AN ANDROID-BASED EXPERT SYSTEM FOR DIAGNOSIS OF SELECTED TROPICAL DISEASES USING FUZZY-ANALYTIC HIERARCHY PROCESS

acy
[P)

BY

ALEGBELEYE OLUBUNMI JEHOFADUNSIN

(CPE/13/1074)

SUBMITTED TO

THE DEPARTMENT OF COMPUTER ENGINEERING,

FACULTY OF ENGINEERING,

FEDERAL UNIVERSITY OYE-EKITI,

NIGERIA.

IN PARTIAL FULFILMENT OF THE

REQUIREMENT FOR THE AWARD OF BACHELOR OF ENGINEERING

(B.ENG.) DEGREE IN COMPUTER ENGINEERING

MARCH, 2019

DECLARATION

This project is my own work and is in my own words (except where I have attributed it to others). I know that plagiarism is wrong. Plagiarism is to use another's work and pretend that it is one's own. I have used the APA convention for citation and referencing. Each contribution to, and quotation in, this project from the work(s) of other people has been attributed and has been cited and referenced. Any section taken from an internet source has been referenced to that source. I have not allowed, and will not allow, anyone, to copy my work with the intention of passing it off as his or her own work.

Alegbeleye Olubunmi — Student's name.

29/3/2019 — Signature & Date.

CERTIFICATION

This project with the title

AN ANDROID-BASED EXPERT SYSTEM FOR DIAGNOSIS OF SELECTED TROPICAL DISEASES USING FUZZY-ANALYTIC HIERARCHY PROCESS.

Submitted by

ALEGBELEYE OLUBUNMI JEHOFADUNSIN

(CPE/13/1074)

Has satisfied the regulations governing the award of the degree of

BACHELOR OF ENGINEERING (B.Eng.) in Computer Engineering

Federal University Oye-Ekiti, Ekiti State Nigeria.

29:03:9

Engr. (Dr.) O. M. Olaniyan
Supervisor

Date

29:03:19

Engr. (Dr.) O.M. Olaniyan
Head of Department

Date

DEDICATION

This project is dedicated to my parents, Mr and Mrs Alegbeleye, for their unfailing support so far.

ACKNOWLEDGEMENTS

My gratitude goes to my Project supervisor, Engr. (Dr.) Olaniyan O.M. for his guidance and patience in the course of undergoing this project work. I want to show my gratitude to the lecturers of the Department of Computer Engineering. I want to thank the entire administrative members of staff of the Department of Computer Engineering.

I want to appreciate my colleagues for their support ever since we got on the train of higher education together.

It would be very unfair to forget to mention the Medical Doctors who offered to contribute their experience to this work through structured or unstructured interview. I express my sincere gratitude.

Finally, my greatest appreciation goes to Jehovah, the one who created all things and gives life and health.

ABSTRACT

Nigeria, a country in the tropics, is not free from tropical diseases. The commonest ones being malaria, typhoid and tuberculosis. Malaria alone results in over 300,000 deaths every year. Adding to this, there is a serious shortage of Doctors. A survey by the World Health Organization has shown that there is approximately 1 Doctor to about 2,660 Nigerians. Due to this, the Health sector is stressed beyond its carrying capacity and Doctors cannot perform at their best. Nigerians are discouraged from going to the hospital because when they do, they meet with unending queues and angry health personnel. These result in self-medication, drug abuse and deaths. Several works have been developed to aid diagnosing of tropical diseases using fuzzy logic. This work developed an Android-based solution to shorten patient waiting time.

Artificial Intelligence is being applied in several endeavours including medicine. These Systems have shown greater accuracy than experts in the fields where they serve. This performance benefit should also be harnessed for the diagnosis and management of tropical diseases in Nigeria. This project develops an Android-based expert system for the diagnosis of some tropical diseases using hybridization of Fuzzy logic and Analytic Hierarchy Process. At the knowledge acquisition stage, questionnaires were administered to medical doctors at four hospitals. Fuzzy rules were extracted from the questionnaires using a method called Learning from Example. The system was designed on MATLAB and FuzzyTECH software. Subsequently, it was implemented on the Android platform using Java programming language and XML.

Evaluation was done by comparing the sensitivity and consultation time of the developed system to that of medical doctors. The system showed 72% sensitivity compared to 62% of medical doctors and consultation time was reduced by up to 15%. This study has shown that an Android-based Fuzzy-AHP diagnosis system can help to reduce the stress on the health sector in Nigeria and other developing countries thereby reducing mortality due to tropical diseases.

TABLE OF CONTENT

CHAPTER THREE: METHODOLOGY

CHAPTER FOUR: RESULTS AND DISCUSSION

CHAPTER FIVE: CONCLUSION AND RECOMMENDATIONS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS AND ABBREVIATIONS

$\mu \underline{\mathrm{B}}(x)$	Degree of membership of element x in fuzzy set B
AHP	Analytic Hierarchy Process
AI	Artificial Intelligence
ANP	Analytic Network Process
F-AHP	Hybridization of Analytic Hierarchy Process and Fuzzy Logic
IDE	Integrated Development Environment
KB(E)S	Knowledge Based (Expert) System
LFE	Learning From Example
MCDA	Multiple Criteria Decision Analysis
MDSS	Medical Decision Support System
PDA	Personal Digital Assistant
SDK	Software Development Kit
SGML	Standard Generalized Mark-up Language
XML	Extensible Mark-up Language

CHAPTER ONE

INTRODUCTION

1.1 GENERAL OVERVIEW

Artificial Intelligence (AI) is the field of computer science concerned with designing computer systems and computational models that exhibit the characteristics we associate with intelligence in human behaviour. These characteristics include understanding language, learning, reasoning, perception, problem-solving and so on. In simple terms, Artificial Intelligence is the design of computer systems that think and act like humans (Russell & Norvig, 1995). This discipline was formally initiated in 1956. The first AI conference, organised by John McCarthy, Marvin Minsky, Nathaniel Rochester and Claude Shannon was held in that year at Dartmouth College in New Hampshire. This conference was the first organised effort in the field of artificial intelligence. The Dartmouth conference paved the way for the development that occurred in the field of AI (Jones, 2008). Artificial Intelligence has found usage in natural language processing, robotics, computer vision, gaming systems and machine learning among others. Over the years, Artificial Intelligence has been applied to medicine in various ways. It has found uses ranging from diagnostic aid, therapeutic aid, training, research aid, managing data record, doing repetitive jobs, data analysis, drug creation to health monitoring (Fieschi, 1990).

One of the major advancements of AI is the development of computer systems which separate the underlying knowledge from the procedural part of the system. This allows the knowledge to be stored and edited without changing a single line of the program. This feature is a major advantage of expert systems over conventional programs. These systems are called Knowledge-Based Systems (KBS). If such systems contain the knowledge of experts in a field, they are called Expert System (ES) or more comprehensively, Knowledge-Based Expert System (KBES). In its simplest form, an expert system will contain a knowledge base and an inference engine. An expert system is an intelligent computer program that simulates the decisions and actions of an expert in a field by consulting a knowledge database constructed from the knowledge of experts in that field and uses inference procedures to solve problems.

From the foregoing, it is evident that expert systems can be used as decision support systems. Decision support systems that have been used in the past years include Aaphelp, Internist I. Mycin, Emycin, Casnet/Glaucoma, Pip, Isabel and so on. Mycin was used in the 1970s to provide consultative advice on diagnosis and therapy for infectious diseases of the blood. These systems have helped health practitioners to diagnose diseases faster and with more accuracy (Uzoka *et al,* 2016).

Analytic Hierarchy Process (AHP) was developed in the 1970s by Thomas L. Saaty. It is a technique for organizing and analysing complex decisions, based on mathematics and psychology. It provides a framework for making better decisions by decomposing a decision problem into a hierarchy of more easily comprehended sub-problems, each of which can then be analysed independently. By reducing complex decisions to a series of pairwise comparisons, and then synthesizing the results, AHP helps to capture both subjective and objective aspects of a decision. In addition, the AHP incorporates a useful technique for checking the consistency of the decision maker's evaluations, thus reducing the bias in the decision-making process (Saaty, 1980).

Fuzzy logic is an information processing technique that uses linguistic variables (such as slight, heavy, and so on) that are vague to process data. It allows for the representation of partial truth. Fuzzy system is useful when an exact solution is not necessary but an approximate and fast solution is needed. Fuzzy logic also becomes necessary when the inputs to our problem are vague, ambiguous or not known at all. Fuzzy logic was first proposed by Lotfi A. Zadeh of the University of California at Berkeley in 1965 (Dadios, 2012).

Fuzzy AHP (FAHP) is an extension of the classical AHP method that was developed to accommodate the acknowledged possible uncertainty in the subjective judgements made in AHP. In most real-world problems, some of the decision data can be precisely assessed while others cannot. Humans are unsuccessful in making quantitative predictions, whereas they are comparatively efficient in qualitative forecasting. Essentially, the uncertainty in the preference judgments gives rise to uncertainty in the ranking of alternatives as well as difficulty in determining the consistency of preferences. Despite the convenience of AHP in handling both quantitative and qualitative criteria of multi-criteria decision-making problems based on decision maker's judgments, fuzziness and vagueness existing in many decision-making problems may contribute to the imprecise judgments of decision makers in conventional AHP approaches. Thus the need for Fuzzy-AHP.

Medical informatics is the scientific field that deals with the application of computing systems, devices and methods to optimize the storage, retrieval and management of biomedical information for problem-solving and decision-making. In simple form, medical informatics, or bioinformatics, is the integration of biology and technology. In this field, computers are used to analyse biological information using statistical techniques and algorithms. Medical Informatics harnesses the power of information technology to expedite the transfer and analysis of data, leading to improved efficiencies and knowledge. The field also interfaces with other fields such as the clinical sciences, computer sciences, library sciences and public health sciences, to mention a few. Several fields in medical informatics include quality management, patient care, patient safety, medical education, disease management, evidence-based care, pharmacy, laboratory radiology and medical record (Hoyt *et al.*, 2009).

Android is a complete, open source software stack based on the Linux kernel and designed for mobile devices. It is championed by Google and owned by Open Handset Alliance. The Android Software Development Kit (SDK) 1.0 was released in 2008 (Gargenta, 2011). The SDK is a collection of tools required for developing Android applications. Android includes an operating system, a middleware, some key applications and a set of API libraries for writing mobile applications. The operating system is based on the Linux Kernel. The middleware is the software that interfaces the operating system with higher level applications such as user applications. It is partly written in Java. Android applications are written in Java. Android also offers access to sensors and devices such as camera, accelerometer, compass, Bluetooth and Global Positioning System (GPS) to mention a few. Today, the Android Operating System has grown to be one of the most popular operating systems in the world.

Tropical diseases, according to the World Health Organization (2016a), are infectious diseases that thrive in hot, humid conditions. Examples include malaria, river blindness, Lassa fever, Ebola, haemorrhagic fever, tuberculosis, leishmaniasis, schistosomiasis, African trypanosomiasis and dengue. These diseases are common in Nigeria (a tropical country) due to the prevalence of disease carriers like the mosquito. The diagnosis of tropical diseases involves a measure of uncertainty. This is as a result of the way humans feel and express their symptoms. For example, a patient might say that he feels a slight headache. Therefore, it is inaccurate to use Boolean logic to model such a problem domain. This makes Fuzzy Logic more applicable to modelling the diagnosis of tropical diseases.

1.2 STATEMENT OF PROBLEM

According to the World Health Organization (2016b), Nigeria has a physician density of 0.376 per 1000 persons. This means that approximately one doctor is available for every 2,660 persons. There are an estimated 100 million malaria cases with over 300,000 deaths per year in Nigeria (United States Embassy in Nigeria, 2011). These scanty doctors have to attend to many of these cases. Consequently, the doctors are stressed and cannot perform at their best. When people go to the hospitals to be treated, they meet long queues that discourage them and lead to self-medication. Therefore, there is the need to reduce patients' waiting time and the work overload on the medical practitioners.

Patel & Patel (2017) defined patient waiting time as the length of time from when the patient enters the out-patient clinic to the time the patient actually leaves the OPD. The cause of waiting time was identified as shortage of health workers by Oche & Adamu (2013). Obamiro (2013) also attribute long queues at health centers to few Medical Doctors. He went on to state that waiting time is one of the key predictors of patient satisfaction and therefore very useful to evaluate the efficiency of a health care system. He explained that a hospital in Nigeria, by the name Medicare has adopted the use of information technology in their operation with a reduction in patient waiting time.

According to Uzoka *et al.* (2016), another major challenge in the diagnosis of tropical diseases is that a number of tropical diseases present with confusable symptoms making accurate diagnosis very difficult. An example is the frequent confusion in diagnosing Malaria and Typhoid fever. In addition, most people view health cost as a luxury and resort to self-diagnosis and self-medication. In some places, people travel very long distances before they can get quality health care. In such cases, they would rather stay back at home and use locally made herbs for a disease that has been diagnosed wrongly by themselves.

The use of decision support systems, like the system developed can solve the problems identified. Application of the system will reduce the workload on doctors by shortening consultation time required for diagnosis. The system will increase the efficiency of medical practitioners by preventing the human error that results from confusable symptoms of tropical diseases. The system will greatly reduce patient waiting time consequently reducing the frustration that results when patients meet with long queues at the health centres.

1.3 AIM AND OBJECTIVES

The aim of this project is to develop an Android-based application for diagnosing selected tropical diseases using Fuzzy-Analytic Hierarchy Process

The objectives are:

1. To design an expert system for diagnosing selected tropical diseases using Fuzzy-AHP.
2. To implement the designed system on the Android mobile platform.
3. To evaluate the performance of the system implemented.

1.4 SCOPE OF STUDY

In the project, an Android application that uses fuzzy logic and Analytic Hierarchy Process to diagnose some tropical diseases was developed. The tropical diseases considered in the project are Malaria, Tuberculosis, Pneumonia and Diarrhoea. Expert Knowledge was gathered using questionnaires administered to health practitioners in the Ekiti State University Teaching Hospital (EKSUTH) Ado Ekiti, Federal Teaching Hospital, Ido-Ekiti (FETHI), Obafemi Awolowo University Teaching Hospital (OAUTH), Ile-Ife and University College Hospital (UCH), Ibadan. The knowledge gathered was used to develop the knowledge base for the expert system.

1.5 SIGNIFICANCE OF STUDY

The Android Operating System (OS) is very popular. Even those who view yearly medical check-ups as a luxury have one or more Android phones. If this project is able to develop the expert system on the Android O.S., it will bring cheap and effective medical care to the masses. This system will ease the pressure on our few medical practitioners and facilities.

The project will document expert knowledge on the selected tropical diseases in a corpus that is readily available to be used for further study. It will also encourage other studies to develop new expert systems or port existing ones to the Android platform thereby making them available to more people. The developed system will be applicable in health centres situated in the rural areas with little or no network coverage because it does not require internet access.

CHAPTER TWO

LITERATURE REVIEW

2.1 ARTIFICIAL INTELLIGENCE

Artificial Intelligence is a discipline that took form in the 1950s. About this time, Turing defined intelligent behaviour as the ability to achieve human-level performance in all cognitive tasks, sufficient to fool an interrogator. What he meant is that the computer should be interrogated by a human through a teletype, and if the interrogator cannot tell if it is a human or a computer at the other end, then the computer can be said to be intelligent (Russell & Norvig, 1995).

Jones (2008) defined intelligence as a set of properties of the mind which include the ability to plan, solve problems, and in general, reason. The author also provided a simpler definition of intelligence as the ability to make the right decision given a set of inputs and a variety of possible actions. Artificial Intelligence currently encompasses a huge variety of subfields, from general-purpose areas such as perception and logical reasoning to specific tasks such as playing chess, proving mathematical theorems, writing poetry and diagnosing diseases.

2.2 EXPERT SYSTEM

Expert systems are computer programs that use domain-specific knowledge to emulate the reasoning process of human experts. They are capable of offering solutions to specific problems in a given domain or which are able to give advice, both in a way and at a level comparable to that of experts in the field. Building expert systems for specific application domains has even become a separate subject known as knowledge engineering. Expert systems are mostly designed to function in problem domains that require considerable human expertise. Examples of such problem domains are medical diagnosis, finance, product design and so on (Lucas & van der Gaag, 1991).

Expert systems are designed to solve complex problems by reasoning about knowledge, like an expert, and not by following the procedure of a developer as is the case in conventional programming. An expert system has a unique structure, different from conventional programs. It is divided into two parts, one fixed, independent of the expert system: the inference engine, and one variable: the knowledge base. To run an expert system, the inference engine reasons about the knowledge base like a human.

Knowledge acquisition is the process of extracting, structuring, and organizing knowledge from several knowledge sources, usually human experts so that the problem-solving expertise can be captured and transformed into a computer-readable form. Knowledge is the most important component of expert systems. The captured knowledge forms the basis for the reasoning process of an expert system. Without explicitly represented knowledge, an expert system is no more than a computer program.

2.2.1 The Basic Components of Expert System

A typical Expert System consists of the user interface, knowledge base and inference engine. The Knowledge Base is expressed with natural language as "IF... THEN ..." rules, referred to as production rules. A production rule is a conditional statement that indicates under which condition a particular conclusion may be drawn. This is the sub-system that represents human expertise about a specific problem area in the form of facts and heuristics. It is important to clarify between knowledge and data. Isolated data points do not become knowledge until they are analysed and summarised. According to Conejar and Kim (2014), knowledge is a theoretical or practical understanding of a subject or a domain. Knowledge is the sum of what is currently known.

The Inference Engine is a computer program designed to produce a reasoning on rules. In order to produce a reasoning, it should be based on logic. There are several kinds of logic: propositional logic, epistemic logic, modal logic, fuzzy logic and others. The engine has two ways to run: batch or conversational. In batch, the expert system has all the necessary data to process from the beginning. The conversational method becomes necessary when the developer knows he cannot ask the user for all the necessary data all at once, the problem being too complex. The software gradually requests the missing data from the user. The result gives the impression of a dialogue led by an expert. To guide a dialogue, the engine may have several levels of sophistication: "forward chaining", "backward chaining" and "mixed chaining".

The process of knowledge acquisition determines the performance of the expert system. Therefore, great attention should be paid to this process. Several knowledge acquisition techniques and tools have been developed for use. They include: interviewing, structured interviewing (questionnaire), observations, protocol analysis and repertory grid analysis.

The user interface is the means through which the system's end users interact with the system. The User Interface should include menu systems, an interactive graphics facility and explanation facilities to make the system user-friendly. The relationship between these components is illustrated in figure 2.1.

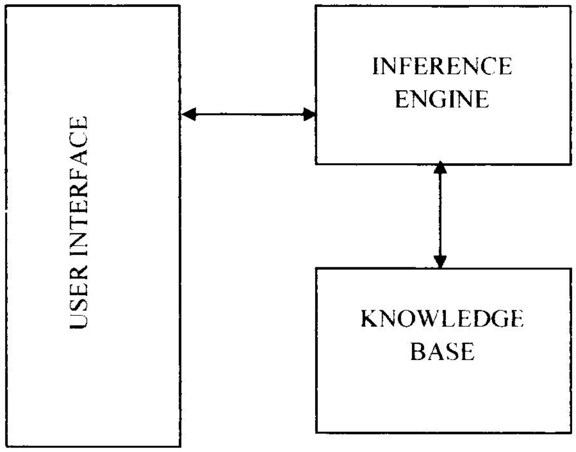


Figure 2.1: The basic components of an expert system.

2.3 MEDICAL DECISION SUPPORT SYSTEMS (MDSS)

Medical Decision Support Systems (MDSS) are computer systems designed to assist or support healthcare personnel in making clinical decisions. They can help healthcare personnel to organize, store and apply the exponentially increasing volume of medical knowledge thereby improving the quality of medical care (Conejar & Kim, 2014).

Attempts to create decision support tools for medical diagnosis started in the 1950s. Research into the use of Artificial Intelligence for medical diagnosis did not start until the early 1970s (Uzoka *et al*, 2016). Over the years, several systems have been developed and tested. Some of these systems are discussed in this section. According to Shortliffe *et al.* (1979), the goals for developing expert systems for medicine are as follows:

1. To improve the accuracy of clinical diagnosis through approaches that are systematic, complete, and able to integrate data from diverse sources.

2. To improve the reliability of clinical decisions by avoiding unwarranted influences of similar but not identical cases.
3. To improve the cost efficiency of tests and therapies by balancing the expenses of time, inconvenience against benefits, and risks of definitive actions.
4. To improve our understanding of the structure of medical knowledge, with the associated development of techniques for identifying inconsistencies and inadequacies in that knowledge.
5. To improve our understanding of clinical decision-making, in order to improve medical teaching and to make the system more effective and easier to understand.

Some Medical decision support systems that have been used in the past are discussed below.

2.3.1 AAPHelp

This system was developed at Leeds University to support the diagnosis of acute abdominal pain and the need for surgery. The decision-making process was based on the naïve Bayesian approach. The system's overall diagnostic accuracy was 91.8 % which was significantly higher than the accuracy of the members of the clinical team at 79.6% (Jaiswal & Sarode, 2015).

2.3.2 INTERNIST I

It is a rule-based expert system. It was designed in the University of Pittsburgh in 1974 for the diagnosis of complex problems in general internal medicine. It is based on a tree-structured database that links diseases with symptoms. The drawback of the system includes the cost of running the program and maintaining the database and introducing some model of disease evolution in time. (Jaiswal & Sarode, 2015).

2.3.3 MYCIN

MYCIN was developed at Stanford University. It is a rule-based expert system designed to diagnose and recommend treatment for certain blood infections. Eventually, it was extended to handle other infectious diseases. Its knowledge base is represented as a set of "IF-THEN" rules with certainty factors attached to diagnoses.

2.3.4 CASNET

CASNET (Causal Associational Network) is a general tool for building expert systems for the diagnosis and treatment of diseases. It was used in developing several systems, one of which is CASNET/Glaucoma used for the diagnosis and treatment of glaucoma.

2.3.5 ONCOCIN

It is a rule-based expert system for oncology protocol management. It was designed to assist physicians with the treatment of cancer patients receiving chemotherapy. It uses a customized flowchart language and takes into consideration the history of past events and the duration of actions.

2.3.6 DENDRAL

The development of DENDRAL started at Stanford University in the 1960s. The HEURISTIC DENDRAL system offers assistance in the field of organic chemistry in determining the structural formula of a chemical compound that has been isolated from a given sample. In determining a structural formula, information concerning the chemical formula, the source the compound has been taken from, as well as information that has been obtained by subjecting the compound to physical, chemical and spectrometric tests are used. The method employed is called generate-and-test. The system first generates all plausible molecular structures as hypotheses and test each against the observed data (Lucas & van der Gaag, 1991).

A summary of the Medical Decision Support Systems discussed is presented in table 2.1.

Table 2.1: Summary of existing Medical Decision Support Systems (MDSS).

NAME	METHOD	APPLICATION AREA
AAPHelp	Naïve Bayesian approach	Diagnosis of abdominal pain
INTERNIST I	Tree structured database	Diagnosis in internal medicine
MYCIN	Rule-based	Diagnosis and treatment of blood infections
CASNET/Glaucoma	-	Diagnosis of Glaucoma
ONCOCIN	Rule-based	Oncology protocol management
DENDRAL	Generate-and-test	Organic chemistry

2.4 MULTIPLE CRITERIA DECISION ANALYSIS

There are several types of decision analysis methods. One is the Multiple Criteria Decision Analysis. The Multiple Criteria Decision Analysis problem is a decision-making problem in which a wide range of criteria is considered to select from a number of alternative courses of action. Multi-criteria decision analysis (MCDA) methods have been developed to support the decision maker in their unique and personal decision process. MCDA methods provide techniques for finding a compromise solution.

MCDA methods place the decision maker at the centre of the process. They lead to different solutions for every decision maker, but they incorporate subjective information. This subjective information, also known as preference information, is provided by the decision maker, which leads to the difference (Ishizaka & Nemery, 2013). The two main branches of Multiple Criteria Decision-making are Multiple Objective Decision-making (MODM) and Multiple Attribute Decision-making (MADM). The major difference between these two branches is that MODM concentrates on problems with continuous decision spaces while MADM concentrates on problems with discrete decision space. (Zimmerman. 2001).

2.4.1 Types of Decision

1. The choice problem. The goal is to select the single best option or reduce the group of options to a subset of equivalent or incomparable 'good' options. An example is a person who wants to decide on what model of car to buy.
2. The Sorting problem. In this case, options are to be sorted into ordered and predefined groups called categories. Sorting methods are useful for repetitive or automatic use. They can also be used as an initial screening to reduce the number of options to be considered in a subsequent step.
3. The ranking problem. In this case, options are ordered from best to worst by means of scores or pairwise comparisons. An example is the ranking of universities according to several criteria.
4. The descriptive problem. Here, the goal is to describe options and their consequences. This is usually done in the first step to understand the characteristics of the decision problem

2.4.2 Multi-Criteria Decision Analysis Methods

The number of MCDA methods available is enormous. This usually makes it difficult for the decision maker to decide which method is best for the problem at hand.

2.4.2.1 Analytic Hierarchy Process

The Analytic Hierarchy Process (AHP), was developed by Saaty. It is a Multiple Criteria Decision Analysis tool. It is a particularly useful method when the decision maker is unable to construct a utility function.

According to Mu and Pereyra-Rojas (2017), for the Analytical Hierarchy Process to be useful for decision-making, the following steps must be taken to decompose the problem.

1. Develop a model for the decision. In this stage, a hierarchy is developed to analyse the decision. This is in the form of a tree with the first level or the root as the goal of the problem. The second level is constituted by the criteria that will be used to reach the goal. The third level consists of available alternatives. Such a model is illustrated in figure 2.2. In the process of structuring the problem this way, the understanding of the problem increases.

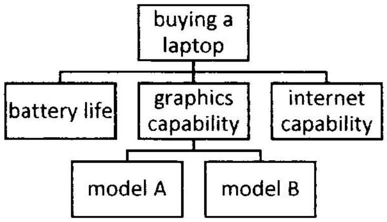


Figure 2.2: A model for decision-making (Mu and Pereyra-Rojas, 2017).

2. Derive priorities (weights) for the criteria. Each criterion in level two of the hierarchy have different importance (or priorities or weight). Saaty suggests using the scale shown in table 2.2 to compare the relative importance of the criteria.
To perform a pairwise comparison of the criteria, a comparison matrix of the criteria involved in the decision is prepared. Each cell in the matrix will have a value from the numerical scale shown above. If criteria A is equally important as B, the A-B comparison cell will contain 1. If criteria A is strongly more important as B, the A-B comparison cell will contain 5. Conversely, the B-A comparison cell will contain 1/5. Also, when a criterion is compared with itself, the comparison cell will contain 1. Table 2.3 is the pairwise comparison matrix of the criteria for buying a computer system for a managing director who travels a lot.
The next step is to calculate the priorities or weights of each criterion. The two methods for doing this are the exact and approximate methods. The approximate method is simpler and will be applied. The first step is to perform normalization of the comparison matrix. This is done by dividing each cell by the total of the column it belongs. The normalized matrix is shown in table 2.4.

From the normalized matrix, the priorities are calculated as the average value of each row. This average is shown in the priority column of table 2.5. The priorities in the table are then adjusted for consistency.

Table 2.2: Priority scale (Mu and Pereyra-Rojas, 2017).

Verbal judgement	Numeric value
Extremely important	9
	8
Very strongly more important	7
	6
Strongly more important	5
	4
Moderately more important	3
	2
Equally important	1

Table 2.3: Pairwise comparison matrix of the criteria.

Buying a laptop	Battery life	Graphics capability	Internet capability
Battery life	1	7	3
Graphics capability	1/7	1	1/3
Internet capability	1/3	3	1

Table 2.4: Normalized matrix.

Buying a laptop	Battery life	Graphics capability	Internet capability
Battery life	0.677	0.636	0.692
Graphics capability	0.097	0.091	0.077
Internet capability	0.226	0.273	0.231

Table 2.5: Priorities of Criteria.

Buying a laptop	Battery life	Graphics capability	Internet capability	Priority
Battery life	0.677	0.636	0.692	0.669
Graphics capability	0.097	0.091	0.077	0.088
Internet capability	0.226	0.273	0.231	0.243

3. Derive local priorities (preferences) for the alternatives. This step derives the relative priorities of the alternatives with respect to each criterion. In the foregoing example, this means that we
    a. Compare model A with model B with respect to battery life
    b. Compare model A with model B with respect to Graphics capability
    c. Compare model A with model B with respect to Internet capability

This results in table 2.6 below. These results are called the local priorities for each alternative.

Table 2.6: Local Priorities of Alternatives.

Alternative	Battery life	Graphics capability	Internet capability
Model A	0.875	0.167	0.100
Model B	0.125	0.833	0.900

4. Derive overall priorities (Model Synthesis). In this step, we calculate the overall priority or final priority of each alternative taking into consideration the weights of each criterion.

To calculate the overall priority, each cell in the table above is multiplied by the weight of the corresponding criteria. Each row is then added up. This gives:

a. The overall priority of model A as 0.624
b. The overall priority of model B as 0.376

Considering the foregoing, model A is preferable (with an overall priority of 0.624) compared to model B.

5. Perform sensitivity analysis. This step allows one to check the robustness of the decision. It is used to determine which criteria determined our results. It is performed by changing the weights of the criteria and seeing how that affects the overall priorities.
6. Making a final decision

2.4.2.2 Analytic Network Process (ANP)

ANP is a generalization of the analytic hierarchy process (AHP) which deals with dependencies. In AHP, as with the other methods presented in this book, we assume that criteria are independent. If they are not independent, correlated criteria would result in an over evaluated weight in the decision. In cases when criteria are dependent. The ANP method allows these dependencies, also called feedbacks, to be modelled. They are closer to reality and, as a result, yield more accurate results.

The ANP problem structure is very similar to that of AHP, with three levels of clusters: goal, criteria and alternatives. The difference is the additional loop over the cluster criteria, which indicates an inner dependency. In addition to the pairwise comparisons in traditional AHP, matrices modelling the inner dependency are required. These matrices aim to capture the relative importance of the criteria when another dependent criterion has already been evaluated. Two software that is popularly used for ANP is Super Decisions and Decision Lens.

2.4.2.3 Multi-Attribute Utility Theory

MAUT is based on the main hypothesis that every decision maker tries to optimize, consciously or implicitly, a function which aggregates all their points of view. This means that the decision maker's preferences can be represented by a function, called the utility function. This function is not necessarily known at the beginning of the decision process, so the decision maker needs to construct it first. The utility function is a way of measuring the desirability or the preference of objects, called alternatives. These can be consumer goods (cars, smartphones, etc.) or services. The utility score is the degree of well-being those alternatives provide to the decision maker. The utility function is composed of various criteria which enable the assessment of the global utility of an alternative.

Considering the foregoing, model A is preferable (with an overall priority of 0.624) compared to model B.

5. Perform sensitivity analysis. This step allows one to check the robustness of the decision. It is used to determine which criteria determined our results. It is performed by changing the weights of the criteria and seeing how that affects the overall priorities.
6. Making a final decision

2.4.2.2 Analytic Network Process (ANP)

ANP is a generalization of the analytic hierarchy process (AHP) which deals with dependencies. In AHP, as with the other methods presented in this book, we assume that criteria are independent. If they are not independent, correlated criteria would result in an over evaluated weight in the decision. In cases when criteria are dependent. The ANP method allows these dependencies, also called feedbacks, to be modelled. They are closer to reality and, as a result, yield more accurate results.

The ANP problem structure is very similar to that of AHP, with three levels of clusters: goal, criteria and alternatives. The difference is the additional loop over the cluster criteria. which indicates an inner dependency. In addition to the pairwise comparisons in traditional AHP, matrices modelling the inner dependency are required. These matrices aim to capture the relative importance of the criteria when another dependent criterion has already been evaluated. Two software that is popularly used for ANP is Super Decisions and Decision Lens.

2.4.2.3 Multi-Attribute Utility Theory

MAUT is based on the main hypothesis that every decision maker tries to optimize, consciously or implicitly, a function which aggregates all their points of view. This means that the decision maker's preferences can be represented by a function, called the utility function. This function is not necessarily known at the beginning of the decision process, so the decision maker needs to construct it first. The utility function is a way of measuring the desirability or the preference of objects, called alternatives. These can be consumer goods (cars, smartphones, etc.) or services. The utility score is the degree of well-being those alternatives provide to the decision maker. The utility function is composed of various criteria which enable the assessment of the global utility of an alternative.

2.4.2.4 MACBETH

MACBETH means 'Measuring Attractiveness by a Categorical Based Evaluation Technique'. From a user point of view, MACBETH has many similarities with AHP. A novice may even not see the difference. Both methods are based on pairwise comparisons entered by the user, but MACBETH uses an interval scale while AHP adopts a ratio scale. The calculation process behind AHP is also different from MACBETH (Ishizaka & Nemery, 2013).

2.4.2.5 PROMETHEE

PROMETHEE means 'Preference Ranking Organization METHod for Enriched Evaluation'. Thus the PROMETHEE method will provide the decision maker with a ranking of actions (choices or alternatives) based on preference degrees. A preference degree is a score (between 0 and 1) which expresses how an action is preferred over another action, from the decision maker's point of view. A preference degree of 1 thus means a total or strong preference for one of the actions on the criterion considered. If there is no preference at all, then the preference degree is 0. On the other hand, if there is some preference but not a total preference, then the intensity will be somewhere between 0 and 1.

2.4.2.6 ELECTRE

The *ELimination Et Choix Traduisant la REalite* (elimination and choice expressing reality) methods, referred to as ELECTRE, belong to the outranking methods. The outranking methods are based on pairwise comparisons of the options. This means that every option is compared to all other options. Based on these pairwise comparisons, final recommendations can be drawn. The main characteristic and advantage of the ELECTRE method is that they avoid compensation between criteria and any normalization process, which distorts the original data (Ishizaka & Nemery, 2013).

2.4.2.7 TOPSIS

TOPSIS is the acronym for 'Technique of Order Preference Similarity to the Ideal Solution'. The TOPSIS method requires only a minimal number of inputs from the user and its output is easy to understand. The only subjective parameters are the weights associated with the criteria. The fundamental idea of TOPSIS is that the best solution is the one which has the shortest distance to the ideal solution and the furthest distance from the anti-ideal solution. The TOPSIS method is based on five computation steps. The first step is the gathering of the performances of the alternatives on the different criteria. These performances need to be normalized in the second step. The normalized scores are then weighted and the distances to an ideal and anti-ideal point are calculated. Finally, the closeness is given by the ratio of these distances.

Two other Multi-Criteria Decision Analysis methods worth mentioning are goal programming and Data Envelopment Analysis (DEA). Table 2.7 below shows a summary of some MCDA methods and the type of decision problem they solve.

Table 2.7: Summary of Multi-criteria decision methods (Ishizaka & Nemery, 2013).

Choice problems	Ranking problems	Sorting problems	Description problem
AHP	AHP	AHPSort	-
ANP	ANP	-	-
MAUT/UTA	MAUT/UTA	UTADIS	-
MACBETH	MACBETH	-	-
PROMETHEE	PROMETHEE	FlowSort	GAIA, FS-Gaia
ELECTRE I	ELECTRE III	ELECTRE -Tri	-
TOPSIS	TOPSIS	-	-
Goal Programming	-	-	-
DEA	DEA	-	-

Some software has also been developed for applying the MCDA methods. Table 2.8 shows a few of those software and the MCDA method they support.

Table 2.8: Software tools for MCDA problems (Ishizaka & Nemery, 2013).

MCDA Methods	Software
PROMETHEE-GAIA	Decision Lab, D-Sight, Smart Picker Pro, Visual Promethee
PROMETHEE	DECERNS
ELECTRA	Electre IS, Electre III-IV
UTA	Right Choice, UTA+, DECERNS
AHP	MakeltRational, ExpertChoice, Decision Lens, HIPREE 3+, RightChoicesDSS, Criterium, EasyMind, Questfox, ChoiceResults, 123SHP, DECERNS
ANP	Super Decisions, Decisions, Decision Lens
MACBETH	M-MACBETH
TOPSIS	DECERNS
DEA	Win4DEAP, Efficiency Measurement System, DEA Solver Online, DEAFrontier, DEA-Solver PRO, Frontier Analyst
FlowSort – FS-GAIA	Smart Picker Pro
ELECTRE-Tri	Electre Tri, IRIS

2.5 FUZZY LOGIC

Fuzzy logic is a form of logic that is governed by fuzzy set theory as against conventional set theory. Fuzzy logic allows for imprecision and incompleteness of data. Fuzzy logic allows for imprecision in that it allows a varying degree of truth as against Boolean logic which only allows two degrees of truth (true and false). In the real world, problems are very rarely true or false. That is the origin of the cliché that things are rarely so black and white. The natural world is filled with variables that are continuous and takes on several values between two extremes. Our understanding of most physical processes is based largely on imprecise human reasoning.

Fuzzy logic can be used to solve very complex problems that cannot be solved within a reasonable time with the current computers. Take the "travelling salesman" problem (Ross, 2010) as an example. A sales representative wants to minimize the total distance travelled by considering various itineraries and schedules between a series of cities on a particular trip. If the number of cities involved is very small, finding an optimal route is very easy. As the number of cities grows, the complexity of the problem grows astronomically. For example, for 100 cities there are about 10^{200} possible routes to consider. No computers exist today that can solve this problem through the brute-force enumeration of every possible route. An approximate solution through fuzzy logic can allow such problems to be solved within reasonable time limits.

Fuzzy logic is the computing technique of choice in the field of medical diagnosis due to its capability to make decisions in an environment of imprecision, uncertainty and incompleteness of information. In addition, another advantage of choosing fuzzy logic is due to the fact that fuzzy logic closely resembles the process of human decision-making and it has the ability to work from approximate reasoning and ultimately find a precise solution. Fuzzy logic is a logically consistent way of reasoning that can cope with uncertainty, vagueness and imprecision inherent in medical diagnosis (Jaiswal & Sarode, 2015). Fuzzy systems are very useful in situations involving highly complex systems whose behaviours are not well understood and in situations where an approximate, but fast solution is warranted. (Ross, 2010).

2.5.1 Fuzzy Set Membership

A fuzzy set is a set containing elements that have varying degrees of membership in the set. Fuzzy set was introduced by Lofti A. Zadeh (1965). Fuzzy sets contain objects that satisfy imprecise properties of membership, that is, membership of an object in a fuzzy set can be approximate. Conversely, classical sets contain objects that satisfy precise properties of membership (Ross, 2010). In a classical set, an object can either be a member or not. There are no middle grounds. For example, the set of heights from 5 to 7 feet is precise; the set of heights in the region around 6 feet is imprecise, or fuzzy. Figure 2.3 below illustrate the difference between a classical set and a fuzzy set using the given example.

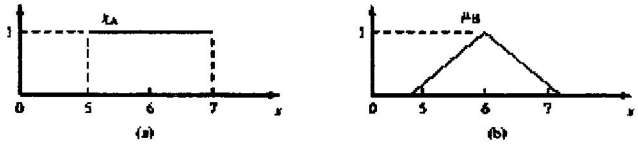


Figure 2.3: Comparison between a classical set (a) and a fuzzy set (b) (Zimmerman, 2001).

The mathematical representation of membership in a fuzzy set is shown in equation 2.1 below. It indicates that an element x in a fuzzy set B can have a degree of membership ranging from 0 to 1.

$$\mu\underset{\sim}{\mathrm{B}}(x) \in [0,1] \tag{2.1}$$

2.5.2 Fuzzy Set Operations

Union $\qquad \mu\underset{\sim}{\mathrm{B}} \cup \underset{\sim}{\mathrm{D}}(x) = \mu\underset{\sim}{\mathrm{B}}(x) \vee \mu\underset{\sim}{\mathrm{D}}(x)$

Intersection $\qquad \mu\underset{\sim}{\mathrm{B}} \cap \underset{\sim}{\mathrm{D}}(x) = \mu\underset{\sim}{\mathrm{B}}(x) \wedge \mu\underset{\sim}{\mathrm{D}}(x)$

Complement $\qquad \mu\underset{\sim}{\mathrm{B}}'(x) = 1 - \mu\underset{\sim}{\mathrm{B}}(x)$

De Morgan's principles for classical sets also hold for fuzzy sets. The only two axioms that hold for classical sets but do not hold for fuzzy sets are the excluded middle axioms. They are shown below in equations 2.2 and 2.3.

$$\underset{\sim}{\mathrm{B}} \cup \underset{\sim}{\mathrm{B}}' \neq X \tag{2.2}$$

$$\underset{\sim}{\mathrm{B}} \cap \underset{\sim}{\mathrm{B}}' \neq X \tag{2.3}$$

2.5.3 Fuzzy Inference System

Fuzzy inference refers to the mapping from a given input to an output using fuzzy logic and involves the application of Membership Functions, Logical Operations and If-Then Rules (Mathworks, 2012). A Fuzzy Inference System is a system that does fuzzy inference. Several types of fuzzy inference systems can be distinguished, to include Mamdani inference system, Sugeno inference system, Larsen inference system and Tsukamoto fuzzy inference system.

2.5.4 Components of a Fuzzy Inference System

2.5.4.1 Rule Base

The mapping of the inputs to the outputs for a fuzzy system is in part characterized by a set of condition → action rules, in the form of IF antecedent THEN consequent. In general, the input to an if-then rule is the current value for the input variables (such as the intensity of each symptom) and the output is an entire fuzzy set. Batch Least Squares (BLS), Recursive Least Squares (RLS), Gradient Method (GM), Learning From Example (LFE), Modified Learning From Example (MLFE), and Clustering Method (CM) are some of the algorithms available for developing a fuzzy model. The LFE technique generates a rule-base for a fuzzy system by using numerical data from a physical system and possibly linguistic information from a human expert (Ross, 2010).

2.5.4.2 Membership Function

A membership function is a mathematical equation that maps elements of a universe of discourse to their corresponding membership values. It is a curve that defines how each point in the input space is mapped to a membership value (or degree of membership). A few standard membership functions are discussed below.

Triangular Membership Function. The triangular membership function has the shape of figure 2.4. It can be represented by either of the two equations (2.5 and 2.6). Equation 2.6 being more succinct and easy to code into a program.

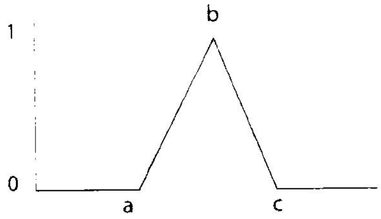


Figure 2.4: Triangular membership function

$$\mu(X) = \left\{ \begin{matrix} 0\, if\ x \le a \\ \frac{x-a}{b-a} if\ a \le x \le b \\ \frac{b-x}{c-b} if\ b \le x \le c \\ 0\, if\ c \le x \end{matrix} \right\} \qquad (2.5)$$

$$trimf\ (x; a, b, c) = \max(\min\left(\frac{x-a}{b-a}, \frac{c-x}{c-b}\right), 0) \qquad (2.6)$$

Trapezoidal membership function. The trapezoidal membership function has the shape of figure 2.5. It can be represented by either of the two equations (2/7 and 2.8) below. Equation 2.8 being more succinct and easy to code into a program.

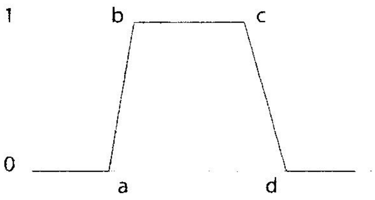


Figure 2.5: Trapezoidal membership function.

$$\mu(X) = \begin{Bmatrix} 0 & if\ x \leq a \\ \frac{x-a}{b-a} & if\ a \leq x \leq b \\ 1 & if\ b \leq x \leq c \\ \frac{d-x}{d-c} & if\ c \leq x \leq d \\ 0 & if\ d \leq x \end{Bmatrix} \tag{2.7}$$

$$trapmf\ (x; a, b, c, d) = \max(\min\left(\frac{x-a}{b-a}, 1, \frac{d-x}{d-c}\right), 0) \tag{2.8}$$

Gaussian Membership Function. It has the shape of a normal distribution curve. It is represented mathematically by the equation 2.9 below.

$$gaussmf\ (x; a, b, c) = e^{-\frac{1}{2}\left(\frac{x-c}{\sigma}\right)^2} \tag{2.9}$$

Other membership functions worth identifying are generalized bell and sigmoid.

2.5.5 Stages of Fuzzy Inference

Fuzzy inference process comprises of five stages. They are fuzzification, composition, implication, aggregation and defuzzification. These stages are shown in figure 1.6.

Fuzzification. Fuzzification is the process of converting a crisp input quantity to a fuzzy value. It is done by applying the membership functions to each input to determine its degree of truth for each rule premise.

Composition. Once fuzzification is completed for a particular rule, composition is required if the antecedent of the rule has more than one part. The fuzzy operators (AND and OR) are then applied to obtain one number that represents the result of the rule antecedent. This number is applied to the output function. Therefore, the input to the composition process is two or more membership values from fuzzified input variables, while the output is a single truth value.

Aggregation. Rule-based systems generally contain more than one rule, each rule producing its own consequent. The process of obtaining the overall consequent from the individual consequents contributed by each rule in the rule-base is known as aggregation of rules. The input of the aggregation process is the list of truncated output functions returned by the implication process for each rule. The output of the aggregation process is one fuzzy set for each output variable. As long as the aggregation method is commutative, then the order in which the rules are executed is unimportant.

An aggregation operator combines the results of the individual rules, individual rule-based inference (functional approach), or composition-based inference (relational approach). Aggregation operations on fuzzy sets are operations by which several fuzzy sets are combined in a desirable way to produce a single fuzzy set. Common aggregation operations are averaging operations and ordered weighted averaging operations.

Implication. This is the process of calculating the relevance of each rule to the current input values. The implication of a rule ranges from 0 to 1. An implication of 0 means that the rule is not relevant to the current input values and therefore has no effect on the final result of the system. An implication of 1 means that the rule is most relevant to the current input values.

Defuzzification. There may be situations where the output of a fuzzy process needs to be a single scalar quantity as opposed to a fuzzy set. Defuzzification is the conversion of a fuzzy quantity to a precise quantity. The input for the defuzzification process is a fuzzy set (the aggregate output fuzzy set) and the output is a single number.

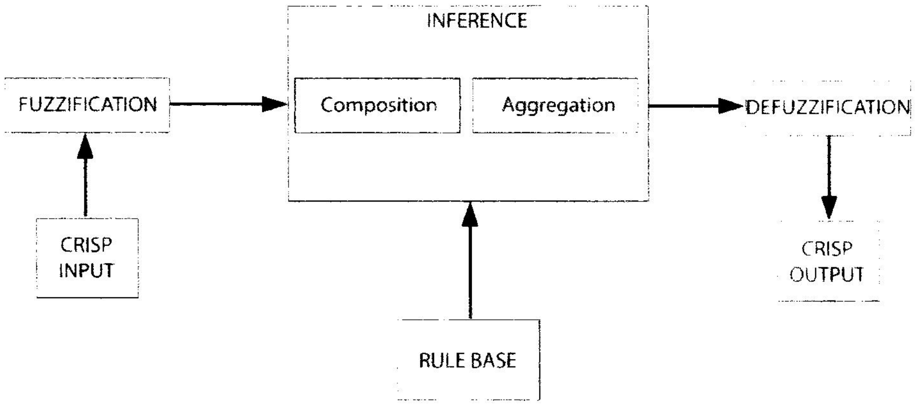


Figure 1.6: Stages and components of a fuzzy system

There are several methods of defuzzification. They include:

Extreme Value Strategies: These fuzzification methods use extreme values of the membership function (generally the maxima) to define the crisp equivalent value. In this method group, we have Left of Maximum (LOM), Right of Maximum (ROM) and Centre of Maximum (COM).

Centroid Strategies (Area Methods): The information taken into account in the above strategies is very limited. If more information shall be considered, which is available via the membership function of the fuzzy set to be defuzzified, then one normally resorts to centroid strategies. The most popular methods are Centre of Area (COA) and Centre of Gravity (COG).

The Centre of Area (COA) method chooses the control action that corresponds to the centre of the area with membership greater than zero. The Centre of gravity (COG) method is the most trivial weighted average and has a distinct geometrical meaning that is the centre of gravity or centre of mass. From a mathematical point of view, the COG corresponds to the expected value of probability. It is represented mathematically as shown below:

$$u_{COG} = \frac{\int u.\mu(u)du}{\int \mu(u)du} \qquad (2.10)$$

The two methods described above have their limitations. This resulted in the development of two other methods: Extended Centre of Gravity (XCOG) and Extended Centre of Area (XCOA). These methods are only worthy of note in this work but are discussed in details in (Zimmerman, 2001). The author did a comprehensive write-up comparing the methods that have been mentioned, stating their comparative strength and weaknesses.

2.6 HYBRIDIZATION OF FUZZY LOGIC AND ANALYTIC HIERARCHY PROCESS

Tang and Beynon (2005) applied the Fuzzy Analytic Hierarchy Process to solve a capital investment problem. A small rental company wanted to decide on which type of car they would adopt. The first step was to identify the criteria for decision-making which were: equipment, comfort, safety, image and price. The numbers of cars identified were five: Proton Persona, Honda New Civic, Vauxhall Merit, Volkswagen Polo and Daewoo Lanos. The linguistic variables used to make the pairwise comparisons were those associated with the standard 9-unit scale shown in table 3 above. Tang and Beynon decided to fuzzify with Triangular Fuzzy Numbers (TFNs) because of their computational simplicity

2.7 BIOMEDICAL INFORMATICS

Biomedical informatics is the field that deals with the application of computers, information and communication technology and systems to the storage, retrieval, sharing and management of biological information to deliver quality healthcare. Medical Informatics emphasizes information brokerage; the sharing of a variety of information back and forth between people and healthcare entities. Examples of medical information that needs to be shared are laboratory test results, x-ray results, vaccination status, medication allergy status, consultant's notes and hospital discharge summaries.

Medical Informatics harnesses the power of information technology to expedite the transfer and analysis of data, leading to improved efficiencies and knowledge. The field also interfaces with other fields such as the clinical sciences, computer sciences, library sciences and public health sciences, to mention a few. Information Technology has provided support for many medical activities. Figure2.7 shows some medical activities that medical informatics have provided support for. This results in more efficient health care systems, migration from paper-based records, reduction in medication errors and easier retrieval of educational and patient related information. According to Hoyt *et al.* (2009), some of the forces driving the adoption of health information technology are cost reduction, improved patient safety and improved quality of health care.

Mobile technology has its origin in the early 1990s. The Apple Newton appeared around this period but did not succeed due to its cost and physical size. Subsequently, Jeff Hawkins invented the Palm Pilot 1000 in 1994. Portable devices became popular with the medical profession when the Epocrates application was released in 1999. Due to this application, drug facts could be retrieved much more rapidly with the PDA compared to the Physician Desk Reference (PDR). According to Kho, Henderson & Dressler (2006), 50 per cent of practising physicians and about 60-70 per cent of medical students used PDAs on a regular basis. Figure 2.8 shows the increasing popularity of PDAs with physicians

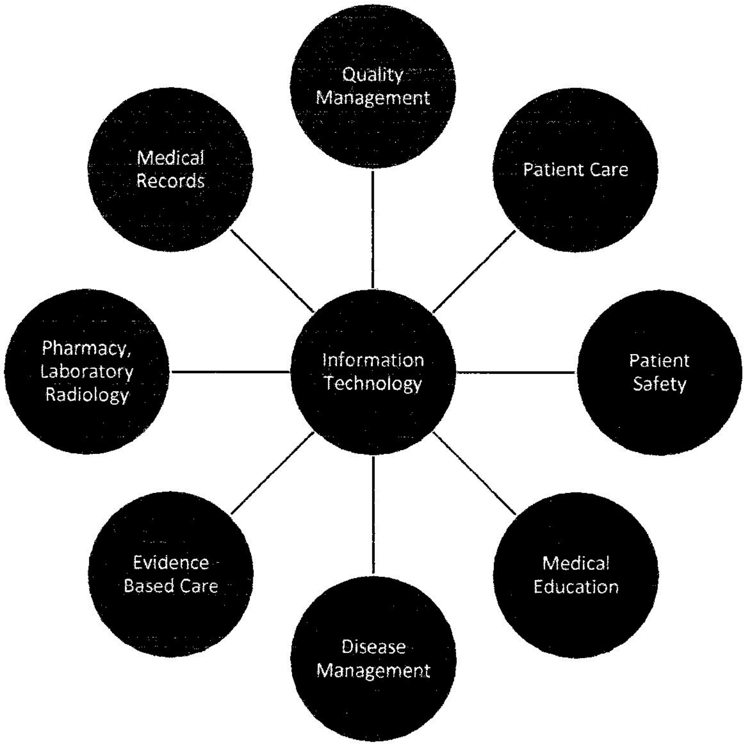


Figure 2.7: Healthcare functions and information technology (Hoyt *et al*. 2009).

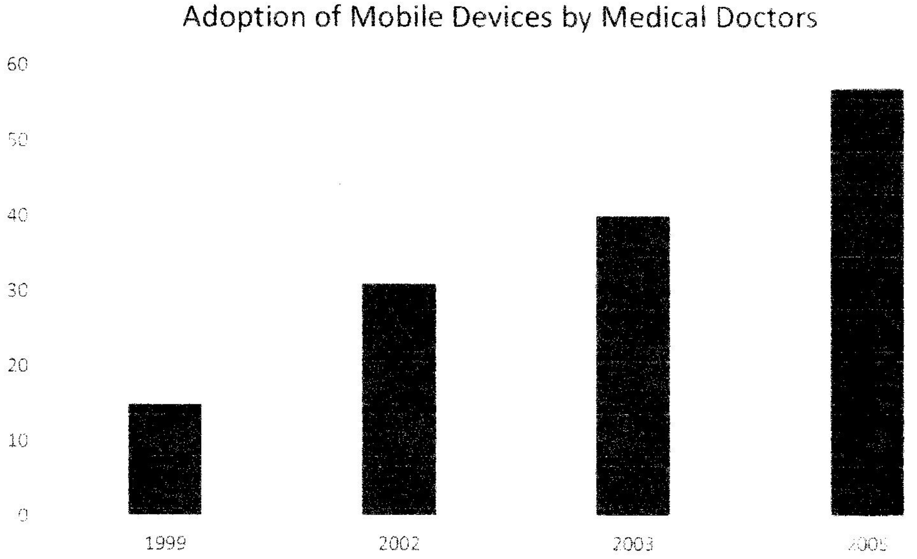


Figure 2.8: Popularity of mobile devices among physicians (Hoyt *et al.* 2009).

Some medical software that has been used was discussed in Hoyt *et al.* (2009). Cardiac Clearance was used for cardiac clearance before surgery. The physician answers a few standard questions and the software indicates if the patient needs further testing prior to going to the operating room. Cholesterol estimates the risk of heart disease in ten years and offers treatment recommendations. Depression is used to calculate the degree of depression. Growth Chart takes a child's age, gender, height and weight and plots how the child compares to normative data. MedCalc includes Intravenous (IV) infusion rate calculators that should improve medication safety. Archimedes is very similar to MedCalc. ABG Pro interprets arterial blood gases. Preg Track tracks pregnancies. Pneumonia Severity Index is a Windows software that calculates the severity and mortality of community-acquired pneumonia. The authors also mentioned that a search for the word "medical" in 2009 returned 617 software for smartphones.

Another interesting area of applying information technology in medicine is in electronic prescribing. Electronic prescribing has replaced handwritten prescription in many countries due to the medical errors that occur from unclear or incomplete prescriptions. An example is found in Hoyt (2009), where the pharmacist interpreted the drug prescribed as Plendil instead of Isordil, due to the doctor's bad handwriting, resulting in the death of the patient. This resulted in the U.S. Institute of Medicine issuing a directive that all prescription should be electronic by the year 2010. Considering the impressive results of biomedical engineering in the past few decades, it is evident that it is a growing field of endeavour with more development to happen in the near future.

2.8. ANDROID

Android is a complete, open-source software stack for mobile devices based on the Linux kernel. The stack includes an operating system, middleware and some key apps. (Smith & Friesen, 2014). Android 1.0 was released by Google in 2008.

2.8.1. Components of an Android Application

An Android Application consists of loosely coupled components, bound using a project manifest that describes each component and how they interact.

2.8.1.1 Activity

An activity is usually a single screen that the user sees on the device at one time and can interact with in order to do something. An activity can be thought of as being a container for the User Interface as well as the code that runs it. Android applications are made up of one or more activities, that is, an Android application must contain at least one activity, but can contain several. An Android application has a main activity, usually, the one that is shown first when you launch the application. For example, an email app might have one activity that shows a list of new emails, another activity to compose an email, and another activity for reading emails. An activity is implemented as a subclass of Activity (Google, 2019a).

2.8.1.2 Services

A service is a component that runs in the background to perform long-running operations or to perform work for remote processes. A service does not provide a user interface. They can perform the same actions as activities but without any user interface. Services are useful for actions that need to be performed, regardless of what is on the screen. For example, a service might play music in the background while the user is in a different app, or it might fetch data over the network without blocking user interaction with an activity. Another component, such as an activity, can start the service and let it run or bind to it in order to interact with it. A service is implemented as a subclass of Service. Services have a much simpler life cycle than activities (Google, 2019b).

2.8.1.3 Content Providers

A content provider manages a shared set of app data. You can store the data in the file system, an SQLite database, on the web, or any other persistent storage location your app can access. Through the content provider, other apps can query or even modify the data (if the content provider allows it). For example, the Android system provides a content provider that manages the user's contact information. As such, any app with the proper permissions can query part of the content provider (such as ContactsContract.Data) to read and write information about a particular person. A content provider is implemented as a subclass of *ContentProvider* (Google, 2019b).

2.8.1.4 Broadcast Receivers

A broadcast receiver is a component that responds to system-wide broadcast announcements. Many broadcasts originate from the system - for example, a broadcast announcing that the screen has turned off, the battery is low, or a picture was captured. Apps can also initiate broadcasts - for example, to let other apps know that some data has been downloaded to the device and is available for them to use. Although broadcast receivers don't display a user interface, they may create a status bar notification to alert the user when a broadcast event occurs. More commonly, though, a broadcast receiver is just a gateway to other components and is intended to do a very minimal amount of work. For instance, it might initiate a service to perform some work based on the event.

A broadcast receiver is implemented as a subclass of BroadcastReceiver and each broadcast is delivered as an Intent object (Google, 2019b).

2.8.1.5 Intent

Intents make up the core message system that runs Android. Intents bind individual components to each other at runtime. Intents are used to start activities, services and broadcast receivers and to communicate among various parts of the Android system. An application can either broadcast an intent or receive an intent. Intents can be viewed as the messengers that request an action from other components. For activities and services, an intent defines the action to perform (for example, to "view" or "send" something) and may specify the URI of the data to act on (Google, 2019b).

2.8.2 Android Application Development

An Android application's functionality is written in Java while the looks can be written in XML (Extensible Mark-up Language) or Java.Java is a programming language developed in the early 1990s by Sun Microsystems which was later acquired by Oracle (Baesens, Backiel & Broucke, 2015). The major characteristic of Java that made it popular is that it is platform independent and portable, that is, it can be executed on different hardware platform and architectures. It can also run on several Operating Systems, thereby earning itself the "write once, run everywhere" title. XML is a subset of Standard Generalized Mark-up Language (SGML). It is a text-based language that can be used to mark up data - that is, add meta-data - in a way that is self-describing.

To develop for Android, the following tools are needed.

1. Android SDK (Software Development Kit).
2. JDK (Java Development Kit). Java Development Kit includes the tools for developing, debugging and monitoring Java applications. The JDK includes a private Java Virtual Machine and the resources needed to finish the development of a java application
3. Gradleware Gradle. This is a build tool that allows you to manage libraries and library projects, run instrumentation tests and create conditional builds.

4. An IDE (Integrated Development Environment). An IDE is an application that offers programmers facilities for developing software. IDEs include tools that support all aspects of software development, including creating, debugging, compiling, and running the code. IDEs check the code for syntax errors. Debugging support allows the programmer to move slowly and methodically through the program to find errors. IDEs also keep track of projects and programs, and the files associated with them, easily organizing your work. The Android Studio is the official IDE for Android Development. Android Studio is a collaboration between JetBrains and Google. It is built on top of JetBrain's IntelliJ.

2.8.3 Android User Interface

The Android User Interface is the part of the application that the user can see and interact with. Android provides a variety of pre-build UI components such as structured layout objects and UI controls for building the graphical user interface for an application. Android also provides other UI modules for special interfaces such as dialogues, notifications, and menus.

Views and Viewgroups

All user interface elements in an Android app are built using View and ViewGroup objects. A View is an object that draws something on the screen that the user can interact with. A ViewGroup is an object that holds other View (and ViewGroup) objects in order to define the layout of the interface. The user interface for each component of an application is defined using a hierarchy of View and ViewGroup objects.

Layout

A layout defines the visual structure for a user interface, such as the UI for an activity or app widget. Two common layouts are the LinearLayout and Relative Layout. Linear Layout arranges its children into a single horizontal or vertical row. Relative Layout allows the specifying the position of a child object relative to the parent or relative to other child objects.

Some views in Android are Buttons, TextView, EditText, CheckBox, RadioButton, DatePicker, ProgressBar, Spinner and ListView.

2.9 TROPICAL DISEASES

The World Health Organization (WHO) lists eight diseases that occur exclusively or especially in the tropics and states that, for all practical purposes, the designation refers to infectious diseases that proliferate in hot and humid weather conditions (WHO, 2016). Some of these diseases are caused by protozoa, such as malaria, leishmaniasis, Chagas' disease and sleeping sickness. Others are caused by worms, including schistosomiasis, onchocerciasis and lymphatic filariasis. One is viral, dengue fever. The eight tropical diseases are transmitted to humans by various means, but always include a vector that is generally a hematophagous insect. Schistosomiasis has no vector, but rather intermediary hosts – snails – that release in water the infectious forms for humans.

Take malaria as a case study. If the current world distribution of malaria is examined, it will be seen that it prevails and has greater incidence in countries located between the tropic of Cancer and the tropic of Capricorn, that is, between latitudes 27° 23' north and south. Exceptions are the Middle Eastern countries, notably Afghanistan, northern India and parts of southern China, where malaria is residual. It is arguable that the underdevelopment and poverty of the tropical region contribute to the proliferation of tropical diseases in this region.

2.10 REVIEW OF RELATED WORKS

Application of Artificial Intelligence to Medical Diagnosis dates back to the 1970s (Uzoka *et al*, 2016). Several research works have been carried out in improving the performance of the expert systems that are used in medical diagnosis. The following review of literature shows some soft-computing technologies that have been applied to the development of Medical Decision Support Systems.

2.10.1 Application of Artificial Neural Network to Medical Diagnosis

Al-Shayea (2011) proposed a feed-forward back propagation neural network-based system for the diagnosis of Nephritis disease and heart disease. The system consisted of three layers: the input, hidden and output layers. The input samples were divided into training, validation and test sets. The training set was used to teach the network. The inputs to the system are symptoms and medical images. The data set contained 120 patients: 90 for training, 30 for testing the network. Neural network toolbox from MATLAB 7.9 was used to evaluate the performance of the proposed networks. The system was very accurate at diagnosing.

2.10.2 Application of Bayesian Network to Medical Diagnosis

Burnside, Rubin & Shachter (2004) improved on a system they had previously designed. This system uses predictive imaging features to determine the likely underlying breast disease by using the standardized lexicon established in breast imaging, the Breast Imaging Reporting and Data System (BI-RADS), which defines mammogram feature distinctions and the terminology used to describe them. The objective of the work was to confirm that the Bayes' Net for mammography is capable of predicting the likelihood of malignancy for microcalcifications on mammography, can predict the likelihood of invasive disease as opposed to in situ changes in order to help guide appropriate surgical management and is well-calibrated to the task of predicting malignancy. They identified 26 diseases of the breast that represent the most likely diagnoses to be made on mammography. Twelve of these diseases are malignant and fourteen are benign.

In the research, the characterization of microcalcifications is of interest. When microcalcifications are identified, the radiologist needs to describe the morphology of the microcalcifications as well as their distribution in the breast. The study included 44 consecutive image-guided biopsies performed for microcalcifications detected and deemed suspicious by radiologists. The radiologist used a web-based interface to input mammography findings and his estimate of the likelihood of malignancy into the Bayes' Net. To construct the belief net and perform inference, the authors used the GeNIe modelling environment developed by the Decision Systems Laboratory of the University of Pittsburgh. They made probability assessments from the medical literature and expert opinion. They obtained pre-test probabilities, the age-specific and risk factor specific distribution of diseases from census data and large randomized trials. They derived many of the joint probabilities from studies of the radiologic/pathologic correlation of individual breast diseases.

The authors showed that the developed expert system has approximately the same ability as a sub-specialist mammographer to discriminate benign and malignant disease. In addition, it also performed as well as a full-time, fellowship-trained mammographer in assessing the significance of microcalcifications.

2.10.3 Application of Fuzzy Logic to Medical Diagnosis

Gorgulu and Akilli (2016) presented a study to assist in decision-making and knowledge management in the Decision Support Systems developed for use in the field of health. In their study, they presented several Fuzzy Logic based Decision Support Systems and their different application in medicine. They also stated that the mean success level of the Fuzzy Logic based Decision Support Systems was ninety per cent. Consequently, they concluded that Fuzzy Logic based Decision Support Systems has been providing a significant contribution to disease diagnosis in the examined studies.

A consultation system to be used for diagnosis by patients with chest pain or discomfort was presented by Al-Hamadani (2016). The system consists of two subsystems. The first one, fuzzy-ontology detector, receives the signs and symptoms entered by the patient as linguistic variables. The fuzzy inputs are processed using fuzzy concept to determine the cause of the Chest Pain. The output of the first subsystem is entered to the second. The second subsystem, the crisp-ontology advisor, which is based on ontology engineering infers the required recommendations and/or treatments to save the patient's life when necessary. Al-Hamadani gathered data through interviews with specialists in the field. Triangular membership function was used for fuzzification. The Fuzzy-Ontology Detector stage uses "IF... THEN..." fuzzy rules for inference. Defuzzification was by the Centre of Gravity (COG) technique. Both the Fuzzy- Ontology Detector and the Crisp-Ontology Advisor were designed using Protégé, an open source ontology editor and knowledge management system.

Obi, Eke, and Osagba (2018) proffered a diagnostic system that will aid medical practitioners in the fast and accurate diagnosis of tuberculosis, early treatment by prescribing appropriate medications based on the patient complaints, the signs and symptoms and isolation of carrier to curtail further spread of the disease. The authors wrote that the traditional method of diagnosis based totally on cultured specimens, the outcome of which takes weeks to achieve, was inefficient. They also stressed the importance of fast and accurate diagnosis to the worldwide control of tuberculosis. They made use of Fuzzy Logic Mining Techniques to model uncertainty inherent in diagnosis and implement the system making use of asynchronous techniques which improves the performance of the system and produce results of diagnosis without delay. The inputs to the system are the patient's complain, signs and symptoms.

A knowledge base deduced from 36 fuzzy rules was used for diagnosis. The fuzzy system had 11 attributes. The inputs made use of the following linguistic variables (mild, moderate, severe and very severe, yes, no) where applicable to describe the membership function.

The proposed system was simulated using MATLAB (R2010a) using the Fuzzy Logic Design Toolbox version 7.10.0. The inputs were fuzzified using the trapezoidal membership function and the output was defuzzified using the Weighted Average method. The system was implemented using Hypertext Preprocessor. MYSQL was used as the database and WAMP server as the server technology. The proposed system makes use of forward chaining reasoning to determine the rules that fire.

Singla (2013) used Prolog to design an expert system that is used to diagnose lung diseases. He proposed a rule-based expert system. The system contains the Knowledge base, the Fact base, the Inference engine, the User Interface, the Explanation module and the Developer Interface. The Knowledge Base contained information about thirty-two lung diseases which are represented as a set of if-then production rules. The Fact Base contained facts which are used to match against the antecedent part of rules stored in the knowledge base. The Inference Engine carries out reasoning by linking the rules with facts and deducing new facts. The User Interface is used to communicate between the user and the expert system. The Explanation Module enables the user to ask the expert system how a particular conclusion is reached and why a specific fact is needed. The Developer Interface is used to modify the knowledge. During testing, the system recorded an accuracy of 70%.

Latifi, Hosseini and Mazinai (2015) introduced a fuzzy inference system (FIS) for diagnosing of acute lymphocytic leukaemia in children. The main components of the fuzzy inference system are fuzzifier, rule-base, inference engine and defuzzifier. They designed the proposed system in MATLAB software. The system was evaluated with a dataset including 100 patients. The proposed system achieved an accuracy of 95%.

Djam *et al* (2011) identified Malaria as the commonest cause of mortality in the tropics. The authors also stated the reason that necessitates medical decision support systems. They pointed out that the task of disease diagnosis and management is complex because of the numerous variables involved. It is made more so because of a lot of imprecision and uncertainties. Patients cannot describe exactly how they feel, doctors and nurses cannot tell exactly what they observe, and laboratories results are dotted with some errors caused either by the carelessness of technicians or malfunctioning of the instrument. Medical researchers cannot precisely characterize how diseases alter the normal functioning of the body. They also argued that medical decision support systems are needed more in tropical countries. In most tropical countries, most of which are developing countries, medical personnel and facilities are not adequate for effective tackling of tropical diseases. In rural areas, medical attention is grossly inadequate. The authors made the decision to apply fuzzy logic in their study because of its capability to make decisions in an environment of imprecision, uncertainty and incompleteness of information. Also, fuzzy logic resembles human decision-making with its ability to work from approximate reasoning and ultimately find a precise solution.

The authors designed a fuzzy expert system for the management of Malaria. The developed system composed of four components which include the Knowledge base, the Fuzzification, the Inference engine and Defuzzification components. The fuzzy inference method employed in the research is the Root Sum Square (RSS). Triangular membership function was used to show the degree of participation of each input parameter. On the basis of domain experts' knowledge, both input and output parameters selected for the research were described with four linguistic variables (mild, moderate, severe and very severe). The defuzzification technique employed by the authors is the Centre of Gravity (CoG). According to them, the approach was adopted because it is computationally simple and intuitively plausible. The fuzzy expert system was designed based on clinical observations, medical diagnosis and the expert's knowledge. They selected 35 patients with malaria and computed the results that were in the range of predefined limit by the domain experts. The fuzzy rules for the system were developed with the assistance of five medical doctors. The system generated a percentage that shows the possibility that the patient has Malaria. The authors were able to show that Fuzzy Systems are effective for diagnosing malaria and will reduce the doctors' workload.

Kadhim, Alam & Kaur (2011) discussed the architecture of a fuzzy expert system that used fuzzy rules to diagnose back pain diseases based on symptoms. The parameters considered to determine back pain diseases include patient's history, Body Mass Index (BMI), age, and gender of the patient. The input age, BMI and symptoms for the patient is converted to linguistic values using fuzzification process. The linguistic values and corresponding membership function have been determined by the aid of the experts. Knowledge is represented as a set of fuzzy rules. The fuzzy rules consist of two parts, head of the rule (consequent) and body of the rule (antecedent). The head of the rules consist of two arguments, disease name and region number (1-5). The body of the rules consists of all symptoms and their severity. They are stored in the Knowledge Base. The other component of the Knowledge Base is the treatment of the diseases. The working memory contains all the temporary results during the reasoning process. Backward chaining strategy was used in the inference engine of the proposed system. The accuracy of the system diagnosis is evaluated by comparing with the accuracy of the diagnosis of specialists (doctor). The authors stated that the system accuracy in diagnosing back pain diseases is 90%.

Borgohain & Sanyal (2012) developed a rule based expert system for the diagnosis of some neuromuscular disorders which included Cerebral Palsy, Parkinson's disease, Multiple Sclerosis and Muscular. The system was implemented using the Java Expert System Shell (JESS). The Expert System has a graphical user interface where the user of the system is asked to answer a few yes or no questions which are prepared according to the symptoms shown by neuromuscular disorder patients. The system made use of backward chaining for the inference engine. According to the feedback of the user, the expert system uses the RETE algorithm to search the knowledge base and matches the patterns of the symptoms to those in the knowledge base. If the pattern matches, the expert system shows the user the diagnosis of his disease. The system also advises the user to undergo tests for confirmation of the disease. It also offers the patient with the treatment options. If there is a rule in the knowledge base which matches the symptoms of the patient, the system shows the possible diagnosis in the recommendation window. It also advises the patient about the tests to confirm the disease. Moreover, the patient is also given the different treatment options for treating the disease. The primary source for knowledge acquisition for the designed system was consultation with neurology doctors, internet and medical books. The knowledge based consisted of acquiring the symptoms of the disease, treatment and test options. The knowledge base was represented in the form of rules using the Java Expert System Shell. The authors tested the system against some proven cases of neuromuscular disorders and it showed accurate results when the symptoms of the user were given as input.

2.10.4 Hybridization of Soft-Computing Techniques in Medical Diagnosis

Obi and Imianvan (2011) analysed the medical diagnosis of leukaemia using neuro-fuzzy inference procedure. The system contained 14 symptoms needed for the diagnosis of Leukaemia. The inference engine consists of a reasoning algorithm driven by production rules. These production rules are evaluated by using forward chaining approach of reasoning. The system is interactive and could tell the patient his current condition as regards Leukaemia. If the patient is having five or more of the enlisted symptoms, the patient is experiencing "severe Leukaemia" and should go for treatment urgently. If it is approximately four of the symptoms the patient is experiencing, the patient "might be suffering from Leukaemia" and hence should see a physician right away, but if it is three or less of the enlisted symptoms, the patient is not "suffering from Leukaemia".

The system contains two interesting subsystems among others: the Cognitive filter and the emotional filter. The cognitive filter ranks the patient on the presence or absence of Leukaemia disease while the emotional filter ranks the patient on the extent of his Leukaemia disease. Another interesting feature of the system is that it can also be configured to handle other kinds of diseases. The Expert system was developed using Microsoft Window XP Professional operating system, Microsoft Access Database Management system, Visual Basic Application Language and Microsoft Excel. Neuro-Solution and Crystal Report were used for Neural Networks analysis and graphical representation respectively.

Nagarajasri and Padmavathamma (2013) developed a hardware system that uses threshold neuro fuzzy expert system for the diagnosis of breast cancer. The hardware used is the ARM Cortex-M3. The neuro-fuzzy system has a 3 layered feed forward architecture. The first layer corresponds to the input variables and has five input units (mass shape, mass margin, mass density, calcification and calcification distribution). The second layer holds the fuzzy rules. The third layer consists of six output units. The aim of the work is to assist physicians and radiologists in clinical diagnosis.

Uzoka *et al* (2016) presented a framework for diagnosis of confusing tropical diseases based on the Fuzzy-Cognitive Map Hybridization. Malaria, Typhoid, Chicken Pox, Measles, Hepatitis, Yellow Fever and Urinary Tract Infection were considered in the study and an accuracy of 85% was achieved in diagnosis. The work intended to collect data from 200 physicians in Nigeria. 60% of collected data would be used as a training set while 40% would be used as a test set. The knowledge base was composed of quantitative (structured) and qualitative (unstructured) knowledge of medical diagnosis. The structured knowledge is concerned with facts, rules and events. The unstructured knowledge is heuristic knowledge.

2.10.5 Summary of Related Work

The application of Artificial Intelligence in medicine is not a new concept. The literature reviewed showed how Artificial Neural Network, Bayesian Net and Fuzzy Logic has been applied to medical diagnosis with high accuracy. A plausible trend is the hybridization of two or more computing techniques to improve the accuracy of the system developed. Table 2.9 below presents a summary of the literature reviewed.

From the foregoing, it is evident that it has become a popular trend to develop Medical Decision Support Systems based on the hybridization of two soft-computing techniques. Most of the Expert System discussed attained accuracy that is comparative or even better than that of Doctors in the field of operation. Also, it is evident that very few medical decision support systems have been developed on the Android platform for application in hospitals. Considering the recent increase in popularity of the Android Operating System, it is important to have a diagnosis system that resides on the Android OS.

Table 2.9: Summary of related works.

WORK	METHOD	LIMITATION
Al-Shayea (2011)	Feed-forward back propagation neural network.	It is not a stand-alone system.
Burnside, Rubin & Shachter (2004)	Baye's net This system uses predictive imaging features to determine the likely underlying breast disease	-
Al-Hamadani (2016)	Fuzzy Inference System based on IF-THEN fuzzy rules. The author used Triangular membership function for defuzzification and Centre of Gravity for defuzzification.	-
Obi, Eke, and Osagba (2018)	Fuzzy Logic Mining Techniques. A knowledge base deduced from 36 fuzzy rules was used for diagnosis.	The accuracy of the system is below the average for current expert systems for disease diagnosis.
Singla (2013)	Rule-based (if-then production rules) expert system.	The system has an accuracy of 70% which is very low.
Latifi, Hosseini and Mazinai (2015)	Fuzzy Inference System (FIS) for diagnosing of acute lymphocytic leukaemia in children.	It was designed in MATLAB and is not a stand-alone system.
Djam *et al* (2011)	Fuzzy Logic Expert System. Root Sum Square (RSS) was used for fuzzy inference. The author used Triangular membership function for defuzzification and Centre of Gravity for defuzzification.	Malaria is the only disease considered in the study.
Kadhim, Alam & Kaur (2011)	Fuzzy expert system	The system uses Body Mass Index (BMI) which makes it useable only to medical doctors.
Borgohain & Sanyal (2012)	Rule-based expert system. The system made use of backward chaining for the inference engine. The expert system uses the RETE algorithm to search the knowledge base	-
Obi and Imianvan (2011)	The developed system used neuro-fuzzy inference procedure. Production rules are evaluated using forward chaining approach.	It only diagnoses leukaemia
Nagarajasri and Padmavathamma (2013)	The system uses threshold neuro-fuzzy expert system for the diagnosis	It is limited to being used by physicians and radiologist.
Uzoka *et al* (2016)	Fuzzy-Cognitive Map Hybridization.	-

CHAPTER THREE

METHODOLOGY

3.1 RESEARCH METHODOLOGY

In this research, a hybridization of fuzzy logic and Analytic Hierarchy Process was used. Pairwise comparison matrix of the symptoms was generated using a nine-point scale. The matrix was solved to generate the eigenvalues (weights) of each symptom in diagnosing each disease.

3.1.1 Data Collection

This study gathered data on the following tropical diseases: Malaria, Pneumonia, Tuberculosis and Typhoid. Data on tropical diseases were collected from Doctors in four health Institutions. The Institutions are Federal Teaching Hospital, Ido-Ekiti (FETHI), Ekiti State University Teaching Hospital (EKSUTH), Obafemi Awolowo University Teaching Hospital (OAUTH) and University College Hospital (UCH). Data was collected through a structured interview. Questionnaires were shared in the health institutions mentioned above. The questionnaire used in this study is shown in Appendix A – Questionnaire on Diagnosis of Tropical Diseases. A very huge challenge faced during data collection was the very busy schedule of the medical doctors in the Nigerian health sector. They could barely spend the time to complete the questionnaire. Table 3.1 shows the symptoms of each of the tropical diseases included in this work. It should be noted that these symptoms list is not exclusive. The accuracy and design complexity of the system increases proportionately with the number of symptoms considered.

3.1.2 Data Analysis

3.1.2.1 Analytic Hierarchy Process (AHP)

The steps are described in details below.

Step 1: Develop a pairwise comparison matrix. In this study, a nine-point priority scale presented in table 2.2 was used to compare the symptoms to one another. To perform a pairwise comparison of the criteria, a comparison matrix (square matrix) of the symptoms involved in the diagnosis of each disease is prepared. Each cell in the matrix will have a value from the numerical scale shown above. The matrix is filled as explained after the table.

Table 3.1: Symptoms of diseases.

Symptoms	Malaria	Pneumonia	Tuberculosis	Typhoid
Abdominal Pain	✓	✓	✓	✓
Blood in cough/urine	-	-	✓	-
Chest pain	✓	✓	✓	✓
Chills	✓	✓	✓	✓
Constipation	-	-	-	✓
Cough	✓	✓	✓	✓
Dehydration	✓	✓	✓	✓
Diarrhoea	✓	✓	✓	✓
Fatigue	✓	✓	✓	✓
Fever	✓	✓	✓	✓
Headache	✓	✓	✓	✓
Loss of Appetite	✓	✓	✓	✓
Nausea	✓	✓	✓	✓
Rash	-	-	-	✓
Shortness of breath	✓	✓	✓	-
Sweating	-	✓	✓	✓
Vomiting	✓	✓	✓	✓
Weight loss	-	-	✓	✓

If symptom in row i is equally important as the symptom in column j in the diagnosis of the disease under consideration, the cell a_{ij} will contain 1. If symptom in row i is more important than the symptom in column j in the diagnosis of the disease under consideration, the cell a_{ij} will contain n (where n is a number from 1 to 9 as appropriate) while the cell a_{ji} will contain $\frac{1}{n}$. Tables 3.2 to 3.5 shows the pairwise comparison matrix for malaria, pneumonia, tuberculosis and typhoid respectively.

Table 3.2: Pairwise comparison matrix for malaria.

	Abdominal pain	Chest pain	Chills	Cough	Dehydration	Diarrhoea	Fatigue	Fever	Headache	Loss of appetite	Nausea	Shortness of breath	Vomiting
Abdominal pain	1	3	1/5	3	1	1	1/5	1/7	1/5	1/3	1/4	1	1
Chest pain	1/3	1	3	1/3	2	1	1/2	1/7	1/5	1/3	1/3	3	1/3
Chills	5	1/3	1	6	5	7	2	1/4	2	2	5	5	1
Cough	1/3	3	1/6	1	1	1	1/4	1/8	1/5	1/3	1/3	1	1/4
Dehydration	1	½	1/5	1	1	1	1/3	1/7	1/5	1//3	1/4	1/2	1/3
Diarrhoea	1	1	1/7	1	1	1	1/3	1/7	1/5	1/3	1/4	1/2	1/3
Fatigue	5	2	1/2	4	3	3	1	1/7	1/5	1/3	1/4	5	4
Fever	7	7	4	8	9	7	7	1	5	5	5	7	7
Headache	5	5	1/2	5	7	5	5	1/5	1	1/2	3	7	5
Loss of appetite	3	3	1/2	3	5	3	3	1/5	2	1	5	6	6
Nausea	4	3	1/5	3	3	4	4	1/5	1/3	1/5	1	3	1
Shortness of breath	1	1/3	1/5	1	1	2	1/5	1/7	1/7	1/6	1/3	1	1/2
Vomiting	1	3	1	4	4	3	1/4	1/7	1/5	1/6	1	2	1

Table 3.3: Pairwise comparison matrix for pneumonia

	Abdominal pain	Chest pain	Chills	Cough	Dehydration	Diarrhoea	Fatigue	Fever	Headache	Loss of appetite	Nausea	Shortness of breath	Sweating	Vomiting
Abdominal pain	1	1/7	1/5	1/5	1/3	1/2	1/5	1/5	1/5	1	1	1/7	1/3	1
Chest pain	7	1	3	3	5	5	3	5	5	7	3	1	7	7
Chills	5	1/3	1	3	5	7	5	5	5	7	7	1	7	7
Cough	5	1/3	1/3	1	5	5	4	4	4	7	6	2	7	9
Dehydration	3	1/5	1/5	1/5	1	1	1/3	1/3	1/5	1/3	1	1/7	2	1
Diarrhoea	2	1/5	1/7	1/5	1	1	1/2	1/5	1/7	1/2	1	1/7	2	3
Fatigue	5	1/3	1/5	¼	3	2	1	1/2	1/5	1/2	1	1/5	1/2	3
Fever	5	1/5	1/5	¼	3	5	2	1	1/5	3	1/3	1/7	1/2	3
Headache	5	1/5	1/5	¼	5	7	5	5	1	1	5	1/5	3	4
Loss of appetite	1	1/7	1/7	1/7	3	2	2	1/3	1	1	1	1/7	1/2	1
Nausea	1	1/3	1/7	1/6	1	1	1	3	1/5	1	1	1/9	1/3	1
Shortness of breath	7	1	1	½	7	7	5	7	5	7	9	1	7	9
Sweating	3	1/7	1/7	1/7	½	½	2	2	1/3	2	3	1/7	1	1
Vomiting	1	1/7	1/7	1/9	1	1/3	1/3	1/3	1/4	1	1	1/9	1	1

Table 3.4: Pairwise comparison matrix for tuberculosis

	Abdominal pain	Blood in cough/urine	Chest pain	Chills	Cough	Dehydration	Diarrhoea	Fatigue	Fever	Headache	Loss of appetite	Nausea	Shortness of breath	Sweating	Vomiting	Weight loss
Abdominal pain	1	1/9	1/7	1/5	1/5	1/3	1/2	1/5	1/5	1/5	1	1	1/7	1/3	1	1/9
Blood in cough/urine	9	1	7	7	8	9	9	8	7	6	5	7	6	9	9	1/5
Chest pain	7	1/7	1	3	3	5	5	3	5	5	7	3	1	7	7	1/7
Chills	5	1/7	1/3	1	3	5	7	5	5	5	7	7	1	7	7	1/7
Cough	5	1/8	1/3	1/3	1	5	5	4	4	4	7	6	2	7	9	1/5
Dehydration	3	1/9	1/5	1/5	1/5	1	1	1/3	1/3	1/5	1/3	1	1/7	2	1	1/9
Diarrhoea	2	1/9	1/5	1/7	1/5	1	1	1/2	1/5	1/7	1/2	1	1/7	2	3	1/9
Fatigue	5	1/8	1/3	1/5	¼	3	2	1	1/2	1/5	1/2	1	1/5	1/2	3	1/7
Fever	5	1/7	1/5	1/5	¼	3	5	2	1	1/5	3	1/3	1/7	1/2	3	1/7
Headache	5	1/6	1/5	1/5	¼	5	7	5	5	1	1	5	1/5	3	4	1/5
Loss of appetite	1	1/5	1/7	1/7	1/7	3	2	2	1/3	1	1	1	1/7	1/2	1	1/7
Nausea	1	1/7	1/3	1/7	1/6	1	1	1	3	1/5	1	1	1/9	1/3	1	1/9
Shortness of breath	7	1/6	1	1	½	7	7	5	7	5	7	9	1	7	9	1/3
Sweating	3	1/9	1/7	1/7	1/7	½	½	2	2	1/3	2	3	1/7	1	1	1/9
Vomiting	1	1/9	1/7	1/7	1/9	1	1/3	1/3	1/3	1/4	1	1	1/9	1	1	1/9
Weight loss	9	5	7	7	5	9	9	7	7	5	7	9	3	9	9	1

Table 3.5: Pairwise comparison matrix for typhoid

	Abdominal pain	Chest pain	Chills	Constipation	Cough	Dehydration	Diarrhoea	Fatigue	Fever	Headache	Loss of appetite	Nausea	Rashes	Sweating	Vomiting	Weight loss
Abdominal pain	1	3	1/5	3	3	1	1	1/5	1/7	1/5	1/3	1/4	5	4	1	3
Chest pain	1/3	1	3	2	1/3	2	1	1/2	1/7	1/5	1/3	1/3	7	6	1/3	3
Chills	5	1/3	1	4	6	5	7	2	1/4	2	2	5	9	9	1	7
Constipation	1/3	1/2	1/4	1	3	1	1/3	1/7	1/9	1/5	1/3	1	4	5	1	1/3
Cough	1/3	3	1/6	1/3	1	1	1	1/4	1/8	1/5	1/3	1/3	3	2	1/4	1
Dehydration	1	½	1/5	1	1	1	1	1/3	1/7	1/5	1//3	1/4	3	2	1/3	2
Diarrhoea	1	1	1/7	3	1	1	1	1/3	1/7	1/5	1/3	1/4	4	4	1/3	2
Fatigue	5	2	1/2	7	4	3	3	1	1/7	1/5	1/3	1/4	7	7	4	7
Fever	7	7	4	9	8	9	7	7	1	5	5	5	9	8	7	7
Headache	5	5	1/2	5	5	7	5	5	1/5	1	1/2	3	6	4	5	4
Loss of appetite	3	3	1/2	3	3	5	3	3	1/5	2	1	5	5	5	6	1/3
Nausea	4	3	1/5	1	3	3	4	4	1/5	1/3	1/5	1	4	3	1	1/5
Rashes	1/5	1/7	1/9	¼	1/3	1/3	1/4	1/7	1/9	1/6	1/5	1/4	1	1/3	1/5	1/7
Sweating	1/4	1/6	1/9	1/5	1/2	1/2	1/4	1/7	1/8	1/4	1/5	1/3	3	1	1/5	1/3
Vomiting	1	3	1	1	4	4	3	1/4	1/7	1/5	1/6	1	5	5	1	1/2
Weight loss	1/3	1/3	1/7	3	1	1/2	1/2	1/7	1/7	1/4	3	5	7	3	2	1

Step 2: Normalize the matrix. This is done by dividing each cell by the total of the column it belongs. The mathematical representation of the normalization process is shown in equation 3.1.

$$E_{nj} = \frac{a_{nj}}{\sum_{n=1}^{i} a_{nj}} \qquad (3.1)$$

Where E_{nj} is the eigenvalue of cell a_{nj}.

Tables 3.6 to 3.9 shows the normalized matrix for malaria, pneumonia, tuberculosis and typhoid respectively.

Table 3.6: normalized matrix for malaria.

	Abdominal pain	Chest pain	Chills	Cough	Dehydration	Diarrhoea	Fatigue	Fever	Headache	Loss of appetite	Nausea	Shortness of breath	Vomiting
Abdominal pain	0.0289	0.0933	0.0172	0.0744	0.0233	0.0256	0.0083	0.0480	0.0168	0.0300	0.0114	0.0238	0.0360
Chest pain	0.0095	0.0311	0.2584	0.0082	0.0465	0.0256	0.0208	0.0480	0.0168	0.0300	0.0150	0.0714	0.0119
Chills	0.1443	0.0103	0.0861	0.1488	0.1163	0.1795	0.0831	0.0840	0.1685	0.1816	0.2274	0.1190	0.0360
Cough	0.0095	0.0933	0.0144	0.0248	0.0233	0.0256	0.0104	0.0420	0.0168	0.0300	0.0150	0.0238	0.0090
Dehydration	0.0289	0.0155	0.0172	0.0248	0.0233	0.0256	0.0137	0.0480	0.0168	0.0300	0.0114	0.0119	0.0119
Diarrhoea	0.0289	0.0311	0.0123	0.0248	0.0233	0.0256	0.0137	0.0480	0.0168	0.0300	0.0114	0.0119	0.0119
Fatigue	0.1443	0.0622	0.0431	0.0992	0.0698	0.0769	0.0416	0.0480	0.0168	0.0300	0.0114	0.1190	0.1442
Fever	0.2020	0.2177	0.3445	0.1984	0.2093	0.1795	0.2909	0.3361	0.4211	0.4540	0.2274	0.1667	0.2523
Headache	0.1443	0.1555	0.0431	0.1240	0.1628	0.1282	0.2078	0.0672	0.0842	0.0454	0.1364	0.1667	0.1802
Loss of appetite	0.0866	0.0933	0.0431	0.0744	0.1163	0.0769	0.1247	0.0672	0.1685	0.0908	0.2274	0.1429	0.2163
Nausea	0.1154	0.0933	0.0172	0.0744	0.0698	0.1026	0.1663	0.0672	0.0278	0.0182	0.0455	0.0714	0.0360
Shortness of breath	0.0289	0.0103	0.0172	0.0248	0.0233	0.0513	0.0083	0.0480	0.0120	0.0151	0.0150	0.0238	0.0180
Vomiting	0.0289	0.0933	0.0861	0.0992	0.0930	0.0769	0.0104	0.0480	0.0168	0.0151	0.0455	0.0476	0.0360

Table 3.7: normalized matrix for pneumonia

	Abdominal pain	Chest pain	Chills	Cough	Dehydration	Diarrhoea	Fatigue	Fever	Headache	Loss of appetite	Nausea	Shortness of breath	Sweating	Vomiting
Abdominal pain	0.0196	0.0305	0.0284	0.0212	0.0081	0.0113	0.0064	0.0059	0.0088	0.0254	0.0248	0.0221	0.0084	0.0196
Chest pain	0.1373	0.2131	0.4259	0.3187	0.1225	0.1128	0.0957	0.1475	0.2200	0.1780	0.0744	0.1543	0.1788	0.1373
Chills	0.0980	0.0703	0.1420	0.3187	0.1225	0.1579	0.1594	0.1475	0.2200	0.1780	0.1736	0.1543	0.1788	0.1373
Cough	0.0980	0.0703	0.0468	0.1062	0.1225	0.1128	0.1276	0.1180	0.1760	0.1780	0.1488	0.3087	0.1788	0.1765
Dehydration	0.0588	0.0426	0.0284	0.0212	0.0245	0.0226	0.0105	0.0097	0.0088	0.0084	0.0248	0.0221	0.0511	0.0196
Diarrhoea	0.0392	0.0426	0.0203	0.0212	0.0245	0.0226	0.0159	0.0059	0.0063	0.0127	0.0248	0.0221	0.0511	0.0588
Fatigue	0.0980	0.0703	0.0284	0.0266	0.0735	0.0451	0.0319	0.0148	0.0088	0.0127	0.0248	0.0309	0.0128	0.0588
Fever	0.0980	0.0426	0.0284	0.0266	0.0735	0.1128	0.0638	0.0295	0.0088	0.0763	0.0082	0.0221	0.0128	0.0588
Headache	0.0980	0.0426	0.0284	0.0266	0.1225	0.1579	0.1594	0.1475	0.0440	0.0254	0.1240	0.0309	0.0766	0.0784
Loss of appetite	0.0196	0.0305	0.0203	0.0152	0.0735	0.0451	0.0638	0.0097	0.0440	0.0254	0.0248	0.0221	0.0128	0.0196
Nausea	0.0196	0.0703	0.0203	0.0177	0.0245	0.0226	0.0319	0.0885	0.0088	0.0254	0.0248	0.0171	0.0084	0.0196
Shortness of breath	0.1373	0.2131	0.1420	0.0531	0.1714	0.1579	0.1594	0.2066	0.2200	0.1780	0.2232	0.1543	0.1788	0.1765
Sweating	0.0588	0.0305	0.0203	0.0152	0.0122	0.0113	0.0638	0.0590	0.0145	0.0509	0.0744	0.0221	0.0255	0.0196
Vomiting	0.0196	0.0305	0.0203	0.0118	0.0245	0.0074	0.0105	0.0097	0.0110	0.0254	0.0248	0.0171	0.0255	0.0196

Table 3.8: normalized matrix for tuberculosis

	Abdominal pain	Blood in cough/urine	Chest pain	Chills	Cough	Dehydration	Diarrhoea	Fatigue	Fever	Headache	Loss of appetite	Nausea	Shortness of breath	Sweating	Vomiting	Weight loss
Abdominal pain	0.0145	0.0140	0.0076	0.0095	0.0089	0.0056	0.0080	0.0043	0.0042	0.0059	0.0195	0.0178	0.0092	0.0058	0.0145	0.0336
Blood in cough/urine	0.1304	0.1264	0.3745	0.3326	0.3569	0.1530	0.1444	0.1726	0.1462	0.1779	0.0974	0.1243	0.3876	0.1575	0.1304	0.0604
Chest pain	0.1014	0.0181	0.0535	0.1426	0.1338	0.0850	0.0802	0.0647	0.1044	0.1483	0.1364	0.0533	0.0646	0.1225	0.1014	0.0432
Chills	0.0725	0.0181	0.0177	0.0475	0.1338	0.0850	0.1123	0.1079	0.1044	0.1483	0.1364	0.1243	0.0646	0.1225	0.1014	0.0432
Cough	0.0725	0.0158	0.0177	0.0157	0.0446	0.0850	0.0802	0.0863	0.0835	0.1186	0.1364	0.1065	0.1292	0.1225	0.1304	0.0604
Dehydration	0.0435	0.0140	0.0107	0.0095	0.0089	0.0170	0.0160	0.0071	0.0069	0.0059	0.0064	0.0178	0.0092	0.0350	0.0145	0.0336
Diarrhoea	0.0290	0.0140	0.0107	0.0068	0.0089	0.0170	0.0160	0.0108	0.0042	0.0042	0.0097	0.0178	0.0092	0.0350	0.0435	0.0336
Fatigue	0.0725	0.0158	0.0177	0.0095	0.0112	0.0510	0.0321	0.0216	0.0104	0.0059	0.0097	0.0178	0.0129	0.0087	0.0435	0.0432
Fever	0.0725	0.0181	0.0107	0.0095	0.0112	0.0510	0.0802	0.0431	0.0209	0.0059	0.0584	0.0059	0.0092	0.0087	0.0435	0.0432
Headache	0.0725	0.0211	0.0107	0.0095	0.0112	0.0850	0.1123	0.1079	0.1044	0.0297	0.0195	0.0888	0.0129	0.0525	0.0580	0.0604
Loss of appetite	0.0145	0.0253	0.0076	0.0068	0.0064	0.0510	0.0321	0.0431	0.0069	0.0297	0.0195	0.0178	0.0092	0.0087	0.0145	0.0432
Nausea	0.0145	0.0181	0.0177	0.0068	0.0074	0.0170	0.0160	0.0216	0.0626	0.0059	0.0195	0.0178	0.0072	0.0058	0.0145	0.0336
Shortness of breath	0.1014	0.0211	0.0535	0.0475	0.0223	0.1190	0.1123	0.1079	0.1462	0.1483	0.1364	0.1598	0.0646	0.1225	0.1304	0.0997
Sweating	0.0435	0.0140	0.0076	0.0068	0.0064	0.0085	0.0080	0.0431	0.0418	0.0098	0.0390	0.0533	0.0092	0.0175	0.0145	0.0336
Vomiting	0.0145	0.0140	0.0076	0.0068	0.0050	0.0170	0.0053	0.0071	0.0069	0.0074	0.0195	0.0178	0.0072	0.0175	0.0145	0.0336
Weight loss	0.1304	0.6321	0.3745	0.3326	0.2231	0.1530	0.1444	0.1510	0.1462	0.1483	0.1364	0.1598	0.1938	0.1575	0.1304	0.3020

Table 3.9: normalized matrix for typhoid

	Abdominal pain	Chest pain	Chills	Constipation	Cough	Dehydration	Diarrhoea	Fatigue	Fever	Headache	Loss of appetite	Nausea	Rashes	Sweating	Vomiting	Weight loss
Abdominal pain	0.0288	0.0910	0.0166	0.0685	0.0679	0.0226	0.0261	0.0082	0.0430	0.0159	0.0226	0.0089	0.0610	0.0585	0.0326	0.0773
Chest pain	0.0095	0.0303	0.2495	0.0457	0.0075	0.0451	0.0261	0.0205	0.0430	0.0159	0.0226	0.0117	0.0854	0.0878	0.0108	0.0773
Chills	0.1438	0.0100	0.0832	0.0914	0.1359	0.1128	0.1826	0.0819	0.0752	0.1588	0.1372	0.1771	0.1098	0.1317	0.0326	0.1803
Constipation	0.0095	0.0152	0.0208	0.0228	0.0679	0.0226	0.0086	0.0058	0.0334	0.0159	0.0226	0.0354	0.0488	0.0732	0.0326	0.0085
Cough	0.0095	0.0910	0.0139	0.0075	0.0226	0.0226	0.0261	0.0102	0.0376	0.0159	0.0226	0.0117	0.0366	0.0293	0.0082	0.0258
Dehydration	0.0288	0.0152	0.0166	0.0228	0.0226	0.0226	0.0261	0.0135	0.0430	0.0159	0.0226	0.0089	0.0366	0.0293	0.0108	0.0515
Diarrhoea	0.0288	0.0303	0.0119	0.0685	0.0226	0.0226	0.0261	0.0135	0.0430	0.0159	0.0226	0.0089	0.0488	0.0585	0.0108	0.0515
Fatigue	0.1438	0.0607	0.0416	0.1599	0.0906	0.0677	0.0783	0.0409	0.0430	0.0159	0.0226	0.0089	0.0854	0.1024	0.1305	0.1803
Fever	0.2013	0.2123	0.3326	0.2056	0.1812	0.2030	0.1826	0.2865	0.3010	0.3969	0.3430	0.1771	0.1098	0.1171	0.2285	0.1803
Headache	0.1438	0.1517	0.0416	0.1142	0.1132	0.1579	0.1304	0.2047	0.0602	0.0794	0.0343	0.1062	0.0732	0.0585	0.1632	0.1030
Loss of appetite	0.0863	0.0910	0.0416	0.0685	0.0679	0.1128	0.0783	0.1228	0.0602	0.1588	0.0686	0.1771	0.0610	0.0732	0.1958	0.0085
Nausea	0.1150	0.0910	0.0166	0.0228	0.0679	0.0677	0.1044	0.1637	0.0602	0.0262	0.0137	0.0354	0.0488	0.0439	0.0326	0.0052
Rashes	0.0058	0.0043	0.0092	0.0057	0.0075	0.0074	0.0065	0.0058	0.0334	0.0132	0.0137	0.0089	0.0122	0.0048	0.0065	0.0037
Sweating	0.0072	0.0051	0.0092	0.0046	0.0113	0.0113	0.0065	0.0058	0.0376	0.0198	0.0137	0.0117	0.0366	0.0146	0.0065	0.0085
Vomiting	0.0288	0.0910	0.0832	0.0228	0.0906	0.0902	0.0783	0.0102	0.0430	0.0159	0.0114	0.0354	0.0610	0.0732	0.0326	0.0129
Weight loss	0.0095	0.0100	0.0119	0.0685	0.0226	0.0113	0.0130	0.0058	0.0430	0.0198	0.2058	0.1771	0.0854	0.0439	0.0653	0.0258

Step 3: Derive the eigenvectors. This step derives the weights for each symptom needed to diagnose each of the four diseases under consideration. The eigenvector for each row is calculated by finding the average of that row. Mathematically, the eigenvector is given in equation 3.2.

$$\lambda_i = \frac{E_{ij}}{n} \tag{3.2}$$

Where n is the number of columns in the matrix.

Tables 3.10 to 3.13 shows the eigenvectors (or weights) for malaria, pneumonia, tuberculosis and typhoid respectively.

Table 3.10: Eigenvectors for malaria.

Symptom	Eigenvectors
Abdominal pain	0.0336
Chest pain	0.0456
Chills	0.1219
Cough	0.0260
Dehydration	0.0215
Diarrhoea	0.0223
Fatigue	0.0697
Fever	0.2692
Headache	0.1266
Loss of appetite	0.1176
Nausea	0.0696
Shortness of breath	0.0228
Vomiting	0.0536

Table 3.11: Eigenvectors for pneumonia

Symptom	Eigenvectors
Abdominal pain	0.0172
Chest pain	0.1797
Chills	0.1613
Cough	0.1406
Dehydration	0.0252
Diarrhoea	0.0263
Fatigue	0.0384
Fever	0.0473
Headache	0.0830
Loss of appetite	0.0304
Nausea	0.0285
Shortness of breath	0.1694
Sweating	0.0341
Vomiting	0.0184

Table 3.12: Eigenvectors for tuberculosis

Symptom	Eigenvectors
Abdominal pain	0.0114
Blood in cough/urine	0.1920
Chest pain	0.0908
Chills	0.0900
Cough	0.0816
Dehydration	0.0160
Diarrhoea	0.0169
Fatigue	0.0240
Fever	0.0307
Headache	0.0535
Loss of appetite	0.0210
Nausea	0.0179
Shortness of breath	0.0995
Sweating	0.0223
Vomiting	0.0126
Weight loss	0.2197

Table 3.13: Eigenvectors for typhoid

Symptom	Eigenvectors
Abdominal pain	0.0406
Chest pain	0.0493
Chills	0.1153
Constipation	0.0277
Cough	0.0244
Dehydration	0.0242
Diarrhoea	0.0303
Fatigue	0.0795
Fever	0.2287
Headache	0.1085
Loss of appetite	0.0920
Nausea	0.0572
Rashes	0.0093
Sweating	0.0131
Vomiting	0.0488
Weight loss	0.0512

3.1.2.2. Learning From Example (LFE)

Learning From Example was used to extract fuzzy rules from the questionnaires. The LFE technique generates a rule-base for a fuzzy system by using numerical data from a physical system and possibly linguistic information from a human expert. The LFE is not capable of generating membership functions. Therefore, the membership functions are determined through other means. 70% of the questionnaire was used as training data to form fuzzy rules. Since the data got in this study is not from a physical system, each questionnaire will likely generate a rule. If two questionnaires result in two rules with the same antecedents and consequent, the rules are represented only once. Applying LFE to the questionnaires generated 69 fuzzy rules. These rules were used as the rule base of the system designed. The rules are presented in Appendix B – Rule Base

3.2 DESIGN METHODOLOGY

3.2.1. Hybridization of Fuzzy Logic and Analytic Hierarchy Process

In this work, fuzzy logic and Analytic Hierarchy Process was hybridized to exploit the strengths of the two techniques. The eigenvectors generated through AHP were used to normalize the degree of membership of each symptom during inference. The eigenvectors as presented in table 3.10 to 3.13 show the importance of each symptom in diagnosing a particular tropical disease.

Once Fuzzification is done on the user input for a symptom, the degree of membership of the user input is normalized using the corresponding weight for that symptom. The weights were derived through AHP. For example, table 3.10 shows the eigenvectors of the symptoms required to diagnose malaria. From the table, it can be seen that fever is most important in diagnosing malaria. During inference, the fuzzified value of the user input for each symptom is multiplied by the eigenvectors from table 3.10 to 3.13. Therefore, if the user inputs 60% for fever, it will be fuzzified as 0.8133 for the fuzzy set mild. Before this number is used in inference, it is multiplied by the eigenvector of malaria in the table.

3.2.2 Fuzzy Logic

Fuzzy Logic was adopted in this study because of its ability to handle the uncertainty and incompleteness of information inherent in medical diagnosis. Fuzzy Logic involves fuzzification, inference, aggregation and defuzzification.

3.2.2.1. Fuzzification

This system designed uses five (5) membership values: Very low, low, mild, high, very high. Symptom weights are fuzzified using the membership functions in equations 3.3 to 3.7.

$$\mu_{very\_low}(X) = \begin{Bmatrix} \frac{37.5-x}{37.5-0} & if\ 0 \leq x \leq 37.5 \\ 0 & if\ 37.5 \leq x \end{Bmatrix} \tag{3.3}$$

$$\mu_{low}(X) = \begin{Bmatrix} 0 & if\ x \leq -16 \\ \frac{x-(-16)}{21.5-(-16)} & if\ -16 \leq x \leq 21.5 \\ 1 & if\ 21.5 \leq x \leq 26.5 \\ \frac{64-x}{64-26.5} & if\ 26.5 \leq x \leq 64 \\ 0 & if\ 64 \leq x \end{Bmatrix} \tag{3.4}$$

$$\mu_{mild}(X) = \begin{Bmatrix} 0 & if\ x \leq 10.5 \\ \frac{x-10.5}{b-10.5} & if\ 10.5 \leq x \leq 48 \\ 1 & if\ 48 \leq x \leq 53 \\ \frac{90.5-x}{90.5-53} & if\ 53 \leq x \leq 90.5 \\ 0 & if\ 90.5 \leq x \end{Bmatrix} \tag{3.5}$$

$$\mu_{high}(X) = \begin{Bmatrix} 0 & if\ x \leq 36.75 \\ \frac{x-36.75}{74.25-36.75} & if\ 36.75 \leq x \leq 74.25 \\ 1 & if\ 74.25 \leq x \leq 79.25 \\ \frac{116.75-x}{116.75-79.25} & if\ 79.25 \leq x \leq 116.75 \\ 0 & if\ 116.75 \leq x \end{Bmatrix} \tag{3.6}$$

$$\mu_{very\_high}(X) = \begin{Bmatrix} 0 & if\ x \leq 63 \\ \frac{x-63}{100.5-63} & if\ 63 \leq x \leq 100.5 \end{Bmatrix} \tag{3.7}$$

Trapezoidal membership function was used in this work to fuzzify the inputs to the system. It was also used to represent the output of the fuzzy rules in the system. The points of the membership function for the five linguistic variables are shown below

Table 3.14: Trapezoidal membership function used to represent input and output.

Membership function	Points
Very low	-42.5, -5, 0, 37.5
Low	-16, 21.5, 26.5, 64
Moderate	10.5, 48, 53, 90.5
Intense	36.75, 74.25, 79.25, 116.75
Very intense	63, 100.5, 105.5, 143

3.2.2.2. Fuzzy Inference

Usually, a rule in a fuzzy system will more than one antecedent connected by a fuzzy operator such as AND or OR. There are several ways of applying these fuzzy operators. The most common way is to use minimum for the AND operator and maximum for the OR operator. One of the features of a fuzzy system that contributes to its fuzziness is that the mathematical application of the fuzzy operators is not set in stone. Therefore, unlike the Boolean AND whose truth table can be drawn, the truth table cannot be drawn for the fuzzy AND adding to the fact that application will vary from system to system.

All the 69 fuzzy rules in this study have more than one antecedent. The antecedents are connected by fuzzy AND operator. Average was used to model the fuzzy AND operator in this study. Equation 3.8 shows the AND operator.

$$\mu(A * B) = average(\mu A, \mu B) \tag{3.8}$$

The antecedent connective uses the average value of all the degrees of membership in the antecedent as the truth value of the aggregated rule antecedent. The average implication method truncates the output membership function of each rule at the value of the average of the truth values of the rule antecedent.

3.2.2.3. Aggregation

The rule base contains more than one rule. Therefore the outputs of the rules have to be combined together to form a fuzzy set. Minimum is an aggregation method that is commonly used. In this method, the rules with the minimum truth value for a particular fuzzy set are chosen. In this study, the average method is used for aggregation. Therefore, the average of the truth values of rules with the same output is used. For example, if five rules have output MALARIA is very low and the truth value of the output of each of the rules are 0.8, 1, 0.5, 0.2 and 0.6, the average of the five numbers (0.62) will be chosen as the aggregate of the five rules

3.2.2.4. Defuzzification

The output from the fuzzy engine is a fuzzy set. This output makes no sense to the medical practitioner or a patient. It must be converted to a form that is understandable to them. The process is called defuzzification. It can be likened to the reverse of fuzzification since fuzzification takes input that makes sense to the user and converts it to elements in a fuzzy set. Commonly used defuzzification techniques include extreme value strategies and centroid strategies. They have been discussed in the literature review. In this study, the Centre of Area (COA) method was used for defuzzification. The Centre of Area (COA) method chooses the control action that corresponds to the centre of the area with a membership greater than zero. It is represented mathematically in equation 3.9.

$$u_{COA} = \frac{\int u.\mu(u)du}{\int \mu(u)du} \tag{3.9}$$

3.2.3. System Architecture

The architecture of the system designed in this study is shown in figure 3.2. The system is made up of the user interface, fuzzifier, inference engine, rule base and defuzzifier. The user interacts with the system through the user interface. The user input is fuzzified by the fuzzifier.

User Interface. The User Interface is the medium through which the user interacts with the system. The user keys in his/her symptoms and receives diagnosis through the user interface. The User Interface is written in Extensible Markup Language (XML).

Fuzzifier. The user interacts with the system through a slider to show the intensity of each symptom felt. The slider generates an integer between 0 and 100. This integer serves as the input into the fuzzy system. The fuzzifier takes this integer and generates its membership degree in each of the five fuzzy sets used in this study.

Rule Base. The rule base contains the rules on which the system operates. It is a two-dimensional matrix. It contains 69 rules of the form:

```
If Abdominal Pain is very low AND Chest Pain is very low AND
Chills is very low AND Cough is very low AND Dehydration is
very low AND Diarrhoea is very low AND Fatigue is very low
AND Fever is very low AND Headache is very low AND Loss of
```

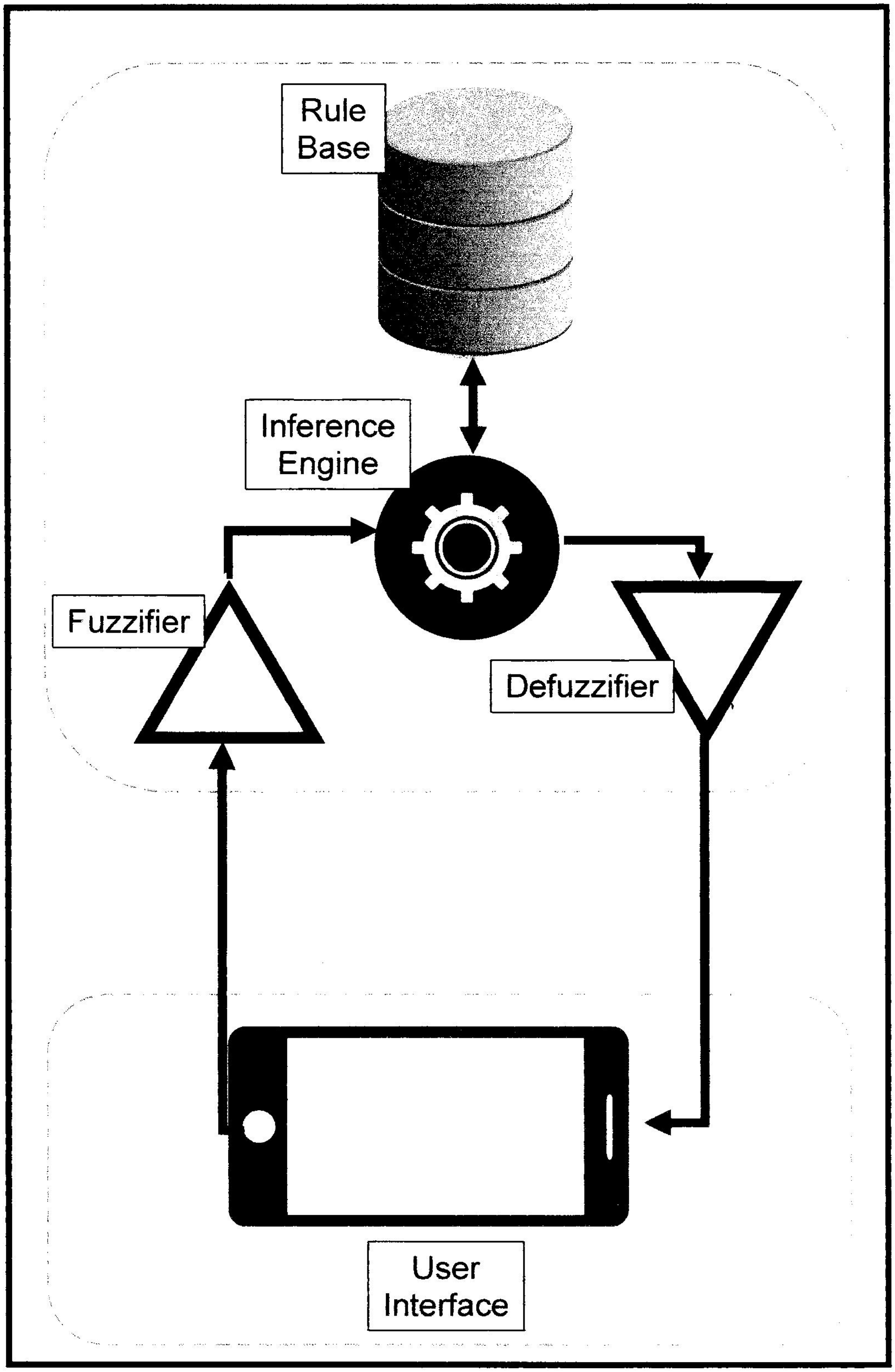


Figure 3.1: System architecture of the designed system

```
Appetite is very low AND Nausea is very low AND Vomiting is
very low then malaria is Low.
```

Inference Engine. The inference engine is the part of the system that performs reasoning. It receives the user inputs stored in an array of integer. It fires each rule in the rule base, gets the required membership degrees for that rule. It does aggregation by finding the average of the degree of membership of the terms in the antecedent. It also calculates the implication (or weight) of that rule. The input to the Inference engine is a set of membership degrees while the output is a set of fuzzy sets. The Inference engine then performs aggregation on this fuzzy sets. Aggregation is done by the max method.

Defuzzifier. The defuzzifier takes the output of aggregation from the inference engine and generates a single crisp value that best represents its input. Therefore the inputs to the defuzzifier is a set of fuzzy sets while its output is a crisp value. The crisp value represents the probability that the patient under consideration has the disease under consideration.

3.2.4. Algorithms

The algorithm of operation of the Android application

1. Display splash screen for three seconds.
2. Accept user name, age and gender.
3. Request the user to select one or more symptoms from the symptom list.
4. Save the selected symptoms using an array of Boolean.
5. Request the user to select intensity for each symptom selected in 4 above.
6. Store the intensities in an array of integer.
7. Create an Inference object and pass the intensities in 6 to the object.
8. Run diagnosis
9. Display result of diagnosis to the user.

The algorithm of the Fuzzy Inference System

```
1. Start
2. Cycle through each rule in the rule base
        a. Cycle through each statement in the current rule.
                i. Check if there is a corresponding user input.
                        1. If YES
                a. Calculate the input's membership function.
                b. Normalize and add to list A
            2. If NO, add "0" to the list
        b. Exit loop if there is no next statement in the current rule.
        c. Calculate the implication of the current rule as the average of list A and
           add the average to list B.
3. Exit the loop if there is no next rule.
4. Aggregate list B
5. Defuzzify
6. Stop
```

3.2.5. Flowchart

The flowchart for the developed system is presented in figure 3.3.

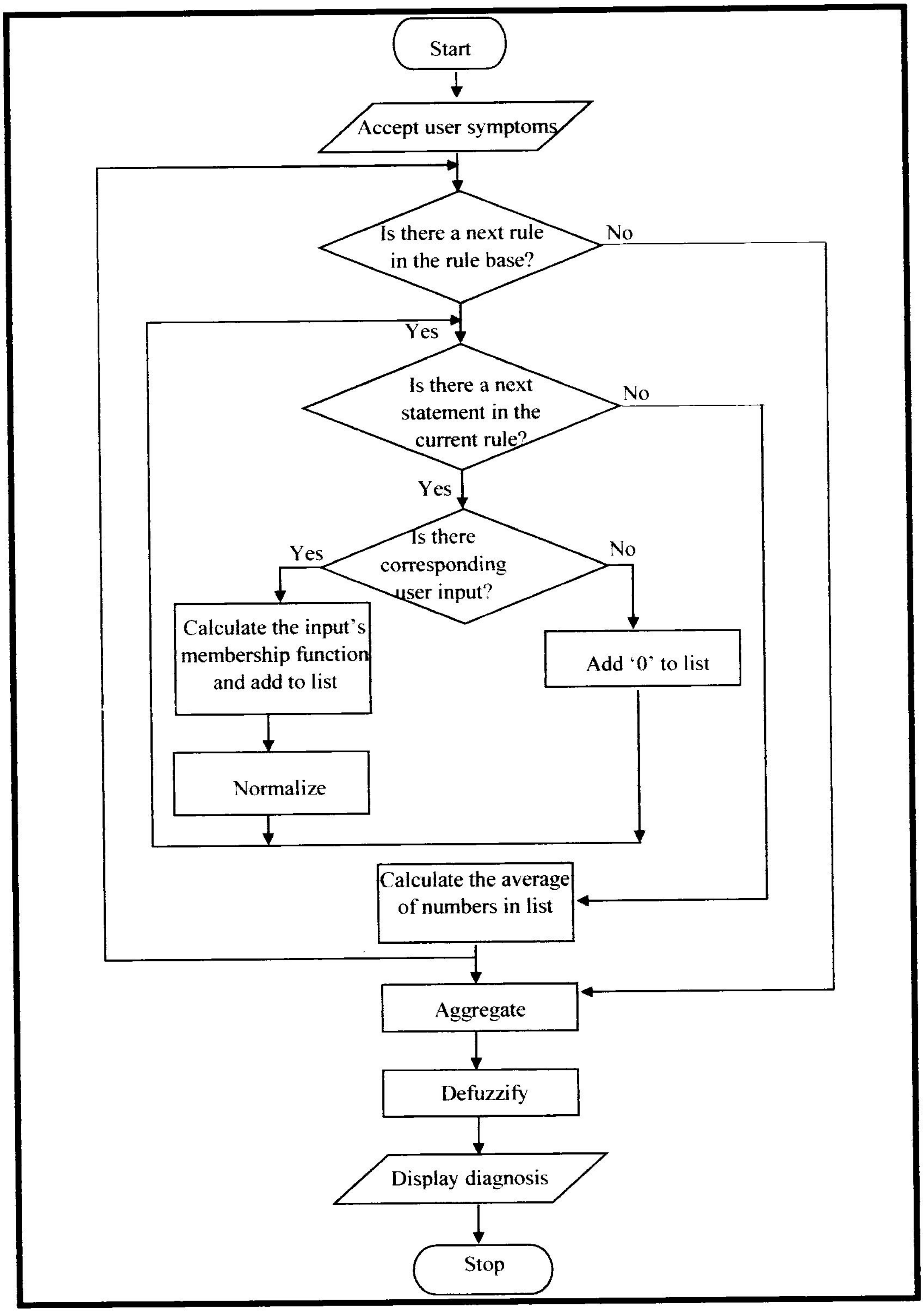

Start
Accept user symptoms
Is there a next rule in the rule base?
No
Yes
Is there a next statement in the current rule?
No
Yes
Is there corresponding user input?
Yes
No
Calculate the input's membership function and add to list
Add '0' to list
Normalize
Calculate the average of numbers in list
Aggregate
Defuzzify
Display diagnosis
Stop

Figure 3.2: Flowchart of the designed system.

3.3 DEVELOPMENT METHODOLOGY

The system was developed using Java programming language. Java is a programming language that can be easily used to program Android application, making it the programming language of choice for this project. The user interface was written in XML. Android Studio 2.2 was used in writing the Android application. The code listing of the Fuzzy Inference System is presented in Appendix C – Fuzzy Inference Class – Code Listing. Fuzzification was done by trapezoidal membership function. The function responsible for fuzzification is *Fuzzifier()*. The inference was done by average as against the usual and or product method due to peculiarities of medical diagnosis. The function that does the inference in the code listing is *Inference()*. Defuzzification is by Centre of Area defuzzification.

The designed system is a Multiple Inputs Multiple Outputs (MIMO) system with 18 inputs and four outputs. This presents a very great challenge during design and development. This challenge was simplified by a concept in Passino and Yurkovich (1998). The authors stated that a MIMO system is a sum of several Multiple Inputs Single Output (MISO) systems. Therefore the MIMO system was designed and developed as four MISO system. The authors also stated a way to simplify MIMO rules. Every MIMO rules can be broken down into several MISO rules. For Example, the rule:

If A & B & C & … Then X & Y & Z can be broken down into

If A & B & C & … Then X;

If A & B & C & … Then Y;

If A & B & C & … Then Z;

3.4 EVALUATION

The designed system was developed on the Android system. It was tested with real-life data of patients obtained by questionnaires. This system was developed to reduce patient waiting time while maintaining the accuracy of the conventional method of diagnosing. Therefore, the performance of the developed system was compared with the performance of medical doctors who helped fill the questionnaires based on sensitivity and consultation time. 30% of the questionnaires gathered in this study was used as test data to evaluate the system.

CHAPTER FOUR
RESULTS AND DISCUSSION

4.1 RESULT FROM MATLAB

A script was developed in MATLAB to calculate the normalized matrixes and generate the weight of each symptom for each disease. This MATLAB script is presented in Appendix D – MATLAB Script. Figure 4.1 to 4.4 presents a bar chart showing the weight of the symptoms for diagnosing malaria pneumonia, tuberculosis and typhoid respectively. From the charts, the symptoms that are more significant for diagnosing each of the diseases is obvious. For example, the bar chart for malaria shows that fever is most important in diagnosing malaria.

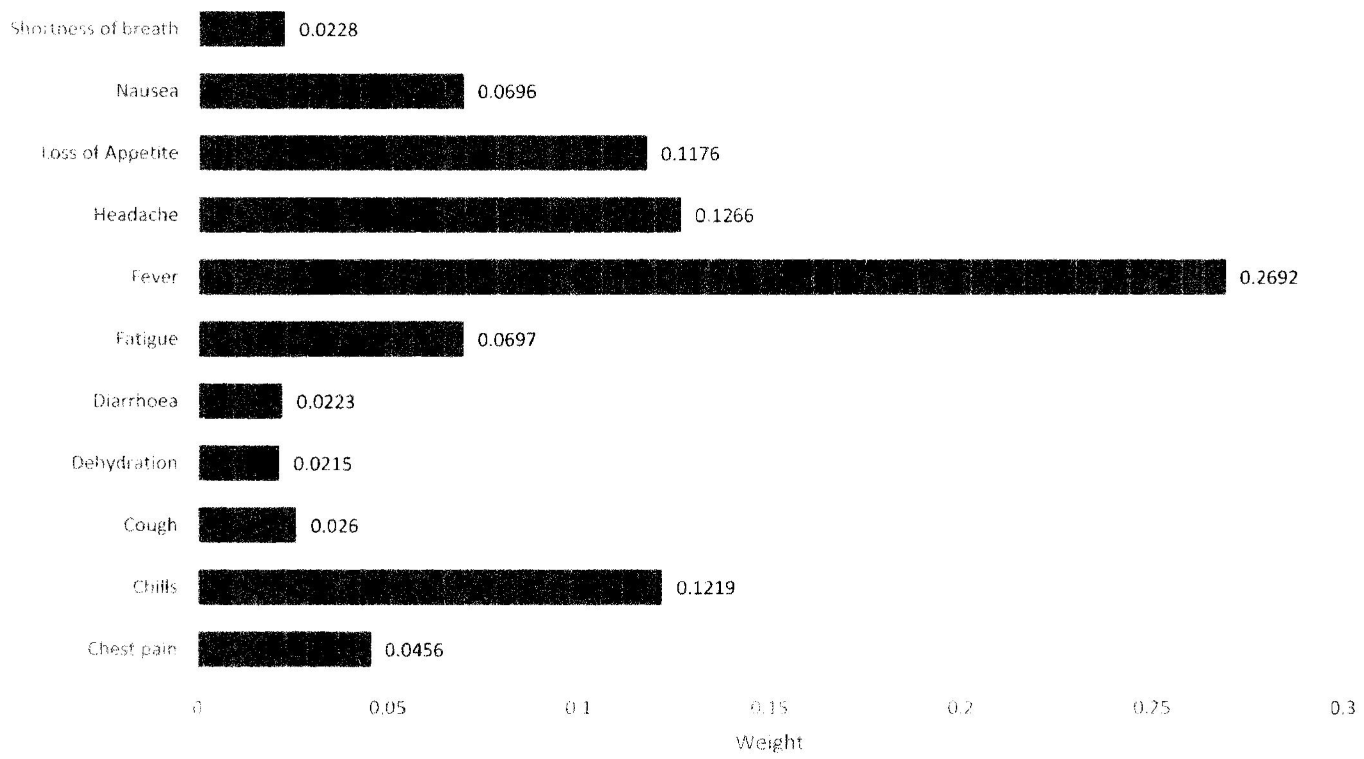


Figure 2.1: Bar chart showing weights generated for malaria.

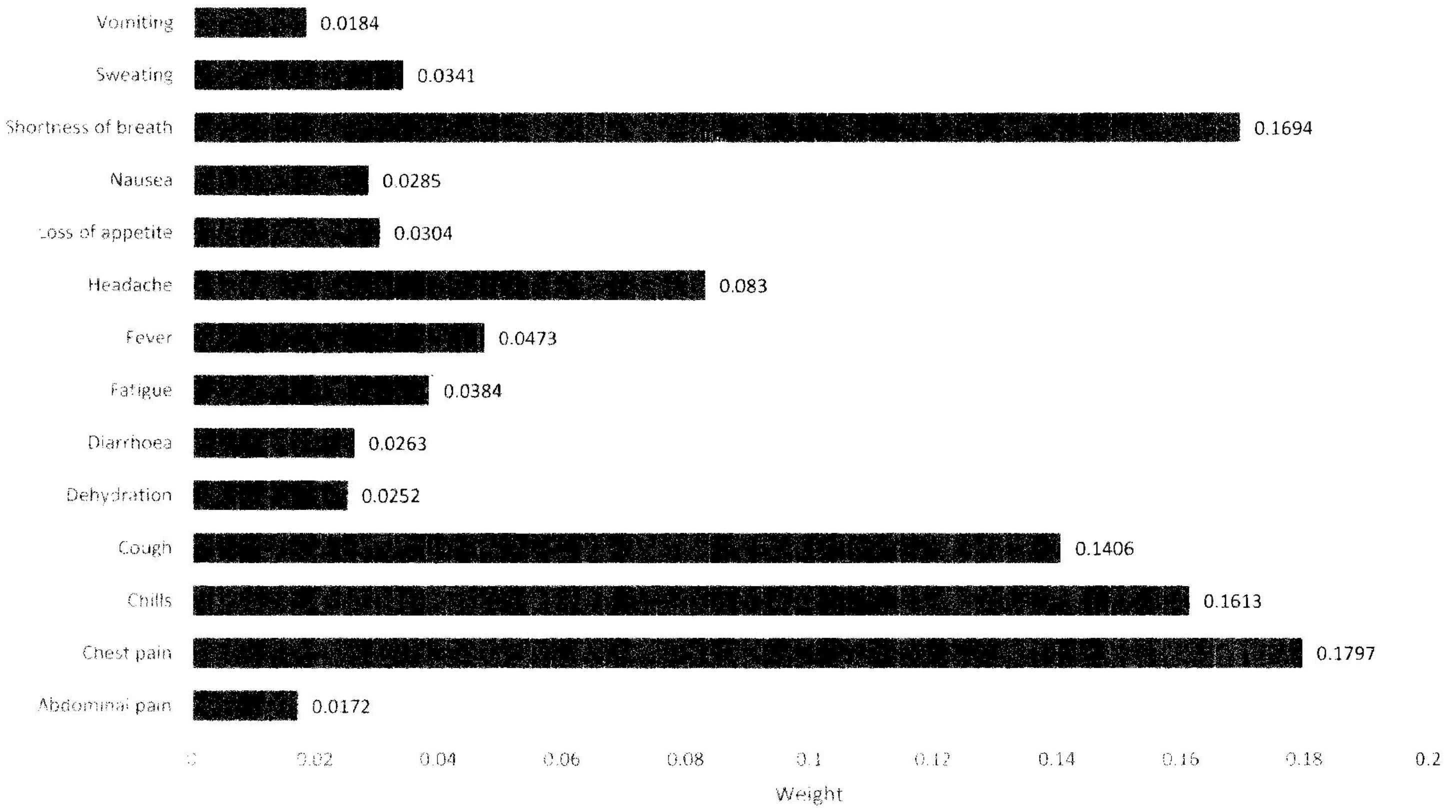


Figure 4.2: Bar chart showing weights generated for pneumonia

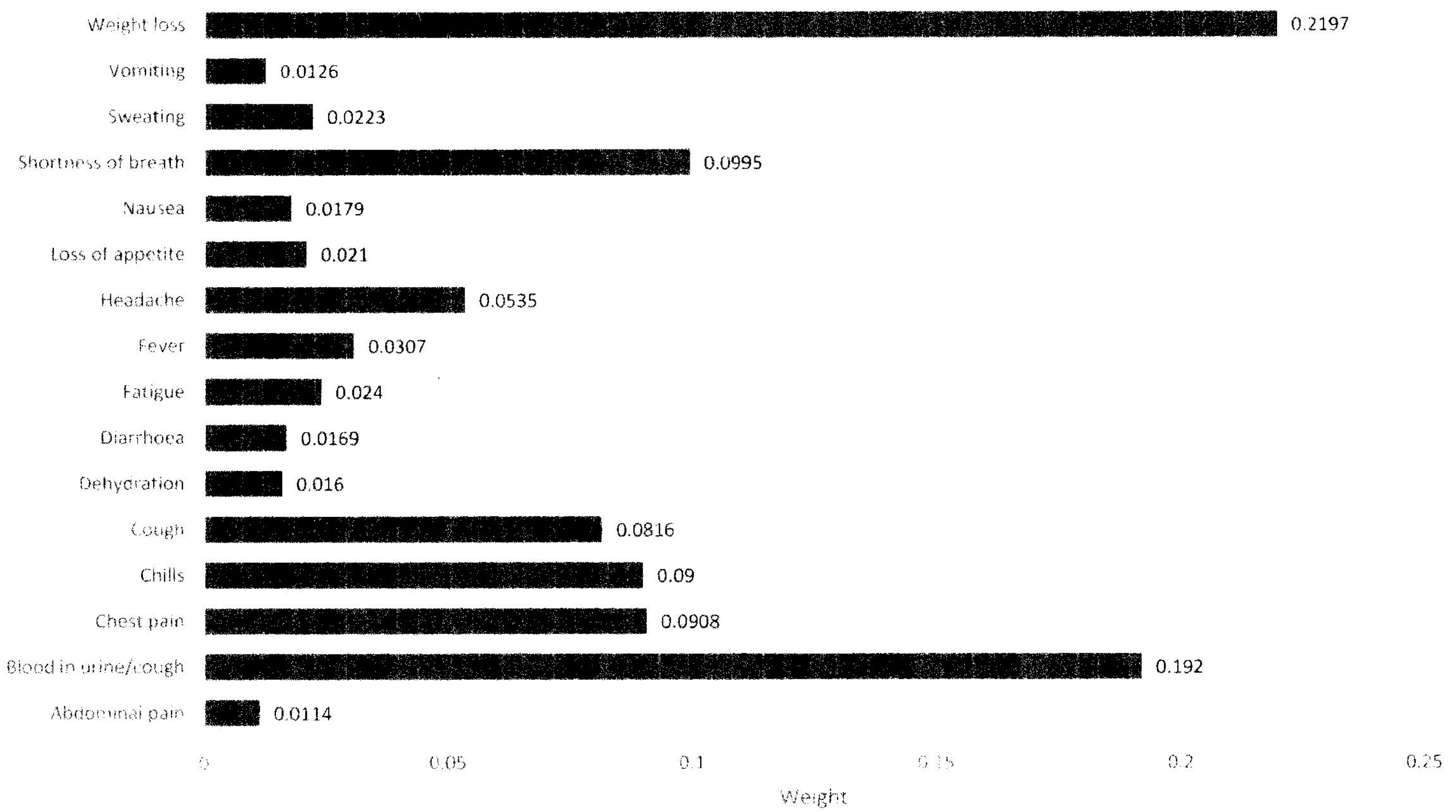


Figure 4.3: Bar chart showing weights generated for tuberculosis

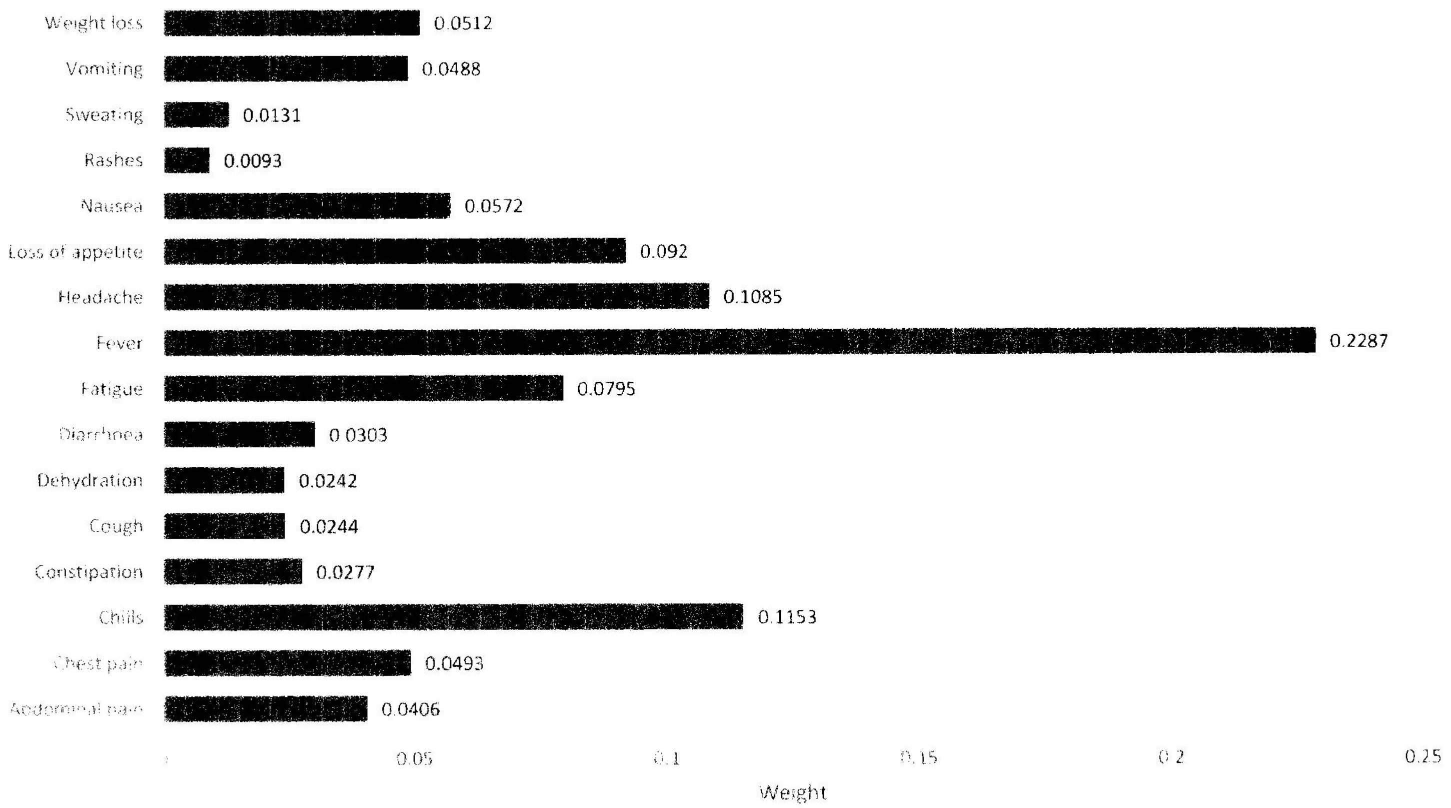


Figure 4.4: Bar chart showing weights generated for typhoid

4.2 RESULT FROM FuzzyTECH

The system was designed and simulated in FuzzyTECH 8.5b professional. FuzzyTECH is a powerful tool for design and simulation of fuzzy systems. The symptoms served as the inputs to the system. The membership functions were set in the software. The system was designed as four Multiple Inputs Single Output system (MISO). Each of the diseases under consideration served as output to each of the MISO systems. The rules were programmed into a rule block. Figure 4.5 shows the inputs, rule blocks and outputs that form the designed fuzzy system.

The system contains four rule blocks named RB_malaria, RB_pneumonia, RB_tuberculosis and RB_typhoid. RB_malaria has 13 inputs, 24 rules and is connected to output malaria. RB_pneumonia has 14 inputs, 21 rules and is connected to output pneumonia. RB_tuberculosis has 16 inputs, 25 rules and is connected to output tuberculosis. RB_typhoid has 16 inputs, 27 rules and is connected to output typhoid.

The fuzzyTECH software was then used to simulate the system. The software provides an interactive debug window where the inputs to the system can be changed while monitoring the output. The value of each input is changed by highlighting it and moving the slider at the bottom-right corner of the window. The window is shown in figure 4.6.

The fuzzyTECH professional software generates 3D graphs to show the relationship between any two symptoms and any of the outputs. Some of the 3D graphs generated are shown in figure 4.7 to 4.10. Figure 4.7 shows how malaria changes with abdominal pain and diarrhoea. Figure 4.8 shows how pneumonia varies with chest pain and cough. Figure 4.9 shows how tuberculosis changes with chest pain and cough. Figure 4.10 shows how typhoid changes with vomiting and fever.

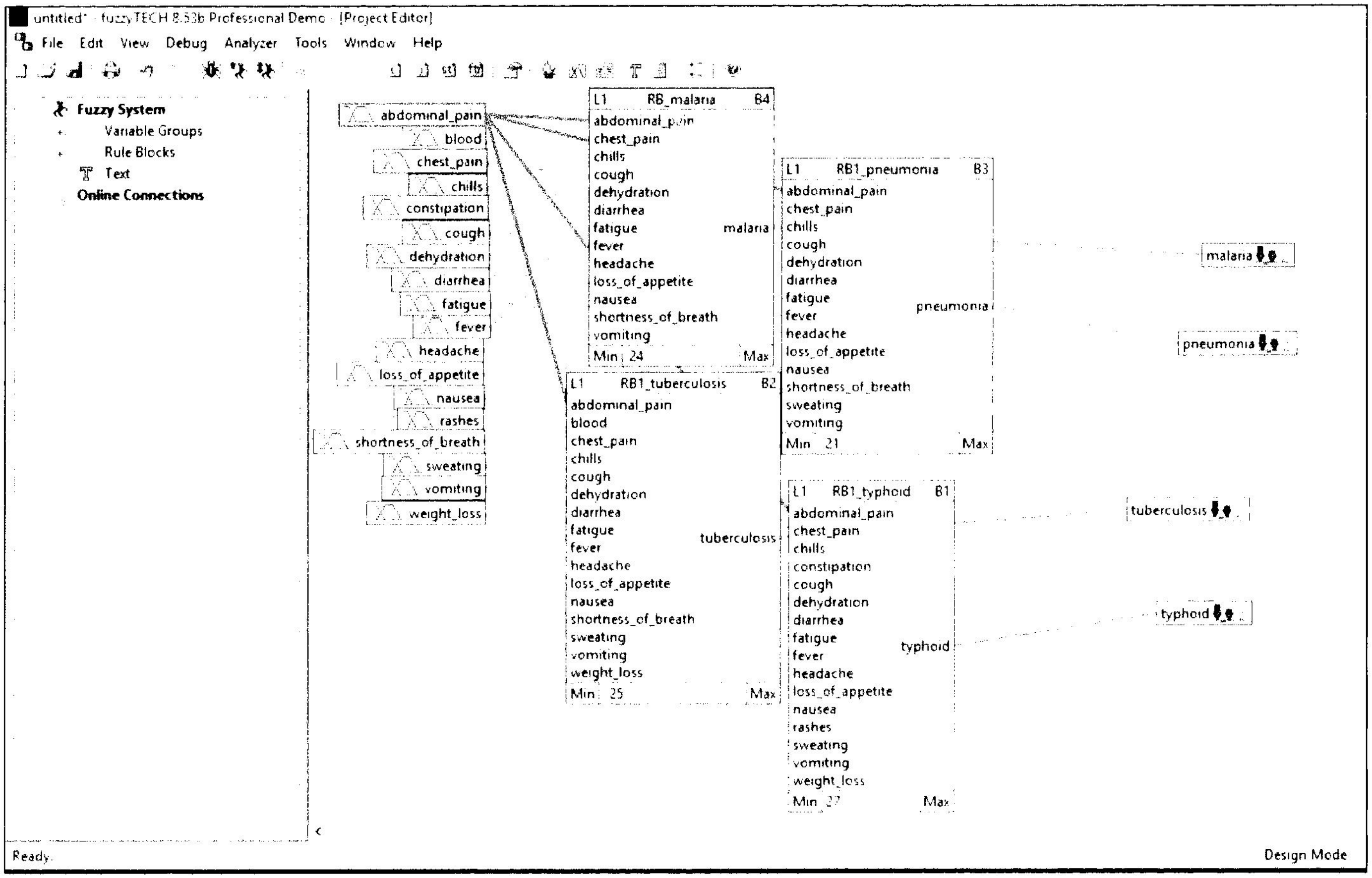
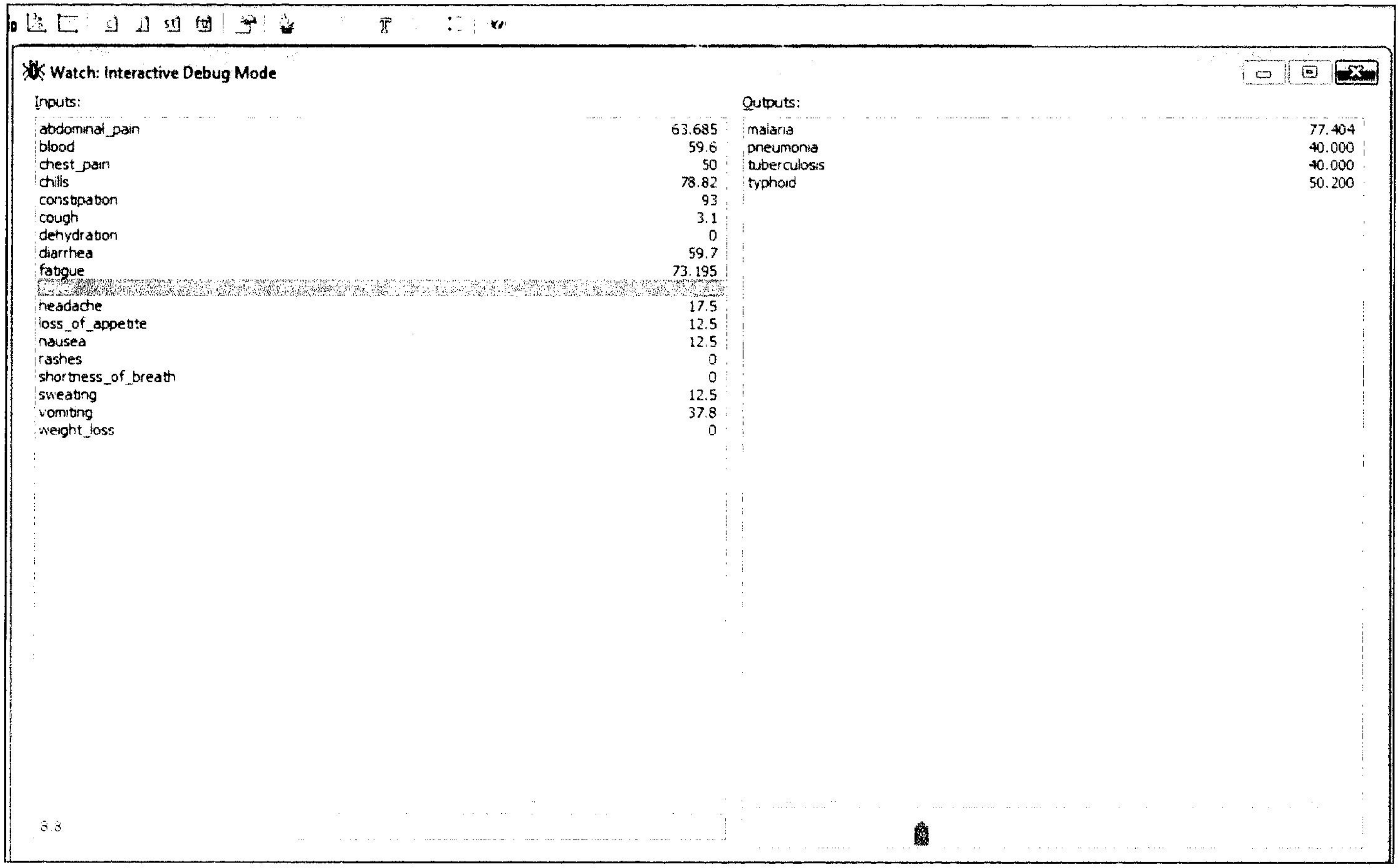


Figure 4.5: The fuzzyTECH software window showing the inputs, rule blocks and outputs.



```
Watch: Interactive Debug Mode

Inputs:                                      Outputs:
abdominal_pain                   63.685      malaria                    77.404
blood                              59.6      pneumonia                  40.000
chest_pain                           50      tuberculosis               40.000
chills                            78.82      typhoid                    50.200
constipation                         93
cough                               3.1
dehydration                           0
diarrhea                           59.7
fatigue                          73.195
[illegible]
headache                           17.5
loss_of_appetite                   12.5
nausea                             12.5
rashes                                0
shortness_of_breath                   0
sweating                           12.5
vomiting                           37.8
weight_loss                           0
```

Figure 4.6: FuzzyTECH's interactive debug mode window.

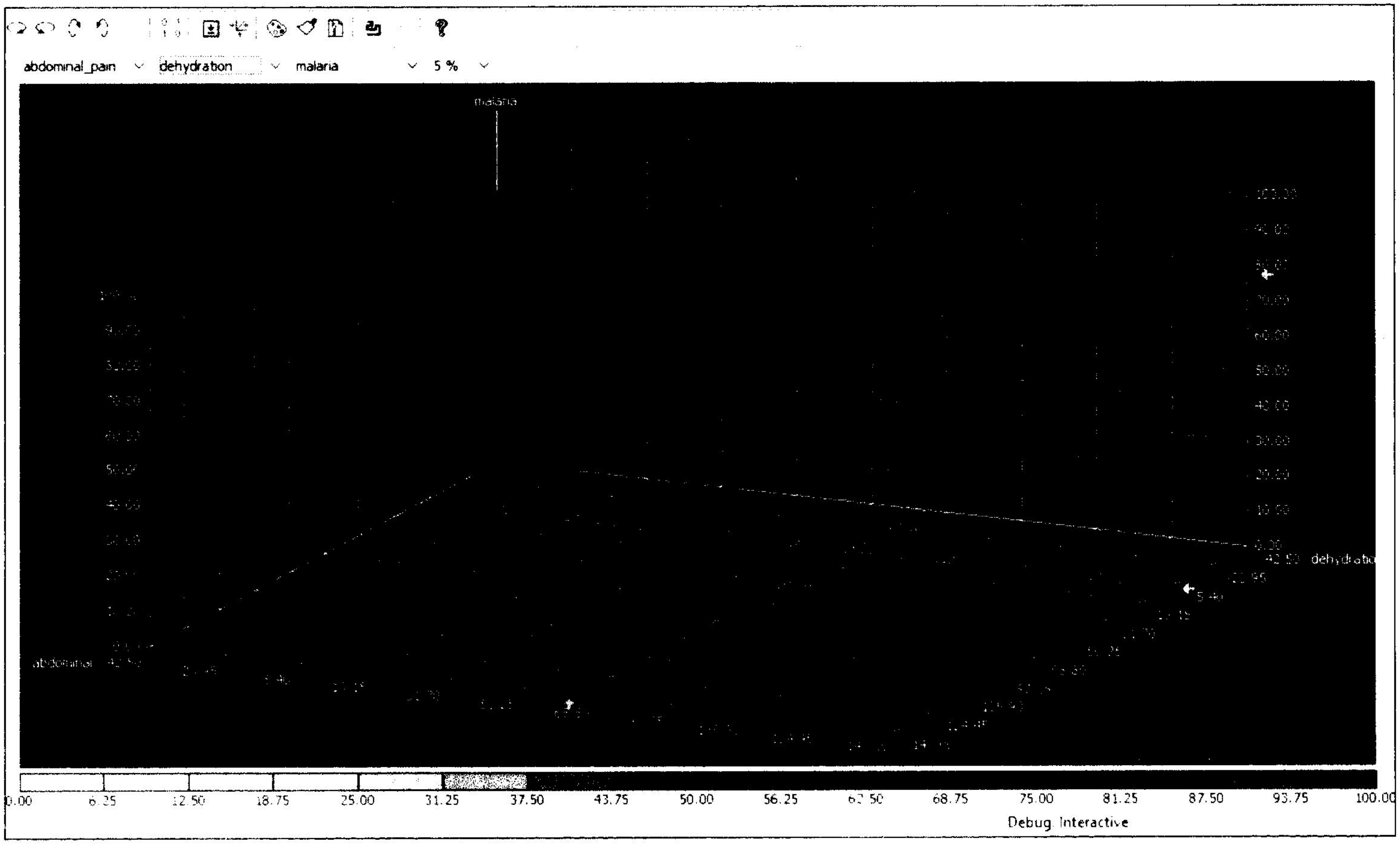


Figure 4.7: FuzzyTECH 3D plot window showing how malaria varies with two inputs.

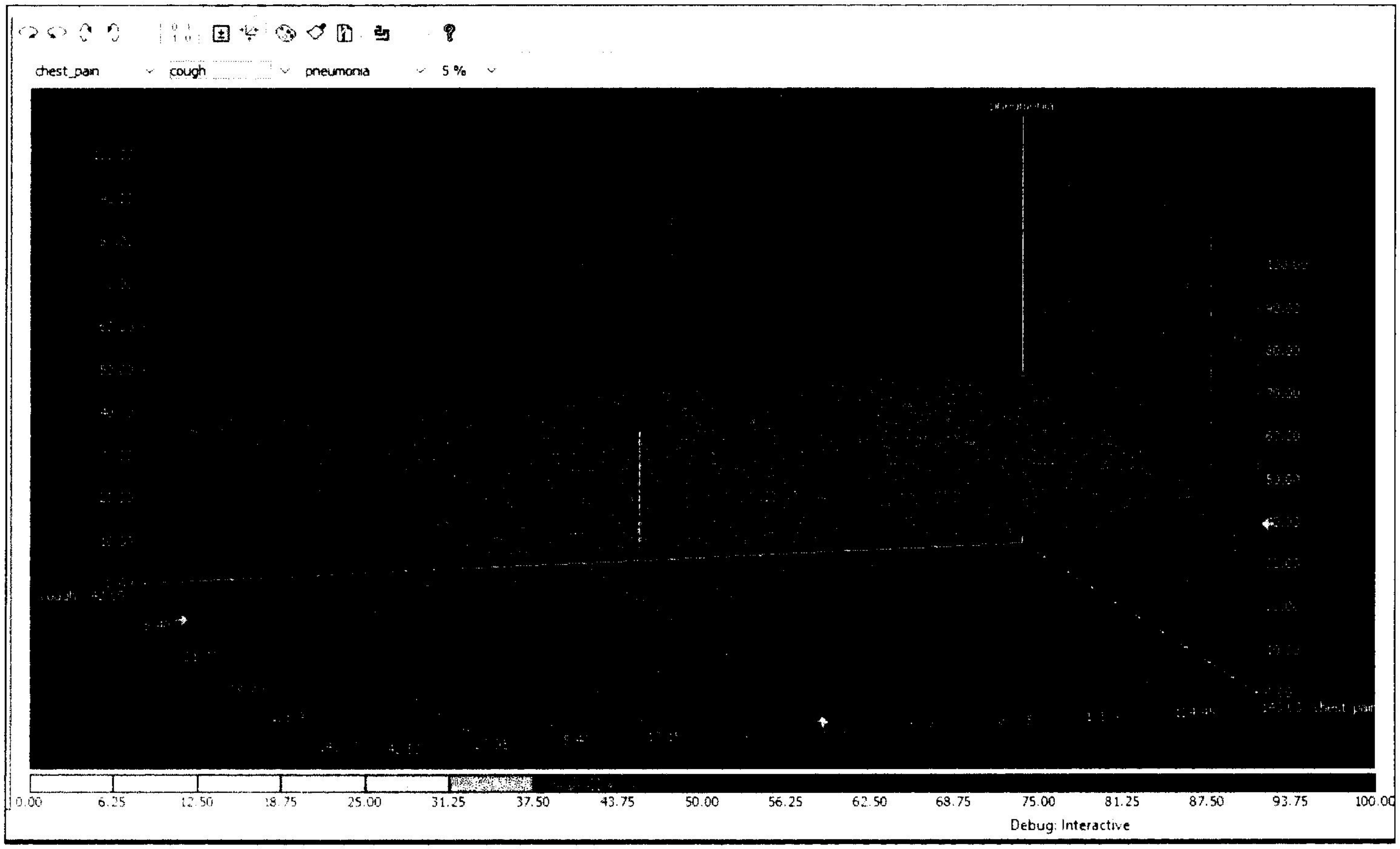


Figure 4.8: FuzzyTECH 3D plot window showing how pneumonia varies with two inputs.

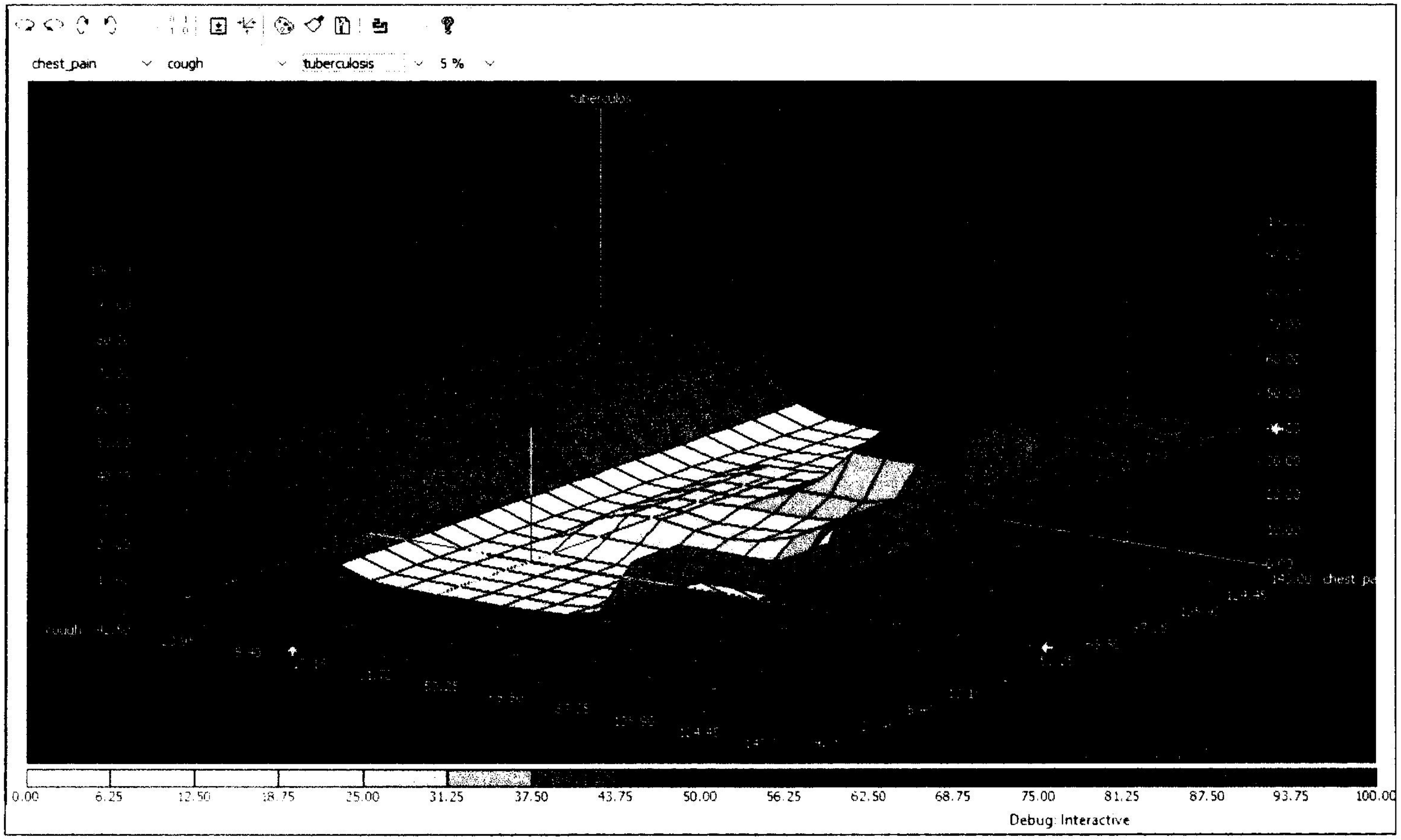


Figure 4.9: FuzzyTECH 3D plot window showing how tuberculosis varies with two inputs.

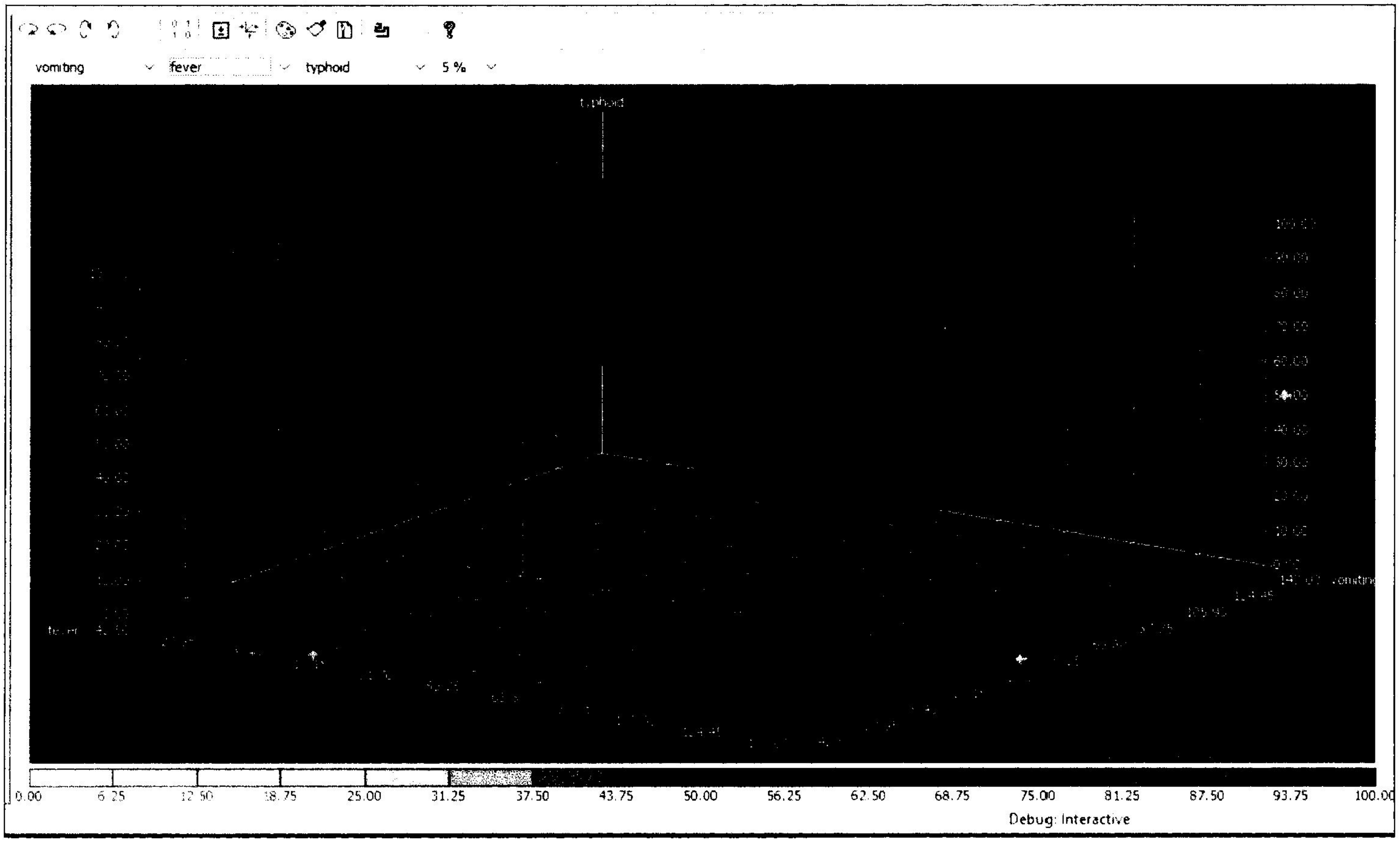


Figure 4.10: FuzzyTECH 3D plot window showing how typhoid varies with two inputs

4.3 RESULT FROM THE DEVELOPED SYSTEM

The system was previously designed and simulated in MATLAB R2015a and FuzzyTECH 8.5b. The system was then implemented on the Android platform. Java programming language was used to program the backend of the system while XML was used to program the user interface. The Development tool used is Android Studio 2.2. When development was done, the application was installed on an Android phone running Android 5.0 (Lollipop). The following screens were captured during the operation of the application.

The Splash screen (Figure 4.11). This screen is the first screen that shows up on starting the application. It contains basic information about the application. The information it contains include the name of the application, the school logo and copyright.

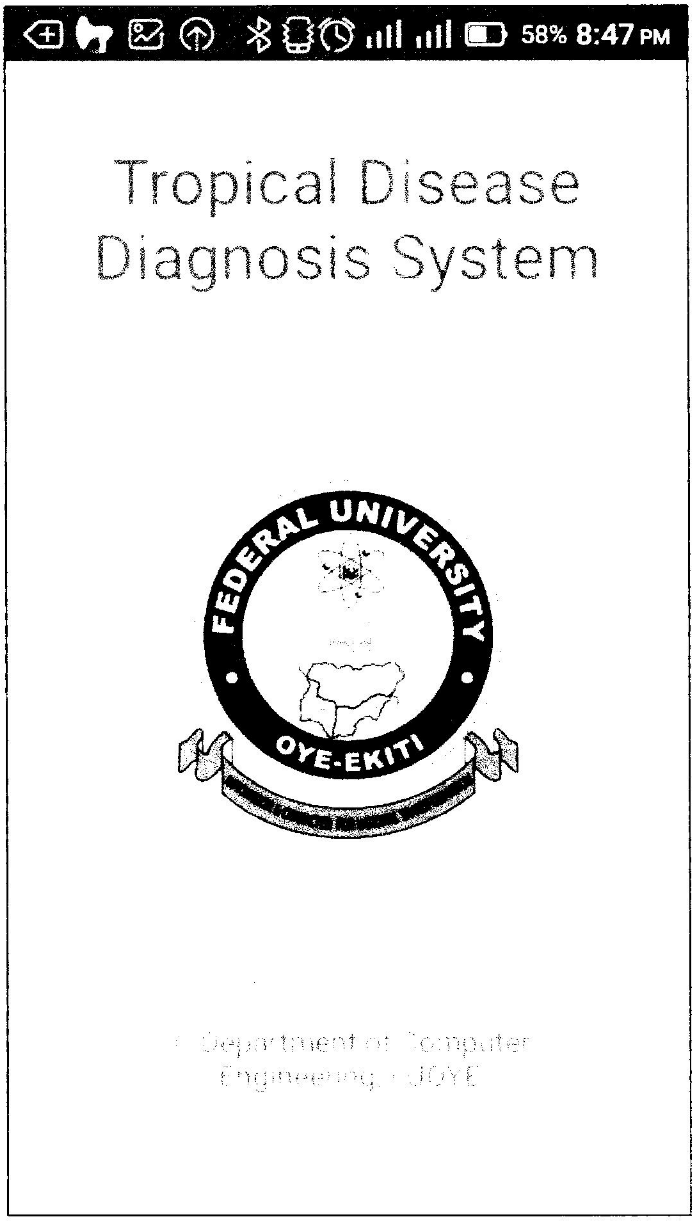
The Login screen (Figure 4.12). This screen requests the user to key in his name, age and gender. These data may be needed in diagnosing some diseases such as tuberculosis and pneumonia. The screen has validation constraint built in. Therefore, if a user leaves any field empty, the application will alert the user and prevent him from moving to the next page.

The select symptom screen (Figure 4.13). This screen requests the user to select one or more symptoms that will serve as inputs to the diagnostic system. This page has validation constraint built in too. Therefore, if a user does not select any symptom, he will be presented with a message requesting him to select one or more symptoms. The application will not go further until at least one symptom is selected.

The confirm symptom screen (Figure 4.14). This screen presents the user with a summary of his choice from the previous screen for the purpose of verification. If the user needs to change any symptom, he clicks the "back" button and makes the changes.

The select symptom weight screen (Figure 4.15). This screen presents the user with the symptoms selected in the select symptom screen one after the other. The user can then move the slider to indicate the intensity of each of the symptoms they have selected.

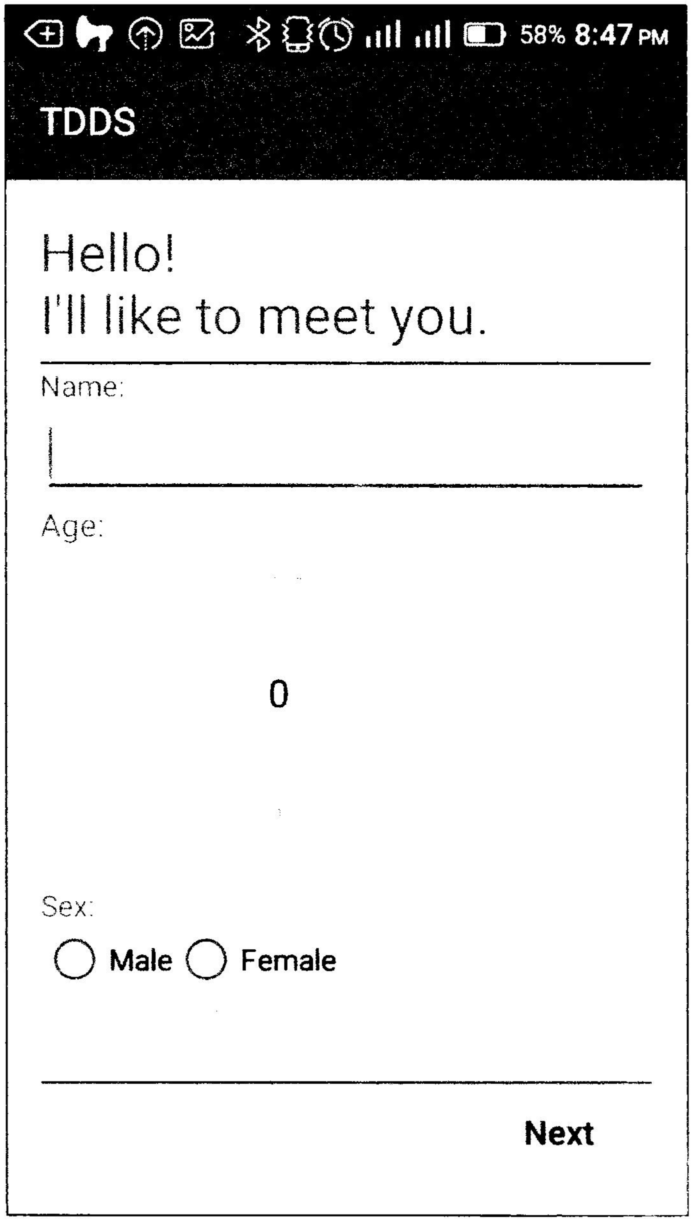
The diagnosis result screen (Figure 4.16). This screen presents the user with the result of the diagnosis arranged in order from the disease with the most probability to that with the lowest.



Tropical Disease Diagnosis System

© Department of Computer Engineering - FUOYE

Figure 4.11: The splash screen of the developed system



Hello!
I'll like to meet you.

Name:

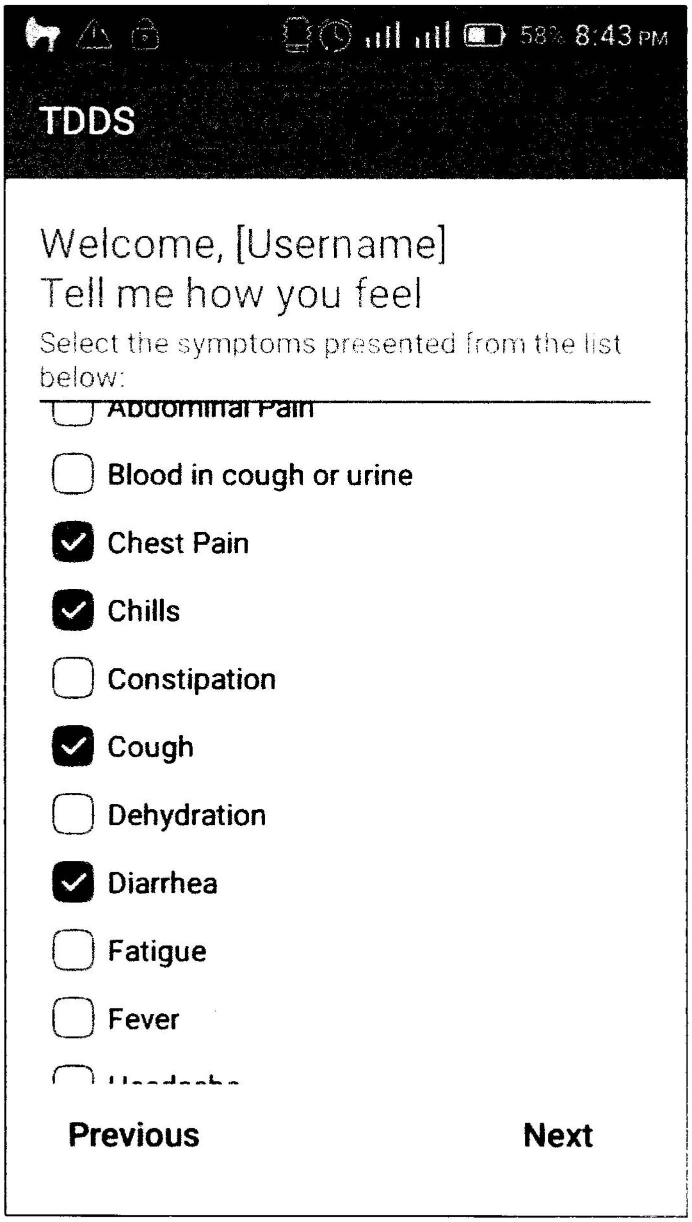
Age:

0

Sex:

○ Male ○ Female

Figure 4.12: The login screen of the developed system.

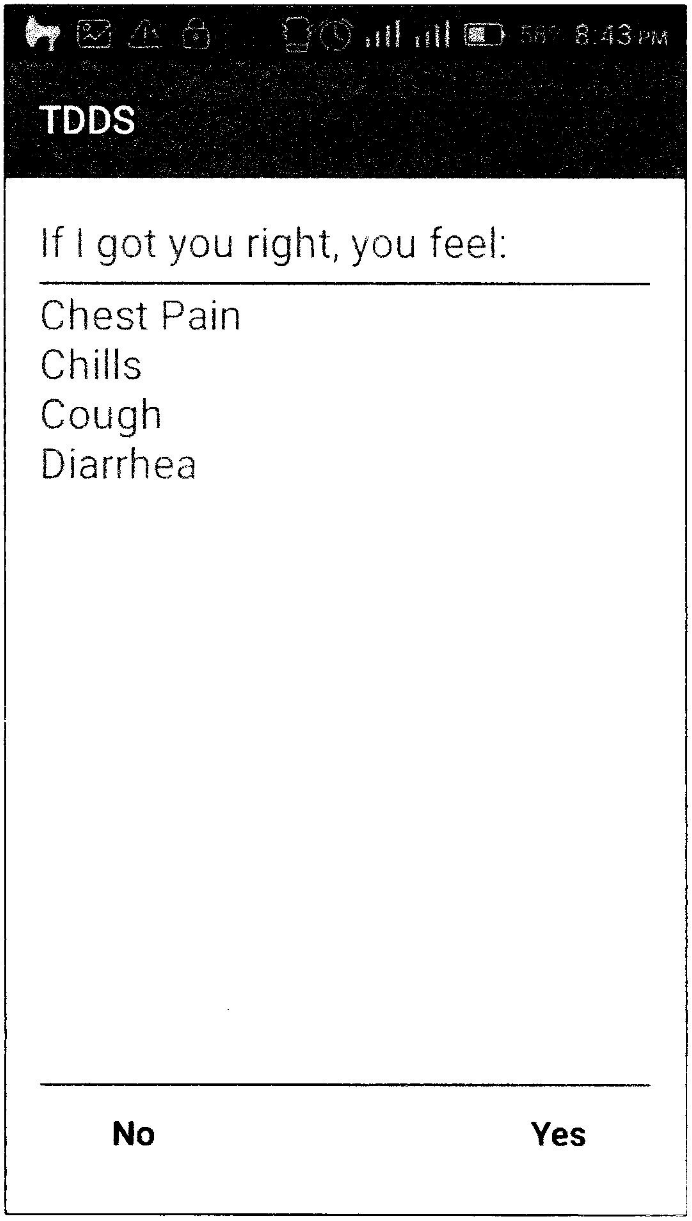


Welcome, [Username]
Tell me how you feel

Select the symptoms presented from the list below:

- ☐ Abdominal Pain
- ☐ Blood in cough or urine
- ☑ Chest Pain
- ☑ Chills
- ☐ Constipation
- ☑ Cough
- ☐ Dehydration
- ☑ Diarrhea
- ☐ Fatigue
- ☐ Fever
- ☐ Headache

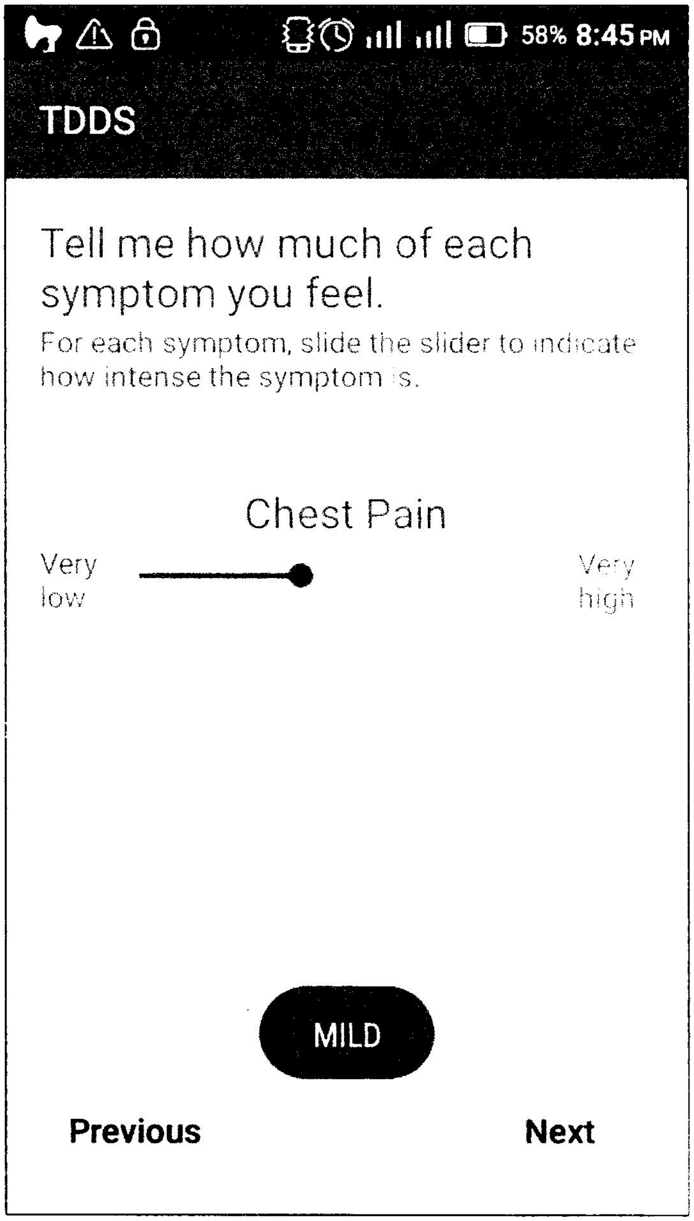
Figure 4.13: The select symptom screen of the developed system



If I got you right, you feel:

Chest Pain
Chills
Cough
Diarrhea

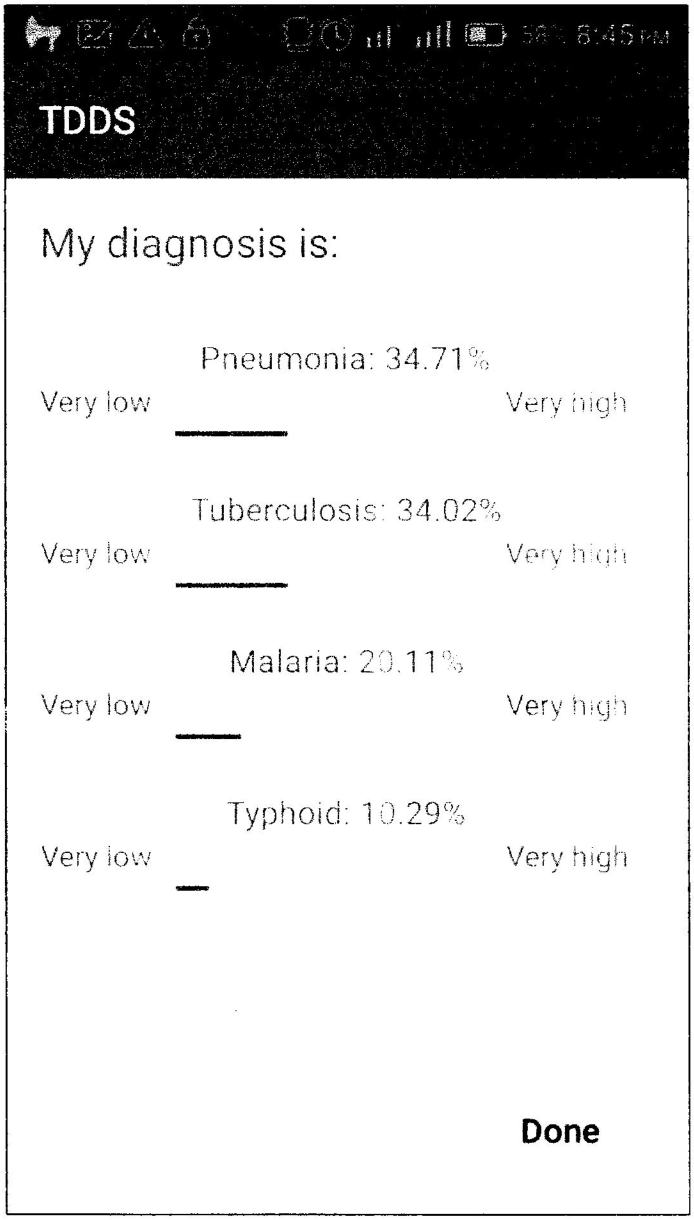
Figure 4.14: The confirmed symptom screen of the developed system.



Tell me how much of each symptom you feel.

For each symptom, slide the slider to indicate how intense the symptom is.

Figure 4.15: The select symptom weight screen of the developed system.



TDDS

My diagnosis is:

Pneumonia: 34.71%

Very low Very high

Tuberculosis: 34.02%

Very low Very high

Malaria: 20.11%

Very low Very high

Typhoid: 10.29%

Very low Very high

Figure 4.16: The diagnosis result screen of the developed system.

4.4 RESULT FROM EVALUATION OF THE DEVELOPED SYSTEM

Evaluation is the systematic determination of the significance of the developed system. The developed system was evaluated by comparing its diagnosis with that of medical doctors using the conventional method. The detailed result of the evaluation is presented in the table in appendix E – System Evaluation. Sensitivity for the two systems was calculated. Sensitivity is given by the formula:

$$sensitivity = \frac{true\ positive\ (TP)}{true\ positive\ (TP) + false\ negative\ (FN)} \tag{4.1}$$

Sensitivity measures the percentage of correctly identified true positive case. The higher the sensitivity of a system, the more accurate the system is. Sensitivity for the developed system was calculated as 72% while sensitivity for the conventional method was calculated as 62%.

The developed system was also evaluated based on the extent of reduction in consulting time. Consulting time is the average time a doctor spends with a patient before diagnosis is completed. If the consulting time is long, doctors will attend to fewer patients resulting in lengthy queues. Conversely, if the consulting time is short, doctors will attend to more patients and queues will reduce. The developed system was demonstrated to reduce consulting time by up to 15%

CHAPTER FIVE
CONCLUSION AND RECOMMENDATIONS

5.0 CONCLUSION

Tropical diseases have a very high mortality rate in countries in the tropics, such as Nigeria. The case of malaria was cited earlier with over a million reported cases resulting in over three hundred thousand deaths yearly. A problem faced by the health sector in Nigeria and other developing countries is the shortage of medical practitioners. The health sector is constantly stressed beyond its carrying capacity. This results in inefficient health care delivery.

Artificial Intelligence is a development in computing that has been applied to solve several problems with interesting results. This prompts the idea of applying artificial intelligence to solve the problems existing in the health sector. Computers are steadily gaining recognition in the health sector. They are being used in the storage, retrieval, sharing and management of biological information to deliver quality healthcare. This has resulted to cost reduction, improved patient safety and improved quality of health care.

This work combined two soft computing techniques, Analytic Hierarchy Process and Fuzzy Logic, to diagnosis four tropical diseases. The tropical diseases considered are malaria, pneumonia, tuberculosis and typhoid. Analytic Hierarchy Process was used to generate membership functions. The rule base was constructed by applying the learning from example technique to expert knowledge collected through structured interviewing. The system was designed using FuzzyTECH 8.5b professional and MATLAB R2015a. The resulting design was implemented successfully on the Android platform. The system was evaluated based on accuracy and reduction of consulting time.

The project has been able to show that hybridization of Fuzzy Logic and Analytic Hierarchy Process can diagnose tropical diseases with an accuracy of up to 72%. The Android application developed in this work can provide a cheap and efficient way of diagnosing tropical diseases even in rural areas. The system can also reduce the consulting time for diagnosing tropical diseases, consequently reducing the physician's workload and increasing their efficiency. Summarily, the system will reduce the stress on the health sector in Nigeria and other developing countries.

5.1 RECOMMENDATION AND FUTURE RESEARCH

This project included just four out of the numerous tropical disease common in Nigeria. Further research to incorporate more diseases in this work is suggested. This will lead to a more robust system. The database of the system should be expanded to contain more symptoms and more rules. This will greatly increase the accuracy of the system.

The fuzzification and defuzzification techniques are very important to the accuracy and performance of a fuzzy inference system. This work did not focus on selecting the optimum fuzzification and defuzzification techniques for the system designed. It is suggested that better techniques for fuzzification and defuzzification be researched and implemented. This will increase the accuracy of the system.

REFERENCES

Akhlaghi, H. & Asadi, H. (2002). Essentials of Telemedicine and Telecare. *BMJ Clinical Research. 324*(7345). 1104-1111.

Al-Hamadani, B. (2016). Chest-Pain Consultation System Based on Fuzzy Semantic Rules. *Computer and Information Science. 9*(3), 26-36.

Al-Shayea, Q.K. (2011). Artificial Neural Networks in Medical Diagnosis. *International Journal of Computer Science Issues. 8*(2). 150-154.

Baesens. B.. Backiel, A. & Broucke, S. (2015). Beginning Java Programming: The Object-Oriented Approach. Indiana, USA: Wiley.

Borgohain, R., Sanyal, S., (2012). Rule Based Expert System for Diagnosis of Neuromuscular Disorders. International Journal of Advanced Networking and Applications. *4*(01), 1509-1513.

Burnside, E.S., Rubin, D.L. & Shachter, R.D. (2004). Using a Bayesian Network to Predict the Probability and Type of Breast Cancer Represented by Microcalcifications on Mammography. Studies in Health Technology and Informatics. *107*(1). 13-17.

Conejar, R.J. & Kim, H. (2014). A Medical Decision Support System (DSS) for Ubiquitous Healthcare Diagnosis System. *International Journal of Software Engineering and Its Applications. 8*(10), 237-244.

Dadios E.P. (Ed). (2012). *Fuzzy Logic: Algorithms, Techniques and Implementations.*

Djam, X.Y.. Wajiga, G. M., Kimbi, Y. H. & Blamah, N.V. (2011). A Fuzzy Expert System for the Management of Malaria. International Journal of Pure and Applied Sciences and Technology. *5*(2), 84-108.

Fieschi, M. (1990). Artificial Intelligence in Medicine: Expert Systems. Translated by D. Cramp. Dartmouth, New Hamshire: Springer.

Gargenta, M. (2011). Learning Android. California, USA: O'Reilly.

Giorgio, A. (2011). Innovative Medical Devices for Telemedicine Applications. In Grschew, G. & Rakowsky, S. (Eds.) *Telemedicine Techniques and Application* (pp. 3-44). Croatia: Intech.

Google. (2019a). Activities. Retrieved from: https://developer.Android.com/guide/components/activities.

Google. (2019b). App Components. Retrieved from: https://developer.Android.com/guide/components.

Gorgulu, O. & Akilli, A. (2016). Use of Fuzzy Logic Based Decision Support Systems in Medicine. *Studies on Ethno-Medicine. 10*(4), 393-403.

Hoyt, R.E., Sutton, M., Yoshihashi, A. (Eds.). (2009). Medical Informatics. Practical Guide for the Healthcare Professional. (3rd Ed.). Lulu.com.

Ishizaka, A. & Nemery, P. (2013). Multi-Criteria Decision Analysis. Methods and Software. West Sussex, United Kingdom: Wiley.

Jaiswal R.S. and Sarode M.V. (2015). *A Survey on Fuzzy Logic Applications in Medical Diagnosis*. International Journal of Computer Science Information and Engineering Technologies. *1*(2), 1-5.

Jones, M.T. (2008). Artificial Intelligence: A systems Approach. Hingham, Massachusetts: Infinity Science Press LLC.

Kadhim, M.A., Alam, M.A. & Kaur, H. (2011). Design and Implementation of Fuzzy Expert. International Journal of Innovative Technology and Creative Engineering. *1*(9), 16-22.

Latifi, F., Hosseini, R. & Mazinai, M. (2015). A Fuzzy Expert System for Diagnosis of Acute Lymphocytic Leukemia in Children. *International Journal of Information, Security and System Management. 4*(2), 424-429.

Lucas, J.F.P., & van der Gaag, L.C. (1991). Principles of Expert Systems. Addison-Wesley.

M. A. Kadhim, M. A. Alam & H. Kaur. (2011). Design and Implementation of Fuzzy expert System of Back Pain Diagnosis. *International Journal of Innovative technology & Creative Engineering. 1*(9), 16-22.

Masic, I., Pandza, H., Kulasin, I., Masic, Z. &Valjevic, S. (2009). Tele-education as Method of Medical Education. *Medical Archive. 63*(6), 350-353.

Mu E. & Pereyra-Rojas M. (2017). *Practical Decision-making using Super Decisions v2. An Introduction to the Analytic Hierarchy Process.* Switzerland: Springer International Publishing AG. pp. 7-21.

Nagarajasri, B. & Padmavathamma, M. (2013). Threshold Neuro Fuzzy Expert System for Diagnosis of Breast Cancer. *International Journal of Computer Applications. 66*(8), 6-10.

Obamiro, J.K. (2013). Effects of Waiting Time on Patient Satisfaction: Nigerian Hospitals Experience. *Journal of Economic Behaviour. 3*, 117-126.

Obi, J.C. & Imianvan, A.A. (2011). Interactive Neuro-Fuzzy Expert System for Diagnosis of Leukemia. *Global Journal of Computer Science and Technology. 11*(12), 43-50.

Obi, M.O., Eke, B.O. & Asagba, P.O. (2018). An Improved Asynchronous Tuberculosis Diagnosis System using Fuzzy Logic Mining Techniques. *International Journal of Computer Trends and Technology. 59*(1), 20-25.

Oche, M.O. & Adamu, H. (2013). Determinants of Patient Waiting Time in the General Outpatient Department of a Tertiary Institution in North Western Nigeria. *Annals of Medical & Health Sciences Research. 3*(4). 588-592.

Passino, K.M. & Yurkovich, S. (1998). Fuzzy Control. California: Addison Wesley Longman.

Patel, R. & Patel,H.R. (2017). A Study on Waiting Time and Out-Patient Satisfaction at Gujarat Medical Education Research Society Hospital, Valsad, Gujarat, India. *International Journal of Community Medicine and Public Health. 4*(3). 857-863.

Peter J.F. Lucas & Linda C. van der Gaag. (1991). *Principles of Expert Systems.* Amsterdam: Addison-Wesley. pp. 3-4.

Ross, T.J. (2010). *Fuzzy Logic with Engineering Applications.* 3rd Ed. West Sussex: John Wiley & Sons, Ltd. pp. 8, 15.

Russell, S.J., Norvig, P. (1995). Artificial Intelligence: a Modern Approach. Eaglewood Cliffs, New Jersey: Prentice Hall.

Saaty, T.L. (1980). The Analytic Hierarchy Process. New York: McGraw-Hill.

Singla, J. (2013). The Diagnosis of Some Lung Diseases in a Prolog Expert System. *International Journal of Computer Applications. 78*(15), 37-40.

Smith, D. & Friesen, J. (2014). *Android Recipes: A Problem-Solution Approach.* (3rd Ed.). New York: Apress.

Tang, Y., Beynon, M.J. (2005). Application and Development of a Fuzzy Analytic Hierarchy Process within a Capital Investment Study. *Journal of Economics and Management. 1*(2), 207-230.

United States Embassy in Nigeria. (2011). *Nigeria Malaria Fact Sheet.* Retrieved from: www.who.int/malaria/publications/country-profiles/profile_nga_en.pdf

Uzoka, F.E., Akinnuwesi, B., Amoo, T., Aladi, F., Fashoto, S., Olaniyan, M. & Osuji, J. (2016). A framework for Early Differential Diagnosis of Tropical Confusable Diseases using the Fuzzy Cognitive Map Engine. *International Journal of Health and Medical Engineering. 10*(2), 346-353.

World Health Organization. (2016a). *Tropical Diseases.* Retrieved from: www.who.int/topics/tropical_diseases.

World Health Organization. (2016b). *Nigeria – key indicators.* Retrieved from: apps.who.int/gho/data/node.cco.ki-NGA.

Zadeh, L. A. (1965). Fuzzy sets, Information and Control. 8, 338-353.

Zimmerman, H.J. (2001). *Fuzzy Set Theory and Its Applications.* (4th Ed.). New York: Springer.

APPENDIX A
QUESTIONNAIRE ON DIAGNOSIS OF TROPICAL DISEASES

Dear Sir/Ma,

This questionnaire seeks to obtain information on the diagnosis of the following tropical diseases: **Malaria, Typhoid Fever, Chicken Pox, Yellow fever, Tuberculosis, Pneumonia and Cholera.** The end result of this study is to develop a decision support system that would aid doctors in efficiently diagnosing these tropical diseases and also attempt to optimize hospital efficiency in terms of the number and duration of patient examinations. The questionnaire is designed to be used alongside interrogation of patients. Your cooperation in completing the questionnaire is highly appreciated.

Bunmi Alegbeleye

(08036565951).

Patient and Physician Demographic Information

1. Age of Patient (yrs.): □ Less than 31. □ 31-40. □ 41-50. □ 51-60. □ Over 60.

2. Patient's gender: □ Male. □ Female.

3. Years of experience of Physician: □ 0-5. □ 6-10. □ 11-15. □ 16-20. □ Over 20.

4. Physician's professional qualification: ______________________________

Symptoms Analysis Diagnosis of Common Tropical Diseases

Based on the initial interrogation, please indicate (by a tick ✓) the **patient's level of presentation** of the following clinical symptoms. Kindly add a confidence (probability) factor for each symptom analysis. The confidence level should be on a **scale of 1-10** (1 being extremely low confidence and 10 being extremely high confidence)

Section A. Clinical Presentation (Symptom Analysis)

Symptoms	Illnesses						
	Absent	Very low	Low	Moderate	High	Very High	Confidence level
Abdominal Pain							
Chest pain							
Chills							
Constipation							
Cough							
Dehydration							
Diarrhea							
Fatigue							
Fever							
Headache							
Jaundice							
Loss of Appetite							
Nausea							
Rashes							
Runny nose							
Shortness of breath							
Vomiting							

Section B. Suspected Disease (diagnosis Hypotheses)

Based on the symptom analysis (for patient), how would you rate the level of manifestation of each of the following diseases (diagnosis hypotheses)? Please, also indicate your level of confidence in each diagnosis hypothesis on a scale of 1-10 (1 being very low confidence and 10 being very high confidence). Note that a patient's symptom could be an indication of none, one or more of the diseases at different levels of intensities.

Disease	Intensity of Attack or Level of Manifestation					Doctor's confidence level
	Very low	Low	Moderate	High	Very High	
Chicken Pox						
Cholera						
Hepatitis B						
Malaria						
Pneumonia						
Tuberculosis						
Typhoid						
Yellow Fever						

How much time in minutes did you require to formulate your diagnosis hypothesis? _____ minutes.

Section C. Further Investigation

Please check where applicable further investigation(s) you carried out to confirm your diagnosis.

Illness	FBC	Blood film	Body Fluid Culture				CXR	AFB
			Sputum	Urine	Stool	Blood		
Chicken Pox								
Cholera								
Hepatitis B								
Malaria								
Pneumonia								
Tuberculosis								
Typhoid								
Yellow Fever								

UTI: Urinary Tract Infection, FBC: Full Blood Count, CXR: Chest X-Ray, AFB: Acid Fast Bacilli

Section D. Confirmed Diagnosis

Based on your further investigation (and without reference to the diagnosis hypotheses), what are your confirmatory findings in terms of the intensity of attack of the following diseases? Also, indicate your level of confidence on a scale of 1-10 (1 being very low confidence and 10 being very high confidence).

Disease	Illnesses					Doctor's confidence level
	Very low	Low	Moderate	High	Very High	
Chicken Pox						
Cholera						
Hepatitis B						
Malaria						
Pneumonia						
Tuberculosis						
Typhoid Fever						
Yellow Fever						

Thank you for taking your time to complete this Questionnaire!

APPENDIX B
RULE BASE

Rule 1: If Chills is very low AND Fatigue is very low AND Fever is very low AND Headache is very low AND Nausea is very low AND Vomiting is very low then malaria is VL.

Rule 2: If Chills is low AND Fatigue is low AND Fever is low AND Headache is low AND Nausea is low AND Vomiting is low then malaria is L.

Rule 3: If Chills is mild AND Fatigue is mild AND Fever is mild AND Headache is mild AND Nausea is mild AND Vomiting is mild then malaria is M.

Rule 4: If Chills is high AND Fatigue is high AND Fever is high AND Headache is high AND Nausea is high AND Vomiting is high then malaria is H.

Rule 5: If Chills is very high AND Fatigue is very high AND Fever is very high AND Headache is very high AND Nausea is very high AND Vomiting is very high then malaria is VH.

Rule 6: If Abdominal Pain is very low AND Chest Pain is very low AND Chills is very low AND Cough is very low AND Dehydration is very low AND Diarrhoea is very low AND Fatigue is very low AND Fever is very low AND Headache is very low AND Loss of Appetite is very low AND Nausea is very low AND Vomiting is very low then malaria is L.

Rule 7: If Abdominal Pain is low AND Chest Pain is low AND Chills is low AND Cough is low AND Dehydration is low AND Diarrhoea is low AND Fatigue is low AND Fever is low AND Headache is low AND Loss of Appetite is low AND Nausea is low AND Vomiting is low then malaria is M.

Rule 8: If Abdominal Pain is mild AND Chest Pain is mild AND Chills is mild AND Cough is mild AND Dehydration is mild AND Diarrhoea is mild AND Fatigue is mild AND Fever is mild AND Headache is mild AND Loss of Appetite is mild AND Nausea is mild AND Vomiting is mild then malaria is H.

Rule 9: If Abdominal Pain is high AND Chest Pain is high AND Chills is high AND Cough is high AND Dehydration is high AND Diarrhoea is high AND Fatigue is high AND Fever is high AND Headache is high AND Loss of Appetite is high AND Nausea is high AND Vomiting is high then malaria is VH.

Rule 10: If Abdominal Pain is very high AND Chest Pain is very high AND Chills is very high AND Cough is very high AND Dehydration is very high AND Diarrhoea is very high AND Fatigue is very high AND Fever is very high AND Headache is very high AND Loss of Appetite is very high AND Nausea is very high AND Vomiting is very high then malaria is M.

Rule 11: If Chest Pain is very low AND Chills is very low AND Cough is very low AND Fatigue is very low AND Fever is very low AND Headache is very low AND Loss of Appetite is very low AND Shortness of Breath is very low AND Sweating is very low then pneumonia is VL.

Rule 12: If Chest Pain is low AND Chills is low AND Cough is low AND Fatigue is low AND Fever is low AND Headache is low AND Loss of Appetite is low AND Shortness of Breath is low AND Sweating is low then pneumonia is L.

Rule 13: If Chest Pain is mild AND Chills is mild AND Cough is mild AND Fatigue is mild AND Fever is mild AND Headache is mild AND Loss of Appetite is mild AND Shortness of Breath is mild AND Sweating is mild then pneumonia is M.

Rule 14: If Chest Pain is high AND Chills is high AND Cough is high AND Fatigue is high AND Fever is high AND Headache is high AND Loss of Appetite is high AND Shortness of Breath is high AND Sweating is high then pneumonia is H.

Rule 15: If Chest Pain is very high AND Chills is very high AND Cough is very high AND Fatigue is very high AND Fever is very high AND Headache is very high AND Loss of Appetite is very high AND Shortness of Breath is very high AND Sweating is very high then pneumonia is VH.

Rule 16: If Chest Pain is very low AND Chills is very low AND Cough is very low AND Dehydration is very low AND Diarrhoea is very low AND Fatigue is very low AND Fever is very low AND Headache is very low AND Loss of Appetite is very low AND Nausea is very low AND Shortness of Breath is very low AND Sweating is very low AND Vomiting is very low then pneumonia is L.

Rule 17: If Chest Pain is low AND Chills is low AND Cough is low AND Dehydration is low AND Diarrhoea is low AND Fatigue is low AND Fever is low AND Headache is low AND Loss of Appetite is low AND Nausea is low AND Shortness of Breath is low AND Sweating is low AND Vomiting is low then pneumonia is M.

Rule 18: If Chest Pain is mild AND Chills is mild AND Cough is mild AND Dehydration is mild AND Diarrhoea is mild AND Fatigue is mild AND Fever is mild AND Headache is mild AND Loss of Appetite is mild AND Nausea is mild AND Shortness of Breath is mild AND Sweating is mild AND Vomiting is mild then pneumonia is H.

Rule 19: If Chest Pain is high AND Chills is high AND Cough is high AND Dehydration is high AND Diarrhoea is high AND Fatigue is high AND Fever is high AND Headache is high AND Loss of Appetite is high AND Nausea is high AND Shortness of Breath is high AND Sweating is high AND Vomiting is high then pneumonia is VH.

Rule 20: If Chest Pain is very high AND Chills is very high AND Cough is very high AND Dehydration is very high AND Diarrhoea is very high AND Fatigue is very high AND Fever is very high AND Headache is very high AND Loss of Appetite is very high AND Nausea is very high AND Shortness of Breath is very high AND Sweating is very high AND Vomiting is very high then pneumonia is VH.

Rule 21: If Chest Pain is very low AND Chills is very low AND Cough is very low AND Fatigue is very low AND Fever is very low AND Loss of Appetite is very low AND Shortness of Breath is very low AND Sweating is very low AND Weight loss is very low then tuberculosis is VL.

Rule 22: If Chest Pain is low AND Chills is low AND Cough is low AND Fatigue is low AND Fever is low AND Loss of Appetite is low AND Shortness of Breath is low AND Sweating is low AND Weight loss is low then tuberculosis is L.

Rule 23: If Chest Pain is mild AND Chills is mild AND Cough is mild AND Fatigue is mild AND Fever is mild AND Loss of Appetite is mild AND Shortness of Breath is mild AND Sweating is mild AND Weight loss is mild then tuberculosis is M.

Rule 24: If Chest Pain is high AND Chills is high AND Cough is high AND Fatigue is high AND Fever is high AND Loss of Appetite is high AND Shortness of Breath is high AND Sweating is high AND Weight loss is high then tuberculosis is H.

Rule 25: If Chest Pain is very high AND Chills is very high AND Cough is very high AND Fatigue is very high AND Fever is very high AND Loss of Appetite is very high AND Shortness of Breath is very high AND Sweating is very high AND Weight loss is very high then tuberculosis is VH.

Rule 26: If Abdominal Pain is very low AND Blood in cough/urine is very low AND Chest Pain is very low AND Chills is very low AND Cough is very low AND Fatigue is very low AND Fever is very low AND Loss of Appetite is very low AND Rashes is very low AND Shortness of Breath is very low AND Sweating is very low then tuberculosis is L.

Rule 27: If Abdominal Pain is low AND Blood in cough/urine is low AND Chest Pain is low AND Chills is low AND Cough is low AND Fatigue is low AND Fever is low AND Loss of Appetite is low AND Rashes is low AND Shortness of Breath is low AND Sweating is low then tuberculosis is M.

Rule 28: If Abdominal Pain is mild AND Blood in cough/urine is mild AND Chest Pain is mild AND Chills is mild AND Cough is mild AND Fatigue is mild AND Fever is mild AND Loss of Appetite is mild AND Rashes is mild AND Shortness of Breath is mild AND Sweating is mild then tuberculosis is H.

Rule 29: If Abdominal Pain is high AND Blood in cough/urine is high AND Chest Pain is high AND Chills is high AND Cough is high AND Fatigue is high AND Fever is high AND Loss of Appetite is high AND Rashes is high AND Shortness of Breath is high AND Sweating is high then tuberculosis is VH.

Rule 30: If Abdominal Pain is very high AND Blood in cough/urine is very high AND Chest Pain is very high AND Chills is very high AND Cough is very high AND Fatigue is very high AND Fever is very high AND Loss of Appetite is very high AND Rashes is very high AND Shortness of Breath is very high AND Sweating is very high then tuberculosis is VH.

Rule 31: If Abdominal Pain is very low AND Constipation is very low AND Diarrhoea is very low AND Fatigue is very low AND Fever is very low AND Headache is very low AND Loss of Appetite is very low then typhoid is VL.

Rule 32: If Abdominal Pain is low AND Constipation is low AND Diarrhoea is low AND Fatigue is low AND Fever is low AND Headache is low AND Loss of Appetite is low then typhoid is L.

Rule 33: If Abdominal Pain is mild AND Constipation is mild AND Diarrhoea is mild AND Fatigue is mild AND Fever is mild AND Headache is mild AND Loss of Appetite is mild then typhoid is M.

Rule 34: If Abdominal Pain is high AND Constipation is high AND Diarrhoea is high AND Fatigue is high AND Fever is high AND Headache is high AND Loss of Appetite is high then typhoid is H.

Rule 35: If Abdominal Pain is very high AND Constipation is very high AND Diarrhoea is very high AND Fatigue is very high AND Fever is very high AND Headache is very high AND Loss of Appetite is very high then typhoid is VH.

Rule 36: If Abdominal Pain is very low AND Chills is very low AND Constipation is very low AND Cough is very low AND Dehydration is very low AND Diarrhoea is very low AND Fatigue is very low AND Fever is very low AND Headache is very low AND Loss of Appetite is very low AND Nausea is very low AND Rashes is very low AND Sweating is very low AND Vomiting is very low AND Weight loss is very low then typhoid is L.

Rule 37: If Abdominal Pain is low AND Chills is low AND Constipation is low AND Cough is low AND Dehydration is low AND Diarrhoea is low AND Fatigue is low AND Fever is low AND Headache is low AND Loss of Appetite is low AND Nausea is low AND Rashes is low AND Sweating is low AND Vomiting is low AND Weight loss is low then typhoid is M.

Rule 38: If Abdominal Pain is mild AND Chills is mild AND Constipation is mild AND Cough is mild AND Dehydration is mild AND Diarrhoea is mild AND Fatigue is mild AND Fever is mild AND Headache is mild AND Loss of Appetite is mild AND Nausea is mild AND Rashes is mild AND Sweating is mild AND Vomiting is mild AND Weight loss is mild then typhoid is H.

Rule 39: If Abdominal Pain is high AND Chills is high AND Constipation is high AND Cough is high AND Dehydration is high AND Diarrhoea is high AND Fatigue is high AND Fever is high AND Headache is high AND Loss of Appetite is high AND Nausea is high AND Rashes is high AND Sweating is high AND Vomiting is high AND Weight loss is high then typhoid is H.

Rule 40: If Abdominal Pain is very high AND Chills is very high AND Constipation is very high AND Cough is very high AND Dehydration is very high AND Diarrhoea is very high AND Fatigue is very high AND Fever is very high AND Headache is very high AND Loss of Appetite is very high AND Nausea is very high AND Rashes is very high AND Sweating is very high AND Vomiting is very high AND Weight loss is very high then typhoid is VH.

Rule 41: If Abdominal Pain is mild AND Chills is low AND Dehydration is low AND Diarrhoea is mild AND Fever is low AND Headache is low AND Nausea is mild AND Rashes is mild AND Vomiting is low then typhoid is H.

Rule 42: If Headache is very low AND Loss of Appetite is very low AND Nausea is very low AND Vomiting is low then malaria is M.

Rule 43: If Cough is mild AND Headache is high AND Loss of Appetite is mild then malaria is H.

Rule 44: If Cough is mild AND Headache is high AND Loss of Appetite is mild then typhoid is M.

Rule 45: If Chest Pain is mild AND Cough is high AND Headache is high then malaria is H.

Rule 46: If Chest Pain is mild AND Cough is high AND Headache is high then typhoid is VL.

Rule 47: If Fatigue is mild AND Fever is high AND Headache is mild AND Loss of Appetite is very high then malaria is H.

Rule 48: If Fatigue is mild AND Fever is high AND Headache is mild AND Loss of Appetite is very high then typhoid is M.

Rule 49: If Fever is very high AND Headache is high then malaria is H.

Rule 50: If Chills is low AND Cough is mild AND Fever is low then tuberculosis is VL.

Rule 51: If Chilis is low AND Cough is mild AND Fever is low then pneumonia is M.

Rule 52: If Dehydration is very low AND Headache is mild AND Loss of Appetite is mild AND Nausea is low AND Vomiting is low then typhoid is L.

Rule 53: If Dehydration is very low AND Headache is mild AND Loss of Appetite is mild AND Nausea is low AND Vomiting is low then malaria is H.

Rule 54: If Cough is very low AND Dehydration is low AND Fatigue is low AND Fever is mild AND Headache is high AND Loss of Appetite is low AND Nausea is low then malaria is H.

Rule 55: If Cough is very low AND Dehydration is low AND Fatigue is low AND Fever is mild AND Headache is high AND Loss of Appetite is low AND Nausea is low then typhoid is L.

Rule 56: If Fatigue is mild AND Fever is very high AND Headache is high then malaria is VH.

Rule 57: If Fatigue is mild AND Fever is very high AND Headache is high then typhoid is M.

Rule 58: If Chest Pain is mild AND Cough is high AND Headache is mild then tuberculosis is M.

Rule 59: If Dehydration is high AND Diarrhoea is very high AND Fever is mild then malaria is M.

Rule 60: If Fever is very high AND Vomiting is very high then malaria is H.

Rule 61: If Fever is mild AND Headache is high AND Loss of Appetite is high then malaria is VH.

Rule 62: If Abdominal Pain is very high AND Fatigue is mild AND Loss of Appetite is high AND Vomiting is high then malaria is VH.

Rule 63: If Abdominal Pain is very high AND Fatigue is mild AND Loss of Appetite is high AND Vomiting is high then typhoid is VH.

Rule 64: If Chest Pain is very high AND Cough is high AND Shortness of Breath is high then tuberculosis is H. If Chest Pain is very high AND Cough is high AND Shortness of Breath is high then tuberculosis is H.

Rule 65: If Chest Pain is very high AND Cough is high AND Shortness of Breath is high then pneumonia is H.

Rule 66: If Abdominal Pain is low AND Nausea is mild AND Vomiting is high then malaria is M.

Rule 67: If Chest Pain is low AND Cough is high AND Shortness of Breath is very low then malaria is M.

Rule 68: If Chest Pain is low AND Cough is high AND Shortness of Breath is very low then tuberculosis is VL.

Rule 69: If Abdominal Pain is low AND Chest Pain is mild AND Fatigue is high AND Fever is high AND Headache is mild AND Loss of Appetite is mild AND Nausea is mild AND Vomiting is low then malaria is H.

APPENDIX C
FUZZY INFERENCE SYSTEM CLASS – CODE LISTING

```
import java.util.*;

import java.util.Arrays;

public class FuzzyInferenceSystem {

   private int[] weight;

   private final int MALARIA = 0;

   private final int PNEUMONIA = 1;

   private final int TUBERCULOSIS = 2;

   private final int TYPHOID = 3;

   // a mxn array that represent the fuzzy rules the system depends on

   private final String rulesDB[][] = {

      {"-", "-", "-", "VL", "-", "-", "-", "-", "VL", "VL", "VL", "-", "VL", "-", "-", "-", "VL", "-", "malaria", "VL"},

      {"-", "-", "-", "L", "-", "-", "-", "-", "L", "L", "L", "-", "L", "-", "-", "-", "L", "-", "malaria", "L"},

      {"-", "-", "-", "M", "-", "-", "-", "-", "M", "M", "M", "-", "M", "-", "-", "-", "M", "-", "malaria", "M"},

      {"-", "-", "-", "H", "-", "-", "-", "-", "H", "H", "H", "-", "H", "-", "-", "-", "H", "-", "malaria", "H"},

      {"-", "-", "-", "VH", "-", "-", "-", "-", "VH", "VH", "VH", "-", "VH", "-", "-", "-", "VH", "-", "malaria", "VH"},

      {"VL", "-", "VL", "VL", "-", "VL", "VL", "VL", "VL", "VL", "VL", "VL", "VL", "-", "-", "-", "VL", "-", "malaria", "L"},

      {"L", "-", "L", "L", "-", "L", "L", "L", "L", "L", "L", "L", "L", "-", "-", "-", "L", "-", "malaria", "M"},

      {"M", "-", "M", "M", "-", "M", "M", "M", "M", "M", "M", "M", "M", "-", "-", "-", "M", "-", "malaria", "H"},

      {"H", "-", "H", "H", "-", "H", "H", "H", "H", "H", "H", "H", "H", "-", "-", "-", "H", "-", "malaria", "VH"},

      {"VH", "-", "VH", "VH", "-", "VH", "VH", "VH", "VH", "VH", "VH", "VH", "VH", "-", "-", "-", "VH", "-", "malaria",
"M"},

      {"-", "-", "VL", "VL", "-", "VL", "-", "-", "VL", "VL", "VL", "VL", "-", "-", "VL", "VL", "-", "-", "pneumonia", "VL"},

      {"-", "-", "L", "L", "-", "L", "-", "-", "L", "L", "L", "L", "-", "-", "L", "L", "-", "-", "pneumonia", "L"},

      {"-", "-", "M", "M", "-", "M", "-", "-", "M", "M", "M", "M", "-", "-", "M", "M", "-", "-", "pneumonia", "M"},

      {"-", "-", "H", "H", "-", "H", "-", "-", "H", "H", "H", "H", "-", "-", "H", "H", "-", "-", "pneumonia", "H"},

      {"-", "-", "VH", "VH", "-", "VH", "-", "-", "VH", "VH", "VH", "VH", "-", "-", "VH", "VH", "-", "-", "pneumonia", "VH"},

      {"-", "-", "VL", "VL", "-", "VL", "VL", "VL", "VL", "VL", "VL", "VL", "VL", "-", "VL", "VL", "VL", "-", "pneumonia",
"L"},
    {"-", "-", "L", "L", "-", "L", "L", "L", "L", "L", "L", "L", "L", "-", "L", "L", "L", "-", "pneumonia", "M"},

    {"-", "-", "M", "M", "-", "M", "M", "M", "M", "M", "M", "M", "M", "-", "M", "M", "M", "-", "pneumonia", "H"},

    {"-", "-", "H", "H", "-", "H", "H", "H", "H", "H", "H", "H", "H", "-", "H", "H", "H", "-", "pneumonia", "VH"},

    {"-", "-", "VH", "VH", "-", "VH", "VH", "VH", "VH", "VH", "VH", "VH", "VH", "-", "VH", "VH", "VH", "-", "pneumonia",
"VH"},

    {"-", "-", "VL", "VL", "-", "VL", "-", "-", "VL", "VL", "-", "VL", "-", "-", "VL", "VL", "-", "VL", "tuberculosis", "VL"},

    {"-", "-", "L", "L", "-", "L", "-", "-", "L", "L", "-", "L", "-", "-", "L", "L", "-", "L", "tuberculosis", "L"},

    {"-", "-", "M", "M", "-", "M", "-", "-", "M", "M", "-", "M", "-", "-", "M", "M", "-", "M", "tuberculosis", "M"},

    {"-", "-", "H", "H", "-", "H", "-", "-", "H", "H", "-", "H", "-", "-", "H", "H", "-", "H", "tuberculosis", "H"},

    {"-", "-", "VH", "VH", "-", "VH", "-", "-", "VH", "VH", "-", "VH", "-", "-", "VH", "VH", "-", "VH", "tuberculosis",
"VH"},

    {"VL", "VL", "VL", "VL", "-", "VL", "-", "-", "VL", "VL", "-", "VL", "-", "VL", "VL", "VL", "-", "-", "tuberculosis", "L"},

    {"L", "L", "L", "L", "-", "L", "-", "-", "L", "L", "-", "L", "-", "L", "L", "L", "-", "-", "tuberculosis", "M"},

    {"M", "M", "M", "M", "-", "M", "-", "-", "M", "M", "-", "M", "-", "M", "M", "M", "-", "-", "tuberculosis", "H"},

    {"H", "H", "H", "H", "-", "H", "-", "-", "H", "H", "-", "H", "-", "H", "H", "H", "-", "-", "tuberculosis", "VH"},

    {"VH", "VH", "VH", "VH", "-", "VH", "-", "-", "VH", "VH", "-", "VH", "-", "VH", "VH", "VH", "-", "-", "tuberculosis",
"VH"},

    {"VL", "-", "-", "-", "VL", "-", "-", "VL", "VL", "VL", "VL", "VL", "-", "-", "-", "-", "-", "-", "typhoid", "VL"},

    {"L", "-", "-", "-", "L", "-", "-", "L", "L", "L", "L", "L", "-", "-", "-", "-", "-", "-", "typhoid", "L"},

    {"M", "-", "-", "-", "M", "-", "-", "M", "M", "M", "M", "M", "-", "-", "-", "-", "-", "-", "typhoid", "M"},

    {"H", "-", "-", "-", "H", "-", "-", "H", "H", "H", "H", "H", "-", "-", "-", "-", "-", "-", "typhoid", "H"},

    {"VH", "-", "-", "-", "VH", "-", "-", "VH", "VH", "VH", "VH", "VH", "-", "-", "-", "-", "-", "-", "typhoid", "VH"},

    {"VL", "-", "-", "VL", "VL", "VL", "VL", "VL", "VL", "VL", "VL", "VL", "VL", "VL", "-", "VL", "VL", "VL", "typhoid", "L"},

    {"L", "-", "-", "L", "L", "L", "L", "L", "L", "L", "L", "L", "L", "L", "-", "L", "L", "L", "typhoid", "M"},

    {"M", "-", "-", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "M", "-", "M", "M", "M", "typhoid", "H"},

    {"H", "-", "-", "H", "H", "H", "H", "H", "H", "H", "H", "H", "H", "H", "-", "H", "H", "H", "typhoid", "H"},

    {"VH", "-", "-", "VH", "VH", "VH", "VH", "VH", "VH", "VH", "VH", "VH", "VH", "VH", "-", "VH", "VH", "VH", "typhoid",
"VH"},

    {"M", "-", "-", "L", "-", "-", "L", "M", "-", "L", "L", "-", "M", "M", "-", "-", "L", "-", "typhoid", "H"},//1

    {"-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "VL", "VL", "VL", "-", "-", "-", "L", "-", "malaria", "M"},

    {"-", "-", "-", "-", "-", "M", "-", "-", "-", "-", "H", "M", "-", "-", "-", "-", "-", "-", "malaria", "H"},
    {"-", "-", "-", "-", "-", "M", "-", "-", "-", "-", "H", "M", "-", "-", "-", "-", "-", "-", "typhoid", "M"},
    {"-", "-", "M", "-", "-", "H", "-", "-", "-", "-", "H", "-", "-", "-", "-", "-", "-", "-", "malaria", "H"},
    {"-", "-", "M", "-", "-", "H", "-", "-", "-", "-", "H", "-", "-", "-", "-", "-", "-", "-", "typhoid", "VL"},
    {"-", "-", "-", "-", "-", "-", "-", "-", "M", "H", "M", "VH", "-", "-", "-", "-", "-", "-", "malaria", "H"},
    {"-", "-", "-", "-", "-", "-", "-", "-", "M", "H", "M", "VH", "-", "-", "-", "-", "-", "-", "typhoid", "M"},
    {"-", "-", "-", "-", "-", "-", "-", "-", "-", "VH", "H", "-", "-", "-", "-", "-", "-", "-", "malaria", "H"},//6
    {"-", "-", "-", "L", "-", "M", "-", "-", "-", "L", "-", "-", "-", "-", "-", "-", "-", "-", "tuberculosis", "VL"},
    {"-", "-", "-", "L", "-", "M", "-", "-", "-", "L", "-", "-", "-", "-", "-", "-", "-", "-", "pneumonia", "M"},
    {"-", "-", "-", "-", "-", "-", "VL", "-", "-", "-", "M", "M", "L", "-", "-", "-", "L", "-", "typhoid", "L"},
    {"-", "-", "-", "-", "-", "-", "VL", "-", "-", "-", "M", "M", "L", "-", "-", "-", "L", "-", "malaria", "H"},
    {"-", "-", "-", "-", "-", "VL", "L", "-", "L", "M", "H", "L", "L", "-", "-", "-", "-", "-", "malaria", "H"},//9
    {"-", "-", "-", "-", "-", "VL", "L", "-", "L", "M", "H", "L", "L", "-", "-", "-", "-", "-", "typhoid", "L"},//9
    {"-", "-", "-", "-", "-", "-", "-", "-", "M", "VH", "H", "-", "-", "-", "-", "-", "-", "-", "malaria", "VH"},
    {"-", "-", "-", "-", "-", "-", "-", "-", "M", "VH", "H", "-", "-", "-", "-", "-", "-", "-", "typhoid", "M"},
    {"-", "-", "M", "-", "-", "H", "-", "-", "-", "-", "M", "-", "-", "-", "-", "-", "-", "-", "tuberculosis", "M"},
    {"-", "-", "-", "-", "-", "-", "H", "VH", "-", "M", "-", "-", "-", "-", "-", "-", "-", "-", "malaria", "M"},
    {"-", "-", "-", "-", "-", "-", "-", "-", "-", "VH", "-", "-", "-", "-", "-", "-", "VH", "-", "malaria", "H"},
    {"-", "-", "-", "-", "-", "-", "-", "-", "-", "M", "H", "H", "-", "-", "-", "-", "-", "-", "malaria", "VH"},
    {"VH", "-", "-", "-", "-", "-", "-", "-", "M", "-", "-", "H", "-", "-", "-", "-", "H", "-", "malaria", "VH"},//15
    {"VH", "-", "-", "-", "-", "-", "-", "-", "M", "-", "-", "H", "-", "-", "-", "-", "H", "-", "typhoid", "VH"},
    {"-", "-", "VH", "-", "-", "H", "-", "-", "-", "-", "-", "-", "-", "-", "H", "-", "-", "-", "tuberculosis", "H"},//16
    {"-", "-", "VH", "-", "-", "H", "-", "-", "-", "-", "-", "-", "-", "-", "H", "-", "-", "-", "pneumonia", "H"},//16
    {"L", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "-", "M", "-", "-", "-", "H", "-", "malaria", "M"},
    {"-", "-", "L", "-", "-", "H", "-", "-", "-", "-", "-", "-", "-", "-", "VL", "-", "-", "-", "malaria", "M"},
    {"-", "-", "L", "-", "-", "H", "-", "-", "-", "-", "-", "-", "-", "-", "VL", "-", "-", "-", "tuberculosis", "VL"},
    {"L", "-", "M", "-", "-", "-", "-", "-", "H", "H", "M", "M", "M", "-", "-", "-", "L", "-", "malaria", "H"} };
//this array holds the points of the membership functions
double mf[][] = {
    {-42.5, -5, 0, 37.5}, //VL
    {-16, 21.5, 26.5, 64}, //L
```

```
        {10.5, 48, 53, 90.5}, //M
        {36.75, 74.25, 79.25, 116.75}, //H
        {63, 100.5, 105.5, 143} //VH
    };

    double eigenValues[][] = {
        {0.0336, 0.0000, 0.0456, 0.1219, 0.0000, 0.0260, 0.0215, 0.0223, 0.0697, 0.2692, 0.1266, 0.1176, 0.0696,
0.0000, 0.0228, 0.0000, 0.0536, 0.0000}, //malaria
        {0.0172, 0.0000, 0.1797, 0.1613, 0.0000, 0.1406, 0.0252, 0.0263, 0.0384, 0.0473, 0.0830, 0.0304, 0.0285,
0.0000, 0.1694, 0.0341, 0.0184, 0.0000}, //pneumonia
        {0.0114, 0.1920, 0.0908, 0.0900, 0.0000, 0.0816, 0.0160, 0.0169, 0.0240, 0.0307, 0.0535, 0.0210, 0.0179,
0.0000, 0.0995, 0.0223, 0.0126, 0.2197}, //tuberculosis
        {0.0406, 0.0000, 0.0493, 0.1153, 0.0277, 0.0244, 0.0242, 0.0303, 0.0795, 0.2287, 0.1085, 0.0920, 0.0572,
0.0093, 0.0000, 0.0131, 0.0488, 0.0512} //typhoid
    };
    public FuzzyInferenceSystem(int[] weight) {
        this.weight = weight;
    }

    private double Fuzzifier(String ling_var, int symptom_weight) {
        //setting up the membership function matrix reference
        int X = 0;
        switch (ling_var) {
            case "VL":
                X = 0;
                break;
            case "L":
                X = 1;
                break;
            case "M":
                X = 2;
                break;
            case "H":
                X = 3;
                break;
            case "VH":
                X = 4;
                break;
        }
        int a = 0;
        int b = 1;
        int c = 2;
        int d = 3;
        int w = symptom_weight;
        double i = 0;
        double j = 0;
        i = ((w - mf[X][a]) / (mf[X][b] - mf[X][a]));
        j = ((mf[X][d] - w) / (mf[X][d] - mf[X][c]));
        return Math.max(Math.min(Math.min(i, 1), j), 0);
    }

    private double Defuzzifier(double[] aggregate) {
        double numerator = 0;
        double denominator = 0;
        double[] centre = {-2.5, 24, 50.5, 76.75, 103};
        for (int i = 0; i < 5; i++) {
            numerator = numerator + (integral(aggregate[i]) * centre[i]);
            denominator = denominator + integral(aggregate[i]);
        }
        return Math.min(100, (numerator / denominator));
    }
```

```
private double Compositor(List<Double> md) { //average
    Iterator iterator = md.iterator();
    double sum = 0;
    while (iterator.hasNext()) {
        sum = sum + (double) iterator.next();
    }
    double ave = sum / md.size();
    return ave;
}

private double Normalizer(double degree, int symptom, String disease) {
    int X = -1;
    switch (disease) {
        case "malaria":
            X = 0;
            break;
        case "pneumonia":
            X = 1;
            break;
        case "tuberculosis":
            X = 2;
            break;
        case "typhoid":
            X = 3;
            break;
    }

    return degree * eigenValues[X][symptom];
}
```

```
private double[] Aggregator(List<Object[]> md) {
  Iterator iterator = md.iterator();
  double sum[] = {0, 0, 0, 0, 0};
  double counter[] = {0, 0, 0, 0, 0};
  while (iterator.hasNext()) {
    Object[] set = (Object[]) iterator.next();
    switch ((String) set[0]) {
      case "VL":
        sum[0] = sum[0] + (double) set[1];
        counter[0]++;
        break;
      case "L":
        sum[1] = sum[1] + (double) set[1];
        counter[1]++;
        break;
      case "M":
        sum[2] = sum[2] + (double) set[1];
        counter[2]++;
        break;
      case "H":
        sum[3] = sum[3] + (double) set[1];
        counter[3]++;
        break;
      case "VH":
        sum[4] = sum[4] + (double) set[1];
        counter[4]++;
        break;

    }
```

```
        }
        double[] average = new double[5];
        for (int i = 0; i < 5; i++) {
            average[i] = sum[i] / counter[i];
        }
        return average;
    }

    private List InferenceEngine() {
        List<ArrayList<Object[]>> inference = new ArrayList<>();
        ArrayList<Object[]> malaria = new ArrayList<>();
        ArrayList<Object[]> pneumonia = new ArrayList<>();
        ArrayList<Object[]> tuberculosis = new ArrayList<>();
        ArrayList<Object[]> typhoid = new ArrayList<>();
        for (int i = 0; i < rulesDB.length; i++) { //loops through each rule
            List<Double> md = new ArrayList<Double>(); //temporary holding for each rule
            for (int j = 0; j < rulesDB[i].length - 2; j++) { //loops through each statement of each rule
                String ling_var = rulesDB[i][j];
                String disease = rulesDB[i][18];
                if (!ling_var.equals("-")) {
                    if (weight[j] >= 0) {
                        md.add(Normalizer(Fuzzifier(ling_var, weight[j]), j, disease));
                    } else {
                        md.add(0.0);
                    }
                }
            }
            if (Compositor(md) > 0) {
                Object[] fuzzySet = new Object[2];
                fuzzySet[0] = rulesDB[i][19];  //severity
                fuzzySet[1] = Compositor(md);
                switch (rulesDB[i][18]) {
                    case "malaria":
                        malaria.add(fuzzySet);
                        break;
                    case "pneumonia":
                        pneumonia.add(fuzzySet);
                        break;
                    case "tuberculosis":
                        tuberculosis.add(fuzzySet);
                        break;
                    case "typhoid":
                        typhoid.add(fuzzySet);
                        break;
                }
            }
        }
        inference.add(malaria);
        inference.add(pneumonia);
        inference.add(tuberculosis);
        inference.add(typhoid);
        return inference;
    }

    public double integral(double h) {
        return (80 * (h - ((15 * h * h) / 32)));
    }
    public int[] order(double[] temp) {
        int[] order = new int[4];
        for (int i = 0; i <= 3; i++) {
            double current = temp[i];
            int position = 1;
            for (int j = 0; j <= 3; j++) {
                if (current < temp[j]) {
                    position++;
                }
            }
            order[i] = position;
        }
        return order;
    }
    public double[] DiagnosisManager() {
        List<ArrayList<Object[]>> inference = InferenceEngine();
        double[] diagnosis = new double[4];
        diagnosis[MALARIA] = Defuzzifier(Aggregator(inference.get(MALARIA)));
        diagnosis[PNEUMONIA] = Defuzzifier(Aggregator(inference.get(PNEUMONIA)));
        diagnosis[TUBERCULOSIS] = Defuzzifier(Aggregator(inference.get(TUBERCULOSIS)));
        diagnosis[TYPHOID] = Defuzzifier(Aggregator(inference.get(TYPHOID)));
        return diagnosis;
    }
}
```

APPENDIX D
MATLAB SCRIPT

```
function eig = normalize(x)
norm = zeros(length(x));
        for i = 1:length(x)
          for j = 1:length(x)
            norm(i,j) = x(i,j)/sum(x(:,j));
          end
        end
norm
eig = zeros(length(x), 1);
        for k = 1:length(x)
          eig(k,1) = sum(norm(k,:))/length(x);
        end
end
```

APPENDIX E
SYSTEM EVALUATION

S/N	Initial hypothesis	Confirmed diagnosis	Accuracy (Doctor)	FAHP diagnosis	Accuracy (FAHP)
1.	Typhoid-M	Typhoid-H	0	Typhoid-H(69)	1
2.	Malaria-H	Malaria-H	1	Malaria-H(81)	1
		Typhoid-M	0	Typhoid-M(52)	1
3.	Malaria-H	Malaria-H	1	Malaria-H(65)	1
	Pneumonia-M	Tuberculosis-VL	0	Tuberculosis-M(55)	0
4.		Malaria-VH	0	Malaria-L(32)	0
5.	Malaria-H	Malaria-H	1	Malaria-H(79)	1
		Typhoid-M	0	Typhoid-M(64)	1
6.	Malaria-H	Malaria-H	1	Malaria-H(67)	1
7.	Pneumonia-M	Pneumonia-M	1	Pneumonia-M(44)	1
		Tuberculosis-VL	0	Tuberculosis-VL(10)	1
8.	Malaria-H	Malaria-H	1	Malaria-H(90)	1
		Typhoid-L	0	Typhoid-M(55)	0
9.	Malaria-H	Malaria-H	1	Malaria-H(90)	1
	Typhoid-L	Typhoid-L	1	Typhoid-M(59)	0
10.	Malaria-H	Malaria-VH	0	Malaria-H(97)	1
		Typhoid-M	0	Typhoid-M(63)	1
11.	Malaria-H	Tuberculosis-M	0	Tuberculosis-M(55)	1
	Pneumonia-M				
12.	Malaria-M	Malaria-M	1	Malaria-L(38)	1
13.	Malaria-VH	Malaria-VH	1	Malaria-M(46)	0
14.	Malaria-VH	Malaria-VH	1	Malaria-VH(80)	1
15.	Typhoid-VH	Malaria-VH	0	Malaria-VH(71)	1
		Typhoid-VH	1	Typhoid-VH(85)	1
16.	Pneumonia-VH	Tuberculosis-H	0	Pneumonia-H(69)	1
		Pneumonia-H	0	Tuberculosis-H(76)	1
17.	Malaria-M	Malaria-M	1	Malaria-M(37)	1
18.	Malaria-M	Malaria-M	1	Tuberculosis-M(42)	0
19.	Malaria-H	Malaria-H	1	Malaria-H(90)	1
20.	Malaria-VH	Malaria-VH	1	Malaria-L(35)	0
	Typhoid-H	Typhoid-H	1	Pneumonia-L(11)	1
	Tuberculosis-VL	Tuberculosis-L	0	Tuberculosis-L(17)	1
		Pneumonia-L	0	Typhoid-H(66)	1
21.	Malaria-VH	Malaria-VH	1	Malaria-M(58)	0
	Typhoid-H	Typhoid-VH	0	Pneumonia-L(38)	1
	Tuberculosis-L	Tuberculosis-L	1	Tuberculosis-L(34)	1
	Pneumonia-M	Pneumonia-M	1	Typhoid-M(50)	0

S/N	Initial hypothesis	Confirmed diagnosis	Accuracy (Doctor)	FAHP diagnosis	Accuracy (FAHP)
22.	Malaria-VH	Malaria-H	0	Malaria-L(26)	0
	Typhoid-H	Typhoid-H	1	Pneumonia-L(35)	1
	Tuberculosis-VL	Tuberculosis-VL	1	Tuberculosis-VL(10)	1
	Pneumonia-L	Pneumonia-L	0	Typhoid-L(17)	0
23.	Malaria-VH	Malaria-VH	1	Malaria-M(50)	0
	Typhoid-H	Typhoid-VH	0	Pneumonia-M(38)	1
	Tuberculosis-L	Tuberculosis-L	1	Tuberculosis-L(28)	1
	Pneumonia-M	Pneumonia-M	1	Typhoid-L(31)	0
24.	Malaria-VH	Malaria-VH	1	Malaria-VH(90)	1
	Typhoid-H	Typhoid-H	1	Pneumonia-M(41)	1
	Tuberculosis-L	Tuberculosis-L	1	Tuberculosis-L(34)	1
	Pneumonia-M	Pneumonia-M	1	Typhoid-H(65)	1
25.	Malaria-VH	Malaria-VH	1	Malaria-H(92)	1
	Typhoid-VH	Typhoid-H	0	Pneumonia-M(62)	0
	Tuberculosis-VL	Tuberculosis-L	0	Tuberculosis-L(32)	1
	Pneumonia-L	Pneumonia-L	1	Typhoid-H(65)	1
26.	Malaria-VH	Malaria-VH	1	Malaria-VH(92)	1
	Typhoid-VH	Typhoid-H	0	Pneumonia-L(32)	1
	Tuberculosis-L	Tuberculosis-L	1	Tuberculosis-L(25)	1
	Pneumonia-M	Pneumonia-L	1	Typhoid-M(44)	0
27.	Malaria-VH	Malaria-VH	1	Malaria-L(30)	0
	Typhoid-H	Typhoid-H	1	Pneumonia-L(33)	1
	Tuberculosis-L	Tuberculosis-L	1	Tuberculosis-L(27)	1
	Pneumonia-L	Pneumonia-L	1	Typhoid-L(14)	0
28.	Malaria-VH	Malaria-VH	1	Malaria-VH(92)	1
	Typhoid-H	Typhoid-H	1	Pneumonia-L(35)	1
	Tuberculosis-VL	Tuberculosis-L	0	Tuberculosis-L(34)	1
	Pneumonia-M	Pneumonia-L	0	Typhoid-H(65)	1
29.	Malaria-VH	Malaria-VH	1	Malaria-VH(98)	1
	Typhoid-H	Typhoid-H	1	Pneumonia-M(56)	0
	Tuberculosis-L	Tuberculosis-L	0	Tuberculosis-M(60)	0
	Pneumonia-M	Pneumonia-L	0	Typhoid-H(90)	1

S/N	Initial hypothesis	Confirmed diagnosis	Accuracy (Doctor)	FAHP diagnosis	Accuracy (FAHP)

22.	Malaria-VH
	Typhoid-H
	Tuberculosis-VL
	Pneumonia-L
		Malaria-H
		Typhoid-H
		Tuberculosis-VL
		Pneumonia-L
			0
			1
			1
			0
				Malaria-L(26)
				Pneumonia-L(35)
				Tuberculosis-VL(10)
				Typhoid-L(17)
					0
					1
					1
					0

23.	Malaria-VH
	Typhoid-H
	Tuberculosis-L
	Pneumonia-M
		Malaria-VH
		Typhoid-VH
		Tuberculosis-L
		Pneumonia-M
			1
			0
			1
			1
				Malaria-M(50)
				Pneumonia-M(38)
				Tuberculosis-L(28)
				Typhoid-L(31)
					0
					1
					1
					0

24.	Malaria-VH
	Typhoid-H
	Tuberculosis-L
	Pneumonia-M
		Malaria-VH
		Typhoid-H
		Tuberculosis-L
		Pneumonia-M
			1
			1
			1
			1
				Malaria-VH(90)
				Pneumonia-M(41)
				Tuberculosis-L(34)
				Typhoid-H(65)
					1
					1
					1
					1

25.	Malaria-VH
	Typhoid-VH
	Tuberculosis-VL
	Pneumonia-L
		Malaria-VH
		Typhoid-H
		Tuberculosis-L
		Pneumonia-L
			1
			0
			0
			1
				Malaria-H(92)
				Pneumonia-M(62)
				Tuberculosis-L(32)
				Typhoid-H(65)
					1
					0
					1
					1

26.	Malaria-VH
	Typhoid-VH
	Tuberculosis-L
	Pneumonia-M
		Malaria-VH
		Typhoid-H
		Tuberculosis-L
		Pneumonia-L
			1
			0
			1
			1
				Malaria-VH(92)
				Pneumonia-L(32)
				Tuberculosis-L(25)
				Typhoid-M(44)
					1
					1
					1
					0

27.	Malaria-VH
	Typhoid-H
	Tuberculosis-L
	Pneumonia-L
		Malaria-VH
		Typhoid-H
		Tuberculosis-L
		Pneumonia-L
			1
			1
			1
			1
				Malaria-L(30)
				Pneumonia-L(33)
				Tuberculosis-L(27)
				Typhoid-L(14)
					0
					1
					1
					0

28.	Malaria-VH
	Typhoid-H
	Tuberculosis-VL
	Pneumonia-M
		Malaria-VH
		Typhoid-H
		Tuberculosis-L
		Pneumonia-L
			1
			1
			0
			0
				Malaria-VH(92)
				Pneumonia-L(35)
				Tuberculosis-L(34)
				Typhoid-H(65)
					1
					1
					1
					1

29.	Malaria-VH
	Typhoid-H
	Tuberculosis-L
	Pneumonia-M
		Malaria-VH
		Typhoid-H
		Tuberculosis-L
		Pneumonia-L
			1
			1
			0
			0
				Malaria-VH(98)
				Pneumonia-M(56)
				Tuberculosis-M(60)
				Typhoid-H(90)
					1
					0
					0
					1