how fast the activation function goes from the minimum to the maximum. To train neural networks where the output values should be at the extremes (usually 0 and 1, depending on the activation function), a steep activation steepness can be used (like 1.0) [25].

Based on this, the activation steepness for the hidden and output neurons are set to be 1.o.

Step 4 :setting the activation functions and train stop function

ANN:set_activation_function_hidden(fann::fANN_SIGMOID

ANN:set_activation_function_output(fann::fANN_SIGMOID)

ANN:set_train_stop_function(fann::fANN_STOPFUNC_MSE)

This is a function used to transform the activation level of a neuron into an output signal. A number of activation functions are supported in the FANN library, but the sigmoid function is selected for this research since its supported range lies between 0 and 1 which is appropriate to the training data tagged for fertilizer recommendation of onion.

The train stop function is used to determine the stop criteria during training. I selected train stop function to be mean squared error.

ANN:set_train_stop_function(fann::fANN_STOPFUNC_MSE)

There is no special reason for selecting this function among the two functions supported by FANN. fANN_STOPFUNC_MSE sets the stop criterion to be the mean square error (the mean value of the squared difference between the actual and desired output of the network). fANN_STOPFUNC_BIT sets the stop criterion to be the number of bits that fail. The limit is the maximum accepted difference between the desired output and the actual output during training. Each output that diverges more than this limit is counted as an error bit [25].

Step 5: Training the network

ANN:train_on_file("onion_Fertilizer.train", 1000, 0.10000)

Once the important parameters are set, the network was trained by adjusting the parameters based on the mean square error (MSE) reported during the training. The training file is tagged and saved to the file "onion_Fertilizer.train" inside the Exe folder of my prolog file. The value 0.10000 indicates the desired mean square error to train the network. This means the network will stop training when the mean square error falls below the specified value.

The FANN library supports four types of training algorithms (INCREMENTAL, BATCH, QUICK PROP AND RPROP). I selected the incremental method since I found the desired error at smaller epoch as compared to the rest of the three algorithms. With this algorithm, I get the desired error at 45th epoch. It is found that MSE is obtained at 58th epoch when the QUICK algorithm is used while it is obtained at 174th epoch when the RPROP algorithm is used. The BATCH algorithm did not converge to the desired error even within the 1000 epoch limit set for the training. Figure 6.1 and 6.2 below show the error report using the incremental and QUICK method of training. See Annex IV to look at the report for the rest of the algorithms.

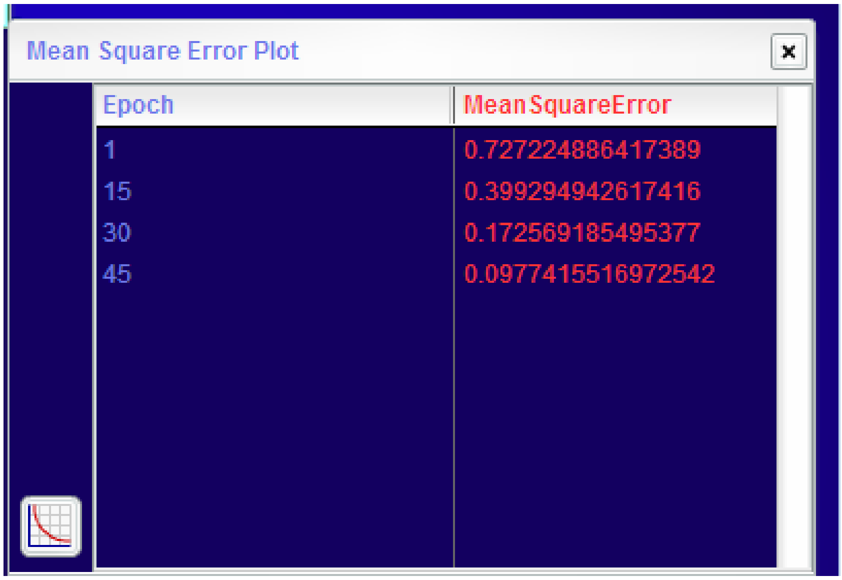

Fig 6.1 Mean Squared Error Values Using the Incremental Method of Training

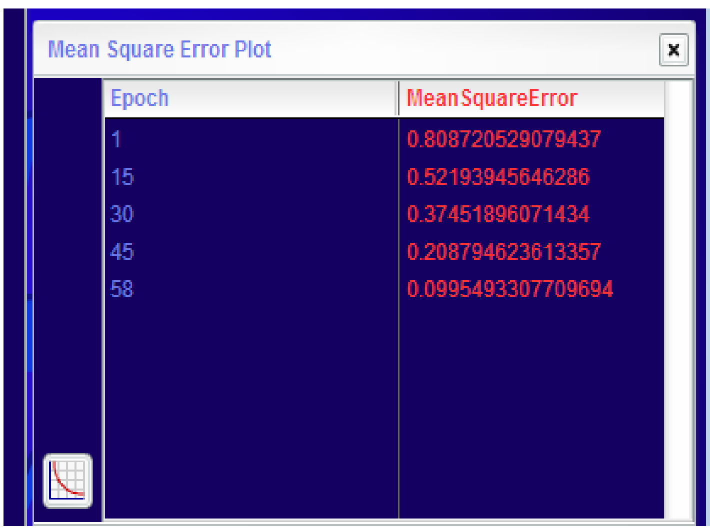

Fig 6.2 Mean square error report using the QUICK method of Training

Step 6: using the network

Once the network model is developed by setting the appropriate parameters, it can be called in the hybrid system when the rule base module fails.

onion_display_Decision (ANN:run(inputList())).

The function onion_display_Decision is prepared to compare the value returned by ANN:run() with the threshold limit which is set to be 0.6 by observing the result of the experiment . Then, this function displays the recommended fertilizer to the user. InputList() is a function written to pass the list that contains the current content of the fact variables. That is, the fact variables were initialized with empty facts; which will be later updated with the new value as the result of a dialog interaction between the system and the user.

The artificial neural network model for disease control of onion is created with twenty three neurons in the input layer, twelve neurons in the hidden layer and five neurons in the output layer.

As it has been explained in the previous sections, the hybrid model developed in this research has two main modules**.** The performance of the rule based system is dependant on the kind of user request as it requires exact match of the request with the rules in the knowledge base. However, attempt has been made to minimize this weakness with the help of the second module (ANN module) developed by training the network with similar data used to develop the rule based module.

Once the prototype is developed, ten random queries of consultation requests were presented to both the rule based module (before integration) and the hybrid system (after integration). The hybrid system responded to eight of these questions while the rule based module alone provided answers to only five of these questions.

6.2 User Interface

The rules and the neural network data discussed in chapter five are only understandable to developers of expert system. They can not be easily used by ordinary users as they are. They must be provided to the user with simple and clear communication media between the user and the system. For this purpose, a user interface is designed for the prototype expert system.

User interface to enter to the main window

When the program starts, user will find two options under the consultation menu where he can select the kind of consultation service he wants (Fertilizer Recommendation or disease control as shown on Fig 6.3). Once he/she selected the main category of consultation request, again the user will be provided with other options to select about which vegetable crop he/she may want to ask for consultation.

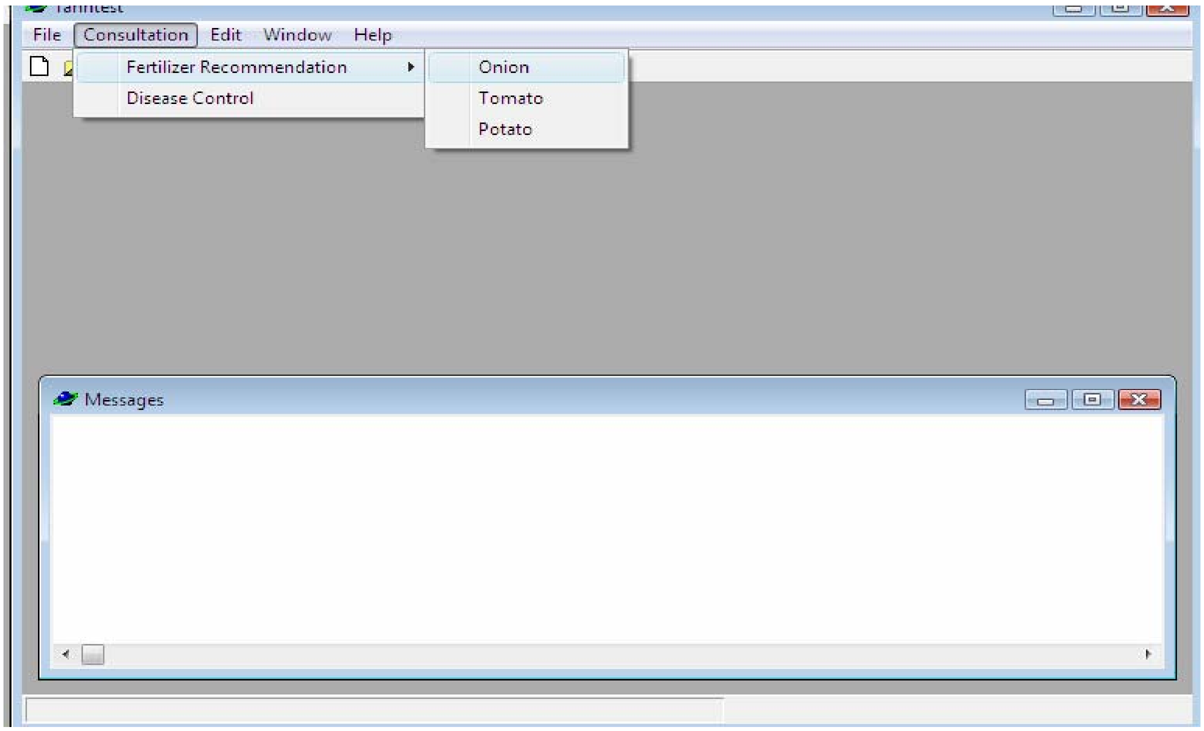

Fig 6.3 Main Window for Hybrid Vegetable Expert System

If the user selected “Fertilizer Recommendation” in the main window, a dialog window will pop up, displaying a series of questions. The following snapshots are dialog cases taken from one possible situation when the user asks for consultation on fertilizer recommendation of onion.

Let us take the case where user selected *onion* under “Fertilizer Recommendation”. The system will start the dialog by asking first, the soil color as it is the most important criteria for determining the type of fertilizer required for onion. See Fig 6.4 for schematic demonstration.

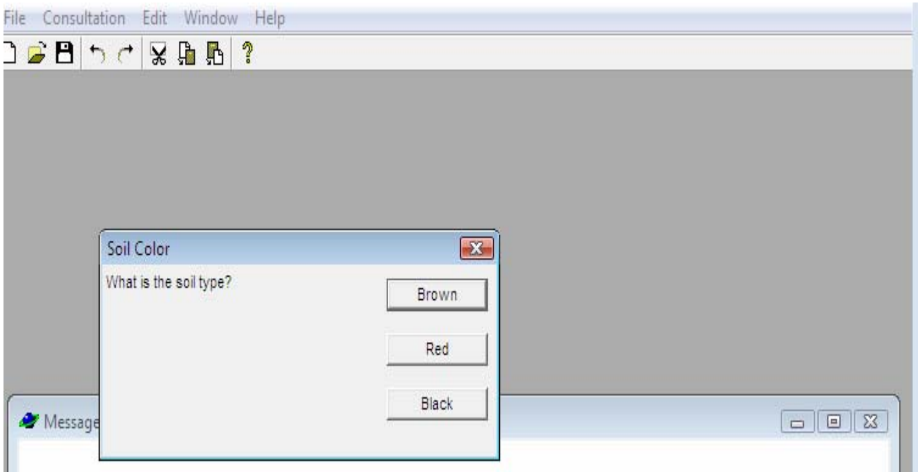

Fig 6.4 Dialog Interface to Collect Information about the Soil Color

If the user clicked brown in the above dialog, another window that will ask for information about the type of crop that had been sowed in the previous season will pop up as shown in Fig 6.5.

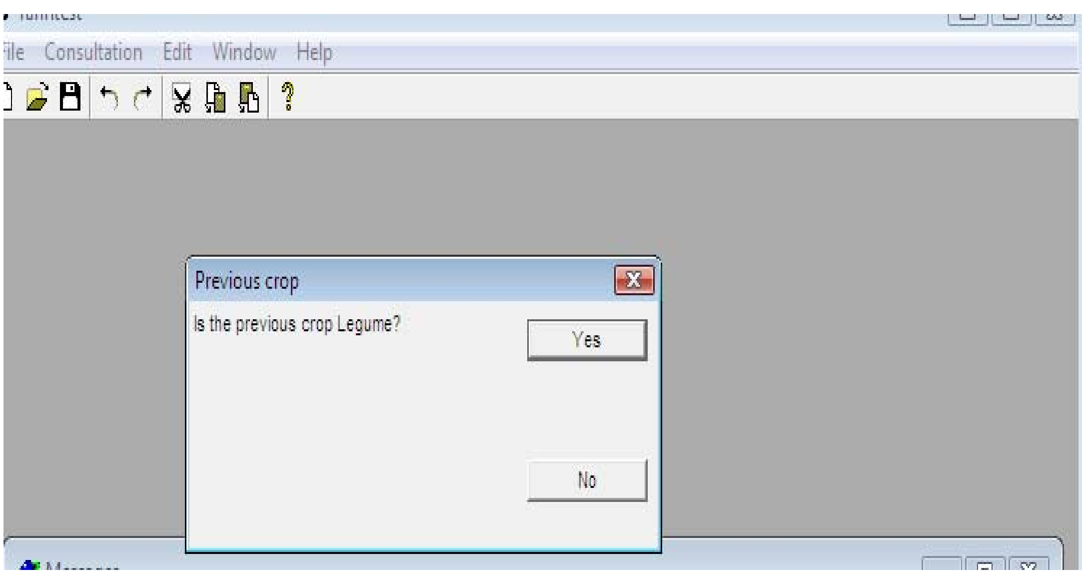

Fig 6.5 Dialog Window to Ask the Type of Previous Crop

If the user clicks yes in the above dialog, it means that the previous crop is legume. This fact will be asserted in to corresponding fact variable in the rule based module. Then, the following dialog that will ask information about the textural property of the soil as shown in Fig 6.6:

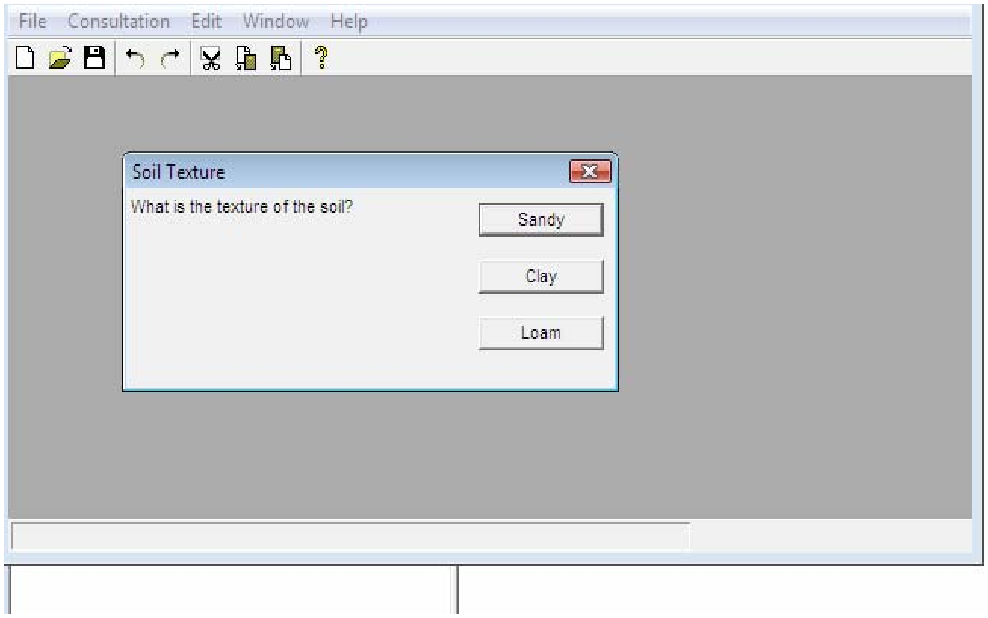

Fig 6.6 Dialog Window to Collect Information about Soil Texture

Suppose the user has clicked loam soil in the above dialog. The system is left with one important information to forward the solution using the knowledge in the rule base. That information is the pH of the soil with which this crop is planed to be sowed/ planted. The dialog in Fig 6.7 is used to prompt this information.

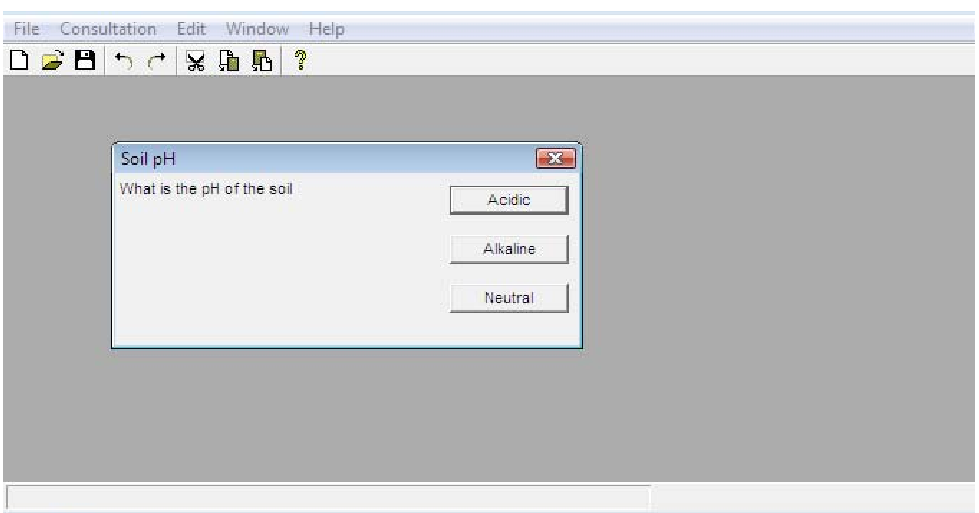

Fig 6.7 Dialog Interface to Collect Information about Soil PH

If the user clicked acidic, a message window that displays the recommendation will pop up. This is shown in Fig 6.8 below.

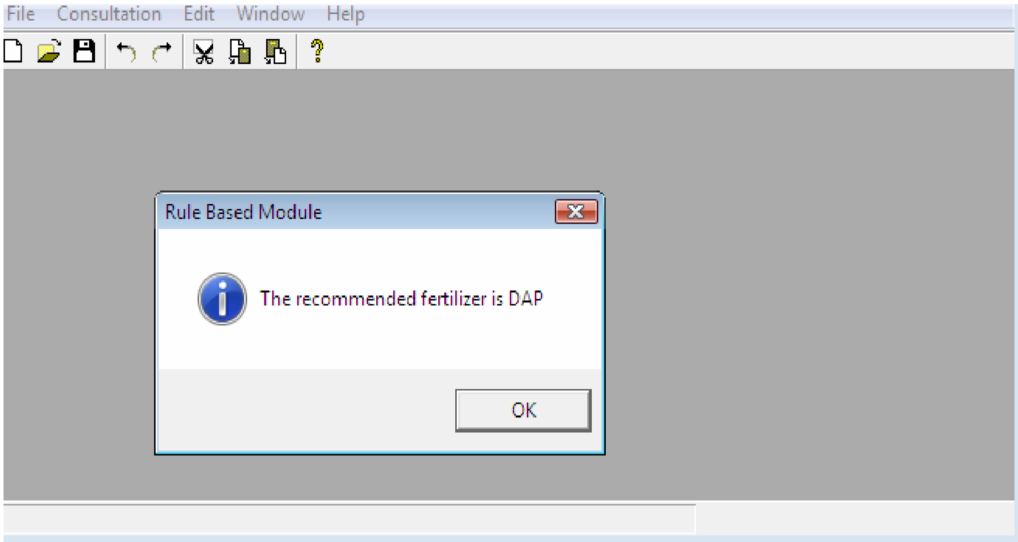

Fig 6.8 Recommendation of the Rule Based Module

In most cases, user may want to know how the system reached to this conclusion. To provide answer for that, an explanation facility is provided as follows:

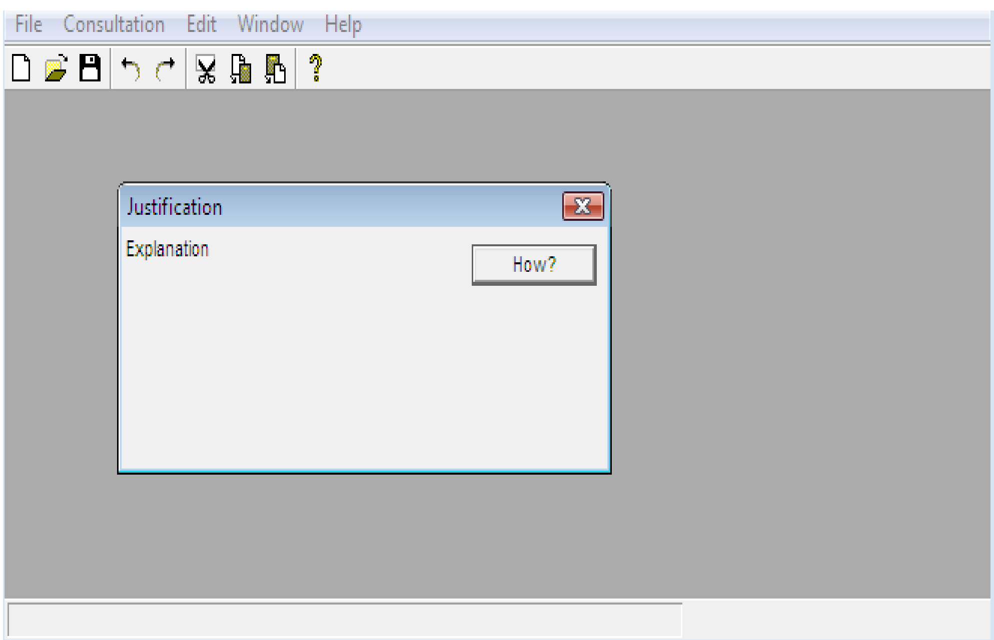

Fig 6.9 Dialog Window That Provide Option for Explanation

When the user clicked the “How” button in the above dialog, the system will display the justification of how it reached to the conclusion of deciding the recommended fertilizer. See Fig 6.10 for schematic demonstration.

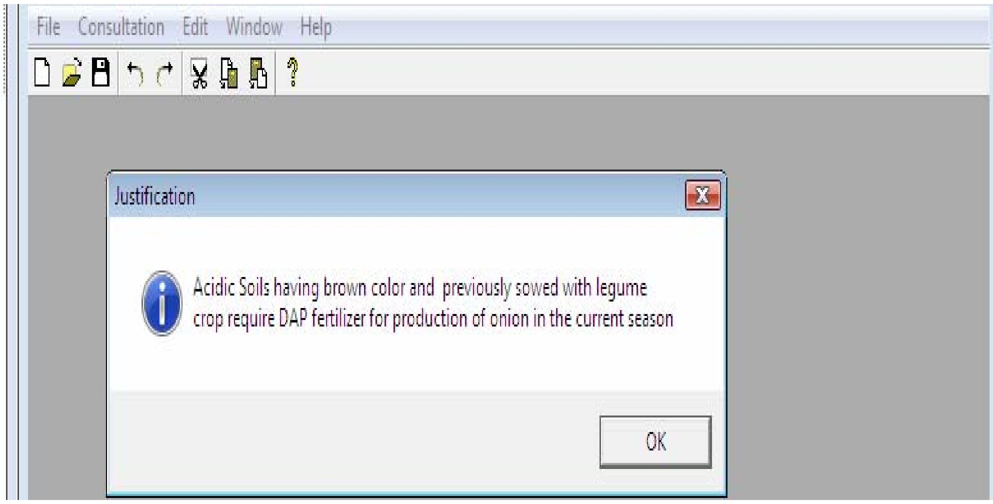

Fig 6.10 Justification of the Conclusion

User may want to calculate the rate at which the recommended fertilizer should be applied to the land. This option is also provided as shown in Fig 6.11. The sample implementation codes are attached in Annex **III** of this document.

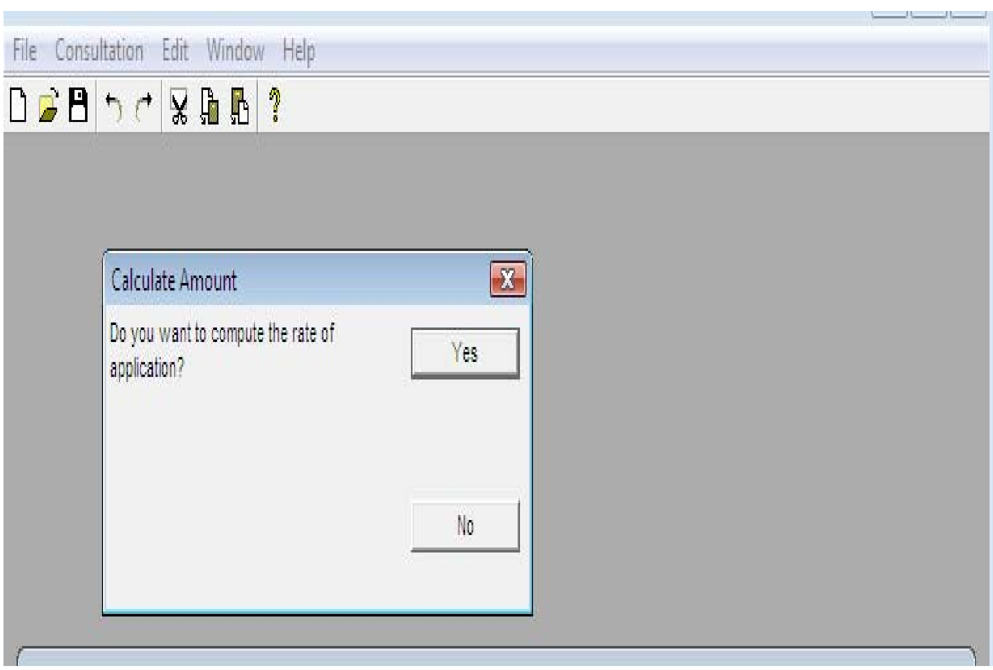

Fig 6.11 Window to Ask User’s Interest to Compute Rate of Application

If the user clicked "yes" in the above dialog, the system will first ask for the time of application of the fertilizer. The task of calculating the required amount is considered to be easier for the lower level professionals as it only requires the information about the time of application since fertilizer dosage is dependent on this information. Generally, DAP is required at planting time and Urea is required at leaf stage (after planting) for healthy vegetative growth of the plant. When both are required, a mixture of half of the normal rate from each fertilizer is calculated and applied. This is presented in Fig 6.12.

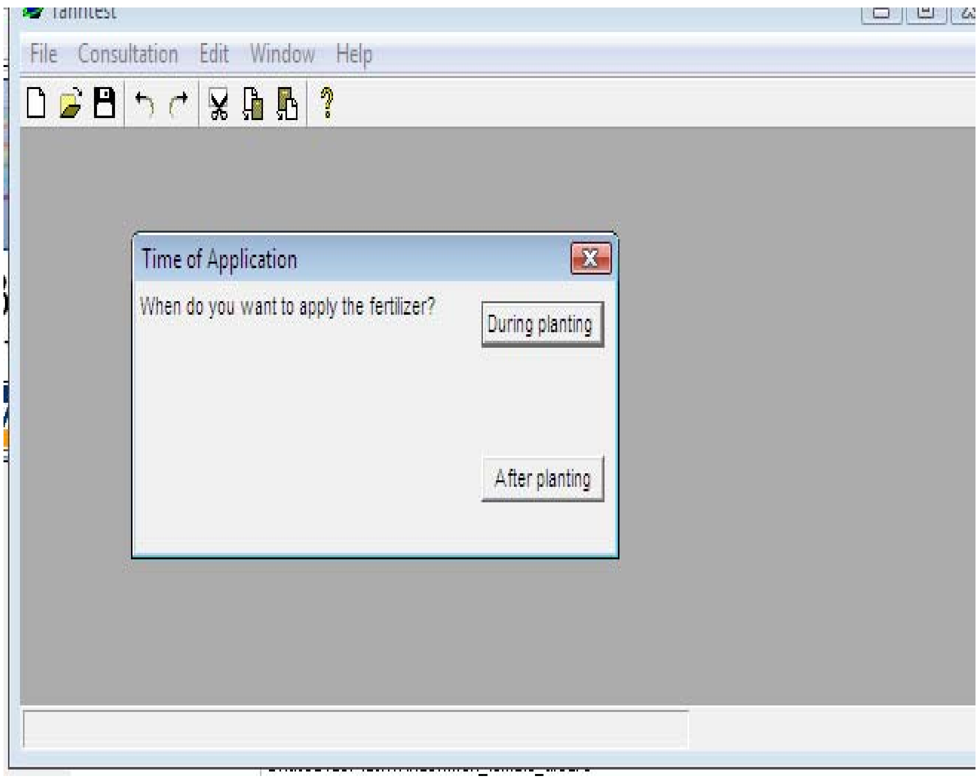

Fig 6.12 User Interface To Select the Time of Application of the Fertilizer

The last interface for the typical case taken from the system in the above dialogs will be a window that displays the rate of application of the recommended fertilize. This is presented in Fig 6.13 below:

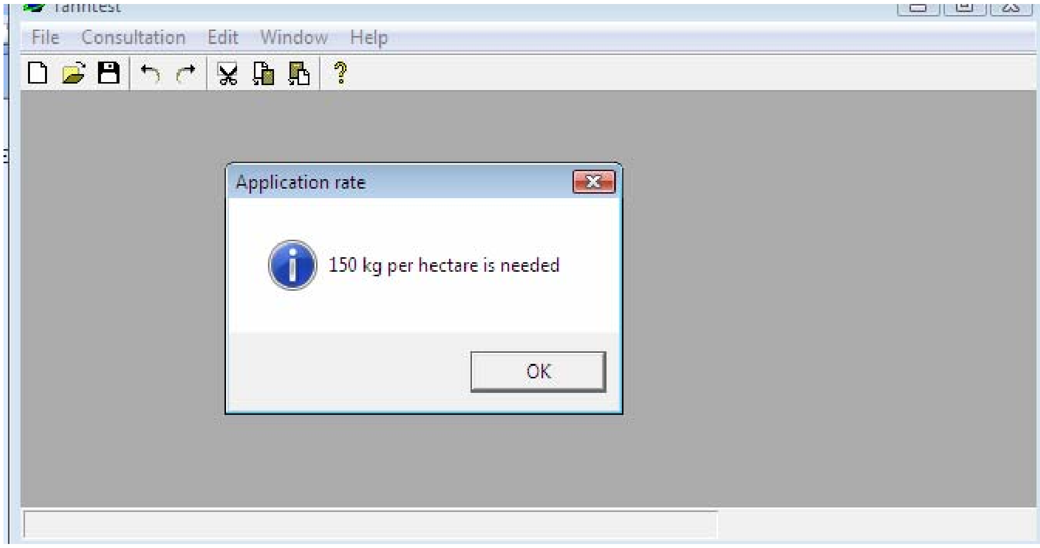

Fig 6.13 Window to Display the Rate of Application of the Given Fertilizer

When the rule based module in the above dialog does not find a match in the rule base, all the answers collected from the user will be transferred to the ANN module for decision as input parameters. In this case, the ANN module will be called to take the decision. If the result returned from the ANN module passes a thresh hold value, the decision will be communicated to the user through the user interface. By looking at the out put of the network, I set the thresh hold value to be 0.6 in this research.

Chapter Seven

Conclusion and Recommendation

7.1 Conclusion

Providing proper technical assistance to farmers demands availability of well qualified agricultural experts at a community level. However, with the staff size and educational level of most employees currently assigned to work with the farmers, provision of this service at the desired level is practically challenging. That means, assigning specialists at a community level is difficult due to shortage of high level specialists. The problem is more intense particularly in areas of vegetable production. The researcher believe that this problem can somehow be lessened by assisting the expert decision making process of lower level professionals, with the help of an expert system technology. This research is conducted with an aim to contribute to the effort in developing a reasoning system that can aid the day to day decision making process of lower level professionals in the production of vegetable crops.

There are a number of approaches for developing expert systems ranging from rule based approaches to methods that employ neural network techniques. Rule based expert systems developed in the previous days are excellent at providing explanation to their decision. However, they have the limitation to deal with problem situations that are not seen in the rule base. In order to provide comprehensive consultation service to users, they should be integrated with systems developed through machine learning approaches.

The work in this research is conducted on developing a hybrid vegetable expert system model through combination of the rule based and artificial neural network approach. To build the hybrid model, knowledge acquisition, data preprocessing, rule generation, knowledge representation and model integration tasks had been performed. From the knowledge collected for the research, rule based module is prepared for fertilizer recommendation and disease control. The artificial neural network module is created by making use of the C functions implemented in the fast artificial neural network libraries (used after compiling and importing them in the prolog environment). The data for training the artificial neural network model was prepared after preprocessing and eliminating some decision attributes that carry similar meaning and have no effect on the overall decision. Finally the rule based and the neural network modes are integrated to operate on one single system.

Based on the result obtained after implementing the hybrid model using a prototype, the effort made in this work produced interesting result. Although there still remained some gap left to complete this work, the researcher believes that the output of this research can be implemented after minor adjustments and extensions and it can serve as an important input to future works in this area.

7.2 Recommendations

The following are the most important issues the researcher believed that future work on this area should address to fill the gap identified in this work.

The rules used for developing the prototype hybrid vegetable expert system are manually generated by looking at the results of the decision tree. It is better to improve this process by looking at other alternatives to automatically generate the rules.

The algorithm used to train the neural network is TRAIN_INCREMENTAL. Future research work in this area may need to incorporate a comparative study on the performance of other algorithms such as TRAIN_BATCH, TRAIN_RPROP and TRAIN_QUICKRPROP by changing the other parameters of the network so that one best algorithm can be selected for the domain.

The result obtained with the hybrid model is based on limited data collected from domain experts. Future researchers on this area may incorporate more data and conduct experiments to see the change on the performance of the model.

The decisions made by the neural network are accepted when the output neurons return a value beyond an already set threshold value. As indicated in the model, rules can be extracted and added back to the neural network module to enhance the learning and decision process. However, this process is not implemented in this work. Future work in this domain may implement and test how the system reacts to the change in terms of performance.

Bibliography

1. National Bank of Ethiopia. 2007. *Annual report. National Bank of Ethiopia*, Addis Ababa.
2. Ministry of Agriculture and rural development. 2006. *Agricultural policies, programs and target plan for accelerated and sustainable development to end poverty. Ministry Agriculture*, Addis Ababa.
3. M.L Chadha, Efren C. Altoveros, R. Nono-Womdim and H. Mandiga .2000. *Variety evaluation and seed production of vegetables*. AVRDC, African regional program
4. Ravi kochhar. 2000. *The falling cost of computing*. University of Wisconsin – Madison, united states.
5. African press international (API). 2007. *Ethiopian Electrification program*. *Retrieved* November 16, 2008 from http:// www.africanpress.com.
6. Sanchez.E and Zadeh.L.A, ed. 1988. *Approximate reasoning in intelligent systems, decisions and control*. Pergamon press inc.UK.
7. University of Texas. Expert systems. Retrieved July 22, 2008 from http://www.eco..utexas.edu/faculty/Norman/long.extra/info.s98/exp/tree.html
8. Masoud Yazdani, ed.1986. *Artificial intelligence principles and applications.* University of Exeter.
9. AAAI. 20008. *Expert systems, AI news*. Retrieved Aug 20,2008 from http://www.aaai.org/AITopics/newsTopics/mail.html
10. Ben Krose and Patrick van der Smagt.1996. *An introduction to neural network.* University of Amsterdam, the Netherlands.
11. Zemede Asfaw.1997. *Conservation and use of traditional vegetables in Ethiopia*. Faculty of Science, Addis Ababa University, Addis Ababa.
12. Crop Research Directorate. 2000. *Horticultural crop research strategy*. Ethiopian Agricultural research organization (EARO), Addis Ababa.
13. M.L Chadha, Efren C. Altoveros, R. Nono-Womdim and H. Mandiga .2000. *Variety evaluation and seed production of vegetables*. AVRDC, African regional program.

14. Lemma Desalegn.2002. *Tomatoes research experiences and production prospects*. Ethiopian Agricultural research organization (EARO), Addis Ababa

15. Dr Tekalign Tsegaw. 2007. *Production of commonly grown vegetables in Ethiopia.* Haramaya university, college of agriculture, Haramaya, Ethiopia .

16. Thomas A. O'callaghan.2003. *SHYSTER-MYCIN, a hybrid legal expert system*. Australia University. Canberra, Australia.

17. V. Van Hoof et al. 2004. *Medical expert system developed in J.MD*. Amesterdam IOS press, Amsterdam.

18. Thomas Siesmeier et al. 2005. *Use of an expert system for diagnosis of suspected alzheimer's disease.* Department of nuclear medicine, Mainz, Germany.

19. Ermias wondie.2005.*An expert based system for social security determination in Ethiopia* . A thesis submitted to the department of information science, Addis Ababa university, Addis Ababa.

20. Rediet Alemu. 2006. *Design and development of prototype knowledge base system for HIV pre test counseling*. A thesis submitted to the department of information science, Addis Ababa university, Addis Ababa.

21. Seifu Tatek. 2005. *A rule based/neural network hybrid legal expert system.* A thesis submitted to the department of information science, Addis Ababa University, Addis Ababa.

22. Lemma Dessalegne. 2002. *Major vegetable crop varieties and their production practices.* EARO, Addis Ababa.

23. Mohamed Yesuf. 2002. *Major vegetable crop disease and their control measures*. EARO, Addis Ababa.

24. J.R. Quinlan. *Decision trees for supervised learning.* Retrieved November 10, 2008 from http://grb.mnsu.edu/grbts/doc/manual/J48_Decision_Trees.html

25. Steffen Nissen. 2003. *Implementation of a fast artificial neural network library.* Department of computer science, university of Copenhagen (DIKU).

26. Rajkishore Prasad, Dr. Ashok Kumar Sinha. *Role of Expert System in Natural Resources Management,* retrieved Aug 15, 2008 from http:// www.gisdevelopemnet.net /applicatiion/nrm/overview.htm

27. S. ALY, I. VRANA.2006. *Toward efficient modeling of fuzzy expert systems.* Czech University of Agriculture, Prague, Czech Republic.

28. D.Ashagari. *A survey of parasitic vegetable diseases in eastern Ethiopia.* International society for horticultural science. Retrieved November 16, 2008 from http://www.pubhort.org

29. Thomas W. de Boer. 2003. *A beginner's guide to visual prolog*. Retrieved September 5, 2008 from http://download.pdc.dk/vip/71/deBoer.

Annexes

Annex I

Sample questions that were asked during the interview with vegetable experts

1. What are the major problems of vegetable production in the agriculture sector
2. How is the assignment(allocation) of budget and vegetable experts for weredas and zones
3. To what level is the knowledge of development agents adequate to support farmers
4. What are the major fertilizers commonly distributed to farmers for vegetable production
5. What are the problems usually encountered in fertilizer application
6. What are the parameters that should be taken in to account in recommending fertilizer dosage for a given type of vegetable
7. What kind of questions are usually asked by farmers in relation to seeking expertise advice on vegetables
8. How do you solve problems that arise due to over dosage of fertilizer and disease infestation
9. Can you explain the different vegetable diseases you know including their symptoms, control measures and level of severity (economic thresh hold level)

Annex II

Questions commonly asked by farmers seeking help from experts

See the attached papers that are filled by experts when they are asked to write down the kind of questions they usually encounter.

Annex III

Sample Rules Taken from the Rule Based Module

```
clauses
   onion_DAP():-onion_DAPRule1(),! .
   onion_DAP():-onion_DAPRule2(),! .
   onion_DAP():-onion_DAPRule3() ,!.
clauses
   onion_Urea():-onion_UreaRule1(),!.
   onion_Urea():-onion_UreaRule2(),!.
   onion_Urea():-onion_UreaRule3(),!.
 clauses
   onion_Both():-onion_BothRule1(),!.
   onion_Both():-onion_BothRule2(),!.
   %onion_Both():-onion_BothRule3(),!.
clauses
onion_DAPRule1():-

soilColorBrown:=0,soilColorRed:=0,soilColorBlack:=0,
SoilColor=ask("Soil Color", "What is the soil type?",["Brown","Red","Black"]),
if SoilColor=0 then
soilColorBrown:=1

elseif SoilColor=1 then
soilColorRed:=1
elseif SoilColor=2 then
soilColorBlack:=1
end if,

prevCropLegume:=0,prevCropNonLegume:=0,
PrevCrop=ask("Previous crop","Is the previous crop Legume?",["Yes","No"]),

if PrevCrop=0 then
prevCropLegume:=1
```

```
elseif  PrevCrop=1 then
prevCropNonLegume:=1
end if,

soilTextureSandy:=0,soilTextureClay:=0,soilTextureLoam:=0

Texture=ask("Soil Texture","What is the texture of the soil?",["Sandy","Clay","Loam"]),
if Texture=0 then
soilTextureSandy:=1

elseif Texture=1 then

soilTextureClay:=1

else
soilTextureLoam:=1
end if,

soilpHAcidic:=0,soilpHAlkaline:=0,soilpHNeutral:=0
PH=ask("Soil pH","What is the pH of the soil",["Acidic","Alkaline","Neutral"]),
if PH=0 then
soilpHAcidic:=1

elseif PH=1 then
soilpHAlkaline:=1

else
soilpHNeutral:=1
end if,
soilColorBrown=1,prevCropLegume=1,soilTextureLoam=1,soilpHAcidic=1,note("Recommen
dation","The recommended fertilizer is DAP"),
clauses
onion_DAPRule2():-
Temp=ask("How hot is the average daily temperature?",["High","Medium","Low"]),
if Temp=0 then
temperatureHigh:=1,temperatureMedium:=0,temperatureLow:=0
elseif Temp=1 then
temperatureLow:=0,temperatureMedium:=1,temperatureLow:=0
else
```

```
temperatureHigh:=0,temperatureMedium:=0,temperatureLow:=1
end if,

soilColorRed=1,temperatureLow=1,soilTextureClay=1,note("Recommendation","The recom
mended fertilizer is DAP"),

onion_DAPRule3():-

Fixation=ask("How is the fixation rate of the soil?",["Low","High"]),
if Fixation = 1 then
fixationHigh:=1,fixationLow:=0
else
fixationHigh:=0,fixationLow:=1
end if,
fixationLow=1,soilColorBlack=1,soilTextureSandy=1,
note("Recommendation","The recommended fertilizer is DAP"),
```

Annex IV

Mean Square Error Report during ANN Training

Mean Square Error Plot

Epoch	MeanSquareError
1	0.808720529079437
15	0.666515290737152
30	0.552276909351349
45	0.43748489022255
60	0.349486917257309
75	0.252598345279694
90	0.186488002538681
105	0.133270159363747
120	0.107516489923
126	0.0995986759662629

Error report using the RPROP algorithm of training the neural network

Mean Square Error Plot

Epoch	MeanSquareError
870	0.200802952051163
885	0.197304353117943
900	0.193774282932281
915	0.190186694264412
930	0.186531111598015
945	0.18282113969326
960	0.179092094302177
975	0.175386846065521
990	0.171741619706154
1000	0.169357046484947

Error report using the BATCH algorithm of training the neural network

Annex V

Sample codes used in the modules

```
predicates
 convertSoilColor:(real,real,real)->string.
 clauses
 convertSoilColor(First,Second,_)=SoilColor:-
 if First=1 then
 SoilColor="Brown "
 elseif Second=1 then
 SoilColor="Red  "
 else
 SoilColor="Black "
 end if.
  predicates
  convertPH:(real,real,real)->string.
 clauses
 convertPH(First,Second,_)=PH:-
  if First=1 then
  PH=concat("Acid "," soils having ")
   elseif Second=1 then
   PH=concat(" Alkalin "," soils having ")
   else
   PH=concat(" Neutral "," soils having ")
   end if.
 predicates
 convertFixation:(real,real)->string.
 clauses
 convertFixation(First,_)=Fixation:-
  if First=1 then
  Fixation=concat(" high "," fixation rate, ")
   else
   Fixation=concat(" low "," fixation rate, ")
   end if.
 predicates
 convertTexture:(real,real,real)->string.
 clauses
 convertTexture(First,Second,_)=Texture:-
  if First=1 then
  Texture=concat(" sandy ", " textural property, ")
   elseif Second=1 then
   Texture=concat(" clay "," textural property, ")
   else
   Texture=concat("loam","property")
   end if.
```

Declaration

The thesis is my own and has not been presented for a degree in any other university and all the sources of material used for the thesis have been duly acknowledged.

Tsegaw Kelela

January, 2009

This thesis has been submitted with my approval as a university advisor

----

Dr. Dejene Ejigu

Advisor