Optimal feature selection for firewall log analysis using Machine learning and Hybrid Metaheuristic algorithms

1st Seungwoo Han  
Kwangwoon University  
Seoul, Korea  
seungwoohan0108@gmail.com

2nd Gil Hong  
Sahmyook University  
Seoul, Korea  
wideroad95@nate.com

3rd Jewan Kim  
Sahmyook University  
Seoul, Korea  
wpdhrs6@gmail.com

4th Jeuk Yu  
Sahmyook University  
Seoul, Korea  
jipsina71@gmail.com

5th Sangjun Lee  
Sahmyook University  
Seoul, Korea  
tjdlf1234@naver.com

6th Byongok Cho  
Sahmyook University  
Seoul, Korea  
winjbo1021@gmail.com

7th Jusung Jeon  
Sahmyook University  
Seoul, Korea  
gggg1371@naver.com

Abstract—Firewall log classification is important to monitor network traffic. Most firewall log classification via machine learning has shown good result by network-related features and classifiers. However, feature with many dimensions take a lot of time to do classification. In this paper, we applied a method of feature selection using optimized bee swarm optimization with reinforcement learning. We evaluated average performance by accuracy, macro-averaged precision, macro-averaged recall, and macro-averaged F1 score in 5-stratified folds using a random forest, k-nearest neighbor, and naïve bayes classifier. As a results, it could be applied for an automatic firewall log analysis system.

Keywords—bee swarm optimization, reinforcement learning

I. INTRODUCTION

Firewalls of computers are used to ensure that the network is functioning properly and safely. In particular, as the security of personal information [1] becomes more important and robust, it needs to be conscious to protect networks. Firewall logs are key evidence to identify intruder attacks, including insider and outsider threats [2]. In addition to the existing traditional classification methods, with the development of machine learning and deep learning, a study on log classification and intrusion prevention using it has been conducted [3–8]. Log analysis and intrusion detection defense should attempt to classify efficiently with fewer parameters for quick response. At this time, the number of features used for classification or regression is based on the researcher's experience. For this reason, research is also being conducted to intensively select the optimal feature and parameter via information gain, and genetic algorithms, and reinforcement learning [9–12]. In this paper, we performed optimal feature search using the optimized bee swarm optimization algorithm along with reinforcement learning [13].

II. EXPERIMENT

A. Data acquisition

We used Internet Firewall Data Data Set [14] in UCI Machine Learning Repository. The data have 11 features and 4 labels. Total data points are 65532. Data profile and feature information are shown in Table 1 and Table 2. Four label is allow, deny, drop, reset-both, respectively.

Table 1: Profile of Dataset

<table>
<thead>
<tr>
<th>Name</th>
<th>Data points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allow</td>
<td>37640</td>
</tr>
<tr>
<td>Deny</td>
<td>14987</td>
</tr>
<tr>
<td>Drop</td>
<td>12851</td>
</tr>
<tr>
<td>Reset-both</td>
<td>54</td>
</tr>
<tr>
<td>Total</td>
<td>66532</td>
</tr>
</tbody>
</table>

Table 2: Feature information

<table>
<thead>
<tr>
<th>Name</th>
<th>bytes received</th>
<th>bytes sent</th>
<th>destination port</th>
</tr>
</thead>
<tbody>
<tr>
<td>elapsed time</td>
<td>NAT destination port</td>
<td>packets</td>
<td>packets received</td>
</tr>
<tr>
<td>NAT source port</td>
<td>packets sent</td>
<td>source port</td>
<td></td>
</tr>
</tbody>
</table>
B. Methods

1. Bee swarm optimization algorithm

Bee swarm optimization algorithm (BSO) is an algorithm that is inspired by the social behavior of bees. Each bee is an object working together to solve the optimization problem, and they search for the fitness function using a feature combination in iterations. The fitness function of this research is set to average accuracy.

2. Reinforcement learning

Reinforcement learning (RL) refers to an algorithm in which an agent defined in the environment recognizes the current state and finds an action that maximizes the reward among actions. The RL algorithm applied in this paper is Q-learning [15]. Local search and experience of bee replace Q-learning. In this process, the reward is given differently depending on the accuracy of the current and next states. The reward-setting condition is shown Table 3.

Table 3: Reward-setting condition

<table>
<thead>
<tr>
<th>Condition</th>
<th>Reward</th>
</tr>
</thead>
<tbody>
<tr>
<td>if the next state accuracy is higher than the current state</td>
<td>next state accuracy</td>
</tr>
<tr>
<td>if the current state accuracy is high</td>
<td>next state accuracy - current state accuracy</td>
</tr>
<tr>
<td>if the number of features in the current state is greater than the number of features in the next state</td>
<td>1/4 * next state accuracy</td>
</tr>
<tr>
<td>if the number of features in the current state is less than the number of features in the next state</td>
<td>-1/4 *next state accuracy</td>
</tr>
</tbody>
</table>

As a result, the agent tries to get the best accuracy while getting fewer features. Moreover, to reduce the space in the search space, we applied the XOR operation on the best solution and the current state solution.

3. Classifier

In this paper, we compared performance using several classifier: random forest (RF), k-nearest neighbor (KNN), and naïve baye (Bernoulli NB).

III. HYPER-PARAMETER SETTING

Table 4 shows the hyper-parameters applied to this experiment.

Table 4: Hyper-parameter lists

<table>
<thead>
<tr>
<th>Name</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flip</td>
<td>5</td>
<td>To calculate Search Region space</td>
</tr>
<tr>
<td>Max Chance</td>
<td>3</td>
<td>Number of chances to escape local minima</td>
</tr>
<tr>
<td>Number of Bee</td>
<td>3</td>
<td>Worker to solve</td>
</tr>
<tr>
<td>Max iteration</td>
<td>3</td>
<td>Number of total iteration</td>
</tr>
<tr>
<td>Local iteration</td>
<td>3</td>
<td>Number of iteration in local search</td>
</tr>
<tr>
<td>alpha</td>
<td>0.8</td>
<td>Learning rate of reinforcement learning</td>
</tr>
</tbody>
</table>

The machine learning classifier was used python library called scikit-learn (version 0.24.2). hyper-parameter of classifier was used default parameter.

IV. RESULT

The optimal combination list of features in training set selected by the feature selection method is shown Table 5.

Table 5: Optimal feature lists

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Optimal feature combination lists</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>Destination port</td>
</tr>
<tr>
<td></td>
<td>NAT Source Port</td>
</tr>
<tr>
<td></td>
<td>NAT Destination Port</td>
</tr>
<tr>
<td></td>
<td>Bytes</td>
</tr>
<tr>
<td></td>
<td>Bytes Received</td>
</tr>
<tr>
<td></td>
<td>Packets</td>
</tr>
<tr>
<td></td>
<td>Elapsed Time</td>
</tr>
<tr>
<td></td>
<td>Packets sent</td>
</tr>
<tr>
<td></td>
<td>Packets received</td>
</tr>
<tr>
<td>KNN</td>
<td>Destination Port</td>
</tr>
<tr>
<td></td>
<td>NAT Destination Port</td>
</tr>
<tr>
<td></td>
<td>Bytes</td>
</tr>
<tr>
<td></td>
<td>Bytes Sent</td>
</tr>
<tr>
<td></td>
<td>Packets</td>
</tr>
<tr>
<td></td>
<td>Elapsed Time</td>
</tr>
<tr>
<td>Bernoulli NB</td>
<td>Source Port</td>
</tr>
<tr>
<td></td>
<td>Bytes</td>
</tr>
<tr>
<td></td>
<td>Bytes Sent</td>
</tr>
<tr>
<td></td>
<td>Elapsed Time</td>
</tr>
<tr>
<td></td>
<td>Packets received</td>
</tr>
</tbody>
</table>

And we evaluate accuracy performance each classifier. Table 6 show the best accuracy among combination lists each classifiers.

Table 6: Best accuracy (%)

<table>
<thead>
<tr>
<th>Classifiers</th>
<th>Best accuracy among combination lists</th>
</tr>
</thead>
<tbody>
<tr>
<td>RF</td>
<td>99.87</td>
</tr>
<tr>
<td>KNN</td>
<td>99.78</td>
</tr>
<tr>
<td>Bernoulli NB</td>
<td>80.22</td>
</tr>
</tbody>
</table>

And we measure random forest performance by average accuracy and average macro-averaged precision (Macro-precision), average macro-averaged recall (Macro-recall), average macro-averaged F1 (Macro-F1) score. We compared the optimal selected feature results with the case of applying all features using random forest. The results are shown in Table 7.
Table 7: Comparing all and optimal features using RF (%)

<table>
<thead>
<tr>
<th>Feature selected</th>
<th>Average Accuracy</th>
<th>Macro-precision</th>
<th>Macro-recall</th>
<th>Macro-F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>All</td>
<td>99.80</td>
<td>93.06</td>
<td>81.71</td>
<td>84.22</td>
</tr>
<tr>
<td>Optimal</td>
<td><strong>99.87</strong></td>
<td><strong>96.62</strong></td>
<td><strong>88.73</strong></td>
<td><strong>91.27</strong></td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this paper, we classified firewall logs using optimal feature via BSO with reinforcement learning feature selection method. The results of using optimal features outperformed using all features and it could be applied to a firewall log analysis.

REFERENCES


