

# ANN BASED FEATURES SELECTION APPROACH USING HYBRID GA-PSO FOR siRNA DESIGN

<sup>1</sup>Ranjan Sarmah, <sup>2</sup>Mahendra K. Modi, <sup>1</sup>Shahin Ara Begum\*

<sup>1</sup>Department of Computer Science, Assam University Silchar, Assam, India

<sup>2</sup>Department of Agril. Biotechnology, Assam Agricultural University, Assam, India

<sup>1</sup>Department of Computer Science, Assam University Silchar, Assam, India

\*Corresponding Author

## ABSTRACT

*siRNA has become an indispensable tool for silencing gene expression. It can act as an antiviral agent in RNAi pathway against plant diseases caused by plant viruses. However, identification of appropriate features for effective siRNA design has become a pressing issue for researchers which need to be resolved. Feature selection is a vital pre-processing technique involved in bioinformatics data set to find the most discriminative information not only for dimensionality reduction and detection of relevance features but also for minimizing the cost associated with features to design an accurate learning system. In this paper, we propose an ANN based feature selection approach using hybrid GA-PSO for selecting feature subset by discarding the irrelevant features and evaluating the cost of the model training. The results showed that the performance of proposed hybrid GA-PSO model outperformed the results of general PSO.*

## KEYWORDS

*SIRNA, PSO, GA-PSO, Features Selection, ANN, Cost Evaluation, GA-BPNN, heuristic optimization*

## 1. INTRODUCTION

In the present scenario, RNAi is the most promising technology used to study the Gene function and drug target identification as shown by [1, 2] which is mediated by small 21-30 base pair long double stranded small RNA molecules i.e. miRNA (micro RNA) and siRNA (small interfering RNA), generated endogenously or exogenously. The small RNAi molecules interfere in RNAi pathways thereby disrupting the post transcriptional product via dicer and RNA induced silencing complex (RISC) formation thus controlling the gene function. At present there are number of tools available on public servers for siRNA prediction against gene of interest. However, these tools generate the bunch of siRNA with varied efficacy and not the optimal number of siRNAs [3] because of challenges like bias dataset and over fitting issues of siRNA design as reported by [4]. As a result, designing the most effective siRNA, based upon optimal features selection has posed to be the one of the greatest challenges in RNAi technology. Hence it becomes necessary to identify the suitable features for designing effective siRNA.

In siRNA design, the motivation for features selection technique has shifted from being a mere illustrative example to becoming a real pre-requisite for model building. Often based on local neighbourhood searches, heuristic solution methods for optimization are sensitive to starting point conditions and tend to get trapped in local minima. In order to avoid these types of problems, Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO) and Genetic Algorithm (GA) randomizes the search space stochastically so as to explore the search space [5-10] which provides global optima with proper tuning of the parameter. In addition, these approaches use only a

simple scalar performance measure that does not require or use derivative information. PSO method is one of the heuristic optimization techniques which is effectively applied in various optimization problems.

Although several works have been reported on feature selection techniques but the studies are found to be very limited in siRNA efficacy prediction in plant dataset. Prasad et al. [11] have shown that siRNA efficacy prediction using SVM classifier integrated with evolutionary and natural computing heuristics provide significant improvement in the prediction result and obtain the most appropriate set of features for prediction of siRNA efficacy. Another work reported by Jain and Prasad [12] in the year 2009 presented an ant colony optimization based meta-heuristic methodology to identify the features subset and the results of which were analyzed using linear regression and ANCOVA methods. Authors reported that both sequence and thermodynamic features are equally important in the effective designing of siRNA.

The present study is a wrapper feature selection approach based on ANN (BPNN) algorithm where ANN is used for evaluating the classification performance of the selected features obtained from hybrid GA-PSO feature selection approach on a plant siRNA dataset. The experiments carried out in this study try to demonstrate the application of PSO in order to select a subset of features so as to identify important features for effective siRNA design. Further, to optimize the performance of PSO, ANN based hybrid GA-PSO approach is implemented by combining the merits of both PSO and GA. Furthermore, different combination of siRNA properties and their associated features is performed and tested with hybrid GA-BPNN to enhance the predictive accuracy for designing potential siRNA.

## **2. MATERIALS AND METHODOLOGY**

### **2.1.Preparation of Dataset**

In the present experiment, a plant dataset with 1100 siRNA sequences has been taken into consideration as given by Sarmah et al. [13]. Five different properties namely presence of motifs in a sequence ( $M_p$ ), absence of motif in a sequence ( $M_a$ ), presence of specific nucleotide at a particular position ( $NP_p$ ), absence of specific nucleotide at a particular position ( $NP_a$ ) and thermodynamic characteristics ( $T_c$ ) and the number of original features 15, 15, 14, 15 and 11 respectively have been considered for the study.

### **2.2.Feature Selection**

Feature selection is mainly composed of two approaches - Filter method and Wrapper method. In filter method, the selection of features is not dependent on the classifier used, whereas, the performance of classification of the selected features is used as the evaluation criteria for wrapper feature selection approach. Although the performances of wrapper method may be better, compared to filter method but it requires greater computational resources [14]. For this very reason, a hybrid approach known as embedded method has emerged that integrates both filter and wrapper method.

#### **2.2.1. Filter method**

The basic structure of filter method is depicted in Figure 1. The filter methods are used as a basic pre-processing step where selection of features is made on the basis of their score obtained in different statistical tests.

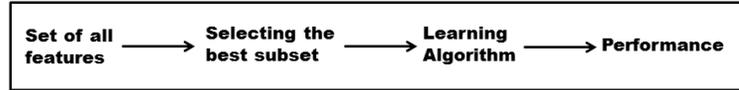


Figure 1. Filter method

### 2.2.2. Wrapper method

The basic structure of wrapper method is depicted in Figure 2. A predictive model is used in wrapper methods to set score to the feature subsets. The new subset so obtained is used to train a model, which is tested on a hold-out set. The number of mistakes is counted on the hold-out set (the error rate of the model) and a score is provided for that subset. As the wrapper methods train a new model for each subset, they provide the best performing feature set compared to filter method despite being computationally intensive.

Some of the popular classification algorithms such as Support Vector Machines (SVMs), Artificial Neural Network (ANN), Naïve Bayes (NB), Decision Tree (DT), k-nearest neighbour (KNN) and Linear Discriminant Analysis (LDA) have been applied to wrapper approach for feature selection [15-17].

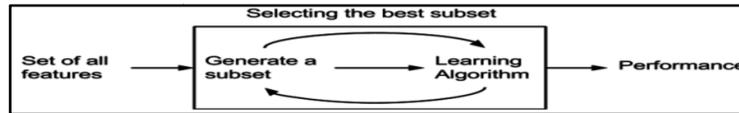


Figure 2. Wrapper method

### 2.2.3. Embedded method

The basic structure of the embedded method is depicted in Figure 3. The merits of both filter and wrapper method combine to constitute the embedded method. It is implemented by algorithms that have its own built-in feature selection methods.

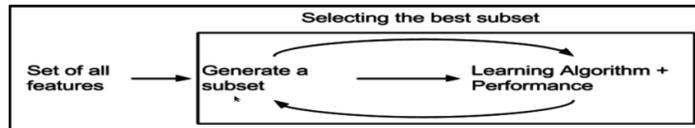


Figure 3. Embedded method

## 2.3. Particle Swarm Optimization (PSO)

Particle swarm optimization (PSO) is an evolutionary computational technique to solve problems whose solution can be represented as a point in an n-dimensional solution space. Each individual particle utilizes two important information in a decision processes. The first one is their own experience; i.e. they observe the “fitness” of themselves and the second one is other agent’s experiences i.e. they have knowledge of how their neighbours have performed and “emulate” successful neighbours by moving towards them. Particle swarm algorithm is a simple approach and has been found to be effective in various problem domains. The inspiration drawn from the study of Frank Heppner on bird flocking behaviour led James Kennedy and Russell Eberhart to develop PSO in 1995 [18-24].

The PSO is initialized with a population of random solutions. A random velocity is given to each potential solution, called a particle (agent) and is flown through the problem space. Each agent has memory that helps to keep track of its previous best position (called the  $P_{Best}$ ) and its corresponding fitness. For each agent, a number of  $P_{Best}$  exists in the swarm and the agent with highest fitness is called the global best ( $G_{Best}$ ) of the swarm. Each particle in an n-dimensional space is treated as a point. So the  $i^{th}$  particle is represented as  $X_i = (x_{i1}, x_{i2}, \dots, x_{in})$  and the

previous best position of the  $i^{\text{th}}$  particle ( $P_{\text{Best}i}$ ) that provides the best fitness value is represented as  $P_i = (p_{i1}, p_{i2}, \dots, p_{in})$ . Among all the particles, the best particle in the population is represented by  $P_g = (p_{g1}, p_{g2}, \dots, p_{gn})$ . The velocity of the particle obtained from the change in position for particle  $i$  is represented as  $V_i = (v_{i1}, v_{i2}, \dots, v_{in})$ . The particles are manipulated according to the following equations (the superscripts denote the iteration):

$$v_i^{k+1} = w \times v_i^k + c_1 \times r_1 \times (p_i - x_i^k) + c_2 \times r_2 \times (p_g - x_i^k), \quad (1)$$

$$x_i^{k+1} = x_i^k + v_i^{k+1} \quad (2)$$

where,  $i = 1, 2, \dots, N$ , where  $N$  refers to the population size;  $w$  refers to inertia weight and  $c_1$  and  $c_2$  are two positive constants called the cognitive and social parameter respectively;  $r_1$  and  $r_2$  are random numbers uniformly distributed within the range  $[0, 1]$ . Equation (1) is used to determine the  $i^{\text{th}}$  particle's new velocity  $v_i^{k+1}$ , at each iteration, while Equation (2) provides the new position of the  $i^{\text{th}}$  particle  $x_i^{k+1}$ , adding its new velocity  $v_i^{k+1}$ , to its current position  $x_i^k$ . Figure 4 shows the description of position and velocity updates of a particle for a 2-dimensional parameter space.

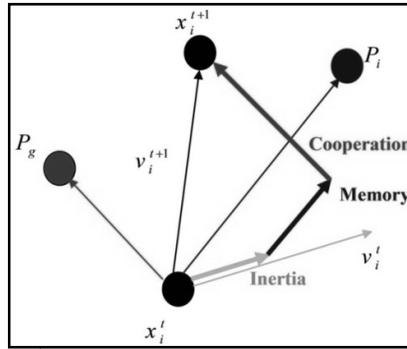


Figure 4. Description of velocity and position updates in PSO for a 2-dimensional parameter space. The pseudo code of the general PSO algorithm is shown in algorithm 1.

Algorithm 1. PSO algorithm

---

```

Randomly initialize positions and velocities of all particles.
Do:
Set PBest and GBest
Calculate particle velocity according to Equation (1)
Update particle position according to Equation (2)
Evaluate the fitness function
While reaching a satisfactory solution
    
```

---

**2.4.Hybrid GA-PSO**

A number of works have been reported on the hybridization of PSO with other heuristic optimization techniques including the hybrid approach of PSACO (particle swarm ant colony optimization) proposed by Sheloker et al. [25] for highly non convex optimization problems. Another work has been reported by Kao and Zahara in the year 2008 [26] for global optimization of multimodal functions by combining GA with PSO as a hybrid method.

But the application of feature selection for effective siRNA designing by using GA-PSO hybrid model for plant dataset is scanty. So, the present investigation is mainly focused on the implementation of a new ANN based hybrid GA-PSO model for selection of important feature subset and evaluating the cost associated with the features.

### 2.4.1. Genetic Algorithm

Genetic Algorithms (GA) was discovered by Holland in the 1960s and was further described by Goldberg [27]. The GAs have been broadly applied in optimization problems, neural networks, fuzzy logic control, scheduling, expert systems etc. [28]. For a particular problem, the GA describes a solution as an individual chromosome. Initial population of those individuals are then defined which represents a part of the solution space of the problem. Hence, the search space is defined to mean the solution space where each feasible solution is represented by a distinct chromosome.

The GA algorithm performs the following steps:

- 1) Initial population is generated by the randomly chosen chromosome set from the search space.
- 2) Fitness evaluations of individuals are performed by a specific objective function.
- 3) Selection, crossover and mutation are applied to obtain a new generation of chromosomes which is expected to be better than that of the previous.
- 4) This process is repeated until the final solution is reported.

Algorithm 2 shows the pseudo code of the general GA algorithm.

---

#### Algorithm 2 GA algorithm

---

```
Generate the initial population;  
Evaluate fitness of individuals in the population;  
Do:  
    Select parents from the population;  
    Recombine (crossover and mutation operator) parents to produce children;  
    Evaluate fitness of the children;  
    Replace some or all of the population by the children;  
    While a satisfactory solution has been found
```

---

### 2.4.2. Hybrid GA-PSO Algorithm

From the literature [29, 30] it has been found that most of the evolutionary techniques have the following procedure:

- 1) Random generation of an initial population.
- 2) Calculate fitness value for each particle that depends on the optimum distance.
- 3) Population reproduction on the basis of fitness values.
- 4) If optimal solutions are found, then stop. Else go to 2.

The above procedure shows that PSO shares several common points with GA. PSO and GA both starts with randomly generated population. Both these algorithms have fitness values that are used to evaluate the population. Both update the population and with random techniques search for the optimum. However none of the systems guarantee success.

PSO however, does not have genetic operators such as crossover and mutation. In PSO, the particles update themselves with the internal velocity. The information sharing mechanism in PSO is significantly different from GA where chromosomes are used to share information with each other. Even in the local version in most cases all the particles in PSO tends to converge to the best solution quickly as compared to GA.

The proposed hybrid GA-PSO for siRNA design aims to combine best features of GA and PSO by integrating the two algorithms where the best solution obtained from PSO is further optimized by GA using selection, crossover and mutation operator. The pseudo code of the proposed hybrid GA-PSO algorithm used in the present study is shown in algorithm 3.

---

### Algorithm 3 Hybrid GA-PSO algorithm

---

```
Initialize the PSO and GA parameters
While travel not completed
  While sub-travel not completed
    Evaluate the fitness function values of all particles by using Equation 3
    Set  $P_{Best}$  and  $G_{Best}$ 
    Updating the velocity and position according to Equation 1 and 2
    Evolution of infeasible particle
  End while
  While evolution not completed
    Updating  $P_{Best}$  and  $G_{Best}$ 
    Ranks individual according to the fitness value
    Selection
    Crossover
    Mutation
    Repair the infeasible of the population to be feasible
  End while
End while
```

---

## 2.5. Fitness Evaluation

In certain situations, a user is not only interested in maximising the performance of the model but also minimizing the cost associated with features [31]. The cost associated with a feature may come from economy, time, or other resources used to obtain feature values of objects [32, 33]. Furthermore, the cost may be associated with computational issues also [34].

A single fitness function used in the present study is a combination of maximization of classification accuracy and minimization of the cost of an ANN.

### 2.5.1. ANN Classifier

A back propagation neural network (BPNN) with one hidden layer is used for the present study. The numbers of input features are 15, 15, 15, 14 and 11 for  $M_a$ ,  $M_p$ ,  $NP_a$ ,  $NP_p$  and  $T_c$  respectively. After a number of trial and error, the optimum number of hidden layer neuron found for the network is 11. The output node is set as 1 which is the efficiency of siRNA. The 'log-sigmoid' activation function has been used for both hidden layer and output layer. In each training and testing process, 70% of the dataset is used for training, 15% for testing and rest 15% for validation. The performance function is measured by MSE.

### 2.5.2. Cost Evaluation

The following steps are involved in calculating the cost used in the present study.

- Step 1: Read the data element.
- Step 2: Create permutation using random keys (sort).
- Step 3: Select features by using feature selection method.
- Step 4: Initialize the weights of Training and Testing Errors.

Step 5: Create and train ANN.

Step 6: Calculate the overall error (E) by using the equation (3)

$$E = wTrain * \text{training performance} + wTest * \text{testing performance} \tag{3}$$

where,  $wTrain$  = weight of train and  $wTest$  = weight of test

Step 7: Repeat step 5 and step 6 till the error is minimized.

Step 8: Calculating the final cost by ‘mean’ of ‘E’ obtained from equation (3).

### 3. RESULTS AND DISCUSSIONS

#### 3.1. Selection of Parameter

The population size of PSO for the present experiment has been considered between 10 and 40. The reason for a lower population size is that it significantly lowers the computation time. This is because during initialization, all the particles must be in the feasible space. Randomly initialized particles are not always in the feasible space where feasible space are the positions in which a particle can fly freely, keeping the constraints of the problem intake [35]. Initialization may take a longer time if the population is too large. However, for complex cases, a larger population size is preferred. In PSO, there are not many parameters that need to be tuned. Only the following parameters need to be adjusted: maximum velocity  $VelMax$ , inertia weight  $w$ , acceleration coefficient  $C1$  and  $C2$ . In GA, the following parameters and operators need to be adjusted: Crossover rate ( $Pc$ ), Mutation rate ( $Pm$ ), Selection operator and Crossover operator. The parameters used for the present implementation are tabulated in Table 1.

Table1. Parameter value of PSO and GA-PSO

Parameter	Values
Maximum velocity of the particle( $VelMax$ )	$VarMax-VarMin$
Inertia Weight ( $w$ )	1.0
Generation Gap	0.9
Crossover rate ( $Pc$ )	0.7
Mutation rate ( $Pm$ )	0.5
Crossover operator	Single point
Mutation operator	Real value
PSO iteration	10
GA generation	10

#### 3.2. Experimental Results

The experimental dataset of plant siRNA is divided into two groups, considering 80% samples for training and 20% samples for testing.

The results of feature selection using PSO and GA-PSO for five different properties viz.,  $M_a$ ,  $M_p$ ,  $NP_a$ ,  $NP_p$ ,  $T_c$  with 15, 15, 15, 14 and 11 features respectively have been tabulated in Table 2 and Table 3. The results show the reduced numbers of features for individual properties with best cost (lower the value better is the cost) against different numbers of iteration and population.

Table 2. Results of Feature Selection using PCO

	Properties	I.F.	R.F.	Positions	Population	No. of Iteration	Best Cost
PSO	$M_a$	15	11	1,2,3,5,6,7,8,10,11,13,14	10	10	133.180
			12	1,2,3,4,5,6,7,9,11,12,13,15	20	10	132.555
			12	2,3,4,5,6,9,10,11,12,13,14,15	30	10	133.612
			11	1,2,3,4,5,7,10,11,12,13,15	40	10	132.512

	M <sub>p</sub>	15	11	1,2,5,6,7,8,9,11,12,14,15	10	10	125.312
			11	1,2,3,5,6,7,8,10,12,13,15	20	10	121.833
			12	1,2,4,5,7,8,9,10,11,12,13,12	30	10	120.874
			<b>12</b>	<b>1,2,3,4,5,7,8,10,11,13,14,15</b>	<b>40</b>	<b>10</b>	<b>119.828</b>
	NP <sub>a</sub>	15	14	1,2,3,5,6,7,8,9,10,11,12,13,14,15	10	10	55.743
			13	1,2,3,4,6,7,8,9,11,12,13,14,15	20	<b>10</b>	54.219
			13	1,2,3,4,6,8,9,10,11,12,13,14,15	30	10	53.186
			<b>14</b>	<b>1,2,3,4,5,6,7,8,9,10,11,13,14,15</b>	<b>40</b>	10	<b>51.118</b>
	NP <sub>p</sub>	14	13	1,2,3,4,5,6,7,8,10,11,12,13,14	10	10	91.122
			12	1,2,3,4,5,6,7,8,11,12,13,14	20	<b>10</b>	93.967
			13	1,2,3,4,5,6,7,8,10,11,12,13,14	30	10	96.624
			<b>13</b>	<b>1,2,3,4,5,6,7,8,10,11,12,13,14</b>	<b>40</b>	10	<b>89.163</b>
	T <sub>c</sub>	11	8	1,3,4,5,6,7,9,11	10	<b>10</b>	64.419
			8	1,2,3,4,5,6,7,11	20	10	66.050
			8	1,3,4,5,6,8,9,11	30	10	64.209
			<b>9</b>	<b>1,2,3,4,5,6,7,8,9</b>	<b>40</b>	10	<b>60.606</b>

I.F: Initial Features, R.F: Reduced Features

The best case results for each properties are shown in bold.

Table 2. Results of Feature Selection using GA-PSO

	Properties	I.F.	R.F.	Positions	Population	No. of Iteration	Best Cost
	M <sub>a</sub>	15	8	1,3,5,7,10,11,13,15	10	10	101.505
			10	1,2,3,5,6,10,11,13,14,15	20	10	102.735
			12	1,2,3,5,7,9,10,11,12,13,14,15	30	10	102.262
			<b>10</b>	<b>1,2,3,4,5,10,11,12,13,15</b>	<b>40</b>	<b>10</b>	<b>101.006</b>
	M <sub>p</sub>	15	11	2,3,4,5,7,8,11,12,13,14,15	10	10	111.963
			12	1,2,3,5,6,8,9,10,11,13,14,15	20	10	108.852
			<b>9</b>	<b>1,2,3,5,8,11,13,14,15</b>	<b>30</b>	<b>10</b>	<b>105.858</b>
			11	1,2,5,6,7,8,10,11,13,14,15	40	10	109.478
	NP <sub>a</sub>	15	11	1,2,3,5,6,7,8,10,11,14,15	10	10	41.623
			11	1,2,3,4,5,6,7,8,9,11,15	20	10	36.384
			<b>13</b>	<b>1,2,3,4,5,6,7,8,9,11,13,14,15</b>	<b>30</b>	<b>10</b>	<b>35.442</b>
			13	1,2,3,4,5,6,7,9,10,11,12,14,15	40	10	36.522
	NP <sub>p</sub>	14	10	1,2,3,4,5,7,8,11,13,14	10	10	70.699
			10	1,2,5,6,7,8,10,11,13,14	20	10	68.866
			<b>12</b>	<b>1,2,3,4,5,6,7,9,11,12,13,14</b>	<b>30</b>	<b>10</b>	<b>68.169</b>
			13	1,2,3,4,5,6,7,8,10,11,12,13,14	40	10	69.269
T <sub>c</sub>	11	<b>7</b>	<b>1,3,4,5,6,8,11</b>	<b>10</b>	<b>10</b>	<b>47.528</b>	
		8	1,2,3,4,5,6,8,9	20	10	48.674	
		8	1,3,4,6,8,9,10,11	30	10	48.839	
		9	1,2,3,5,6,7,9,10,11	40	10	48.369	

I.F: Initial Features, R.F: Reduced Features

The best case results for each properties are shown in bold.

From the Table 2, it has been found that the best cost in terms of minimum value for M<sub>a</sub>, M<sub>p</sub>, NP<sub>a</sub>, NP<sub>p</sub> and T<sub>c</sub> using PSO are 132.512, 119.828, 51.118, 89.163 and 60.606 respectively, whereas, the features of motif avoided is reduced from 15 to 11, motif preferred from 15 to 12, nucleotide

position avoided from 15 to 14, nucleotide position preferred from 14 to 13 and the features of thermodynamic properties reduced from 11 to 9.

The performance of PSO is further optimized by applying GA to PSO forming a hybrid GA-PSO model and the best case results in each of the properties for proposed hybrid GA-PSO is tabulated in Table 3.

From the experimental results of the proposed hybrid GA-PSO it has been found that the number of features subset obtained for  $M_a$ ,  $M_p$ ,  $NP_a$ ,  $NP_p$  and  $T_c$  properties have been reduced compared to PSO and the cost is also reduced.

A comparison of the best case result in terms of best cost (lower the value, better is the cost) is tabulated in Table 4.

Table 3. Comparison of PSO and GA-PSO in terms of Best Cost

Properties	IF	PSO		GA-PSO	
		RF	Best Cost	RF	Best Cost
$M_a$	15	11	132.512	10	101.006
$M_p$	15	12	119.828	9	105.858
$NP_a$	15	14	51.118	13	35.442
$NP_p$	14	13	89.163	12	68.169
$T_c$	11	9	60.606	7	47.528

I.F: Initial Features, R.F: Reduced Features

A hybrid GA-BPNN model as proposed by Sarmah et al. [13] has been employed for training and testing of the reduced features obtained from feature selection using PSO and GA-PSO. These new reduced features are considered as the input parameter to the model for individual properties i.e.,  $M_a$ ,  $M_p$ ,  $NP_a$ ,  $NP_p$  and  $T_c$ . The results so obtained are tabulated in Table 5 and Table 6.

Table 4. Training and Testing results of GA-BPNN against the reduced feature subset obtained by using PSO algorithm

Properties	Training		Testing	
	RMSE	CC	RMSE	CC
$M_a$	0.1504	0.4596	0.1549	0.4286
$M_p$	0.1550	0.3865	0.1621	0.3663
$NP_a$	0.0830	0.8671	0.0845	0.8637
$NP_p$	0.1239	0.7045	0.1244	0.7031
$T_c$	0.1012	0.7814	0.1031	0.7690

Table 5. Training and Testing results of GA-BPNN against the reduced subset obtained by using GA-PSO algorithm

Properties	Training		Testing	
	RMSE	CC	RMSE	CC
$M_a$	0.1486	0.4629	0.1516	0.4497
$M_p$	0.1507	0.4041	0.1596	0.3973
$NP_a$	0.0831	0.8829	0.0840	0.8767
$NP_p$	0.1233	0.7149	0.1241	0.7093
$T_c$	0.0987	0.8021	0.1010	0.7957

A comparative analysis has been carried out for the best case testing results of GA-BPNN obtained in Sarmah et al. [13] with original features and the best case results obtained in Table 5 and Table 6 using PSO and Hybrid GA-PSO feature selection techniques are tabulated in Table 7.

Table 7. Comparison of testing results of GA-BPNN with original feature and with reduced features

Properties	Original features			Reduced features obtained from PSO algorithm			Reduced features obtained from GA-PSO algorithm		
	Features	Testing RMSE	Testing CC	Features	Testing RMSE	Testing CC	Features	Testing RMSE	Testing CC
$M_a$	15	0.1521	0.4412	11	0.1549	0.4286	<b>10</b>	<b>0.1516</b>	<b>0.4497</b>
$M_p$	15	0.1626	0.3358	12	0.1621	0.3663	<b>9</b>	<b>0.1596</b>	<b>0.3973</b>
$NP_a$	15	0.0874	0.8314	14	0.0845	0.8637	<b>13</b>	<b>0.0840</b>	<b>0.8767</b>
$NP_p$	14	0.1245	0.6778	13	0.1244	0.7031	<b>12</b>	<b>0.1241</b>	<b>0.7093</b>
$T_c$	11	0.1037	0.7495	9	0.1031	0.7690	<b>7</b>	<b>0.1010</b>	<b>0.7957</b>

From Table 7, it has been found that the hybrid GA-BPNN results of the reduced features obtained from hybrid GA-PSO algorithm outperformed the results obtained with methods with original features and features obtained with PSO.

The best results for training, testing and cost efficacy prediction of hybrid GA-PSO for all the properties viz.,  $M_a$ ,  $M_p$ ,  $NP_a$ ,  $NP_p$  and  $T_c$  are plotted in Figure 5-9.

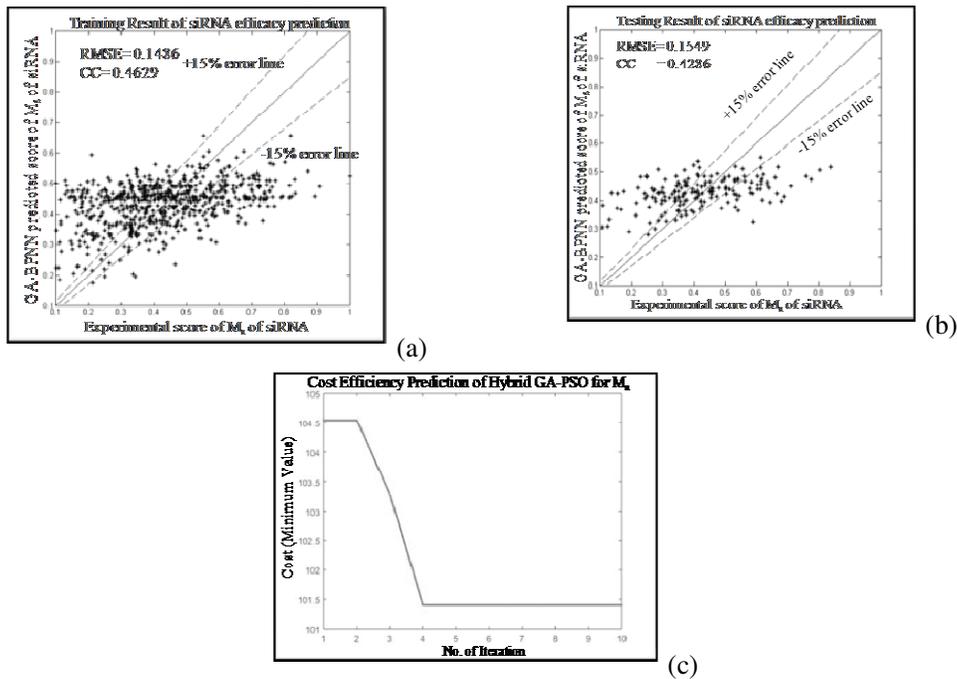


Figure 5. Results of (a) GA-BPNN Training (b) GA-BPNN Testing (c) Cost Efficiency Prediction of Hybrid GA-PSO for  $M_a$

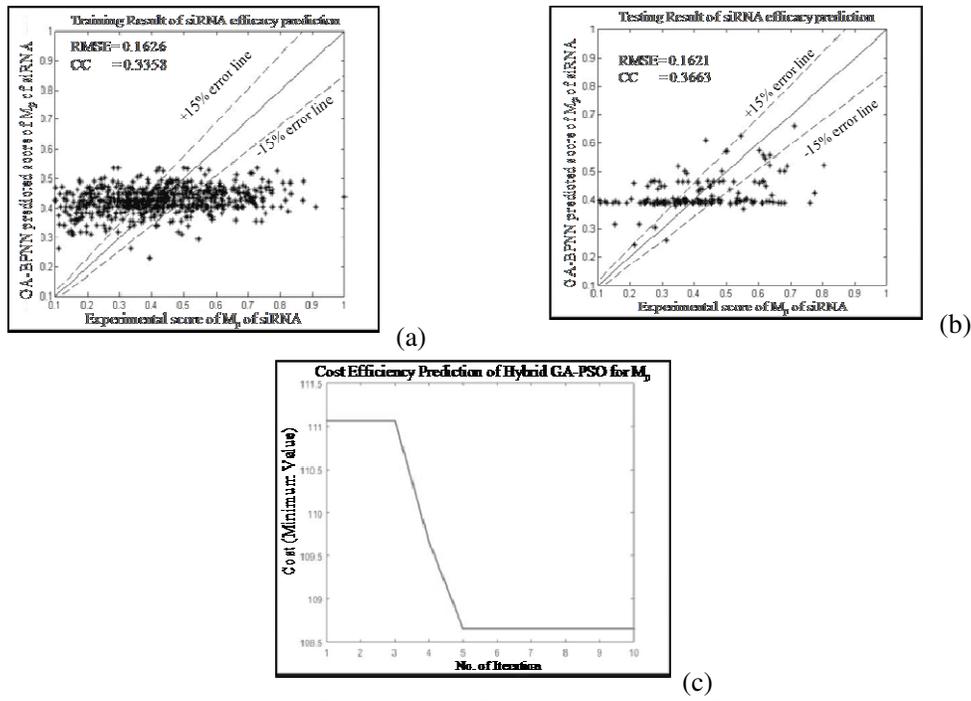


Figure 6. Results of (a) GA-BPNN Training (b) GA-BPNN Testing (c) Cost Efficiency Prediction of Hybrid GA-PSO for  $M_p$

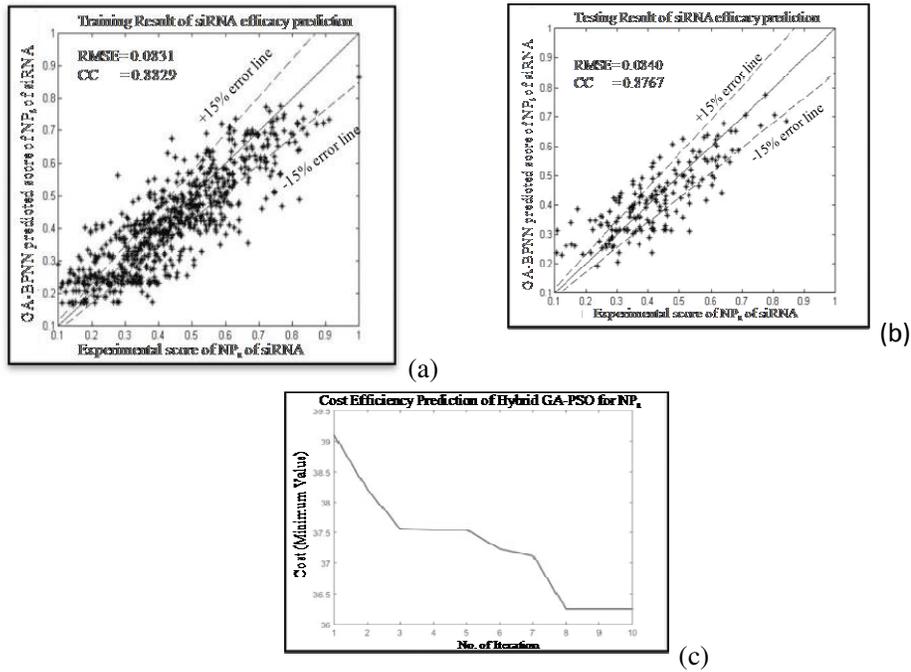


Figure 7. Results of (a) GA-BPNN Training (b) GA-BPNN Testing (c) Cost Efficiency Prediction of Hybrid GA-PSO for  $NP_a$

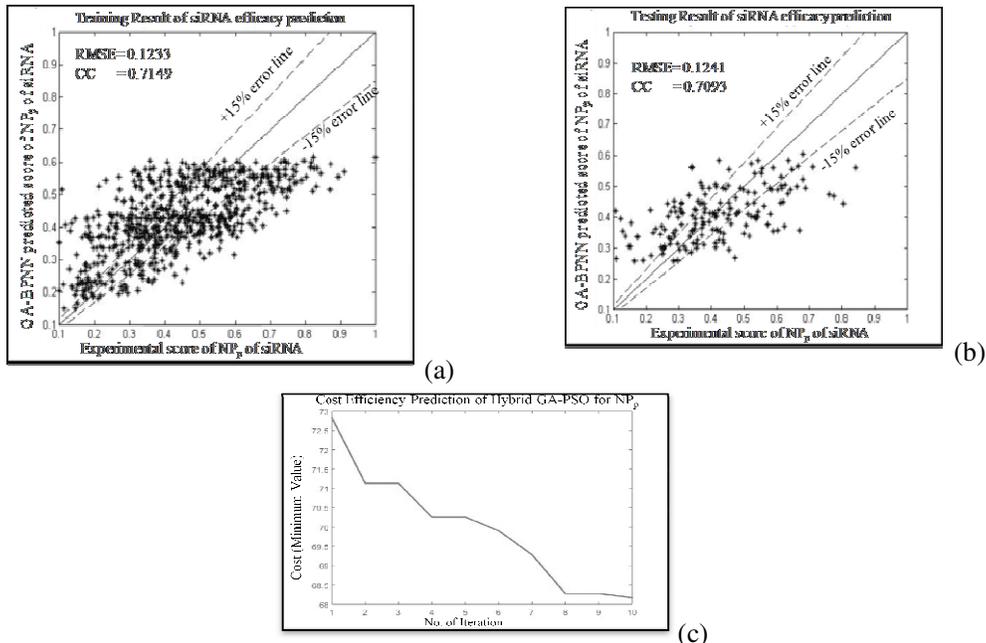


Figure 8. Results of (a) GA-BPNN Training (b) GA-BPNN Testing (c) Cost Efficacy Prediction of Hybrid GA-PSO for  $NP_p$

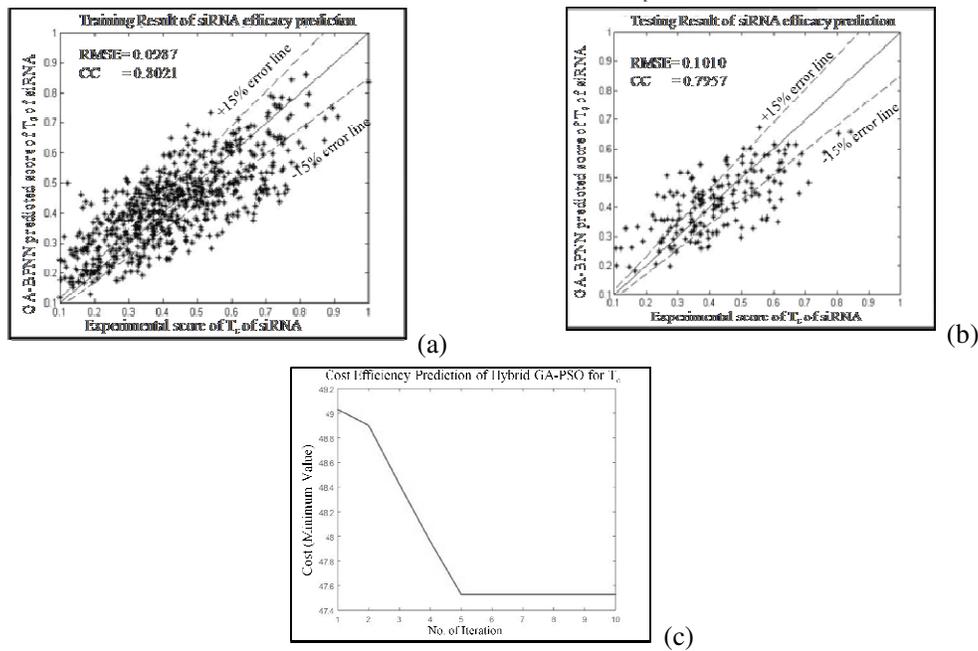


Figure 9. Results of (a) GA-BPNN Training (b) GA-BPNN Testing (c) Cost Efficacy Prediction of Hybrid GA-PSO for  $T_c$

To further optimize the designing of productive siRNA and their silencing efficiency, combination of different properties with their reduced features obtained from proposed hybrid GA-PSO model has been implemented. The performance analysis of these feature combinations have been evaluated by using the proposed hybrid GA-BPNN model given by Sarmah et al. [13]. The combination of properties and the number of combined reduced features are tabulated in Table 8.

Table 8. List of combined properties and associated reduced features

No.	Combined Properties	No. of combined Features
1.	$M_a + M_p$	19 (10 from $M_a$ and 9 from $M_p$ )
2.	$NP_a + NP_p$	25 (13 from $NP_a$ and 12 from $NP_p$ )
3.	$M_a + M_p + T_c$	26 (10 from $M_a$ , 9 from $M_p$ and 7 from $T_c$ )
4.	$NP_a + NP_p + T_c$	32 (13 from $NP_a$ , 12 from $NP_p$ and 7 from $T_c$ )
5.	$M_a + M_p + NP_a + NP_p$	44 (10 from $M_a$ , 9 from $M_p$ , 13 from $NP_a$ and 12 from $NP_p$ )
6.	$M_a + M_p + NP_a + NP_p + T_c$	51 (10 from $M_a$ , 9 from $M_p$ , 13 from $NP_a$ , 12 from $NP_p$ and 7 from $T_c$ )

#### 4. PERFORMANCE ANALYSIS OF FEATURE COMBINATION

The performances of six feature combination methods have been analyzed by training and testing with the proposed GA-BPNN hybrid model. The experimental results obtained with GA-BPNN for six feature combination predictive methods as shown in Table 8 are tabulated in Table 9-14. The best case results of training and testing for each combined properties is plotted in Figure 10-15.

Table 9. GA-BPNN Training and Testing results for  $M_a+M_p$

Hybrid method	Epoch	Training			Testing		
		MAE	RMSE	CC	MAE	RMSE	CC
$M_a + M_p$	100	0.1211	0.1571	0.4110	0.1235	0.1583	0.4096
	200	0.1312	0.1618	0.3764	0.1321	0.1627	0.3751
	300	0.1295	0.1589	0.3866	0.1310	0.1601	0.3850
	400	0.1316	0.1620	0.3661	0.1324	0.1638	0.3648
	500	0.1286	0.1554	0.4106	0.1298	0.1561	0.4097
	600	0.1263	0.1549	0.4211	0.1279	0.1559	0.4201
	<b>700</b>	<b>0.1168</b>	<b>0.1480</b>	<b>0.4661</b>	<b>0.1179</b>	<b>0.1493</b>	<b>0.4648</b>
	800	0.1173	0.1495	0.4615	0.1191	0.1508	0.4598
	900	0.1181	0.1506	0.4603	0.1198	0.1521	0.4587
	1000	0.1191	0.1511	0.4544	0.1206	0.1529	0.4526
	1100	0.1201	0.1565	0.4120	0.1216	0.1581	0.4101

Table 10. GA-BPNN Training and Testing results for  $NP_a+NP_p$

Hybrid method	Epoch	Training			Testing		
		MAE	RMSE	CC	MAE	RMSE	CC
$NP_a + NP_p$	100	0.0682	0.0844	0.8714	0.0691	0.0856	0.8701
	200	0.0681	0.0840	0.8703	0.0690	0.0854	0.8694
	300	0.0685	0.0845	0.8698	0.0696	0.0857	0.8685
	<b>400</b>	<b>0.0666</b>	<b>0.0819</b>	<b>0.8745</b>	<b>0.0673</b>	<b>0.0828</b>	<b>0.8738</b>
	500	0.0688	0.0851	0.8675	0.0697	0.0863	0.8659
	600	0.0684	0.0850	0.8699	0.0694	0.0861	0.8678
	700	0.0677	0.0853	0.8673	0.0684	0.0865	0.8657
	800	0.0691	0.0857	0.8658	0.0701	0.0877	0.8643
	900	0.0688	0.0851	0.8675	0.0697	0.0863	0.8660
	1000	0.0695	0.0863	0.8648	0.0708	0.0878	0.8636
	1100	0.0701	0.0874	0.8629	0.0718	0.0881	0.8618

Table 11. GA-BPNN Training and Testing results for  $M_a + M_p+T_c$

Hybrid method	Epoch	Training			Testing		
		MAE	RMSE	CC	MAE	RMSE	CC
	100	0.0876	0.1088	0.7705	0.0898	0.1112	0.7684

$M_a + M_p + T_c$	200	0.0861	0.1079	0.7714	0.0887	0.1106	0.7691
	300	0.0874	0.1084	0.7702	0.0896	0.1109	0.7681
	400	0.0866	0.1081	0.7709	0.0891	0.1108	0.7685
	<b>500</b>	<b>0.0844</b>	<b>0.1054</b>	<b>0.7814</b>	<b>0.0868</b>	<b>0.1064</b>	<b>0.7793</b>
	600	0.0851	0.1060	0.7806	0.0872	0.1079	0.7787
	700	0.0879	0.1096	0.7694	0.0902	0.1121	0.7689
	800	0.0888	0.1107	0.7687	0.0912	0.1132	0.7671
	900	0.0849	0.1059	0.7807	0.0870	0.1078	0.7793
	1000	0.0862	0.1082	0.7717	0.0890	0.1109	0.7698
	1100	0.0857	0.1063	0.7794	0.0879	0.1096	0.7784

Table 12. GA-BPNN Training and Testing results for  $NP_a + NP_p + T_c$

Hybrid method	Epoch	Training			Testing		
		MAE	RMSE	CC	MAE	RMSE	CC
$NP_a + NP_p + T_c$	100	0.0394	0.0492	0.9569	0.0441	0.0512	0.9549
	200	0.0417	0.0522	0.9532	0.0468	0.0541	0.9519
	300	0.0420	0.0525	0.9531	0.0473	0.0548	0.9516
	400	0.0404	0.0520	0.9553	0.0451	0.0539	0.9521
	<b>500</b>	<b>0.0390</b>	<b>0.0486</b>	<b>0.9579</b>	<b>0.0436</b>	<b>0.0506</b>	<b>0.9561</b>
	600	0.0398	0.0496	0.9568	0.0451	0.0518	0.9554
	700	0.0409	0.0529	0.9526	0.0468	0.0545	0.9511
	800	0.0412	0.0517	0.9536	0.0471	0.0533	0.9522
	900	0.0426	0.0531	0.9519	0.0481	0.0567	0.9509
	1000	0.0431	0.0544	0.9493	0.0493	0.0574	0.9481
1100	0.0415	0.0521	0.9529	0.0472	0.0561	0.9512	

Table 13. GA-BPNN Training and Testing results for  $M_a + M_p + NP_a + NP_p$

Hybrid method	Epoch	Training			Testing		
		MAE	RMSE	CC	MAE	RMSE	CC
$M_a + M_p + NP_a + NP_p$	100	0.0600	0.0734	0.9008	0.0620	0.0746	0.8961
	200	0.0607	0.0746	0.8994	0.0621	0.0761	0.8973
	<b>300</b>	<b>0.0590</b>	<b>0.0727</b>	<b>0.9047</b>	<b>0.0601</b>	<b>0.0737</b>	<b>0.9011</b>
	400	0.0610	0.0751	0.9007	0.0628	0.0772	0.8861
	500	0.0598	0.0735	0.9038	0.0618	0.0746	0.9002
	600	0.0617	0.0761	0.8997	0.0639	0.0784	0.8836
	700	0.0595	0.0731	0.9029	0.0611	0.0743	0.9017
	800	0.0604	0.0738	0.9001	0.6226	0.0752	0.8974
	900	0.0593	0.0729	0.9039	0.0609	0.0740	0.9004
	1000	0.0604	0.0741	0.8998	0.0624	0.0755	0.8971
1100	0.0621	0.0758	0.8937	0.0653	0.0769	0.8817	

Table 14. GA-BPNN Training and Testing results for  $M_a + M_p + NP_a + NP_p + T_c$

Hybrid method	Epoch	Training			Testing		
		MAE	RMSE	CC	MAE	RMSE	CC
$M_a + M_p + NP_a + NP_p + T_c$	100	0.0357	0.0454	0.9634	0.0371	0.0471	0.9605
	200	0.0409	0.0517	0.9531	0.0420	0.0532	0.9510
	300	0.0394	0.0501	0.9556	0.0411	0.0520	0.9529
	400	0.0386	0.0489	0.9586	0.0399	0.0502	0.9557
	<b>500</b>	<b>0.0350</b>	<b>0.0451</b>	<b>0.9641</b>	<b>0.0367</b>	<b>0.0466</b>	<b>0.9611</b>
	600	0.0380	0.0486	0.9594	0.0393	0.0497	0.9574
	700	0.0387	0.0491	0.9585	0.0403	0.0511	0.9558
	800	0.0413	0.0519	0.9528	0.0427	0.0537	0.9502
	900	0.0359	0.0460	0.9621	0.0372	0.0483	0.9594
	1000	0.0368	0.0467	0.9625	0.0380	0.0491	0.9601
	1100	0.0386	0.0497	0.9572	0.0398	0.0519	0.9547

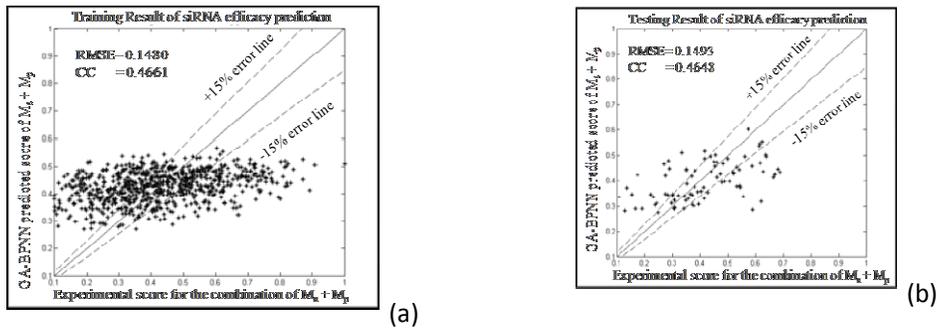


Figure 10. GA-BPNN (a) Training (b) Testing for  $M_a + M_p$

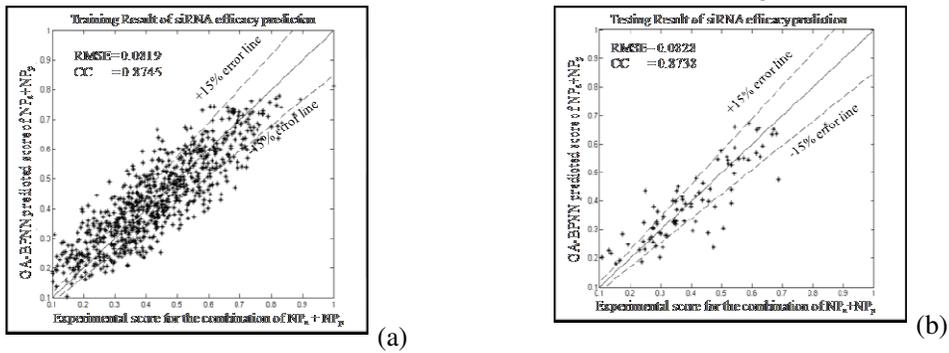


Figure 11. GA-BPNN (a) Training (b) Testing for  $NP_a + NP_p$

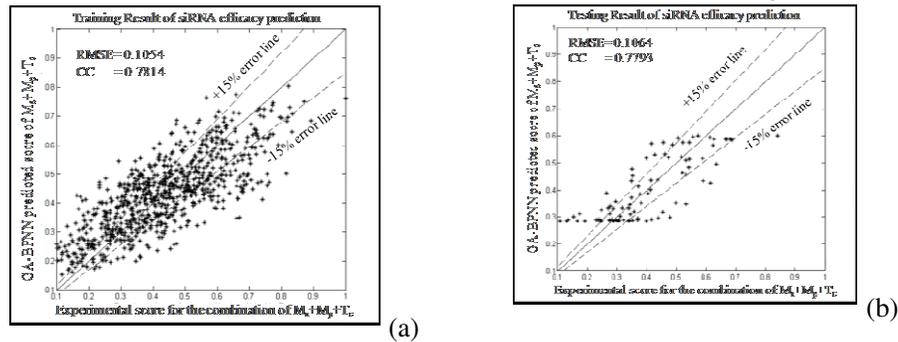


Figure 12. GA-BPNN (a) Training (b) Testing for  $M_a + M_p + T_c$

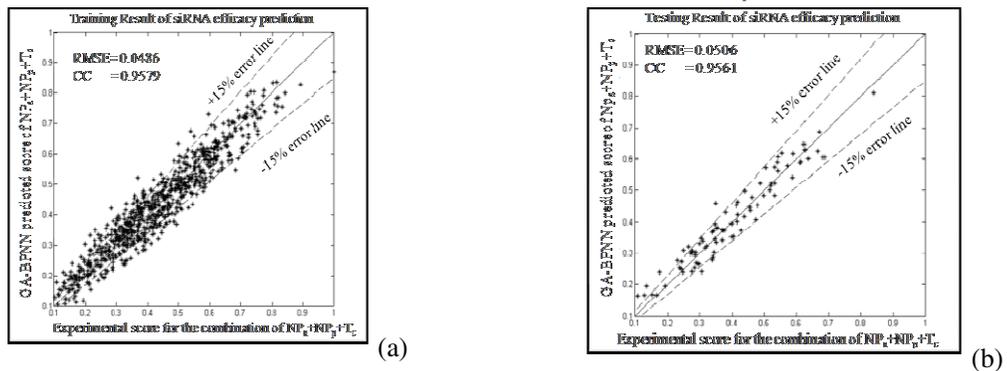


Figure 13. GA-BPNN (a) Training (b) Testing results for  $NP_a + NP_p + T_c$

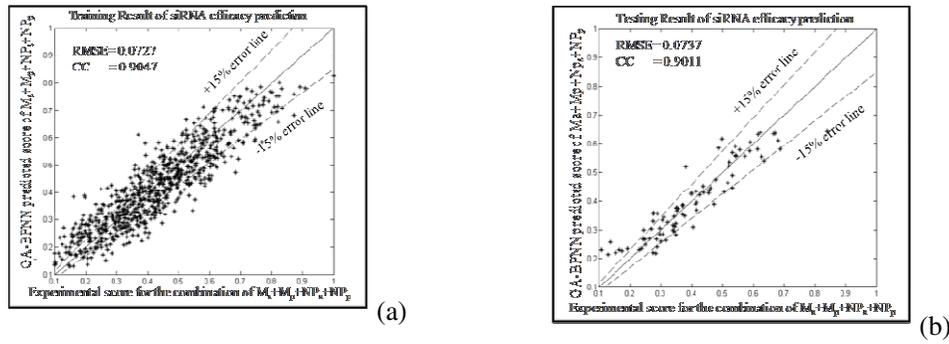


Figure 14. GA-BPNN (a) Training (b) Testing results for  $M_a + M_p + NP_a + NP_p$

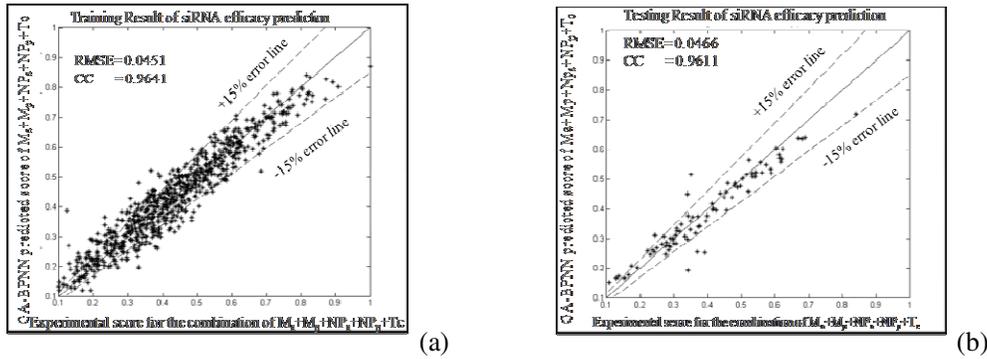


Figure 15. GA-BPNN (a) Training and (b) Testing results for  $M_a + M_p + NP_a + NP_p + T_c$

Table 15. Comparison of Results of Feature Combinations

Combined Properties	Training		Testing	
	RMSE	CC	RMSE	CC
$M_a + M_p$	0.1480	0.4661	0.1493	0.4648
$NP_a + NP_p$	0.0819	0.8745	0.0828	0.8738
$M_a + M_p + T_c$	0.1054	0.7814	0.1064	0.7793
$NP_a + NP_p + T_c$	0.0486	0.9579	0.0506	0.9561
$M_a + M_p + NP_a + NP_p$	0.0727	0.9047	0.0737	0.9011
$M_a + M_p + NP_a + NP_p + T_c$	<b>0.0451</b>	<b>0.9641</b>	<b>0.0466</b>	<b>0.9611</b>

From Table 15 it is found that the combination of different properties show high correlation coefficient (CC) and low root mean square error (RMSE) compared to the results of individual properties. It is also noticeable that the CC value for the combined features of  $M_a$  and  $M_p$  is lower compared to other combination of features but when  $T_c$  is added to  $M_a$  and  $M_p$  combination, the CC value goes up from 0.4648 to 0.7793 in testing case. The similar case is also seen in case of  $N_a$  and  $N_p$  where the addition of  $T_c$  to  $M_a$  and  $M_p$  combination brings up the GA-BPNN testing value of CC from 0.8738 to 0.9561. It is also noticeable that although  $M_a + M_p$  is not performing well but when added with  $NP_a$ ,  $NP_p$  and  $T_c$ , it shows highest correlation coefficient and lowest RMSE value among all other combinations used in this study. So, it may be concluded that all the combination of properties viz.,  $M_a$ ,  $M_p$ ,  $NP_a$ ,  $NP_p$  and  $T_c$  and their associated features obtained from ANN based hybrid GA-PSO feature selection approach is an important aspect for designing an effective siRNA in plant pathogens for the considered dataset.

## 5. CONCLUSIONS

In this study, ANN based feature selection approach using hybrid GA-PSO has been implemented by integrating the merits of both GA and PSO to reduce the number of features and the cost

associated with the features. With the investigation presented in this paper, it is concluded that wrapper feature selection method using hybrid GA-PSO approach based on ANN (BPNN) algorithm have shown better results in term accuracy of the model and minimization of cost as against general PSO algorithm. The training and testing results of hybrid GA-BPNN method with the reduced features obtained from hybrid GA-PSO model have yielded better siRNA efficacy prediction results compared to the original features for the considered plant dataset. Furthermore, to find the optimal algorithm for designing efficient siRNA sequences further experimentation has been carried out with the combination of different features. It may also be concluded with the discussion made in the previous section that the combination of  $M_a$ ,  $M_p$ ,  $NP_a$ ,  $NP_p$  and  $T_c$  properties with their reduced features obtained from ANN based hybrid GA-PSO model is an important aspect for designing an effective siRNA in plant pathogens for the considered dataset.

Further experimentation may be carried out on different plant dataset considering different properties.

## REFERENCES

- [1] McManus, M.T., and Sharp, P.A.: 'Gene silencing in mammals by small interfering RNAs', *Nat Rev Genet*, 2002, 3
- [2] Hannon, G.J., and Rossi, J.J.: 'Unlocking the potential of the human genome with RNA interference', *Nature*, 2004, 431
- [3] Lu, Z.J., and Mathews, D.H.: 'OligoWalk: an online siRNA design tool utilizing hybridization thermodynamics', *Nucleic Acids Res*, 2008, 36, (Web Server issue), pp. W104-108
- [4] Saetrom, P., and Snove, O.: 'A comparison of siRNA efficacy predictors', *Biochem Biophys Res Commun*, 2004, 321
- [5] Basiri, M.E., and Nemati, S.: 'A novel hybrid ACO-GA algorithm for text feature selection', in Editor (Ed.)<sup>(Eds.)</sup>: 'Book A novel hybrid ACO-GA algorithm for text feature selection' (2009, edn.), pp. 2561-2568
- [6] Aghdam, M.H., Ghasem-Aghaee, N., and Basiri, M.E.: 'Text feature selection using ant colony optimization', *Expert Systems with Applications*, 2009, 36, (3, Part 2), pp. 6843-6853
- [7] Yang, J., and Honavar, V.: 'Feature subset selection using a genetic algorithm', *IEEE Intelligent Systems and their Applications*, 1998, 13, (2), pp. 44-49
- [8] Zhao, X., Li, D., Yang, B., Ma, C., Zhu, Y., and Chen, H.: 'Feature selection based on improved ant colony optimization for online detection of foreign fiber in cotton', *Applied Soft Computing*, 2014, 24, pp. 585-596
- [9] Liu, Y., Qin, Z., Xu, Z., and He, X.: 'Feature Selection with Particle Swarms', in Zhang, J., He, J.-H., and Fu, Y. (Eds.): 'Computational and Information Science: First International Symposium, CIS 2004, Shanghai, China, December 16-18, 2004. Proceedings' (Springer Berlin Heidelberg, 2005), pp. 425-430
- [10] Mojtaba Ahmadieh, K., Mohammad, T., and Mahdi Aliyari, S.: 'A novel binary particle swarm optimization', in Editor (Ed.)<sup>(Eds.)</sup>: 'Book A novel binary particle swarm optimization' (2007, edn.), pp. 1-6
- [11] Prasad, Y., Biswas, K.K., and Jain, C.K.: 'SVM Classifier Based Feature Selection Using GA, ACO and PSO for siRNA Design', in Tan, Y., Shi, Y., and Tan, K.C. (Eds.): 'Advances in Swarm Intelligence: First International Conference, ICSI 2010, Beijing, China, June 12-15, 2010, Proceedings, Part II' (Springer Berlin Heidelberg, 2010), pp. 307-314
- [12] Jain, C.K., and Prasad, Y.: 'Feature selection for siRNA efficacy prediction using natural computation', in Editor (Ed.)<sup>(Eds.)</sup>: 'Book Feature selection for siRNA efficacy prediction using natural computation' (2009, edn.), pp. 1759-1764
- [13] Sarmah, R., Begum, S. A. and Modi, M.K.: 'A hybrid GA-ANN approach in building efficient model for prediction of siRNA knockdown efficiency in plant pathogens', *International Journal of Computer Science and Information Security*, 2016, 14, (12), pp. 15
- [14] Sánchez-Marroño, N., Alonso-Betanzos, A., and Tombilla-Sanromán, M.: 'Filter Methods for Feature Selection – A Comparative Study', in Yin, H., Tino, P., Corchado, E., Byrne, W., and Yao, X. (Eds.): 'Intelligent Data Engineering and Automated Learning - IDEAL 2007: 8th

- International Conference, Birmingham, UK, December 16-19, 2007. Proceedings' (Springer Berlin Heidelberg, 2007), pp. 178-187
- [15] Liu, H., and Zhao, Z.: 'Manipulating Data and Dimension Reduction Methods: Feature Selection', in Meyers, R.A. (Ed.): 'Computational Complexity: Theory, Techniques, and Applications' (Springer New York, 2012), pp. 1790-1800
- [16] Liu, Q., Xu, Q., Zheng, V.W., Xue, H., Cao, Z., and Yang, Q.: 'Multi-task learning for cross-platform siRNA efficacy prediction: an in-silico study', BMC Bioinformatics, 2010, 11
- [17] Xue, B., Zhang, M., Browne, W.N., and Yao, X.: 'A Survey on Evolutionary Computation Approaches to Feature Selection', IEEE Transactions on Evolutionary Computation, 2016, 20, (4), pp. 606-626
- [18] A Kachitvichyanukul, V.: 'Recent Advances in Adaptive Particle Swarm Optimization Algorithms', in Editor (Ed.)^(Eds.): 'Book Recent Advances in Adaptive Particle Swarm Optimization Algorithms' (2008, edn.), pp.
- [19] Arumugam, M.S., and Rao, M.V.C.: 'On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square (RMS) variants for computing optimal control of a class of hybrid systems', Applied Soft Computing, 2008, 8, (1), pp. 324-336
- [20] Chen, W.N., Zhang, J., Chung, H.S.H., Zhong, W.L., Wu, W.G., and Shi, Y.h.: 'A Novel Set-Based Particle Swarm Optimization Method for Discrete Optimization Problems', IEEE Transactions on Evolutionary Computation, 2010, 14, (2), pp. 278-300
- [21] <http://www.swarmintelligence.org/tutorials.php>
- [22] Kennedy, J., and Eberhart, R.: 'Particle Swarm Optimization', 1995
- [23] Li, L., and Zhang, Y.: 'An Improved Genetic Algorithm for the Traveling Salesman Problem', in Huang, D.-S., Heutte, L., and Loog, M. (Eds.): 'Advanced Intelligent Computing Theories and Applications. With Aspects of Contemporary Intelligent Computing Techniques: Third International Conference on Intelligent Computing, ICIC 2007, Qingdao, China, August 21-24, 2007. Proceedings' (Springer Berlin Heidelberg, 2007), pp. 208-216
- [24] Liao, Y.-F., Yau, D.-H., and Chen, C.-L.: 'Evolutionary algorithm to traveling salesman problems', Computers & Mathematics with Applications, 2012, 64, (5), pp. 788-797
- [25] Shelokar, P.S., Siarry, P., Jayaraman, V.K., and Kulkarni, B.D.: 'Particle swarm and ant colony algorithms hybridized for improved continuous optimization', Applied Mathematics and Computation, 2007, 188, (1), pp. 129-142
- [26] Kao, Y.-T., and Zahara, E.: 'A hybrid genetic algorithm and particle swarm optimization for multimodal functions', Applied Soft Computing, 2008, 8, (2), pp. 849-857
- [27] Goldberg, D.E.: 'Genetic Algorithms in Search, Optimization and Machine Learning' (Addison-Wesley Longman Publishing Co., Inc., 1989. 1989)
- [28] Man, K.F., TANG, K.S., and Kwong, S.: 'Genetic Algorithms: Concepts and Designs' (Springer London, 2001. 2001)
- [29] Voratas, K.: 'Comparison of Three Evolutionary Algorithms: GA, PSO, and DE', Industrial Engineering & Management Systems, 2012, 11, (3), pp. 215-223
- [30] Hook, J.V., Sahin, F., and Arnavut, Z.: 'Application of Particle Swarm Optimization for Traveling Salesman Problem to lossless compression of color palette images', in Editor (Ed.)^(Eds.): 'Book Application of Particle Swarm Optimization for Traveling Salesman Problem to lossless compression of color palette images' (2008, edn.), pp. 1-5
- [31] Zhang, Y., Gong, D.w., and Cheng, J.: 'Multi-Objective Particle Swarm Optimization Approach for Cost-Based Feature Selection in Classification', IEEE/ACM Transactions on Computational Biology and Bioinformatics, 2017, 14, (1), pp. 64-75
- [32] Turney, P.D.: 'Cost-Sensitive Classification: Empirical Evaluation of a Hybrid Genetic Decision Tree Induction Algorithm', Journal of Artificial Intelligence Research, 1995, 2, pp. 41
- [33] Min, F., Hu, Q., and Zhu, W.: 'Feature selection with test cost constraint', International Journal of Approximate Reasoning, 2014, 55, (1, Part 2), pp. 167-179
- [34] Haralick, R.M., Shanmugam, K., and Dinstein, I.: 'Textural Features for Image Classification', IEEE Transactions on Systems, Man, and Cybernetics, 1973, SMC-3, (6), pp. 610-621
- [35] Abraham, S., Sanyal, S., and Sanglikar, M.: 'Particle swarm optimisation based Diophantine equation solver', Int. J. Bio-Inspired Comput., 2010, 2, (2), pp. 100-114

**Authors**

Name of Authors	Institutions	Designation	Educational Qualification	Area of Specialization	Contact Details	Photograph
Ranjan Sarmah	<i>Department of Computer Science, Assam University Silchar-788011, Assam, India.</i>	Research Scholar	MCA	Soft Computing Techniques, Computational Biology	Email: <a href="mailto:sarmah.ranjan@gmail.com">sarmah.ranjan@gmail.com</a>	
Dr. Mahendra K Modi	<i>Department of Agricultural Biotechnology, Assam Agricultural University, Jorhat-785013, Assam, India.</i>	Professor	PhD	Bioinformatics, Molecular Biology	Email: <a href="mailto:mkmodi@gmail.com">mkmodi@gmail.com</a>	
Dr. Shahin Ara Begum	<i>Department of Computer Science, Assam University Silchar-788011, Assam, India.</i>	Associate Professor	PhD	Soft Computing Techniques, Data Mining, Pattern Recognition	Email: <a href="mailto:shahin.begum.ara@gmail.com">shahin.begum.ara@gmail.com</a>	