An improved bio-inspired based intrusion detection model for a cyberspace

Samera Uga Otor, Bodunde Odunola Akinyemi, Temitope Adegboye Aladesanmi, Ganiyu Adesola Aderounmu and B. H. Kamagaté

Cogent Engineering (2021), 8: 1859667
COMPUTER SCIENCE | RESEARCH ARTICLE

An improved bio-inspired based intrusion detection model for a cyberspace

Samera Uga Otor¹, Bodunde Odunola Akinyemi²*, Temitope Adegbeye Aladesanmi³, Ganiyu Adesola Aderounmu⁴ and B. H. Kamagate⁵

Abstract: Bio-inspired intrusion detection solutions provide better detection accuracy than conventional solutions in securing cyberspace. However, existing bio-inspired anomaly-based intrusion detection systems are still faced with challenges of high false-positive rates because the algorithms were tuned with unpredictable user-defined parameters, which led to premature convergence, exploration and exploitation discrepancies, algorithm complexity, and unrealistic results. In this paper, an intrusion detection system based on the foraging behavior of the social spider was developed. It employed signal transmission variables such as frequency of vibration to achieve a system that can evaluate real-life signals transmitted by computers and computing devices in the cyberspace to detect intrusion. This intrusion detection system was formulated using a social spider colony optimization model to generate a classifier that was tested using the standard NSL-KDD and live network traffic OAUnet datasets. The performance of the proposed intrusion

ABOUT THE AUTHORS

Samera Uga Otor is a lecturer at Benue State University, Makurdi, Nigeria. Her research interests are Cybersecurity and Green technology. She is a Member of the Nigeria Computer Society (MNCS).

Bodunde Odunola Akinyemi is a Senior Lecturer at Obafemi Awolowo University, Ile-Ife, Nigeria. Her current research interest include Data communication and networking. She is a Member of the Nigeria Computer Society (MNCS) and a chartered IT practitioner (MCPN).

Temitope Adegbeye Aledesanmi is an IT staff at Obafemi Awolowo University, Ile-Ife, Nigeria. He is a chartered IT practitioner (MCPN); a life member of the Nigeria Computer Society (MNCS).

Ganiyu Adesola Aderounmu is a professor at Obafemi Awolowo University, Ile-Ife, Nigeria. He is a Full member of the Nigeria Society of Engineers (FNSE), Nigeria Computer Society (FNCS), a chartered IT practitioner (CPN) and certified Engineer (COREN).

B. H. Kamagate is a Lecturer and researcher at ESATIC (Ecole Supérieure Africaine des Technologies de l’Information et de la Communication), and LARIT (Laboratoire de Recherche en Informatique et Télécommunication), Abidjan, Côte d’Ivoire.

PUBLIC INTEREST STATEMENT

Attacks are known to compromise the assets, resources, rules, policy, and guidelines of cyberspace. These compromises adversely affect the cyberspace performances by altering or tampering with the Confidentiality, Integrity, and Availability (CIA) of the network. The upsurge of intrusions has significantly deteriorated the Quality of Services (QoS) of cyberspace. The degeneration of the performance of networks has given rise to researches on the detection of intrusions efficiently and accurately. In securing the cyberspace, bio-inspired intrusion detection solutions provide better detection capabilities than conventional solutions. However, the bio-inspired algorithms for intrusion detections in cyberspace require a self-tuned search pattern and predictable user-defined parameters to achieve a fast convergence detection process with a high detection accuracy and low false-positive rates. In this study, a bio-inspired optimization model capable of self-tuning its search pattern to achieve fast convergence is employed to detect intrusions in the cyberspace to enhance its performances and facilitate efficient QoS.
detection system was evaluated by benchmarking it with existing classifiers using detection accuracy, sensitivity, and specificity as performance metrics. Results showed that the proposed model was more effective in terms of higher detection accuracy, sensitivity, specificity, and F-measure with a low false-positive rate. This showed that the spider model is a robust computational scheme that improves intrusion detection with a minimal false-positive rate in cyberspace.

**Subjects**: Biology; Algorithms & Complexity; Computing & IT Security; Computer Science; General

**Keywords**: cyberspace; intrusion detection; foraging behavior; bio-inspired

1. Introduction

Intrusion problem is inevitable in a continuously growing network with the influx of new network-enabled devices as attackers continue to devise means of breaching even existing security measures. Organizations that best manage these cybersecurity challenges will be those that quickly detect and have a strategy in place to tackle them (WaterISAC, 2015). Delay in the detection of these breaches results in several losses to individuals and the nation as a whole. In the light of this awareness, intrusion detection is one of the twelve (12) cyber-security areas the United States Government devoted attention to due to its vital nature to the economy (Andress, 2014).

Cybersecurity has been a subject of attraction to researchers due to the insecure nature of the cyberspace. According to Maurer (2011), “cyberspace is an operational domain framed by the use of electronics to exploit information via interconnected systems and their associated infrastructure.” It comprises both physical and virtual properties, hardware and software, all computer networks in the world including the Internet as well as other networks not linked to the Internet.

The interconnectedness of cyberspace as well as the alarming rate at which the network and the network-enabled devices are growing, and the increase in the knowledge of attackers and intruders puts the cyberspace under threat. The effect of these threats could result in a huge loss to both individuals and the economy. Therefore, securing cyberspace or cyber-security has become crucial. In cyberspace, an intrusion is any set of actions that attempt to compromise the Confidentiality, Integrity, or Availability (CIA) of a resource (Aissa & Guerroumi, 2016). Since the overall goal of network management is to maximize its performance, the compromise of the CIA affects the performance of a network. Thus, the need for a measure to minimize these compromises (Akinyemi et al., 2015).

Security measures such as Intrusion Detection System (IDS) and Intrusion Prevention Systems (IPS) among others are examples of cyber-security tools, concepts, and technologies used to minimize these threats. Intrusion Detection and Prevention Systems are systems used to monitor the activities occurring in a computer system or network and scrutinizing them for signs of likely violations or forthcoming threats of violation of computer security policies, acceptable use policies, or standard security practices and attempting to stop detected possible occurrences. These systems primarily focus on identifying possible instances, logging information about them, attempting to stop them, and reporting them to security administrators (Scarfone & Mell, 2012). Intrusion detection systems based on the techniques used to detect attacks (Liao et al., 2013), can be classified as Signature or Misuse-based and Anomaly-based. This paper focuses on Anomaly-based intrusion detection system in cyberspace.

Anomaly-based systems first build a model of the normal behavior of the protected target, then monitors the characteristics of cyberspace activities over a period to detect any deviations in these activities from the model previously built. Consequently, any deviation from this model is considered as suspicious. Anomaly-based intrusion detection systems are usually faced with challenges
of alarming false alarm rates because it is difficult to differentiate what is normal from abnormal both resulting in the consumption of computing resources. Therefore, the need to develop a system that can model real-life transmission parameters such as the frequency to achieve a low false alarm rate.

There exist conventional and bio-inspired solutions to these cyberspace’s intrusion problems. Interestingly, bio-inspired intrusion detection solutions provide better detection accuracy than their counterparts in securing cyberspace (Cuevas et al., 2013). However, Yangi (2014) stated that, though bio-inspired algorithms are highly successful, they still have room for improvement by learning more and more from nature and by carrying out ever-increasingly detailed, systematic studies, to develop truly “smart” self-evolving algorithms so that such smart algorithms can automatically fine-tune their behavior to find the most efficient way of solving a complex problem. Also, it was noted in Otor et al. (2019) that existing bio-inspired algorithms were tuned with unpredictable user-defined parameters, which led to premature convergence, exploration and exploitation discrepancies, algorithm complexity (in terms of the number of user-defined parameters), and unrealistic results. Meanwhile, studies regarding bio-inspired optimization models with a self-tuned search pattern and fast convergence for intrusion detection in cyberspace are scarce.

Thus, these identified problems of bio-inspired algorithms were addressed in the model proposed by Otor et al. (2019). The authors formulated a bio-inspired optimization model that mimics the foraging behavior of a social spider called the Adaptive Social Spider Colony Optimisation (ASSCO) model. ASSCO modeled frequency which is a measurable parameter in signal transmission, thus, enabling real-life applications in computing.

In furtherance, an attempt was made in this paper to develop an intrusion detection model that leverage on Otor et al. (2019)’s bio-inspired optimization model that is capable of self-tuning its search pattern to achieve fast convergence while maintaining its balance in exploration and exploitation to avoid premature convergence. This was intending to achieve improved detection accuracy, sensitivity, specificity, and low false-positive rate for intrusion detection in cyberspace.

The remaining part of the paper is organized as follows: section two gives a brief review of related works, section three gives the detailed background of the study, section four describes the materials and methodology employed, section five gives a detailed analysis of results obtained from simulation and section six is the conclusion and future research areas opened up.

2. Related works
Several bio-inspired intrusion detection systems have been developed by researchers using bio-inspired optimization algorithms as feature selection, classification, and clustering of intrusion datasets as stand-alone or hybridized with other machine learning techniques like Complex Tree, Naïve Bayes, Support Vector Machine and K-Nearest Neighbor among others. They include; The Ant Colony Optimization (ACO) algorithms. These are bio-inspired optimization algorithm based on the foraging behavior of ants (Colorni et al., 1991; Dorigo et al., 1996, 1991). They have been employed to develop intrusion detection systems as data classifiers (Zhu & Li, 2016), to filter packet in the firewall rule (Sreelaja & Vijayalakshmi, 2010), to improve classifiers’ performance in terms of accuracy and running time (Bamakan et al., 2015) and as a feature selection method to reduce dimensionality of intrusion dataset (Chung & Wahid, 2012). It was also used to select important features used to build an intrusion system for efficient and effective detection with low computational complexity (Aghdam & Kabiri, 2016).

To further improve the results of feature selection, ACO was hybridized with feature weighting Support Vector Machine (SVM) classifier (Xingzhu, 2015), to find the optimal combination of parameters and to improve the learning performance and generalization ability of the SVM model and also to establish the optimal data classification model (Zhu & Li, 2016). Furthermore,
ACO was combined with a Genetic Algorithm (GA) as a hybrid method to obtain an effective system (Wan et al., 2016). Also, as a clustering method, ACO has been used for the clustering of intrusion data sets (Sharma & Gaur, 2016) and network vulnerability model to ensure network security (Chhikara & Patel, 2013; Sreelaja & Vijayalakshmi, 2014).

Another bio-inspired optimization algorithm employed to develop IDS is the Particle Swarm Optimization (PSO). In computer security (i.e. intrusion detection) and data mining, it has been applied either as a standalone or as a hybrid bio-inspired technique in combination with machine learning methods as follows: it was applied in combination with the Fuzzy learning technique for data clustering (Kavitha & Ranjitha, 2013; Kumutha & Palaniammal, 2014; Niu & Huang, 2011); with K-means clustering as a classifier (Li et al., 2011; Zheng, 2011) and to enhance the sensitivity problem of initial cluster centroid as well as premature convergence of clustering methods (Aljarah & Ludwig, 2013).

PSO has also been hybridized with other bio-inspired algorithms such as Genetic Algorithm (GA) to produce good classification rule for IDS (Kumar, 2013). Other application areas include; classification (Sujitha & Kavitha, 2015; Tsai & Chen, 2015), feature selection of intrusion dataset (Nejad et al., 2015), and parameter tuning problem of multiple criteria linear programming (MCLP) classifier to achieve better performance of MCLP classification of intrusion dataset (Bamakan et al., 2015).

Artificial Bee Colony Optimization (ABC) is also another bio-inspired optimization model employed to solve the intrusion problem. ABC and Multilayer Perception (MLP) were used as a hybrid method to address the problem of intrusion detection (Mahmod & Alnaish, 2015). In the same vein, modified Artificial Bee Colony was integrated with Enhanced PSO to find out better optimization results and classification accuracies through cross-validation in IDS data (Amudha et al., 2015).

More so, the Bat Algorithm, another bio-inspired algorithm has been used as a classifier for the intrusion dataset (Khan et al., 2011) and in combination with K-Means for data clustering (Komarasamy & Wahi, 2012).

The spider algorithm developed by Cuevas et al. (2013) has also been applied as a feature selection method (Anter et al., 2015) and clustering of a high dimensional dataset (Shukla & Nanda, 2016; Zhou et al., 2017; Chandran et al., 2017).

Therefore, this work seeks to further address the problem of intrusion detection with better accuracy and low false alarm rate by leveraging on the advantages of bio-inspired models and the uniqueness of the spider models. The main strength of the spider model is that their communication is based on signals which can be quantified rather than pheromone and other measures that are rather chemical perceptions and can only be assumed.

3. Materials and method
The conceptual framework of the proposed model is presented in Figure 1. The details of the proposed model's concept and formulation are as follows:

3.1. Biological concept of social spider colony
The concept of the foraging behavior of spider starts by the spiders building a web (a hyperdimensional surface) which can be likened to devices linked up to form a cyberspace, after which the spider retires either to the hub or a shrub (similar to IDS host) attached to the web (cyberspace) waiting for a prey to be caught on the web. As soon as the prey is caught, a vibration signal is sent to the spider, this signal transmits the information about the position of the prey to the spider. The design of the webs of some species such as Achaearaneaesteselata and Leucauagemariana also transmits information regarding the location of the prey trapped on the sheet to reach the resting spider inside the retreat.
The first response of the spider in her retreat is to turn and face the prey; the spider then climbs down along the mesh threads following almost a straight line to the prey. Some species such as the *Leucaugemariana* spider stops intermittently for some seconds, jerks the web as they move towards the prey to sense the direction of vibration from the prey and move towards it until they get to the prey. The prey may escape if the spider does not get to it on time; therefore, prey capture success is partly determined by the speed with which the spider reaches the prey, which in turn is probably largely dependent on the information the spider receives regarding prey location in the web (Biere & Uetz, 1981). The average time between the times of descending to the time of attack for the *Achaearaneatessellata specie* is 0.11 seconds and the deviation between spiders approaching direction and that of the prey position ranges from a minimum of zero degrees to a maximum of one hundred and eighty degrees (0° to 180°) respectively (Barrantes & Weng, 2006). The spiders also communicate between themselves through signals; therefore, the spiders can differentiate the vibration of neighboring spiders from that of the prey no matter how minute, with all sense of accuracy. This accuracy of detection is represented in this work as the difference in frequency composition of vibration (signal) of the prey (attacks) to that of neighboring spiders (monitoring devices) sensed by the spiders on the web threads (links).
Social spider foraging is a result of effective communication of vibration signals perceived by spiders either from neighboring spiders or from the prey. The spider is known to differentiate between the vibrations from the prey and that of neighboring spiders; therefore, the proposed model employed this behavior by using the Adaptive Social Spider Colony Optimization (ASSCO) to select optimal features from the intrusion dataset as the search space using the griewank function as the objective function of the optimization problem. Furthermore, these features were used to generate a trained classifier through simulation which was tested using test datasets for intrusion detection. The choice of ASSCO is because it employed a frequency of the signal as the parameter that determines the solution quality of the optimization problem, as computers exchange information through the signal transmission.

3.2. Cyberspace and social spider colony commonalities

The cyberspace comprises computers, internet-enabled devices such as mobile phones, servers, routers, and other components of the internet infrastructures linked together. Relating it to the Adaptive Social Spider Colony Optimization (ASSCO) Model Otor et al. (2019), which is based on the foraging behavior of a social spider, the spider web is the virtual space (similar to cyberspace created by the linked devices, i.e. the computers, internet-enabled devices such as mobile phones, servers, routers, and other components of the internet infrastructures) while the colonies of social spider on the web are monitoring devices that monitor intrusion in the space and the attackers (i.e. intruders) are the prey.

This research, therefore, modeled the spider foraging behavior using a simplified wave transmission model with prey (attacks) as a source of wave, producing a signal that propagates through the web (cyberspace) to the spiders (monitoring devices) as receivers. The prey/source can be characterized by the frequency composition of its signals and the spider/receiver is characterized by its frequency-dependent sensitivity to signals. These signals are described by a transfer function proposed by Otor et al. (2019).

Thus, the conceptual framework presented in Figure 1 showed the Spiders (monitoring devices) and prey (attackers) on the web (cyberspace) communicate via vibration cues (signals) transmitted over the web.

3.3. Model design

In this paper, an intrusion detection system based on the foraging behavior of a social spider to alleviate the false alarm rate of anomaly-based intrusion detection systems was formulated. The Intrusion Detection System (IDS) developed is an anomaly-based IDS known as the Adaptive Social Spider Colony Intrusion Detection System (ASSCO-IDS). ASSCO-IDS simulates the spiders as the monitoring agents, the preys as intruders i.e. spider web as cyberspace.

In this study, the model design leveraged on the design constraints of ASSCO (Otor et al., 2019). As an optimization problem, the dataset represents the search space, the features of the dataset represent the positions of the spider on the search space, and the attack type i.e. intrusions represents the prey. The Adaptive Social Spider Colony Optimization Intrusion Detection system (ASSCO-IDS) parameter set up is as shown in Table 1. ASSCO-IDS consists of three modules, namely: the data collection, feature selection, and data classification modules described as follows:

3.3.1. Data collection module of the adaptive social spider colony optimization intrusion detection system

The data sources used in this study are the Network Socket Layer Knowledge Discovery in Data Cup 99 competition (NSL-KDDCUP’99), a subset of KDDCUP’99 dataset, and live network traffic data that were captured on Obafemi Awolowo University network (OAUnet) by Akinyemi et al. (2019). OAUnet dataset was captured in the Cybersecurity laboratory of the African Center of Excellence, which is a control room where outbound and inbound traffic are free from intrusions. The
environment also allows one to execute real-life attack scripts against some targeted machine which would be unacceptable if run against production machines. The environment also made it easy to filter out unwanted packets during network attack traffic collection. The detailed setup of the network environment is as presented in Figure 2 and the detailed description of the capture method is described in Akinyemi et al. (2019).

### 3.3.2. Feature selection module of the adaptive social spider colony optimization intrusion detection system

The feature selection approach employed for the ASSCO intrusion detection system is an enhanced wrapper method. It is an enhanced wrapper method because feature selection and classification were not done simultaneously as in wrapper methods. However, the method employed characteristics of the wrapper method by using the accuracy of a classifier to evaluate the feature subset to generate a trained classifier. The feature selection method proceeded as follows:

| Parameters            | Values                                      |
|-----------------------|---------------------------------------------|
| Number of Spiders     | Total features for each attack type         |
| Number of spiders     | Attack type                                 |
| Maximum Iterations    | 2000                                        |
| Method                | Bag                                         |
| Type                  | Classification                              |
| Learner               | Tree                                        |
| Number of Trees       | 30                                          |
| Cross-validation Partition | 25%         |
| Split Criteria        | Gini Diversity Index (GDI)                 |
(a) Normalizing Categorical Data: The test data contains some categorical feature values that need to be converted to numeric data to enable the developed social spider model (ASSCO) to perform mathematical operations on them. Such features include the class, the flag, the protocol type, and service. An algorithm was developed that assigned values ranging from 1 ... n with n being the maximum number of unique data points and the class data assigned the value 0 for anomaly and 1 for normal. The features after normalization were then outputted to an excel table. The algorithm is as presented in Algorithm 1.

(b) Selecting Features for Classification: To enhance the accuracy and training speed of the classifier, the dimensionality of the datasets was reduced through feature selection as follows:

(i) Given a full feature set $S_i = \{S_j|i = 1 \ldots N, J\}$ of the normalized datasets, which is the total data points in the dataset, a random dataset $A_M = \{x_{i1}, x_{i2}, \ldots x_{im}\}$ was generated within the lower bound and upper bound of the feature set $S_i$ using Equation (1) (Otor et al., 2019).

$$x_i = x_i(l) + \text{rand}(1, d) \cdot (x_i(u) - x_i(l)) \text{ for } i = 1, 2, \ldots n \tag{1}$$

Where $\text{rand}(1, d)$ helps to distribute the positions randomly and $d$ is the dimension of the search space defined in the problem.

This represents the position of spiders in the search space (i.e. the dataset) with $d$ equal to the total number of features. Each spider’s position $A_M$ is a column vector of the feature values they represent.

(ii) An initial subset of features consisting of the Class feature $S_j$ was also selected for a start. This is because the class feature represents the decision point (Attack/Normal or Anomaly/Normal) which the supervised learning algorithm relies on to learn the data pattern.

(iii) The subset $A_M$ is then evaluated using the fitness function to generate the optimal feature. The fitness function is defined as the Griewank test function, given as Equation (2):

$$f_{15}(x) = \frac{1}{4000} \sum_{i=1}^{n} x_i^2 - \prod_{i=1}^{n} \cos \left( \frac{x_i}{\sqrt{i}} \right) + 1 \tag{2}$$

(iv) For each iteration, the ASSCO algorithm (Otor et al., 2019) was applied to generate the best feature from the data points $A_M = \{x_{i1}, x_{i2}, \ldots x_{im}\}$ used to update $f_{best}$ which is the best vibration among the spiders.

(v) For each iteration also, the optimal feature obtained from minimization, the $\text{Ftar}$ is added to the subset $S_j$ generated in (iii)

(vi) Step (iii) is repeated until the optimal subset $S_j$ is generated (i.e. until maximum iteration is reached).

(vii) The subset generated is then outputted.

The feature selection algorithm is as shown in Algorithm 2.
Algorithm 1: Categorical features Normalization Algorithm

```
Input {read table(training_feature.xlsx)
Output write table(training_sample.xlsx)}
1. read flag
   read unique feature of flag
   Initialize labels 1
   for i = 1: size of flag
   label1(ismember(flag)) = i
   end for
2. read protocol_type
   read unique protocol feature
   Initialize label 2
   for j = 1: size of protocol_type
   label2(ismember(protocol_type)) = j
   end for
3. read service
   read unique service
   Initialize label 3
   for k = 1: size of service
   label3(ismember(service)) = k
   end for
4. read class
   read unique class
   classval1 = "normal"
   classval2 = compare (classval1, unique class)
   class = classval2
   newtraining = data
   writetable(newtraining, "training_sample.xlsx")
5. end
```

Algorithm 2: Feature Selection Algorithm

```
Input {read table(Normalized_features)
Output selected_features}
1. Initialize subset
   \( S_j = \{Class\} \)
2. generate data points
   for i = 1: total features
   \( S_i = lowerbound(i) + rand(upperbound(i) - lowerbound(i)) \)
   end for
3. While not maximum iteration
   apply ASSCO algorithm(Otor et al., 2019) and update \( f_{best} \)
   iterate until \( f_{star} \) is found
   \( S_j = \{Class\} + f_{star} \)
4. repeat step 3 until maximum iteration is reached
5. output \( S_j \)
6. end
```

3.3.3. Classification method of adaptive social spider colony optimization intrusion detection system

The feature subset generated from the feature selection stage was evaluated using a 25% cross-validation partition to obtain a trained classifier. The 25% cross-validation partition divides the
dataset into a 75% training set and a 25% test set. An ensemble bagged tree learning algorithm was used to grow the subset of features generated by ASSCO to generate a trained classifier (the ASSCO-IDS learner). The ensemble bagged tree learner algorithm uses a bootstrap aggregation to reduce the variance of the decision tree algorithm. Variance is a measure of how often these algorithms disagree. It consists of several decision tree algorithms. The decision tree algorithm for a binary tree splits the branches at each node using a split criterion the Gini diversity index with each node representing the features, the splits (branches) represent the classification rule, and the leaves the decision points. The ensemble bagged tree algorithm is presented in Algorithm 3.

Algorithm 3: The Ensemble Bagged Learning Algorithm

```plaintext
Input { optimalFeatures }
Output Classification Accuracy, confusion matrix ROC curve
1. Load optimal features
   predictorsName=optimalFeatures
   response=Class
   IsCategorical={true, False...}
2. Set up holdout validation using cross validation partition(cvp)
   cvp = 0.25
3. Train the classifier
   for i=1:totalFeatures
     classificationEnsemble = fitensemble (trainingPredictors, trainingResponse,
     aggregationMethod=Bag, numberOfTrees=30, classification=tree, split criterion=gdi,,
     classNames={normal; normal})
   endfor
4. Create the result structures with predict function
   ensemblePredictFunction= predict(classificationEnsemble, predictors)
   validationPredictionFunction=ensemblePredictFunction(predictors)
5. Compute validation accuracy
   validationPredictors=predictors(cvp.test)
   validationResponse=response(cvp.test)
   [validationPredictions, validationScores] = validationPredictFcn(validationPredictors)
   correctPredictions = strcmp(validationPredictions, validationResponse)
   validationAccuracy = sum(correctPredictions)/length(correctPredictions)
   confusionmat=confusionmat(validationPredictions, validationResponse)
   [FpD2, TpD2, T, AUC, OPTROCP,TD2] = perfcurve(validationResponse, validationScores(:,2), 'normal')
   Plot ROC
6. output Accuracy, confusion Matrix, ROC curve
7. end
```

The Adaptive Social Spider Colony Optimization (ASSCO) model (Otor et al., 2019) was used to optimize the classification accuracy of the bagged tree classifier by selecting the best set of features based on the position of the spiders (features) with the strongest vibration (best feature set) in the entire population of spiders (total features). This was done by optimizing the frequency of vibration and the distance between individuals using Euclidean distance to select the optimized feature subset. The ASSCO-IDS learner algorithm consists of the following steps:

(a) For $N$ observations with $M$ features in the training dataset, a subset of features was generated using the ASSCO algorithm (Otor et al., 2019) with replacement. This enhances the classifier’s accuracy by selecting optimal features. This differs from the existing ensemble bagged tree that selects features at random.

(b) The features selected in (a) were used to grow each of the trees in the bag of trees in the ensemble learner using the tree algorithm to generate classification accuracy, starting with the feature that gives the best split using the Gini Diversity Index split criterion.
\[ \text{GDI} = 1 - \sum_{i=1}^{n} p(i)^2 \quad \text{where } p(i) \text{ is the observed fraction of classes with class } i \text{ that reaches the node.} \]

(c) Step (a) is repeated until the tree is fully grown and the maximum iteration is reached.

(d) The classification accuracies obtained from each of the trees were averaged to generate the ASSCO-IDS accuracy.

(e) The ASSCO-IDS accuracy is outputted alongside the confusion matrix and Receiver Operative Curve (ROC). The ASSCO-IDS algorithm is as shown in Algorithm 4.

Algorithm 4: The Adaptive Social Spider Colony Optimisation Intrusion Detection Classifier Algorithm

```plaintext
Input {read table(Normalized_features)
Output selected_features, Accuracy, confusion matrix ROC curve
1. initialize subset S_j = \{Class\} 
2. generate data_points
   for i=1:totalfeatures
      S_i = lowerbound(i) + rand(upperbound(i) - lowerbound(i))
   end for
3. While not maximum iteration
   apply algorithm 1 and update f_best
   iterate until f_tar is found
   S_j = \{Class\} + f_tar
3. Load optimal features=S_j
   predictorsNames = optimalFeatures
   response = Class
   IsCategorical = [true, False ...]
4. Set up holdout validation using cross-validation partition(cv)
   cvp = 0.25
5. Train the classifier
   TotalAccuracy = 0.000
   for i=1:totaloptimalfeatures
      classificationEnsemble
      = fitemensly (trainingPredictors, trainingResponse, aggregationMethod = Bag, numberOfTrees = 30, classification = tree, split criterion = gdi, classNames\{anomaly, normal\})
   6. Create the result structures with predict function
      ensemblePredictFunction = predict(classificationEnsemble, predictors)
      validationPredictFunction = ensemblePredictFunction(predictors)
7. Compute validation accuracy
   validationPredictors = predictors(cvp.test)
   validationResponse = response(cvp.test)
   [validationPredictions, validationScores] = validationPredictFcn(validationPredictor: correctPredictions = strcmp(validationPredictions, validationResponse)
   validationAccuracy = sum(correctPredictions) / length(correctPredictions)
   confusionMatrix = confusionmat(validationPredictions, validationResponse)
   [FalsePositive, TruePositive, T, AUC, OPTROCPT] = perfcurve(validationResponse, validationScores(:,2), 'normal'
) for endfor
   TotalAccuracy = TotalAccuracy + validationAccuracy
8. repeat step 3 until maximum iteration is reached
9. Accuracy = TotalAccuracy/maximum Iteration
endWhile
10. output S_j, Accuracy, confusionMatrix, ROC curve
11. end
```
4. Results and discussions
A simulation was carried out in MATLAB 2016a environment to simulate the proposed model.

The Network Socket Layer Knowledge Discovery in Data Cup 99 competition (NSL-KDDCUP’99) a subset of KDDCUP’99 dataset and a live dataset collected in the Cybersecurity Laboratory of Obafemi Awolowo University (OAUnet Dataset) was used to test the model. Both datasets were used to detect intrusions of Denial of Service (DoS), User to Root (U2R), Probe, and Remote to Local (R2L) attacks by employing the proposed ASSCO-IDS model. The NSL_KDDCUP’99 dataset consists of 42 features including the class feature while the OAUnet dataset consists of 52 features including the class feature. The datasets were pre-processed and the results obtained are as follows:

4.1. Feature selection results
From the full feature sets of the normalized datasets, a random dataset was generated within the lower bound and upper bound (i.e. 1, 41 for NSL-KDD dataset and 1, 52 for OAUnet dataset) of the ASSCO model, these represent the position of spiders in the search space (i.e. the dataset)

The features selected using the ASSCO model for both the OAUnet and the NSL-KDD datasets are presented in Tables 2 and Tables 3. For the OAUnet dataset, a total of 45, 39, 35, and 35 features out of 52 features were selected for DoS, Probe, Remote to Local (R2L), and User to Root (U2R) attacks, respectively. This represents a dimensionality reduction of 13.5%, 25%, and 32.7% in the features of DoS, Probe, Remote to Local, and User to Root attack types.

For the NSL-KDD dataset, a total of 34, 29, 24, and 18 features out of 41 features were selected for DoS, Probe, Remote to Local, and User to Root, respectively. This represents a dimensionality reduction of 17.1%, 29.27%, 41.46%, and 56.1% in features of each of the attack types. This dimensionality reduction reduces the classifier complexity and enhances the speed of the classifier.

4.2. Classification results
For validation of the datasets obtained from the feature selection stage, a cross-validation technique was applied to split the datasets into 75% training datasets and 25% test datasets. To detect intrusions in the datasets, the proposed model ASSCO-IDS was trained and tested with 75% and 25% of each dataset, respectively. The data statistics are presented in Tables 4 and

| Table 2. Selected Features of the OAUnet datasets |
|-----------------------------------------------|
| **Attack Category** | **Number of Features Selected** |
| DoS                | 45 |
| Probe              | 39 |
| R2L                | 35 |
| U2R                | 35 |

| Table 3. Selected Features of the NSL-KDD datasets |
|-----------------------------------------------|
| **Attack Category** | **Number of Features Selected** |
| DoS                | 34 |
| Probe              | 29 |
| R2L                | 24 |
| U2R                | 18 |
Tables 5 for the OAUnet and NSL-KDD datasets, respectively. Also, an existing bagged tree classification model was used in the same datasets. The choice of the bagged ensemble method was because ASSCO-IDS employed an ensemble bagged learner algorithm in its design and also bagging model trees outperform state-of-the-art decision tree ensembles on problems with numeric and binary attributes, and, more often than not, on problems with multi-valued nominal attributes too (Kotsiantis et al., 2005). They are also known for high performance in terms of accuracy specificity and sensitivity (Al-Barazanchi et al., 2017).

The results obtained from the simulation for DoS, probe, Remote to Local, and User to Root attack types are presented in Tables 6 and Tables 7, for the OAUnet and NSL-KDD datasets, respectively. In Table 6, the result showed that ASSCO-IDS was able to classify all attack types efficiently and misclassified only 1 out of 6311, 2115, and 1578 non-attacks as an attack for DoS, Probe, R2L and U2R attack types. Also, in Table 7, all attacks were correctly classified for U2R and 1 attack for each DoS and probe, three attacks for R2L were misclassified out of 11,256, 2913, and 249 while 1 non-attack for each probe and U2R out of 1164 and 5 were misclassified. Consequently, this implied that ASSCO-IDS was able to achieve low false-positive rates for both datasets with Area Under Curve (AUC) of the Receiver Operative Curve (ROC) equal to 1 as shown in Figure 3.

Also, the True Positives, True Negatives, False Positives, and False Negatives rates obtained from bagged tree and ASSCO-IDS are presented in Tables 8 and Tables 9 for all attack types using both the NSL-KDD and the OAUnet datasets.

| Table 4. OAUnet Dataset Statistics |
|------------------------------------|
| Attack Types | Total No. of Dataset | Anomaly | Normal | Training Set | Test Set |
| DoS | 67176 | 46138 | 21038 | 47024 | 20152 |
| Probe | 23503 | 7051 | 16457 | 16453 | 7050 |
| User2Root | 22140 | 6642 | 15498 | 16605 | 5335 |
| Remote2Local | 21052 | 6315 | 14737 | 15789 | 5263 |

| Table 5. NSL-KDD Dataset Statistics |
|-------------------------------------|
| Attack Types | Total No. of Dataset | Anomaly | Normal | Training Set | Test Set |
| DoS | 67025 | 45022 | 22003 | 50269 | 16756 |
| Probe | 16318 | 11656 | 4662 | 12239 | 4079 |
| User2Root | 73 | 52 | 21 | 55 | 18 |
| Remote2Local | 1393 | 995 | 398 | 1045 | 348 |

| Table 6. Simulation Results for all Attack Classification on ASSCO-IDS using OAU Dataset |
|----------------------------------------|
| Attack Types | True Positives | True Negatives | False Positives | False Negatives |
| DoS | 13840 | 6311 | 0 | 1 |
| Probe | 4934 | 2115 | 0 | 1 |
| R2L | 3684 | 1578 | 0 | 1 |
| U2R | 3874 | 1661 | 0 | 0 |
4.3. Model performance evaluation results

The performance of the proposed ASSCO-IDS classifier was evaluated by benchmarking it with an existing ensemble bagged tree classifier and the model developed by Akinyemi et al. (2019), through

| Attack Types | True Positives | True Negatives | False Positives | False Negatives |
|--------------|----------------|----------------|-----------------|-----------------|
| DoS          | 11256          | 5499           | 1               | 0               |
| Probe        | 2913           | 1164           | 1               | 1               |
| R2L          | 249            | 96             | 3               | 0               |
| U2R          | 12             | 5              | 0               | 1               |

Figure 3. Receiver operative curve for simulation of ASSCO-IDS.

Table 7. Simulation Results for all Attack Classification on ASSCO-IDS using NSL-KDD Dataset

| Attack Types       | True Positive | False Positive | True Negative | False Negative |
|--------------------|---------------|----------------|---------------|----------------|
| DoS                | 11255         | 4              | 5496          | 1              |
| Probe              | 2909          | 8              | 1157          | 5              |
| R2L                | 245           | 6              | 94            | 3              |
| U2R                | 10            | 0              | 5             | 3              |

Table 8. Confusion Matrix Results of ASSCO-IDS vs Bagged tree Using NSL-KDD datasets

Key
E = Existing(bagged tree)
P = Proposed(ASSCOIDS)
simulation using detection accuracy, sensitivity, specificity, and F-measure as metrics. The choice of Akinyemi et al. (2019) too was because ASSCO-IDS employed the network set up and the live data used in Akinyemi et al. (2019).

The detection accuracy, sensitivity, and specificity of the ASSCO-IDS, bagged tree, and Akinyemi et al. (2019)'s model were evaluated using the OAUnet dataset and NSL-KDD dataset. The detailed results are as shown in Tables 10 and Tables 11.

The detection accuracy graphs of ASSCO-IDS vs bagged tree shown in Figure 3 proved that the developed ASSCO-IDS detects intrusions in the OAUnet dataset with a slightly higher detection accuracy of 0.01%, 0.03% for DoS and probe attacks while maintaining the same detection accuracy for Remote to Local and User to Root attacks. In the NSL-KDD dataset, ASSCO-IDS shows higher detection accuracy over the bagged tree for all the attack types. In conclusion, results show that the proposed ASSCO-IDS performs better in detecting anomaly than the existing bagged tree classifier which is known for its high detection rate.

Also, the sensitivity graphs of ASSCO-IDS vs bagged tree are shown in Figure 4. Sensitivity represents the proportion of actual positives that are predicted positive. It is equivalent to the true positive rate. It tests the ability of the system to predict attacks correctly. In the OAUnet dataset, ASSCO-IDS achieved a slightly higher sensitivity rate of 0.01% for the DoS attack only. It maintains the same rate for other attacks despite the reduction in features. This implied that some of the features in the original data set were redundant. Meanwhile, in the NSL-KDD dataset, ASSCO-IDS achieved higher sensitivity rates of 0.14%, 1.21%, and 15.39% for Probe, R2L, and U2R attacks, respectively, while maintaining the same sensitivity rate for DoS attacks.

Also, the specificity graph of ASSCO-IDS vs bagged tree is shown in Figure 5. Specificity represents the proportion of actual negatives that are predicted negative. It represents the true negative rates and also evaluates how well a classifier predicts attacks. In the OAUnet dataset, ASSCO-IDS achieved significant higher specificity rate of 0.02% and 0.09% for DOS and Probes attacks, respectively, while maintaining the same rate for other attack types. While in the NSL-KDD dataset, ASSCO-IDS achieved higher sensitivity rates of 0.06%, 2.97%, and 0.60% for DoS, R2L, and probe attack types, respectively. It maintained the same specificity rate for U2R attacks.

In furtherance, the F-Measure of the two models was evaluated. In a statistical analysis of binary classification, F-measure is a measure of a test’s accuracy. The F-measure score is the harmonic average of the precision and recall, where an F-measure score reaches its best value at 1 (100%) (Perfect precision and recall) and worst at 0 (0%). In the OAUnet dataset, F-measure rates achieved by ASSCO-IDS were 0.01% and 0.02% slightly higher than bagged tree rates for the DoS,

| Attack Types       | True Positive | False Positive | True Negative | False Negative |
|--------------------|---------------|----------------|---------------|----------------|
|                    | E  | P  | E  | P  | E  | P  | E  | P  |
| DoS                | 13839 | 13840 | 1  | 0  | 6310 | 6311 | 2  | 1  |
| Probe              | 4934  | 4934  | 2  | 0  | 2113 | 2115 | 1  | 1  |
| Remote to Local    | 3683  | 3684  | 0  | 0  | 1579 | 1578 | 1  | 1  |
| User to Root       | 3875  | 3874  | 0  | 0  | 1660 | 1661 | 0  | 0  |

Key
E = Existing(baggedtree)
P = Proposed(ASSCOIDS)
| Existing/ proposed Model | DOS | Probe | R2L | U2R |
|--------------------------|-----|-------|-----|-----|
|                          | Accuracy (%) | Sensitivity (%) | Specificity (%) | Accuracy (%) | Sensitivity (%) | Specificity (%) | Accuracy (%) | Sensitivity (%) | Specificity (%) | Accuracy (%) | Sensitivity (%) | Specificity (%) |
| BaggedTree               | 99.985 | 99.985 | 99.984 | 99.957 | 99.979 | 99.905 | 99.981 | 99.973 | 100.000 | 100.000 | 100.000 |
| Akinyemi et al. (2019)   | 91.420 | 87.670 | 99.650 | 86.570 | 86.080 | 86.630 | 85.890 | 95.530 | 83.350 | 91.140 | 77.340 | 95.020 |
| ASSCO-IDS                | 99.995 | 99.992 | 100.000 | 99.985 | 99.979 | 100.000 | 99.981 | 99.973 | 100.000 | 100.000 | 100.000 | 100.000 |
| Existing/ proposed Model | DOS | Probe | R2L | U2R |
|--------------------------|-----|-------|-----|-----|
|                          | Accuracy (%) | Sensitivity (%) | Specificity (%) | Accuracy (%) | Sensitivity (%) | Specificity (%) | Accuracy (%) | Sensitivity (%) | Specificity (%) |
| BaggedTree               | 99.976         | 100.000          | 99.927          | 99.681        | 99.828          | 99.313          | 97.414        | 98.790          | 94.000          |
| Akinyemi et al. (2019)   | 95.976         | 99.981           | 98.023          | 96.875        | 86.080          | 89.080          | 85.971        | 97.303          | 83.971          |
| ASSCO-IDS                | 99.994         | 100.000          | 99.981          | 99.951        | 99.965          | 99.914          | 98.854        | 100.000         | 96.970          |
and Probe attacks, respectively. It maintained the same F-measures rate for Remote to Local and User to Root attack, respectively. While in the NSL-KDD dataset, F-measure rates achieved by ASSCO-IDS were significantly higher than the bagged tree for the DoS, Probe, Remote to Local, and User to Root attack, respectively Figure 6.

Figure 7 shows the evaluation results of the three models. These results showed that the developed social spider optimization intrusion detection system (ASSCO-IDS) was able to detect intrusions with higher detection accuracy, sensitivity, and specificity as compared to other models.
5. Conclusion
Cybersecurity is an ongoing trend in computing. New attacks on cyberspace are deployed continuously by attackers to breach existing security measures. Once a preventive measure is devised attackers device new means of attacks to counter it. Existing anomaly intrusion detection systems are faced with a high false alarm rate due to the difficulty in determining the normal traffic. In this paper, an intrusion detection model for cyberspace based on the foraging behavior of a social spider was proposed. The proposed IDS is known as an Adaptive Social Spider Colony Optimization (ASSCO-IDS). This is an IDS that employs the Adaptive Social Spider Colony Optimization (ASSCO) Model, a model whose fitness parameter is frequency. ASSCO was used to optimize classification accuracy based on the bagged tree classification technique to generate a trained classifier that was tested on two datasets, the NSL-KDD and OAUnet dataset (captured at the Obafemi Awolowo University Cybersecurity Laboratory) respectively. The social spider optimization intrusion detection system developed was able to detect intrusion on both NSL-KDD and OAUnet datasets with higher detection accuracy, specificity, sensitivity, and low false-positive rates for U2R, DoS, R2L, and probe attack types over the bagged tree. Therefore, the intrusion detection system developed can be adapted to alleviate intrusion detection in cyberspace. A real-time application is recommended to be deployed in the future work as the ASSCO model used measures the frequency of the signal which compares favorably with the signal transmission in computing.

Funding
This research was funded by the TETFund and Africa Centre of Excellence OAK-Park, Obafemi Awolowo University, Ile-Ife.

Author details
Samera Uga Otor
E-mail: jopadev@gmail.com
ORCID ID: http://orcid.org/0000-0002-9283-2909
Bodunde Odunola Akinremi
E-mail: bakinremi@oaife.edu.ng
ORCID ID: http://orcid.org/0000-0001-6943-7137
Temitope Adegbeye Aladesanmi
E-mail: taladesanmi@oaife.edu.ng
ORCID ID: http://orcid.org/0000-0003-2046-8738
Ganyu Adesola Aderounmu
E-mail: gaderoun@oaife.edu.ng
ORCID ID: http://orcid.org/0000-0002-7992-514X
B. H. Kamagaté
E-mail: beman2017@gmail.com
ORCID ID: http://orcid.org/0000-0002-8015-1815
1 Department of Computer Science, Benue State University, Makurdi, Nigeria.
2 Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria.
3 INTECU, Obafemi Awolowo University, Ile-Ife, Nigeria.
4 Department of Computer Science and Engineering, Obafemi Awolowo University, Ile-Ife, Nigeria.
5 Laboratoire LARIT, Cocody Danga, Abidjan, Cote D’Ivoire.

Citation information
Cite this article as: An improved bio-inspired based intrusion detection model for a cyberspace, Samera Uga Otor, Bodunde Odunola Akinremi, Temitope Adegbeye Aladesanmi, Ganyu Adesola Aderounmu & B. H. Kamagaté, Cogent Engineering (2021), 8: 1859667.

References
Aghadam, M. H., & Kabiri, P. (2016). Feature selection for intrusion detection system using ant colony optimization. International Journal of Network Security, 18(3), 420–432. http://ijns.femto.com.tw/contents/ijns-v18-n3/ijns-2016-v18-n3-p420-432.pdf
Aissa, N. B., & Guerroumi, M. (2016). Semi-supervised statistical approach for network anomaly detection. In Procedia Computer Science, The 6th International Symposium on Frontiers in Ambient and Mobile Systems (Vol. 83, pp. 1090–1095). Elsevier.
Akinyemi, B. O., Adekunle, J. B., Aladesanmi, T. A., Aderounmu, G. A., & Kamagaté, B. H. (2019). An improved anomalous intrusion detection model. FUOYE Journal of Engineering and Technology, 4(2), 81–88. https://doi.org/10.46792/fuyojet.v4i2.418
Akinyemi, B. O., Amoo, A. O., & Aderounmu, G. A. (2015). Performance prediction model for network security risk management. Communications on Applied
Kavitha, K., & Ranjitha, K. (2013). Particle Swarm Optimization For Adaptive Anomaly-Based Intrusion Detection System Using Fuzzy Controller. International Journal of Computer Trends and Technology (IJCTT), 4(10), 3536–3541. https://www.ijcttjournal.org/archives/1220-ijctt-v4i10p130

Khan, K., Nikov, A., & Sahai, A. (2011). A fuzzy bat clustering method for ergonomic screening of office workplaces. Advances in Intelligent and Soft Computing, 101, 59–66

Kamarasamy, G., & Wahi, A. (2012). An optimized k-means clustering technique using bat algorithm. European Journal of Scientific Research, 84 (2), 263–273. http://www.europeanjournalscientificsearch.com

Kotsiantis, S., Tsekovras, G., & Pintelas, P. (2005). Bagging model trees for classification problems. In P. Bozanis & E. N. Houstis (Eds.), Proceedings of advances in Informatics, 10th Panhellenic Conference on Informatics (pp. 328–337). PCI 2005, LNCS 3746.

Kumar, K. P. M. (2013). Intrusion Detection System for Malicious Traffic by using PSO-GA Algorithm. International Journal of Computer Science & Engineering Technology, 3(6), 236–238. https://ijceet.net/docs/Volumes/volume3issue6/IJCEET2013030606.pdf

Kumutha, V., & Palanimammal, S. (2014). Improved fuzzy clustering method based on intuitionistic fuzzy particle swarm optimization. Journal of Theoretical and Applied Information Technology, 62(1), 8–15. http://www.jatit.org/volumes/V8i2014/1091.pdf

Li, Z., Li, Y., & Xu, L. (2011). Anomaly intrusion detection method based on K-means clustering algorithm with particle swarm optimization. In Proceedings of IEEE International Conference of Information Technology, Computer Engineering and Management Sciences . IEEE Computer Society, pp. 157–161.

Liao, H. J., Lin, C. H. R., Lin, C. Y., & Tung, K. Y. (2013). Intrusion Detection System: A Comprehensive Review. Journal of Network and Computer Applications, Elsevier, 36(1), 16–24. https://doi.org/10.1016/j.jnca.2012.09.004

Maurer, T. (2011). Cyber Norm Emergence at the United Nations: An Analysis of the UN’s Activities Regarding Cyber-security. In Explorations in Cyber International Relations Discussion Paper Series,Discussion Paper 2011–11. Harvard’s Belfer Center for Science and International Affairs.(pp. 02138).

Nejad, N. K., Jabbehdari, S., & Moattar, M. H. (2015). A Hybrid Intrusion Detection System Using Particle Swarm Optimization For Feature Selection. International Journal of Computer Science and Technology Network, Web & Security, 13(4), 19–22. https://globaljournals.org/GJCS_Volume13/Enhancing-Network-Security.pdf

Niu, Q., & Huang, X. (2011). An improved fuzzy c-means clustering algorithm based on PSO. Journal of Software, 6(5), 873–879. https://doi.org/10.4304/jsw.6.5.873-879

Otor, S. U., Akinjemi, B. O., Aladesami, T. A., Aderounmu, G. A., & Kamogaté, B. H. Saurabh Pratap (Reviewing editor). (2019). An adaptive bio-inspired optimisation model based on the foraging behaviour of a social spider. Cogent Engineering, 6(1), 1588681. https://doi.org/10.1080/23311916.2019.1588681

Scarfone, K., & Mell, P. (2012). Guide to Intrusion Detection and Prevention Systems (IDPS) (Draft): Recommendations of the National Institute of
Standards and Technology NIST Special Publication. 800–894, Revision 1 (Draft). https://www.hqAscii.org/?view&id=718849

Sharma, N., & Gaur, B. (2016). An Approach for efficient intrusion detection based on R-ACO. International Journal of Advanced Technology and Engineering Exploration, 3(20), 98–104. https://doi.org/10.19101/IJATEE.2016.320005

Shukla, U. P., & Nanda, S. J. (2016). Parallel social spider clustering algorithm for high dimensional datasets. Engineering Applications of Artificial Intelligence. Elsevier, 56(2016) 75–90. https://doi.org/10.1016/j.enganappai.2016.08.013

Sreelaja, N. K., & Vijayalakshmi, P. G. A. (2010). Ant Colony Optimization based approach for efficient packet filtering in firewall. Applied Soft Computing, 10(4), 1222–1236. https://doi.org/10.1016/j.asoc.2010.03.009

Sreelaja, N. K., & Vijayalakshmi, P. G. A. (2014). Swarm intelligence based approach for sinkhole attack detection in wireless sensor networks. Applied Soft Computing, 19(2014), 68–79. https://doi.org/10.1016/j.asoc.2014.01.015

Sujitha, B. B., & Kavitha, V. (2015). Layered approach for intrusion detection using multiobjective particle swarm optimization. International Journal of Applied Engineering Research Research 10(12), 31999–32014.

Tsai, C. Y., & Chen, C. J. (2015). A PSO-AB classifier for solving sequence classification problems. Applied Soft Computing. Elsevier, 27(2015), 11–27. https://doi.org/10.1016/j.asoc.2014.10.029

Wan, Y., Wang, M., Ye, Z., & Lai, X. (2016). A feature selection method based on modified binary coded ant colony optimization algorithm. Applied Soft Computing, 49(2016), 248-258.https://doi.org/10.1016/j.asoc.2016.08.011

WaterISAC. (2015). Best practice to reduce exploitable weaknesses and attacks: 10 Basic cybersecurity measures. https://ics-cert.us-cert.gov/sites/default/files/documents/10_Basic_Cybersecurity_Measures-WaterISAC_June2015_S508C.pdf

Xingzhu, W. (2015). ACO and SVM selection feature weighting of network intrusion detection method. International Journal of Security and Its Applications, 9(4), 129–270. https://doi.org/10.14257/ijsia.2015.9.4.24

Yang, X. (2014). A framework for self-tuning algorithms. In Nature-inspired optimization algorithms -Chapter 12. Elsevier. pp. 175–182. https://doi.org/10.1016/B978-0-12-416743-8.00012-9.

Zheng, H. (2011). An efficient hybrid clustering-PSO algorithm for anomaly intrusion detection. Journal Of Software. Academy Publisher, 6(12), 2350–2360. https://doi.org/10.4304/jsw.6.12.2350-2360

Zhou, Y., Zhou, Y., Luo, Q., & Abdel-Basset, M. (2017). A simplex method-based social spider optimization algorithm for clustering-analysis. Engineering Applications of Artificial Intelligence. Elsevier, 64 (2017), 67–82. https://doi.org/10.1016/j.engappai.2017.06.004

Zhu, H., & Li, X. (2016). Research on a new method based on improved ACO algorithm and SVM model for data classification. International Journal of Database Theory and Application, 9(1), 217–226. https://doi.org/10.14257/ijdta.2016.9.1.19
