PREDICTION OF DENGUE OUTBREAKS IN PERSPECT OF BANGLADESH USING ARTIFICIAL INTELLIGENCE

JONY AKTER

MASTER OF SCIENCE IN INFORMATION AND COMMUNICATION ENGINEERING

NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

PREDICTION OF DENGUE OUTBREAKS IN PERSPECT OF BANGLADESH USING ARTIFICIAL INTELLIGENCE

JONY AKTER

This Thesis Report submitted in partial fulfillment of the requirements
for the award of the degree of
Master of Science in Information and Communication Engineering

Department of Information and Communication Engineering
NOAKHALI SCIENCE AND TECHNOLOGY UNIVERSITY

MARCH 2021

SUPERVISOR'S DECLARATION

We hereby declare that we have checked this thesis and in our opinion, this thesis is adequate in terms of scope and quality for the award of the degree of Master of Science in Department of Information and Communication Engineering of Noakhali Science and Technology University.

_____

Name of Supervisor: DR. MD. ASHIKUR RAHMAN KHAN

Position: PROFESSOR& CHAIRMAN

Date:

_____

Name of Co-Supervisor: ZAYED-US-SALEHIN

Position: ASSISTANT PROFESSOR

Date:

STUDENT'S DECLARATION

I hereby declare that the work in this thesis is my own except for quotations and summaries which have been duly acknowledged. This thesis report has not been accepted for any degree and is not concurrently submitted for the award of other degrees.

_____

Name: JONY AKTER

ID Number: BKH1811MSc107F

Date:

ACKNOWLEDGEMENTS

I am grateful and would like to express my sincere gratitude to my supervisor Professor Dr. Md. Ashikur Rahman Khan for his germinal ideas, invaluable guidance, continuous encouragement, and constant support in making this research possible. He has always impressed me with his outstanding professional conduct, his strong conviction for science, and his belief that an MSc program is only the start of a life-long learning experience. I appreciate his consistent support from the first day I applied to the graduate program to these concluding moments. I am truly grateful for his progressive vision about my training in science, his tolerance of my naive mistakes, and his commitment to my future carrier. I also would like to express very special thanks to my co-supervisor Zayed-Us-Salehin for his suggestions and co-operations throughout the study. I also sincerely thank you for the time spent proofreading and correcting my many mistakes.

My sincere thanks go to all my lab mates and members of the staff of the Information and Communication Engineering Department, NSTU, who helped me in many ways and made my stay at NSTU pleasant and unforgettable.

I acknowledge my sincere indebtedness and gratitude to my parents for their love, dream, and sacrifice throughout my life. Special thanks should be given to my committee members. I would like to acknowledge their comments and suggestions which were crucial for the successful completion of this study.

ABSTRACT

Dengue Fever is a disease that has grown worldwide in the last few years. The information about the patients can be maintained with clinical documents. By keeping huge volume of clinical documents we can easily predict the occurrence of dengue disease in the patients. Dengue is considered to be one of the vital diseases which are spreading in more than 110 countries. It is a vector borne disease caused by the mosquitoes of female Aedes Albopictus and Aedes Aegypti which are well suited human environment. With nearly 45,000 cases reported all over Bangladesh in the last year. Dengue fever has become a major health hazard in Bangladesh over the past few years. Several studies show that which variable is related to the disease, however, as far as we know there is no effective study in Bangladesh that reveals this relation. This research shows the accuracy of different algorithms of Artificial Neural Network (ANN) to predict Dengue outbreaks. Firstly, data is collected from different hospitals, and then data is normalized. Data is analyzed to see the infection rate of different parameters. Then data is splitted for training and testing and finally find the best accuracy for different Multi-layer perceptron algorithms such as Scaled Conjugate Gradient (SCG), Levenberg–Marquardt (LM) based on back propagation algorithm and Learning Vector Quantization (LVQ). In this work, an analysis of the influence of variables is performed and shows the efficiency of the neural networks is used to predict the number of disease cases. Accuracy of SCG, LM and LVQ algorithm is 87.1%, 95% and 90.3% with MSE 0.137, 0.0241 and 0.0967 respectively. And decision tree algorithm gives 98.92% in training stage but 90% accuracy gives in validation and testing. LM outperforms with minimum MSE than other algorithm. Thus, this work finds an efficient prediction model for dengue cases for districts of Bangladesh.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

Symbol	Meaning
μ	Mu
λ	Lambda
α	Alpha

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
DF	Dengue Fever
LS- SVM	Least Square – Support Vector Machines
GIS	Geographic Information System
IDE	Integrated Development Environment
MLP	Multilayer Perceptron
AIC	Akaike Information Criterion
BIC	Bayesian Information Criterion
MAPE	Mean Absolute Percentage Error
MFNN	Multilayer Feed-Forward Neural Networks
NNM	Neural Network Model
NLRM	Nonlinear Regression Model
DHF	Dengue Hemorrhagic Fever
SVR	Support Vector Regression
ADTree	Alternating Decision Tree
REP	Reduced-error Pruning
CART	Classification And Regression Tree
SMO	Sequential Minimal Optimization
EWSORA	Entropy Weighted Score based Optimal Ranking Algorithm
BIA	Bioelectrical Impedance Analysis
PSO	Particle Swarm Optimization
RS	Rank Search
KNN	K – Nearest Neighbor
MMRE	Mean Magnitude of Relative Error
MSE	Mean Squared Error
ROC	Receiver Operating Characteristic
WBC	White Blood Cells
IgG	Immunoglobulin G
IgM	Immunoglobulin M
LM	Lavenberg-Marquardt

SCG	Scaled Conjugate Gradient
LVQ	Learning Vector Quantization
TN	True Negative
TP	True Positive
FN	False Negative
FP	False Positive
TPR	True Positive Rate
DT	Decision Tree

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

The introductory study of this thesis is carried out in this chapter. Since this research focuses on the prediction model for dengue outbreaks in Bangladesh; thus, background analysis on these sectors is carried out first. The problem statement is then introduced and the research objectives are described on this basis. At the end of this chapter the outline of this thesis is presented.

1.2 BACKGROUND

Dengue is a viral infection spread by mosquitoes that causes a mild flu-like disease and often causes a potentially fatal complication called extreme dengue. Around half of the world's population is at risk, affecting infants, young children and adults. Over the past 50 years, the prevalence of dengue has risen 30-fold. In over 100 endemic countries, up to 50-100 million infections are now estimated to occur annually. The primary vector that transmits the virus that causes dengue is the Aedes aegypti mosquito. Via the bites of an infectious female Aedes mosquito, which primarily gets the virus when feeding on an infected person's blood, the virus is transmitted to humans. Dengue fever virus' life cycle includes mosquitoes' function as transmitters (or vectors) and human beings as the primary victim and source. Humans become the main carriers and propagators of the virus after they get infected and serve as source of the virus to uninfected mosquitoes. The virus circulates in an infected individual's blood for 2 to 7 days, roughly at the same time as the individual experiences the fever. The infection can be spread by the Aedes mosquito, which usually occurs within 4 to 5 days, to a

maximum of 12 days, after the first symptoms appear [1]. In Bangladesh, an effective prediction model is not developed to predict dengue outbreaks. The limited variables are used as an input parameter to predict dengue disease previously. There are many variables that can be used as an input parameter to predict dengue outbreaks. Such as Week, Year ,Age, Sex, District , Headache, Myalgia, Arthralgia, Retro-ocular pain, Cough and dyspnea, Shock and so many. The four important features namely mean temperature, mean relative humidity, total rainfall and the total number of dengue confirmed-cases were very effective in predicting the number of dengue confirmed-cases. In many areas ranging from computer vision to business forecasting through available data, ANNs have succeeded in solving many engineering problems. They are great tools because of their ability to learn from instances, because there is no need to mathematically model every individual event. A large number of linked unit networks contain inputs, outputs and hidden nodes. The interconnected units a single work unit essentially summarizes the weighted activation of its input, converts this number to an active function and transfers the resulting function to its output. Neural networks provide a closer approach to human perception and recognition than traditional computing [2]. The prediction of dengue cases in Singapore using the average temperature, the relative humidity and the total amount of rain per week, obtaining good results by using artificial neural networks [3]. The prediction model of dengue outbreaks in Bangladesh is obtained by using the data of 2000 to 2007 which is collected from metrological department of Bangladesh. Climatic factors, i.e. rainfall, maximum temperature and relative humidity are used as inputs [4].

1.3 PROBLEM STATEMENT

The first official outbreak of dengue fever in Bangladesh was in 2000, and since then the number of hospitalized patients has exceeded 3000 patients six times—6232 in 2002, 3934 in 2004, 3162 in 2015, 6060 in 2016, 10148 in 2018, and100107 as of Nov 30, 2019, with estimated projections of more than 112000 cases by the end of 2019. In parallel with the major epidemics in 2018 and last year’s outbreak, 26 deaths and 129 deaths, respectively, had been officially documented by the government surveillance systems with a clear predominance of cases and fatalities during the summer months

(July to November), even if the death tally is likely to be much higher because of actual under-reporting. The geographical spread of the cases reported in 2019 affects all districts of the country, exhibits a clear predominance in men (64·11% for men vs 35·89% for women), and primarily encompassed younger adults (whereby people aged 15–35 years accounted for 51·42% of 29855 total cases). As national surveillance is passive and only government hospitals are included, it is highly likely that substantial underreporting is taking place. Furthermore, the operational surveillance is not based on appropriate methods, such as the WHO projection done in July, 2019, where an estimated 358 960 people were deemed to be infected compared with only 7179 cases in official reports.8 Based on the aforementioned official number of cases in November, 2019, as many as 3–4 million people could have conceivably been infected. We cannot underscore the seriousness of the current epidemic, which is unfortunately being handled with great laxity by the country's authorities, as shown not only by the marked under-reporting, but also by the absence of health awareness campaigns targeting both the general public and health professionals. Such campaigns that aim for earlier and more consistent recognition and supportive clinical management of dengue cases are a major factor underlying reduced mortality for this highly contagious disease, and are urgently needed. Dengue fever has become a major health hazard in Bangladesh over the past few years. Many people died of dengue in Bangladesh in 2019. It was a situation where people started fearing dengue like all other major diseases. Despite many attempts, government and different health organization couldn't control the spread of dengue. The existing dengue prediction model for Bangladesh based on the definition of an epidemiological outbreak. The current time for decision-making in the analysis of dengue cases is limited to a maximum of one week in advance, which may not be enough. The determination of the optimal parameters values for a neural network destined to make predictions of dengue cases is a complex task due to the large number of possible combinations. The main problem to be solved is the lack of a predictive model of the number of dengue cases with a reasonable anticipation time that will allow the authorities to make better decisions in a timely manner in Bangladesh.

The motto of this work is to overcome previous difficulty and predict dengue outbreaks using many variables by using artificial neural network in the perspective of Bangladesh and increase performance of prediction model of dengue outbreaks.

1.4 RESEARCH OBJECTIVES

This research paper is worked on artificial neural networks and classification algorithms and provides better result that will directly help to the physicians for accurate diagnosis. The main purposes of research are following:

i. To analyze the symptoms of dengue disease to detect the reasons of dengue fever.
ii. To predict dengue outbreaks in respect of Bangladesh.
iii. To make a comparison among distinct methods in prediction of dengue outbreak.
iv. To find out the optimal technique for dengue disease prediction.

1.5 SCOPE OF THE STUDY

The data used in this study to predict sourced mainly from two locations. The data is collected from peoples who tested for dengue disease from different hospitals of Noakhali and Dhaka. Data is collected from people who are tested for dengue by dataset paper. The required tools to carry out this work are Matlab and Anaconda3.

1.6 THESIS OUTLINE

This thesis consists of 5 chapters and these chapters are organized as follows, in chapter 1 it provides a detailed discussion on the importance of the work that has been done and why the current topic is selected for thesis. In the next that is chapter 2 is about some related works of this thesis is described in this section. It gives the information about dengue outbreaks of different countries using different classification algorithms and techniques. Next is chapter 3 which introduces the dataset and the methods and explains deep understanding about the algorithms and evaluating criteria. Then the chapter 4 describes experimental plans and prediction test results on evaluation criteria and shows comparison of different algorithm's performance. Finally, in Chapter

5, paper will end up as allowing to know why this study is a valuable for effective techniques in medical diagnosis in the conclusion.

CHAPTER 2

LITERATURE REVIEW

2.1 INTRODUCTION

In this chapter, different past works related with dengue data collection, prediction methods and prediction accuracy are described in short. These three things which are related to dengue disease prediction are discussed in literature review. And lastly, the summary of this chapter which is the outcome of literature review is described briefly.

2.2 LITERATURE REVIEW

The literature review is performed on the works which relates on these sectors. The novelty of this thesis compared to the related previous works can be easily understood from the summary of literature which is presented in Table 2.1.

Nishanthi et al. [1] had tried to create an Artificial Neural Network Model (ANN) that predicts Dengue outbreaks by analyzing the impact on current Dengue cases from weather variables as well as previous Dengue cases. In this paper, the criteria used to forecast dengue outbreaks include average temperature, average relative humidity, rainy days per week, cumulative rainfall and previous cases. A correlation analysis between each individual variable and current Dengue cases determines the parameters and the particular time lags. As a result of this report, the ANN was established to predict Dengue outbreaks in the Kandy district in Sri Lanka. The probable number of Dengue cases Aburas et al. [2] attempts to estimate, by using an ANN model. This model forecasts three weather parameters – mean temperature, mean rainfall and average

relative humidity of the region during the period under consideration, based on a total number of cases recorded during the preceding period. ANN gave promising results in Singapore's Dengue prediction and correlation to 0.86 to 0.91.Gultekin et al. [3] proposed the use of artificial neural networks for the prediction of dengue cases in Singapore using the average temperature, the relative humidity and the total amount of rain per week. Data is collected from Singaporean National Environment Agency and prediction output results correlation to 0.70 to 0.76. Climatic factors, i.e. rainfall, maximum temperature and relative humidity were significantly correlated with monthly reported dengue cases in Dhaka city is shown in [4]. Yuhanis et al. [5] had used Least Square – Support Vector Machines (LS – SVM) model to predict dengue cases and prediction accuracy is approximately 86.84%. Data is collected from five districts of Selangor. In [6], an ANN model is developed to predict Dengue using the records of Western and Southern provinces. The data includes epidemiological records on DF cases reported, meteorological data on weather parameters and census data. It focuses on identifying the risk factors for DF outbreak using a regression analysis, predicting DF cases using an ANN and simulating the results using a GIS application. The accuracy of the model for the test set is stated as 86.5% and 89.5% for the two locations considered. Balasaravanan et al. [7] had implemented a data mining technique called ANN is used here for classification of data to classify diseases. In this paper, they analyzed weather variable i.e. rainfall, humidity rate, temperature rate and number of dengue cases and experimented with Weka and Netbeans IDE and showed the more accuracy than the CART algorithm which is 92%. Ughelli et al. [8] had studied an analysis of the influence of variables is performed and the efficiency of the neural networks is used to predict the number of disease cases. Climatic variables like rainfall, humidity rate, temperature rate and number of dengue cases of Paraguay is used as inputs. This model can be changed for each district, since the conditions are different, reason why the training and prediction was realized for each of them individually, avoiding to generate noise in the prediction of others. ANN-MLP and SVM polynomials had performed as classifiers in the diagnosis of dengue disease with accuracy 96%, sensitivity 96%, and specificity 97%, within the spatial and temporal context determined by the dataset is shown in [9]. The risk factors for the presence of Aedes mosquitoes, to identify the types of most productive and key containers and to

estimate the effects of climatic factors on Aedes abundance in the city of Dhaka, Bangladesh is determined by Kumar et al. [10].

Padet et al. [11] demonstrated the important roles of female mosquito infection rates from the previous season and climate factors (represented as seasons) in dengue outbreaks. Incorporating these two factors in the model significantly improves the predictive power of dengue hemorrhagic fever forecasting models, as confirmed by AIC, BIC, and MAPE. The exclusive use of climate factors from similar locations decreases a model's prediction power. The multivariate Poisson regression, however, effectively forecasts even when climate variables are slightly different. A proficient methodology – SVM with random forest classifier with its associated Gini feature is presented by Fathima et al. [12] for better performance and accuracy 90.78% is found. Shakil et al. [13] proposed approach i.e. Naive Bayes and J48 with 10 cross validation to evaluate data and compare results by using Weka tool. Furthermore in order to validate their approach, they used a dengue dataset with 108 instances but weka used 99 rows and 18 attributes to determine the prediction of disease and their accuracy using classifications of different algorithms to find out the best performance. Iqbal et al. [14] proposed to develop a machine learning model to predict Dengue. They took various machine learning classifiers ranging from simple classifiers, like Decision Tree, Naïve Bayes, Model Tree, to complex algorithms such as Support Vector Machines, Neural Networks, Gene Expression Programming, Genetic Programming and ensemble classifiers. The algorithm giving the highest prediction accuracy will be considered for the development of Dengue Prediction Tool. Ishaq et al. [15] presented a systematic literature review on expert systems which are used for identification and overcoming of Dengue fever. These expert systems require knowledge of the relevant problem and techniques to infer the result in order to make decisions. The literature review provides a comparison of techniques, methodologies, limitations and advantages of different Dengue expert systems. Ibrahim et al. [16] described a non-invasive prediction system for predicting the day of defervescence of fever in dengue patients using artificial neural network. The developed system bases its prediction solely on the clinical symptoms and signs and uses the multilayer feed-forward neural networks (MFNN). The results show that the proposed system is able to predict the day of defervescence in dengue patients with 90% prediction accuracy. Husin et al. [17] presented architectures for predicting

dengue outbreak incorporating Neural Network (NNM) and Nonlinear Regression (NLRM) models. The study was done using dataset of dengue cases and rainfall level for five districts in Selangor. The data were from 2004 to 2005. From the undertaken experiment, it is shown that NNM yields better output compared to NLRM where NNM's prediction accuracy is 90%. Rachata et al. [18] studied predicting dengue hemorrhagic fever (DHF) in Thailand. In the study, an automatic prediction system for DHF is proposed by utilizing entropy technique and Artificial Neural Network (ANN). Entropy is used to extract the relevant information that affects the prediction accuracy. Later, the supervised neural network is applied to predict future DHF outbreak. The entropy technique produces 85.92% accuracy while only 78.16% when entropy technique is not applied. Lee et al. [19] studied on Wavelet transformation for data pre-processing before implementing Support Vector Machines (SVM)-based Genetic Algorithm in analyzing and predicting the dengue outbreak. The model was developed based on data collected in Singapore, from 2001 to 2006. From their study, they found that, for predicting, Support Vector Regression (SVR) performed better compared to a simple linear regression and also more reliable, even with the present of over-fitting. An alternating decision trees for early diagnosis of dengue fever is used. The ADTree correctly classifies 84 % of cases as compared to J48 which can classify only 78% of cases correctly are shown in [20].

In [21], an analysis of dengue infection is presented using data mining decision tree. In this paper, two datasets have been used from two different hospitals Srinagarindra Hospital and Songklanagarind Hospital, each having more than 400 attributes. Four classification algorithms have been used in this paper for experimental purpose. The first and second experiment test got an accuracy of 97.6% and 96.6%. The third experiment extracts useful knowledge. In fourth experiment of day0 accuracy is very low as compared to other three experiments. Mutsuddy et al. [22] emphasized only in terms of environment and seasonal variation on the epidemiology aspects of dengue-related morbidity and mortality. The incidence of dengue has also shifted as more dengue cases are recorded during the pre-monsoon season as a result of climate change and other factors, such as quick, unplanned urban growth and other urban parameters. They also projected that the climate changes after 2014 may have triggered such environmental imbalances that would lead to more dengue cases in the pre-monsoon

season. Bhavani et al. [23] tried to highly reliably predict dengue fever. For this reason were used the five classifiers REP Tree, J48, SMO, ZeroR and Random Tree. These techniques have been used with Weka Data Mining Tool on dengue data collection. These techniques have been tested and comparable to their performance. From the study, they have shown that SMO and J48 are well above other algorithms and have achieved a precision of 84% and 76%. In [24], an effective and helpful algorithm for medical data analysis and forecasts has been shown to have a novel feature selection algorithm called the Entropy Weighted Score based Optimal Ranking Algorithm (EWSORA). Dengue Dataset is framed by the compilation in real time of test results from several patients in the Thanjavur region of Tamil nadu. The observation is performed with the proposed approach on a real-time data base and the findings produced outperform current methods. A method used in dengue patients using the bioelectrical (BIA) and artificial neural network (ANN) diagnosis and classification of early risk in [25]. Patients with dengue were subsequently classified and quantified in terms of their risk categories and corresponding BIA parameters. For training and testing ANN, four criteria were used: fever, reaction, sex and risk group quantification. In dengue patients, the device can identify and diagnose the risk with a total predictive precision of 96.27%. A comparison analysis was performed on the basis of the Simple Classification and Regression Tree (CART), multi-layer perception (MLP) and C4.5 algorithms, which showed that the Simple CART algorithm is 100% effective in the classification of the patient affected or not in [26]. Classification methods are used for evaluating and contrasting their output in [27]. These are the Naive Bayesian, REP Tree, Random Tree, J48 and SMO techniques. The dataset was taken from Jhelum District Hospital (DHQ). WEKA has been used for the classification of data as a data mining tool. The systematic review of literature between 1980 and 2017 and the empirical evidence on climate change impacts in China on DF in [28]; the related DF incidence models and findings on how changes in meteorological factors can affect DF occurrences in China are discussed in addition. In China there is a lack of knowledge as regards how the spatio temporary distribution of DF will respond to climate change compared with some well-known research projects in the western countries. In 2000, during a community-wide outbreak of dengue fever and dengue hemorrhagic fever in Dhaka, Bangladesh, Ali et al. [29] used conventional and spatial analytical tools to characterize patterns of transmission. In order to obtain data on mosquito species and

breeding as well as disease consistent with dengue, a comprehensive household-level mosquito vector survey and interview was conducted. In this urban area, where the focus was mainly on indoor mosquito breeding and transmission, the study had effects to check the transmission of dengue. Classification algorithms were considered for evaluating classification performance in terms of Accuracy, Precision, Sensitivity, Specificity and false negative rate in classifying dengue haemraghic fever and dengue fever from patient's dataset which is collected hospitals of Pakistan in [30]. A model and prediction study was conducted in [31] using various techniques for data mining, especially by using three different selection algorithms such as Particle Swarm Optimization (PSO) and Rank Search (RS). Based on the characteristics chosen, the detection of dengue outbreak was based on three predictive modeling techniques (J48, DTNB and Naive Bayes). Data from the Department of Public Health, Seremban, Negeri Sembilan, Malaysia was obtained in this study.

Sheng et al. [32] studied Genetic Algorithm-Support Vector Machine (GA-SVM) which is utilized in predicting tax gross. In the developed model, GA is implemented to automatically tune the SVM parameters. The research was using tax gross data sets in China, from 1990- 2001. In order to improve the operation speed and generalization performance in the proposed model, datasets used is normalized before training process. The empirical results revealed that the proposed approach yields better results as compared to artificial neural network (ANN) and grey model (GM) in tax gross prediction. Dhamodharan et al. [33] had done prediction of liver disease using Bayesian Classification through Naive Bayes and FT tree algorithms. With the help of data mining techniques they have predicted and analyzed liver diseases using weka tool. They have also compared the outputs obtained from Naïve Bayes and FT tree algorithms and concluded that Naive Bayes algorithm plays a key role in predicting liver diseases. Solanki et al. [34] used weka as a data mining technique for classification of sickle cell disease prevalent in Gujarat. They have compared J48 and Random tree algorithms and have given a predictive model for classification with respect to a person's age of different blood group types. From there experimentation it can be inferred that Random tree is better algorithm as it produces more depth decisions respect to J48 for sickle cell diseases. Joshi et al. [35] had done diagnosis and prognosis of breast cancer using classification rules. By comparing classification rules such as

Bayes Net, Logistic, Multilayer Perceptron, SGD, Simple Logistic, SMO, AdaBoostM1, Attribute Selected, Classification via Regression, Filtered Classifier, Multiclass Classifier and J48,They have inferred that LMT Classifier gives more accurate diagnosis i.e. 76 % healthy and 24 % sick patients. David et al. [36] had used classification techniques for leukemia disease prediction. K- Nearest Neighbor, Bayesian Network, Random tree, J48 tree compared on the basis of accuracy, learning time and error rate. According to them Bayesian algorithm has better classification accuracy amongst others. Vijayarani et al. [37] had compared the analysis of classification function techniques for heart disease prediction. Classification was done using algorithms such as Logistic, Multilayer Perception and Sequential Minimal Optimization algorithms for predicting heart disease. In this classification comparison logistic algorithm trained out to be best classifier for heart disease having more accuracy and least error rate. Durairaj M. and Ranjani V. [38] had compared different data mining applications in healthcare sector. Algorithms such as Naïve, J48, KNN and C4.5 were used for Classification in order to diagnose diseases like Heart Disease, Cancer, AIDS, Brain Cancer, Diabetes, Kidney Dialysis, Dengue, IVF and Hepatitis C. Comparison study analysis revealed high accuracy i.e. 97.77% for cancer prediction and around 70% for IVF treatment through data mining techniques. Sugandhi C. et al. [39] analyzed a population of cataract patient's database by weka tool. In this study, weka has been used to classify the results and for comparison purpose. They have concluded that Random Tree gives 84% classify accuracy which means better performance as compared to other algorithms used for classification accuracy performance of Naïve Bayes, SMO, J48, REP Tree and Random Tree. Thus according to their study Random Tree is the best performance classification algorithm for cataract patient disease. Yasodha P. and Kannan M. [40] performed analysis of a population of diabetic patient database using weka tool. They have classified the data and then outputs were compared by using Bayes Network, REP Tree, J48 and Random Tree algorithms. Finally the results conclude that these algorithms help to determine and identify the stage or state in which a of disease like diabetes is in by entering patients daily glucose rate and insulin dosages thereby predicting and consulting the patients for their next insulin dosage .

In [41], modify Levenberg-Marquardt algorithm for MLP neural network learning. The modified algorithm has good convergence. This method reduces the amount of

oscillation in learning procedure. In [42], the intelligence based on artificial neural networks is incorporated into cross platform application development by using an Artificial Neural Network software component. An ANN model is a software based model that can be used in predicting certain parameters based on the past data provided. The model learns through the past data and produces a forecast for the future. It has many advantages over the other types of computing models and statistical models, its ability to learn from the data itself, being one remarkable advantage [43].

The accuracy of dengue disease prediction is not good enough using the methods which were used in these previous works. In previous work, they can work with Average Temperature, Average Relative Humidity, Rainy Days per Week, Total Rainfall and the Previous Cases are identified with a time parameters in Sri-Lanka, Singapore, Malaysia, Paraguay, Thailand etc. countries. There is no work done yet to predict dengue disease prediction with so many attributes of real data using ANN in Bangladesh. In Bangladesh, there is no more efficient work for dengue prediction before. There are some works where some specific parameters are used i.e. Average monthly humidity, rainfall, minimum and maximum temperature and number of dengue cases reported and prediction accuracy is low.

In this thesis work, 25 parameters are used to predict dengue outbreaks in the perspective of Bangladesh. Attributes of dengue disease are Age, Sex, Month, Residence, No. of days(fever), Current temperature, White Blood Cell(WBC), Back pain of eyes, Joint pain, Muscle pain, Headache, Vomiting, Hemoglobin, Hematocrit, Platelets, Shock, Rash, Pleural effusion, Ascites, Bleeding, Slow heart rate, Igg, Igm, NS1, Travelling story etc.

The summary of past works is described in short in the table 2.1 which is shown in below:

Table 2.1: Summary of literature

Sl No.	**Author, Year**	**Implemented Methods**	**Remark**

[1]	P.H.M.Nishanthi Herath, A.A.I. Perera and H.P.Wijekoon (2014)	Multi - Layer Perceptr on	• Average Temperature, Average Relative Humidity, Rainy Days per Week, Total Rainfall and the Previous Cases are identified with a time • Kandy, Sri Lanka • Accuracy of 68.5% according to Pred (25). • Explore attributes
[2]	Hani M. Aburas, B. Gultekin Cetiner and Murat Sari (2010)	ANN model	• Mean temperature, mean relative humidity, total rainfall and the total number of dengue confirmed-cases • Singaporean National Environment Agency (NEA) • Correlation to 0.86 from 0.91. • Explore attributes
[3]	B. Gultekin Cetiner , Murat Sari and Hani M. Aburas (2009)	ANN model	• Mean temperature, mean relative humidity, total rainfall and number of dengue confirmed cases • Singaporean National Environment Agency (NEA) • correlation to 0.70 from 0.76
[4]	Md. Nazmul Karim, Saif Ullah Munshi, Nazneen Anwar & Md. Shah Alam (2012)	multiple regressio n	• Average monthly humidity, rainfall, minimum and maximum temperature and number of dengue cases reported • Bangladesh Meteorological Department at Dhaka. • Area under ROC = 0.89, 95% CI = 0.89-0.98 • More information will be added for the predication
[5]	Yuhanis Yusof and Zuriani Mustaffa (2011)	Least Squares Support Vector Machine s	• Dengue fever (DF) cases, neighborhood dengue cases and total of Rainfall. • From five districts in Selangor. • Approximately 86.84%. • Explore attributes
[6]	Aloka. Munasinghe, H.L. Premaratne, and M.G.N.A.S. Fernando (2013)	regressio n model, ANN and GIS model	• Minimum, maximum, and average values of rainfall, temperature, and relative humidity, Conventional indices data and dengue cases data and population density • Metrology Department, Colombo and the Faculty of Agriculture, University of Ruhuna, Sri Lanka • Accuracy 86.5% in Matara district and 89.5% in Colombo district • Explore attributes
[7]	K. Balasaravanan and M. Prakash (2018)	ANN	• Rainfall, humidity rate, temperature rate, number of dengue cases • accuracy 92%

[8]	V.Ughelli, Y.Lisnichuk, J. Paciellol and J. Pane (2017)	ANN	• Rainfall, humidity rate, temperature rate, number of dengue cases • DINAC, Health Surveillance Office of Paraguay • Explore attributes
[9]	Jorge D. Mello-Roman ,Julio C. Mello-Roman, Santiago Gomez-Guerrero, and Miguel Garcia-Torres (2019)	ANN multilayer perceptron	• Week, Year, Age, Sex, District, Hospitalised, Headache, Myalgia, Arthralgia ,Retro-ocular pain, Pruritus, Sickness, Cough and dyspnea, Oligoanuria, Epistaxis, Gingivorrhagia, Hemoptysis, Melena, Black vomiting , Exanthema, Shock, Conjunctival injection, Bi-palpebral edema etc. • 96% accuracy, 96% sensitivity, and 97% specificity
[11]	Padet Siriyasatien, Atchara Phumee, Phatsavee Ongruk, Katechan Jampachaisriand Kraisak Kesorn (2016)	Multivariate poisson regression	• Seasonal temperature, rainfall, humidity, and wind, Ae. aegypti larvae infection rate, Female mosquito infection rate, Male mosquito infection rate, Season, Population and Dengue cases • Thai Meteorological department, Parasitology Department, Chulalongkorn University and NTCAESI • p-value = 0.9639
[12]	Shameem Fathima, Nisar Hundewale (2011)	SVM with Random Forest	• Patient demographic details, admission reasons, discharge details, lab tests outcome, and complaints on reporting, course of treatment • Clinics of Chennai and Tirunelveli from India • 0.9078
[13]	Kashish Ara Shakil, Shadma Anis And Mansaf Alam (2015)	Naive Bayes and J48	• Fever Temperature, WBC, Platelets, Severe Headache, Vomiting, Metallic Taste, Joint Pain, Appetite, Diarrhea, Hematocrit,Hemoglobin, and days. • Naive Bayes is 100% accurate and J48 is 99.70% accurate
[14]	Naiyar Iqbal and Mohammad Islam (2017)	Decision Tree, Naive Bayes, Support Vector Machines, Neural Networks	• Rainfall, Humidity, Temperature, Wind Velocity, Fever Temperature, WBC, Platelets, Severe Headache, Vomiting, Metallic Taste, Joint Pain, Appetite, Diarrhea, Hematocrit, Hemoglobin, Rash, Ascites, Depression, Itching, Shock, Residence, Abdominal Pain etc.
[16]	Fatimah Ibrahima, Mohd	Multilayer feed-	• Fever, headache, retro-orbital pain, arthralgia, myalgia, rash, taste aberration,

	Nasir Taib, Wan Abu Bakar Wan Abas, Chan Chong Guan, Saadiah Sulaiman (2005)	forward neural networks (MFNN).	vomiting, blood stained saliva, ecchymoses • National University of Malaysia Hospital (HUKM), and Hospital Kuala Lumpur in Malaysia • 90% prediction accuracy. • More information will be added for the predication
[17]	Nor Azura Husin, Naomie Salim, Ab Rahman Ahmad (2008)	NNM	• Dengue cases data (independent variable), rainfall data (independent variable) and approximate location dengue cases data (independent variable) • State Health Department of Selangor (SHD) and Malaysian Meteorological Service • 0.9 of learning rate, 0.9 of momentum rate.
[18]	Napa Rachata, Phasit Charoenkwan, Thongchai Yooyativong, Kosin Chamnongthai, Chidchanok Lursinsap , and Kohji Higuchi (2008)	Artificial neural network.	• Maximum, minimum, average temperature, rainfall, relative humidity and Dengue Haemorrhagic Fever cases data • Bureau of Epidemiology Department of Disease Control Ministry of Public Health and Thai Meteorological Department • 85.92% accuracy.
[20]	M. Naresh Kumar (2013)	Alternating Decision tree	• Fever duration, vomiting, body pains, rashes, pulse, headache, restlessness, abdominal pain and hemoglobin, WBC, platelet, packed cell volume, IgM and IgG • Accuracy of 89%.
[21]	Daranee Thitiprayoonwongse, Prapat Suriyaphol And Nuanwan Soonthornphisaj (2012)	Decision Tree	• Shock, leakage, platelet_min, liver_size_avg, JE vaccin • Srinagarindra Hospital, Khon Kaen, Thailand (KK) • Accuracy rate 97.6%.
[23]	M.Bhavani and S.Vinod kumar (2016)	SVM, J48 Classification	• Fever, bleeding, metallic taste, Fatigue • Accuracy of 84 % • Explore attributes.
[26]	T.Sajana, M.Navya, YVSSV.Gayathri, N.Reshma (2018)	CART, Multi-layer perception and	• WBC, Temperature, Pain,Abdominal pain, Vomiting, Diarrhoea, Severe headache, Haemoglobin, Platelets, Dengue, Metallic Taste, Joint/Muscle pain • Vijayawada hospital

		C4.5 algorithms	• Simple CART algorithm shows 100% accuracy
[27]	Kamran Shaukat, Nayyer Masood, Sundas Mehreen and Ulya Azmeen (2015)	Naive Bayesian, REP Tree, Random tree, J48 and SMO.	• Fever, Bleeding, Myalgia, Flu, Fatigue. • District Headquarter Hospital Jhelum. • Accuracy of Naive bayes is 92% • More information will be added for the predication

2.3 SUMMARY

The analysis of literature review had broadened the scope of dengue disease prediction issues in the perspective of Bangladesh. The information and findings collected from this chapter is used as a guidance to develop the proposed systems. The methods used in previous studies did not have sufficient precision in predicting dengue disease. They have previously worked with Average Temperature, Average Relative Humidity, Rainy Days per Week, Total Rainfall, and Previous Cases are found with time parameters in countries such as Sri Lanka, Singapore, Malaysia, Paraguay, and Thailand. In Bangladesh, no work has been done to predict dengue disease using ANN with so many attributes of real data. There has never been a more effective job for dengue prediction in Bangladesh. Some works employ precise metrics, such as average monthly humidity, rainfall, and minimum and maximum temperatures and their prediction accuracy is low.

Twenty-five parameters are used in this thesis work to forecast dengue outbreaks in Bangladesh. Age, sex, month, residence, number of days (fever), current temperature, white blood cell(WBC), Back eyes, Joint pain, Headaches, Vomiting, Hemoglobin, Hematocrit, Pleural effusion, Shock, Rash, Slurred Heart, Bleeding, Slow heart rate, Igg, Igm, NS1, Story of journey and so forth. This chapter has also demonstrated the necessities of an efficient technique which is able to predict dengue disease and find the effective factors.

CHAPTER 3

METHODOLOGY

3.1 INTRODUCTION

This chapter presents an outline of research methods that were followed in the study. It provides information on the workflow diagram, data collection and preprocessing, artificial neural network modeling, decision tree algorithms and performance analysis terminologies used in the study. The requirements of this work and the procedures that were followed to carry out this study are included.

3.2: STRATEGY OF WORK FRAME

First of all, attributes are selected for collecting data. After collecting data, data is preprocessing and normalizing for training algorithm. Then normalized data is used in training, testing and validation. If the output of train, test and validation is acceptable after comparing among all other training algorithms of ANN and decision tree, then final model is selected for predicting disease. The workflow of diagram of this thesis work is shown in figure 3.1.

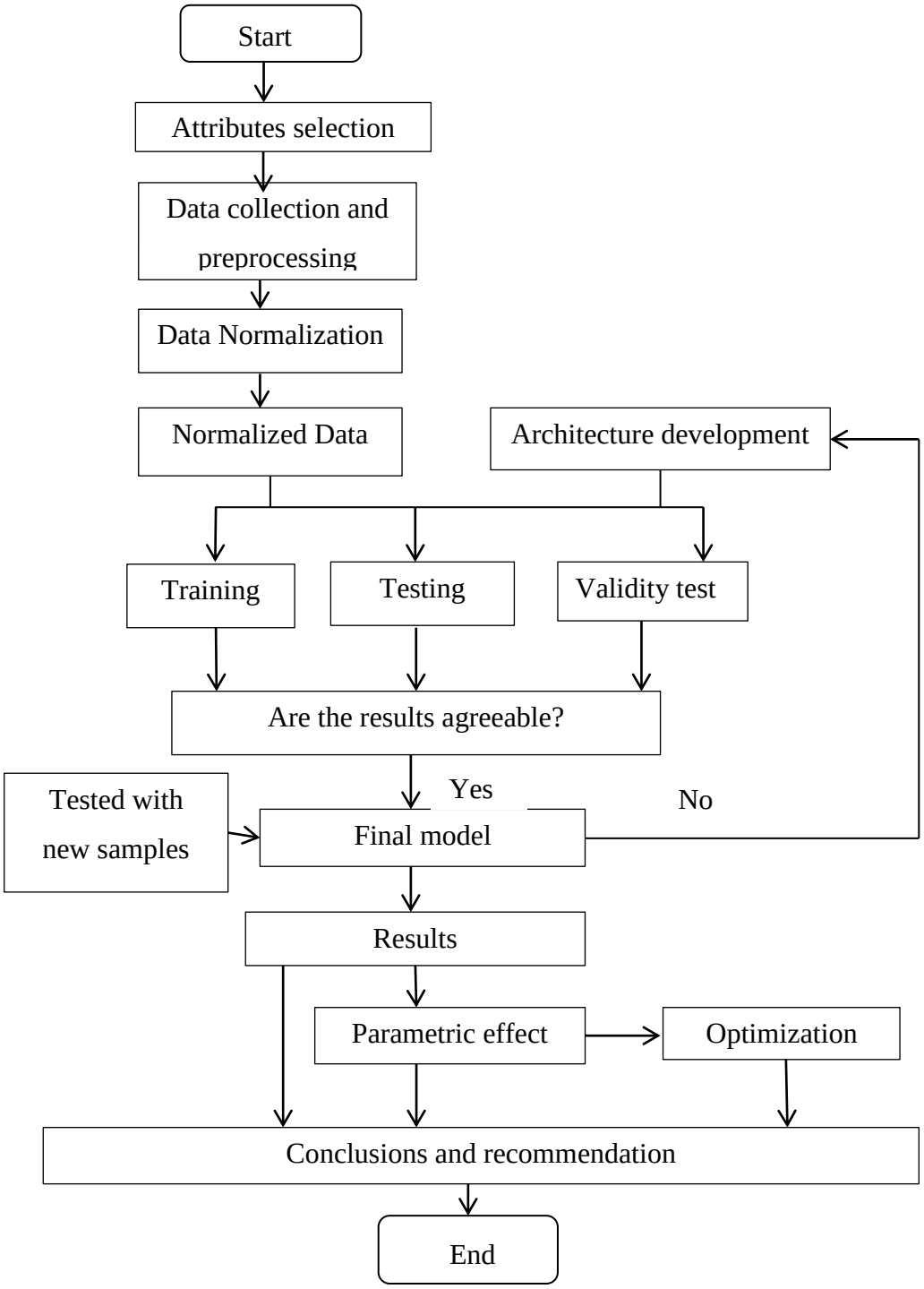

Start
Attributes selection
Data collection and preprocessing
Data Normalization
Normalized Data
Architecture development
Training
Testing
Validity test
Are the results agreeable?
Yes
No
Tested with new samples
Final model
Results
Parametric effect
Optimization
Conclusions and recommendation
End

Figure 3.1: Workflow diagram

3.3 DATA COLLECTION AND DATA PREPROCESSING

3.3.1 Attributes Selection

For the proper diagnosis, it is necessary to evaluate some of the main attributes of dengue patient's dataset. Accurate predict of the dengue disease is a challenging task for doctors. Different classification techniques are used to classify the data and predict the dengue disease by using the datasets of dengue patients. Attributes of dengue disease are Age, Sex, Month, Residence, No. of days(fever), Current temperature, Average Temperature, Average Relative Humidity, Rainy Days per Week, Total Rainfall, , Myalgia, Arthralgia ,Retro-ocular pain , Pruritus , Sickness, Cough and dyspnea, Oligoanuria , Epistaxis, Gingivorrhagia, Hemoptysis , Melena, Black vomiting , Exanthema, Ae. aegypti larvae infection rate, Female mosquito infection rate, Male mosquito infection rate, White Blood Cell(WBC), Back pain of eyes, Joint pain, Muscle pain, Headache, Vomiting, Hemoglobin, Hematocrit, Platelets, Shock, Rash, Pleural effusion, Ascites, Bleeding, Slow heart rate, Igg, Igm, NS1, Travelling story etc. From these attributes 25 parameters are selected for this thesis work. These attributes are Age, Sex, Month, Residence, No. of days(fever), Current temperature, White Blood Cell(WBC), Back pain of eyes, Joint pain, Muscle pain, Headache, Vomiting, Hemoglobin, Hematocrit, Platelets, Shock, Rash, Pleural effusion, Ascites, Bleeding, Slow heart rate, Igg, Igm, NS1, Travelling story etc. Description of these attributes is shown in table 3.1.

Table 3.1: Attributes list and data description

Attributes	Description
Age	Patient's age
Sex	Gender(male or female)
Month	Affected Month
Residence	Address
No. of days(fever)	Number of days of fever
Current temperature	Fever degree in Celsius
White Blood Cell(WBC)	Number of White Blood Count
Back pain of eyes	Low/Medium/High
Joint pain	Low/Medium/High
Muscle pain	Low/Medium/High
Headache	Low/Medium/High
Vomiting	Low/Medium/High

Table 3.1: (Continued)

Attributes	Description
Hemoglobin	Range of Hemoglobin

Hematocrit	Range of Hematocrit
Platelets	Number of Platelets
Shock	Fall in Blood Pressure (Low/Medium/High)
Rash	Low/Medium/High
Pleural effusion	Excess fluid(Low/Medium/High)
Ascites	Abdominal Swelling(Low/Medium/High)
Bleeding	Low/Medium/High
Slow heart rate	Rate of heart beats(Low/Medium/High)
Igg	Immunoglobulin G
Igm	Immunoglobulin M
NS1	Nonstructural Protein 1
Travelling story	Yes/No

3.3.2 Data Collection

The data used in this study to predict sourced mainly from two locations. The data is collected from peoples who tested for dengue disease from different hospitals of Noakhali and Dhaka. Data is collected from people who are tested for dengue by dataset paper. Data is collected from Good Heal Complex, Good Heal Hospital, Noakhali sadar Hospital, Prime Hospital, Holy land Hospital, Medinova Hospital and Dhaka Community Hospital etc. The total number of data points used in the study is 206, after removing the anomalies. 70% of data is used for training, 15% testing and 15% validating. And 20 unused samples are used for cross check network accuracy. Raw Dataset is shown in appendix A-1.

3.3.3. Data Preprocessing

Before introducing the collected data into the classification algorithm for training, it should be noted that there are differences in the range and measure units of the variables. Using this data directly for the training of a classification algorithm could cause a considerably delay in convergence. A solution to this problem is to preprocess the training and validation data before using them. First of all, after taking all collected raw data samples are included in the excel file and save file with ‘csv’ extension. There are several parameters whose value are a string and first, converted them into numeric via python. The developed code of data preprocessing is given below:

Code

```
import numpy as py
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns
dataset = pd.read_csv("Thesisdata.csv")
x=dataset.iloc[:,:-1].values
y=dataset.iloc[:,25].values
#missing value "NaN" show korle
from sklearn.preprocessing import Imputer
imputer=Imputer(missing_values="NaN",strategy='most_frequent',    fill_value='None')
imputer=imputer.fit(x[:,1:25])
x[:,1:25]= imputer.transform(x[:,1:25])
#preprocessing string variable as numeric
from sklearn.preprocessing import LabelEncoder, OneHotEncoder
labelencoder_x2=LabelEncoder()
x[:,1]=labelencoder_x2.fit_transform(x[:,1])
x[:,2]=labelencoder_x2.fit_transform(x[:,2])
x[:,3]=labelencoder_x2.fit_transform(x[:,3])
x[:,7]=labelencoder_x2.fit_transform(x[:,7])
x[:,8]=labelencoder_x2.fit_transform(x[:,8])
x[:,9]=labelencoder_x2.fit_transform(x[:,9])
x[:,10]=labelencoder_x2.fit_transform(x[:,10])
x[:,11]=labelencoder_x2.fit_transform(x[:,11])
x[:,15]=labelencoder_x2.fit_transform(x[:,15])
x[:,16]=labelencoder_x2.fit_transform(x[:,16])
x[:,17]=labelencoder_x2.fit_transform(x[:,17])
x[:,18]=labelencoder_x2.fit_transform(x[:,18])
x[:,19]=labelencoder_x2.fit_transform(x[:,19])
x[:,20]=labelencoder_x2.fit_transform(x[:,20])
x[:,21]=labelencoder_x2.fit_transform(x[:,21])
x[:,22]=labelencoder_x2.fit_transform(x[:,22])
x[:,23]=labelencoder_x2.fit_transform(x[:,23])
x[:,24]=labelencoder_x2.fit_transform(x[:,24])
```

Preprocess Dataset is shown in appendix A-2.

3.3.4 Data Normalization

Many algorithms attempt to find trends in the data by comparing features of data points. However, there is an issue when the features are on drastically different scales**.** For this data normalization is needed for best performance of classification algorithm. We use Min-Max normalization formula for normalization. Min-max normalization is one of the most common ways to normalize data. For every feature, the minimum value of that feature gets transformed into a 0, the maximum value gets transformed into a 1, and every other value gets transformed into a decimal between 0 and 1. Here is the data normalization formula

$$Xnormalized = \frac{x - \min(x)}{\max(x) - \min(x)} \quad (3.1)$$

Where $Xnormalized$ is the new normalized value.

All supervised learning methods start with an input data matrix, called X here. Each tuple represents one observation. Each column represents one variable. For classification, Y can be any data types, here used numeric type- 0 or 1 for the response respectively two class 'Yes' or 'No'. Normalized dataset is shown in appendix A-3.

3.4 DATA ANALYSIS

This thesis is showed what kind of parameters affect the occurrence of dengue. It is important to find out the symptoms for patients with dengue. For example, how much men or women are affected, urban or rural people who are more affected etc. is find out. This analysis is done by Microsoft office excel sheet.

3.5. ARTIFICIAL NEURAL NETWORK MODELING

The ANN used to predict Dengue cases is designed as a Multi-Layer Perceptron (MLP) network with three initial layers, namely – Input layer, Output layer and one hidden layer. The input layer consists of the input parameters identified in the previous

step. The hidden layer is designed with five neurons and to be increased as the training process progresses on demand. Output layer consists of a two parameter which is the prediction of number of Dengue cases or not.

In this work, different attributes such as Age, Sex, Month, Residence, No. of days(fever), Current temperature, White Blood Cell(WBC), Back pain of eyes, Joint pain, Muscle pain, Headache, Vomiting, Hemoglobin, Hematocrit, Platelets, Shock, Rash, Pleural effusion, Ascites, Bleeding, Slow heart rate, Igg, Igm, NS1, Travelling story etc. work as an input layer. The hidden layer is designed with ten neurons and to be increased as the training process progresses on demand. The structure of MLP of this work is shown in figure 3.2.

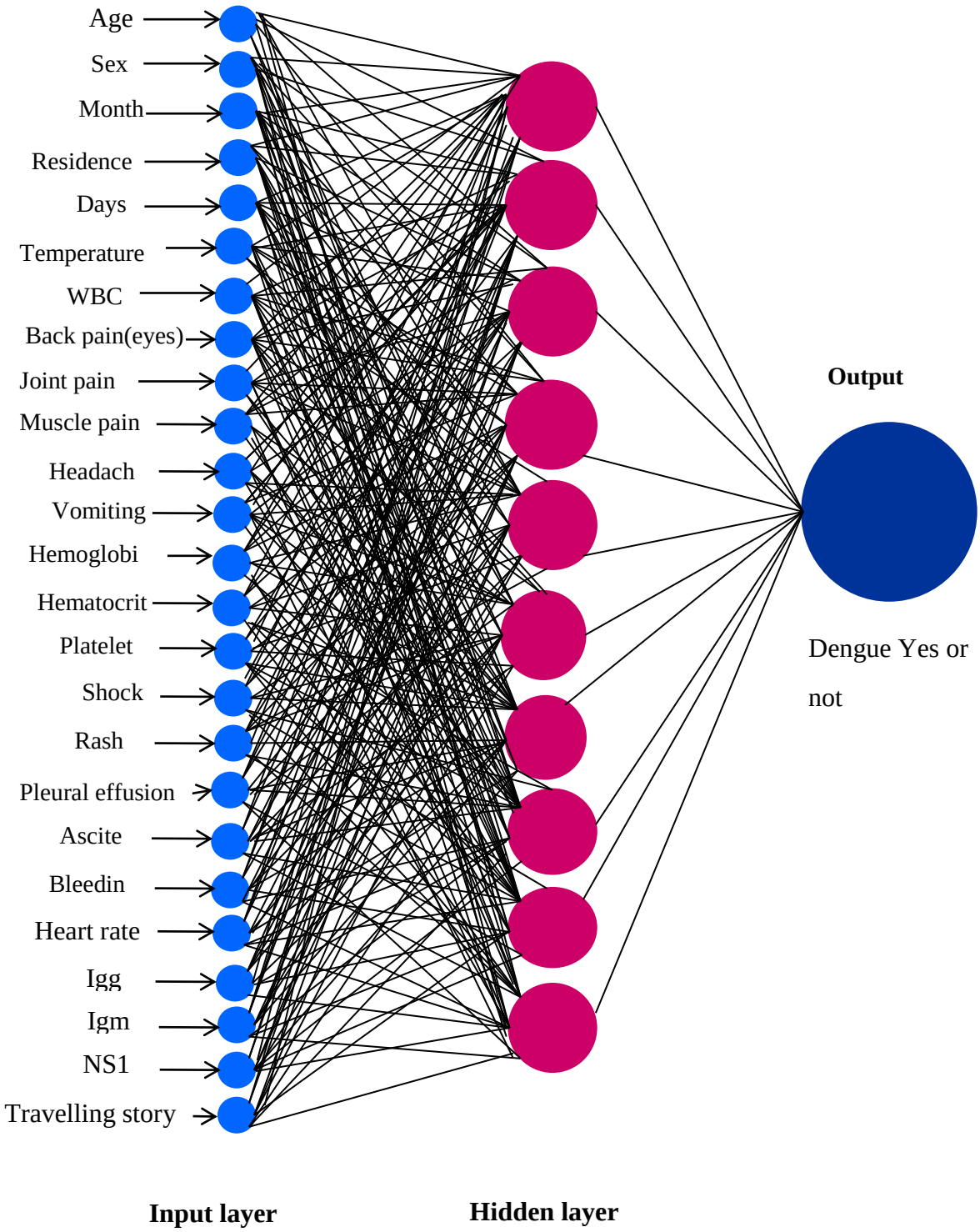


Figure 3.2: MLP structure for dengue disease prediction

3.5.1 Division of Dataset and Configuration of Distinct Algorithm

The complete data set is divided into three portions, training set, validation set and the test set as shown in Table 3.2. The training set, which is used to train the ANN, consist of 70% of the data set. Training process is carried out using the training set, alternatively using Log Sigmoid and Tan Sigmoid functions as the activation function for the hidden layer. For each activation function, the number of hidden layer neurons is increased from 1 neuron up to 10 neurons and trained with re-initialized weights.

Table 3.2: Proportions of data in each dataset

Data set	Proportion
Training set	70%
Validation set	15%
Test set	15%

15% of unused data from the entire data set is used for testing and validating the ANN, respectively. The validation set is particularly used to prevent over fitting the data, and thus using local minima in generating the result. The test set is the data which generates the real prediction.

In this work, different multi-layer perceptron neural networks algorithm is used. These are Lavenberg-Marquardt (LM), Scaled Conjugate Gradient (SCG) Backpropagation algorithms and Learning Vector Quantization (LVQ) algorithm. The ANNs with minimum value of Mean Square Error (MSE) for the results generated for both training and test sets were extracted to select the ANN most suitable and fit for the purpose. LM, SCG and LVQ algorithms are used to train the network and the corresponding configuration parameters are as shown in Table 3.3

Table 3.3: Training algorithms configuration parameters

Configuration Parameters	SCG	LM	LVQ
Maximum number of epochs to train	5000	5000	1000
Performance goal	0	0	0
Maximum validation failures	6	6	6
Minimum performance gradient	1e-7	1e-7	1e-7
Initial μ	0.001	0.001	0.001
μ decrease factor	0.1	0.1	0.1
μ increase factor	10	10	10
Maximum μ	1e10	1e10	1e10

3.5.2 Scaled Conjugate Gradient

In this work, neural network pattern classification technique is proposed for the prediction. A feed forward neural network is trained using different training functions that update weight and bias values. In this work training algorithm such as scaled conjugate gradient are studied for the implementation of prediction techniques.

Scaled conjugate gradient (SCG) is a supervised learning algorithm used for feed forward neural networks (NN) and the member of the class of conjugate gradient methods. SCG uses similar concepts of the general optimization strategy, but chooses the search direction and step size more effectively by using information from the second order approximation is represented by equation 3.2

$$E\ w + y \approx E\ w + E'\ (w)\ T + 1/2\ yTE''(w)y \tag{3.2}$$

In SCG, each iteration computes optimal distance. The line search is then performed to determine the optimal distance to move along the current search direction as equation 3.3.

$$wk+1 = wk + ak * pk \tag{3.3}$$

Then the next search direction is performed so that, the conjugated to previous search instructions. Actually pk is a function of ak, the Hessian matrix of the error function and also the matrix of the second derivatives. SCG uses a scalar ak that is supposed to regulate the indefiniteness of the Hessian matrix [44].

Development code of scaled conjugate is given below:

Code

```
% patternnet= Pattern recognition network
load dengu_dataset.mat
x = dengu_inputs;
t = dengu_targets;
% Syntax : patternnet(hiddenSizes,trainFcn,performFcn)
hiddenSizes = 10; % default setting
trainFcn='trainscg'; % default setting % Scaled conjugate gradient backpropagation
performFcn='crossentropy'; % default setting
net=patternnet(hiddenSizes,trainFcn,performFcn);
% Or net=patternnet(10,'trainscg','crossentropy'); % default setting
 % Or net=patternnet(hiddenLayerSize); Or net = patternnet(10);
view(net)
net.trainParam.epochs = 1000;% Maximum number of epochs to train
net.trainParam.goal=0; % Performance goal
net.trainParam.max_fail=6; % Maximum validation failures
net.trainParam.min_grad=1e-7; % Minimum performance gradient
net.trainParam.mu=0.001; % Initial mu
net.trainParam.mu_dec=0.1; % mu decrease factor
net.trainParam.mu_inc=10; % mu increase factor
net.trainParam.show=25; %Epochs between displays (NaN for no displays)
net.trainParam.showCommandLine=false; %Generate command-line output
net.trainParam.time=inf; %Maximum time to train in seconds
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
[net,tr] = train(net,x,t); % Or net=train(net,x,t);
view(net)
y = net(x);
errors = gsubtract(t,y);
perf = perform(net,t,y); % classes = vec2ind(y);
```

```
% To draw Target versus Output curve or plot
figure
plot(t,y, 'LineWidth',2)
title('Figure 1: Target(t) versus Network Output(y)');
xlabel('Target');
ylabel('Output');
xlim([0 1]);
ylim([0 1]);
lgd = legend({'data','Fit'},'FontSize',12,'TextColor','black','Location','NorthEastoutside')
% To evaluate the network with about few (here 20 samples) new samples
xi=inputs_for_final_evaluation;
yi = net(xi);
ti=target_for_final_evaluation;
errors = gsubtract(yi, ti);
% It is also possible to calculate the network performance only on the  test set, by using the testing indices, which are located in the  training record.
tInd = tr.testInd;
tstOutputs = net(x(:,tInd));
tstPerform = perform(net,t(:,tInd),tstOutputs)
% Plot the training, validation, and test performance.
figure, plotconfusion(t,y)
figure, ploterrhist(errors)
figure, plotregression(t,y,'Regression')
figure, plotroc(t,y)
figure, plotperform(tr)
figure, plottrainstate(tr)
perf2 = perform(net,ti,yi);
```

3.5.3 Levenberg Marquardt Algorithm

Levenberg – Marquardt algorithm is specifically designed to minimize sum-of-square error functions [45] of the form.

$$E = \frac{1}{2}\varepsilon\, k\, (\, e_k)^2 = \frac{1}{2}\, ||e||^2 \qquad (3.4)$$

Where e_k *is* the error in the kth exemplar or pattern and e is a vector with element e_k. If the difference between the pervious weight vector and the new weight vector is small, the error vector can be expanded to first order by means of a Taylor series.

$$e_{(j+i)} = e_{(j)} + \frac{\partial e_k}{\partial w_i}\left(w_{(j+i)} - w_{(j)}\right) \tag{3.5}$$

As a consequence, the error function can be expressed as

$$E = \frac{1}{2}||e_{(j)} + \frac{\partial e_k}{\partial w_i}\left(w_{(j+i)} - w_{(j)}\right)||^2 \tag{3.6}$$

Minimizing the error function with respect to the new weight vector, gives

$$w_{(j+1)} = w_{(j)} - (Z^T Z)^{-1} Z^T e_{(j)} \tag{3.7}$$

Where, $(Z)_{ki} = \frac{\partial e_k}{\partial w_i}$

Since the Hessian for the sum-of-square error function is

$$(H)_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j} = \varepsilon\{\left(\frac{\partial e_k}{\partial w_i}\right)\left(\frac{\partial e_k}{\partial w_i}\right) + e_k \frac{\partial^2 e_k}{\partial w_i \partial w_j} \tag{3.8}$$

Neglecting the second term, the Hessian can be written as $H = Z^T$

Updating of the weights therefore involves the inverse Hessian or an approximation thereof for nonlinear networks. The Hessian is relatively easy to compute, since it is based on first order derivatives with respect to the network weights that are easily accommodated by back propagation. Although the updating formula could be applied iteratively to minimize the error function, this may result in a large step size, which would invalidated the linear approximation on which the formula is based.

In the Levenberg-Marquardt algorithm, the error function is minimized, while the step size is kept small in order to ensure the validity of the linear approximation. This is accomplished by use of a modified error function of the form.

$$E = \frac{1}{2}||e_{(j)} + \frac{\partial e_k}{\partial w_i}\left(w_{(j+i)} - w_{(j)}\right)||^2 + \lambda||\left(w_{(j+i)} - w_{(j)}\right)||^2 \tag{3.9}$$

Where is a parameter governing the step size. Minimizing the modified error with respect to w(j+1) gives

$$w_{(j+1)} = w_{(j)} - (Z^T Z + \lambda I)^{-1} Z^T e_{(j)} \tag{3.10}$$

Development code of scaled conjugate is given below:

Code

```
% patternnet= Pattern recognition network
load dengu_dataset.mat
x = dengu_inputs;
t = dengu_targets;
% Syntax : patternnet(hiddenSizes,trainFcn,performFcn)
hiddenSizes = 10; % default setting
trainFcn='trainlm'; % default setting % Scaled conjugate gradient backpropagation
performFcn='crossentropy'; % default setting
net=patternnet(hiddenSizes,trainFcn,performFcn);
% Or net=patternnet(10,'trainlm','crossentropy'); % default setting
 % Or net=patternnet(hiddenLayerSize); Or net = patternnet(10);
view(net)
net.trainParam.epochs = 1000;          % Maximum number of epochs to train
net.trainParam.goal=0; % Performance goal
net.trainParam.max_fail=6; % Maximum validation failures
net.trainParam.min_grad=1e-7; % Minimum performance gradient
net.trainParam.mu=0.001; % Initial mu
net.trainParam.mu_dec=0.1; % mu decrease factor
net.trainParam.mu_inc=10; % mu increase factor
net.trainParam.show=25; %Epochs between displays (NaN for no displays)
net.trainParam.showCommandLine=false; %Generate command-line output
net.trainParam.time=inf; %Maximum time to train in seconds
net.divideParam.trainRatio = 70/100;
net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
[net,tr] = train(net,x,t); % Or net=train(net,x,t);
view(net)
y = net(x);
errors = gsubtract(t,y);
perf = perform(net,t,y); % classes = vec2ind(y);
```

```
% To draw Target versus Output curve or plot
figure
plot(t,y, 'LineWidth',2)
title('Figure 1: Target(t) versus Network Output(y)');
xlabel('Target');
ylabel('Output');
xlim([0 1]);
ylim([0 1]);
lgd = legend({'data','Fit'},'FontSize',12,'TextColor','black','Location','NorthEastoutside')
% To evaluate the network with about few (here 20 samples) new samples
xi=inputs_for_final_evaluation;
yi = net(xi);
ti=target_for_final_evaluation;
errors = gsubtract(yi, ti);
% It is also possible to calculate the network performance only on the  test set, by using the testing indices, which are located in the  training record.
tInd = tr.testInd;
tstOutputs = net(x(:,tInd));
tstPerform = perform(net,t(:,tInd),tstOutputs)
% Plot the training, validation, and test performance.
figure, plotconfusion(t,y)
figure, ploterrhist(errors)
figure, plotregression(t,y,'Regression')
figure, plotroc(t,y)
figure, plotperform(tr)
figure, plottrainstate(tr)
perf2 = perform(net,ti,yi);
```

3.5.4 Learning Vector Quantization

Learning Vector Quantization LVQ, different from Vector quantization VQ and Kohonen Self-Organizing Maps KSOM, basically is a competitive network which uses supervised learning. We may define it as a process of classifying the patterns where

each output unit represents a class. As it uses supervised learning, the network will be given a set of training patterns with known classification along with an initial distribution of the output class. After completing the training process, LVQ will classify an input vector by assigning it to the same class as that of the output unit.

Following figure 3.3 shows the architecture of LVQ which is quite similar to the architecture of KSOM. As we can see, there are "n" number of input units and "m" number of output units. The layers are fully interconnected with having weights on them.

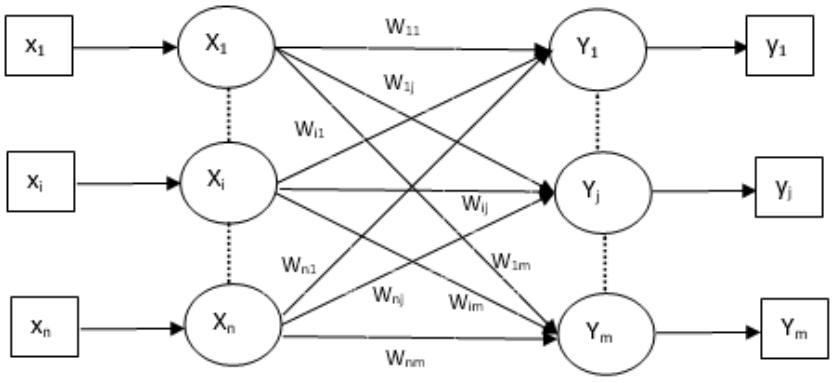


Figure 3.3: Architecture of Learning Vector Quantization

Algorithm of Learning Vector Quantization is given below:

Step 1 − Initialize reference vectors, which can be done as follows −

Step 1a − From the given set of training vectors, take the first "m" numberofclusters training vectors and use them as weight vectors. The remaining vectors can be used for training.

Step 1b − Assign the initial weight and classification randomly.

Step 1c − Apply K-means clustering method.

Step 2 − Initialize reference vector α

Step 3 − Continue with steps 4-9, if the condition for stopping this algorithm is not met.

Step 4 − Follow steps 5-6 for every training input vector x.

Step 5 − Calculate Square of Euclidean Distance for j = 1 to m and i = 1 to n

$$D(j)=\sum i=1n\sum j=1m(xi-wij)2$$

Step 6 − Obtain the winning unit J where Dj is minimum.

Step 7 − Calculate the new weight of the winning unit by the following relation −

if T = Cj then wj(new)=wj(old)+α[x−wj(old)]

if T ≠ Cj then wj(new)=wj(old)−α[x−wj(old)]

Step 8 − Reduce the learning rate α.

Step 9 − Test for the stopping condition. It may be as follows −

Maximum number of epochs reached.

Learning rate reduced to a negligible value.

Development code of Learning Vector Quantization is given below:

Code

```
% lvqnet= Learning vector quantization neural network
load dengu_dataset.mat
x = dengu_inputs;
t = dengu_targets; % or [x,t] = dengu_dataset;
% Syntax: lvqnet(hiddenSize,lvqLR,lvqLF)
hiddenSize=10; % default settings
lvqLR=0.01; % default settings
lvqLF='learnlv1'; % default settings
      % or lvqLF='learnlv2';
 net.divideParam.trainRatio = 70/100;
 net.divideParam.valRatio = 15/100;
net.divideParam.testRatio = 15/100;
net=lvqnet(hiddenSize,lvqLR,lvqLF);
% Or net = lvqnet(10, 0.01, 'learnlv1'); % default settings
net.trainParam.epochs = 1000; % Maximum number of epochs to train
net.trainParam.goal=0; % Performance goal
net.trainParam.max_fail=6; % Maximum validation failures
net.trainParam.min_grad=1e-7; % Minimum performance gradient
net.trainParam.mu=0.001; % Initial mu
net.trainParam.mu_dec=0.1; % mu decrease factor
net.trainParam.mu_inc=10; % mu increase factor
%net.trainParam.mu_max=1e10; % Maximum mu
```

```
net.trainParam.show=25; %Epochs between displays (NaN for no displays)
net.trainParam.showCommandLine=false; %Generate command-line output
% net.trainParam.showWindow=true: % Show training GUI
net.trainParam.time=inf;
[net,tr] = train(net,x,t); % Or net = train(net,x,t);
view(net)
y = net(x);
perf = perform(net,t,y)% Or performance = perform(net,t,y);
errors = gsubtract(t,y);
% To draw target versus output curve or plot
figure
plot(t,y,'LineWidth',2, 'MarkerSize',10)
title('Figure 1: Target(t) versus Network Output(y)');
xlabel('target');
ylabel('Output');
xlim([0 1]);
ylim([0 1]);
lgd = legend({'data','Fit'},'FontSize',12,'TextColor','black','Location','NorthEastoutside')
 % Or figure, plotregression(t,y,'Regression')
```

% Test the network. After the network has been trained, you can use it to compute the network outputs. The following code calculates the network outputs, errors and overall performance.

```
xi=inputs_for_final_evaluation;
yi = net(xi);
ti=target_for_final_evaluation;
errors = gsubtract(yi, ti);
```

% It is also possible to calculate the network performance only on the test set, by using the testing indices, which are located in the training record.

```
tInd = tr.testInd;
tstOutputs = net(x(:,tInd));
tstPerform = perform(net,t(:,tInd),tstOutputs)
perf2 = perform(net,ti,yi)% Or performance = perform(net,t,y);
```

% Plot the training, validation, and test performance.

figure, plotconfusion(t,y)

figure, ploterrhist(errors)

figure, plotregression(t,y,'Regression')

figure, plotroc(t,y)

figure, plotperform(tr)

figure, plottrainstate(tr)

3.6 DECISION TREE ALGORITHM

Decision Tree is a supervised learning technique that can be used for both classification and regression problems, but mostly it is preferred for solving classification problems. It is a tree-structured classifier, where internal nodes represent the features of a dataset, branches represent the decision rules and each leaf node represents the outcome. In a decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. Decision trees classify instances by sorting them down the tree from the root to some leaf node, which provides the classification of the instance. An instance is classified by starting at the root node of the tree, testing the attribute specified by this node, then moving down the tree branch corresponding to the value of the attribute as shown in the figure 3.4. This process is then repeated for the sub tree rooted at the new node.

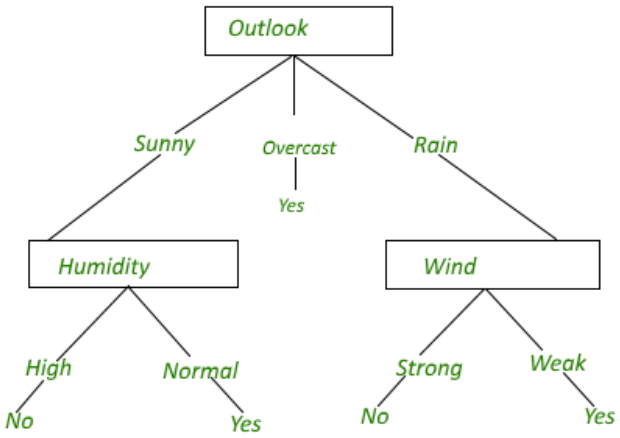


Figure 3.4: Example of decision tree for the concept of play tennis

The decision tree in above figure classifies a particular morning according to whether it is suitable for playing tennis and returning the classification associated with the particular leaf.(in this case Yes or No).

For example, the instance

(Outlook = Rain, Temperature = Hot, Humidity = High, Wind = Strong) would be sorted down the leftmost branch of this decision tree and would therefore be classified as a negative instance.

In other words we can say that decision tree represent a disjunction of conjunctions of constraints on the attribute values of instances.

(Outlook = Sunny ^ Humidity = Normal) v (Outllok = Overcast) v (Outlook = Rain ^ Wind = Weak)

Algorithm

Step-1: Begin the tree with the root node, says S, which contains the complete dataset.

Step-2: Find the best attribute in the dataset using Attribute Selection Measure (ASM).

Step-3: Divide the S into subsets that contains possible values for the best attributes.

Step-4: Generate the decision tree node, which contains the best attribute.

Step-5: Recursively make new decision trees using the subsets of the dataset created in step -3. Continue this process until a stage is reached where you cannot further classify the nodes and called the final node as a leaf node.

Developed code is shown below:

Code

```
% Decision Tree
load dengue_data_nor_all.mat
X=inputs;% Or
Y=diseases; % Or
ctree = fitctree(X, Y,'PredictorNames',{'Age' 'Sex' 'Month' 'Residence'..... 'Days' 'Temp' 'WBC' 'Back_pain_of_eyes' 'Joint_pain' 'Muscle_pain' 'Headache' 'Vomiting' 'Hemoglobin' 'Hematocrit' 'Platelets' 'Shock'..... 'Rash' 'Effusion' 'Ascites' 'Bleeding' 'Heart rate' 'Igg' 'Igm' 'NS1' 'Travelling'}); % Or
rng(1); % For reproducibility % Use when need CrossVal1 is reqired.
```

```
ctree = fitctree(X, Y,'PredictorNames',{'Age' 'Sex' 'Month' 'Residence'.....'Days' 'Temp' 'WBC' 'Vomiting' 'Back_pain_of_eyes' 'Joint_pain' 'Muscle_pain' 'Headache' 'Hemoglobin' 'Hematocrit' 'Platelets' 'Shock'..... 'Rash' 'Effusion' 'Ascites' 'Bleeding' 'Heartrate' 'Igg' 'Igm' 'NS1' 'Travelling'},..... 'MaxNumSplits',20,'MinLeafSize',1,'MinParentSize',12,'AlgorithmForCategorical','Pull Left',..... 'CrossVal','on', 'KFold',5); % KFold=number of fold for cross validation, % Or
ctree = fitctree(inputs, diseases,'PredictorNames',{'Age' 'Sex' 'Month' 'Residence'.....
 'Days' 'Temp' 'WBC' 'Vomiting' 'Hemoglobin' 'Hematocrit' 'Platelets' 'Shock'.....
 'Rash' 'Effusion' 'Ascites' 'Bleeding' 'Heart rate' 'Igg' 'Igm' 'NS1' 'Travelling'},.....
 'MaxNumSplits',7,'CrossVal','on'); % and so on, % Or
MdlDefault = fitctree(X,Y,'CrossVal','on');% Or
ctree = fitctree(X,Y) % Or tree = fitctree(X,Y) % Or tree = fitctree(TBL,Y) % Or
view(ctree) % Or view(tree) % text description % For without CrossVal
view(ctree,'mode','graph')
% Specify to grow each tree using a minimum leaf size in leafs.
leafs = logspace(1,2,10);
rng('default')
N = numel(leafs);
err = zeros(N,1);
for n=1:N
t = fitctree(X,Y,'CrossVal','On',...
'MinLeafSize',leafs(n));
err(n) = kfoldLoss(t);
end
xlabel('Min Leaf Size');
ylabel('cross-validated error');
plot(leafs,err);
% Compare the near-optimal tree with at least 40 observations per leaf with the default tree,
ctree = fitctree(inputs,diseases);
view(ctree,'Mode','Graph')
OptimalTree = fitctree(inputs,diseases,'MinLeafSize',40);
view(OptimalTree,'mode','graph')
```

```
% Draw a histogram of the number of imposed splits on the trees. Also, view one of the trees.
numBranches = @(x)sum(x.IsBranch); %When ctree = fitctree(X,Y,'CrossVal','on'); % Or MdlDefault = fitctree(X,Y,'CrossVal','on');
mdlDefaultNumSplits = cellfun(numBranches, MdlDefault.Trained);
figure;
histogram(mdlDefaultNumSplits)
view(MdlDefault.Trained{1},'Mode','graph') %or
% Compare the cross validation classification errors of the models.
classErrorDefault = kfoldLoss(MdlDefault)
resuberror = resubLoss(ctree)
% Plot Confusion Matrix
load dengu_rbutnotnor_dataset_for_all_classification % Or
ctree = fitctree(inputs, diseases);
outputs=predict(ctree,inputs);
figure; plotconfusion(Y,outputs); % Or plotconfusion(targets,outputs)
% Find the predicted classification of a point at the mean (average) of the ionosphere data.
Xnew=inputs_for_final_evaluation; % CMdl = fitctree(X,Y); % Ynew = predict(CMdl,X);
Ynew = predict(ctree,Xnew);
```

3.7 PERFORMANCE ANALYSIS FOR BOTH ARTIFICIAL NEURAL NETWORKS AND DECISION TREE

This paper carried out model development in order to assess the performance and usefulness of different pattern classification algorithms for predicting dengue disease. After performing different classification algorithm and value of different term for each classification algorithm are listed in a table to measure and explore the performance on the classification methods. Comparison is based on different performance metrics like TP rate, FP rate, and Precision, Error, Recall and ROC area are for different classifiers. Model evaluation or Performance Analysis is based on confusion matrix and the all

related measures like- ROC curve, True positives, True negatives, False positives, False negatives, Error rate (ERR), Accuracy, Sensitivity, Specificity, Precision etc.

3.7.1 Confusion Matrix

Confusion matrix is one of the most common and easiest metrics used for finding the correctness and accuracy of the model. It is used for Classification problem where the output can be of two or more types of classes. The confusion matrix is a table with two dimensions "Actual class" and "Predicted class" in both dimensions. Actual classifications are Rows and Predicted ones are columns. In the dataset there are two classes one is Class 1 and another is Class 2. A confusion matrix has created as follows:

Table 3.4: Confusion Matrix

		Predicted	
Actual		Positive(Class 1)	Negative(Class 2)
	Positive(Class 1)	TP	FN
	Negative(Class 2)	FP	TN

3.7.2 AUC – ROC curve

AUC - ROC is a graphical representation for performance measurement for classification problem. ROC is a probability curve and AUC represents degree or measure of area. It expresses how much model is capable of distinguishing between classes. Higher the AUC, better the model is at predicting 0s as 0s and 1s as 1s. By analogy, Higher the AUC, better the model is at distinguishing between patients with disease and no disease. The ROC curve is plotted with TPR against the FPR where TPR is on y-axis and FPR is on the x-axis.

True Positives (TP): True positives are the cases when the actual class of the data point was True and the predicted is also True.

True Negatives (TN): True negatives are the cases when the actual class of the data point was False and the predicted is also false.

False Positives (FP): False positives are the cases when the actual class of the data point was False and the predicted is True.

False Negatives (FN): False negatives are the cases when the actual class of the data point was true and the predicted is False.

Sensitivity (Recall or True Positive Rate): Sensitivity (SN) is calculated as the number of correct positive predictions divided by the total number of positives. It is also called recall (REC) or true positive rate (TPR).

$$SN = (TP)/(TP+ FN) = (TP)/P \quad (3.10)$$

Specificity (True Negative Rate): Specificity (SP) is calculated as the number of correct negative predictions divided by the total number of negatives. It is also called true negative rate (TNR).

$$SP = TN/(TN+ FP) = (TN)/N \quad (3.11)$$

Precision (Positive Predictive Value): It is also called positive predictive value (PPV). The best precision is 1.0, whereas the worst is 0.0.

$$PREC = (TP)/(TP+ FN) \quad (3.12)$$

Accuracy (ACC) is calculated as the number of all correct predictions divided by the total number of the dataset. Accuracy comparison is based on the performance among the four classification algorithms.

$$ACC = (TP + TN)/(TP+TN+FP+FN) = (TP+TN)/(P+N) \quad (3.13)$$

3.6.3 Error

Error rate is one of the performance measurement parameter. Error rate (ERR) is calculated as the number of all incorrect predictions divided by the total number of the dataset. The best error rate is 0.0, whereas the worst is 1.0.

$$ERR = (FP + FN)/(TP + TN + FP + FN) = (FP + FN)/(P + N) \quad (3.14)$$

3.6.4 Accuracy Testing For New Samples

After whole model development and performance analysis with different classification algorithm, the values of different models are investigated with the

performance on the classification methods. Then accuracy testing for new samples is done by using 20 numbers of new samples. These samples are completely new for the model. These will run on the model and then accuracy is testing by the values.

New samples data is given in appendix A-4.

3.8 SUMMARY

This methodology chapter has discussed about the dataset and how to process dataset and pattern classification algorithms and basic terminologies which are used in this work such as Scaled Conjugate Gradient (SCG) algorithm, Levenberg Marquardt algorithm (LM) Based on Back Propagation and Learning Vector Quantization algorithm and decision tree classification algorithm. Performance analysis terminologies such as confusion matrix, roc curve, error calculation etc. also have included in this chapter.

CHAPTER 4

RESULTS AND DISCCUSION

4.1 INTRODUCTION

In this chapter, the results of different ANN pattern recognition techniques and decision tree classification algorithm are discussed. Result produces from all the selected algorithms are compared with result from the evaluation process. In this paper different classification algorithm works with SCG, LM, LVQ and DT algorithm. Their performance was evaluated and selects a final model with minimum MSE.

4.2 DATA ANALYSIS

Result of data analysis of dengue disease which is obtained under Section 3.4 is described below:
Gender parameter's infection rate is shown in table 4.1.

Table 4.1: Infection rate of gender parameter

Gender	Infected person	Infection Rate (%)
Male	72	63.16
Female	42	36.84
	Total=114	

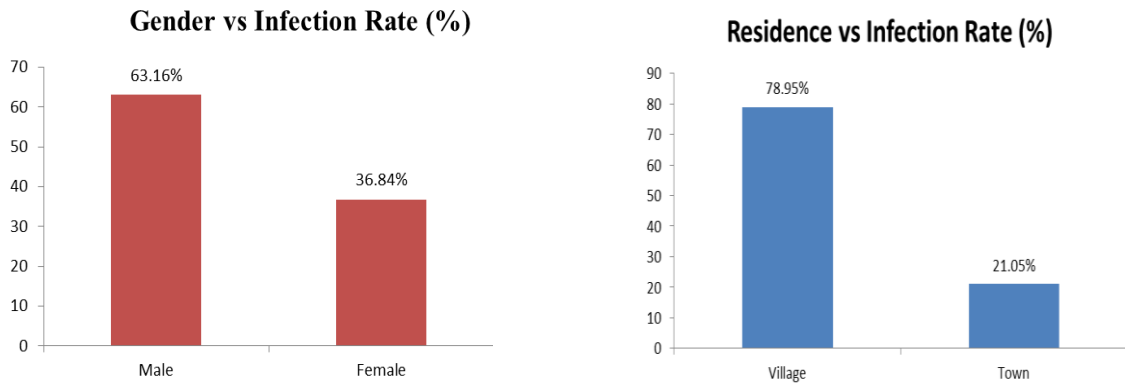
From above table 4.1, it shows that infection rate of male is 63.16% and female infection rate is 36.84%.

Residence parameter's infection rate is shown in table 4.2

Table 4.2: Infection rate of residence parameter

Residence	Infected person	Infection Rate (%)
Village	90	78.95
Town	24	21.05

The relation between gender parameter, residence parameter etc. and dengue disease occurrence in the term of infection rate (%) is shown in figure 4.1.



a b

Figure 4.1: Infection rate of different parameter ((a) Gender parameter, (b) Residence parameter)

The above figure 4.1(a) shows male is more affected than female and the infection rate of male is 63.16% and female is 36.84%. The above figure 4.1(b) shows people who are living in village are more affected than town's people. The infection rate of village is 78.05% and town is 21.05%.

4.3 PERFORMANCE OF DIFFERENT MULTILAYER NEURAL NETWORKS

Performance of different multilayer neural networks which is obtained under Section 3.4 is described below:

4.3.1 Scaled Conjugate Gradient (SCG) Algorithm

For train dataset with Scaled Conjugate Gradient algorithm; load dataset, initialize input parameter and target value. There are 25 parameters are used as an input in input layer, one hidden layer with 10 neurons which is shown in figure 4.2.

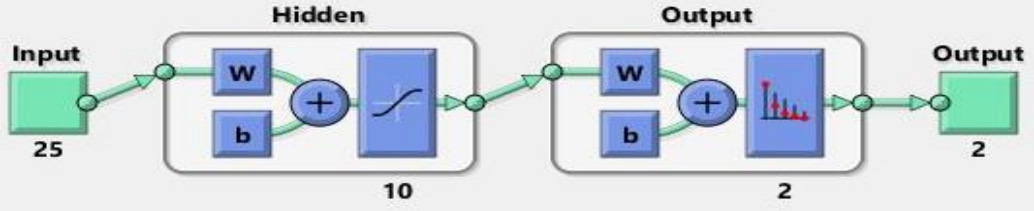
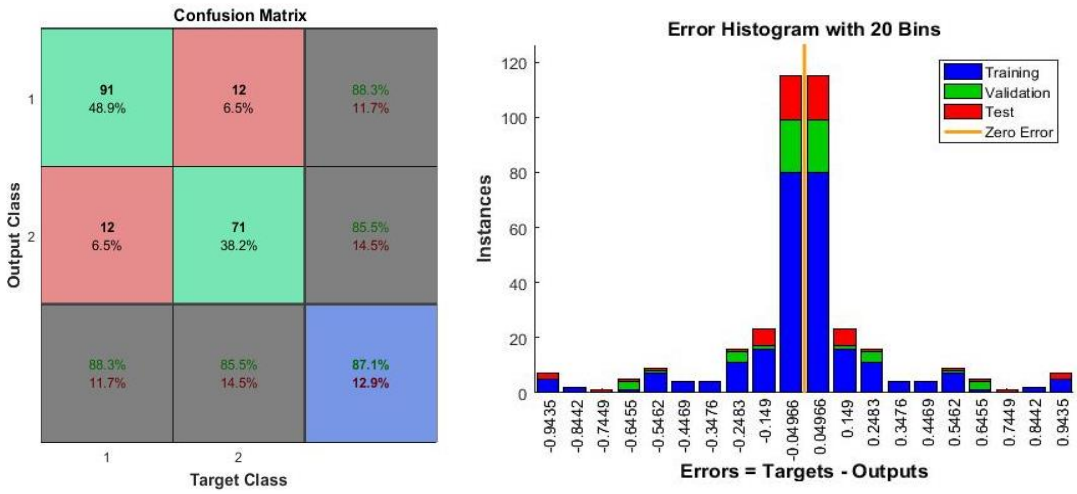


Figure 4.2: Architecture of Scaled Conjugate Gradient algorithm

The figure 4.3(a) is stand for confusion matrix by scaled conjugate gradient algorithm where the green cell is shown where the output class is matched with the target class. And the red cell is shown where the output class is not matched with the target class. The figure 4.3(b) shows the error histogram for SCG algorithm. X axis represents errors (Targets-Outputs) and Y axis represents instances.



a b

Figure 4.3: (a) Confusion matrix for Scaled Conjugate Gradient
(b) Error histogram for Scaled Conjugate Gradient

In the figure 4.3(a), the model uses dataset which is dengue dataset, all rows classes stand for the output class. The columns show the target classes. In the first column below, 88.3% of the class1 which is the true positive rate and 11.7% are incorrectly classified so 11.7% is the false negative rate. In the next column, 85.5% of the class2 are correctly classified so 85.5% is the true negative rate for correctly classified points in this class and 14.5% are misclassified which is false positive rate. In the figure 4.3(b), error range is 0.2483 to -0.2483 which is the closest of zero error.

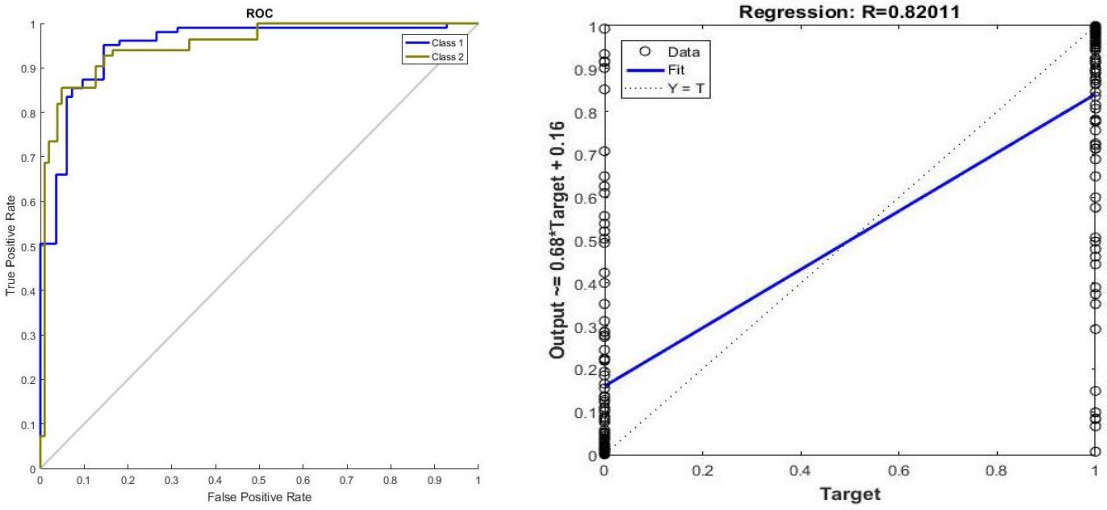
The confusion matrix is generated from the "trainscg" function by using SCG ANN algorithm and the results value is shown in Table 4.3

Table 4.3: Accuracy result of Scaled Conjugate Gradient algorithm

Class	**No. of Instances**	**Correctly predicted instances**	**Incorrectly classified instances**	**Error (%)**	**Accuracy (%)**
Class '1'	103	91	12	11.7	88.3
Class '2'	83	71	12	14.5	85.5
Total	186	162	24	12.9	87.1

In the table 4.3, correctly classified instances are 162 and incorrectly classified instances are 24 which lead 87.1% and 12.9% respectively.

In figure 4.4(a) is for ROC curve for measuring performance. The Y axis is stand for true positive rate and X axis is for false positive rate. Area under ROC for both Class 1 and Class 2 is 0.95.



a b

Figure 4.4: (a) ROC Curve for measuring performance of Scaled Conjugate Gradient algorithm

(b) Regression graph of Scaled Conjugate Gradient algorithm

Regression graph of SCG algorithm is shown in figure 4.4(b). The dashed line in each plot represents the perfect result – outputs = targets. The solid line represents the best fit linear regression line between outputs and targets. The R value is an indication of the relationship between the outputs and targets. If R = 1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Here R = 0.82011 which is closer to 1.

Performance graph is shown in figure 4.5 for training, validating and testing of dengue dataset by SCG algorithm. X axis represents 'Cross-Entropy' which is a loss function and Y axis represents epochs. Best validation performance is found 0.102 at 11 epochs.

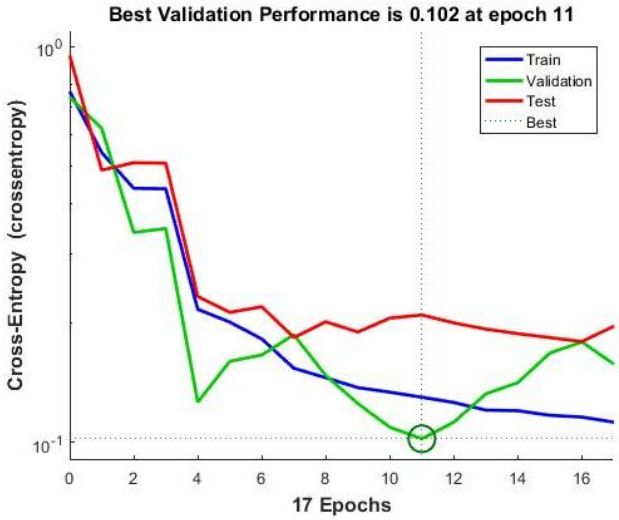


Figure 4.5: Performance graph of train, validation and test for Scaled Conjugate Gradient algorithm

4.3.2 Levenberg Marquardt Algorithm (LM)

For train dataset with Levenberg Marquardt algorithm; load dataset, initialize input parameter and target value. There are 25 parameters are used as an input in input layer, one hidden layer with 10 neurons. Basic structure of LM algorithm is shown in figure 4.6.

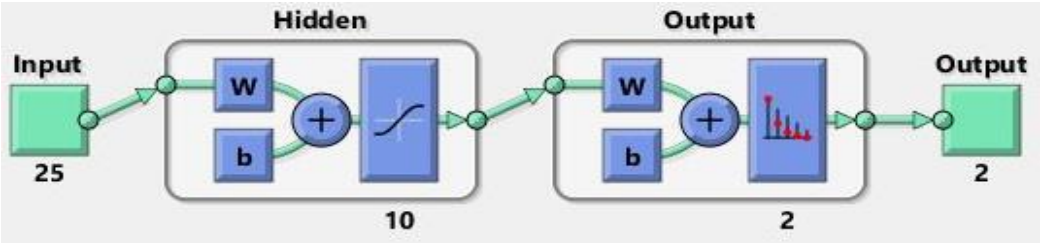
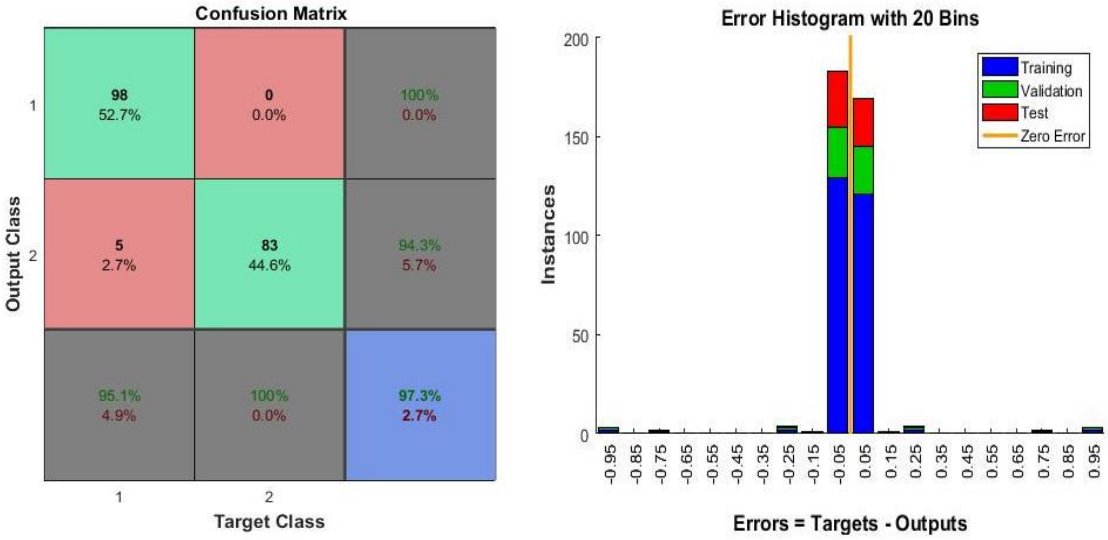


Figure 4.6: Architecture of Levenberg Marquardt algorithm

The figure 4.7(a) is stand for confusion matrix by LM algorithm where the green cell is shown where the output class is matched with the target class. And the red cell is shown where the output class is not matched with the target class. The following figure 4.7(b) shows the error histogram for LM algorithm. X axis represents errors (Targets-Outputs) and Y axis represents instances.



a b

Figure 4.7: (a) Confusion matrix of Levenberg Marquardt algorithm
(b) Error histogram of Levenberg Marquardt algorithm

In this figure 4.7(a), the model uses dataset which is dengue dataset, all rows classes stand for the output class. The columns show the target classes. In the first column below, 95.1% of the class1 which is the true positive rate and 4.9% are incorrectly classified so 4.9% is the false negative rate. In the next column, 100% of the class2 are correctly classified so 100% is the true negative rate for correctly classified points in this class and false positive rate is zero. In this figure 4.7(b), error range is the very closest to zero error. Error range is -0.05 to 0.05.

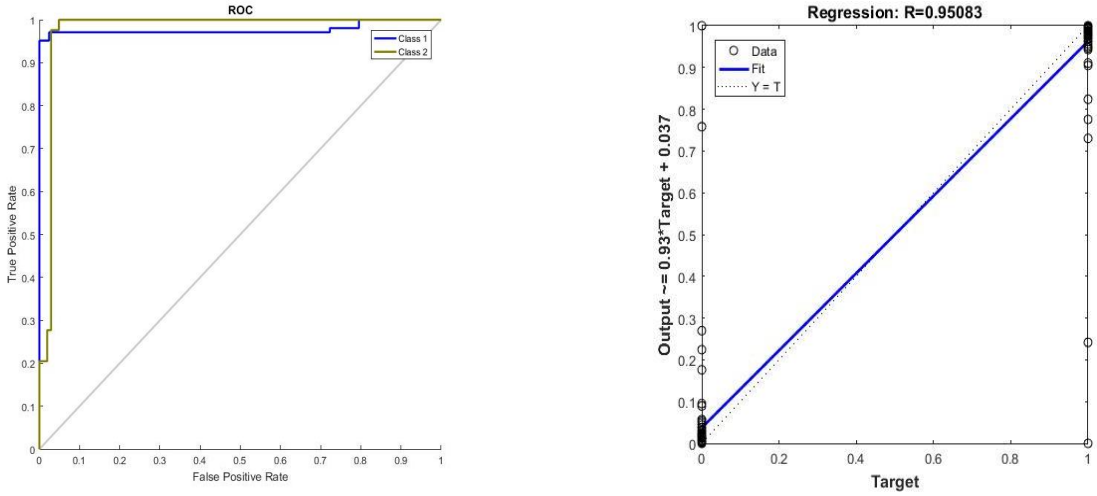
The confusion matrix is generated from the "trainlm" function by using LM ANN algorithm and the output value is shown in Table 4.4.

Table 4.4: Accuracy result of Levenberg Marquardt algorithm

Class	No. of Instances	Correctly predicted instances	Incorrectly classified instances	Error (%)	Accuracy (%)
Class '1'	103	98	5	4.9	95.1
Class '2'	83	83	0	0	100
Total	186	181	5	2.7	97.3

In the table 4.4 correctly classified instances are 181 and incorrectly classified instances are 5 which lead 97.3% of accuracy and 2.7% of error respectively.

In figure 4.8(a) is for ROC curve for measuring performance. The Y axis is stand for true positive value and X axis is for false positive value. Regression graph of LM algorithm is shown in figure 4.8(b). The dashed line in each plot represents the perfect result – outputs = targets. The solid line represents the best fit linear regression line between outputs and targets.



a b

Figure 4.8: (a) ROC Curve for measuring performance of Levenberg Marquardt
(b) Regression graph of Levenberg Marquardt algorithm

In figure 4.8(a), area under ROC for both Class 1 and Class 2 is 0.99. In the figure 4.8(b), the R value is an indication of the relationship between the outputs and targets. If R = 1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Here R = 0.95083 which is closer to 1.

Performance graph is shown in figure 4.9 for training, validating and testing of dengue dataset of LM algorithm. X axis represents 'Mean Square Error (MSE)' which is a loss

function and Y axis represents epochs. Best validation performance is found 0.058467 at 18 epochs.

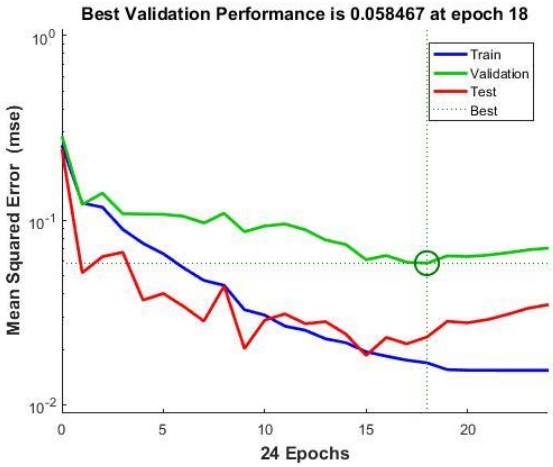


Figure 4.9: Performance graph of train, validation and test for Levenberg Marquardt algorithm

4.3.3 Learning Vector Quantization Algorithm

For train dataset with Learning Vector Quantization algorithm; load dataset, initialize input parameter and target value. There are 25 parameters are used as an input in input layer, one hidden layer with 10 neurons. There is no bias in this method. Basic structure of LVQ algorithm is shown in figure 4.10.

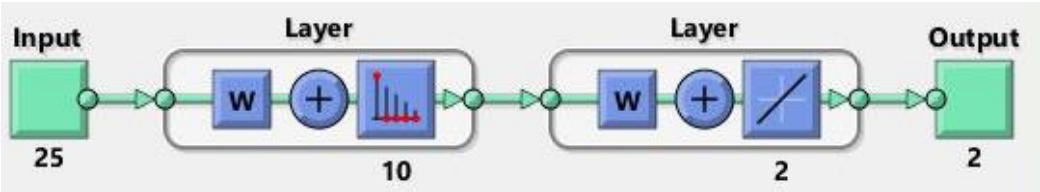
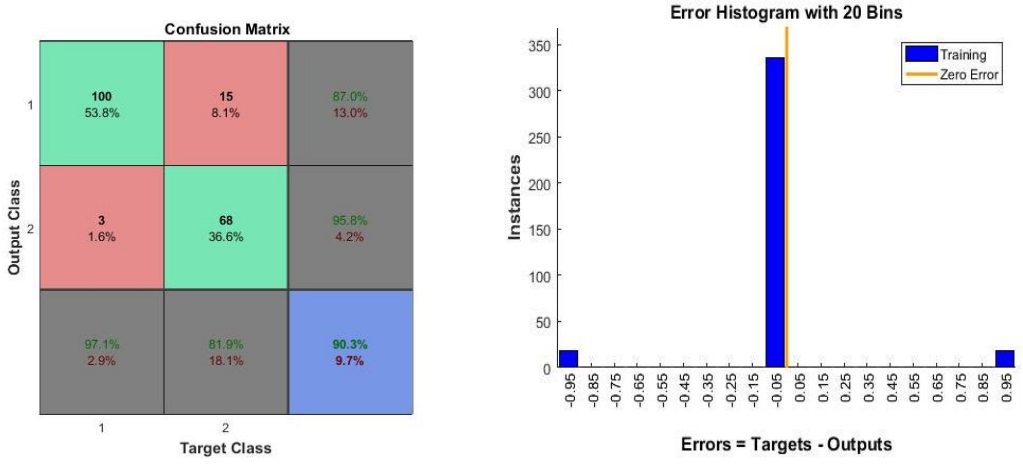


Figure 4.10: Architecture of Learning Vector Quantization algorithm

The figure 4.11(a) is stand for confusion matrix by LVQ where the green cell is shown where the output class is matched with the target class. And the red cell is shown where the output class is not matched with the target class. The following figure 4.11(b) shows the error histogram for LVQ algorithm. X axis represents errors (Targets-Outputs) and Y axis represents instances.



a b

Figure 4.11: (a) Confusion matrix of Learning Vector Quantization algorithm
(b) Error histogram for Learning Vector Quantization algorithm

In this figure 4.11(a), the model uses dataset which is dengue dataset, all rows classes stand for the output class. The columns show the target classes. In the first column below, 97.1% of the class1 which is the true positive rate and 2.9% are incorrectly classified so 2.9% is the false negative rate. In the next column, 81.9% of the class2 are correctly classified so 81.9% is the true negative rate for correctly classified points in this class and 18.1% are misclassified which is false positive rate. In figure 4.11(b), Small instances occurred error in the -0.95, 0.95 and large instance occurred error show in -0.05 which is the closest of zero error.

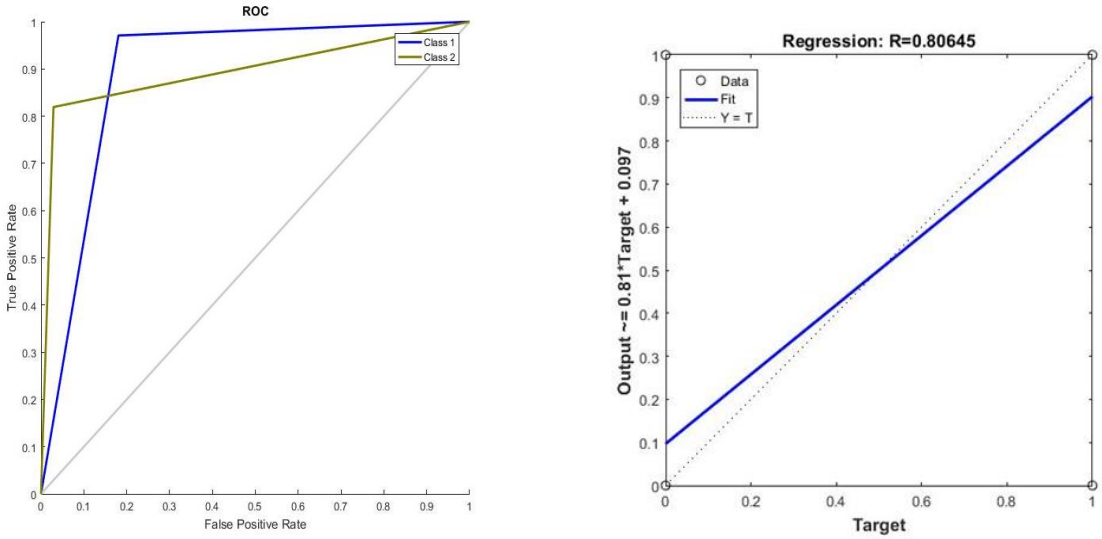
The confusion matrix is generated from the LVQ ANN algorithm and the output value is shown in Table 4.5.

Table 4.5: Accuracy result of Learning Vector Quantization algorithm

Class	No. of Instances	Correctly predicted instances	Incorrectly classified instances	Error (%)	Accuracy (%)
Class '1'	103	100	3	2.9	97.1
Class '2'	83	68	15	18.1	81.9
Total	186	168	18	9.7	90.3

In the above table 4.5 correctly classified instances are 168 and incorrectly classified instances are 18 which lead 90.3% and 9.7% respectively.

Following curve shown in figure 4.12(a) is for ROC curve for measuring performance. Y axis is stand for true positive rate and X axis is for false positive rate. Regression graph of LVQ algorithm is shown in figure 4.12(b). The dashed line in each plot represents the perfect result – outputs = targets. The solid line represents the best fit linear regression line between outputs and targets.



a b

Figure 4.12: (a) ROC Curve for measuring performance of Learning Vector Quantization algorithm

(b) Regression graph of Learning Vector Quantization algorithm

In figure 4.12(a), area under ROC for both Class 1 and Class 2 is 0.97. In figure 4.12(b), R value is an indication of the relationship between the outputs and targets. If R = 1, this indicates that there is an exact linear relationship between outputs and targets. If R is close to zero, then there is no linear relationship between outputs and targets. Here R = 0.80645 which is closer to 1.

Performance graph is shown below for training, validating and testing of dengue dataset by LVQ algorithm. X axis represents 'Cross-Entropy' which is a loss function and Y axis represents epochs. Best validation performance is found 0.086022 at 224 epochs.

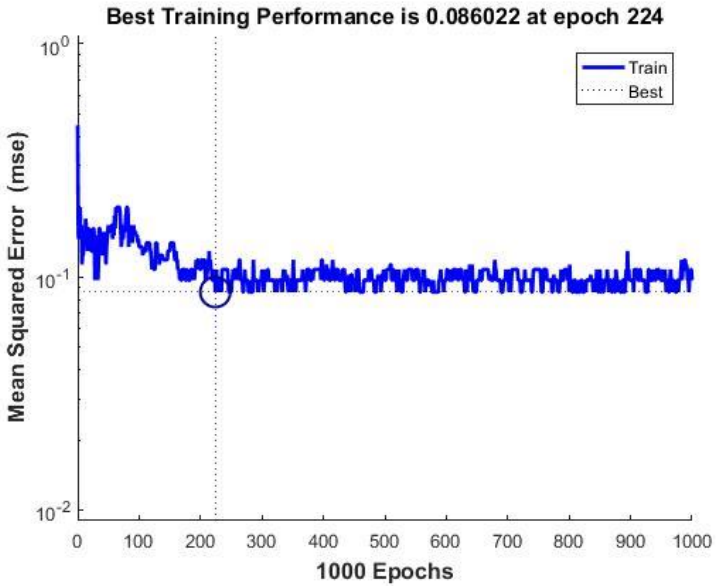
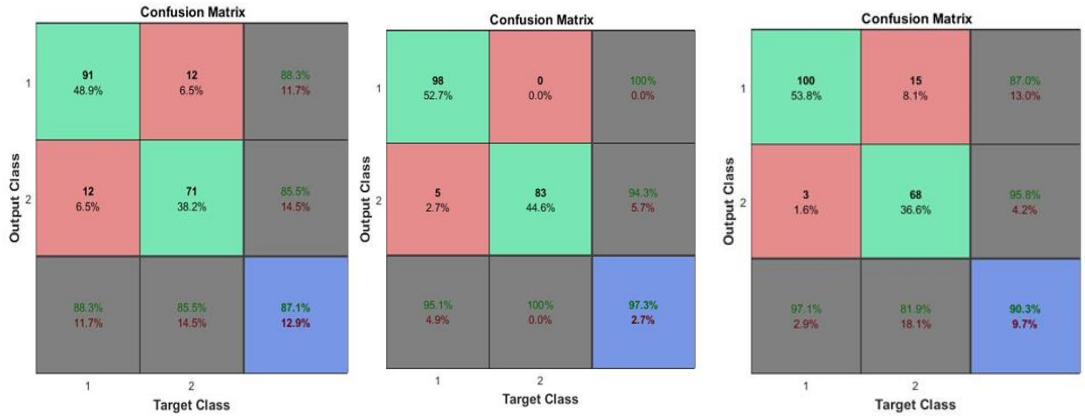


Figure 4.13: Performance graph of train, validation and test for Learning Vector Quantization algorithm

4.4 COMPARATIVE ANALYSIS OF DIFFERENT ANN

Comparison of different performance terminologies like confusion matrix, ROC, regression value curve, error histogram, performance graph etc. of different artificial neural networks are shown below :

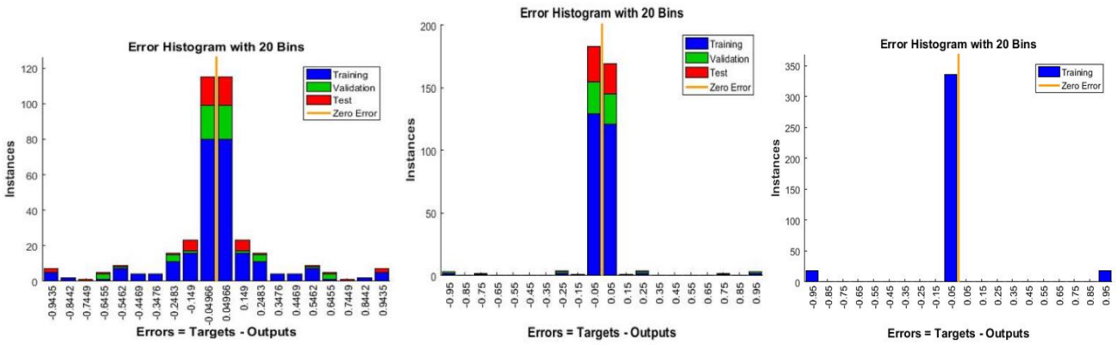
Comparison of confusion matrix between different artificial neural networks is shown in figure 4.14. In this figure 4.14(a) represents confusion matrix of SCG algorithm, 4.14(b) represents confusion matrix of LM algorithm and 4.14(c) represents confusion matrix of LVQ algorithm. This figure shows that SCG accuracy is 87.1%, LM accuracy is 97.3% and LVQ accuracy is 87.3%.



a b c

Figure 4.14: Comparison of confusion matrix between different artificial neural networks (a) SCG, b) LM and c) LVQ)

Comparison of error histogram between different artificial neural networks is shown in figure 4.15. X axis represents errors (Targets-Outputs) and Y axis represents instances. In this figure 4.15(a) represents error histogram of SCG algorithm, 4.15(b) represents error histogram of LM algorithm and 4.15(c) represents error histogram of LVQ algorithm.



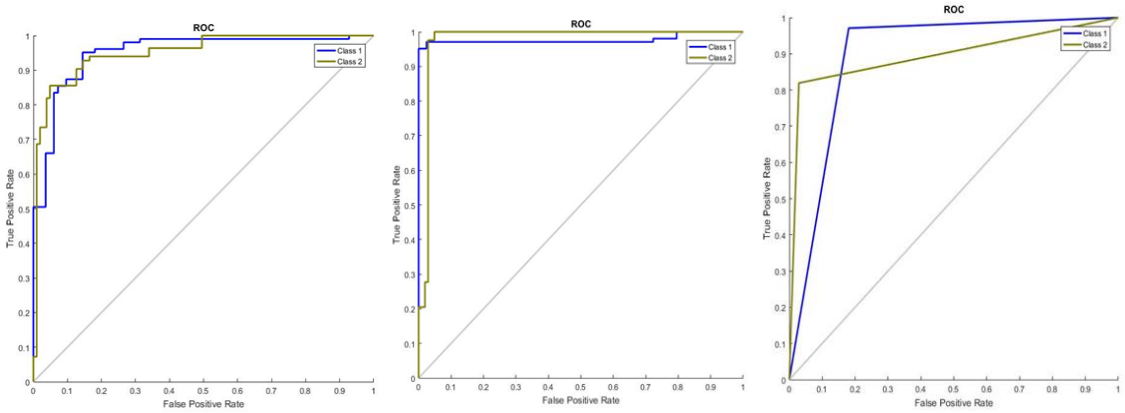
a b c

Figure 4.15: Comparison of error histogram between different artificial neural networks (a) SCG, b) LM and c) LVQ)

In above figure, 4.15(a) shows that error range is 0.2483 to -0.2483 which is the closest of zero error, 4.15(b) shows that error range is the very closest to zero error. Error range

is -0.05 to 0.05 and 4.15(c) shows that small instances occurred error in the -0.95 to 0.95 and large instance occurred error show in -0.05 which is the closest of zero error.

Comparison of ROC between different artificial neural networks is shown in figure 4.16. The Y axis is stand for true positive rate and X axis is for false positive rate. In this figure 4.16(a) represents ROC of SCG algorithm, 4.16(b) represents ROC of LM algorithm and 4.16(c) represents ROC of LVQ algorithm.



a b c

Figure 4.16: Comparison of ROC between different artificial neural networks (a) SCG, b) LM and c) LVQ)

In above figure, 4.16(a) shows that area under ROC for both Class 1 and Class 2 is 0.95 4.16(b) shows that area under ROC for both Class 1 and Class 2 is 0.99 and 4.16(c) shows that area under ROC for both Class 1 and Class 2 is 0.95. So ROC of LM algorithm is higher than other algorithms.

Comparison table of prediction accuracy, error accuracy, Regression value, ROC and MSE between SCG, LM and LVQ algorithm is shown table 4.6.

Table 4.6: Comparison of performance between different algorithms

Name of algorithm	Accuracy (%)	Error (%)	Regression value	ROC	MSE
SCG	87.1	12.9	0.82011	0.95	0.13767
LM	97.3	2.7	0.95083	0.99	0.02410
LVQ	90.3	9.7	0.80645	0.97	0.09677

In above table 4.6 shows that LM algorithm's prediction accuracy is higher than other algorithms which value is 97.3%.

Comparison of accuracy and error between different ANN algorithms is shown in figure 4.17.

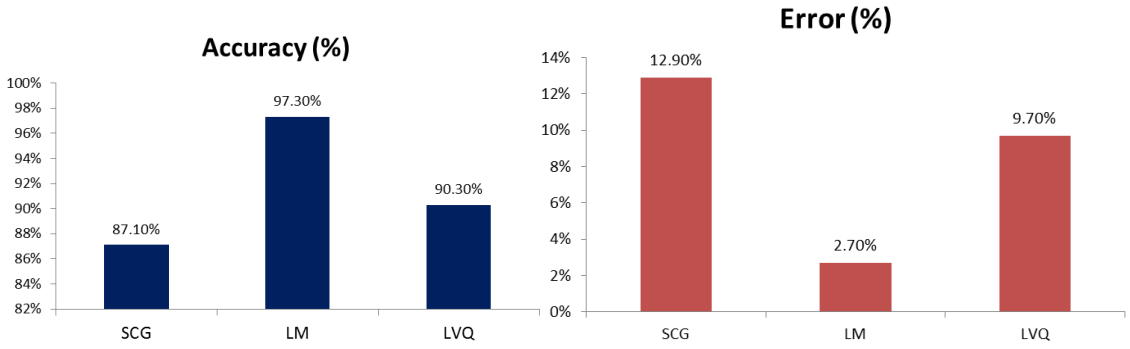


Figure 4.17: Comparison chart of accuracy and error of different neural networks algorithms

The above figure 4.17 shows that LM algorithm has higher accuracy rate which is 97.3% and lower error rate which is 2.70%. These bar charts are built in excel sheet.

Comparison of regression value and ROC between different ANN algorithms is shown in figure 4.18.

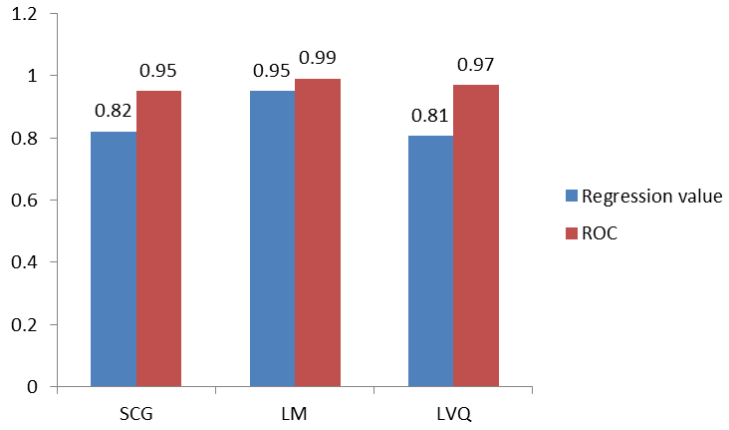


Figure 4.18: Comparison chart of regression value and ROC of different algorithms

From the above figure 4.18, it visibly shows that Levenberg–Marquardt (LM) algorithm has maximum regression value 0.95 and ROC is 0.99 which is larger than other algorithms. This bar chart is built in excel sheet.

From discussion of this section, we selected the ANN developed with the data set as the input variables, is a Multi-Layer Perceptron Model with three layers. There is one (01) hidden layer which has four (10) neurons. The activation function of the hidden layer the Log Sigmoid function and the Pure Linear function is used as the output function. The learning algorithms used for this ANN is Levenberg – Marquardt algorithm.

The Marquardt algorithm was experienced on several function approximation problems, and it was compared with the conjugate gradient algorithm and with variable learning rate backpropagation. The results indicated that the Marquardt algorithm is very efficient when training networks which have up to a few hundred weights. Although the computational requirements are much higher for each iteration of the Marquardt algorithm, this is more than made up for by the increased efficiency. This is specially true when high accuracy is essential. It is also found that in many cases the Marquardt algorithm converged when the conjugate gradient and variable learning rate algorithms failed to converge [45].

Now Levenberg – Marquardt algorithm is used for testing with new samples which are unused before.

The steps of the prediction process of LM algorithm are given bellow:

1. Test the dataset with new 20 instances. The new instance data set is in the main dataset. The last 20 samples of total samples are used for prediction accuracy. All data samples are shown in appendices section.
2. Initialize the inputs and targets data.
3. For prediction process, after training process train the new value with training network ; yi = net(xi)
4. Then compute errors by subtracting predicted values from targets value (errors = gsubtract(yi,ti))
5. Performance is calculated by 'perform' function (perf = perform(net,ti,yi)) where perf is the MSE value.

One of the main tasks of the research is to show the prediction with new samples. Since the main goal of the research is to find out the best algorithm which will give better accuracy than the early prediction system of dengue disease. To this end, the output of the prediction is obtained after following the 4.4 section procedure and shown in table 4.7.

Table 4.7: Target and Predicted value of new samples of LM algorithm

Sample no	Target	Predicted	Remarks
1	Class 2	Class 2	Right
2	Class 1	Class 1	Right
3	Class 1	Class 1	Right
4	Class 1	Class 1	Right
5	Class 2	Class 2	Right
6	Class 1	Class 1	Right
7	Class 1	Class 1	Right
8	Class 2	Class 1	Wrong
9	Class 2	Class 2	Right
10	Class 1	Class 1	Right
11	Class 2	Class 2	Right
12	Class 2	Class 2	Right
13	Class 1	Class 1	Right
14	Class 1	Class 1	Right
15	Class 2	Class 2	Right
16	Class 1	Class 1	Right
17	Class 2	Class 2	Right
18	Class 2	Class 2	Right

19	Class 2	Class 2	Right
20	Class 1	Class 1	Right

The accuracy of pred(20) of LM algorithm = (19*100)/20 = 95%

4.5 PERFORMANCE OF DECISION TREE ALGORITHM

4.5.1 Decision Tree

In a Decision tree, there are two nodes, which are the Decision Node and Leaf Node. Decision nodes are used to make any decision and have multiple branches, whereas Leaf nodes are the output of those decisions and do not contain any further branches. From figure 4.19, we can say x1, x3, x15 and x17 variables are the most significant variables to predict dengue disease. Here x1, x3, x15 and x17 variables are respectively Age, Month, Platelets and Rash.

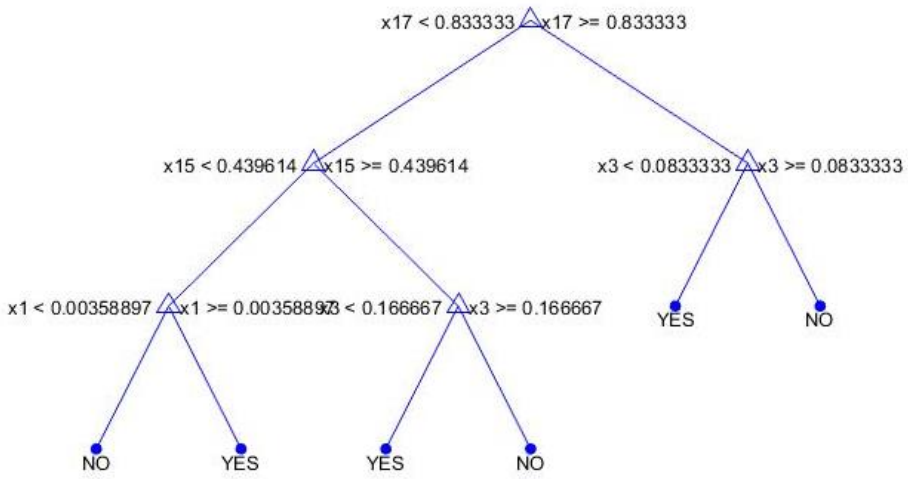


Figure 4.19: Decision tree

4.5.2 Accuracy Result of Decision Tree Algorithm

The model uses dataset which is dengue dataset. Prediction accuracy of dengue disease by decision tree is shown in table 4.8. 103 samples class are ‘yes’ and 83

samples class are ‘no’ among total 186 samples. The first row show that 103 samples are correctly predicted positive class. So 100% of the class ‘yes’ is the true positive rate. The second row show that 81 samples are correctly predicted negative class and 2 samples are incorrectly predicted positive class. So 97.60% of the class ‘no’ is the true negative rate and 2.4% is the false negative rate.

Table 4.8: Prediction accuracy of Decision Tree algorithm

Class	Target	Predicted	Error (%)	Accuracy (%)
‘Yes’	103	103	0	100
‘No’	83	81	2.4	97.60
Total	186	184	1.08	98.92

Unused new 20 samples are used for test. Target values and predicted values of decision tree are given below in table 4.9:

Table 4.9: Target and predicted value of new samples of Decision tree algorithm

Sample no	Target	Predicted	Remarks
1	'NO'	'NO'	Right
2	'YES'	'YES'	Right
3	'YES'	'YES'	Right
4	'YES'	'YES'	Right
5	'NO'	'NO'	Right
6	'YES'	'YES'	Right
7	'YES'	'YES'	Right
8	'NO'	'NO'	Right
9	'NO'	'NO'	Right
10	'YES'	'YES'	Right
11	'NO'	'NO'	Right
12	'NO'	'NO'	Right
13	'YES'	'YES'	Right
14	'YES'	'YES'	Right
15	'NO'	'YES'	Wrong
16	'YES'	'YES'	Right
17	'NO'	'NO'	Right
18	'NO'	'NO'	Right
19	'NO'	'YES'	Wrong

20	'YES'	'YES'	Right

The accuracy of decision tree algorithm of pred(20) = (18*100)/20 = 90%.

4.6 COMPARATIVE ANALYSIS OF DIFFERENT NEURAL NETWORKS ALGORITHMS AND DECISION TREE ALGORITHM

Comparison of training accuracy of different training artificial neural network's algorithms and decision tree is given table 4.10.

Table 4.10: Comparison of training accuracy between different neural Networks algorithms and Decision tree algorithm

Name of algorithm	Accuracy (%)	Error (%)
SCG	87.1	12.9
LM	97.3	2.7
LVQ	90.3	9.7
DT	98.92	1.08

From the above table, it shows that decision tree algorithm gives higher accuracy in training than other training algorithms.

Comparison of training accuracy between different neural Networks algorithms and Decision tree algorithm is shown in figure 4.19.

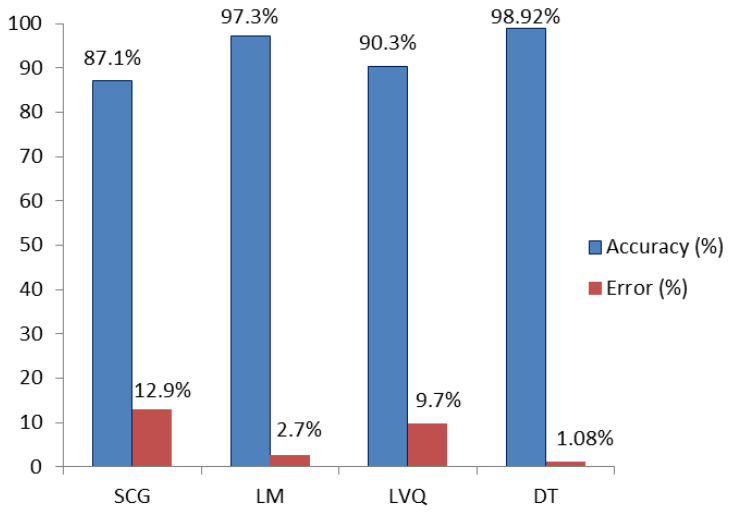


Figure 4.20: Comparison chart of training accuracy between different Neural Networks algorithms and Decision Tree algorithm

From above figure 4.19, it can visibly show that decision tree algorithm gives the higher accuracy and lower error in training than other training algorithms.

Accuracy Comparison of tested with new samples both DT and LM algorithm is shown in figure 4.20. This bar chart is built by excel sheet.

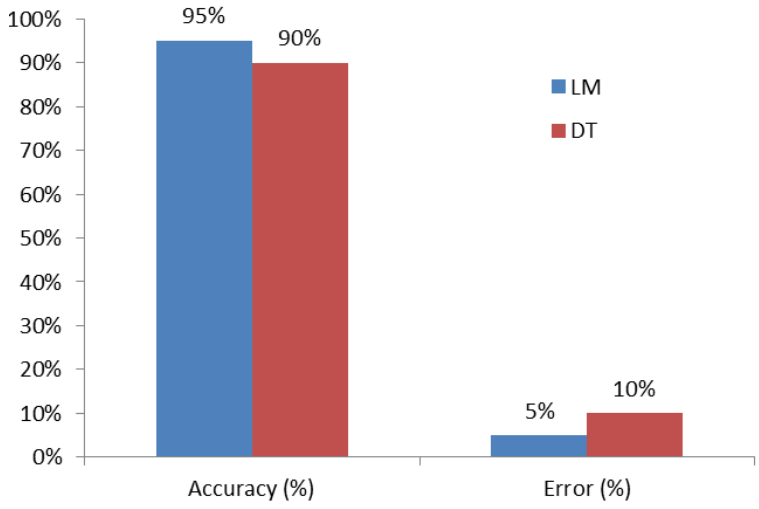


Figure 4.21: Comparison chart of Levenberg Marquardt algorithm and Decision tree algorithm

From above figure 4.20, it visibly shows that though decision tree algorithm has higher training accuracy than LM algorithm, decision tree gives lower accuracy than LM algorithm in testing with new samples which is 95%.

4.7 COMPARISON WITH PREVIOUS RELATED WORK

P.H.M. Nishanthi Herath, A.A.I. Perera and H.P.Wijekoon in their paper "Prediction of Dengue Outbreaks in Sri Lanka using Artificial Neural Network" showed Multi -Layer Perceptron Model with three layers, Log Sigmoid function and the Pure Linear output function and Levenberg – Marquardt algorithm gives 68.5 % accuracy [1]. Md. Nazmul Karim, Saif Ullah Munshi, Nazneen Anwar & Md. Shah Alam in their paper "Climatic factors influencing dengue cases in Dhaka city: a model for dengue prediction" showed Multiple regression analysis which gives Area under ROC = 0.89, 95% CI = 0.89-0.98 [4]. Yuhanis Yusof and Zuriani Mustaffa in their paper "Dengue Outbreak Prediction: A Least Squares Support Vector Machines Approach" showed Least Squares Support Vector Machine (LS-SVM) gives approximately 86.84% [5]. In [6], they showed prediction accuracy 86.5% in Matara district and 89.5% in Colombo district by ANN. K. Balasaravanan and M. Prakash showed 92% accuracy in their paper by using ANN model [8]. In [16], they used MFNN to predict dengue disease and showed 90% prediction accuracy. They showed 85.92% accuracy by using Entropy technique and artificial neural network in their paper [18]. M. Naresh Kumar showed 89% accuracy by using alternating decision tree [20]. In [23], they showed 84% accuracy by using SVM in their paper.

Comparison of this work and previous related work is shown in below:

Sl No.	Author	Methods/ Techniques	Accuracy of previous work (%)	Accuracy of this work (%)
[1]	P.H.M. Nishanthi Herath, A.A.I. Perera and H.P.Wijekoon	Multi -Layer Perceptron Model	Accuracy of 68.5%	Accuracy of 95%
[4]	Md. Nazmul Karim, Saif Ullah Munshi, Nazneen Anwar & Md. Shah Alam	Multiple regression analysis	Area under ROC = 0.89, 95% CI = 0.89-0.98	Area under ROC = 0.99, 95%
[5]	Yuhanis Yusof and Zuriani Mustaffa	Least Squares Support Vector Machine (LS-SVM)	approximately 86.84%	95%
[6]	Aloka. Munasinghe, H.L. Premaratne, and M.G.N.A.S. Fernando	ANN	86.5% accuracy in Matara district and 89.5% accuracy in Colombo district	Accuracy 95% in Noakhali disctrict in Bangladesh
[8]	K. Balasaravanan and M. Prakash	ANN	Accuracy 92%	Accuracy 95%
[16]	Fatimah Ibrahima, Mohd Nasir Taib, Wan Abu Bakar Wan Abas, Chan Chong Guan, Saadiah Sulaiman	Multilayer feed-forward neural networks (MFNN).	90% prediction accuracy	95% prediction accuracy
[18]	Napa Rachata, Phasit Charoenkwan, Thongchai Yooyativong, Kosin Chamnongthai, Chidchanok Lursinsap , and Kohji Higuchi	Entropy technique and artificial neural network.	85.92% accuracy	95% accuracy
[20]	M. Naresh Kumar	Alternating Decision tree	accuracy of 89%	accuracy of 95%
[23]	M.Bhavani and S.Vinod kumar	SVM	accuracy of 84%	accuracy of 95%

4.8 SUMMARY

In this chapter, analyze the parameters, results of all algorithms are discussed in details and select the final model among neural networks and decision tree algorithm which is considered for this work. And show the comparison of previous related work with this work.

CHAPTER 5

CONCLUSION

5.1 INTRODUCTION

This chapter describes the conclusion and future research scope of this thesis work. First the overall summary of this thesis work is presented in the conclusion section. Then the potential future research directions are provided in future research scope section

5.2 CONCLUSION

This research focuses on the use of different supervised classification techniques to enhance the early detection dengue disease. Dengue disease is a fatal disease which may cause life threatening complication such as death sometimes. Here used real time dataset of dengue patients which are collected from hospital and clinic. Parameters are analyzed to find parameter's infection rate in this paper. Infection rate of male is 63.16%, female infection rate is 36.84%, village residence people's infection rate is 78.25% and town residence people's infection rate is 21.75%. In proposed model, all the experiment is based on real dataset and MATLAB software is used for experiments. This research work used multi-layer perceptron neural networks algorithms namely SCG, LM and LVQ and decision tree classification algorithm. 25 parameters are used as an input in input layer, one hidden layer with 10 neurons is used for each training algorithm. Each algorithm's outcome is used to determine the better classifiers among them. Comparisons of these selected algorithms are based on the performance factors.

In training state, Scaled Conjugate Gradient (SCG) Algorithm gives 87.1% accuracy, 12.9% error rate, MSE 0.138, regression value 0.82 and ROC 0.95. Levenberg–Marquardt (LM) algorithm gives 97.3% accuracy, 2.7% error rate, MSE 0.024, regression value 0.95 and ROC 0.99. Learning Vector Quantization (LVQ) Algorithm

gives 90.3% accuracy, 9.7% error rate, MSE 0.096, regression value 0.81 and ROC 0.97 and Decision tree algorithm gives 98.92% accuracy. But in testing with new unused samples, Levenberg–Marquardt (LM) algorithm gives 95% accuracy and Decision tree gives 90% accuracy.

By the evaluation the experimental results of all algorithms, this work concludes that, Levenberg–Marquardt (LM) algorithm can be considered as a best algorithm because of its highest prediction accuracy 95%. As LM gives better accuracy it can be used for early dengue disease prediction.

This research paper's goal is to provide better performance and prediction. For calculating performance used MLPNN with learning algorithm LM.

5.3 FUTURE RESEARCH SCOPE

There is a possibility to generate a new algorithm or a hybrid one which will provide better accuracy than the existing algorithms. Data from several districts of Bangladesh can be used for wide research in this topic in future. Weather variables like rainfall, humidity, wind speed, geographic region etc. can be used with the clinical dataset as an input parameters.

REFERENCES

1. Nishanthi, P. H. M., A. A. I. Perera, and H. P. Wijekoon. "Prediction of dengue outbreaks in Sri Lanka using artificial neural networks." International Journal of Computer Applications 101.15 (2014).
2. Aburas, Hani M., B. Gultekin Cetiner, and Murat Sari. "Dengue confirmed-cases prediction: A neural network model." Expert Systems with Applications 37.6 (2010): 4256-4260.
3. Cetiner, B. Gultekin, Murat Sari, and H. M. Aburas. "Recognition of dengue disease patterns using artificial neural networks." 5th International Advanced Technologies Symposium (IATS'09). 2009.
4. Karim, Md Nazmul, et al. "Climatic factors influencing dengue cases in Dhaka city: a model for dengue prediction." The Indian journal of medical research 136.1 (2012): 32.
5. Yusof, Yuhanis, and Zuriani Mustaffa. "Dengue outbreak prediction: A least squares support vector machines approach." International Journal of Computer Theory and Engineering 3.4 (2011): 489.
6. Munasinghe, Aloka, H. Premaratne, and M. G. N. A. S. Fernando. "Towards an early warning system to combat dengue." International Journal of Computer Science and Electronics Engineering 1.2 (2013): 252-256.
7. Balasaravanan, K., and M. Prakash. "Detection of dengue disease using artificial neural network based classification technique." Int. J. Eng. Technol. 7.13 (2018): 13-15.
8. Ughelli, V., et al. "Prediction of Dengue Cases in Paraguay Using Artificial Neural Networks." The 3rd Int'l Conf on Health Informatics and Medical Systems. 2017.

9. Mello-Román, Jorge D., et al. "Predictive Models for the Medical Diagnosis of Dengue: A Case Study in Paraguay." Computational and mathematical methods in medicine 2019 (2019).
10. Paul, Kishor Kumar, et al. "Risk factors for the presence of dengue vector mosquitoes, and determinants of their prevalence and larval site selection in Dhaka, Bangladesh." PloS one 13.6 (2018): e0199457.
11. Siriyasatien, Padet, et al. "Analysis of significant factors for dengue fever incidence prediction." BMC bioinformatics 17.1 (2016): 166.
12. Fathima, Shameem, and Nisar Hundewale. "Comparison of classification techniques-SVM and naives bayes to predict the Arboviral disease-Dengue." 2011 IEEE International Conference on Bioinformatics and Biomedicine Workshops (BIBMW). IEEE, 2011.
13. Shakil, Kashish Ara, Shadma Anis, and Mansaf Alam. "Dengue disease prediction using weka data mining tool." arXiv preprint arXiv:1502.05167 (2015).
14. Iqbal, Naiyar, and Mohammad Islam. "Machine learning for dengue outbreak prediction: An outlook." International Journal of Advanced Research in Computer Science 8.1 (2017).
15. Ahmed, Nadeem, et al. "Role of Expert Systems in Identification and Overcoming of Dengue Fever." International Journal Of Advanced Computer Science And Applications 8.10 (2017): 82-89.
16. Ibrahim, Fatimah, et al. "A novel dengue fever (DF) and dengue haemorrhagic fever (DHF) analysis using artificial neural network (ANN)." Computer methods and programs in biomedicine 79.3 (2005): 273-281.
17. Husin, Nor Azura, and Naomie Salim. "Modeling of dengue outbreak prediction in Malaysia: a comparison of neural network and nonlinear regression model." 2008 International Symposium on Information Technology. Vol. 3. IEEE, 2008.
18. Rachata, Napa, et al. "Automatic prediction system of dengue haemorrhagic-fever outbreak risk by using entropy and artificial neural network." 2008 International Symposium on Communications and Information Technologies. IEEE, 2008.
19. Wu, Yan, et al. "Detect climatic factors contributing to dengue outbreak based on wavelet, support vector machines and genetic algorithm." (2008).

20. Kumar, M. Naresh. "Alternating Decision trees for early diagnosis of dengue fever." arXiv preprint arXiv:1305.7331 (2013).
21. Thitiprayoonwongse D., Suriyaphol P., and Soonthornphisaj N. "Data mining of dengue infection using decision tree." Entropy 2 (2012): 2.
22. Mutsuddy, Pulak, et al. "Dengue situation in Bangladesh: An epidemiological shift in terms of morbidity and mortality." Canadian Journal of Infectious Diseases and Medical Microbiology 2019 (2019).
23. Bhavani, M., and S. Vinod Kumar. "A data mining approach for precise diagnosis of dengue fever." International journal of latest trends in engineering and technology 7.4 (2016).
24. alias Balamurugan, S. Appavu, MS Mohamed Mallick, and G. Chinthana. "Improved prediction of dengue outbreak using combinatorial feature selector and classifier based on entropy weighted score based optimal ranking." Informatics in Medicine Unlocked 20 (2020): 100400.
25. Ibrahim, Fatimah, et al. "Non-invasive diagnosis of risk in dengue patients using bioelectrical impedance analysis and artificial neural network." Medical & biological engineering & computing 48.11 (2010): 1141-1148.
26. Sajana, Navya. "Gayathri, & Reshma.(2018). Classification of Dengue using Machine Learning Techniques." International Journal of Engineering & Technology 7.2: 212-218.
27. Shaukat, Kamran, et al. "Dengue fever prediction: A data mining problem." Journal of Data Mining in Genomics & Proteomics 2015 (2015).
28. Li, Chenlu, et al. "Climate change and dengue fever transmission in China: evidences and challenges." Science of the Total Environment 622 (2018): 493-501.
29. Ali, Mohammad, et al. "Use of a geographic information system for defining spatial risk for dengue transmission in Bangladesh: role for Aedes albopictus in an urban outbreak." The American journal of tropical medicine and hygiene 69.6 (2003): 634-640.
30. Farooqi, Wajeeha, and Sadaf Ali. "A Critical Study of Selected Classification Algorithms for Dengue Fever and Dengue Hemorrhagic Fever." 2013 11th International Conference on Frontiers of Information Technology. IEEE, 2013.

31. SAHANI, MAZRURA, and ZAINUDIN MOHD ALI. "Feature selection algorithms for Malaysian dengue outbreak detection model." Sains Malaysiana 46.2 (2017): 255-265.
32. Lu, Sheng, Zhong-jian Cai, and Xiao-bin Zhang. "Application of GA-SVM time series prediction in tax forecasting." 2009 2nd IEEE International Conference on Computer Science and Information Technology. IEEE, 2009.
33. Dhamodharan, S. "Liver disease prediction using bayesian classification." 4th National Conference on Advanced computing, applications & Technologies. 2014.
34. Solanki, Ashokkumar Vijaysinh. "Data mining techniques using WEKA classification for sickle cell disease." International Journal of Computer Science and Information Technologies 5.4 (2014): 5857-5860.
35. Joshi, Jahanvi, Rinal Doshi, and Jigar Patel. "Diagnosis and prognosis breast cancer using classification rules." International Journal of Engineering Research and General Science 2.6 (2014): 315-323.
36. David, Satish Kumar, A. T. Saeb, and Khalid Al Rubeaan. "Comparative analysis of data mining tools and classification techniques using weka in medical bioinformatics." Computer Engineering and Intelligent Systems 4.13 (2013): 28-38.
37. Vijayarani, S., and S. Sudha. "Comparative analysis of classification function techniques for heart disease prediction." International Journal of Innovative Research in Computer and Communication Engineering 1.3 (2013): 735-741.
38. Durairaj, Manoj, and Veera Ranjani. "Data mining applications in healthcare sector: a study." International journal of scientific & technology research 2.10 (2013): 29-35.
39. Sugandhi, C., P. Ysodha, and M. Kannan. "Analysis of a Population of Cataract Patient Database in WEKA Tool." International Journal of Scientific and Engineering Research 2.10 (2011).
40. Yasodha, P., and M. Kannan. "Analysis of a population of diabetic patients databases in WEKA tool." International Journal of Scientific & Engineering Research 2.5 (2011): 1-5.

41. Suratgar, Amir Abolfazl, Mohammad Bagher Tavakoli, and Abbas Hoseinabadi. "Modified Levenberg-Marquardt method for neural networks training." World Acad Sci Eng Technol 6.1 (2005): 46-48.
42. Cetiner, B. Gultekin, and H. M. Aburas. "Development of a cross-platform artificial neural network component for intelligent systems." JKAU: Eng. Sci 16.2 (2005): 97-113.
43. Lippmann, Richard. "An introduction to computing with neural nets." IEEE Assp magazine 4.2 (1987): 4-22.
44. Nayak, Sasmita, Dr Neeraj Kumar, and Dr BB Choudhury. "Scaled conjugate gradient backpropagation algorithm for selection of industrial robots." International Journal of Computer Application (2250-1797) 7.6 (2017).
45. Sapna, S., A. Tamilarasi, and M. Pravin Kumar. "Backpropagation learning algorithm based on Levenberg Marquardt Algorithm." Comp Sci Inform Technol (CS and IT) 2 (2012): 393-398.

APPENDICES

A. DATASET

A-1 Raw Dataset

Here, Days (no. of days (fever)), TP (Current temperature), Back pain (Back pain of eyes), Hb(Heamoglobin), Hct(Hematocrit), Plt(Platelets). 18 parameters are shown below among 25 parameters.

Sample no	Age	Sex	Month	Residence	Days	TP	WBC	Back pain	Joint pain	Muscle pain	Headache	Vomiting	Hb (g/dL)	Hct (%)	Plt (per cmm)
1	9m	male	march	village	4	102	20380	no	no	no	no	low	9.7	29.6	302000
2	28y	female	july	town	3	102	3200	no	no	medium	high	no	11.5	34.7	198000
3	28y	male	august	village	6	104	4600	no	low	high	high	medium	15.3	44.8	48000
4	9m	male	august	village	4	102	18500	no	no	no	no	high	11.2	34	296000
5	22y	female	august	village	5	103	4500	no	high	high	low	low	10.5	32.3	165000

6	27 y	male	august	village	3	103	4200	no	medium	low	no	no	12.6	37.2	150000
7	1y	female	july	village	3	102	8300	no	no	no	no	low	9.5	28.7	120000
8	28 y	male	august	village	3	104	3400	low	no	medium	high	no	14	39.9	152000
9	65 y	male	august	village	3	104	4200	no	medium	high	medium	medium	12.1	36.3	115000
10	58 y	male	august	village	3	104	4700	no	no	high	medium	no	12.4	36.5	156000
11	18 y	male	august	village	4	103	4300	medium	medium	high	low	no	13.9	41.1	148000
12	50 y	male	august	village	5	104	4600	high	no	high	medium	medium	12.5	36.6	95000
13	24 y	male	august	village	3	102	4500	no	high	high	no	no	11.5	34.3	165000
14	18 y	female	august	village	2	104	4200	low	low	high	high	low	14.4	42.2	155000
15	27 y	male	august	village	7	103	7800	no	medium	high	high	no	12.3	35.2	150000
16	17 y	male	august	village	6	103	6200	low	no	high	high	low	14.7	41.2	110000
17	24 y	male	august	village	3	104	6200	no	low	high	medium	no	13.1	34.9	160000
18	22 y	female	august	village	3	103	4300	high	medium	high	low	medium	11.1	33	175000
19	36 y	male	august	village	4	103	11400	no	low	medium	high	no	13.4	39.3	300000
20	40 y	male	august	village	5	104	4500	high	low	high	low	no	12.4	37.5	183000

21	80 y	male	august	village	7	105	13700	no	medium	high	no	low	10.3	30.3	207000
…	…	.....	……	……..	…	….	……	…….	…….	……	…….	……	….	…	……..
….	…	.....	……	……..	…	….	……	…….	…….	……	…….	……	….	…	……..
….	…	.....	……	……..	…	….	……	…….	…….	……	…….	……	….	…	……..
160	17 y	female	august	town	5	102	7500	no	medium	low	high	low	11.4	29.6	195000
161	12 y	male	february	town	3	103	9500	no	no	medium	high	no	12.8	33.5	230000
162	18 y	male	march	town	4	102	13200	low	medium	no	no	no	13.5	38.3	320000
163	26 y	female	march	village	5	103	10600	no	low	no	high	medium	11.2	37.1	245000
164	24 y	female	march	village	5	102	11500	low	medium	no	no	low	12.5	34.7	275000
165	35 y	male	september	town	7	103	6300	low	medium	no	low	no	11.7	32.3	170000
168	20 y	male	july	town	3	105	8600	medium	low	low	medium	no	12.1	27.3	156000
169	25 y	male	february	village	6	103	12100	no	low	low	no	medium	12.9	39.5	310000
170	40 y	female	august	village	4	103	6700	high	no	low	no	low	10.6	23.5	187000
171	50 y	male	august	village	5	103	5800	no	high	low	medium	low	11.2	24.6	120000
172	22 y	female	december	town	3	102	9600	medium	no	no	low	no	12.6	35.2	250000

173	36 y	male	septemb er	village	4	10 3	4100	no	low	low	medium	no	11.6	34.2	172000
174	56 y	femal e	january	village	4	10 2	8300	no	low	mediu m	low	no	13.2	39.1	279000
175	60 y	male	august	village	3	10 3	4800	high	high	low	medium	low	8.7	30.4	87000
176	72 y	femal e	march	village	10	10 2	7800	no	low	mediu m	no	low	12.5	36.2	295000
177	65 y	femal e	july	town	6	10 3	6200	no	mediu m	low	high	no	10.5	28.4	202000
178	45 y	male	decembe r	village	4	10 1	9700	low	mediu m	low	medium	low	13	38.3	265000
179	32 y	male	july	village	5	10 4	6900	low	high	mediu m	low	high	10.1	32.2	170000
180	55 y	femal e	august	village	7	10 5	5100	no	low	low	medium	no	9.5	27.8	117000
181	40 y	male	august	town	5	10 3	7500	low	high	low	medium	low	11.3	36.2	165000
182	26 y	femal e	january	town	3	10 2	8700	no	low	low	high	no	13.2	41	310000
183	46 y	male	septemb er	village	4	10 3	7400	no	low	mediu m	high	mediu m	10.2	33.6	220000
183	18 y	femal e	july	village	3	10 2	8200	no	mediu m	low	high	no	11.4	31.3	135000
184	35 y	male	march	village	4	10 3	11600	no	low	mediu m	low	no	12.6	37.5	245000
185	32 y	male	august	town	5	10 2	9100	low	high	low	medium	no	9.8	32.5	126000
186	60 y	femal e	february	village	6	10 3	13600	no	low	low	low	no	12.3	34.6	272000

Last 7 parameters are shown in below:

Sample No.	Shock	Rash	Pleural effusion	Ascites	Bleeding	Slow heart rate	Igg	Igm	NS1	Travelling story	Target
1	no	no	no	no	no	no	negative	negative	negative	no	0
2	no	low	no	no	no	no	negative	negative	positive	no	1
3	no	medium	no	no	low	no	negative	negative	positive	no	1
4	no	low	no	no	no	no	positive	positive	negative	no	1
5	no	low	no	no	medium	no	positive	positive	negative	no	1
6	no	medium	no	no	no	no	negative	negative	positive	no	1
7	no	medium	no	no	no	no	negative	negative	positive	no	1
8	no	low	no	no	no	no	negative	negative	positive	no	1
9	no	low	no	low	no	yes	negative	negative	positive	no	1
10	no	medium	no	no	no	no	negative	negative	positive	no	1
11	no	medium	no	no	no	no	positive	positive	negative	yes	1
12	no	low	no	low	low	yes	positive	positive	negative	no	1
13	no	high	no	no	no	no	negative	negative	positive	no	1
14	no	medium	no	no	no	no	negative	negative	positive	no	1
15	no	mediu	no	no	no	no	negative	negative	positive	no	1

		m									
16	no	mediu m	no	no	no	no	negative	negative	positive	no	1
17	no	mediu m	no	no	no	no	negative	negative	positive	no	1
18	no	low	no	no	mediu m	no	negative	negative	positive	no	1
19	no	low	no	no	no	no	positive	positive	negative	no	1
20	no	low	no	no	no	no	positive	positive	negative	no	1
21	no	mediu m	no	no	no	yes	positive	positive	negative	no	1
22	no	low	no	no	no	yes	positive	positive	negative	no	1
23	no	mediu m	no	no	no	no	negative	negative	positive	no	1
24	no	mediu m	no	no	no	no	positive	positive	negative	no	1
25	no	no	no	no	no	no	negative	negative	negative	yes	0
26	no	no	no	no	no	no	negative	negative	negative	yes	0
27	no	low	no	no	mediu m	no	negative	negative	positive	no	1
…….	…… ….	……. ..	……	…… ..	……	……...	…… …	…… ..	…… …	…… …	……
…….	…… ….	……. ..	……	…… ..	……	……...	…… …	…… ..	…… …	…… …	……
…….	…… ….	……. ..	……	…… ..	……	……...	…… …	…… ..	…… …	…… …	……
…….	…… ….	……. ..	……	…… ..	……	……...	…… …	…… ..	…… …	…… …	……
160	no	low	no	no	no	no	positive	positive	negative	yes	1
161	no	no	no	no	no	no	negative	negative	negative	yes	0

162	no	no	no	no	no	no	negative	negative	negative	yes	0
163	no	no	no	no	low	no	negative	negative	negative	no	0
164	no	medium	no	no	high	no	positive	positive	negative	yes	1
165	no	no	no	no	low	no	negative	negative	negative	no	0
166	no	low	no	no	low	yes	negative	negative	positive	yes	1
167	no	low	no	no	no	no	negative	negative	positive	no	1
168	no	no	no	no	no	no	negative	negative	negative	yes	0
169	no	low	no	no	low	no	positive	positive	negative	yes	1
170	no	low	no	no	no	yes	positive	positive	negative	no	1
171	no	no	no	no	no	no	negative	negative	negative	no	0
172	no	high	no	no	no	no	positive	positive	negative	yes	1
173	no	no	no	no	no	no	negative	negative	negative	no	0
174	no	medium	no	yes	no	yes	negative	negative	positive	no	1
175	no	no	no	no	no	yes	negative	negative	negative	no	0
176	no	low	no	no	no	yes	positive	positive	negative	no	1
177	no	no	no	no	no	no	negative	negative	negative	yes	0
178	no	low	no	no	low	no	positive	positive	negative	yes	1
179	yes	medium	no	no	low	yes	positive	positive	negative	no	1
180	no	low	no	no	no	no	positive	positive	negative	yes	1
181	no	no	no	no	no	no	negative	negative	negative	yes	0
182	no	medium	no	no	no	no	positive	positive	negative	no	1
183	no	low	no	no	medium	no	negative	negative	positive	yes	1
184	no	no	no	no	no	no	negative	negative	negative	yes	0
185	no	low	no	no	low	no	positive	positive	negative	no	1

186	no	no	no	no	no	no	negative	negative	negative	yes	0

A-2 Preprocessed Data

Here, Days (no. of days (fever)), TP (Current temperature), Back pain (Back pain of eyes), Hb(Heamoglobin), Hct(Hematocrit), Plt(Platelets). 18 parameters are shown below among 25 parameters.

Sample no	Age	Sex	Month	Residence	Days	TP	WBC	Back pain	Joint pain	Muscle pain	Headache	Vomiting	Hb(g/dL)	Hct(%)	Plt(per cmm)
1	0.75	1	5	1	4	102	20380	3	3	3	3	1	9.7	29.6	302000
2	28	0	4	0	3	102	3200	3	3	2	0	3	11.5	34.7	198000
3	28	1	0	1	6	104	4600	3	1	0	0	2	15.3	44.8	48000
4	0.75	1	0	1	4	102	18500	3	3	3	3	0	11.2	34	296000
5	22	0	0	1	5	103	4500	3	0	0	1	1	10.5	32.3	165000
6	27	1	0	1	3	103	4200	3	2	1	3	3	12.6	37.2	150000
7	1	0	4	1	3	102	8300	3	3	3	3	1	9.5	28.7	120000
8	28	1	0	1	3	104	3400	1	3	2	0	3	14	39.9	152000
9	65	1	0	1	3	104	4200	3	2	0	2	2	12.1	36.3	115000
10	58	1	0	1	3	104	4700	3	3	0	2	3	12.4	36.5	156000
11	18	1	0	1	4	103	4300	2	2	0	1	3	13.9	41.1	148000
12	50	1	0	1	5	104	4600	0	3	0	2	2	12.5	36.6	95000
13	24	1	0	1	3	102	4500	3	0	0	3	3	11.5	34.3	165000
14	18	0	3	1	2	104	4200	1	1	0	0	1	14.4	42.2	155000

15	27	1	0	1	7	103	7800	3	2	0	0	3	12.3	35.2	150000
16	17	1	0	1	6	103	6200	1	3	0	0	1	14.7	41.2	110000
17	24	1	4	1	3	104	6200	3	1	0	2	3	13.1	34.9	160000
18	22	0	0	1	3	103	4300	0	2	0	1	2	11.1	33	175000
19	36	1	0	1	4	103	11400	3	1	2	0	3	13.4	39.3	300000
20	40	1	0	0	5	104	4500	0	1	0	1	3	12.4	37.5	183000
21	80	1	0	1	7	105	13700	3	2	0	3	1	10.3	30.3	207000
22	68	1	0	1	5	104	3700	0	3	1	2	1	12.3	32.3	170000
….	…	.....	…….	……..	…	….	…….	… …	…….	…….	…….	…….	….	… …	……..
….	…	.....	…….	……..	…	….	…….	… …	…….	…….	…….	…….	….	… …	……..
….	…	.....	…….	……..	…	….	…….	… …	…….	…….	…….	…….	….	… …	……..
163	26	0	5	1	5	103	10600	3	1	3	0	2	11.2	37.1	245000
164	35	0	0	1	5	102	7400	3	1	2	3	1	10.5	32.6	115000
165	24	0	5	1	5	102	11500	1	2	3	3	1	12.5	34.7	275000
166	35	1	6	0	7	103	6300	1	2	3	1	3	11.7	32.3	170000
167	20	1	4	0	3	105	8600	2	1	1	2	3	12.1	27.3	156000
168	25	1	2	1	6	103	12100	3	1	1	3	2	12.9	39.5	310000
169	40	0	0	1	4	103	6700	0	3	1	3	1	10.6	23.5	187000
170	50	1	0	1	5	103	5800	3	0	1	2	1	11.2	24.6	120000
171	22	0	1	0	3	102	9600	2	3	3	1	3	12.6	35.2	250000
172	36	1	6	1	4	103	4100	3	1	1	2	3	11.6	34.2	172000
173	56	0	3	1	4	102	8300	3	1	2	1	3	13.2	39.1	279000
174	60	1	0	1	3	103	4800	0	0	1	2	1	8.7	30.4	87000
175	72	0	5	1	10	102	7800	3	1	2	3	1	12.5	36.2	295000
176	65	0	4	0	6	103	6200	3	2	1	0	3	10.5	28.4	202000

177	45	1	1	1	4	101	9700	1	2	1	2	1	13	38.3	265000
178	32	1	4	1	5	104	6900	1	0	2	1	0	10.1	32.2	170000
179	55	0	0	1	7	105	5100	3	1	1	2	3	9.5	27.8	117000
180	40	1	0	0	5	103	7500	1	0	1	2	1	11.3	36.2	165000
181	26	0	3	0	3	102	8700	3	1	1	0	3	13.2	41	310000
182	46	1	6	1	4	103	7400	3	1	2	0	2	10.2	33.6	220000
183	19	0	4	1	3	102	8200	3	2	1	0	3	11.4	31.3	135000
184	34	1	5	1	4	103	11600	3	1	2	1	3	12.6	37.5	245000
185	31	1	0	0	5	102	9100	1	0	1	2	3	9.8	32.5	126000
186	59	0	2	1	6	103	13600	3	1	1	1	3	12.3	34.6	272000

Last 7 parameters are shown in below:

Sample No.	Shock	Rash	Pleural effusion	Ascites	Bleeding	Slow heart rate	Igg	Igm	NS1	Travelling story	Target
1	0	3	1	1	3	0	0	0	0	0	0
2	0	1	1	1	3	0	0	0	1	0	1
3	0	2	1	1	1	0	0	0	1	0	1
4	0	1	1	1	3	0	1	1	0	0	1
5	0	1	1	1	2	0	1	1	0	0	1
6	0	2	1	1	3	0	0	0	1	0	1
7	0	2	1	1	3	0	0	0	1	0	1
8	0	1	1	1	3	0	0	0	1	0	1
9	0	1	1	0	3	1	0	0	1	0	1
10	0	2	1	1	3	0	0	0	1	0	1
11	0	2	1	1	3	0	1	1	0	1	1

12	0	1	1	0	1	1	1	1	0	0	1
13	0	0	1	1	3	0	0	0	1	0	1
14	0	2	1	1	3	0	0	0	1	0	1
15	0	2	1	1	3	0	0	0	1	0	1
16	0	2	1	1	3	0	0	0	1	0	1
17	0	2	1	1	3	0	0	0	1	0	1
18	0	1	1	1	2	0	0	0	1	0	1
19	0	1	1	1	3	0	1	1	0	0	1
20	0	1	1	1	3	0	1	1	0	0	1
21	0	2	1	1	3	1	1	1	0	0	1
22	0	1	1	1	3	1	1	1	0	0	1
…….	…..	…	…	…… …	……	………..	…… …	…..	…..	…… …	……..
…….	…..	…	…	…… …	……	………..	…… …	…..	…..	…… …	……..
163	0	3	1	1	1	0	0	0	0	0	0
164	0	2	1	1	0	0	1	1	0	1	1
165	0	3	1	1	1	0	0	0	0	0	0
166	0	1	1	1	1	1	0	0	1	1	1
167	0	1	1	1	3	0	0	0	1	0	1
168	0	3	1	1	3	0	0	0	0	1	0
169	0	1	1	1	1	0	1	1	0	1	1
170	0	1	1	1	3	1	1	1	0	0	1
171	0	3	1	1	3	0	0	0	0	0	0
172	0	0	1	1	3	0	1	1	0	1	1
173	0	3	1	1	3	0	0	0	0	0	0
174	1	2	1	2	3	1	0	0	1	0	1

175	0	3	1	1	3	1	0	0	0	0	0
176	0	1	1	1	3	1	1	1	0	0	1
177	0	3	1	1	3	0	0	0	0	1	0
178	0	1	1	1	1	0	1	1	0	1	1
179	0	2	1	1	1	1	1	1	0	0	1
180	0	1	1	1	3	0	1	1	0	1	1
181	0	3	1	1	3	0	0	0	0	1	0
182	0	2	1	1	3	0	1	1	0	0	1
183	0	1	1	1	2	0	0	0	1	1	1
184	0	3	1	1	3	0	0	0	0	1	0
185	0	1	1	1	1	0	1	1	0	0	1
186	0	3	1	1	3	0	0	0	0	1	0

A-3 Normalized Dataset

Here, Days (no. of days (fever)), TP (Current temperature), Back pain (Back pain of eyes), Hb(Heamoglobin), Hct(Hematocrit), Plt(Platelets). 18 parameters are shown below among 25 parameters.

Sample no	**Age**	**Sex**	**Month**	**Residence**	**Days**	**TP**	**WBC**	**Back pain**	**Joint pain**	**Muscle pain**	**Headache**	**Vomiting**	**Hb (g/dL)**	**Hct (%)**	**Plt(per cmm)**
1	0.002015	1	0.833333	1	0.2	0.25	1	1	1	1	1	0.333333	0.151515	0.272321	0.613527
2	0.345171	0	0.666667	0	0.1	0.25	0	1	1	0.666667	0	1	0.424242	0.5	0.362319
3	0.3451	1	0	1	0.4	0.75	0.0814	1	0.3333	0	0	0.6666	1	0.9508	0

	71						9		33			67		93	
4	0.002015	1	0	1	0.2	0.25	0.89057	1	1	1	1	0	0.378788	0.46875	0.599034
5	0.269613	0	0	1	0.3	0.5	0.075669	1	0	0	0.333333	0.333333	0.272727	0.392857	0.282609
6	0.332578	1	0	1	0.1	0.5	0.058207	1	0.666667	0.333333	1	1	0.590909	0.611607	0.246377
7	0.005163	0	0.666667	1	0.1	0.25	0.296857	1	1	1	1	0.333333	0.121212	0.232143	0.173913
8	0.345171	1	0	1	0.1	0.75	0.011641	0.333333	1	0.666667	0	1	0.80303	0.732143	0.251208
9	0.811107	1	0	1	0.1	0.75	0.058207	1	0.666667	0	0.666667	0.666667	0.515152	0.571429	0.161836
….	……. …	… ….	….	……. …	……. …	…	……..	…..	…..	……	…… ….	……. …	……..	……	……..
….	……. …	… ….	….	……. …	……. …	…	……..	…..	…..	……	…… ….	……. …	……..	……	……..
….	……. …	… ….	….	……. …	……. …	…	……..	…..	…..	……	…… ….	……. …	……..	……	……..
175	0.899257	0	0.833333	1	0.8	0.25	0.267753	1	0.333333	0.666667	1	0.333333	0.575758	0.566964	0.596618
176	0.811107	0	0.666667	0	0.4	0.5	0.174622	1	0.666667	0.333333	0	1	0.272727	0.21875	0.371981
177	0.559249	1	0.166667	1	0.2	0	0.378347	0.333333	0.666667	0.333333	0.666667	0.333333	0.651515	0.660714	0.524155
178	0.395542	1	0.666667	1	0.3	0.75	0.215367	0.333333	0	0.666667	0.333333	0	0.212121	0.388393	0.294686
179	0.685178	0	0	1	0.5	1	0.110594	1	0.333333	0.333333	0.666667	1	0.121212	0.191964	0.166667

180	0.496285	1	0	0	0.3	0.5	0.250291	0.333333	0	0.333333	0.666667	0.333333	0.393939	0.566964	0.282609
181	0.319985	0	0.5	0	0.1	0.25	0.32014	1	0.333333	0.333333	0	1	0.681818	0.78125	0.63285
182	0.571842	1	1	1	0.2	0.5	0.24447	1	0.333333	0.666667	0	0.666667	0.227273	0.450893	0.415459
183	0.231835	0	0.666667	1	0.1	0.25	0.291036	1	0.666667	0.333333	0	1	0.409091	0.348214	0.210145
184	0.420728	1	0.833333	1	0.2	0.5	0.488941	1	0.333333	0.666667	0.333333	1	0.590909	0.625	0.475845
185	0.382949	1	0	0	0.3	0.25	0.343423	0.333333	0	0.333333	0.666667	1	0.166667	0.401786	0.188406
186	0.73555	0	0.333333	1	0.4	0.5	0.605355	1	0.333333	0.333333	0.333333	1	0.545455	0.495536	0.541063

Last 7 parameters are shown in below:

Sample no	**Shock**	**Rash**	**Pleural effusion**	**Ascites**	**Bleeding**	**Slow heart rate**	**Igg**	**Igm**	**NS1**	**Travelling story**	**Target**
1	0	1	1	1	1	0	0	0	0	0	0
2	0	0.333333	1	1	1	0	0	0	1	0	1
3	0	0.666667	1	1	0.333333	0	0	0	1	0	1
4	0	0.333333	1	1	1	0	1	1	0	0	1
5	0	0.333333	1	1	0.666667	0	1	1	0	0	1
6	0	0.666667	1	1	1	0	0	0	1	0	1

7	0	0.666667	1	1	1	0	0	0	1	0	1
8	0	0.333333	1	1	1	0	0	0	1	0	1
9	0	0.333333	1	0	1	1	0	0	1	0	1
10	0	0.666667	1	1	1	0	0	0	1	0	1
11	0	0.666667	1	1	1	0	1	1	0	1	1
12	0	0.333333	1	0	0.333333	1	1	1	0	0	1
13	0	0	1	1	1	0	0	0	1	0	1
14	0	0.666667	1	1	1	0	0	0	1	0	1
15	0	0.666667	1	1	1	0	0	0	1	0	1
16	0	0.666667	1	1	1	0	0	0	1	0	1
17	0	0.666667	1	1	1	0	0	0	1	0	1
18	0	0.333333	1	1	0.666667	0	0	0	1	0	1
19	0	0.333333	1	1	1	0	1	1	0	0	1
20	0	0.333333	1	1	1	0	1	1	0	0	1
…	…	…	…	…	…	…	….	…	….	…	…
…	…	…	…	…	…	…	….	…	….	…	…
…	…	…	…	…	…	…	….	…	….	…	…
…	…	…	…	…	…	…	….	…	….	…	…
165	0	1	1	1	0.333333	0	0	0	0	0	0
166	0	0.333333	1	1	0.333333	1	0	0	1	1	1
167	0	0.333333	1	1	1	0	0	0	1	0	1
168	0	1	1	1	1	0	0	0	0	1	0
169	0	0.333333	1	1	0.333333	0	1	1	0	1	1
170	0	0.333333	1	1	1	1	1	1	0	0	1
171	0	1	1	1	1	0	0	0	0	0	0
172	0	0	1	1	1	0	1	1	0	1	1
173	0	1	1	1	1	0	0	0	0	0	0

174	1	0.666667	1	2	1	1	0	0	1	0	1
175	0	1	1	1	1	1	0	0	0	0	0
176	0	0.333333	1	1	1	1	1	1	0	0	1
177	0	1	1	1	1	0	0	0	0	1	0
178	0	0.333333	1	1	0.333333	0	1	1	0	1	1
179	0	0.666667	1	1	0.333333	1	1	1	0	0	1
180	0	0.333333	1	1	1	0	1	1	0	1	1
181	0	1	1	1	1	0	0	0	0	1	0
182	0	0.666667	1	1	1	0	1	1	0	0	1
183	0	0.333333	1	1	0.666667	0	0	0	1	1	1
184	0	1	1	1	1	0	0	0	0	1	0
185	0	0.333333	1	1	0.333333	0	1	1	0	0	1
186	0	1	1	1	1	0	0	0	0	1	0

A-4 Unused 20 Samples

Here, Days (no. of days (fever)), TP (Current temperature), Back pain (Back pain of eyes), Hb(Heamoglobin), Hct(Hematocrit), Plt(Platelets).
18 parameters are shown below among 25 parameters.

Sample no	Age	Sex	Month	Residence	Days	TP	WBC	Back pain	Joint pain	Muscle pain	Headache	Vomiting	Hb (g/dl)	Hct (%)	Plt
1	30y	female	september	village	7	105	5800	low	high	low	high	medium	10.4	32.7	167000

2	25y	female	march	village	4	101	9300	no	low	no	medium	low	12.7	36.5	295000
3	12y	female	march	town	3	102	8700	no	low	no	medium	no	11.9	38.3	310000
4	8y	male	january	town	3	102	7300	no	no	no	medium	low	12	37.4	256000
5	42y	male	august	village	6	104	5200	low	medium	no	low	no	10.3	29.7	101000
6	16y	female	february	village	3	103	8500	no	medium	high	low	no	12.4	34.8	277000
7	21y	male	february	town	4	103	9600	low	no	high	medium	low	12.1	35.3	294000
8	12y	female	january	village	4	101	11200	no	no	medium	low	no	12.5	37.2	305000
9	18y	male	december	village	6	102	7600	low	no	no	medium	no	12.9	38.9	287000
10	7y	male	january	village	5	101	8400	low	no	no	no	medium	11.6	36.2	255000
11	64y	male	september	village	9	104	4300	medium	no	low	high	no	10.2	29.2	83000
12	37y	male	august	village	2	102	6200	no	medium	low	low	no	9.6	27.3	147000
13	3y	female	july	village	5	103	7100	no	no	no	no	medium	11.3	31.2	212000
14	26y	male	march	village	4	102	9500	no	medium	low	no	no	12.7	41.2	288000
15	40y	male	august	village	7	103	6300	medium	low	no	low	no	10.3	32.4	132000
16	16y	female	september	village	5	102	4200	medium	low	no	high	no	11.4	33	130000
17	58y	male	august	village	8	104	3700	no	no	low	high	medium	9.2	25.7	62000
18	23y	female	august	village	4	104	5300	high	low	medium	no	medium	12.2	33.4	129000
19	27y	male	february	village	7	101	8900	low	medium	no	low	no	11	36.6	260000
20	26y	male	july	town	12	102	4300	high	no	medium	high	medium	12.3	33.2	55000

Last 7 parameters are shown in below:

Sample no	Shock	Rash	Pleural effusion	Ascites	Bleeding	Slow heart rate	Igg	Igm	NS1	Travelling story	Target
1	no	low	no	no	medium	no	positive	positive	negative	no	1
2	no	no	no	no	low	no	negative	negative	negative	no	0
3	no	no	no	no	no	no	negative	negative	negative	no	0
4	no	no	no	no	no	no	negative	negative	negative	no	0
5	no	medium	low	no	low	no	positive	positive	negative	yes	1
6	no	no	no	no	no	no	negative	negative	negative	no	0
7	no	no	no	no	no	no	negative	negative	negative	no	0
8	no	no	no	no	low	no	negative	negative	negative	no	0
9	no	no	no	no	no	no	negative	negative	negative	no	0
10	no	no	no	no	low	no	negative	negative	negative	no	0
11	yes	medium	low	no	no	yes	positive	positive	negative	yes	1
12	no	no	no	no	no	no	negative	negative	positive	yes	1
13	no	high	no	no	no	no	positive	positive	negative	yes	1
14	no	no	no	no	no	no	negative	negative	negative	no	0
15	no	low	no	no	no	yes	positive	positive	negative	yes	1
16	no	low	no	no	medium	no	positive	positive	negative	yes	1
17	yes	low	low	no	no	no	positive	positive	negative	yes	1
18	no	low	no	no	medium	no	positive	positive	negative	yes	1
19	no	no	no	no	no	no	negative	negative	negative	no	0
20	no	low	no	no	medium	no	positive	positive	negative	yes	1