



International Journal of Emerging Technologies in Computational and Applied Sciences (IJETCAS)

www.iasir.net

Experimental Study on Meta Heuristic Optimization Algorithms for Fake Website Detection

Radha Damodaram*, Dr.M.L.Valarmathi**

*Asst. Professor, Department of BCA, SS & IT, CMS College of Science & Commerce,
Coimbatore, Tamil Nadu, INDIA
E-mail: radhabalaji10@gmail.com

**Associate Professor, Department of Computer Science & Engineering., Government College of
Technology, Coimbatore, Tamil Nadu, INDIA

Abstract: The convenience of online commerce has been embraced by consumers and criminals alike. Phishing, the act of stealing personal information via the internet for the purpose of committing financial fraud, has become a significant criminal activity on the internet. There has been good progress in identifying the threat, educating businesses and customers, and identifying countermeasures. However, there has also been an increase in attack diversity and technical sophistication by the people conducting phishing and online financial fraud. Phishing has a negative impact on the economy through financial losses experienced by businesses and consumers, along with the adverse effect of decreasing consumer confidence in online commerce. This work helps in the detection of phishing websites which is based on Multi-Class Associative Classification optimized with ACO, PSO, BFOA and MBAT algorithms. Rule generation using Associative Classification algorithm over the extracted features of the websites help in creating training data sets to classify their legitimacy. After classification, those results have been optimized ACO, PSO, BFOA & MBAT Algorithms. The experimental results demonstrated and compared the feasibility of using MBAT in real world applications and its better performance.

Keywords: Comparative Study, ACO, PSO, BFOA & MBAT

I. Introduction

The term phishing is a general term for the creation and use by criminals of e-mails and websites – designed to look like they come from well-known, legitimate and trusted businesses, financial institutions and government agencies – in an attempt to gather personal, financial and sensitive information. These criminals deceive Internet users into disclosing their bank and financial information or other personal data such as usernames and passwords, or into unwittingly downloading malicious computer code onto their computers that can allow the criminals subsequent access to those computers or the users' financial accounts.

Although phishing, identity theft and identity fraud are terms that are sometimes used interchangeably, some distinctions are in order. Phishing is best understood as one of a number of distinct methods that identity thieves use to “steal” information through deception – that is, by enticing unwitting consumers to give out their identifying or financial information either unknowingly or under false pretenses, or by deceiving them into allowing criminals unauthorized access to their computers and personal data. The United States and some other countries use the term “identity theft,” and the United Kingdom often uses the term “identity fraud,” to refer broadly to the practice of obtaining and misusing others' identifying information for criminal purposes. Identity fraud also can be used to refer to the subsequent criminal use of others' identifying information to obtain goods or services, or to the use of fictitious identifying information (not necessarily associated with a real living person) to commit a crime. Phishing is committed so that the criminal may obtain sensitive and valuable information about a consumer, usually with the goal of fraudulently obtaining access to the consumer's bank or other financial accounts. Often “phishers” will sell credit card or account numbers to other criminals, turning a very high profit for a relatively small technological investment [1].

2. Website Phishing

A computing scam where the perpetrators try to get sensitive personal information by sending users to fake, but legitimate looking websites is known as website phishing. Fake pages are often designed to look almost the same as the genuine ones in both layout and content. In addition, phishers might insert an arbitrary advertisement banner that redirects users to another malicious Website [9]. To respond to this threat, software vendors and companies have released a variety of anti-phishing toolbars and software. Some of the toolbars have obvious usability problems, which can ultimately affect the performance of these toolbars.

2.1 Fake Website Detection Using Associative Classification

Evolving with the anti-phishing techniques, various phishing techniques and more complicated and hard-to-detect methods are used by phishers. The most straightforward way for a phisher to defraud people is to make the phishing web pages similar to their targets. Actually, there are many characteristics and factors that can distinguish the original legitimate website from the forged phishing website like Spelling errors, Long URL address and Abnormal DNS record [2]. In this research work, 27 phishing features and indicators and clustered them into six Criteria (URL & Domain Identity, Security & Encryption, Source Code & Java script, Page Style & Contents, Web Address Bar and Social Human Factor), and each criteria has its own phishing components. The full list is shown in Table.2.1 which will be later on used for analysis and methodology study.

Table 2.1: Phishing Indicators and their criteria

Criterion	Phishing Indicators
URL & Domain Identity	Using IP address
	Abnormal request URL
	Abnormal URL of anchor
	Abnormal DNS record
	Abnormal URL
Security & Encryption	Using SSL Certificate
	Certificate authority
	Abnormal cookie
	Distinguished names certificate
Source Code & Java script	Redirect pages
	Straddling attack
	Pharming attack
	On Mouse over to hide the Link
	Server Form Handler (SFH)
Page Style & Contents	Spelling Errors
	Copying website
	Using form s with Submit button
	Using pop-ups windows
	Disabling right-click
Web Address Bar	Long URL address
	Replacing similar char for URL
	Adding a prefix or suffix
	Using the @ Symbols to confuse
	Using hexadecimal char codes
Social Human Factor	Emphasis on security
	Public generic salutation
	Buying time to access accounts

We utilized data mining classification and association rule approaches in our new phishing website detection model as shown in Figure.8 to find significant patterns of phishing characteristic or factors in the phishing website archive data. Particularly, we used association and classification technique (MCAR) to compare the relationships of the different phishing classification features and rules.

2.2 Why use Data Mining?

Data Mining is the process of searching through large amounts of data and picking out relevant information. It has been described as "the nontrivial extraction of implicit, previously unknown, and potentially useful information from large data sets. Data mining tools predict future trends and behaviors, allowing businesses to make proactive, knowledge-driven decisions. Association rules mining is another important data mining technique. It is used to find strong and interesting relationships among data items present in a set. A typical example of ARM is market basket analysis. In market basket analysis each record contains a list of items purchased by a customer. We are interested to find out the set of items that are frequently purchased together. The objective is to search for interesting habits of customers. The sets of items occurring together can be written as association rules. These association rules can be written as "IF THEN" statements. IF part is called the

antecedent of rule and THEN contains the consequent of the rule. In ARM the antecedent and consequent are sets of data items called item-set. An item set that contains k items is called k item set. An association rule is written as $A \Rightarrow B$, where A and B are set of items. There are different real world applications of ARM including market basket analysis, customer segmentation, electronic commerce, medical, web mining, finance, and bio informatics[3].

2.3 Theory and Methodology

The attribute values for all our input items which represent the six phishing features and criteria ranged between three fuzzy set values (Genuine, Doubtful and Legitimate) using Fuzzy Logic taking into consideration all the input fuzzy variables for all criteria different components. The output class attribute of our phishing website rate is one of these values (*Very Legitimate, Legitimate, Suspicious, Phishy or Very Phishy*).

Table.2.2 Example of Training Data Set

ID	URL	Security	Java	Style	Address	Social	Class
1	G	G	D	G	G	G	Very Legitimate
2	D	G	G	D	G	D	Legitimate
3	D	D	G	F	D	G	Suspicious
4	F	D	G	D	F	D	Phishy
5	D	F	F	D	F	F	Very Phishy

2.4 Website Phishing Training Data Sets

Two publicly available datasets were used to test our implementation: the "phishtank" from the phishtank.com ([http:// www.phishtank.com/phish_archive.php](http://www.phishtank.com/phish_archive.php), 2010) which is considered one of the primary phishing-report collates both the 2009 and 2010 collections. The PhishTank database records the URL for the suspected website that has been reported, the time of that report, and sometimes further detail such as the screenshots of the website, and is publicly available. The Anti Phishing Working Group (APWG) which maintains a "Phishing Archive" describing phishing attacks dating back to September 2007[4].

3. Enhancement

3.1 Optimization Techniques

The goal of this research is to investigate the suitability of different optimization techniques for fake website detection like Ant Colony Optimization (ACO), Particle Swarm Optimization (PSO), Bat Algorithm and Bacteria Foraging Algorithm which can be learned in reasonable time even for large datasets and comparison of those algorithms.

3.1.1 Optimization

Optimization is nothing but selection of a best element from some set of available alternatives. An optimization problem consists of maximizing or minimizing a real function by systematically choosing input values from within an allowed set and computing the value of the function. To increase more accuracy and reduce encoding time and error rate, the hybrid classification algorithms, combining the idea of association rules mining and supervised classification using following techniques are proposed.

3.2 Ant Colony Optimization (ACO)

Our proposed associative classification algorithm uses ACO algorithm for finding interesting relationships among data items. It uses its evolutionary capability to efficiently find more interesting subsets of association rules. It does not exhaustively search for all possible association rules as conventional ARM approaches does. In each generation of the algorithm a number of rules that satisfies minimum support and confidence threshold are selected for the final classifier. After each generation pheromones values are updated in such a way that better rules can be extracted in next coming generations. The final discovered rule set is the predictive model and is used to classify unseen test samples. In this work, a hybrid classification algorithm called ACO combining the idea of association rules mining and supervised. The proposed technique integrates classification with association rule mining to discover high quality rules for improving the performance of resulting classifier. ACO is used to mine only appropriate subset of class association rules instead of exhaustively searching for all possible rules. The mining process stops when the discovered rule set achieves a minimum coverage threshold [5]. The experiments are performed using a ten-fold cross validation procedure.

3.2.1 Methodology

At the start of the algorithm, discovered rule set is empty and user defined parameters are initialized that include minimum support, minimum confidence, minimum coverage and number of ants used by the algorithm as shown in Figure.5.2.

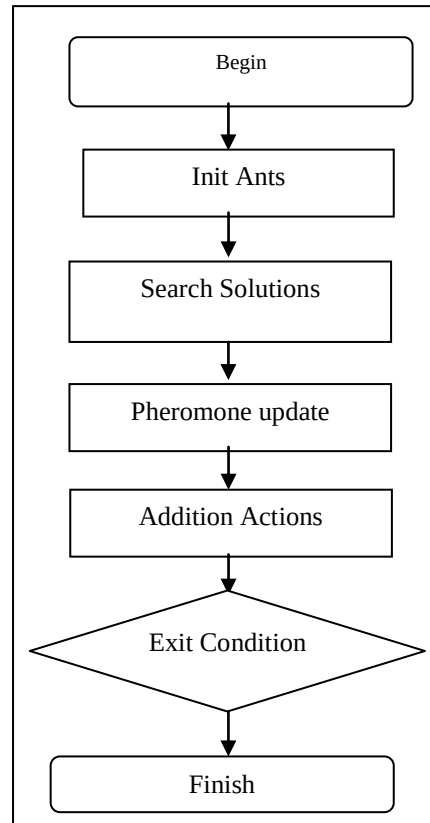


Figure 3.1 Ant Colony Optimization Process.

3.2.2 Rule Construction

Each ant constructs a single item rule in the first generation. In the second generation each ant tries to construct a rule with two items. Similarly we have 3 item rules in 3rd generation and so on. Rules with a maximum k number of items are generated in the k th generation, where k is the number of attributes in training set excluding the class attribute.

3.2.3 Pheromone Initialization

The pheromone values on all edges are initialized before the start of WHILE loop for each new class. The pheromone values on the edges between all items are initialized with the same amount of pheromone. The initial pheromone is:

$$T_{ij}(t=1) = \frac{1}{\sum_{i=1}^a b_i}$$

Where a is the total number of attributes in training set excluding the class attribute and b_i is the number of possible values in the domain of an attribute ai . The pheromone values of all those items are set to zero which do not satisfy a minimum support threshold. The value of zero ensures that these items cannot be selected by ants during rule construction process.

3.2.4 Pheromone Update

The pheromone values are updated after each generation so that in next generation ants can make use of this information in their search. The amount of pheromone on links between items occurring in those rules which satisfy minimum support threshold but whose confidence is below the minimum required confidence (and hence they were removed from the temporary rule set) are updated according to the Equation:

$$T_{ij}(g+1) = (1-p)T_{ij}(g) + \left(1 - \frac{1}{1+Q}\right)T_{ij}(g)$$

Where $\tau_{ij}(g)$ is the pheromone value between $item_i$ and $item_j$ in current generation, ρ represents the pheromone evaporation rate and Q is the quality of the rule constructed by an ant. The pheromones of these rules are increased so that in next generation ants can explore more search space instead of searching around those rules which are already inserted in the discovered rule set. This pheromone strategy increases the diversity of the search by focusing on new unexplored areas of search space [6].

3.2.5 Discovered Rule Set

When the coverage of discovered rule set of the selected class reaches a coverage threshold then we stop the rule discovery process for that class. This process is repeated for all classes. A final discovered rule set (or list) contains discovered rules of all classes.

3.2.6 Experiments and Analysis

We have implemented the proposed algorithm in java. We compare the results of proposed approach with other state of the art, well known classification algorithm MCAR. The performance measures for comparison are predictive accuracy, number of rules, and number of terms per rule. The experiments are performed using a ten-fold cross validation procedure. In ten-fold cross validation a dataset is randomly divided into ten equally sized, mutually exclusive subsets. Each of the subset is used once for testing and the other nine are used for training. The results of these ten runs are averaged and this average is reported as the final result. We use 512 website URLs for comparing results. They also have diversity in terms of number of attributes, number of transactions, number of classes and types of the attributes [7].

3.3 Particle Swarm Optimization

A particle is treated as a point in an M-dimensional search space, and the status of a particle is characterized by its position and velocity. Initialized with a swarm of random particles, PSO is achieved through particle flying along the trajectory that will be adjusted based on the best experience or position of the one particle called local best and the best experience or position ever found by all particles called global best. The M-dimensional position for the i th particle in the t th iteration can be denoted as

$$X_i(t) = \{x_{i1}(t), x_{i2}(t), \dots, x_{iM}(t)\}$$

Similarly, the velocity also a multi dimensional vector, for the i th particle in the t th iteration can be described as

$$V_i(t) = \{v_{i1}(t), v_{i2}(t), \dots, v_{iM}(t)\}$$

Particle swarm has two primary operators : Velocity update and Position Update given by

$$V_i(t) = w(t) V_i(t-1) + c_1 r_1 (X_{iL} - X_i(t-1)) + c_2 r_2 (X_{iG} - X_i(t-1))$$

$$X_i(t) = X_i(t-1) + V_i(t)$$

where X_{iL} represents the local best of the i th particle and X_{iG} represents the global best of the i th iteration, c_1 and c_2 are positive constants and r_1 and r_2 are random numbers between 0 to 1 and $w(t)$ is the inertia weight used to control the impact of the previous velocities on the current velocity, influencing the trade-off between the global and local experiences[8]. In this paper PSO technique has been applied to get optimal solution to reduce the optimization time. It is also used to accelerate the search for the domain block that is most similar to the range block. Here the particles represent the solutions. Initially the particles take some random positions in solutions. Most of evolutionary techniques have the following procedure:

1. Random generation of an initial population
2. Reckoning of a fitness value for each subject. It will directly depend on the distance to the optimum.
3. Reproduction of the population based on fitness values.
4. If requirements are met, then stop. Otherwise go back to 2.

From the procedure, we can learn that PSO shares many common points with GA. Both algorithms start with a group of a randomly generated population, both have fitness values to evaluate the population. Both update the population and search for the optimum with random techniques. Both systems do not guarantee success. Particles update themselves with the internal velocity. They also have memory, which is important to the algorithm. Compared with ACO, the information sharing mechanism in PSO is significantly different. In PSO, only gBest (or lBest) gives out the information to others. It is a one-way information sharing mechanism. The evolution only looks for the best solution. Compared with ACO, all the particles tend to converge to the best solution quickly even in the local version in most cases[11].

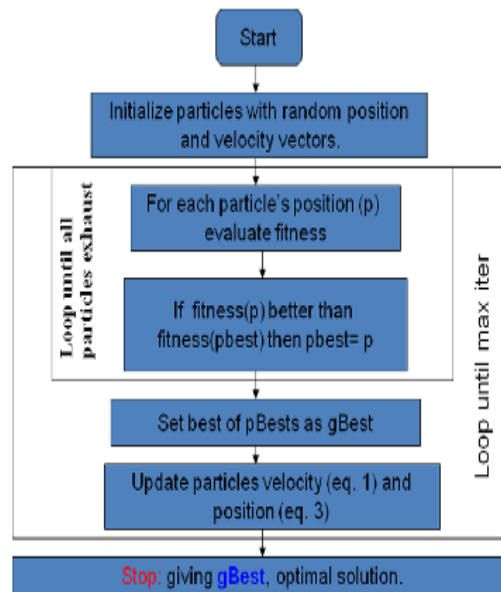


Figure.5.5 PSO Algorithm

3.4 Bacteria Foraging Algorithm

During foraging of the real bacteria, locomotion is achieved by a set of tensile flagella. Flagella help an E.coli bacterium to tumble or swim, which are two basic operations performed by a bacterium at the time of foraging as shown in figure.5.6. When they rotate the flagella in the clockwise direction, each flagellum pulls on the cell. That results in the moving of flagella independently and finally the bacterium tumbles with lesser number of tumbling whereas in a harmful place it tumbles frequently to find a nutrient gradient. Moving the flagella in the counterclockwise direction helps the bacterium to swim at a very fast rate. In the above-mentioned algorithm the bacteria undergoes chemotaxis, where they like to move towards a nutrient gradient and avoid noxious environment. Generally the bacteria move for a longer distance in a friendly environment [9].

Due to the occurrence of sudden environmental changes or attack, the chemotactic progress may be destroyed and a group of bacteria may move to some other places or some other may be introduced in the swarm of concern. This constitutes the event of elimination-dispersal in the real bacterial population, where all the bacteria in a region are killed or a group is dispersed into a new part of the environment.

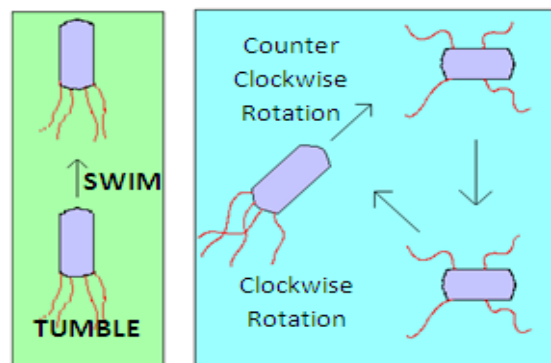


Figure.5.6 Swim And Tumble of a Bacterium

Let us define a chemotactic step to be a tumble followed by a run or a tumble followed by a run. Let j be the index for the chemotactic step. Let k be the index for the reproduction step. Let l be the index of the elimination-dispersal event. Let P is a p -dimensional vector of real numbers. Also let

- P : Dimension of the search space,
- S : Total number of bacteria in the population,
- N_c : The number of chemotactic steps,

- N_s : The swimming length.
 N_{re} : The number of reproduction steps,
 N_{ed} : The number of elimination-dispersal events,
 P_{ed} : Elimination-dispersal probability,
 $C(i)$: Size of the step taken in the random direction specified by the tumble.

- 1) Chemotaxis: Swimming (in the same direction) or Tumbling (in an absolute opposite direction)
 Computational chemotaxis the movement of the bacterium may be represented by

$$\theta^i(j+1, k, l) = \theta^i(j, k, l) + C(i) \frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}$$

Where i th bacterium, j th chemotactic, k th reproductive, l th elimination, $C(i)$ – Size of the Unit taken randomly and $\Delta(i)$ – Vector in arbitrary $[-1,1]$ [41]

- 2) Swarming: The cell-to-cell signaling in E. coli swarm may be represented by the following function

$$j_{cc}(\theta, P(j, k, l)) = \sum_{i=1}^S J_{cc}(\theta, \theta^i, (j, k, l))$$

J_{cc} – Objective function value and P – No. of variables to be optimized

- 3) Reproduction: The least healthy bacteria eventually die while each of the healthier bacteria (those yielding lower value of the objective function) asexually split into two bacteria, which are then placed in the same location. This keeps the swarm size constant.

$$J_{health}^i, \text{ the health of } i\text{th bacterium} \quad J_{health}^i = \sum_{j=1}^{N_c+1} J(i, j, k, l)$$

- 4) Elimination and Dispersal: Gradual or sudden changes in the local environment where a bacterium population lives may occur due to various reasons e.g. a significant local rise of temperature may kill a group of bacteria that are currently in a region with a high concentration of nutrient gradients. To simulate this phenomenon in BFOA some bacteria are liquidated at random with a very small probability while the new replacements are randomly initialized over the search space[10].

Parameters:

Initialize parameters $p, S, N_c, N_s, N_{re}, N_{ed}, P_{ed}, C(i)(i=1,2 \dots S), \Theta(i)$.

Algorithm:

- 1) Elimination-dispersal loop: $l=l+1$
- 2) Reproduction loop: $k=k+1$
- 3) Loop: $j=j+1$
- 4) Do the Chemotaxis process
- 5) If $j < N$, go to step 4. In this case continue chemotaxis since the life of the bacteria is not over.
- 6) Do the Reproduction Process
- 7) If $k < N_{re}$, go to step 3. In this case, we have not reached the number of specified reproduction steps, so we start the next generation of the chemotactic loop.
- 8) Elimination-dispersal: For $i = 1, 2, \dots, S$ with probability P_{ed} , eliminate and disperse each bacterium (this keeps the number of bacteria in the population constant). To do this, if a bacterium is eliminated, simply disperse another one to a random location on the optimization domain. If $l < N_{ed}$, then go to step 2; otherwise end.

3.5 Modified Bat algorithm (MBAT)

By idealizing some of the echolocation characteristics of micro-bats, we can develop various bat-inspired algorithms or bat algorithms. Here we developed Modified Bat Algorithm with Doppler Effect. For simplicity, here some of the approximate or idealized rules [11]. All bats use echolocation to sense distance, and they also 'know' the difference between food/prey and background barriers in some magical way; Bats fly randomly with velocity v_i at position x_i with a fixed frequency f_{min} , varying wavelength λ and loudness A_0 to search for prey. They can automatically adjust the wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission $r \in [0, 1]$, depending on the proximity of their target; Doppler Effect is the change in frequency of a wave for an observer moving relative to the source of the wave. The received frequency is higher (compared to the emitted frequency) during the approach, it is identical at the instant of passing by, and it is lower during the recession. where v_s is positive if the source is moving away from the observer, and negative if the source is moving towards the observer.

$$f = \left(\frac{c}{c + v_s} \right) f_o \quad - (1)$$

(ii) (or) where the similar convention applies: v_r is positive if the observer is moving towards the source, and negative if the

$$f = \left(\frac{c + v_r}{c} \right) f_o \quad - (2)$$

(iii) (or) Single equation with both the source and receiver moving.

$$f = \left(\frac{c + v_r}{c + v_s} \right) f_o \quad - (3)$$

where 'c' is the velocity of waves in the medium, v_r is the velocity of the receiver relative to the medium; positive if the receiver is moving towards the source, and v_s is the velocity of the source relative to the medium; positive if the source is moving away from the receiver.

Although the loudness can vary in many ways, we assume that the loudness varies from a large (positive) A_0 to a minimum constant value A_{min} [12]. We have to define the rules how their positions x_i and velocities v_i in a d-dimensional search space are updated. The new solutions x_i^t and velocities v_i^t at time step t are given by

$$f_i = f_{min} + (f_{max} - f_{min})\beta,$$

$$v_i^t = v_i^{t-1} + (x_i^t - x^*)f_i,$$

$$x_i^t = x_i^{t-1} + v_i^t,$$

Where $\beta \in [0, 1]$ is a random vector drawn from a uniform distribution. Here x^* is the current global best location (solution) which is located after comparing all the solutions among all the n bats. As the product $\lambda_i f_i$ is the velocity increment, we can use either f_i (or λ_i) to adjust the velocity change while fixing the other factor λ_i (or f_i), depending on the type of the problem of interest. In our implementation, we will use $f_{min} = 0$ and $f_{max} = 100$, depending the domain size of the problem of interest. Initially, each bat is randomly assigned a frequency which is drawn uniformly from $[f_{min}, f_{max}]$. For the local search part, once a solution is selected among the current best solutions, a new solution for each bat is generated locally using random walk

$$x_{new} = x_{old} + EA^t,$$

Where $E \in [-1, 1]$ is a random number, while $A^t = \langle A_i^t \rangle$ is the average loudness of all the bats at this time step. The update of the velocities and positions of bats have some similarity to the procedure in the standard particle swarm optimization as f_i essentially controls the pace and range of the movement of the swarming particles.

To a degree, BA can be considered as a balanced combination of the standard particle swarm optimization and the intensive local search controlled by the loudness and pulse rate.

3.5.1 Loudness and Pulse Emission

Furthermore, the loudness A_i and the rate r_i of pulse emission have to be updated accordingly as the iterations proceed. As the loudness usually decreases once a bat has found its prey, while the rate of pulse emission increases, the loudness can be chosen as any value of convenience. For example, we can use

$$A_0 = 100 \text{ and } A_{\min} = 1.$$

For simplicity, we can also use $A_0 = 1$ and $A_{\min} = 0$, assuming $A_{\min} = 0$ means that a bat has just found the prey and temporarily stop emitting any sound. Now we have

$$A_i^{t+1} = \alpha + A_i^t$$

$$r_i^{t+1} = r_i^0 [1 - \exp(-\gamma t)]$$

Where α and γ are constants. In fact, α is similar to the cooling factor of a cooling schedule in the simulated annealing. For any $0 < \alpha < 1$ and $\gamma > 0$, we have

$$A_i^t \longrightarrow 0, r_i^t \longrightarrow r_i^0 \text{ as } t \longrightarrow \infty$$

In the simplicity case, we can use $\alpha = \gamma$, and we have used $\alpha = \gamma = 0.9$ in our simulations. The choice of parameters requires some experimenting. Initially, each bat should have different values of loudness and pulse emission rate, and this can be achieved by randomization. For example, the initial loudness A_i^0 can typically be $[1, 2]$, while the initial emission rate r_i^0 can be around zero, or any value $r_i^0 \in [0, 1]$ if using (6). Their loudness and emission rates will be updated only if the new solutions are improved, which means that these bats are moving towards the optimal solution [13].

4. Results & Comparison

There are many ways to carry out the comparison of algorithm performance and two obvious approaches are: to compare the numbers of function evaluations for a given tolerance or accuracy, or to compare their accuracies for a fixed number of function evaluations.

Error rate: The Modified Bat algorithm results in less error rate when compared to the Association classification using MCAR algorithm supervised with different optimization techniques like ACO, PSO, BFOA & MBAT with good number of iterations during optimization.

Correct prediction: The Modified Bat algorithm predicts the phishing website more accurate than the existing algorithm. There is a significant relation between the two phishing website criteria's (URL and Domain Identity) and (Security and Encryption) for identifying phishing website. Also we found insignificant trivial influence of the (Page Style and Content) criteria along with (Social Human Factor) criteria for identifying phishing websites.

Table 4.1 Comparison of the proposed algorithms with MCAR

Techniques	Accuracy	Time Taken in ms	Error Rate in %
MCAR	84%	12342	12.62
ACO	89%	11276	10.23
PSO	92%	9578	9.13
BFOA	97%	8211	8.67
MBAT	98%	7902	8.56

Table 4.2 Sample Tested data set and their extracted features

Data Set (Websites)	URL & Domain Identity	Security & Encryption	Source Code & Java	Page Style & Contents	Web Address Bar	Social Human Factor	Website Rate
http://www.pcgames.com.cn	Genuine	Genuine	Doubtful	Genuine	Doubtful	Genuine	Legitimate
http://www.redtube1ive.com	Genuine	Genuine	Genuine	Doubtful	Fraud	Genuine	Legitimate
http://www.niux88.com	Genuine	Genuine	Doubtful	Genuine	Doubtful	Genuine	Legitimate
http://www.porn1tube.com	Genuine	Doubtful	Genuine	Doubtful	Fraud	Genuine	Suspicious
http://www.medicinenet.com	Genuine	Doubtful	Genuine	Doubtful	Fraud	Genuine	Suspicious
http://www.yanovich.us/powers/index.php	Fraud	Fraud	Doubtful	Doubtful	Fraud	Doubtful	Phishing
http://www.qdmos.net/link/atpl/related.co.za	Fraud	Doubtful	Doubtful	Doubtful	Fraud	Doubtful	Phishing
http://www.cirakli.com/cha-se-	Genuine	Genuine	Doubtful	Genuine	Doubtful	Genuine	Legitimate
http://spontanmusic.com/cover/reptiles/	Doubtful	Doubtful	Doubtful	Fraud	Fraud	Fraud	Phishing
http://www.buy.com	Genuine	Doubtful	Genuine	Fraud	Genuine	Genuine	Legitimate

No. of Folds : 10

No. of iterations : 10

No. of URLs :1006

No. of Rules : 22

No. of Folds : 10

No. of iterations : 20

No. of URLs :1006

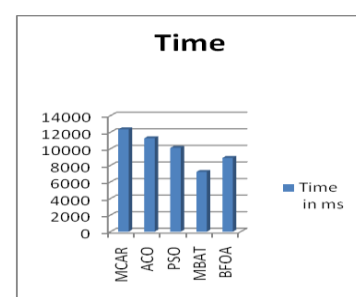
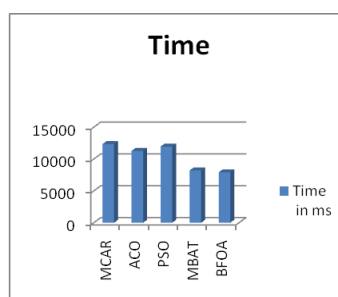
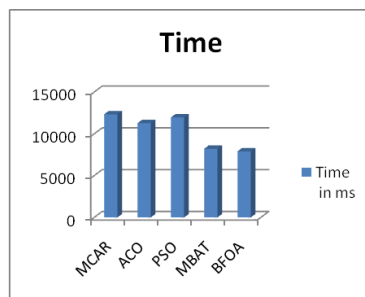
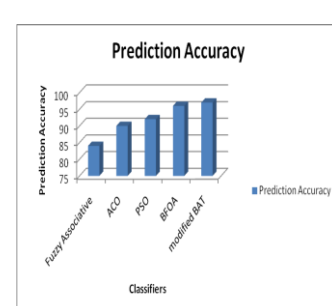
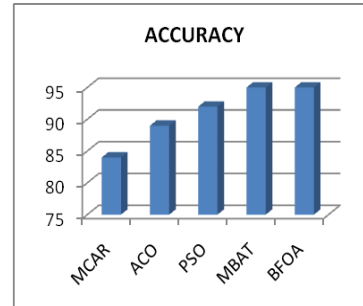
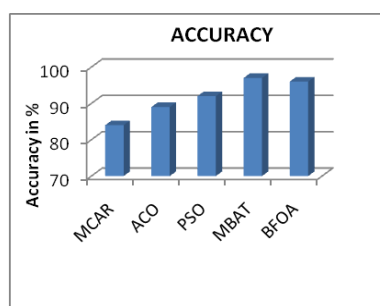
No. of Rules : 27

No. of Folds : 10

No. of iterations : 40

No. of URLs :1006

No. of Rules : 25

**Figure 4.1 After Comparing all Iterations, Modified Bat Algorithm takes minimum time****Figure 4.2 After Comparing all Iterations, Modified Bat Algorithm Having a high Accuracy**

5. References

- Emigh, A (2005). Online identity theft: Phishing technology, chokepoints and countermeasures. ITTC Report on Online Identity Theft Technology and Counter measures; <http://www.antiphishing.org/Phishing-dhs-report.pdf>
- "Fighting Phishing with Discriminative Keypoint Features", Kuan-Ta Chen, Chun-Rong Huang, and Chu-Song Chen, *Institute of Information Science, Academia Sinica, Jau-Yuan Chen Columbia University*
- Associative Classification Techniques for predicting e- Banking Phishing Websites, Maher Aburrous, Dept. of Computing ,University of Bradford, Bradford, UK.
- Maher Aburrous, M. A. Hossain, Keshav Dahal, Fadi Thabtah, "Associative lassification Techniques for predicting e-Banking Phishing Websites", International Conference on MCIT, 2010, IEEE-XPlere. pp – 12.
- Bing Liu, Wynne Hsu, Yiming Ma, "Integrating Classification and Association Rule Mining". Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining (KDD-98, Plenary Presentation), New York, USA
- Ant Colony Optimization, [www. Scholarpedia.org article/AntColonyOptimization](http://www.Scholarpedia.org/article/AntColonyOptimization)

7. "The Techniques of ACO", www.csse.monash.edu.au/~bemdm/CSE460/Lectures/Cse4609.pdf
8. Suraj Pandey, LinlinWu, Siddeswara Guru, Rajkumar Buyya, "A Particle Swarm Optimization(PSO)-based Heuristic for Scheduling Workflow Applications in Cloud Computing Environments",
9. Swagatam Das, Arijit Biswas, Sambarta Dasgupta, and Ajith Abraham; 2008; "Bacterial Foraging Optimization Algorithm: Theoretical Foundations, Analysis, and Applications"
10. Fadi Thabtah, Peter Cowling, Yonghong Peng, "Mining multi-class datasets using Genetic Relation Algorithm for rule reduction", IEEE Trans on Evolutionary Computation, 2009, pp 3249-3255
11. A New Metaheuristic Inspired Algorithm, Xin-She Yang, Department of Engineering, University of Cambridge.
12. "Doppler Effect" -wikipedia
13. "Bats behaviour", www.Swamintelligence.org.