APPLICATION OF AI IN MEDICINE

Submitted by
AMIRUL HISYAM BIN TARMIZI
WEK990171

A FINAL YEAR PROJECT REPORT

Submitted to the
Faculty of Computer Science and Information Technology
University of Malaya
Dissertation submitted in partial fulfilment of the requirement for the
Degree of Computer Science

ABSTRACT

From the beginning of human civilization, mankinds have been trying to find a solution on how to make life easier. They create, modify, and adapt, as they would please their need. From the smallest things in our life to the biggest part of all – technologies, has improved our life tremendously. Computer technology nowadays has an important role in education, communication, medicine, and many more. In the field of medicine, we have seen many support systems that had been developed and used to help human in their daily lives'. Application of AI in medicine was proposed with this reason in mind, as a hope that it can contribute towards the important of human life.

ACKNOWLEDGEMENT

All the praise for god Al-Mighty for given me the strength and confidence, patient and guidance, enough and ample time to complete my final project.

I would like to take this opportunity to express my deepest gratitude to Dr Sameem Abdul Kareem for being so supportive in helping and guiding me with this thesis. I would also like to thank Dr Selvanathan for giving me much of precious comments that I find very helpful and meaningful.

To my family, I would like to thank them for all their support and love. Last but not least were to my friends for their support and ideas in developing this thesis and to all the people who had helped me in this project.

May God bless you.

-AMIRUL HISYAM-

TABLE OF CONTENTS

Chapter 2

Literature Review

Miscellaneous

LIST OF FIGURE

LIST OF TABLE

CHAPTER I

INTRODUCTION

Chapter 1: Introduction

'evolution is a natural way to program'

-Thomas Ray-

For the past 30 years, computer applications in the field of medicine has rapidly evolved from a machine that help doctors in making decisions to a machine that can make decisions, and as their name implies, it performs as well as an expert. Expert systems are ruled-based system, which makes decisions based on rule programmed in the system. One of the major drawback of an expert system is it cannot evolve naturally after the attainment of new knowledge. Due to the static nature of expert system, computer researchers have to think of other better ways in making the system more flexible, so that it can evolve on its own, and after the addition of new data.

In the field artificial intelligence, researchers are always thinking of new way that can make a computer as smart as a human. Based on evolutionary theory by Darwin, evolutionary computing is proposed. Genetic algorithm, genetic programming, and evolutionary programming were born with the hope that it can be used to realize this dream.

In medical research based involving artificial intelligence, genetic algorithm has shown a remarkable performance, especially in areas, such as the diagnosis of cancer. in cancer, prognosis, however there are to many factors involved.

1.1 Project Overview

The proposed system is a survival analysis system that predicts the progress of Nasopharyngeal carcinoma (NPC) patients. This system will be developed by using genetic algorithm and neural network based on data provided by UMMC. This system will act as a support system to help doctors in making prognosis and decisions.

1.2 Prognosis

Prognosis is the process of selectively gathering information concerning a patient, and interpreting it accordingly to previous knowledge, in order to predict the future development of the patient's condition (Moshe Sipper).

1.3 Overview of Nasopharyngeal Carcinoma

Nasopharyngeal cancer is a disease in which malignant (cancer) cells form in the tissues of the nasopharynx. The nasopharynx is the upper part of the pharynx (throat) behind the nose. The pharynx is a hollow tube about 5 inches long that starts behind the nose and ends at the top of the trachea (windpipe) and esophagus (the tube that goes from the throat to the stomach). Air and food pass through the pharynx on the way to the trachea or the esophagus. The nostrils lead into the nasopharynx. An opening on each side of the nasopharynx leads into an ear. Nasopharyngeal cancer most commonly starts in the squamous cells that line the oropharynx (the part of the throat behind the mouth).

1.4 Overview Of Genetic Algorithm

Genetic algorithm was introduced by John Holland in his book ***Adaptation in Natural and*** *Artificial*[ ] He showed how the evolutionary process can be applied to solved a wide variety of problems using a highly parallel technique. This algorithm was based on Darwin's theory of evolution. The genetic algorithm transform a population of individual objects, each with associated fitness value, into a new generation of the population using the Darwinian principle of reproduction and survival of the fittest and naturally occurring genetic operation such as crossover (recombination) and mutation. Each individual in the population represents a possible solution to a given problem. The genetic algorithm attempts to find a very good or best solution to the problem by genetically breeding the population of individual.

1.5 Problem Definition

Behind all development of a system lay some problems that have to be overcome. In this system, some of the obstacles are how the system should be trained to give a correct prognosis result when it comes to rare categories of cancer.

1.6 Project Motivation

This system uses the domain of NPC because of high incidence of NPC in Malaysia. With the accessibility of the NPC data provided from UMMC this system can be work on smoothly and all the prediction can be made as close as possible to expert because of availability of data.

Genetic Algorithm is use to determine the number of input variables that can be use to predict the cancer. While the hybrid system consist of Genetic Algorithm and neural network will be use to improve prediction.

1.7 Objectives

The objectives of this project are to:

I. Create a support system for helping doctors in Nasopharyngeal carcinoma (NPC) prognosis

II. To use hybrid model of genetic algorithm and neural network as a tool to predict the prognosis of Nasopharyngeal carcinoma

III. The research will act as a stepping stone for similar researches in medical prognosis

1.8 Project Scope

I. Research is carried out in the domain of medical cancer, namely, NPC

II. This system will be developed based on evolutionary computing, I.e. genetic algorithm and artificial neural network

III. Only authenticated users are allowed to access the database.

IV. Users will get permission to insert data, but all critical data manipulating like deleting, modifying will have to get a permission from higher level users like doctors.

V. This system will use English as a primary standard

VI. This project consist of two major components, a database and processing unit using genetic algorithm and two minor components, an interface for input and output.

1.9 System Development Life Cycle Model

This is also known as Classic Life Cycle Model (or) Linear Sequential Model (or) Waterfall Method.

System/Information Engineering and Modeling

As software is always of a large system (or business), work begins by establishing requirements for all system elements and then allocating some subset of these requirements to software. This system view is essential when software must interface with other elements such as hardware, people and other resources. System is the basic and very critical requirement for the existence of software in any entity. So if the system is not in place, the system should be engineered and put in place. In some cases to extract the maximum output, system should be re-engineered and spiced up. Once the ideal system is engineered or tuned up, the development team studies the software requirement for the system.

Software Requirements Analysis

This is also known as feasibility study. In this phase, the development team visits the customer and studies their system. They investigate the need for possible software automation in the given system. By the end of the feasibility study, the team furnishes a document that holds the different specific recommendations for the candidate system. It also includes the personnel assignments, costs, project schedule, and target dates. The requirements gathering process is intensified and focussed specially on software. To understand the nature of the program(s) to be built, the system engineer ("analyst") must understand the information domain for the software, as well as required function, behavior, performance and interfacing. The essential purpose of this phase is to find the need and to define the problem that needs to be solved.

System Analysis and Design

In this phase, the software's overall structure and its nuances are defined. In terms of the client/server technology, the number of tiers needed for the package architecture, the database design, the data structure designs etc are all defined in this phase. Analysis and Design are very crucial in the whole development cycle. Any glitch in the design phase could be very expensive to solve in the later stage of the software development. Much care is taken during this phase. The logical system of the product is developed in this phase.

Code Generation

The design must be translated into a machine-readable form. The code generation step performs this task. If design is performed in a detailed manner, code generation can be accomplished with out much complication. Programming tools like Compilers, Interpreters, Debuggers are used to generate the code. Different high level programming languages like C, C++, Pascal, Java are used for coding. With respect to the type of application, the right programming language is chosen.

Testing

Once the code is generated, the program testing begins. Different testing methodologies are available to unravel the bugs that were committed during the previous phases. Different testing tools and methodologies are already available. Some companies built their own testing tools that are tailor made for their own development operations.

Maintenance

Software will definitely undergo change once it is delivered to the customer. There are many reasons for the change. Change could happen because of some unexpected input values into the system. In addition, the changes in the system could directly affect the software operations. The software should be developed to accommodate changes that could happen during the post implementation period.

1.10 Project Schedule

	JUN 2002	JULY 2002	AUG 2002	SEPT 2002	OCT 2002	NOV 2002	DEC 2002	JAN 2002
➢ Project Initiation ➢ Overview ➢ Objectives ➢ Scope	▨	▨	▨					
➢ Literature Research ➢ Information collecting ➢ Analysis on existing system ➢ Research on system requirement ➢ Research database and platform	▨	▨	▨					
➢ System analysis ➢ Review of entire project requirements ➢ Learn MATLAB, Ms Access	▨	▨	▨					
➢ System design ➢ Review on user requirement ➢ Program design ➢ Interface design ➢ Database design ➢ Preparation for viva		▨	▨	▨	▨			
➢ Coding ➢ Prototype development ➢ Develop Genetic algorithm using MATLAB ➢ Develop database					▨	▨	▨	
➢ Testing ➢ Unit testing ➢ Integration testing ➢ Corrections and enhancement						▨	▨	▨
➢ Final documentation ➢ Finalizing of project documentation ➢ Submission of documentation ➢ Preparation for presentation							▨	▨

Figure 1.1: Project Schedule

1.11 Expected Outcome

This system is expected to predict the survival of the patient with the accuracy within a range of 30 days or a month.

CHAPTER II

LITERATURE REVIEW

Chapter 2 : Literature Review

The most important thing in developing a system is how far we understand the system, how the system will work and how the system will be developed to achieve the goal. This chapter is dedicated to help us understand how the data and information that we get can help us in developing the system.

2.1 Data Mining and Facts Finding

Data mining, also known as knowledge discovery, is the process of finding patterns, trends, and regularities by sifting through large amounts of data [vi]. Data mining involves the analysis of data stored in databases to discover associations or patterns, to segment (or cluster) records based on similarity of attributes, and to create predictive (or classification) models.

There are two major approaches to data mining: supervised and unsupervised. In the supervised approach, specific examples of a target concept are given, and the goal is to learn how to recognize members of the class using the description attributes. In the unsupervised approach, a set of examples is provided without any prior classification, and the goal is to discover underlying regularities and patterns, most often by identifying clusters or subsets of similar examples

Clinical databases have accumulated large amounts of data on patients and their medical conditions. The clinical history of a patient generates data that goes beyond the disease being treated. The information, stored along with that of other patients, constitutes a good place to look for new relationships and patterns, and to validate proposed hypotheses.

The range of applications of data mining in medicine is very wide, with the two most popular applications being diagnosis and prognosis. Diagnosis is the process of selectivity gathering information concerning a patient, and interpreting it according to previous knowledge, as evidence for or against the presence or absence of disorders. In a prognostic process, a patient's information is also gathered and interpreted, but the objective is to predict the future development of the patient's condition. Due to the predictive nature of this process, prognostic systems are frequently used as tools to plan medical treatments [vii].

The role played by data mining in the context of diagnosis and prognosis is the discovery of the knowledge necessary to interpret the gathered information. In some cases this knowledge is expressed as probabilistic relationships between clinical features and the proposed diagnosis or prognosis.

Evolutionary computation is usually applied in medical data mining as a parameter finder. Evolutionary techniques search for the parameter values of the knowledge representation set up by the designer so that the mined data are optimally interpreted. For example, evolutionary algorithms can search for the weight of a neural network, the membership function values of a fuzzy system, or the coefficient of a linear regressor.

2.2 Researched on Existing System

After some researches been carried out, there are some system that might be useful have been found.

2.2.1 Early Detection of Ovarian Cancer with Genetic Algorithms and Support Vector Machines

A new classification model for cancer detection from genomic and proteomic data was developed, extending Petrocoin's [1] work. A Genetic Algorithm (GA) selects features from a large featureset available from low-molecular-weight serum protein profiling. The GA wraps around a Support Vector Machine (SVM) that builds the classification model.

Importance of early detection. Current positive prediction rates (number of true positives over all positives detected by a method) for early-stage ovarian cancer using the Cancer Antigen 125 (CA125) biomarker in addition to ultrasound is around 20%. Although easier to detect at later stages (80% for stages II, III and IV), treatment results are better when detected early.

Mass spectroscopy data. New data from SELDI-TOF (surface-enhanced laser desorption and ionization time-of-flight) low-molecular-weight serum protein profiling offers a new potential for the early detection of ovarian cancer. Data collected from 100 ovarian cancer patients and 100 healthy cases using Cyphergen's ProteinChip was downloaded from the Clinical Proteomics Program Databank. Mass scpectroscopy data from each profile consists of 15,153 intensity level readings for a range of mass/energy values (Figure 2.1).

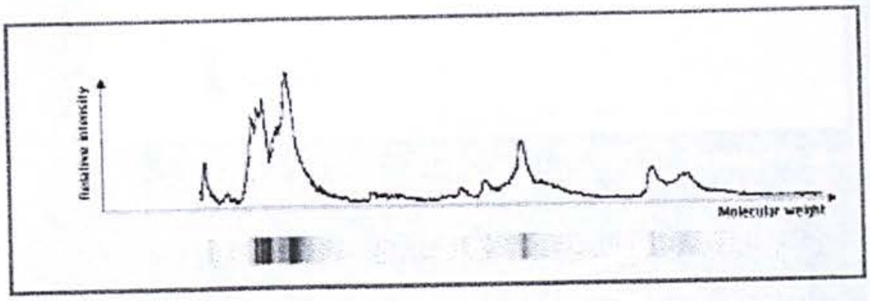


Figure 2.1: Mass Spectroscopy Data

Learning the Classification Model. The Genetic Algorithm selects a number of features from the mass spectroscopy reading (Figures 2.2 and 2.3), and passes the feature set to a Support Vector Machine.

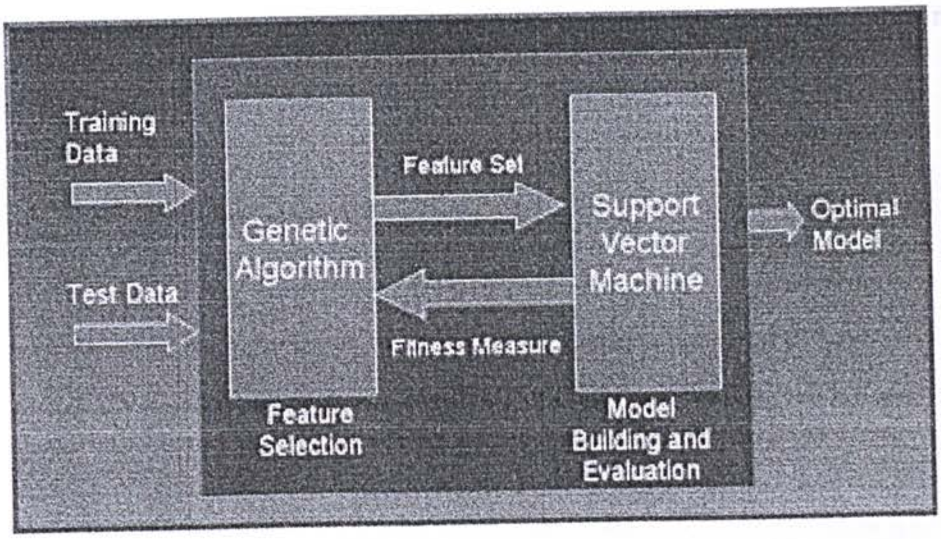


Figure 2.2: Learning the classification model

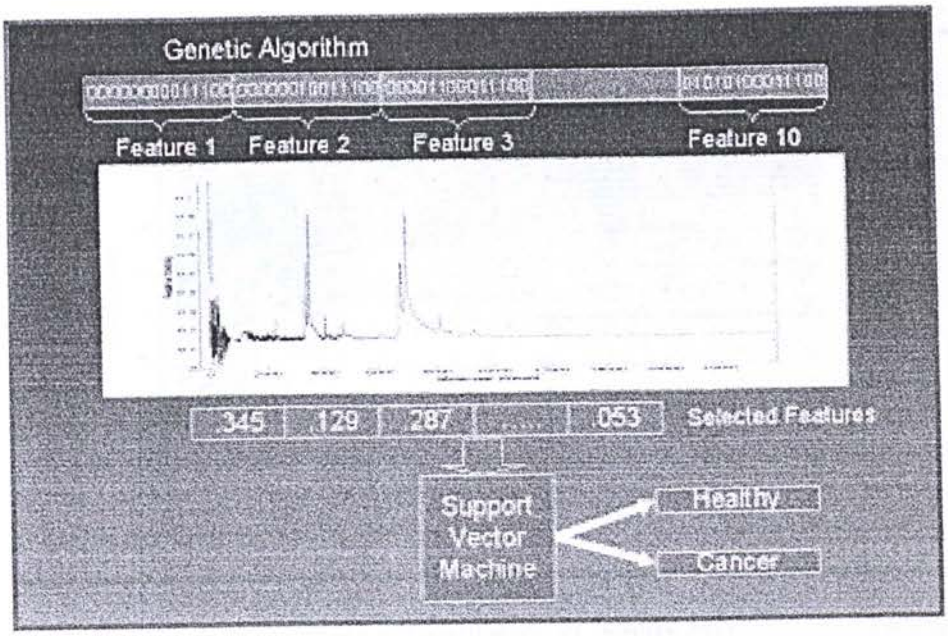


Figure 2.3: Encoding a Features Using a Genetic Algorithm

After the best model is generated with the input data, the accuracy of the model is tested, and the value returned to the Genetic Algorithm as a fitness measure of the feature set. With new generations of the Genetic Algorithm, better featuresets are evolved (Figure 2.4). Some overfitting results from the extreme high dimensional nature of the profiles [2].

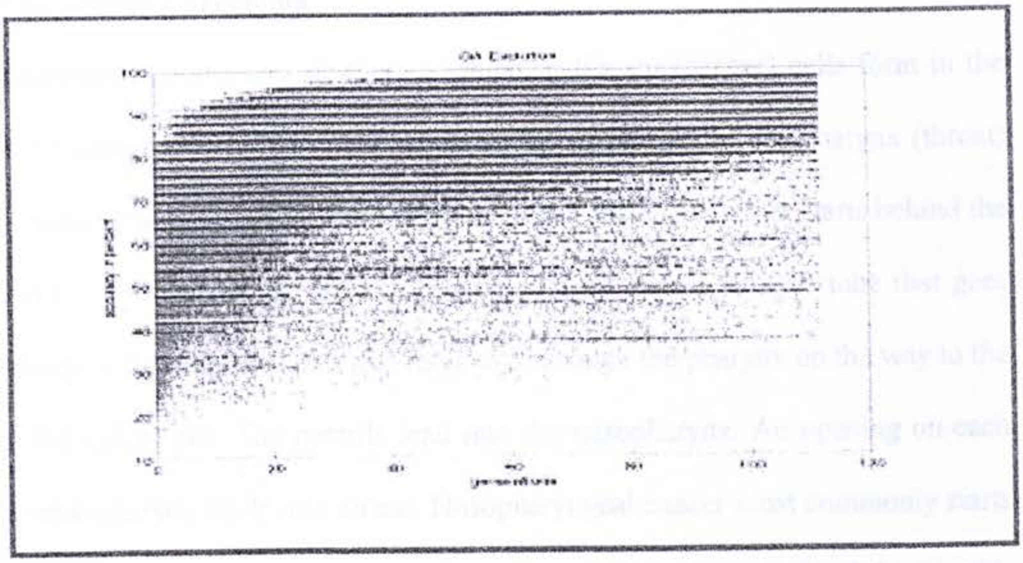
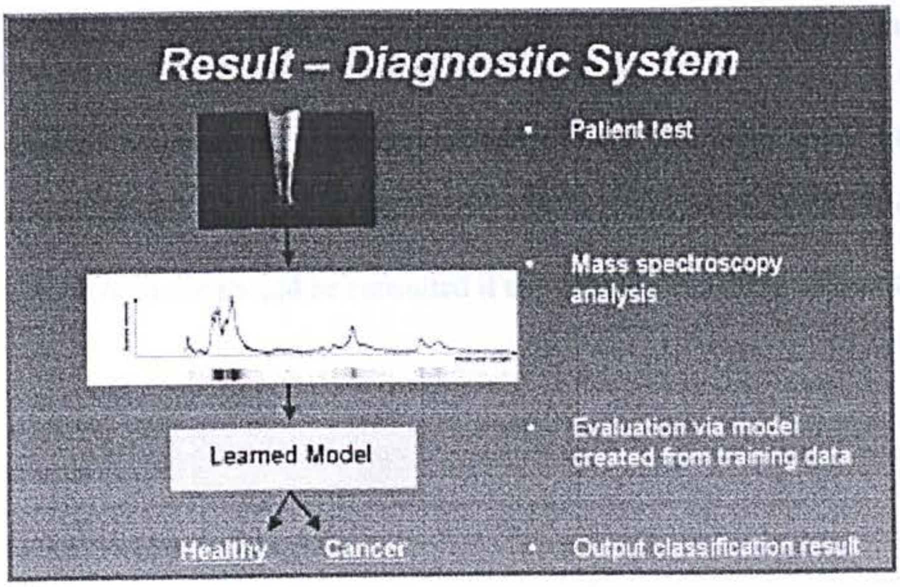


Figure 2.4:Evolution of feature sets

Final Classifier. Mass spectroscopy analysis is carried on a blood sample from a new patient. The learned model (Figure 2.5) uses as input only the subset of features previously identified by the Genetic Algorithm as valuable for the process of discriminating between healthy cases and cancer patients.


Result – Diagnostic System
Patient test
Mass spectroscopy analysis
Learned Model
Evaluation via model created from training data
Healthy
Cancer
Output classification result
Figure 2.5:Final Classifier.

2.3 Nasopharyngeal Carcinoma

Nasopharyngeal cancer is a disease in which malignant (cancer) cells form in the tissues of the nasopharynx. The nasopharynx is the upper part of the pharynx (throat) behind the nose. The pharynx is a hollow tube about 5 inches long that starts behind the nose and ends at the top of the trachea (windpipe) and esophagus (the tube that goes from the throat to the stomach). Air and food pass through the pharynx on the way to the trachea or the esophagus. The nostrils lead into the nasopharynx. An opening on each side of the nasopharynx leads into an ear. Nasopharyngeal cancer most commonly starts in the squamous cells that line the oropharynx (the part of the throat behind the mouth). Ethnic background and exposure to the Epstein-Barr virus can affect the risk of developing nasopharyngeal cancer. Risk factors may include the following:

- Chinese or Asian ancestry.
- Exposure to the Epstein-Barr virus: The Epstein-Barr virus has been associated with certain cancers, including nasopharyngeal cancer and some lymphomas.

Possible signs of nasopharyngeal cancer include trouble breathing, speaking, or hearing. These and other symptoms may be caused by nasopharyngeal cancer or by other conditions. A doctor should be consulted if any of the following problems occur:

- A lump in the nose or neck.
- A sore throat.
- Trouble breathing or speaking.
- Nosebleeds.
- Trouble hearing.

- Pain or ringing in the ear.
- Headaches.

Tests that examine the nose and throat are used to help detect (find) and diagnose nasopharyngeal cancer. The following tests and procedures can help detect nasopharyngeal cancer:

- Physical examination of the throat: An examination in which the doctor feels for swollen lymph nodes in the neck and looks down the throat with a small, long-handled mirror to check for abnormal areas.
- Nasoscopy: A procedure in which a doctor inserts a nasoscope (a thin, lighted tube) into the patient's nose to look for abnormal areas.
- Neurological tests: An examination in which the doctor tests hearing and nerve function.
- Head and chest x-rays: Brief exposure of the skull and chest to radiation to produce images of internal organs and structures.
- Magnetic resonance imaging: A procedure in which a magnet linked to a computer is used to create detailed pictures of areas inside the body. This test is also called MRI or nuclear magnetic resonance imaging (NMRI).
- Computed tomography: A series of detailed pictures of areas inside the body, taken from different angles; the pictures are created by a computer linked to an x-ray machine. This test is also called computerized tomography and computerized axial tomography (CAT) or CT scan.

- Laboratory tests: Medical procedures that involve testing samples of blood, urine, or other substances or tissues in the body to help determine the diagnosis, plan and check treatment, or monitor the course of disease over time.
- Biopsy: The removal of cells, tissues, or fluid to view under a microscope and check for signs of disease.

Certain factors affect prognosis (chance of recovery) and choice of treatment. The prognosis (chance of recovery) and choice of treatment depend on the stage of the cancer (whether it affects part of the nasopharynx, involves the whole nasopharynx, or has spread to other places in the body), the type of nasopharyngeal cancer, the size of the tumor, and the patient's age and general health.

2.3.1 STAGE OF NASOPHARYNGEAL CANCER

After nasopharyngeal cancer has been diagnosed (found), tests are done to find out if cancer cells have spread within the nasopharynx or to other parts of the body. The process used to find out whether cancer has spread within the nasopharynx or to other parts of the body is called staging. It is important to know the stage of the disease in order to plan the best treatment. The results of the tests used to diagnose nasopharyngeal cancer are often also used to stage the disease.

The following stages are used for nasopharyngeal cancer:

STAGE	FOUND	TREATMENT
Stage 0 (carcinoma in Situ)	Lining of the nasopharynx	
Stage I	Nasopharynx only	Radiation therapy
Stage IIA	Spread from nasopharynx to the oropharynx and/or nasal cavity	Chemotherapy combined with radiation therapy.
Stage IIB	Nashopharynx and has spread to lymph nodes on or surrounding the nasopharynx and to lymph nodes on one side of the neck	Chemotherapy combined with radiation therapy. Radiation therapy to the tumor and lymph nodes in the neck

Table 2.1: Stage of Nasopharyngeal Carcinoma

Stage III	Spread to lymph nodes on both sides of the neck. Spread to oropharynx and or the nasal cavity and to lymph nodes on both sides of the neck. Spread to nearby bones or sinuses with/without spreading to lymph nodes on one or both side of the neck.	Chemotherapy combined with radiation therapy. Radiation therapy to the tumor and lymph nodes in the neck. Radiation therapy followed by surgery to remove cancer-containing lymph nodes in the neck that persist or come back after radiation therapy. A clinical trial of chemotherapy before, combined with, or after radiation therapy.

Table 2.1: Stage of Nasopharyngeal Carcinoma(cont..)

Stage IVA	Spread to other areas in the head and may have spread to lymph nodes on one or both sides of the neck, and the involved lymph nodes are smaller than 6 centimeters.	Chemotherapy combined with radiation therapy. Radiation therapy to the tumor and lymph nodes in the neck. Radiation therapy followed by surgery to remove cancer-containing lymph nodes in the neck that persist or come back after radiation therapy. Chemotherapy for cancer that has metastasized (spread) to other parts of the body. A clinical trial of chemotherapy before, combined with, or after radiation therapy
Stage IVB	Spread to lymph nodes above the collarbone and/or the involved lymph nodes are larger than 6 centimeters.	
Stage IVC	Spread beyond nearby lymph nodes to other parts of the body.	

Table 2.1: Stage of Nasopharyngeal Carcinoma(cont..)

2.3.2 TREATMENT OPTION OVERVIEW

There are treatments for all patients with nasopharyngeal cancer. Some treatments are standard, and some are being tested in clinical trials. Before starting treatment, patients may want to think about taking part in a clinical trial. A treatment clinical trial is a research study meant to help improve current treatments or obtain information on new treatments for patients with cancer. When clinical trials show that a new treatment is better than the treatment currently used as "standard" treatment, the new treatment may become the standard treatment.

Clinical trials are taking place in many parts of the country. Information about ongoing clinical trials is available from the NCI cancer.gov Web site. Choosing the most appropriate cancer treatment is a decision that ideally involves the patient, family, and health care team.

Three types of standard treatment are used:

Radiation therapy

Radiation therapy is the use of x-rays or other types of radiation to kill cancer cells and shrink tumors. Radiation therapy may use external radiation (using a machine outside the body) or internal radiation. Internal radiation involves putting radioisotopes (materials that produce radiation) through thin plastic tubes into the area where cancer cells are found. Nasopharyngeal cancer is treated with external and internal radiation. Radiation may be used alone or in addition to chemotherapy or surgery.

External radiation therapy to the thyroid or the pituitary gland may change the way the thyroid gland works. The doctor may wish to test the thyroid gland before and after therapy to make sure it is working properly. Having a dentist evaluate dental health and correct any existing problems is particularly important prior to beginning radiation therapy.

Chemotherapy

Chemotherapy is the use of drugs to kill cancer cells. Chemotherapy may be taken by mouth, or it may be put into the body by inserting a needle into a vein or muscle. Either type of chemotherapy is called systemic treatment because the drugs enter the bloodstream, travel through the body, and can kill cancer cells throughout the body.

Surgery

Surgery is removing the cancer in an operation. Surgery is sometimes used for nasopharyngeal cancer that does not respond to radiation therapy. If cancer has spread to the lymph nodes, the doctor may remove lymph nodes and other tissues in the neck.

Other types of treatment are being tested in clinical trials:

Biological therapy

Biological therapy is treatment to stimulate the ability of the immune system to fight cancer. Substances made by the body or made in a laboratory are used to boost, direct, or restore the body's natural defenses against disease. Biological therapy is sometimes called biological response modifier (BRM) therapy or immunotherapy.

2.4 Evolutionary computing

Over many years, biologists have identified many principles, which govern the *evolution* of living things, at several level of detail. At the highest level, the theory of *natural selection* or *survival of the fittest* govern the evolutionary adaptation of the biological world from the smallest virus to the most complicated mammal. Natural selection operates on the organism through it performance on a specific task – that of producing offspring. The more viable offspring produced by organism, the more successful the organism.

There are several main styles in the evolutionary computation field, distinguished first by the type of structure, which comprise the individual in the population. These differences determine the factors by which one individual may differ from another, and thus the allowable genetic variation. Equally important differences also exist in the genetic operators used to create offspring, as well as many selection procedures based on fitness and a host of other less significant things.

Evolution strategy defines all style of evolutionary computation frequently associated with engineering optimization problems. The structures, which undergo adaptation, are typically sets of real-valued objective variable, which are associated with real-valued strategy variables in an individual.

2.4.1 Genetic Algorithm

History

The idea of evolutionary computing was introduced in the 1960s by I. **Rechenberg** in his work "***Evolution strategies***" (*Evolutionsstrategie* in original). His idea was then developed by other researchers. **Genetic Algorithms** (GAs) were invented by John **Holland** and developed by him and his students and colleagues. This lead to Holland's book "*Adaption in Natural and Artificial Systems*" published in 1975.

In 1992 John **Koza** has used Genetic Algorithm to evolve programs to perform certain tasks. He called his method "**genetic programming**" (GP). LISP programs were used, because programs in this language can expressed in the form of a "parse tree", which is the object the GA works on.

2.4.1.1Biological Background

Chromosome

All living organisms consist of cells. In each cell there is the same set of **chromosomes**. Chromosomes are strings of **DNA** and serves as a model for the whole organism. A chromosome consists of **genes**, blocks of DNA. Each gene encodes a particular protein. Basically can be said, that each gene encodes a **trait**, for example color of eyes. Possible settings for a trait (e.g. blue, brown) are called **alleles**. Each gene has its own position in the chromosome. This position is called **locus**. Complete set of genetic material (all chromosomes) is called **genome**. Particular set of genes in genome is called **genotype**. The genotype is with later development after birth base for the organism's **phenotype**, its physical and mental characteristics, such as eye color, intelligence etc.

Reproduction

During reproduction, first occurs **recombination** (or **crossover**). Genes from parents form in some way the whole new chromosome. The new created offspring can then be mutated. **Mutation** means, that the elements of DNA are a bit changed. This changes are mainly caused by errors in copying genes from parents. The **fitness** of an organism is measured by success of the organism in its life.

2.4.1.2 Search Space

If we are solving some problem, we are usually looking for some solution, which will be the best among others. The space of all feasible solutions (it means objects among those the desired solution is) is called **search space** (also state space). Each point in the search space represent one feasible solution. Each feasible solution can be "marked" by its value or fitness for the problem. We are looking for our solution, which is one point (or more) among feasible solutions - that is one point in the search space. The looking for a solution is then equal to a looking for some extreme (minimum or maximum) in the search space. The search space can be whole known by the time of solving a problem, but usually we know only a few points from it and we are generating other points as the process of finding solution continues.

Algorithm is started with a **set of solutions** (represented by **chromosomes**) called **population**. Solutions from one population are taken and used to form a new population. This is motivated by a hope, that the new population will be better than the old one. Solutions which are selected to form new solutions (**offspring**) are selected according to their fitness - the more suitable they are the more chances they have to reproduce. This is repeated until some condition (for example number of populations or improvement of the best solution) is satisfied.

2.4.1.3 Outline of the Basic Genetic Algorithm

1. **[Start]** Generate random population of *n* chromosomes (suitable solutions for the problem)
2. **[Fitness]** Evaluate the fitness *f(x)* of each chromosome *x* in the population
3. **[New population]** Create a new population by repeating following steps until the new population is complete
    1. **[Selection]** Select two parent chromosomes from a population according to their fitness (the better fitness, the bigger chance to be selected)
    2. **[Crossover]** With a crossover probability cross over the parents to form a new offspring (children). If no crossover was performed, offspring is an exact copy of parents.
    3. **[Mutation]** With a mutation probability mutate new offspring at each locus (position in chromosome).
    4. **[Accepting]** Place new offspring in a new population
4. **[Replace]** Use new generated population for a further run of algorithm

5. **[Test]** If the end condition is satisfied, **stop,** and return the best solution in current population
6. **[Loop]** Go to step **2**

2.4.1.4 Operators of GA

Encoding of a Chromosome

The chromosome should in some way contain information about solution which it represents. The most used way of encoding is a binary string. The chromosome then could look like this:

Chromosome 1	1101100100110110
Chromosome 2	1101111000011110

Table 2.2: Binary coding

Each chromosome has one binary string. Each bit in this string can represent some characteristic of the solution. Or the whole string can represent a number.

There are many other ways of encoding. This depends mainly on the solved problem. For example, one can encode directly integer or real numbers, sometimes it is useful to encode some permutations and so on.

Crossover

After we have decided what encoding we will use, we can make a step to crossover. Crossover selects genes from parent chromosomes and creates a new offspring. The simplest way how to do this is to choose randomly some crossover point and everything before this point copy from a first parent and then everything after a crossover point copy from the second parent.

Crossover can then look like this (| is the crossover point):

Chromosome 1	11011 | 00100110110
Chromosome 2	11011 | 11000011110
Offspring 1	11011 | 11000011110
Offspring 2	11011 | 00100110110

Table 2.3: Binary Crossover

There are other ways on how to make crossover, for example we can choose more crossover points. Crossover can be rather complicated and very depends on encoding of the encoding of chromosome. Specific crossover made for a specific problem can improve performance of the Genetic Algorithm.

made from parts of parents' chromosome. If crossover probability is **100%**, then all offspring is made by crossover. If it is **0%**, whole new generation is made from exact copies of chromosomes from old population (but this does not mean that the new generation is the same!).

Crossover is made in hope that new chromosomes will have good parts of old chromosomes and maybe the new chromosomes will be better. However it is good to leave some part of population survive to next generation.

- **Mutation probability** says how often will be parts of chromosome mutated. If there is no mutation, offspring is taken after crossover (or copy) without any change. If mutation is performed, part of chromosome is changed. If mutation probability is **100%**, whole chromosome is changed, if it is **0%**, nothing is changed.

Mutation is made to prevent falling GA into local extreme, but it should not occur very often, because then GA will in fact change to **random search**.

2.4.1.6 Other Parameters

There are also some other parameters of GA. One also important parameter is population size.

Population size says how many chromosomes are in population (in one generation). If there are too few chromosomes, GA have a few possibilities to perform crossover and only a small part of search space is explored. On the other hand, if there are too many chromosomes, GA slows down. Research shows that after some limit (which depends mainly on encoding and the problem) it is not useful to increase population size, because it does not make solving the problem faster.

2.4.1.7 Selection

Introduction

Chromosomes are selected from the population to be parents to crossover. The problem is how to select these chromosomes. According to Darwin's evolution theory the best ones should survive and create new offspring. There are many methods how to select the best chromosomes, for example roulette wheel selection, Boltzman selection, tournament selection, rank selection, steady state selection and some others.

Roulette Wheel Selection

Parents are selected according to their fitness. The better the chromosomes are, the more chances to be selected they have. Imagine a **roulette wheel** where are placed all chromosomes in the population, every has its place big accordingly to its fitness function, like on the following picture.

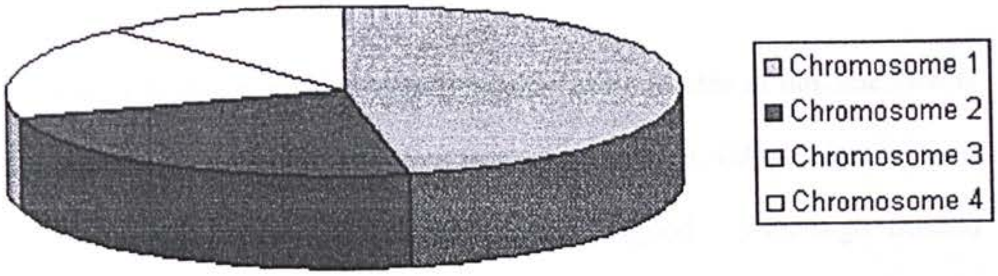


Figure 2.6: Roulette Wheel Selection

Then a marble is thrown there and selects the chromosome. Chromosome with bigger fitness will be selected more times.

This can be simulated by following algorithm.

1. **[Sum]** Calculate sum of all chromosome fitnesses in population - sum ***S***.
2. **[Select]** Generate random number from interval ***(0,S)*** **- *r***.
3. **[Loop]** Go through the population and sum fitnesses from ***0*** - sum ***s***. When the sum ***s*** is greater then ***r***, stop and return the chromosome where you are.

Of course, step **1** is performed only once for each population.

Rank Selection

The previous selection will have problems when the fitnesses differs very much. For example, if the best chromosome fitness is 90% of all the roulette wheel then the other chromosomes will have very few chances to be selected. Rank selection first ranks the population and then every chromosome receives fitness from this ranking. The worst

Steady-State Selection

This is not particular method of selecting parents. Main idea of this selection is that big part of chromosomes should survive to next generation. GA then works in a following way. In every generation is selected a few (good - with high fitness) chromosomes for creating a new offspring. Then some (bad - with low fitness) chromosomes are removed and the new offspring is placed in their place. The rest of population survives to new generation.

Elitism

Idea of elitism has been already introduced. When creating new population by crossover and mutation, we have a big chance, that we will loose the best chromosome. Elitism is name of method, which first copies the best chromosome (or a few best chromosomes) to new population. The rest is done in classical way. Elitism can very rapidly increase performance of GA, because it prevents losing the best-found solution.

2.4.1.8 Encoding

Introduction

Encoding of chromosomes is one of the problems, when you are starting to solve problem with GA. Encoding very depends on the problem.

Binary Encoding

Binary encoding is the most common, mainly because first works about GA used this type of encoding.

In **binary encoding** every chromosome is a string of **bits, 0** or **1**.

Chromosome A	101100101100101011100101
Chromosome B	111111100000110000011111

Table 2.5: Example of chromosomes with binary encoding

Binary encoding gives many possible chromosomes even with a small number of alleles. On the other hand, this encoding is often not natural for many problems and sometimes corrections must be made after crossover and/or mutation.

Permutation Encoding

In **permutation encoding**, every chromosome is a string of numbers, which represents number in a **sequence**.

Chromosome A	1 5 3 2 6 4 7 9 8
Chromosome B	8 5 6 7 2 3 1 4 9

Table 2.6: Example of chromosomes with permutation encoding

Permutation encoding is only useful for ordering problems. Even for this problems for some types of crossover and mutation corrections must be made to leave the chromosome consistent (i.e. have real sequence in it).

Value Encoding

Direct value encoding can be used in problems, where some complicated values, such as real numbers, are used. Use of binary encoding for this type of problems would be very difficult. In **value encoding**, every chromosome is a string of some values. Values can be anything connected to problem, form numbers, real numbers or chars to some complicated objects.

Chromosome A	1.2324 5.3243 0.4556 2.3293 2.4545
Chromosome B	ABDJEIFJDHDIERJFDLDFLFEGT
Chromosome C	(back), (back), (right), (forward), (left)

Table 2.7: Example of chromosomes with value encoding

Value encoding is very good for some special problems. On the other hand, for this encoding is often necessary to develop some new crossover and mutation specific for the problem.

Tree Encoding

Tree encoding is used mainly for evolving programs or expressions, for **genetic programming**. In **tree encoding** every chromosome is a tree of some objects, such as functions or commands in programming language.

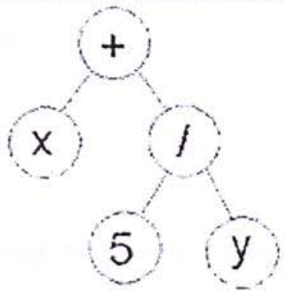
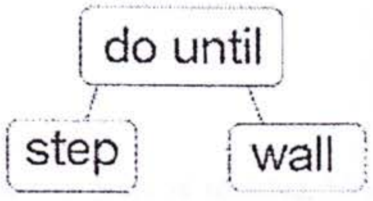
Chromosome A	Chromosome B
	
(+ x (/ 5 y))	(do_until step wall)

Table 2.8:Example of chromosomes with tree encoding

Tree encoding is good for evolving programs. Programming language LISP is often used to this, because programs in it are represented in this form and can be easily parsed as a tree, so the crossover and mutation can be done relatively easily.

2.4.1.9 Crossover and Mutation

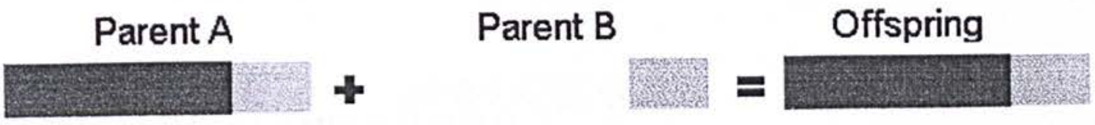
Introduction

Crossover and mutation are two basic operators of GA. Performance of GA very depends on them. Type and implementation of operators depends on encoding and also on a problem.

Binary Encoding

Crossover

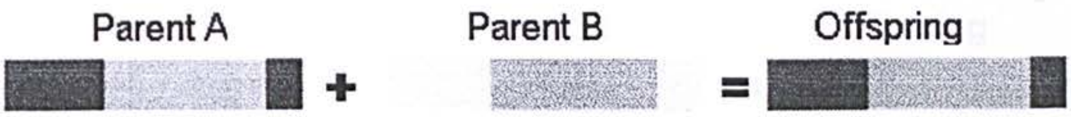
Single point crossover - one crossover point is selected, binary string from beginning of chromosome to the crossover point is copied from one parent, the rest is copied from the second parent



11001011+110**11111** = **11001111**

Figure 2.9: Single Point Crossover

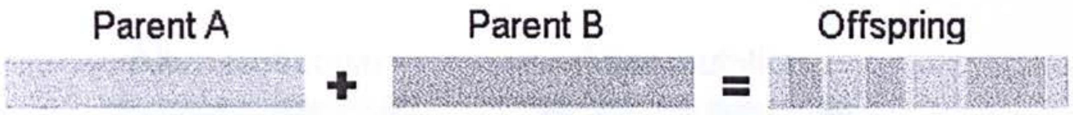
Two point crossover - two crossover point are selected, binary string from beginning of chromosome to the first crossover point is copied from one parent, the part from the first to the second crossover point is copied from the second parent and the rest is copied from the first parent



11001**011** + 11**0111**11 = **11011111**

Figure 2.10: Double Point Crossover

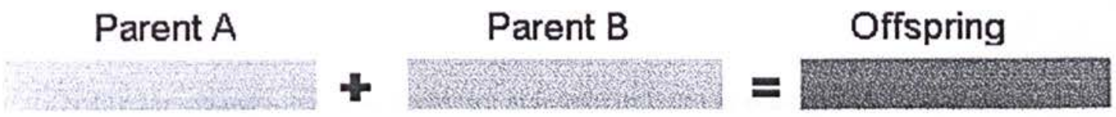
Uniform crossover - bits are randomly copied from the first or from the second parent



11001**011** + **11**0**111**01 = 11011111

Figure 2.11: Uniform Crossover

Arithmetic crossover - some arithmetic operation is performed to make a new offspring

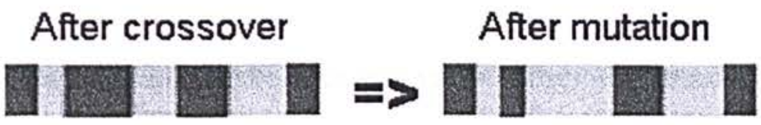


11001011 + 11011111 = 11001001 (AND)

Figure 2.12: Arithmetic Crossover

Mutation

Bit inversion - selected bits are inverted



11001001 => 10001001

Figure 2.13: Mutation

Permutation Encoding

Crossover

Single point crossover - one crossover point is selected, till this point the permutation is copied from the first parent, and then the second parent is scanned and if the number is not yet in the offspring it is added

(**1 2 3 4 5** 6 7 8 9) + (**4 5 3 6 8** 9 7 2 1) = (**1 2 3 4 5** 6 8 9 7)

Mutation

Order changing - two numbers are selected and exchanged

(1 **2** 3 4 5 6 **8** 9 7) => (1 **8** 3 4 5 6 **2** 9 7)

Value Encoding

Crossover

All crossovers from **binary encoding** can be used

Mutation

Adding a small number (for real value encoding) - to selected values is added (or subtracted) a small number

(1.29 5.68 **2.86** **4.11** 5.55) => (1.29 5.68 **2.73** **4.22** 5.55)

Tree Encoding

Crossover

Tree crossover - in both parent one crossover point is selected, parents are divided in that point and exchange part below crossover point to produce new offspring

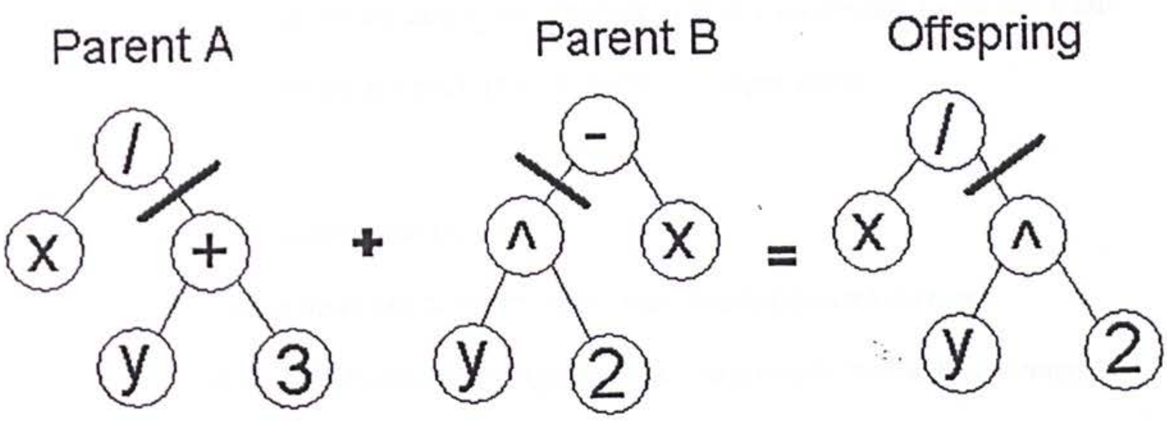


Figure 2.14: Tree Crossover

Mutation

Changing operator, number - selected nodes are changed

2.5 Neural Network

Artificial Neural Network

An *artificial neural network* is an information-processing system that has certain performance characteristics in common with biological neural networks. Artificial neural networks have been develop as generalizations of mathematical models of human cognition or neural biology, based on assumptions that:

I. Information processing occurs at many single elements called neurons.

II. Signals are passed between neurons over connection links.

III. Each connection link has an associated weight, which, in typical neural net, multiplies the signal transmitted.

IV. Each neuron applies an activation function (usually nonlinear) to its net input (sum of weighted input signals) to determine its output signal.

A neural network is characterized by

I. Its pattern of connections between the neurons (called its *architecture*).
II. Its method of determining the weights on the connections (called its *training*, or *learning, algorithm*).
III. Its *activation function.*

A neural net consists of a large number of simple processing elements called *neurons*, *units*, *cells*, or *nodes*. Each neurons is connected to other neurons by means of directed communications links, each with an associated weight. The weights represent information

being used by the net to solve a problem. Neural nets can be applied to a wide variety of problems, such as storing and recalling data or patterns, classifying patterns, performing general mappings from input patterns to output patterns, grouping similar patterns, or finding solutions to constrained optimization problems.

Each neuron has an internal state, called its *activations* or *activity level*, which is a function of the input it has received. Typically, a neuron sends its activation as a signal to several other neurons. A neuron can only send one signal at a time, although the signal is broadcast to several other neurons.

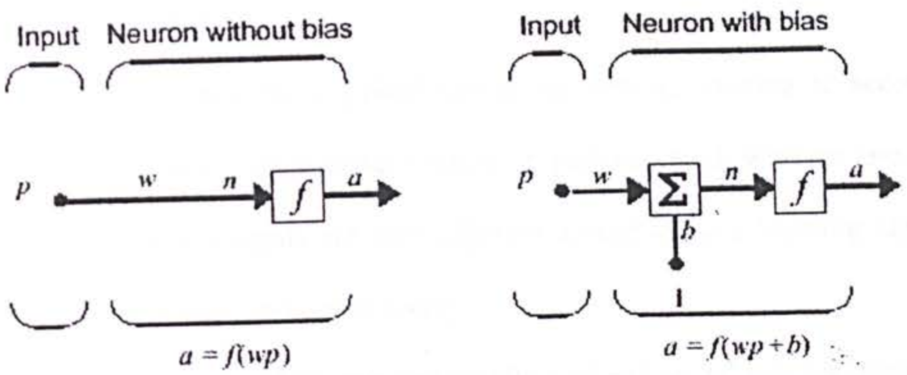


Figure 2.15: A simple (artificial) neurons.

2.5.1 Setting the weight

In addition to the architecture, the method of setting the values of the weights (training) is an important distinguishing characteristic of different neural nets. For convenience, we shall distinguished two type of training – supervised and unsupervised – for a neural network; in addition, there are nets whose weights are fixed without an iterative training process.

Many of the tasks that neural nets can be trained to perform fall into the areas of mapping, clustering, and constrained optimization. Pattern classification and pattern association may be considered special forms of the more general problem of mapping input vectors or patterns to the specified output vectors or patterns.

There is some ambiguity in the labeling of training methods as supervised or unsupervised, that lead to a useful third category, self-supervised training. However, in general, there is a useful correspondence between the type of training that is appropriate and the type problem that we wish to solve.

Supervised training

In perhaps the most typical neural net setting, training is accomplished by presenting a sequence of training vectors or patterns, each with an associated target output vector. The weights are then adjusted according to a learning algorithm. This process is known as supervised training.

Some of the simplest (and historically earliest) neural nets are design to perform pattern classification, i.e., to classify an input vector as either belonging to a given category. In this type of neural net, the output is a bivalent element, say, either 1 (if the input vector belongs to the category) or -1 (if it does not belong).

Pattern association is another special form of a mapping problem, one in which the desired output is not just a "yes" or "no", but rather a pattern. A neural net that is trained to associate a set of input vectors with a corresponding set of output vectors is called an *associative memory*. If the desired output vector is the same as the input vector, the net is an *autoassociative memory*; if the output target vector is different from the input vector, the net is a *heteroassociative memory*. After training, an associative memory can recall a stored pattern when it is given an input vector that is sufficiently similar to a vector it has learn .

Multiplayer neural nets can be trained to perform a nonlinear mapping from an *n*-dimensional space of input vectors (*n*-tuples) to an *m*-dimensional output space.

Unsupervised Learning

Self-organizing neural nets group similar input vectors together without the use of training data to specify what a typical member of each group looks like or to which group each vector belongs. A sequence of input vectors is provided, but no target vectors are specified. The net modifies the weight so that the most similar input vectors are assigned to the same output (or cluster) unit. The neural net will produce an exemplar (representative) vector for each cluster formed. Some of the example are self organizing nets and adaptive resonance theory.

Hybrid Models

Models that used both supervised and unsupervised neural nets concept or models where neural networks are combined with ruled-based, classic statistical, and other type of modeling approach.

Fixed Weight Nets

Type of neural nets that can solve constrained optimization problems. It may work well for problem s that can caused difficulty for traditional techniques, such as problem with conflicting constraints (i.e., not all constraints can be satisfied simultaneously). Often in such cases, a nearly optimal solution (which the net can find) is satisfactory. When these nets are designed, the weights are set to represent the constraints and the quantity to be maximized or minimized. The Boltzmann machine and the continuous Hopfield net used for constrained optimization problems.

2.5.2 Transfer Functions

Three of the most commonly used functions are shown below.

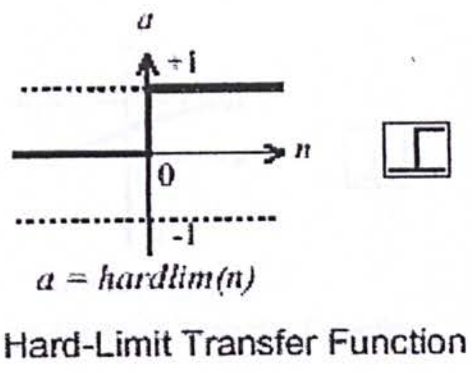


Figure 2.16: Hard Limit Transfer Function

The hard-limit transfer function shown above limits the output of the neuron to either 0, if the net input argument *n* is less than 0; or 1, if *n* is greater than or equal to 0.

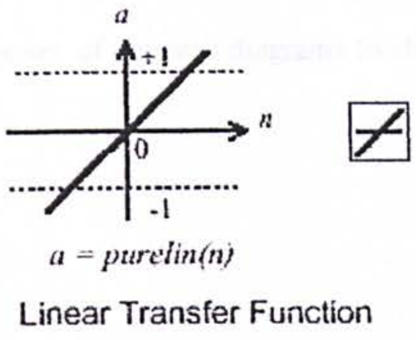


Figure 2.17: Linear Transfer Function

Neurons of this type are used as linear approximators.

The sigmoid transfer function shown below takes the input, which may have any value between plus and minus infinity, and squashes the output into the range 0 to 1.

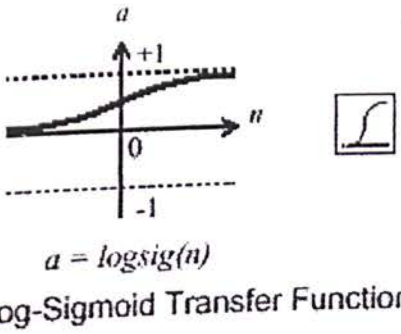


Figure 2.18: Log-Sigmoid Transfer Function

This transfer function is commonly used in backpropagation networks, in part because it is differentiable. The symbol in the square to the right of each transfer function graph shown above represents the associated transfer function. These icons will replace the general *f* in the boxes of network diagrams to show the particular transfer function being used.

CHAPTER III

METHODOLOGY/ SYSTEM ANALYSIS

Chapter 3: Methodology / System Analysis

A system development methodology is a very formal and precise system development process that defines a set of activities, methods, best practices, deliverables, and automated tools for system developers and project managers to use to develop and maintain most or all information system and software.

System analysis is a problem-solving technique that decomposes a system into its component pieces for the purpose of studying how well those components part work and interact to accomplish their purpose. Every system has a different development process. System modeling is important as it form a common understanding of resource and constraint, inconsistencies and omissions of a project.

3.1 Prototype Model

The Prototype model was developed on the assumption that it is often difficult to know all of requirements at the beginning of a project. Typically, users know how many of the objectives that they wish to address with the system, but they do not know all the nuances of the data, nor do they know the details of the system features and capabilities. The prototype model allows for these conditions, and offer a development approach that yield results without first requiring all information up-front.

For the propose system, the development stage is crucial because of the accuracy of the system that predict the survival. One thing come to mind, there is no such thing as half complete system, but it whether complete or not complete.The prototype model was selected because it allowed the system to be test repeatedly until all the needed functionalities were fulfill.

Prototyping is comprised of the following step:

- **Requirements Definition/Collection.** Similar to the Conceptualization phase of the *Waterfall Model*, but not as comprehensive. The information collected is usually limited to a subset of the complete system requirements.
- **Design.** Once the initial layer of requirement information is collected, or new information is gathered, it is rapidly integrated into a new or existing design so that it may folded into the prototype.
- **Prototype Creation/Modification.** The information of the design is rapidly rolled into a prototype. This may mean the creation/modification of paper information, new coding, or modifications to existing coding.

- **Assessment.** The prototype is presented to the customer for review. Comments and suggestions are collected from the customer.
- **Prototype Refinement.** Information collected from the customer is digested and the prototype is refined. The developer revised the prototype to make it more effective and efficient.
- **System Implementation.** In most cases, the system is rewritten once requirements are understood. Sometimes, the iterative process eventually produces a working system that can be cornerstone for the fully functional system.

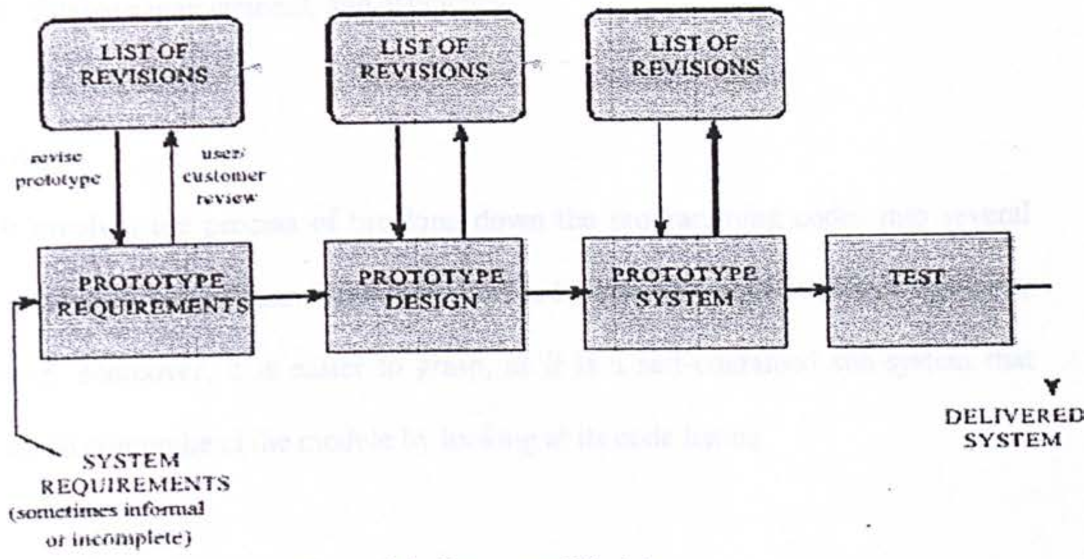


Figure 3.1: Prototype Model

The advantages of using Prototype model

- It improve understanding
- Allows all part of a system to be constructed quickly to understand or clarify issues

- Requirement or design require repeated investigation that ensure the common understanding both of what is needed and what is proposed
- Reduce risk and uncertainty in development

3.2 Requirement Analysis

3.2.1 Non-Functional Requirement

Non-functional requirement describes the restrictions and constraints on the system that limits our choices for constructing a solution to the problems. The non-functional requirement here are modularity, security, user-friendliness, reliability, and accuracy, database management, and efficiency.

Modularity

It involves the process of breaking down the programming codes into several logic and manageable portions. Each individual is functionally cohesive; thus it is easier to maintain. Moreover, it is easier to grasp, as it is a self-contained sub-system that anyone could comprehend the module by looking at its code listing.

Security

Security of the entire system is vital because it prevents loss of confidentiality or privacy, theft, and fraud. Thus, authentication towards entrants of the system should be strengthened. Database security is also another important element as to protect the database against intentional and unintentional threats.

User-friendliness

Graphic User Interface (GUI) is a must for today system. It will help in making the system look attractive thus easy-to-use. It will help in working environment by ease the user compare to command line.

Accuracy and Reliability

For the system that will work with prediction, accuracy is one of the most important things to look at. If the system fail to predict accurately, it is of course will never aid to human life. The system also can be rely on giving a appropriate result.

Database Management

Data integrity is one of the important aspects that should be focused when handling with data. Data integrity refers to validity and consistency of stored data. Besides that, data consistency should also be defined to reduce the risk of occurring data inconsistency.

Efficiency

A good response time towards a request and no delay will help the system in efficiency.

3.3 System Development Tool

After literature researches have been carried out in chapter 2, some system development tools have been selected based on their features and capabilities. These consist of 3 part; platform, database, and programming language.

3.3.1 Platform

Microsoft Windows 98 has been chosen. Although it is not a latest version of Microsoft operating system but still it has specifications that is useful in developing this system.

- *Multitasking*

  Users could run multiple applications simultaneously on the same system. Number of application supported would depend on memory of the system.
- *Plug and Play*

  It easy to install plug and play devices without performing complicated setup
- *User-friendly environment*
- *Easy installation*
- *Availability of technical support*
- *Wide range of use*

3.3.2 Microsoft Access

For the database design, after some research on database builder, Microsoft Access is the most suitable database design software because of properties it shows.

Most of all it runs on Windows 98 platform. Access is a software used for a Relational Database Management System (RBDMS) developed by Microsoft. It is usually used by individuals or a small group of users. Nevertheless, by using interface paradigms such as Remote Data Object (RDO) and Data Access Object (DAO), Access can be used as a database in a client-server architecture environment. Access also provides a complete development environment from the aspect of developing an entity relation table (a relationship). A simple and intuitive interface makes database development easier.

As such with other database software, Access also provides services in the form of SQL applications, form builders, security and much more. But the security component for Access is not very good and can be easily hacked. Because of this, it is advisable that Access be used only for personal or when security is not a priority.

3.3.3 Visual Basic

Visual Basic is a tool that allowed you to create software application for the Windows operating system. It creates Windows desktop applications, reusable software components for building other applications, and applications targeted for Internet and intranets.

Visual Basic incorporates a set of software technology called ActiveX. ActiveX technology allows the creation, integration, and reuse of software components called controls. ActiveX controls are reusable software components that can be integrated into a large number of different software products. More than 2,000 ActiveX controls currently are available. Visual Basic ActiveX technology also allows you to create ActiveX documents. ActiveX documents are applications that can be delivered dynamically over the Internet and intranets with browsers such as Internet Explorer or Netscape Navigator.

3.3.4 MATLAB

MATLAB is a high-performance language for technical computing. It integrates computation, visualization, and programming in an easy-to-use environment where problems and solutions are expressed in familiar mathematical notation. Typical uses include

•Math and computation

•Algorithm development

•Data acquisition

•Modeling, simulation, and prototyping

•Data analysis, exploration, and visualization

•Scientific and engineering graphics

•Application development, including graphical user interface building.

MATLAB is an interactive system whose basic data element is an array that does not require dimensioning. This allows you to solve many technical computing problems, especially those with matrix and vector formulations, in a fraction of the time it would take to write a program in a scalar non-interactive language such as C or Fortran.

The name **MATLAB** stands for *matrix laboratory*. **MATLAB** was originally written to provide easy access to matrix software developed by the **LINPACK** and **EISPACK** projects.

Neural Network Toolbox

One of the toolbox provided in the MATLAB is a neural network toolbox. All the needed build in functions was made to ease the use of neural network such as *learning, creating the net, algorithm for neural network* and many more.

3.4 Hardware Requirement

Hardware also plays an important role along with the system development software. If the hardware fails to communicate with the software, it will lead to system crash and fail to perform smoothly. There are two kinds of hardware requirement used, the development r and end-user computer.

Development computer

Component	**Descriptions**
Microprocessor	AMD Athlon 1.7GHZ
RAM	256 MB
Storage	10GB of hard disk
Input Devices	Mouse, CD-ROM, and keyboard
Output Devices	Monitor and printer

Table 3.1:Development Computer

End-user computer

Component	**Descriptions**
Microprocessor	Pentium II 166MHZ or above
RAM	32 MB
Storage	2G of hard disk
Input Devices	Mouse, CD-ROM, and keyboard
Output Devices	Monitor and printer

Table 3.2: End-user Computer

CHAPTER IV

SYSTEM DESIGN

Chapter 4: System Design

System design is done in order to determine the features of the system, components, and processes in a system and its appearance to the users. System requirements defined in the previous chapter are translated into a system specification to build a complete and executable system.

For the proposed system, it consists of three majors part; the interfaces, the database and the genetic algorithm combine with neural network as the data processing for the system. Genetic algorithm will act as a preprocessing to determine the input variables that can be use, while the neural network will act as a postprocessing to give the final prediction for the data.

The database will store all the patients' data that will be process later to predict the survival of the patients and the interfaces will help the user to read and to input the data.

4.1 System Structure

The system structure is the core of the system design. The system will be develop according to the system structure shown below.

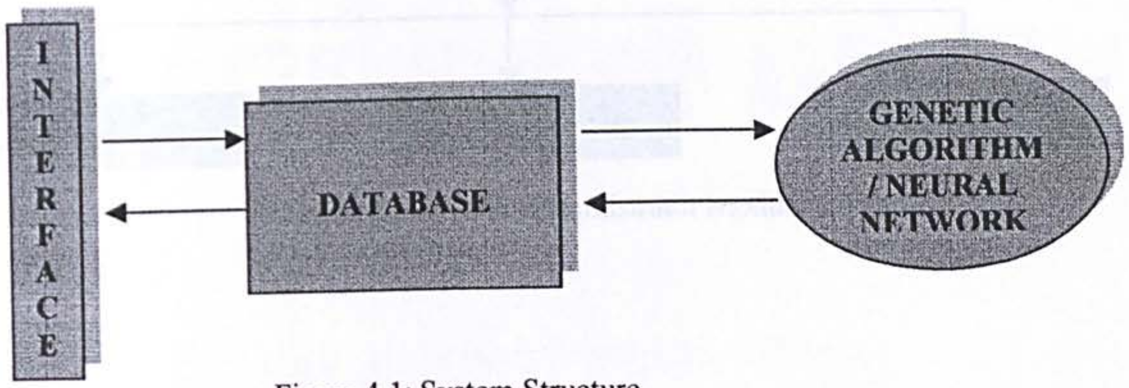


Figure 4.1: System Structure

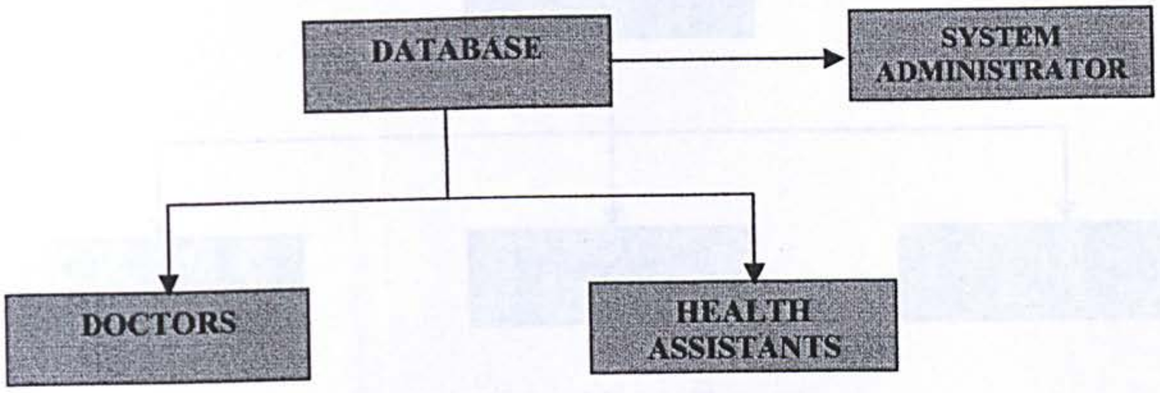


Figure 4.2: Database User Module Structure

Figure 4.2 shows that there is only three type of user that is allowed to access the database.

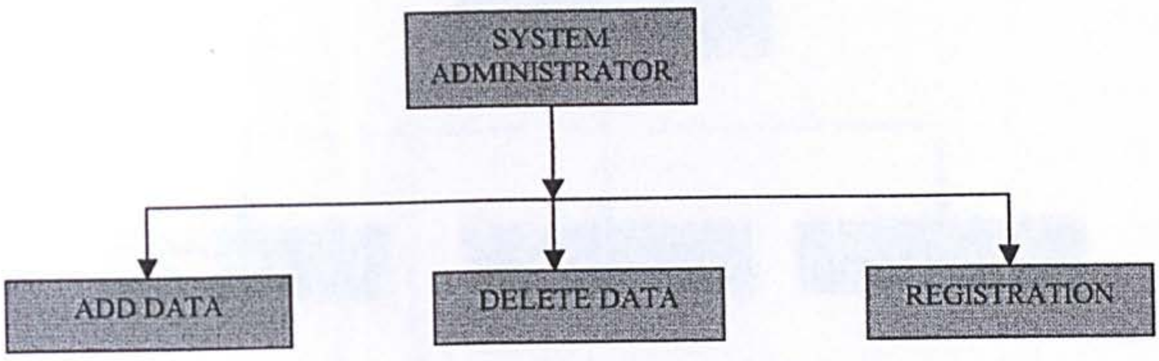


Figure 4.3: System Administrator Module Structure

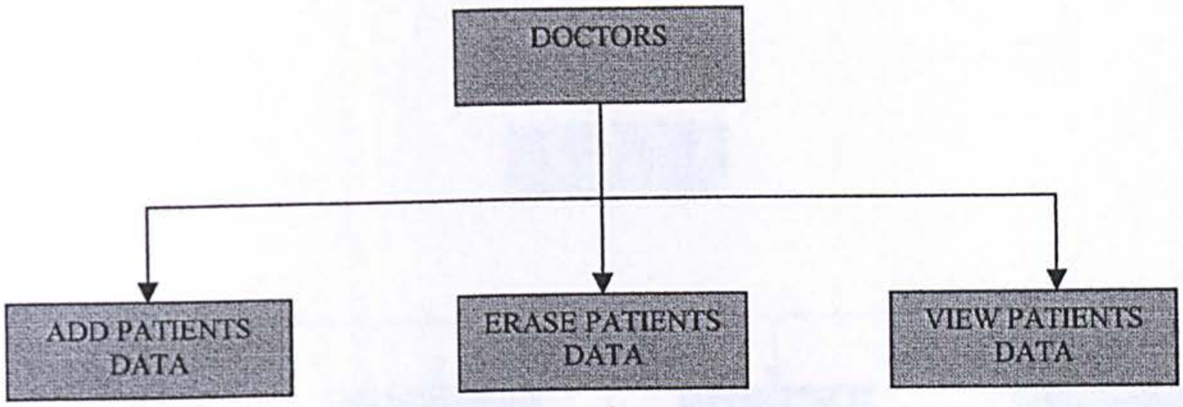


Figure 4.4: Doctors Module Structure

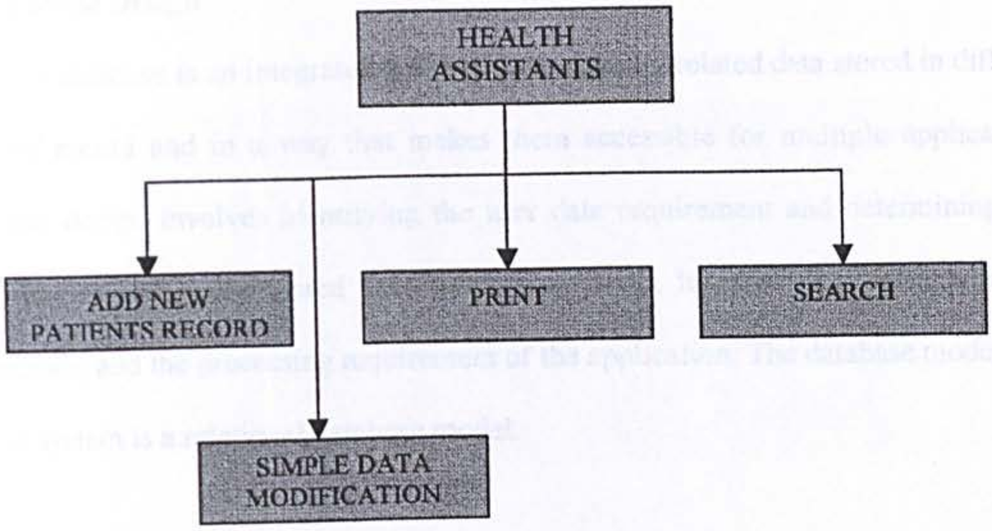


Figure 4.5: Health Assistants Module Structure

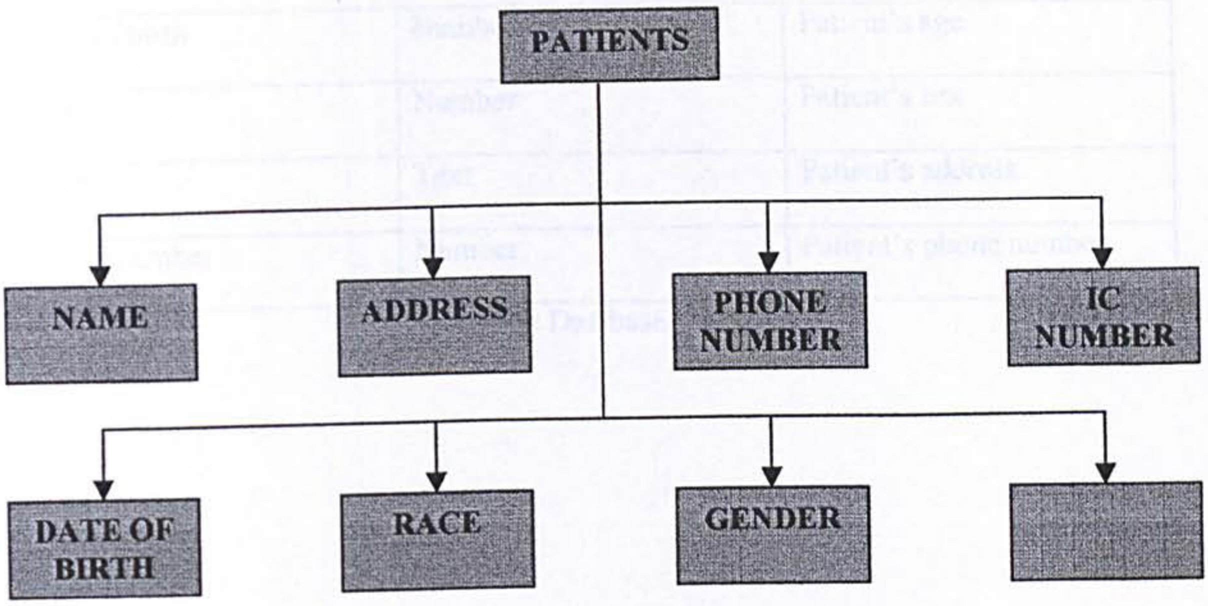


Figure 4.6: Type of Patients Data

4.3 Database Design

A database is an integrated collection of logically related data stored in different types of record and in a way that makes them accessible for multiple applications. Database design involves identifying the user data requirement and determining how these data should be structured for those requirements. It transforms the unstructured information and the processing requirement of the application. The database model used for this system is a relational database model.

Field Name	Data type	Description
Name	Text	Patient's name
IC. Number	Number	Patient's IC Number
Date of birth	Number	Patient's age
Sex	Number	Patient's sex
Address	Text	Patient's address
Phone number	Number	Patient's phone number

Table 4.1: Database Design

4.4 Interface Design

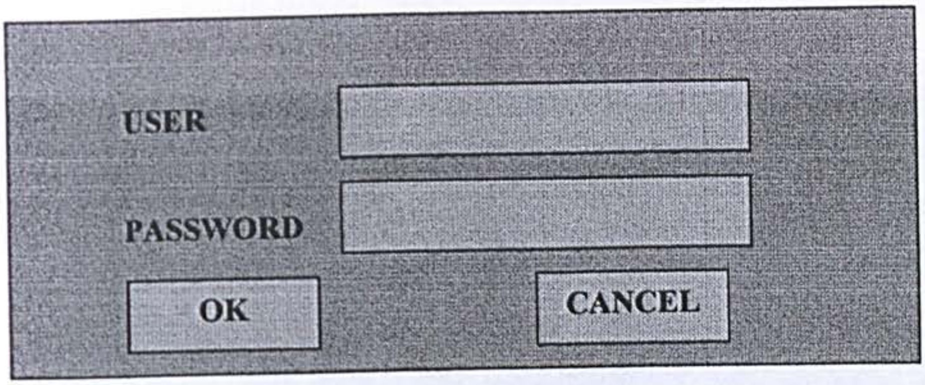


Figure 4.7: Interface of Login Page

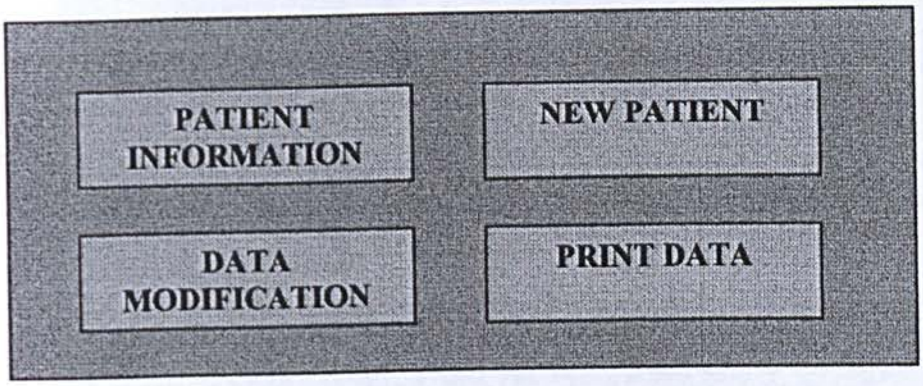


Figure 4.8: Interface of Main Page

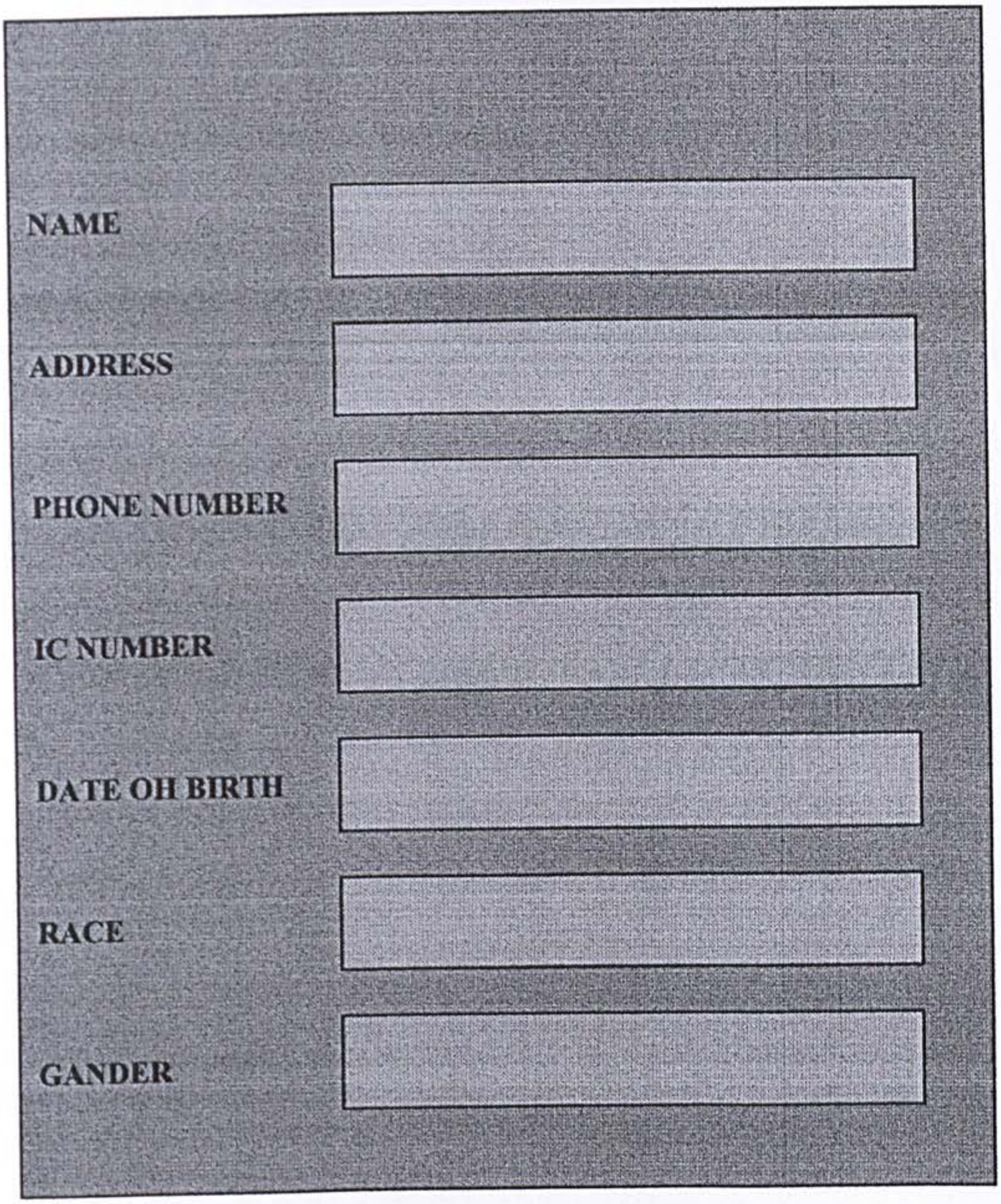


Figure 4.9: Interface of registration page

CHAPTER V

SYSTEM DEVELOPMENT & IMPLEMENTATION

Chapter 5: System Development and Implementation

5.1 System Implementation

System implementation is a process of creating a system by implementing the designed modules or functions described in the system requirement specification. It consists of 5 steps shown in the figure below.

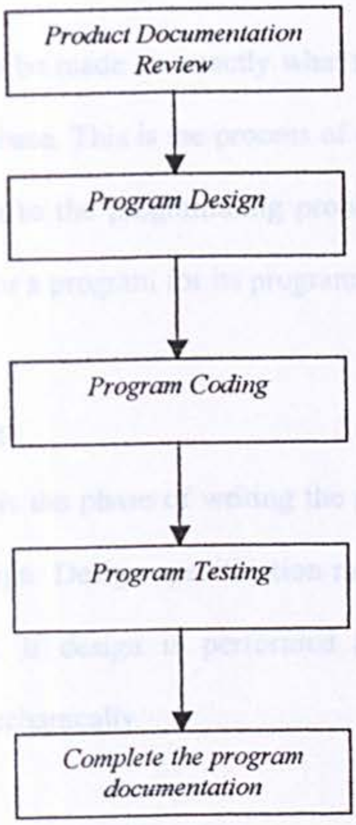

Product Documentation Review
Program Design
Program Coding
Program Testing
Complete the program documentation

Figure 5.1: 5 steps of System Implementation

i. **Product Documentation Review**

Product documentation review was been prepared during the previous phase. The documentation consists of data dictionary entries, process flow and source document. This documentation will give a better understanding of the work that needs to be done during the coding phase.

ii. Program Design

Decisions need to be made on exactly what the program can accomplish during program design phase. This is the process of what is must be done by developing a logical solution to the programming problem. A step-by-step solution is the logical solution for a program for its programming problem.

iii. Program Coding

Program coding is the phase of writing the program instruction that implement the program design. Design specification must be translated into the machine-readable format. If design is performed in details manner, coding can be accomplished mechanically.

iv. Program Testing

Program testing is the phase where the program is being tested. The testing is needed to ensure it functions correctly before the program processes actual data and produces information that user relying on.

v. Complete the program documentation

Completing the program is essential for the successful operation and maintenance of the information system. This documentation includes the system's user manual that may be needed by most of the customers as well as the system administrator.

5.1.1 Program Coding

During this phase, the programs are written and user interfaces are developed and the database is initialized with data. The components built during development are put into operational use. The system is built according to the original design that was done.

5.1.2 Coding Style

Coding style is an important attribute of source code and it determines the intelligibility of a program. An easy to read source code makes the system easier to maintain and enhance. The element of coding style includes internal (source code level) documentation, method for data declaration and approach to statement construction.

5.1.3 Documentation of Coding

Code documentation begin with the selection of identifier (variable and variable names), continues with the composition of connectivity and end with the organization of the program

a) Modularity

Before entering the coding phase, the project has been divided into several modules. The main purpose of modularity is to reduce the complexity of the system. In order to reduce complexity and facilitate changes that result in easier implementation by encouraging parallel development of different parts of a system.

b) Internal documentation

Internal comment provides a clear guide during the maintenance phase of the system. Comments provide the development with means of communicating with readers of the source code. Statement of purpose indicating the function of the module and a descriptive comment that is embedded within the body of the source code is needed to describe processing function.

c) Naming Convention

A good and meaningful naming technique for the variables, controls and modules provides easy identification for the programmer. The naming convention is created with coding consistency and standardization in mind.

d) Maintainability

Codes should be easily revised or corrected. To facilitate maintenance, code should be readable, modular and as general as possible.

e) Readability

Codes should be easy to understand. Adherence to coding conventions such as naming conventions and indentation contribute to program readability.

f) Robustness

The codes should be able to handle cases for user error by responding appropriately. It should be able to avoid any abrupt termination or system failure.

5.2 Development Environment

5.2.1 Application/Tools used in development

Following is the application or tools used throughout the development of the system:

a) Matlab version 6.1 with Neural Network Toolbox

b) Microsoft Access 2000

c) Microsoft Visual Basic 6.0

CHAPTER VI

SYSTEM TESTING

Chapter 6: System Testing

System testing is the process of testing process and approach used to test the system. It is a critical phase in which to make sure that the system fulfills the requirements. In this phase, a systematically test procedure is needed to make sure the system is tested thoroughly and completely. A few steps of test procedure have to go through to complete the system testing; there are unit testing, integration testing, sub-system testing, overall system testing and acceptance testing as shown in the figure below.

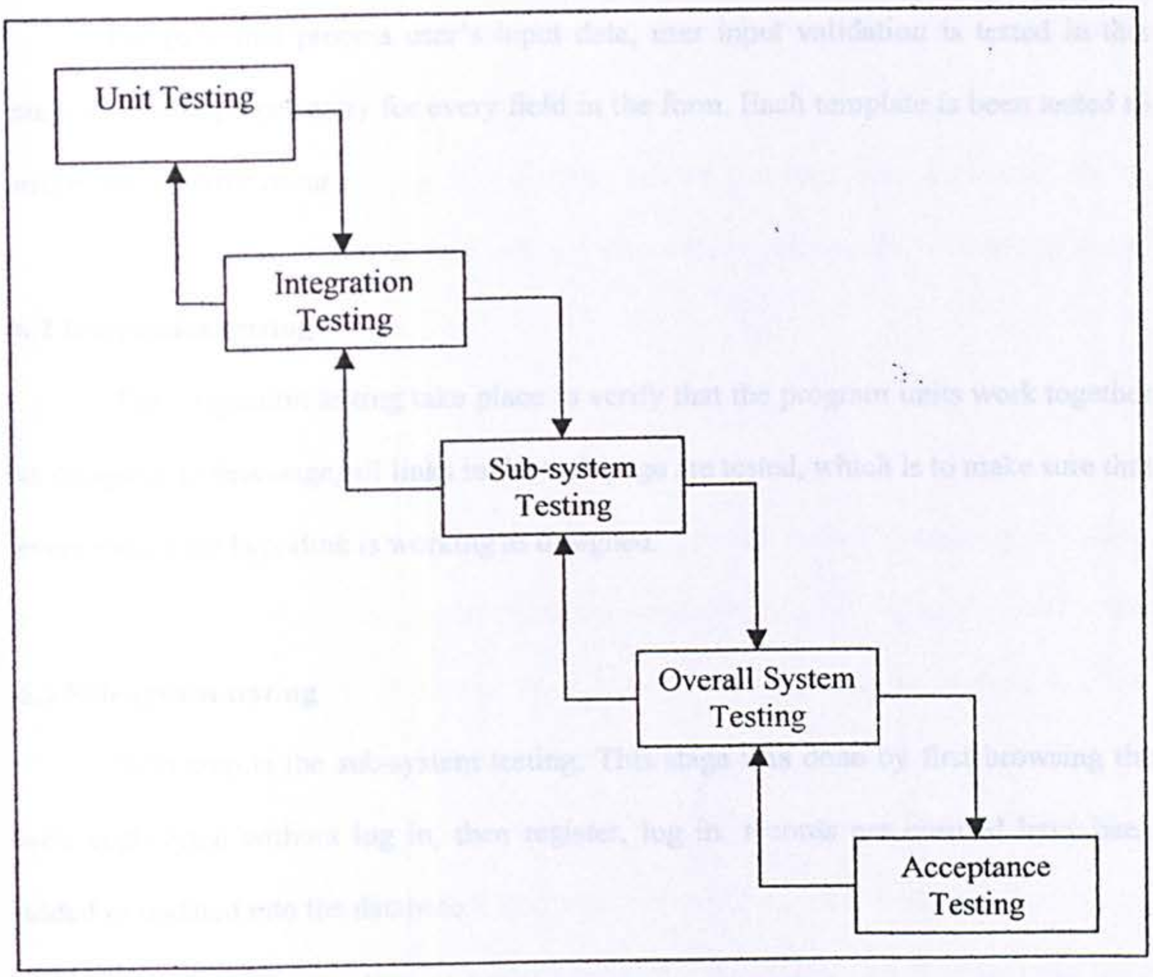


Figure 6.1: System Testing Process

6.1 Unit testing

Each program unit or sub-unit is tested at it owns. The unit testing was conducted throughout the implementation once a new unit was successfully built up. Each unit is tested independently to ensure it operates correctly.

For E-Bibliographic, every function is tested separately. It is to make sure that every function is doing what is exactly and also to ensure every link is correct. And also, it is tested to ensure whatever data input by the user is correct display on particular page.

For page that process user's input data, user input validation is tested in this stage to ensure proper entry for every field in the form. Each template is been tested to make sure no error occur.

6.2 Integration testing

The integration testing take place to verify that the program units work together as designed. In this stage, all links in the web page are tested, which is to make sure that every one of the hyperlink is working as designed.

6.3 Sub-system testing

Next step is the sub-system testing. This stage was done by first browsing the web application without log in, then register, log in. records are ensured have been added or updated into the database.

6.4 Overall System Testing

Then, next testing is the Overall System Testing. This testing is used to ensure that all the components or modules of the system are functioning properly.

6.5 Acceptance Testing

This stage is an important part in the system testing. It is been done by the user of the system. It is commences when the system is ready to use. Users involvement in this stage is to make sure the system meets their understanding of requirement, which be different from the developer. The feedback or comment from the user is important to improve and upgrade the system to fulfill their needs.

The comments and suggestions made by the user are been collected. Every functions of the system either at the back or front end, is work exactly without any error.

6.2 Neural Network Training

Improving Generalization

One of the problems that occurs during neural network training is called overfitting. The error on the training set is driven to a very small value, but when new data is presented to the network the error is large. The network has memorized the training examples, but it has not learned to generalize to new situations.

One method for improving generalization is to use a network that is just large enough to provide an adequate fit. If the larger network had been used, the network can create more complex function. If a small enough network been used, it will not have enough power to overfit the data.

Unfortunately, it is difficult to know before hand how large a network should be for a specific application. Two other methods for improving generalization that are implemented in the neural network:-

Regularization

Regularization involves modifying performance function, which normally chosen to be the sum of squares of the network errors on the training set.

Early Stopping

In this technique the available data is divided into three subsets. The first subsetis the training set, which is used for computing the gradient and updating the network weights and biases. The second subset is the validation set. The error on the validation set is monitored during the training process. The validation error will normally decreased during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set will typically begin to rise. When the validation error increases for a specified number of iterations, the training stopped, and the weights and biases at the minimum of the validation error are returned.

The test set error is not used during the training, but it is used to compare different models. It is also useful to plot the test the test error during the training process. If the error in the test set reaches a minimum at a significantly different iteration number than the validation set error, this may indicate a poor division set of the data.

CHAPTER VII

DISCUSSION

Chapter 7: Discussion

Several problems were encountered in aspects of software (interfaces, database and logic error) and hardware during the development of the system. Some are easily solved meanwhile others take time and some consultation or advise to solve it.

In order to get the system's strengths and limitations, user's involvement is a must. Improvements on the present system and potential future enhancements are based on the suggestions and evaluation result from the user.

7.1 Problems Encountered and Solutions

Below here is a list of encountered problems and its solution. Some solution may come easily but there were those that required an alternative solution.

7.1.1 Problem in development tools

In order to develop a good and dynamic web page, one must possess a good knowledge and experience in the development tool, and lack of this aspects have cost me most of the time in the system development. This is also due to time constraint.

SOLUTION: I have overcome this problem with reading and trying all sorts of ideas that will be used in the development. Beside that, I used the method of trial and error.

7.1.2 Burden of other subject

Beside WXES 3182, I also take another 5 subjects on this semester. It burdens me a lot. Besides developing the system, I have to done some of the assignments from other subject.

SOLUTION: Most of all is the help that come from my supervisor. A lot space been given to me.

7.2 System Strengths

7.2.1 Easy to access

User can easily access the system with the help of the graphical user interfaces.

7.2.2 Transparency

The system is transparent to the user. One does not have to posses' knowledge where the database resides, how the system structured and etc. For example, users don't need to know how to insert and retrieve data from the database.

7.2.3 User friendly

The graphic user interface of the system is easy to use for the user. Simple and easily understandable standard control objects such as add, cancel and close button are available in the system. These predictable user interfaces will enable potential user to shorten the learning curve of using the system.

7.2.4 Security and Integrity

User identification and password must be keyed in and verified, before any changes can be made into the database. With this method, it provides the security to the system from any unwanted parties. Beside than that, it's also ensure the integrity of data been entered in the database.

7.2.5 Search function

The system provides search functions for users to acquire information from the database. The system displays the search results as a summary of the information available with an option to view the full details of their particular result.

7.3 System Limitations

7.3.1 Lack of security features

Security is only enforced through the login function, where a user's identification and user-level information are stored in the session object. Since there are different levels of user access, some measure needs to be done to avoid users from accessing unauthorized pages.

7.3.2 Graphic User Interface

The graphic user interface available in the system, can still be enhance more multimedia technologies in order to attract users to the system.

7.4 Future Enhancements

7.4.1 Development of a Powerful Fitness Function

Maybe the most powerful function in this system goes to fitness function. It cost most of spaces in my head. The system should predict optimally when the fitness function is consistent with the input.

CONCLUSION

Conclusion

Application of AI in Medicine has been carried out with large testing ground from Neural Network alone and hybrid: Neural Network and Genetic Algorithm. From the testing, shown that Neural Network alone fail to train with incomplete data, where all the incomplete data have been set to NaN.

Hybrid model have 2 kind of testing part, where the first part is been carried in which the data is being test and train with one at a time, but there is no regularization and early stopping has been applied to because the data have been expand to 300*15*13 matrices. So the training using the regularization seem impossible to applied to the model, but there is some advantage in using Genetic Algorithm because all the NaN data are ignored during training. The setbacks in using Genetic Algorithm are, all part of the functions have to be developed from scratch, which have take a lot of time until it can be used.

The second part of testing is where all the binary representation has been converted back to decimal and by using regularization and early stopping; it has shown that the data loss in the training was the major setback in the training. Although it inherited the advantages of Genetic Algorithm, but it loses the beneficial part of Neural Network.

MISCELLANEOUS

RESULT

Training with Neural Network

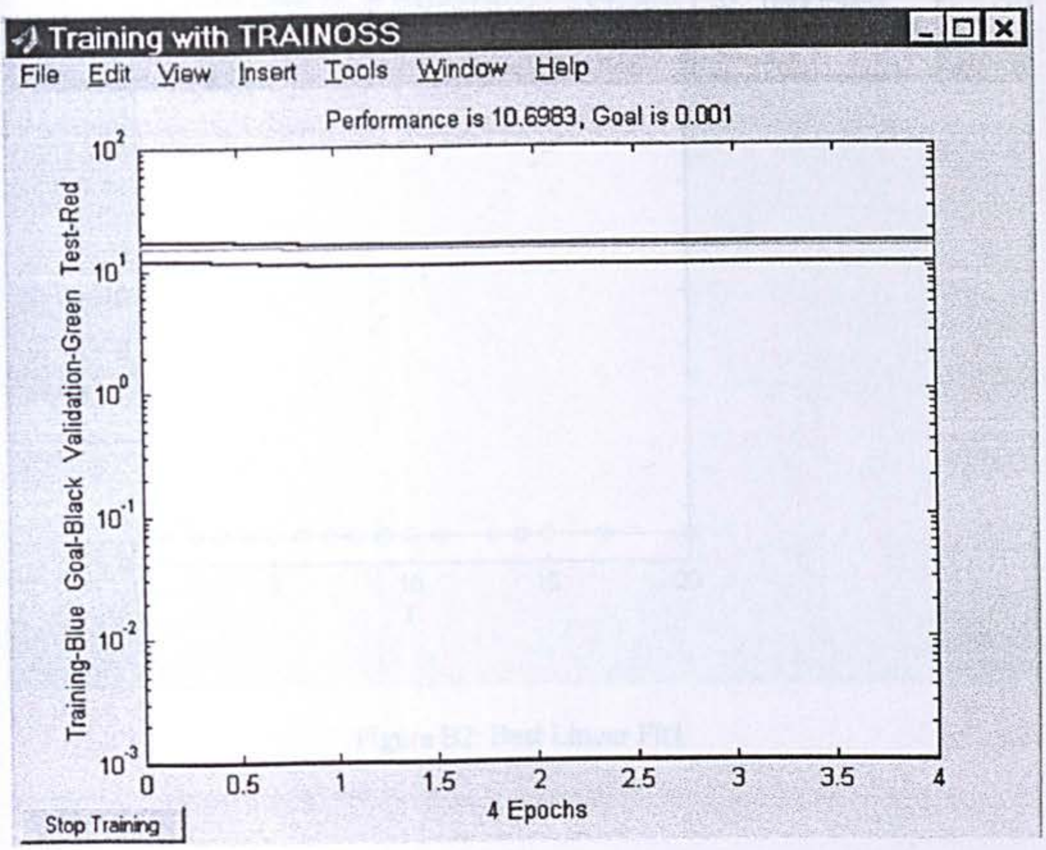

Training with TRAINOSS
File Edit View Insert Tools Window Help
Performance is 10.6983, Goal is 0.001
Training-Blue Goal-Black Validation-Green Test-Red
4 Epochs
Stop Training

Figure B1: Training with TRAINNOSS

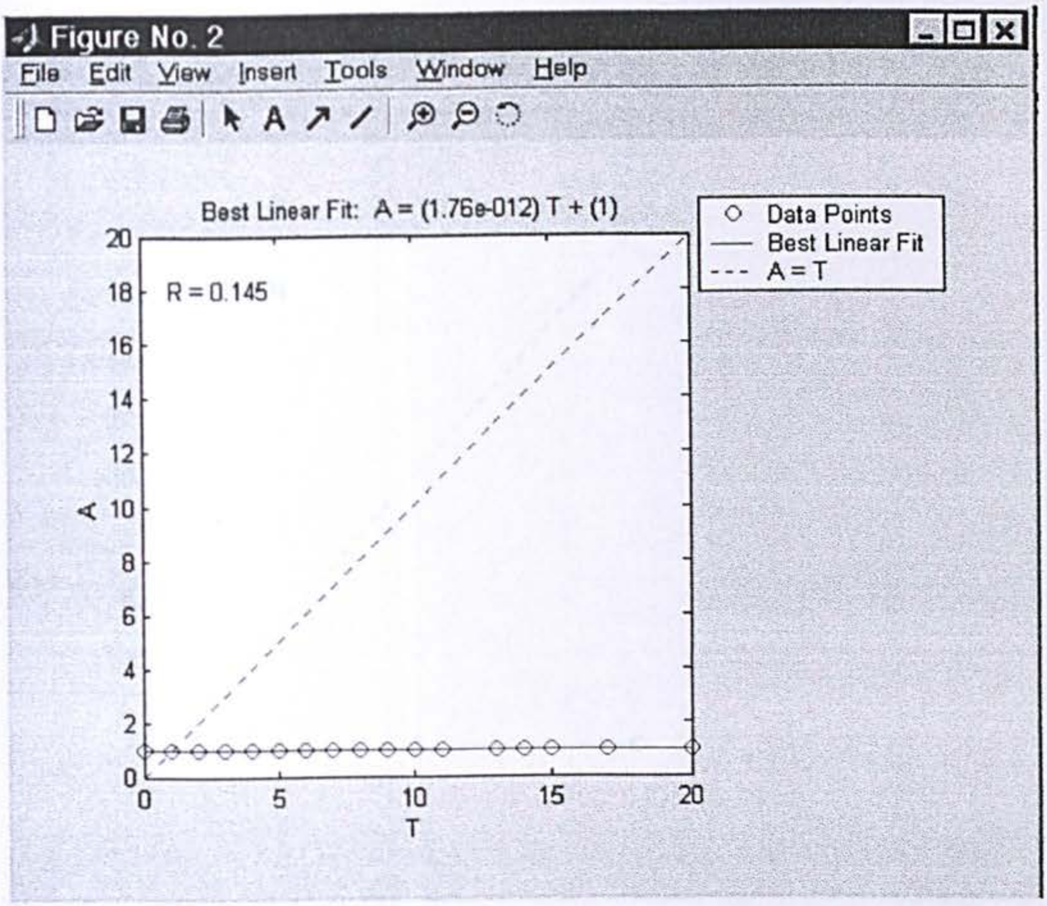


Figure B2: Best Linear Fit1

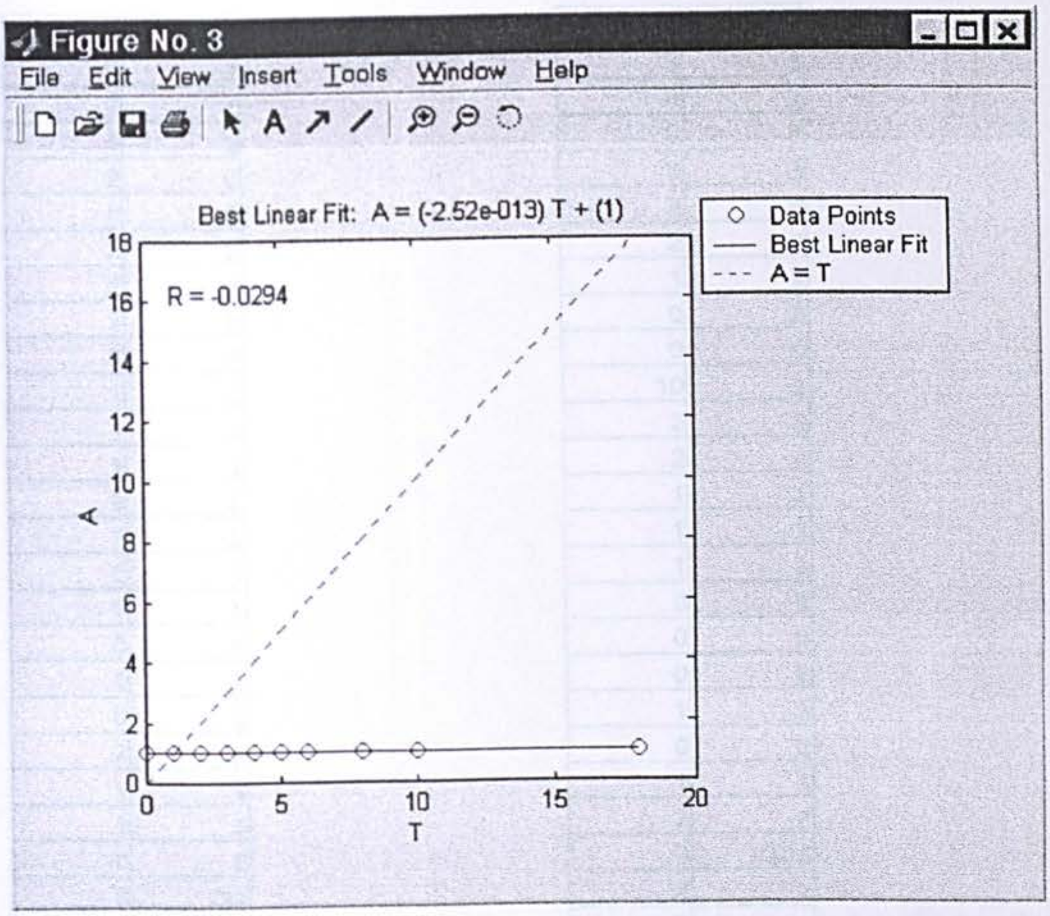


Figure B3: Best Linear Fit2

Training with Genetic Algorithm and Neural Network Without Regularization and Early Stopping

Real values	Predicted values
1	1
1	1
1	1
1	1
0	0
0	0
3	3
6	6
2	2
0	0

Real values	Predicted values
0	0
1	1
1	1
1	1
1	1
1	1
2	2
1	1
1	1
1	1

0	0
3	3
2	2
1	1
2	2
0	0
2	0
2	2
2	2
3	3
1	1
1	1
1	1
2	2
1	1
5	5
1	1
0	0
5	5
0	0
2	2
1	1
1	1
0	0
9	137
1	1
1	1
1	1
1	1
6	6
3	3
4	4
2	2
3	3
0	0
1	1
5	5
1	1
1	1
5	5
0	0
1	1
2	2
1	1
2	2
1	1
2	2
4	4

4	4
3	3
3	3
0	0
3	3
2	2
2	2
1	1
0	0
0	0
10	2
3	3
2	2
1	1
1	1
1	1
0	0
0	0
0	0
1	1
0	0
1	1
2	2
9	4105
4	4
3	3
4	4
6	6
1	1
2	2
13	4109
1	1
1	1
1	1
0	0
1	1
2	2
0	0
1	1
5	5
1	1
3	3
1	1
3	3
2	2
2	2
2	2
15	7

1	1
1	1
1	1
0	0
3	3
1	1
1	1
1	1
1	1
1	1
0	0
0	0
5	5
1	1
1	1
1	1
8	8
1	1
3	3
1	1
2	2
0	0
1	1
6	6
1	1
2	2
0	0
0	0
1	1
6	6
1	1
2	2
0	0
1	1
0	0
1	1
0	0
3	3
1	1
1	1
2	2
3	3
1	1
1	1
2	2
1	1
0	0
4	4

1	1
1	1
1	1
1	1
2	2
1	1
3	3
1	1
4	4
3	3
2	2
1	1
5	5
2	2
2	2
5	5
0	0
3	3
2	0
3	3
2	2
1	1
2	2
3	3
3	3
3	3
6	6
1	1
6	6
6	6
0	0
1	1
1	1
0	0
4	4
0	0
0	0
4	4
1	1
2	2
3	3
4	4
2	2
0	0
8	136
3	3
0	0
0	0

3	3
9	1
0	0
1	1
15	7
0	0
10	10
3	3
0	0
0	0
17	17
2	2
5	5
1	1
0	0
0	0
0	0
2	2
1	1
0	0
1	1
6	6
3	3
1	1
3	3
0	0
5	5
1	1
4	4
1	1
0	0
11	3
0	0
0	0
1	1
1	1
2	2
2	2
0	0
0	0
1	1
1	1
1	1
0	0
0	0

1	1
0	0
1	1
0	0
18	18
1	1
1	1
2	2
4	4
0	0
1	1
1	1
2	2
0	0
1	1
2	2
2	2
3	3
0	0
2	2
1	1
0	0
2	2
0	0
10	2
2	2
10	138
2	2
4	4
1	1
20	20
1	1
0	0
2	2
0	0
9	4105
2	2
1	1
2	2
3	3
0	0
0	0

User Manual

User Manual is a documentation made by developer as users' reference in using the system that has been developed. It gave instruction on how to use the system. Maybe a good user manual should provide deep explanation in how to use the system, maintaining the system and troubleshooting the system.

Login Page

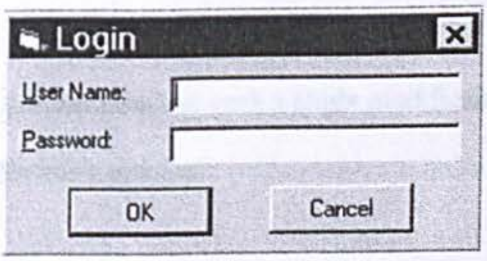
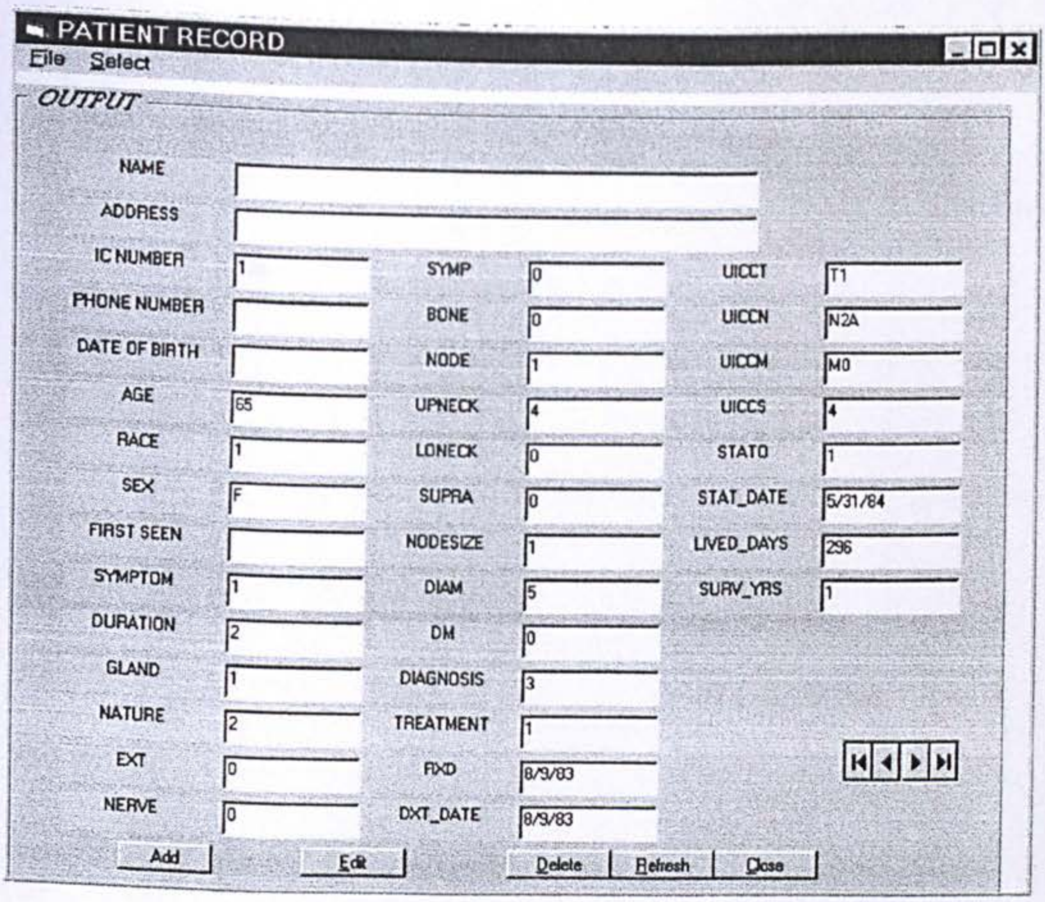


Figure A1: Login page

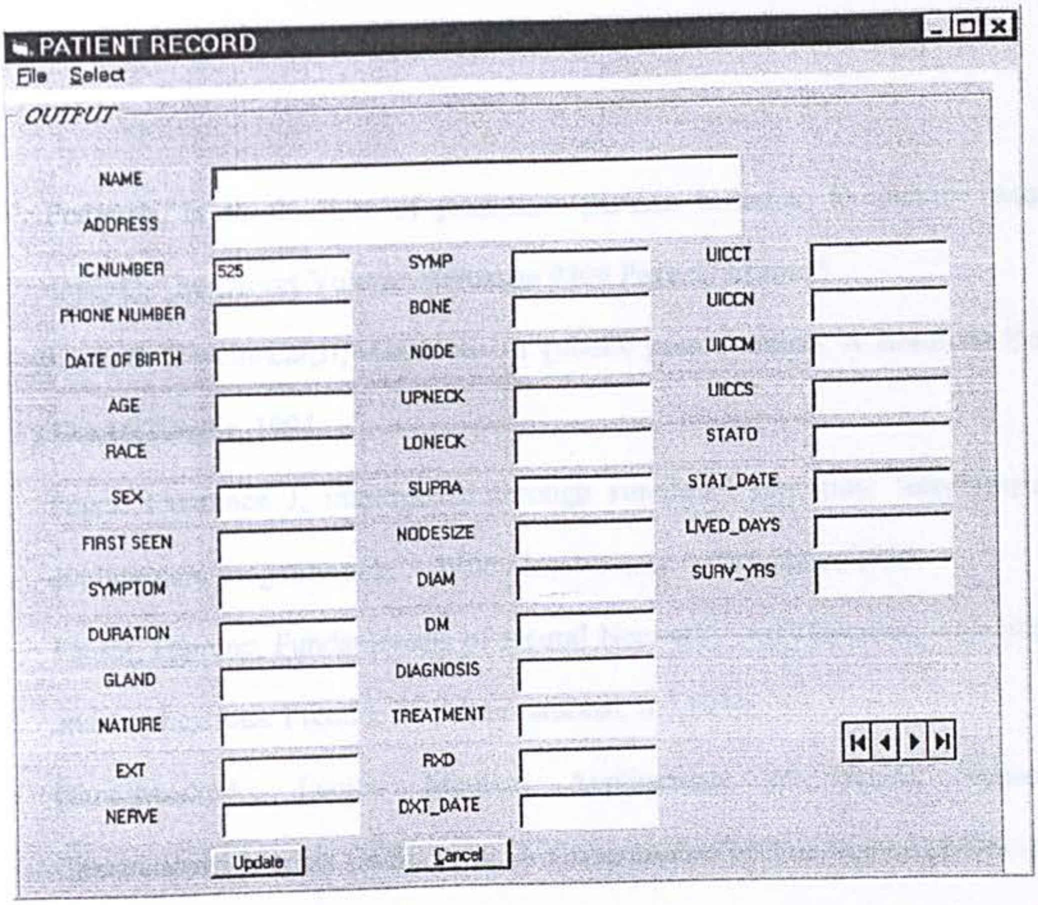
Before one can use the system, he or she should be granted a user name and a password based on position he or she held in a hospital.



NAME					
ADDRESS					
IC NUMBER	1	SYMP	0	UICCT	T1
PHONE NUMBER		BONE	0	UICCN	N2A
DATE OF BIRTH		NODE	1	UICCM	M0
AGE	65	UPNECK	4	UICCS	4
RACE	1	LONECK	0	STATO	1
SEX	F	SUPRA	0	STAT_DATE	5/31/84
FIRST SEEN		NODESIZE	1	LIVED_DAYS	296
SYMPTOM	1	DIAM	5	SURV_YRS	1
DURATION	2	DM	0		
GLAND	1	DIAGNOSIS	3		
NATURE	2	TREATMENT	1		
EXT	0	FIXD	8/9/83		
NERVE	0	DXT_DATE	8/9/83		

Figure A2: Patient Record form

The system uses only a single form but with a slight modification on the button based on the module that the users use.



NAME					
ADDRESS					
IC NUMBER	525	SYMP		UICCT	
PHONE NUMBER		BONE		UICCN	
DATE OF BIRTH		NODE		UICCM	
AGE		UPNECK		UICCS	
RACE		LONECK		STATO	
SEX		SUPRA		STAT_DATE	
FIRST SEEN		NODESIZE		LIVED_DAYS	
SYMPTOM		DIAM		SURV_YRS	
DURATION		DM			
GLAND		DIAGNOSIS			
NATURE		TREATMENT			
EXT		RXD			
NERVE		DXT_DATE			

Figure A3: Add data form

If the users want to update the data, he or she have to click on the add button and the Add Data form will appear. After the entire field have been filled, user might click on update button to add data or cancel if the user wants to terminate, and/or return to the page before.

REFERENCES

i. Petricoin, E. et al., "Use of proteomic patterns in serum to identify ovarian cancer", The Lancet Volume 359 Issue 9306 Page 572 (2002).

ii. Kenneth E. Kinnear,Jr, Advances in genetic programming: A Bradford Book, The MIT Press, 1994.

iii. Fogel, Lawrence J, intelligence through simulated evolution: forty years of evolutionary programming: A Wiley-Interscience publication, 1999.

iv. Fauset, Laurene, Fundamentals of Neural Networks: Architectures, Algorithms, and Applications: Prentice Hall International, Inc,1994.

v. Ohno-Machado, Lucila, Medical Applications of Neural Networks: Connectionist Models Of Survival, A Dissertation For The Degree of Doctor of Philosophy, Department of Computer Science, Stanford University, March 1996.

vi. U.M Fayyad, G. Piatetsky-, P- Smyth, and R. Uthurusamy, editors. Advances in Knowledge Discovery and Data Mining. AIII Press/MIT Press, March 1996.

vii. P.J.F. Lucas. Prognostic Methods in Medicine. Artificial Intelligence in Medicine, 1999.

viii. Pfleeger, Shari Lawrence, Software Engineering: Theory and Practice, Prentice Hall, Inc.