International Journal of Engineering & Technology, 7 (3.34) (2018) 480-486

International Journal of Engineering & Technology

Website: www.sciencepubco.com/index.php/IJET

Research paper

An Improved and Adaptive Attribute Selection Technique to Optimize Dengue Fever Prediction

Sushruta Mishra*, Hrudaya Kumar Tripathy, Amiya Ranjan Panda

School of Computer Science and Engineering, KIIT University, Bhubaneswar, India
**Corresponding author Email: mishra.sushruta@gmail.com*

Abstract

Clinical information mining is rapidly gaining popularity. Restorative information are high dimensional in nature which contains unessential elements that diminish prediction capability. Hence Attribute Optimization is required to retain only the essential features while eradicating irrelevant features. Dengue is one of the major worldwide medical related disease. It has affected millions of people throughout world while a majority of them being women. With constant upgradation of information technology and its application in healthcare domain, several cases relating to diabetes along with its symptoms are properly documented. Our study is centered on developing and implementing a new Adaptive and Dynamic Attribute Optimization algorithm to determine whether patients suffer from Dengue. Our algorithm is evaluated against some vital performance metrics and compared with other sub-modules of the proposed algorithm and traditional Genetic Algorithm. The results indicate our algorithm is more efficient and accurate in determining presence of Dengue disease. This may assist the medical experts in effective diagnosis of patients suffering from Dengue.

Keywords: *Dengue; Data Mining; Attribute Optimization; Genetic Algorithm, Fitness Function;*

1. Introduction

Dengue is a dangerous disease having mortal problems. Dengue fever occurs when an Aedes mosquito infected by a dengue virus bites a human being. Frequent headache, fatigue, rashes in skin, joint and muscle pain and retro-orbital pain are some of the common symptoms of dengue fever [2]. It is also known as bone-breaking illness [3]. Dengue fever infection has affected some billion populations throughout world. Around the globe, about 50 million people suffer from dengue fever every year [1]. Symptoms of dengue are seen on fourth or sixth day from the day it is infected and usually last for around ten days. Patients suffer from frequent headaches. At the starting phases it is tough to distinguish between dengue hemorrhagic fever and dengue fever. Proper medical diagnosis for dengue disease does not exists. Data mining algorithms may be applied to prediction the disease risks of dengue fever.

2. Problem Statement

Medical Data Mining is seen as an effective approach to detect Dengue infection risk disorders turning out to be more viable and early recognition of sickness that help in better diagnosis of patients. A critical constituent of Data Mining includes Attribute Optimization. It reduces the data samples to be investigated without losing any relevant data, enhance the quality of information, upgrade the execution of data mining computations and lessen the execution time of mining calculations [4]. It acts as an optimizing agent thereby eliminating less relevant features from the dataset which further enhances the classification efficiency and performance. Our main research work involves defining and implementing a new improved and dynamic technique of Attribute Optimization based on Genetic Search algorithm on Dengue fever Dataset which selects only relevant symptoms which may be used for classification of patient’s data. Our focus is to implement our new algorithm to efficiently predict presence of Dengue disease in patients with enhanced accuracy.

3. Dengue Fever Dataset Details

The basic objective of our research study is to improve the prediction accuracy of dengue disease with the symptoms being provided. To achieve this, dengue dataset is taken which comprises several attributes along with the result (presence of dengue or not). The dengue data samples consists of many possible symptoms such as Fever temperature, eyes pain, rashes in skin, frequent headaches, fatigue etc. this dataset is developed from discussion with patients suffering from high temperature. This dataset is collected from 100 patients and is saved permanently in csv format. Table 1 depicts the details of dengue attributes.

Table 1: Dengue dataset details

Attributes	Possible Values
Fever Period	3 Days, 4 Days, 5 Days
Temperature in Fever	100°C, 102°C, 104°C
Dark Red Spots	Yes / No
Eyes Pain	Yes / No
Headache	Yes / No
Joint and Muscle Pain	Yes / No
Vomiting or Nausea	Yes / No
Reduced Heart Rate	Yes / No
Fatigue	Yes / No
Outcome	Dengue / No Dengue

4. Related Works

Various research works based on the implementation of data mining algorithms have been undertaken by many researchers. Many classification techniques like Naïve Bayes, KNN algorithm, Multi layer Perceptron and Support vector machine have been used [5][6][7]. These algorithms were evaluated against some performance parameters like classification accuracy, precision, recall, sensitivity etc. Many researchers like Tanner et al. and Tarig et al had carried out important works on classification of dengue fever disease. Tanner and his friends in [8] implemented Decision tree algorithm for classification of dengue disease dataset with six attributes and 1200 patients. They derived a classification accuracy of 84%. Tarig's team in [9] employed multi layer feed forward networks with Self Organizing MAP to cluster patients into two partitions and obtained 70% accuracy rate. [10] Fatimah Ibrahim et.al applied multi layer Perceptron in dengue dataset and got 90% prediction accuracy. Daranee et al. [11] proposed classification of dengue patients into two partitions. Accuracy of 97.6% and 96.6% was the output of the first and second set respectively. The rate of accuracy in testing set was beyond 90%. Wajeeha Farooqi et al. [12] presented data mining classification of dengue fever using decision tree and performed two different demonstrations using the same algorithm. A Malaysian outbreak of dengue disease system model was presented by Noor Diana et al. [13] with the application of three classifiers. Data features were used to classify dengue dataset and its performance was compared with other previous research on dengue fever. Results indicated that the performance was better than other related works. Kashish Ara et al. [14] used weka tool to predict dengue. Dengue dataset was categorized and evaluated against various prediction algorithms with distinct interfaces. [15] analyzed various classifiers to identify the overall population infected by dengue disease in Jhelum district and its neighboring regions. [16] used microscopic blood image report applying an automated and intelligent system model to determine dengue fever. The report was the input and signals were filtered to extract the attribute characteristics. Later these features were fed into artificial neural networks. Classification accuracy of 98% was obtained with classification with back propagation network. Alternating decision trees was used by Kumar M.N. in [17] to detect the dengue fever occurrence at an early stage. It was observed that compared to decision tree which produced 78% accuracy, the ADTree was able to yield an accuracy of 84%. In [18] two distinct datasets of dengue was utilized from two different medical centers (Songklanagarind Hospital and Srinagarindra Hospital) each with 400 features by Thitiprayoonwonsge D. et al. These datasets were further required for analysis of dengue infection with the use of decision tree. An accuracy rate of 97.6% and 96.6% were the output of classification.

5. Proposed Attribute Optimization Algorithm

In our proposed work, the dataset used for diagnosis purpose is Dengue fever. There are 9 attributes in this dataset with 100 instances. Our study consists of proposing a more efficient attribute optimization technique based on Genetic Search Optimization for identifying the presence of Diabetes disease in patients. A 2-point crossover is implemented to the chosen solution space with a probability of 20%. A 1-bit Mutation is used with 20% probability on solution space generated after crossover. In genetic algorithm, crossover and mutation, are known at prior and are static throughout all generations. However if at every round the mutation and crossover probability are varied, more diverse solution space can be explored. Hence a Genetic Algorithm performs better if it can adapt itself to a suitable crossover and mutation rates in every generation. This is an important parameter which we have added in our proposed Attribute Selection method.

Eventually the efficient version for the newly developed Genetic Algorithm is presented and is called Enhanced and Adaptive Genetic Algorithm (IA-GA). IA-GA algorithm is a 10 step algorithm shown in Pseudo code 5. It comprises four different modules which include:

i. *ISS_Generate ($FS^{initial}$, FS^{final}) module:* The initial binary-encoded solution space is generated as shown in pseudo code 1.

ii. *Compute_f (n) module:* A new fitness function for the Enhanced Genetic Search is generated as shown in pseudo code 2.

iii. *Adaptive_CR-MR module:* This module exhibit the dynamic capability of IA-GA algorithm by altering the CR and MR in every round as shown in pseudo code 3.

iv. *R_Mutate module:* A new MSB and LSB based modified Mutation operation is presented as shown in pseudo code 4.

According to *ISS_Generate ($FS^{initial}$, FS^{final})* module as shown in table 2 we have introduced a new method to generate the initial solution space from the original set of attributes. For every attribute we have considered a threshold value. The threshold value may be the mean value for numerical attributes or it may be the attribute value for which the disease occurs. Based on the threshold value the column values of every attribute are modified. For numerical features values less than mean is allotted a 0 while values more than mean are adjusted as 1. Similarly for non-numerical values if the value is identical to the value that causes the disease it is labeled as 1 else it is 0. Then the frequency count of total 1's are done for every attribute and finally based on maximum count of 1's for an attribute the significant features are retained.

Table 2: Pseudo code 1 for ISS_Generate ($FS^{initial}$, FS^{final})

```
Pseudo code 1: ISS_Generate ($FS^{initial}$, $FS^{final}$)
Step 1:Initialize $FS^{initial} = \{F_1, F_2 \ldots F_n\}$ for n features
Step 2:Find Threshold of each Feature, $Feature_i^{th}$
Step 3:Partition column value of each Feature into 2 bounds:
Upper bound (> $Feature_i^{th}$)
Lower bound (< $Feature_i^{th}$)
Step 4:If Feature = Numerical,
Find Mean ($Feature_i$)
If $Feature_i$ (value) >= Mean ($Feature_i$)
$Feature_i$ (value) = 1
Else
$Feature_i$ (value) = 0
Step 5:If Feature! = Numerical,
$DOP^{max}$ (value) = 1 AND $DOP^{min}$ (value) = 0
Step 6:Find [No. of 1's] $Feature_i$ (count)
Step 7:Reject Features $^{Min. Count of 1's}$ ($Rejection^{Prob}$ = 5%)
Step 8: Compute $FS^{final}$ = {1- Features $^{Min. Count of 1's}$}
```

Compute_f (n) module generates a new Fitness function as shown in table 3. The fitness function is assumed to depend on two vital factors which include Misclassification Rate and Frequency count of zeroes in the individual chromosomes. Based on these two factors a new fitness function is evaluated and accordingly the individual solutions are ranked. The least fitness function value is given the topmost priority.

Table 3: Pseudo code 2 for Compute_f (n)

```
Pseudo code 2: Compute_f (n)

Step 1: Create Initial solution space using ISS_Generate ($FS^{initial}$, $FS^{final}$) module
Step 2: Identify Solution Features based on label '1' allotment on individual solution
Step 3: Find Zeroes (count) for each individual
Step 4: Find PAR as:      (TN + TP)/ (TN + TP + FN + FP)
Step 5: Determine M-PAR as: 1 - (TN + TP)/ (TN + TP + FN + FP)
Step 6: Compute f (n) as:   f (n) = M-PAR + βz
Step 7: Priority of solutions based on f (n) value as: Prior (Solution) α 1/f (n)
```

Adaptive_CR-MR procedure deals with variation in Crossover rate and Mutation rate. The pseudo code is shown in table 4. First it determines an initial Crossover Rate (CR) and Mutation Rate (MR) pair. Generally a high Crossover are with a low Mutation rate is preferred in a Genetic Algorithm. As the Crossover and mutation rate are varied here, the crossover rate and mutation rate are set at 0.5 each for the first generation.

Let a Crossover occurs between two parents. Let the fitness summation for the two offspring be f1and the fitness summation of the two parents be f2. The Crossover Variation (CV) is denoted as:

$$CV = f1 - f2 \qquad (1)$$

The Mean Crossover Variation value MCVα for a round with c_n crossover operations is:

$$MCV_\alpha = 1/\, c_n \sum MCV \qquad (2)$$

Thus, MCV_α is a parameter to underline the performance of Crossover for a particular iteration. Similarly let Mutation Variation value (MV) denotes the output of a Mutation as:

$$MV = f_{new} - \qquad (3)$$

where the fitness value for the resultant offspring is f_{new} while f_{old} represents the fitness value of the individual before mutation. The Mean Mutation Variation value (MMV) for a round that witness m_n mutations is:

$$MMV_\alpha = 1/\, m_n \sum MMV \qquad (4)$$

With the use of these mean values, the crossover and mutation probability rates are adjusted accordingly towards the completion of each round. Based on the performance in the previous round the operators are self adjusted. It suggests that the participation of operator with higher mean average value is more frequent and thus enhances its probability in the subsequent generation too and vice versa as shown below:

Condition 1: MCV > MMV
$CR = CR + \beta$, $MR = MR - \mu$
Condition 1: MCV < MMV
$CR = CR - \beta$, $MR = MR + \mu$

where β and μ are the adjustment factors corresponding to CR and MR respectively.

Table 4: Pseudo code 3: Adaptive_CR-MR

```
Pseudo code 3: Adaptive_CR-MR
Step 1: while $round \leq round^{max}$ do
Step 2: Reset the new population space $P_1$;
Step 3: Every individual solution in $P$; is evaluated with a fitness function $f(x)$
        while $|P_1| \leq N$ do
Step 4: Two parents from $P$ are chosen;
Step 5:Crossoveroccurs with aggregation ofCrossover Variation value (CV)
Step 6:Mutationoccurs with aggregation of Mutation Variation value (CV)
Step 7: The offspring is inserted to $P_1$;
End while
Step 8: Compute $MCV_\alpha$ and $MMV_\alpha$ and adjust the crossover rate $CR$ and mutation rate $MR$;
Generation = Generation + 1;
End while
```

A new modification for Mutation operation is highlighted in R_Mutate module as shown in table 5. It is applicable to the last iteration only. Based on the output of crossover operation the solutions are examined carefully. If the fitness function of chromosomes are low enough then the MSB are flipped to adjust for the solution to converge at global optimum and if the fitness function is high enough then the LSB are flipped in order to fine tune the solutions.

Table 5: Pseudo code 4: R_Mutate

```
Pseudo code 4: R_Mutate
Step 1: Input = {Solution set after Crossover}
Step 2: Compute f (n)
Step 3:If f (n) = low then Flip MSB
        else
Step 4: Flip LSB
```

The overall Enhanced and Adaptive Genetic Algorithm Technique is presented in IA-GA module as illustrated in table 6. The initial binary-encoded solution space is generated based on ISS_Generate ($FS^{initial}$, FS^{final}) module. Compute_f (n) module determines the Fitness function for every solution set. The fitness function for the feature set is recorded and stored. A Two-Point Crossover operation is applied to the encoded solution set and then the fitness function for the resultant set after crossover is calculated. If the fitness function is low compared to that of the solution set before crossover then it is interchanged and better fitness function solution set are restored. After crossover normal mutation is implemented to the individual solution. This procedure is repeated for the specified last but one number of generations. Except in the last generation a modified Mutation procedure is used according to R_Mutate module. As a result after mutation the final optimized features set is the output which is used for classification. The dynamic and adaptive nature of IA-GA is derived from the changing value of CR and MR in every generation. Based on the values of Mean Crossover Variation (MCV_α) and Mean Mutation Variation (MMV_α), the Crossover rate and Mutation rate alters in every generation. This is the core part of Adaptive_CR-MR module.

Table 6: Pseudo code 5: IA-GA

```
Pseudo code 5: IA-GA

Step 1: Initialize binary-encoded solution space using ISS_Generate ($FS^{initial}$, $FS^{final}$) module
Step 2: Store f (n) set computed in Compute_f (n) module in f '(n)
Step 3: Apply 2-Point Crossover to solution space based on f '(n) using Adaptive_CR-MR module
Step 4: Calculate and store f (n) set computed after Crossover in f '' (n)
Step 5: Compare f '(n) with f '' (n)
Step 6: Swap low Prior $(Solution)_{Individual}$ in f '' (n) with high Prior $(Solution)_{Individual}$ in f '(n)
Step 6: Apply Mutation operation using Adaptive_CR-MR module
Step 7: Repeat until termination condition met (k generations)
Step 8: Apply R_Mutate module at $k^{th}$ step after step 7
Step 9:Compute Final Feature Set ← $Feature_{Min.\ 0's\ Freq.\ count}$
Step 10: Output = {Optimum feature Set}
```

Basically our proposed IA-GA algorithm is an amalgam of two basic ideas which are:

Swapping of Chromosomes for better Fitness value at every generation: At every round of algorithm better feasible solutions are retained while eliminating the least feasible solutions. The retention of chromosomes are determined by their fitness function value. Better chromosomes are swapped with least preferred chromosomes at every round.

Change in Crossover rate and Mutation rate at every generation: The adaptive nature of our proposed algorithm arises from the dynamic aspect of crossover rate and mutation rate. At every round these two values varies which makes the algorithm more robust and flexible.

The parameters highlighted in the pseudo codes are described in below table 7:

Table 7: Parameters highlighted in the pseudo codes

Parameters highlighted in the pseudo codes	
Name of parameter	*Description*
Optimized_GA	Pseudo code for our proposed Genetic Search algorithm
ISS_Generate ($FS^{initial}$, FS^{final})	Pseudo code for Generation of Initial Feature Set for first iteration
Compute_f (n)	Pseudo code for Calculation of Fitness

	function f(n)
R_Mutate	Pseudo code for Restrictive Mutation
$FS^{initial}$	original set of features in the dataset
FS^{final}	Set of featured after applying Optimized Genetic Search technique
$Feature_i^{th}$	Threshold value of individual features based on which presence of a disease is known.
$Feature_i$ *(value)*	The column value of each feature
Mean $(Feature_i)$	The mean value of each feature
DOP^{max} *(value)*	Maximum value of Disease Occurrence Probability
DOP^{min} *(value)*	Minimum value of Disease Occurrence Probability
[No. of 1's] $Feature_i$ *(count)*	frequency count of 1's in the column of a feature
$Features^{Min.\ Count\ of\ 1's}$	Features with low frequency of 1's
$Features^{Max.\ Count\ of\ 1's}$	Features with high frequency of 1's
Zeroes (count)	Frequency of Zeroes present in an individual binary-encoded solution.
PAR	Prediction Accuracy Rate
TN, TP, FN and FP	True Negatives, True Positives, False Negatives and False Positives respectively.
f(n)	Fitness Function
1 - (TN + TP)/ (TN + TP + FN + FP)	Misclassification Rate = M-PAR
B	fitness constant (usually 0.5)
Z	No. of Zeroes in individual solution set
MSB	Most Significant Bits
LSB	Least Significant Bits
$Crossover^{prob}$	A measure denoting how often will be crossover performed.
f '(n)	array to store the fitness function values before crossover
f "(n)	array to store the fitness function values after crossover
Prior (Solution) $_{Individual}$	Priority of a given individual solution
	based on fitness function value
$Mutation^{prob}$	A measure indicating how often will be parts of chromosome mutated.
K	Total number of generations till the algorithm will run
MCV_α *and* MMV_α	Mean Crossover Variation value and Mean Mutation Variation value respectively.

The sequence of steps of IA-GA algorithm is diagrammatically represented in figure 1 below. The original dataset of Diabetes denoting all attributes is the input. A sub-optimal feature subset is generated using Max. 1's count rule as presented in *ISS_Generate* $(FS^{initial}, FS^{final})$ module. From the reduced feature set obtained random initial solution spaces (chromosomes) are developed. The fitness function is computed according to *Compute_f (n)* module. Priority based ranking of chromosomes are done based on the value of fitness function computed. A 2-point Crossover operation is performed with a 20% probability of chromosomes. The fitness function of chromosomes after Crossover is recomputed and less priority solutions are dropped from the solution space for the subsequent generation. Individuals are again ranked based on their fitness function value. A random 1-bit Mutation is applied on the resultant set with a mutation probability of 20%. The CR and MR are fixed at 20% for the first generation while for subsequent generations these factors are computed according to Adaptive_CR-MR module. In this module the MCVα and MMVα are determined and then the CR and MR value are adjusted accordingly for the following round. The entire process is repeated for the specified k number of generations. At the last round Restrictive Mutation concept is applied to the resultant solution space based on R_Mutate module. Then a final count of number of 1's is done for the final attribute set. Max. 1's count rule is followed based on which features with a low count of 1's in their corresponding column are rejected and the rest are accepted and evaluated as the optimal feature set.

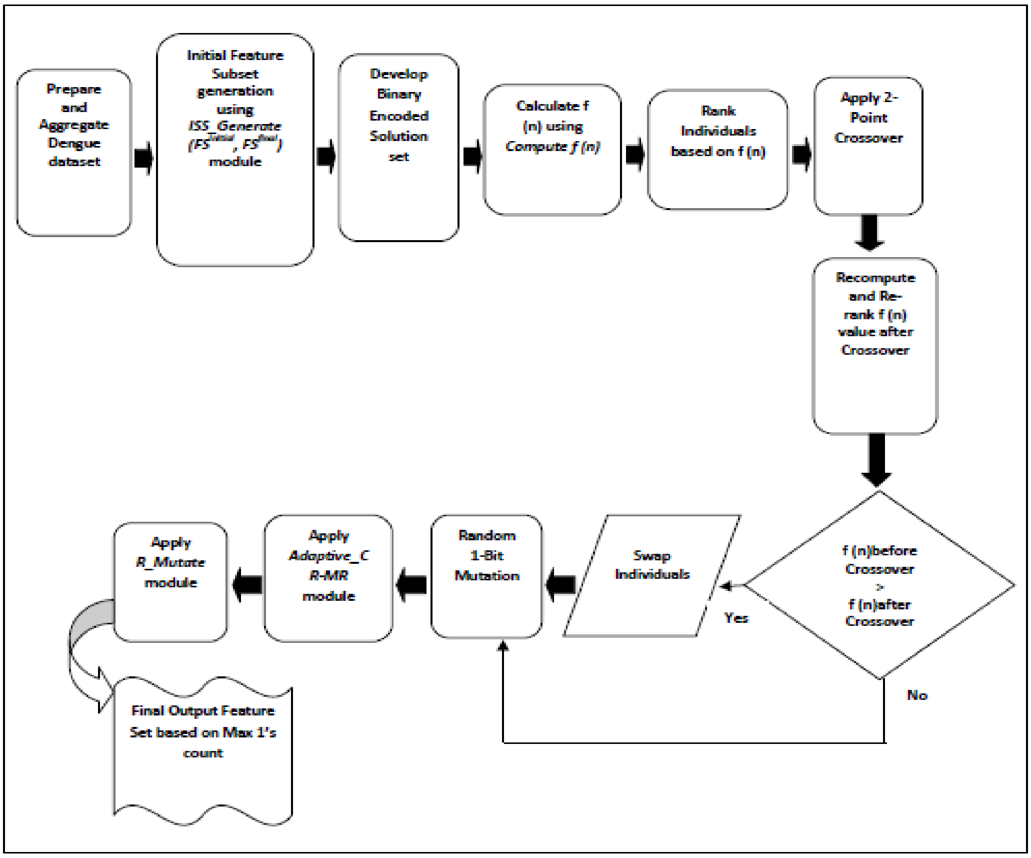
Prepare and Aggregate Dengue dataset
Initial Feature Subset generation using ISS_Generate (FS^Initial, FS^final) module
Develop Binary Encoded Solution set
Calculate f (n) using Compute f (n)
Rank Individuals based on f (n)
Apply 2-Point Crossover
Recompute and Re-rank f (n) value after Crossover
f (n)before Crossover > f (n)after Crossover
Yes
No
Swap Individuals
Random 1-Bit Mutation
Apply Adaptive_CR-MR module
Apply R_Mutate module
Final Output Feature Set based on Max 1's count

Figure 1: Working Principle of Proposed IA-GA Attribute Selection Algorithm

Classification with our newly developed algorithm is represented in figure. 2. The original healthcare related dataset is the input to our proposed Optimized Attribute Selection method. Based on the features the irrelevant attributes are eliminated. It simultaneously interacts with the Classification algorithm (MLP) and an optimized reduced set of features is obtained. This reduced set is subjected to classification using a suitable classifier. Based on the output the performance of classification is computed using performance parameters and the Classification Accuracy is determined.

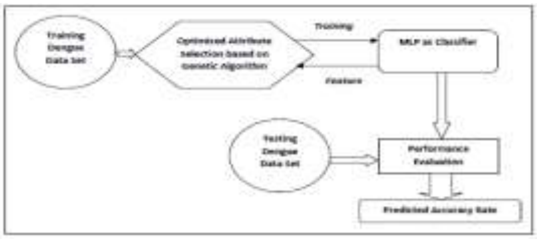

Figure 2: Classification of Dengue Patients with MLP using IA-GA Algorithm

6. Results and Analysis

In our study a newly improved and adaptive version of Genetic Algorithm was developed which was named as Improved and Adaptive Genetic Algorithm (IA-GA). The proposed algorithm worked on Dengue Fever dataset to detect the presence of the dengue in patients. Various performance parameters are used to evaluate the proposed algorithm efficiency. Our implementation is performed for 100 generations. A Confusion Matrix is illustrated and some basic parameters are derived from these matrixes that can represent the effectiveness of machine learning techniques which are shown in table. 8.

Table 8: Parameters of a Confusion matrix

True Positives	Positive samples correctly identified during Classification
True Negatives	Negative samples correctly identified during Classification
False Positives	Negative samples incorrectly identified as negatives during Classification
False Negatives	Positive samples misclassified as negatives

Classification Accuracy is used as a statistical measure that indicates the correctness of a binary classification test. It is the ratio between correctly predicted inferences and total inferences.

$$\text{Prediction Accuracy} = \qquad (5)$$

To have robust model prediction accuracy is not the only performance metric to rely upon. Other than classification accuracy several other metrics are critical. Precision is the ratio between accurately predicted positive inferences and total number of positively predicted inferences. while Recall is the ratio between accurately predicted positive inferences and all inferences in the class.

$$\text{Precision} = \frac{\text{TP}}{\text{TP+FP}} \qquad (6)$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP+FN}} \qquad (7)$$

Though Precision along with Recall metrics are used in computing optimum technique, yet a single parameter is needed to assist in decision making. Hence F-Measure acts as a unique factor in evaluating the effectiveness of a developed model. F-measure is evaluated as the harmonic mean between precision and recall. At times the effectiveness of a classifier may be tough to determine if we consider only positive predictive value and sensitivity parameters. Let us assume performance of two distinct algorithms were evaluated and found that one has a greater positive predictive value but relatively less sensitivity value. In such a scenario another performance metric called as F-Score is used which is a balanced mean between positive predictive value and sensitivity. Higher value of F-Score is a direct measure of the effectiveness of an algorithm.

$$\text{F} - \text{Score} = \frac{2\text{TP}}{2\text{TP+FP+FN}} \qquad (8)$$

The total Latency is computed as the cumulative sum of time taken to build the Classification model and the time to predict the output.

$$\text{Latency} = \text{Model build up time} + \text{Disease prediction time} \qquad (9)$$

Our proposed attribute selection algorithm was used with Multi layer Perceptron for classification of patients having dengue or not. The performance indices used in or research is presented in table 9.

Table 9: Performance Indices used for Dengue Prediction

Performance Metrics	**Description**
Total Number of Tuples	Total number of tuples present in dataset.
Accurately Predicted Tuples	Proportion of testing tuples accurately predicted.
Inaccurately Predicted Tuples	Proportion of testing tuples inaccurately predicted.
Precision	Proportion of retrieved tuples that are relevant.
Recall	Proportion of relevant tuples that are retrieved.
F-Score	Harmonic mean of precision and recall.
Classification Accuracy	Percentage of tuples accurately predicted.
ROC Area	for visualizing and selection of classifier based on their performance

Effectiveness of our proposed IA-GA algorithm is measured by evaluating its performance with simple genetic algorithm as the attribute selection method. A comparative study is performed taking Multi Layer Perceptron (MLP) classifier along with Genetic Algorithm (GA) to predict the presence of dengue fever. It is observed that IA-GA predicts the presence of dengue more efficiently and precisely with 89% accuracy rate. Our proposed Attribute Optimization method shows an optimal Precision value of 0.876 while the Recall value is recorded as 0.88. The recorded value of the Harmonic mean of Precision and recall called F-Score is 0.864 with IA-GA. The results are presented in table 10.

There exists several classification algorithms used earlier for effective prediction of dengue fever using the same dengue dataset. Some of the important classifiers used in [19] include Naïve Bayes, J48, SMO, RANDOM tree and REP tree. These algorithms yielded very good result for dengue fever diagnosis. Attribute selection technique has not been explored much for dengue prediction task. IA-GA attribute selection algorithm has been used here for identifying an optimal subset of attributes of dengue dataset. This attribute subset is treated with MLP classifier for determination of presence of dengue virus or not. IA-GA algorithm is compared with these existing algorithms and evaluated against some vital performance metrics as presented in table 9. The results are depicted in figure 3 to figure 7. As observed in the results, several existing classifiers are used for classification of dengue fever. For the purpose of effective comparative analysis, we have used MLP as classifier without any attribute selection algorithm as well as a combination of MLP and Genetic algorithm as an attribute selection method in our study. An extensive comparative study of these classifications is drawn with classification using our proposed IA-GA algorithm. It is seen that classification with IA-GA technique improves the accuracy of prediction of dengue. Apart from this, evaluation with other performance indicators like Precision, Recall and F-Score also suggest that IA-GA attribute selection method is more optimal in determining the presence of dengue in patients.

	Classification Accuracy	Precision	Recall	F-Measure	ROC Area
MLP	82	0.824	0.82	0.812	0.696
MLP + GA	85	0.836	0.84	0.82	0.72
MLP + IA-GA	89	0.876	0.88	0.864	0.728

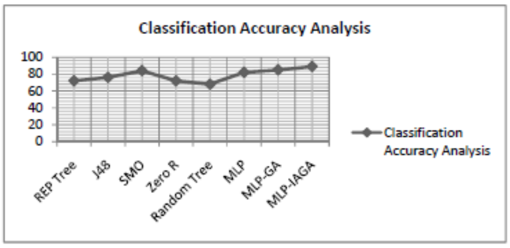

Figure. 3: Comparison of Accuracy rate of Proposed Algorithm IA-GA with MLP Classifier

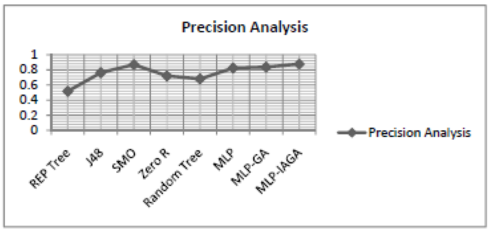

Figure. 4: Comparison of Precision of Proposed Algorithm IA-GA with MLP Classifier

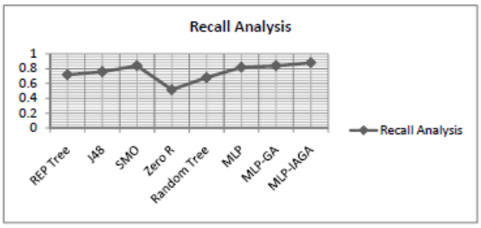

Figure. 5: Comparison of Recall of Proposed Algorithm IA-GA with MLP Classifier

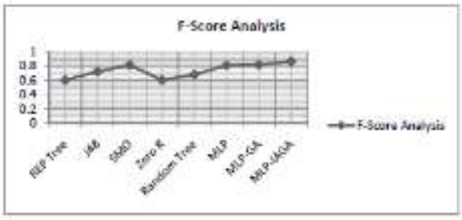

Figure. 6: Comparison of F-Score of Proposed Algorithm IA-GA with MLP Classifier

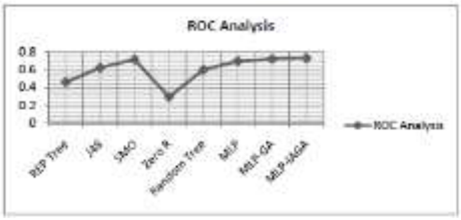

Figure. 7: Comparison of ROC Area of Proposed Algorithm IA-GA with MLP Classifier

Our classification analysis is performed on dengue dataset comprising of 100 instances. Various existing algorithms as stated above are implemented with this dataset earlier. Our IA-GA algorithm integrated with MLP classifier outperforms other classifiers in accurate prediction of dengue fever. Figure 8 illustrates the impact of IA-GA attribute selection algorithm on the dataset. As it is observed the combination of MLP and IA-GA is able to predict 89 tuples accurately while only 11 tuples are incorrectly predicted to the other class. The proportion of incorrectly predicted tuples in other classifiers is higher than our classification with IA-GA.

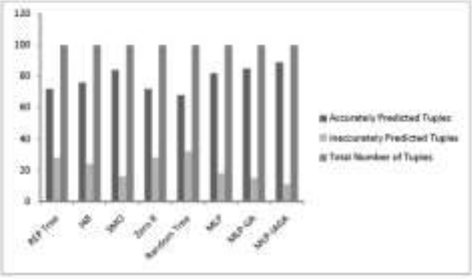

Figure. 8: Comparison of Accuracy of dataset of Proposed Algorithm IA-GA with MLP Classifier

10-fold Cross Validation (CV) technique was used to train the data samples which partition the data into 10 equal sized samples each where 9 sets were being to train and 1 set of data is used for testing. The same procedure is repeated for 10 rounds and an arithmetic mean value of accuracy is computed. The figure.9 highlights the accuracy for every fold while applying varying Cross-Validation method. The mean accuracy for the 10th fold is computed to be an optimal value of 89%.

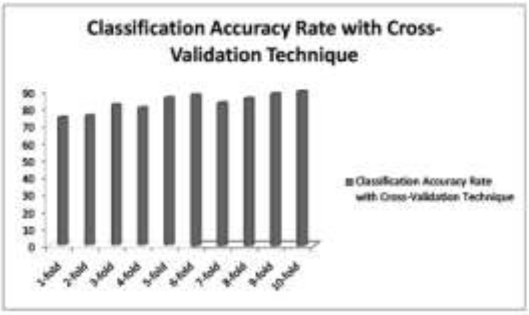

Figure. 9: Prediction Accuracy of classification with IA-GA method using Cross-Validation technique

7. Conclusion

This paper introduced a new version of Attribute Optimization method named as IA-GA which is a new variant of Genetic Algorithm. It is an enhanced and adaptive version of Genetic Algorithm which is fruitful in feature optimization of various data samples. Our proposed work devises an initial solution space using Max. 1’s count Frequency rule. It develops and computes a new Fitness function for chromosomes, develops a new Mutation concept for the last generation. The study deals with efficient and dynamic interaction between Crossover and Mutation by suitable variation in their probability rate. In our work Dengue Fever dataset was taken into account on which IA-GA method was applied with Multilayer Perceptron as a Classifier. The results are evaluated against various critical performance metrics. It is observed that the results are very encouraging and our newly developed algorithm is found to be efficient and effective in determining the presence of Dengue in patients. IA-GA is compared and evaluated with other previously developed techniques and it is seen that our proposed method outperforms other related works. Thus IA-GA method may be suggested to the medical experts as an effective, enhanced and dynamic feature optimization technique to precisely predict the presence of Dengue in patients. Our future work will stress emphasis in developing more adaptive and performance driven attribute selection techniques while revising metrics like size of population and other replacement parameters.

References

[1] Farooqi W, Ali S (2013) A Critical Study of Selected Classification Algorithms for Dengue Fever and Dengue Hemorrhagic Fever. Frontiers of Information Technology (FIT), 11th International Conference on IEEE.

[2] Farooqi W, Ali S, Abdul W (2014) Classification of Dengue Fever Using Decision Tree. VAWKUM Transaction on Computer Sciences 3: 15-22.

[3] Rigau-Pérez JG, et.al. (1998) Dengue and dengue haemorrhagic fever. The Lancet 19: 971-977.

[4] K.C. Tan, E.J. Teoh, Q. Yua, K.C. Goh, (2008). A hybrid evolutionary algorithm for attribute selection in data mining, Elsevier.

[5] Farooqi W, Ali S (2013) A Critical Study of Selected Classification Algorithms for Dengue Fever and Dengue Hemorrhagic Fever. Frontiers of Information Technology (FIT), 11th International Conference on IEEE.

[6] Tanner L, Schreiber M, Low JG, Ong A, Tolfvenstam T, et.al. (2008) Decision Tree Algorithms Predict the Diagnosis and Outcome of Dengue Fever in the Early Phase of Illness. PLoS Neglected Tropical Disease 12: e196.

[7] 5. Phyu TN (2009) Survey of classification techniques in data mining. Proceedings of the International MultiConference of Engineers and Computer Scientists Vol 1.

[8] Vong S, et.al. (2010) Dengue incidence in urban and rural Cambodia: results from population-based active fever surveillance, 2006–2008. PLoS neglected tropical diseases 4: e903.

[9] Faisal T, Ibrahim F, Taib MN (2010) A noninvasive intelligent approach for predicting the risk in dengue patients. Expert Systems with Application 37: 2175-2181.

[10] . Ibrahim F, Taib MN, Abas WA, Guan CC, Sulaiman S (2005) A novel dengue fever (DF) and dengue haemorrhagic fever (DHF) analysis using artificial neural network (ANN). Computer Methods and Programs in Biomedicine 79: 273-281.

[11] Daranee T, Prapat S, Nuanwan S (2012) Data mining of dengue infection using decision tree. Entropy 2: 2.

[12] Rigau-Pérez JG, et.al. (1998) Dengue and dengue haemorrhagic fever. The Lancet 19: 971-977.

[13] Tarmizi NDA, et.al. (2013) Classification of Dengue Outbreak Using Data Mining Models. Research Notes in Information Science 12: 71-75.

[14] Shakil KA, Anis S, Alam M (2015) Dengue disease prediction using weka data mining tool. arXiv preprint arXiv:1502.05167.

[15] N.Subitha and Dr.A.Padmapriya "Diagnosis for Dengue Fever Using Spatial Data Mining", International Journal of Computer Trends and Technology (IJCTT) ,August 2013.

[16] Daranee, Pratap Suriyaphol and Nuanwan," Data Mining of Dengue Infection Using Decision Tree", July 2015.

[17] Kumar MN (2013) Alternating Decision trees for early diagnosis of dengue fever. 1305.7331.

[18] Thitiprayoonwongse D., Suriyaphol P., Soonthornphisaj N., Data mining of dengue infection using decision tree, Entropy, 2: 2, 2012.

[19] M.Bhavani and S.Vinod kumar, "A Data Mining Approach for Precise Diagnosis of Dengue Fever" International Journal of Latest Trends in Engineering and Technology, Vol.(7)Issue(4), pp.352-359, 2013