A Multi-Swarm Particle Swarm Optimization Model for Web Page Security Classification

Mrs. N. Thangammal and Dr. A. Pethalakshmi

Research Scholar, Mother Teresa Women's University, Kodaikanal, India.
Associate Professor & Head, Department of Computer Science, M.V.M.Govt. Arts College for Women, Dindigul, India.

Received 28 January 2017; Accepted 22 May 2017; Available online 28 May 2017

Address For Correspondence:
Mrs. N. Thangammal, Research Scholar, Mother Teresa Women's University, Kodaikanal, India.

ABSTRACT
Web Application Security is increasingly becoming an important concern given the rising implications of security related attacks on web applications. Typically, a web application is composed of multiple modules and some of these modules are more prone to security related risks than others. It is very important that such modules are identified apriori so that the manager can allocate resources for dealing with the security challenge in proportion with the estimated level of security. The classification technique of data mining can be of immense utility here as building a classification model that can classify the security of a module based on some metrics can aid in the discovery of more vulnerable modules. The paper proposes a Multi-Swarm Particle Swarm Optimization approach for building such a classification model.

KEYWORDS: Web Application Security, Particle Swarm Optimization, Multi-Swarm PSO, Classification

INTRODUCTION

The recent spurt in the number of the engineered web applications and the usage of such applications for various security sensitive tasks like banking and e-commerce have catapulted the security challenge to the forefront. Offutt includes security in the list of quality characteristics expected from web applications [2]. Deshpande et.al. list security as one of the major factors that makes web applications distinct from conventional applications [3]. Security is considered as a domain dependent quality factor by Nabil, Mosad and Hefny [4].

Given that Security is one of the crucial quality characteristic of web applications, there is a need to identify mechanisms to improve the security during the web development process so ensure that the process results in a secure web application. Models that can throw light on the likely vulnerability of a module of a web application can greatly aid the developer and manager to focus more resources for such modules predicted to be highly vulnerable.

1.1 Organization of the paper:
The rest of the paper is organized as follows. Section 2 presents a general overview of Particle Swarm Optimization. Section 3 outlines the proposed methodology, Section 4 gives the results and section 5 concludes the research and suggests potential avenues for future research.

2. Particle Swarm Optimization:
Particle Swarm Optimization (PSO) was conceived by observing the social behavior of birds in a flock and attempts to simulate the same in uncovering solutions to problems (Ebbertart & Kennedy; Kennedy &
Ebberhart). The basic idea behind PSO is as follows: each particle flies in the search space with a velocity adjusted by its own memory together with flying experience of the companions. The fitness of each particle is determined by an objective function given as:
\[
v_i^t = w \times v_i^{t-1} + c_1 \times r_1 (p_{id}^t - x_i^t) + c_2 \times r_2 (p_{gd}^t - x_i^t).
\]
Here, \(v_i^t\) represents the ith particle, \(d\) is the dimension of the solution space, \(c_1\) denotes the cognition learning factor, and \(c_2\) indicates the social learning factor, \(r_1\) and \(r_2\) are random numbers uniformly distributed in (0, 1), \(p_{id}^t\) and \(p_{gd}^t\) stand for the position with the best fitness found so far for the ith particle and the best position in neighborhood, \(v_i^{t-1}\) and \(v_i^t\) are the velocities at time \(t\) and time \(t-1\) respectively, and \(x_i^t\) is the position of ith particle at time \(t\). Each particle moves to a potential new solution based on the equation
\[
x_{id}^{t+1} = x_{id}^t + v_{id}^t, d = 1, 2, ..., D.
\]

The main advantage of PSO is its simplicity of implementation with a need to tune very few parameters. It has been widely used to solve optimization and feature selection problems (Liu et al; Huang et al). The performance of PSO can be improved in a variety of ways including adopting a multi-swarm strategy in which each swarm can explore a different portion of the solution space (Blackwell).

### 2.1 Multi-Swarm PSO:

The technique used in this research is the modified Multi-Swarm PSO proposed by Liu et al. Liu et al. use the Multi-Swarm PSO for feature selection. In this approach, a number of sub-swarms are employed and a multi-swarm scheduler monitors and controls each sub-swarm using the following rules [1]:

#### The swarm request rule

Given:
\[
S_i = \begin{cases} 
1, & \text{if } d_i < \frac{t_i - t_{it}}{t_{it}} \times \text{rand()} \times \text{fitness} \\
0, & \text{if } d_i \geq \frac{t_i - t_{it}}{t_{it}} \times \text{rand()} \times \text{fitness} 
\end{cases}
\]

\((1)\)

Here, \(d\) represents a threshold, \(t_{it}\) the maximum iteration number, \(t_i\) the current iteration number. \(\text{rand}()\) is a random number uniformly distributed in \(U(0, 1)\). If \(S_i = 1\), the current sub swell sends the results (it’s corresponding pbest and gbest values) to the multi-swarm scheduler.

The multi-swarm scheduler request rule

The multi-swarm scheduler monitors each sub-swarm and can send a request to obtain a result from any sub-swarm if it is valuable. If the sub-swarm has sent swarm request rules more than a specified number of times, the multi-swarm scheduler will send the rule i.e. the multi-swarm scheduler sends this rule depending on the activity level of the sub-swarm.

#### The multi-swarm collection rule

The multi-swarm scheduler collects results from the alive sub-swarms and updates pbest and gbest values.

#### The multi-swarm destroying rule

The multi-swarm scheduler destroys sub-swarms in 2 cases. In the first case, if the sub-swarm sends the swarm request rule fewer than a threshold number of times and in the second case when the sub-swarm does not change the gbest value within a specified number of iterations.

### 2.2 The Multi-Swarm PSO algorithm:

This section gives the algorithm that uses the 4 rules described above:

**Step 1:** Initialize: Load the data from the source. Initialize the size of swarms randomly. Initialize particle positions and velocities of each swarm with random values. Calculate the objective function and update pbest and gbest for each swarm.

**Step 2:** Parameter Selection: Specify the following parameters for each swarm: Lower and upper bounds on velocity, size of particles, number of iterations, cognition learning factor, social learning factor, threshold \(d_i\) in Eq 1, thresholds required for the multi-swarm destroying rule. Set iteration_number=0

**Step 3:** For each swarm, if iteration_number < max_iterations or gbest has changed in more than 50 iterations, continue with step 4. Otherwise, destroy the swarm and go to step 10. The main module compares the gbest of each swarm with the previous one in the module to decide if gbest needs to be updated using multi-swarm scheduler request rule. In case pbest or gbest is changed, multi-swarm collection rule is executed.

**Step 4:** For each swarm, if current particle number < particle size, continue with step 5. Else, proceed to step 9.
3 Proposed Methodology:

The basic idea is to apply the multi-swarm PSO algorithm for web site security estimation. As already stated, such estimation would greatly aid developers and managers to focus resources on web pages estimated to be insecure. The Motivation for the research work was the research of Liu and Khoshgoftaar who have successfully applied Genetic Programming to build a Software Quality Classification Model [1]. According to them, their work is the first to apply GP for Software Quality Classification. Because Security has several interesting parallels with Quality and vulnerabilities share a lot with “faults”, we explored if the GP model based technique can be replaced with a multi-swarm PSO based technique and to investigate whether the multi-swarm PSO can yield better results. The research work was a direct result of the exploration.

30 Web sites developed by a local software organization were thoroughly studied. The 30 websites had a total of 282 web pages in all. Data pertaining to attacks on these web sites and the root vulnerabilities are known. This data had been collected by the organization over a period of time. For the purpose of the research, a web page was considered insecure if it had more than 5 vulnerabilities. It is considered secure otherwise. A web page is the combination of the client side HTML and the associated server side scripts taken together. We describe each web page with a set of metrics. The degree to which the values of these metrics have an impact on security is unknown. We consider the following metrics.

Table 1: Metrics Considered for the Study

<table>
<thead>
<tr>
<th>Metric</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CSTM</td>
<td>Number of statements of server components executed in response to requests</td>
</tr>
<tr>
<td>ICC</td>
<td>Number of independent paths in the inter-procedural control flow graph</td>
</tr>
<tr>
<td>Sensitive Data Accesses (SDA)</td>
<td>Proportion of the number of “sensitive” data accessed by the server side</td>
</tr>
<tr>
<td>Cyclomatic Complexity (CYCM)</td>
<td>The Cyclomatic Complexity of the server side script</td>
</tr>
<tr>
<td>NClientScriptComp</td>
<td>Total Number of Client Script Components</td>
</tr>
<tr>
<td>NServerScriptComp</td>
<td>Total Number of Server Script Components</td>
</tr>
<tr>
<td>NC</td>
<td>Total Number of Classes</td>
</tr>
<tr>
<td>NM</td>
<td>Total Number of Methods</td>
</tr>
<tr>
<td>NFormE</td>
<td>Total Number of form elements</td>
</tr>
</tbody>
</table>

The Multi-Swarm PSO is applied to build a security classification model based on the above metrics. The fitness function is obtained by the number if misclassifications produced by the model. The lesser misclassifications a model results in, the more fit it is. But, the damage incurred by misclassifying an insecure web page as secure is much more than the damage incurred by misclassifying a secure web page as insecure as the maximum damage that could stem from the later misclassification is wastage of resources. Therefore, the fitness function penalizes the former misclassifications more than the later. If a is the number of former misclassifications and b is the number of later misclassifications, the fitness function is c * a + b, where c is a parameter to be tuned for the application. For the purpose of this research c was chosen to be 2. Since the classes of all the web pages are known apriori, there is little work in determining the number of misclassifications. Following the method outlined in [1], the 282 web pages are classified into two sets – 162 used for training and remaining 120 used for testing.

RESULTS AND DISCUSSION

The classification accuracy of the model, following [Liu etal.] is defined as:

$$\text{accuracy}(N) = \frac{\sum_{i=1}^{N} \text{assess}(n_i)}{|N|} \cdot n_i \in N$$, where \( \text{assess}(n) = \begin{cases} 1 \text{ if } \text{classify}(n) = \text{nc} \\ 0 \text{ otherwise} \end{cases} \)

Here N is the set of web pages to be classified (162 for training, 120 for testing), nc is the class of the item \( n \in N \). The best result for the training data set yielded a classification accuracy of 73.21% while that for the
test data set yielded a classification accuracy of 78.41%. The testing data set yielded better results than the training data set implying that the model has captured the underlying relationships between the data items involved.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Multi-Swarm PSO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>73.21%</td>
</tr>
<tr>
<td>Test</td>
<td>78.41%</td>
</tr>
</tbody>
</table>

To investigate if multi-swarm PSO can yield better results than Genetic Algorithm and plain PSO the research also applied GA and PSO to the problem and obtained the results shown below

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Multi-Swarm PSO</th>
<th>PSO</th>
<th>GA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training</td>
<td>73.21%</td>
<td>64.21%</td>
<td>69.22%</td>
</tr>
<tr>
<td>Test</td>
<td>78.41%</td>
<td>66.91%</td>
<td>71.23%</td>
</tr>
</tbody>
</table>

The lower classification accuracy reported by PSO can be attributed to the fact that PSO has the tendency to converge quickly at local optimal values. Multi-Swarm PSO gives better results than GA. The main advantage of Multi-Swarm PSO over GA is that it entails tuning of a few parameters and therefore the results are sensitive to only these fewer parameters.

5. Conclusion and Future Work:

A Multi-Swarm PSO based classifier model was developed to classify web as “Secure” or “Not Secure” based on metrics. The results are quite promising and are expected to be of great utility to web developers and web development organizations interested in development of secure web sites.

As a part of future work, Multi-Swarm PSO can be used for the selection of optimal parameter values for a SVM that can be used for the said classification. In fact, the work of [1] does just that. Other robust fitness functions can be employed instead of the simple one presented here.

REFERENCES