

Predicting the Risk of Infection with SCHISTOSOMA HAEMATOBIIUM using Machine Learning

Makolo U. Angela

Department of Computer Science, University of Ibadan, Ibadan, Nigeria and University of Ibadan Bioinformatics Research Group (UNIBReG)

Akinyemi M. Oluwatosin

Department of Computer Science, University of Ibadan, Ibadan, Nigeria and University of Ibadan Bioinformatics Research Group (UNIBReG)

ABSTRACT

Most developing countries of the world, particularly in sub-saharan Africa, exhibit high levels of morbidity and mortality associated with the disease Schistosomiasis caused by the parasite *Schistosoma Haematobium*. Most individuals at risk of *Schistosomiasis* reside between latitudes 36°N and 34°S where average fresh water temperatures range from 25° to 30°C, placing African states among the most affected countries. *Schistosoma- Mansoni* and *Schistosoma Haematobium* account for most *Schistosoma* species infection in Africa. In this research work, a computational approach was provided to estimate the prevalence of *Schistosoma Haematobium* using machine learning to predict if an individual is infected with *Schistosomiasis* as well as classify individuals into groups to determine their risk of infection.

The Neural Network machine learning approach was used which involves the use of supervised learning method in which for every set of input, there is an expected output or a set of expected outputs. The algorithm used for the model is almost the same as that used for a neural network single layer perceptron but with little variations in the transformation function applied on the inputs. The algorithm, which was implemented with Java programming language, involves assigning a weight or degree of importance to each input, finding the weighted sum of those input values, applying a transformation function on the weighted sum and adjusting the weight of each input. The inputs supplied to this model are the age, sex and location of a person.

Keywords

Machine Learning, *Schistosoma Haematobium*, Neural network, Supervised Learning.

1. INTRODUCTION

Schistosomiasis is a blood-dwelling parasitic worm that represents a serious health problem in tropical regions of the world, affecting approximately 200 million people. It's one of the most widespread and prevalent parasitic diseases worldwide. They can persist in the bloodstream for decades despite being permanently exposed to the host immune system, indicating that they must deploy a very effective system of immune evasion [12]. *Schistosoma- Haematobium* is a species of *Schistosomiasis* that causes urinary Schistosomiasis disease characterized by blood in the urine (*Haematuria*), hence the infection is often referred to as 'UrinarySchistosomiasis'. Cancer of the bladder is an important complication of infection with *Schistosoma Haematobium* [13].

The apparent seriousness and prevalence of Schistosomiasis especially in sub-saharan Africa calls for measures aimed at managing the scourge. In order to effectively combat this disease, knowledge of the prevalence among other

characteristics will be instrumental to effective interventions. Machine learning is one of the many computational approaches employed in the study of diseases.

Learning, like intelligence, covers such a broad range of processes that it is difficult to define precisely [14]. According to [15], learning is the act of acquiring knowledge or developing the ability to perform new behaviors. Just as humans learn, machines over time have been developed to learn from past experience in order to optimize their future performance. A system is thus said to have learnt if it changes its structure and program in such a way that the expected future performance is improved.

To solve a problem on a computer, we need an algorithm which is a step by step set of instructions that should be carried out to transform an input to output. For the same task, there may be various algorithms and we may be interested in finding the most efficient one, or the one with the optimal solution in terms of time and space. For some tasks, however, we don't have an algorithm at all. For example, to distinguish spam emails from legitimate emails. We know what the input is an email document that in the simplest case is a file of characters. We know what the output should be - a yes/no. We do not know how to transform the input to the output. What can be considered spam changes in time and from individual to individual. What we lack here is knowledge, and we can make up for this lack of knowledge with data. Therefore, we would like the computer machine to extract automatically the algorithm for this task. There is no need to learn to add numbers, we already have algorithms for that; but there are many applications for which we do not have an algorithm but do have example data [16].

The aim of this research work is to provide a model that predicts the prevalence of *Schistosoma Haematobium* which in a way will help to reduce the spread of or even totally eradicate this deadly parasite. The introduction of machine learning to the study of *Schistosoma- Haematobium* will provide a better understanding of the prevalence of the disease.

2. RELATED WORKS

There are many successful applications of Neural Networks used in various research endeavours. Neural Network finds useful application in the prediction and diagnosis of different diseases as they are basically used to improve the objectivity of medical diagnosis.

[17] carried out research that performs a heart disease diagnosis using artificial neural network. Feed-forward back propagation neural network was used as a classifier to distinguish between absence and presence of disease. The data used was obtained from UCI machine learning repository in order to diagnose the disease. The result from

the research showed that the network was able to classify 88% of the cases in the testing set.

In [18], a system was designed that gives the diagnosis of skin disease using neural network. The data used consisted of Patient vital signs, Patient verbal complaints, Patient demographics and Presence of specific symptoms. The data was obtained from the Olivet Clinic, Port Harcourt and from the National Skin Center for Dermatology. The architecture of ANN selected for the skin diseases was based on the feed - forward network. The system recorded a success rate of 90% when suspected and diagnosed based symptoms were supplied to the system.

In the research carried out in [19], an evaluation of artificial neural network in urinary diseases diagnosis was presented. Feed-forward back propagation neural network was used as a classifier to distinguish between infected or non-infected. The two types of urinary disease considered are Inflammation of urinary bladder and nephritis of renal pelvis. The data used for the research was obtained from UCI Machine Learning Repository [20]. The targets for the neural network were identified with 1's as infected and will be identified with 0's as non-infected. The research work recorded a success rate of 99% and thus showed that neural network could be useful for identifying an infected person.

[21] conducted a research on the application of neural networks in diagnosing cancer disease using demographic data. In the network, the input neuron values are the demographic data such as patient's age, sex etc. The data used was obtained from various hospitals. MATLAB was used to train and test the data. The result of the research work shows that neural network can effectively be used for lung cancer diagnosis.

[22] carried out a research work that diagnoses kidney stone disease using three different neural network algorithms. The three algorithms are learning vector quantization (LVQ), two layers feed forward *perceptron* trained with back propagation training algorithm and Radial basis function (RBF) networks. Waikato Environment for Knowledge Analysis (WEKA) version 3.7.5 was used as simulation tool. The data set used for diagnosis is real world data with 1000 instances and 8 attributes. The result of the research work showed that the multilayer *perceptron* and back propagation algorithm is the best model for diagnosis of kidney stone disease. It has an accuracy of 92%.

Considering the various applications of neural networks as applied to diagnoses of diseases neural networks have over time become an important tool used in the prediction of infection of various diseases showing high degree of accuracy. The degree of accuracy of these research shows that it can be used for medical diagnosis of other life threatening ailments.

3. EXPERIMENTAL EXPERIENCE AND DISCUSSION OF RESULTS

Neural networks, take their inspiration from the brain. The brain is an information processing device that has some incredible abilities and surpasses current engineering products in many domains—for example, vision, speech recognition, and learning, to name three. These applications have evident economic utility if implemented on machines. If we can understand how the brain performs these functions, we can define solutions to these tasks as and implement them on computers. [11]

The human brain is quite different from a computer. Whereas a computer generally has one or few number of processors, the brain is composed of a very large (10^{11}) number of processing units known as *neurons*, operating in parallel. Though the details are not known, the processing units are believed to be much simpler and slower than a processor in a computer. What also makes the brain different, and is believed to provide its computational power, is the large connectivity. Neurons in the brain have connections, called *synapses*, to around 104 other neurons, all operating in parallel. In a computer, the processor is active and the memory is separate and passive, but it is believed that in the brain, both the processing and memory are distributed together over the network; processing is done by the neurons, and the memory is in the synapses between the neurons.

3.1 Neural network algorithm

The *perceptron* is an algorithm for supervised classification of an input into one of several possible non-binary outputs [24][25]. It is a type of linear classifier, i.e. a classification algorithm that makes its predictions based on a linear predictor function combining a set of weights with the feature vector describing a given input using the delta rule. A *perceptron* is a program that learns concepts, i.e. it can learn to respond with True or false based on the input presented to it by repeatedly studying examples. The *perceptron* is a binary classifier which maps its input χ (a real-valued vector) to an output value $f(\chi)$ (a single binary value):

$$f(\chi) = \left. \begin{array}{ll} 1 & \text{if } \omega \cdot \chi + b > 0 \\ 0 & \text{otherwise} \end{array} \right\} \dots(1)$$

where ω is a vector of real-valued weights, $\omega \cdot \chi$ is the dot product (which here computes a weighted sum), and b is the 'bias', a term that does not depend on any input value[26]. A bias value enables you to shift the activation function to the left or right, which may be critical for successful learning. A *perceptron* has just two layers of nodes (input nodes and output nodes) [12]. The *perceptron* can be trained by adjusting the weights of the inputs with Supervised Learning. In this learning technique, the patterns to be recognized are known in advance, and a training set of input values are already classified with the desired output. The actual inputs are supplied and the weight of each input is supplied alongside. If the output is correct, no weights are altered. However, if the output is wrong, we have to distinguish which of the patterns we would like the result to be, and adjust the weights on the currently active inputs towards the desired result. Another value used with a Neural network is called the learning Rate. A learning rate is user-designated in order to determine how much the link weights and node biases can be modified based on the change direction and change rate. The higher the learning rate (max. of 1.0) the faster the network is trained.

The algorithm used here is similar to that of a *perceptron* but with little adjustments. The single layer feed forward network consist of a single layer of weights, where the inputs are directly connected to the outputs via a series of weights. The synaptic link carrying weights connects every input to every output but not vice versa. The steps for the algorithm are as follows:

1. Initialize the weights and the threshold. Weights age is initialized to 0.01; sex is initialized to 0.1 and location to 0.2.
2. For each example j in our training set D , perform the following steps over the input x_j and desired output d_j :
Calculate the actual output which is obtained by applying a function on the weighted sum 'I'. 'I' is given by

$$I = \sum_{i=1}^3 \chi_i + \psi_i + b \quad \dots (2)$$

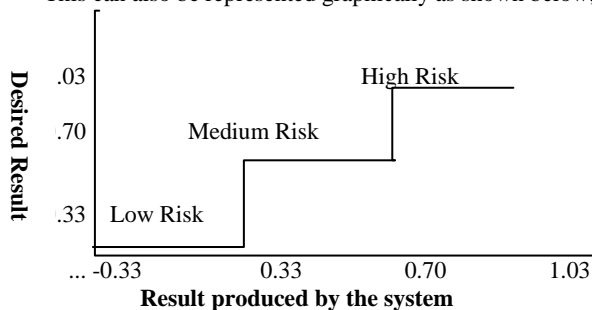
$$I = [(x1 * w1) + b] + [(x2 * w2) + b] + [(x3 * w3) + b] \quad (3)$$

Where $x1$ represents the first input 'age'
 $w1$ represents the weight assigned to 'age'
 $x2$ represents the second input 'sex'
 $w2$ represents the weight assigned to 'sex'
 $x3$ represents the third input 'location'
 $w3$ represents the weight assigned to 'location'
 b represents bias which is assumed to be 0 and
 I represent the weighted sum.

The actual or expected output becomes $f(I)$
 The function is given by:

$$f(I) = \left\{ \begin{array}{l} 1.03 \text{ (High risk) if } 0.71 \leq I \leq 1.03 \\ 0.70 \text{ (Medium risk) if } 0.34 \leq I \leq 0.70 \\ 0.33 \text{ (Low Risk) if } I \leq 0.33 \end{array} \right\} \dots (4)$$

This can also be represented graphically as shown below;



GRAPH 1: graphical representation of the function in equation 4

Save the set in a matrix or array or database

3. Accept $x1, x2, x3$ from the user
4. Apply the function on the data set supplied by the user i.e.

$$f(I) = f[(b+(x1 * w1)) + (b+(x2 * w2)) + (b+(x3 * w3))] = P(o) \quad \dots (5)$$

5. Compute the error factor which is obtained as:
Error factor = Expected output[E(o)] – Produced output[P(o)]
6. Calculate the error or delta using the formula:
delta = Produced output * (1 - Produced output) * errorFactor
7. The weights are adjusted using the following formula:
w1 = $w1 + (\text{learningRate} * 1 * \text{Produced output} * \text{delta})$;

$$w2 = w2 + (\text{learningRate} * 1 * \text{Produced output} * \text{delta});$$

$$w3 = w3 + (\text{learningRate} * 1 * \text{Produced output} * \text{delta});$$

8. Go to step 4, and repeat 100 times, 100 iterations are chosen because at this number of iteration, it is sure that the machine has learned.

The **threshold** for this neural network is 1.03. This means that if an output value exceeds 1.03, the network does not shoot (or produce an output). The **bias (b)** is assumed to be 0 (a constant term that does not depend on any input value).

3.2 Application of the algorithm on the main program

This explains how the algorithm described above will be implemented. Since neural network and its functions works basically on numbers, several assumptions were made. These assumptions are:

1. Ages 6- 25 represents the value '3,' since they stand a higher risk of being infected.
2. Ages 26- 50 represents the value '2' since they are next in line to the age range described previously.
3. Ages 51- 75 represents the value '1' since they are less susceptible to the disease.
4. Ages 1-5 and ages 76 and above represents the value '0' since they are less susceptible to the disease.
5. Males are assigned value '2' since they stand a higher risk of being infected.
6. Females are assigned value '1' since they are less likely to be infected.
7. Countries in Africa are assigned '4' since they stand a higher risk of being infected.
8. Countries in Asia are assigned '3' since they are next in line to Africa in terms of the prevalence of *Schistosoma Haematobium*.
9. Countries in South America are assigned '2' since they are next in line to Asia in terms of the prevalence of *Schistosoma Haematobium*.
10. Countries in North America South America, Europe, Australia and Antarctica are assigned '1' since people in this region stand a very low chance of being infected with *Schistosoma- Haematobium* [12][13][14].

The training set is obtained using the brute force method i.e. it computes the actual output for all possible combinations. This means that the weighted input 'I' has to be computed first, before the transformation function is applied to it. Computing the weighted sum is achieved using the formula in equation (2)

$$I = (b+(x1 * w1)) + (b+(x2 * w2)) + (b+(x3 * w3)) \quad (6)$$

where :-

$w1$ is assumed to be 0.01

$w2$ is assumed to be 0.1

$w3$ is assumed to be 0.2

b is assumed to be 0

While the actual or expected output $E(0)$ is computed by applying the function in equation (4)

Hence, the training set table is given below

Table 1: Database representation of the training set

Age(x1*w1)	Sex(x2*w2)	Location(x3*w3)	Output in value (I)	Evaluation f(I)
3 * 0.01	1 * 0.1	1 * 0.2	0.33	Low risk
3 * 0.01	1 * 0.1	2 * 0.2	0.53	Average Risk
3 * 0.01	1 * 0.1	3 * 0.2	0.73	High Risk
3 * 0.01	1 * 0.1	4 * 0.2	0.93	High Risk
3 * 0.01	2 * 0.1	1 * 0.2	0.43	Average Risk
3 * 0.01	2 * 0.1	2 * 0.2	0.63	Average Risk
3 * 0.01	2 * 0.1	3 * 0.2	0.83	High Risk
3 * 0.01	2 * 0.1	4 * 0.2	1.03	High Risk
2 * 0.01	1 * 0.1	1 * 0.2	0.32	Low risk
2 * 0.01	1 * 0.1	2 * 0.2	0.52	Average Risk
2 * 0.01	1 * 0.1	3 * 0.2	0.72	High Risk
2 * 0.01	1 * 0.1	4 * 0.2	0.92	High Risk
2 * 0.01	2 * 0.1	1 * 0.2	0.42	Average Risk
2 * 0.01	2 * 0.1	2 * 0.2	0.62	Average Risk
2 * 0.01	2 * 0.1	3 * 0.2	0.82	High Risk
2 * 0.01	2 * 0.1	4 * 0.2	1.02	High Risk
1 * 0.01	1 * 0.1	1 * 0.2	0.31	Low risk
1 * 0.01	1 * 0.1	2 * 0.2	0.51	Average Risk
1 * 0.01	1 * 0.1	3 * 0.2	0.71	High Risk
1 * 0.01	1 * 0.1	4 * 0.2	0.91	High Risk
1 * 0.01	2 * 0.1	1 * 0.2	0.41	Average Risk
1 * 0.01	2 * 0.1	2 * 0.2	0.61	Average Risk
1 * 0.01	2 * 0.1	3 * 0.2	0.81	High Risk
1 * 0.01	2 * 0.1	4 * 0.2	1.01	High Risk
0 * 0.01	1 * 0.1	1 * 0.2	0.30	Low Risk
0 * 0.01	1 * 0.1	2 * 0.2	0.50	Average Risk
0 * 0.01	1 * 0.1	3 * 0.2	0.70	Average Risk
0 * 0.01	1 * 0.1	4 * 0.2	0.90	High Risk
0 * 0.01	2 * 0.1	1 * 0.2	0.40	Average Risk
0 * 0.01	2 * 0.1	2 * 0.2	0.60	Average Risk
0 * 0.01	2 * 0.1	3 * 0.2	0.80	High Risk
0 * 0.01	2 * 0.1	4 * 0.2	1.00	High Risk

Table 2: Interpretation of table 1

Age Range	Sex	Location	Output in value (I)	Evaluation f(I)
0- 25	Female	Other continents	0.33	Low risk
0- 25	Female	South America	0.53	Average Risk
0- 25	Female	Asia	0.73	High Risk
0- 25	Female	Africa	0.93	High Risk
0- 25	Male	Other continents	0.43	Average Risk
0- 25	Male	South America	0.63	Average Risk
0- 25	Male	Asia	0.83	High Risk
0- 25	Male	Africa	1.03	High Risk
26-50	Male	Other continents	0.32	Low risk
26-50	Female	South America	0.52	Average Risk
26-50	Female	Asia	0.72	High Risk
26-50	Female	Africa	0.92	High Risk
26-50	Male	Other continents	0.42	Average Risk
26-50	Male	South America	0.62	Average Risk
26-50	Male	Asia	0.82	High Risk
26-50	Male	Africa	1.02	High Risk
51-75	Female	Other continents	0.31	Low risk
51-75	Female	South America	0.51	Average Risk
51-75	Female	Asia	0.71	High Risk
51-75	Female	Africa	0.91	High Risk

51-75	Male	Other continents	0.41	Average Risk
51-75	Male	South America	0.61	Average Risk
51-75	Male	Asia	0.81	High Risk
51-75	Male	Africa	1.01	High Risk
76-100	Female	Other continents	0.30	Low Risk
76-100	Female	South America	0.50	Average Risk
76-100	Female	Asia	0.70	Average Risk
76-100	Female	Africa	0.90	High Risk
76-100	Male	Other continents	0.40	Average Risk
76-100	Male	South America	0.60	Average Risk
76-100	Male	Asia	0.80	High Risk
76-100	Male	Africa	1.00	High Risk

Table 1 describes how the training set is represented in the database based on the assumptions given above. The table 8 tries to interpret the table by replacing the figures with values that can be easily understood.

The produced output P(o) is also computed by first computing the weighted sum before applying the function on the weighted sum.

$$I = (x1 * w1) + (x2 * w2) + (x3 * w3) \quad \dots(7)$$

where

w1 is the weight of the first input supplied by the user

w2 is the weight of the second input supplied by the user

w3 is the weight of the third input supplied by the user

While the output P(o) is computed by applying the function in equation 4.

The error factor is calculated as follows:

$$\text{Error factor} = \text{Expected output}[E(o)] - \text{Produced output}[P(o)]$$

Calculate the error or delta using the formula-

$$\text{delta} = \text{Produced output} * (1 - \text{Produced output}) * \text{errorFactor}$$

The weights are then adjusted using the following formula

$$w1 = w1 + (\text{learningRate} * 1 * \text{Produced output} * \text{delta});$$

$$w2 = w2 + (\text{learningRate} * 1 * \text{Produced output} * \text{delta});$$

$$w3 = w3 + (\text{learningRate} * 1 * \text{Produced output} * \text{delta});$$

The weighted sum, the produced output, the error factor the error and new sets of weights are recalculated 100 times. 100 iterations are chosen because at this number of iteration, it is sure that the machine has learned.

3.3 Classification phase

Another use of a neural network is that it can be used for classification. This Proposed system will be able to perform the function of classification. After a set of data is entered by the user (e.g age, sex and location data), the system should be able to provide information about the set of data entered so far and classify or group them into a set of people with high risk, people with low risk and people with average risk of infection. This will be achieved through the following steps.

1. The system computes the output of each age, sex and location entered by the user. The user enters the age, sex and location of two or more people.
2. The system saves the details of the inputs and the output entered by the user in a list or storage such as a database.
3. The system queries the database in such a way that the data is classified and the classified set of data is displayed to the user.

3.4 Structure of the neural network

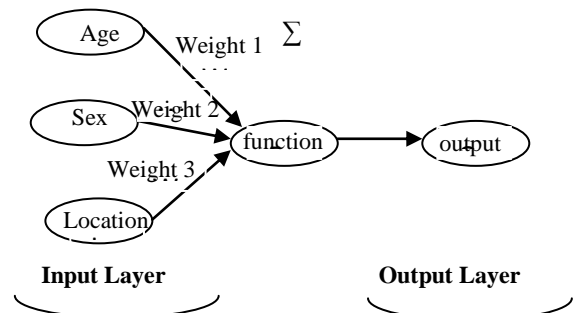
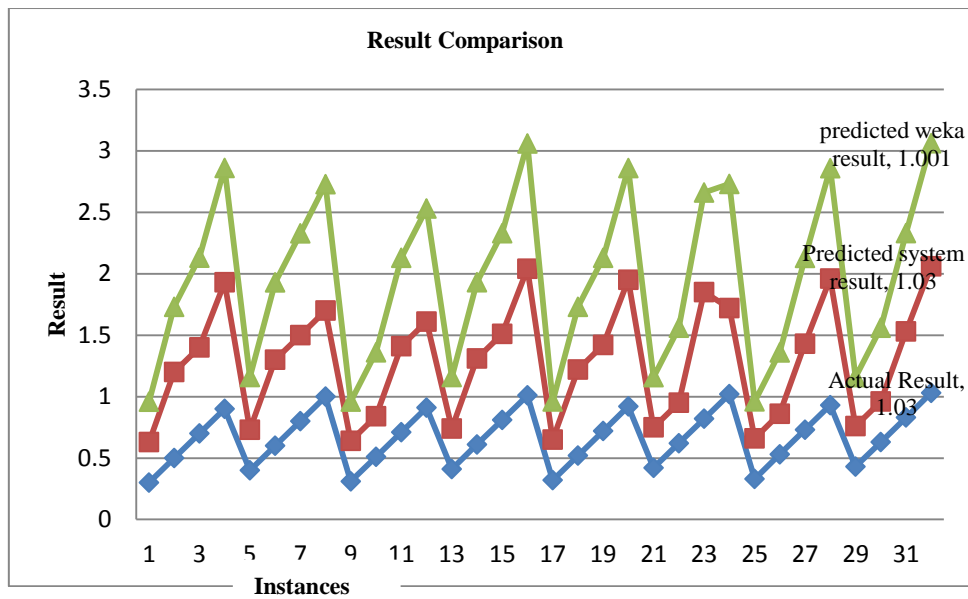


Figure 1: Neural network model

This network consists of two layers; the input layer and the output layer. The input layer is the input values supplied by the user in terms of the age of a person, the sex of a person and the region where the person lives. The system automatically generates weights for the input. A transformation function is applied on the weighted sum of the input to produce an output. The output gives a prediction of if the person stands a high or low risk of being infected.

3.4 Result

The model was tested with test data obtained from UCI machine learning repository [23]. The result produced by the system was compared with that produced with WEKA. It was discovered that the result produced by the system was similar to that produced by WEKA.



Graph 2: Comparison between the expected result, they system result and WEKA result

The x-axis represents the number of instances while the y-axis represents the result produced by both WEKA and the system developed. The blue line represents the expected result as obtained from the training set; the red line represent the result predicted by the system while the green line represents the result as predicted by WEKA. The predicted system result generated a line that is much closer to the actual or expected result as shown in graph 2. The result from the graph shows that designed system produces result that is

much closer to the actual result than those produced by WEKA.

Table 3 shows that the result produced by the system is much closer to the expected result than the result produced by WEKA. The root mean square value produced by the system (0.022248595) is also smaller than that produced by WEKA. This indicates that the system performed better than WEKA.

Table 3: Comparison of the result produced by both the designed system and WEKA

Actual/Expected Result	Predicted weka result	predicted system result	Square weka error (D3^2)	Square system error (E3^2)
0.3	0.33	0.328	0.0009	0.000784
0.5	0.7	0.53	0.04	0.0009
0.7	0.7	0.73	0	0.0009
0.9	1.03	0.93	0.0169	0.0009
0.4	0.33	0.429	0.0049	0.000841
0.6	0.7	0.629	0.01	0.000841
0.8	0.7	0.828	0.01	0.000784
1	0.7	1.028	0.09	0.000784
0.31	0.33	0.319	0.0004	8.1E-05
0.51	0.33	0.52	0.0324	0.0001
0.71	0.7	0.719	0.0001	8.1E-05
0.91	0.7	0.92	0.0441	0.0001
0.41	0.33	0.419	0.0064	8.1E-05
0.61	0.7	0.62	0.0081	0.0001
0.81	0.7	0.819	0.0121	8.1E-05
1.01	1.03	1.019	0.0004	8.1E-05
0.32	0.33	0.31	0.0001	0.0001
0.52	0.7	0.51	0.0324	0.0001

0.72	0.7	0.709	0.0004	0.000121
0.92	1.03	0.909	0.0121	0.000121
0.42	0.33	0.408	0.0081	0.000144
0.62	0.33	0.61	0.0841	0.0001
0.82	1.03	0.81	0.0441	1E-04
1.02	0.7	1.01	0.1024	0.0001
0.33	0.33	0.301	0	0.000841
0.53	0.33	0.499	0.04	0.000961
0.73	0.7	0.698	0.0009	0.001024
0.93	1.03	0.898	0.01	0.001024
0.43	0.33	0.398	0.01	0.001024
0.63	0.33	0.6	0.09	0.0009
0.83	0.7	0.8	0.0169	0.0009
1.03	1.03	1.001	0	0.000841
		Root Mean square error	0.150851748	0.022248595

4. CONCLUSION AND FUTURE WORK

4.1 Conclusion

This research work provides the tool necessary for the biologists or medical scientist to predict the likelihood of an individual being infected with *Schistosoma Haematobium*. The research was carried out using Neural Networks and was implemented using a programming language called JAVA. The designed system produced a more accurate result than the result produced by WEKA.

4.2 Future work

The neural network model may be extended to include other criteria aside from age, sex and location. It may also be extended to include a dynamic number of criteria or input.

Other machine learning models may be exploited on the study of *Schistosoma Haematobium* and conclusions may be drawn on the most effective machine learning algorithm for the prediction of *Schistosoma Haematobium* infection.

5. REFERENCES

- [1] Strickland GT, Ramirez BL. *Schistosomiasis*. In: Strickland GT, editor. *Hunter's Tropical Medicine and Emerging Infectious Diseases*. Vol. 118. Philadelphia: W.B. Saunders Company, 8th edition; 2000. pp. 804–832.
- [2] *Schistosoma Haematobium*, pg 372,375. Retrieved September 02, 2015 from <http://monographs.iarc.fr/ENG/Monographs/vol1100B/mono100B-14.pdf>
- [3] *Schistosoma Haematobium*, pg 372,375. Retrieved September 02, 2015 from http://www.neglecteddiseases.gov/target_diseases/schistosomiasis/
- [4] J. Nielsen M. Farooq et al. The epidemiology of *Schistosoma haematobium* and *S. mansoni* infections in the Egypt-49 project area. Available at:<http://www.ncbi.nlm.nih.gov/pmc/articles/PMC2476086/>
- [5] The Carter Center. "How is *Schistosomiasis* Treated?". Archived from the original on 2008-02-25. Retrieved 2008-07-17.
- [6] The Carter Center. "*Schistosomiasis* Control Program". Retrieved 2008-07-17
- [7] The Carter Center. "*Schistosomiasis* Control Program". Retrieved 2008-07-17
- [8] A ruler killer, Retrieved 9th January, 2016 from <http://www.end.org/whatwedo/ntdoverview/schistosomiasis>
- [9] Thomas Dietterich et al. (2010). *Introduction to Machine learning*, second edition, 2010, pg 1-2.
- [10] Nils J. Nilsson. *Introduction to Machine Learning*. Robotics Laboratory, Department of Computer Science, Stanford University, Stanford, CA 94305.
- [11] Witten, I. H., and E. Frank. 2005. *Data Mining: Practical Machine Learning Tools*
- [12] Variation in blood- dwelling *Schistosome* worms. Retrieved February 06, 2014 from <http://www.ncbi.nlm.nih.gov/pmc/articles/PMC290957/>
- [13] *Schistosomiasis* Research Group. General Pathology Associated with *Schistosome* Infections. Retrieved February 06, 2014 from http://www.path.cam.ac.uk/~schisto//schistosoma/schisto_pathology.html
- [14] Perceptron. Retrieved February 06, 2014 from <http://en.wikipedia.org/wiki/Perceptron>
- [15] Nils J. Nilsson. (2005) *Introduction to machine learning*. Pg 1.

- [16] Learning [text]. (2009). In Microsoft Encarta (version 2.1) [software]. Redmond, WA: Microsoft Corporation.
- [17] Noura Ajam, Heart Diseases Diagnoses using Artificial Neural Network, Network and Complex Systems , ISSN 2224-610X (Paper) ISSN 2225-0603 (Online), Vol.5, No.4, 2015
- [18] Bakpo, F. S.and Kabari, L. G, Diagnosing Skin Diseases Using an Artificial Neural Network. Available at: <http://cdn.intechopen.com/pdfs-wm/14893.pdf>
- [19] Qeethara Kadhim Al-Shayea† and Itedal S. H. Bahia, Urinary System Diseases Diagnosis Using Artificial Neural Networks, IJCSNS International Journal of Computer Science and Network Security, VOL.10 No.7, July 2010
- [20] Dr. N. Ganesan et al, Application of Neural Networks in Diagnosing Cancer Disease Using Demographic Data, International Journal of Computer Applications (0975 - 8887), Volume 1 – No. 26.
- [21] Koushal Kumar and Abhishek, Artificial Neural Networks for Diagnosis of Kidney Stones Disease, I.J. Information Technology and Computer Science,2012, 7, 20-25, Published Online July, 2012 in MECS (<http://www.mecspress.org/>)
- [22] UCI Machine learning repository, Dataset number: 577466(Adult),<http://archive.ics.uci.edu/ml/datasets/Adult>
- [23] Prashan Premaratne, Human Computer Interaction Using Hand Gestures, pg 115.
- [24] Stephen M. Watt et al, Intelligent Computer Mathematics: CICM 2014 Joint Events: Calculemus, DML, MKM, and Systems and Projects 2014, Coimbra, Portugal, July 7-11, 2014. Proceedings.
- [25] Anestis Gkanogiannis and Theodore Kalamboukis, A modified and fast Perceptron learning rule and its use for Tag Recommendations in Social Bookmarking Systems, Department of Informatics Athens University of Economics and Business, Athens, Greece.