Swarm Intelligence Based Feature Selection for Intrusion and Detection System in Cloud Infrastructure

Vaishali Ravindranath
School of Computer Science & Engg.
VIT University
Vellore, India
rvaishali4@gmail.com

Sasikala Ramasamy
School of Computer Science & Engg.
VIT University
Vellore, India
sasikala.ra@vit.ac.in

Ramasubbareddy Somula
Department of IT
VNR VJIET
Hyderabad, India
svramasubbareddy1219@gmail.com

Kshira Sagar Sahoo
Department of Information Technology
VNR VJIET
Hyderabad, India
kshirasagar12@gmail.com

Amir H. Gandomi
Faculty of Engineering & Information Technology
University of Technology Sydney
Sydney, Australia
gandomi@uts.edu.au

***Abstract*—Network intrusion and cyber attacks are the most severe concern for Cloud computing service providers. The vulnerability of attacks is on a hike that manual or simple rule-based detection of cyber-attacks is not robust. In order to tackle cyber attacks in a reliable manner, an automated Intrusion Detection system equipped with a swarm intelligence (SI) based machine learning model (ML) is essential to deploy at entry points of the network. Nowadays, the application of SI with ML is used in various research areas. For an efficient IDS, choosing relevant features from the noisy data is an open question. In this regard, this paper proposes a method that utilizes the Whale Pearson hybrid feature selection wrapper for reducing the irrelevancy in the IDS model. Whale Pearson hybrid wrapper is an improved version of the binary Whale Optimization Algorithm (WOA). The WOA is a type of SI algorithm which is inspired by the behavior of humpback whales. The proposed method has chosen 8 out of 42 features from the Hackereath Network attack prediction data-set, which are sufficient for building an efficient Intrusion detection model. The model trained with the eight features produces an accuracy of 80%, which is 8% greater than the accuracy produced by the original data-set with the KNN algorithm on ten-fold cross-validation.**

***Index Terms*—Intrusion Detection, Cloud Computing, Machine Learning, Feature Selection, Whale Pearson wrapper**

I. Introduction

Cloud computing is a widely deployed technology with fast-growing traction towards all application domains. The metered usage facility enabled through reliable networking capacity has broadened the scope of cloud computing. Along with hiking scope in this domain, the vulnerability to be prone to security attacks increase day by day. Insecure hypervisors, virtual machines, and attacked nodes are major areas of concern. Nowadays, network autonomy with SDN (Software Defined Networks) gained more focus in cloud computing topology [33]. The SDN is prone to security attacks such as Distributed Denial of service, stealthy attack [2], [23] etc. This is a typical threat to the signature features of cloud computing environments like Infrastructure as a service. It is a highly difficult task for any SDN to capture the attack signatures in a real-time networking environment with a simple set of rules [28]. Cloud service providers find predictive model-based analysis as a solution to detect security threats in the network [4], [6]. The concept of intrusion and attack detection is not new in the field of cloud computing. The sample IDS cloud framework has illustrated in Figure 1. Every network holds its past record of attacks, types, vulnerability, and counter mechanism. These historical attacks data becomes a great resource for training the network to stay cognizant of future network attacks. In

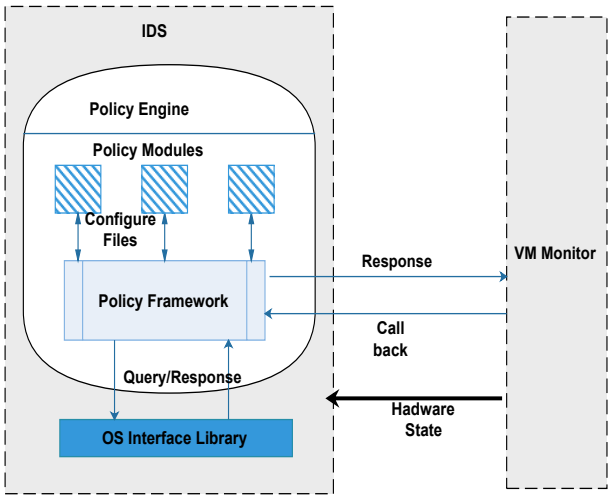

Fig. 1. Representation of intrusion detection systems in cloud computing environment

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

this sector of prediction, machine learning models are widely employed. A machine learning model has the capability to learn from the existing patterns and predict future possibilities explicitly. This capability of machine learning models is exploited to train an automated intrusion detection system with past records of network attacks and protect the network from future possibilities. A survey on Intrusion Detection and Prevention System (IDPS) [7] in cloud computing reveals that the open structure of the cloud infrastructure increases the vulnerability towards network attacks. This work has reviewed fuzzy intrusion detection mechanisms applicable to autonomous security systems deployed in cloud infrastructure. Fuzzy rules are primitive patterns that may not support accurate predictions in all cases. Practical difficulties faced during IDPS deployment in open cloud architectures were briefly discussed in [8]. In [9], authors reviewed security issues in cloud computing from the aspect of reliability, confidentiality, and integrity. This paper recommended and emphasized the necessity of an IDPS model in cloud computing architecture. An IDPS model built with simple rules may not constitute an efficient IDPS due to the lack of robustness. In order to build a robust IDPS, a well-trained machine learning model need to be built on the entry end of the cloud infrastructure. In [10], a semi-supervised IDPS system is proposed with a fuzzy single-layered feed-forward neural network. This method had shown a better classifier rate than Naive Bayes, Random forest, and support vector machine on NSL-KDD data-set. In a dynamic cloud environment, the usage of neural-works is not favorable because of its time complexity incurred for model training and updation. The reason for model complexity is an increase in the dimension space of the past records fed to the model. Dimension refers to the attributes and instances of a data-set. The rise in the number of instances can be tackled by filtering or clustering the most relevant features based on certain objectives and constraints. The increase in attribute space is often addressed as a NP-hard problem in dimension space. Let us imagine a data-set of past records with n feature attributes and one class attribute. Not all feature attributes in the data-set will be relevant to the prediction class. Among $(2^n) - 1$ R possible feature subsets, any one feature subset could be the optimum subset. To find a relevant feature subset, feature selection approaches are widely recommended by researchers in real-time scenarios. Feature Selection is classified into Filters, Wrappers, and Embedded based on their objective functions. An objective function that uses statistical factor analysis is known as the filter approach. This approach is primitive and less complex in terms of search and time. A hybrid feature selection based anomaly detection method has proposed in [11] to perform intrusion detection on NSL-KDD dataset. This work had utilized voting and information again as feature selection factors and trained the model with 20% training samples. The obtained results supported the claim that feature selection will improve classification model accuracy. A weighted feature selection algorithm was proposed for WIFI impersonation detection in [12]. Similarly, many intrusion detection algorithms have used filter-based feature

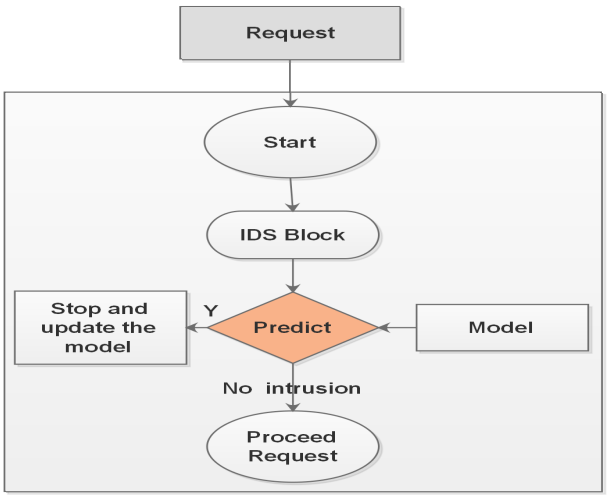

Fig. 2. A machine learning based Automated Intrusion detection system

selection algorithms on feature selection with an exhaustive search strategy. Exhaustive search algorithms test all possible solutions in the universe and provide the best out of all. This mechanism provides the most accurate solution, but it is not suitable for addressing NP-hard problem where the dimension of solution space grows exponentially. In order to overcome the NP-hardness, stochastic processes are introduced. Most of the famous stochastic process is gradient descent (local search), whereas swarm intelligence search algorithms believe in local as well as global search are most suitable for this optimization problem.

Applications of Swarm Intelligence (SI) algorithms for solving various optimization problems are numerous. The specialty of SI algorithms is that it works in a decentralized and self-organized manner. All SI systems typically consist of a population of agents. These agents often interact with themself and their environment. Generally, the inspirations are abstracted from the biological system. Few popular SI algorithms are PSO, ACO, FFA, Gray Wolf, Cuckoo's search etc. [1], [3], [5]. Similarly, the Whale Optimization Algorithm (WOA) is a type of SI which is inspired by the behavior of humpback whales.

In this regard, Fuzzy Genetic algorithm-based feature selection algorithms are popular and applied in [13]–[15], [29] . Genetic algorithms are good examples of global search and stochastic solution exploitation. In comparison with swarm intelligence search algorithms, the solution exploration power of Genetic algorithms is lesser. Hence nowadays, where iteration complexity of an algorithm during the search process is not a constraint, swarm intelligence methods are used. [16]–[22] are some popular feature selection approaches based on swarm intelligence optimizers and single objective fuzzy wrapper evaluation function embedded with k-NN algorithm [24]. This article had briefly discussed the role of soft computing based feature selection algorithms such as Genetic algorithm, Particle Swarm Optimization, and many other bio-inspired methods in IDPS. Among soft computing techniques, swarm intelligence

techniques are identified as a powerful method to identify optimum feature subsets in real-time. Even though Swarm intelligence search techniques are notably appreciated they are blind search techniques. The complexity of the objective function decides the accuracy of selection. Multi-objective optimization functions encounter the Pareto optimality problem. Hence, [25] proposed a single objective fuzzy weighted objective function for selection. Most of these algorithms still lack search optimization and correlation bias optimization. Instead, the algorithm designers have tuned the search path with a hybrid mechanism on the Whale optimization algorithm with simulated annealing [26]. Simulated annealing will increase the complexity of the search, and efficient alternate needs to be designed to improve the exploration process.
Based on the above literature discussion, this paper tries to test thee queries:

1) Is it possible for a feature subset to retain the structure of entire dataset during Intrusion Detection?
2) Does wrapper based correlation tuning on subsets during feature selection improve accuracy of prediction?
3) Does proposed algorithm (Binary Whale Pearson Hybrid Wrapper) perform better than existing Binary Whale simulated annealing hybrid wrapper on intrusion detection [26]?

II. Binary Whale Pearson hybrid Wrapper

Whale Pearson hybrid wrapper is an improved version of the Binary Whale swarm algorithm with simulated annealing as position updation function [27]. Whale optimization algorithm designed with inspiration from whale food search movements. The proposed algorithm follows the steps of its predecessor in the search process. The position updation factor is replaced with a novel correlation based selection algorithm design. As this position updation function follows both correlation and classifier guided fitness function, it will fall under the category of embedded selection methods. The working of the proposed correlation design is discussed below.

Let X be the local optimum solution resulted from Binary Whale wrapper at the end of an iteration. The position updation function takes X and the maximum number of iterations I as input. The function generates I random solutions. All these solutions undergo a correlation bias test with Pearson's correlation method. Let, f be the class attribute either binary or multi-class, and x_i be the feature attributes where i ranges from 0 to T (total number of features in a data-set). The mean correlation between the features and the class attribute is calculated with Equation 1 and Equation 2.

$$mcor = \sum cor(X_i, f) \tag{1}$$

$$cor(x,y) = \frac{\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \overline{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \overline{y})^2}} \tag{2}$$

In equation 2, x stands for the input and y denotes the class attribute containing the output of classification.

Using Mutation function on the current $Gbest$ solution, create I random position vectors. Each vector is evaluated with the novel correlation based objective function described in Equation 3. The best of the obtained solution is considered as the current position of the whale and the search for food quest continues until the maximum number of iterations is met.

$$obj(input) = cor(input, class) \tag{3}$$

After calculation of the obj($Gbest$) and obj(current position), compare the values, the one with maximum correlation value will be initialized as $Gbest$ of the next iteration of Whale Swarm Search Algorithm. This position updation function is used as a substitute for the existing Simulated Annealing module in [27]. The pseudo-code of the updated version of Binary Whale swarm wrapper with Pearson's correlation as objective is given below.

Algorithm 1: Whale Pearson Feature Selection wrapper

```
Initialize: lb = 0; ub = 1; //upper and Lower Boundaries
Initialize: whales,itermax
Initialize: whale_position,food_position
Initialize: whale_fitness, food_fitness
Initialize: i=1 //initial iteration
while i ≤ itermax do
    Calculate fitness of each whale with objective function
    F= Best_Whale
    X=Position of the Best_Whale
    Update whale positions with steering function
    foreach whale(x_i) do
        if obj(whale_position) < obj(food_position) then
            food_position=whale_position
        else
            continue;
        end
    end
    Return food_position
    mutate(food_position,max_iter)
    if (cor(new_position, class) > cor(food_position, class)) ∧
     (fitness(new_position) < fitness(food_position)) then
        food_position=new_position;
    end
end
```

III. Experiment

The experimental setup for the intrusion detection with SI-ML is as follows.

A. System Setup.

Operating System : Windows 10 64 bit
Hardware : Intel i5 5th Gen with 12 Gb RAM
Software : MATLAB 2017A

IV. Dataset

The data-set taken for analysis is obtained from Hackerearth machine learning competitions repository. The theme of the contest is to predict the network attacks with provided training samples of attacks. According to Hackerearth, a networking based company in Japan faced loss due to the intervention of cyber-attacks resulted in a security breach, non-reliable communication, and transfer speed reduction. On conducting an analysis of the TCP packets, the company identified that out of 40 types of cyber attacks on the network, three types of attacks are more malicious. The company has collected the instances of these three different attacks in the network and created a training set for prediction. This training set is supplied to the automated intrusion detection model for the classification of future attacks in the network. An exploratory analysis of the data-set will provide us a clear picture of the internal structures.

Instances in the dataset : 169307
Instances chosen for Experiment : 1:3000
Features in the dataset : 42
Class attribute in the dataset : 1(Target)
Class type : Multiple
Class labels : 3(0,1,2 denoting 3 different attacks)

Among 42 features, there are 18 numeric features, 23 categorical features, and one connection identifier feature. As the sources don't mention the presence of noise in the data-set, it is assumed that the data-set is void of noises such as missing instances and outliers.

V. Methodology

The methodology of the experiment is detailed in the Figure 3.

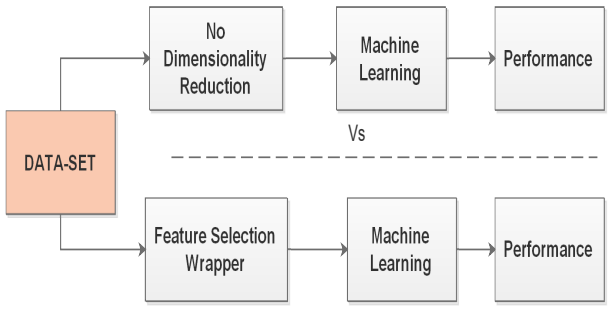

Fig. 3. The conceptual framework of the proposed methodology

VI. Results and Discussions

The experiment is carried out according to the methodology discussed in the previous section. The metrics used in evaluation of the algorithm are feature reduction ratio Error Rate and CPU time. Parameter settings for the algorithms are tabulated in the Table I.

TABLE I
Parameter settings for the algorithms

Algorithm	Parameter	Value
5*Binary Whale Pearson algorithm	Number of whales	30
	Lower bound	0
	Upper bound	1
	Maximum number of iterations	10
	Dimensions	No.of Attributes
3*Objective function-kNN	K	5
	Distance	Euclidean
	Validation	10-fold
4*Pearson objective	Input data	Current food position
	Generation	Mutation
	Itertation	max_iter
	Evaluation	Correlation

$$\text{Feature Reduction Ration} = \frac{n}{N} \tag{4}$$

$$\text{Error Rate of the classifiers } (\%) = 1 - \left(\frac{accuracy}{100}\right) \tag{5}$$

$$\text{CPU Time taken} = \frac{\text{Total Execution Time}}{max_{iter}} \tag{6}$$

Based on the evaluation metrics (4, 5 and 6), a comparative tabulation of results obtained with Whale Simulated annealing algorithm and proposed whale Pearson algorithm is given in Table II.

From the results obtained in Table II, it is observed that there is a significant improvement in the prediction accuracy of machine learning model from 0.72 to 0.8 after feature selection. The obtained result in Figure 5 proves that, for a feature subset selected with wrappers has capacity to improve the accuracy and retain the structure of entire data-set.

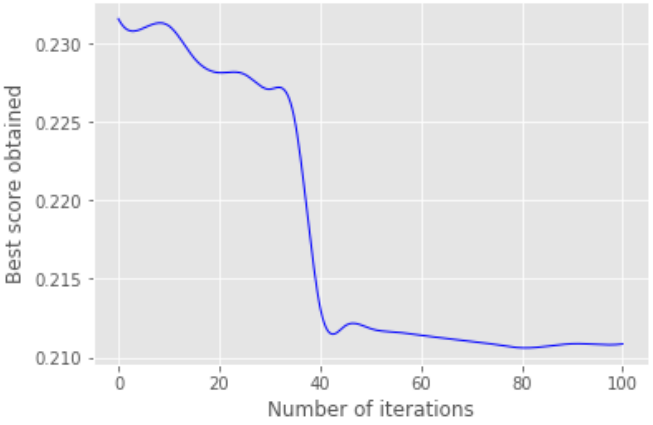

Fig. 4. Convergence of fitness function in 100 iterations of Whale Pearson feature selection wrapper

The feature reduction ratio necessarily has lesser value in the best case. Among 4 columns, Whale Simulated annealing algorithm with 100 iterations has a feature reduction ratio of

TABLE II
RESULTS OF THE BINARY WHALE SWARM WRAPPER BASED FEATURE SELECTION ALGORITHMS

Metrics	**Original**	**Whale Pearson (Proposed) 100 iterations**	**Whale Simulated Annealing 100 iterations**	**Whale Simulated annealing 10 iterations**
Number of Features	42	8	2	12
Feature Reduction Ratio	1	0.19048	0.09524	0.2857
Error Rate	0.28	0.2	0.20	0.198
accuracy	0.72	0.8	0.8	0.802
Fitness	NA	0.20655	0.18991	0.19895
CPU Time Taken (sec/iterations)	NA	1.96	10.7787	1.11631459

0.095 which is lesser than Binary Whale Pearson (0.19048) and Binary Whale Simulated annealing (BWSA) with 10 iterations (0.2857). There is a comparison between BWSA of 100 iterations and BWSA of 10 iterations because BWSA is a complex hybrid algorithm that combines whale search (I iterations) and Simulated Annealing of (I iterations). So each iteration of BWSA runs $i + I$ times, where i= (0 to I). The CPU time is a proof for the complexity of BWSA. 100 iterations of proposed Whale Pearson wrappers takes 1.96 seconds per iteration whereas 100 iterations of BWSA takes 10.7787 seconds/iteration and just 1.1166 seconds/ iteration for $I = 10$. This proves the above claim that each run of BWSA accounts to $i + I$ iterations.

In a minimization problem, objective function should return minimum value of fitness, in the best case it must be more closer to zero. In the Table II, the fitness range of proposed algorithm is close to 0.20. Figure 4 shows the convergence rate of the proposed Whale Pearson wrapper. It does not exhibit fast convergence, however the other algorithms also obtained fitness close to the proposed method.

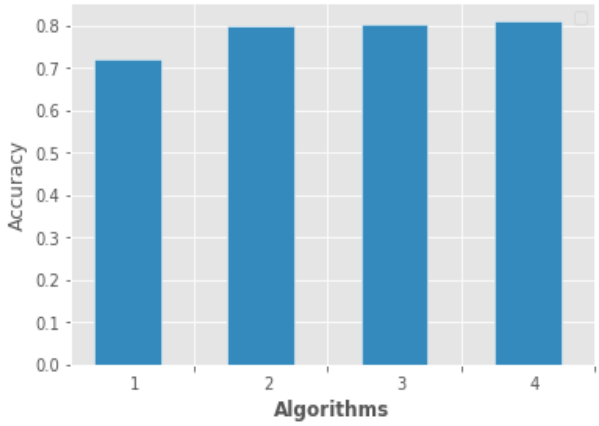

Fig. 5. Accuracy of Intrusion detection with kNN after feature selection 1. Original data 2. Whale Pearson of 100 iterations 3. BWSA of 100 iterations 4. BWSA of 10 iterations

The third hypothesis is to find whether the proposed algorithm is better than the existing BWSA wrapper. In terms of Feature Reduction Ratio, accuracy, Error Rate, and fitness, the existing BWSA wrapper with 100 iterations is a better performer than the proposed method. As discussed above, the exploration capacity of BWSA increases in the rate of $i + I$. Hence, in comparison with BWSA of 10 iterations, the proposed algorithm obtained a better feature reduction ratio than the existing wrapper. The accuracy and fitness of both algorithms are almost closed. Also, on correlation analysis, the subsets obtained with the proposed method are more relevant in all cases than subsets returned by the existing BWSA algorithm. This is because of the special attention given to correlation bias during position updation. Thus in terms of Feature reduction ratio and correlation bias, the proposed method performs better than the existing BWSA wrapper in Intrusion detection.

VII. CONCLUSION

In this paper, we introduced a novel Feature selection wrapper combining Whale Swarm optimizer and Pearson's correlation method for network intrusion detection in cloud computing environments. Model efficiency does matters a lot in real time automated detection systems deployed at network end. In order to train the model with relevant features, a wrapper based feature selection approach was proposed. From the results, it was proven that feature selection wrappers would increase the relevancy levels in the data-set and improve the model prediction accuracy. Also, Whale Pearson, feature selection wrapper had produced decent results close to the existing BWSA with a comparatively less CPU time in 100 iterations. In the proposed SI method, the correlation bias function followed a random initialization method with mutation. This could be the probable reason for the primitive results obtained by the algorithm. In future work, the proposed algorithm should be accompanied by a guided selection procedure for efficiency improvement.

REFERENCES

[1] Yang, Xin-She, et al., eds. Swarm intelligence and bio-inspired computation: theory and applications. Newnes, 2013.

[2] Mauro Conti, Fabio De Gaspari, and Luigi Vincenzo Mancini. A novel stealthy attack to gather sdn configuration-information. *IEEE Transactions on Emerging Topics in Computing*, (1):1–1, 2018.

[3] Yang, Xin-She, and Amir Hossein Gandomi. "Bat algorithm: a novel approach for global engineering optimization." Engineering Computations 29.5 (2012): 464-483.

[4] Salman Iqbal, Miss Laiha Mat Kiah, Babak Dhaghighi, Muzammil Hussain, Suleman Khan, Muhammad Khurram Khan, and Kim-Kwang Raymond Choo. On cloud security attacks: A taxonomy and intrusion detection and prevention as a service. *Journal of Network and Computer Applications*, 74:98–120, 2016.

[5] Gandomi, Amir Hossein, Xin-She Yang, and Amir Hossein Alavi. "Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems." Engineering with computers 29.1 (2013): 17-35.

[6] Ahmed AlEroud and Izzat Alsmadi. Identifying cyber-attacks on software defined networks: An inference-based intrusion detection approach. *Journal of Network and Computer Applications*, 80:152–164, 2017.

[7] Ahmed Patel, Mona Taghavi, Kaveh Bakhtiyari, and Joaquim Celestino JúNior. An intrusion detection and prevention system in cloud computing: A systematic review. *Journal of network and computer applications*, 36(1):25–41, 2013.

[8] Ram Shankar Siva Kumar, Andrew Wicker, and Matt Swann. Practical machine learning for cloud intrusion detection: challenges and the way forward. In *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, pages 81–90. ACM, 2017.

[9] Chirag Modi, Dhiren Patel, Bhavesh Borisaniya, Hiren Patel, Avi Patel, and Muttukrishnan Rajarajan. A survey of intrusion detection techniques in cloud. *Journal of network and computer applications*, 36(1):42–57, 2013.

[10] Rana Aamir Raza Ashfaq, Xi-Zhao Wang, Joshua Zhexue Huang, Haider Abbas, and Yu-Lin He. Fuzziness based semi-supervised learning approach for intrusion detection system. *Information Sciences*, 378:484–497, 2017.

[11] Shadi Aljawarneh, Monther Aldwairi, and Muneer Bani Yassein. Anomaly-based intrusion detection system through feature selection analysis and building hybrid efficient model. *Journal of Computational Science*, 25:152–160, 2018.

[12] Muhamad Erza Aminanto, Rakyong Choi, Harry Chandra Tanuwidjaja, Paul D Yoo, and Kwangjo Kim. Deep abstraction and weighted feature selection for wi-fi impersonation detection. *IEEE Transactions on Information Forensics and Security*, 13(3):621–636, 2018.

[13] Chi-Ho Tsang, Sam Kwong, and Hanli Wang. Genetic-fuzzy rule mining approach and evaluation of feature selection techniques for anomaly intrusion detection. *Pattern Recognition*, 40(9):2373–2391, 2007.

[14] Susan M Bridges, Rayford B Vaughn, et al. Fuzzy data mining and genetic algorithms applied to intrusion detection. In *Proceedings of 12th Annual Canadian Information Technology Security Symposium*, pages 109–122, 2000.

[15] Yingying Zhu, Junwei Liang, Jianyong Chen, and Zhong Ming. An improved nsga-iii algorithm for feature selection used in intrusion detection. *Knowledge-Based Systems*, 116:74–85, 2017.

[16] Hossam M Zawbaa, E Emary, and Crina Grosan. Feature selection via chaotic antlion optimization. *PloS one*, 11(3):e0150652, 2016.

[17] Xin-She Yang. Firefly algorithm, stochastic test functions and design optimisation. *International Journal of Bio-Inspired Computation*, 2(2):78–84, 2010.

[18] Seyedali Mirjalili, Amir H Gandomi, Seyedeh Zahra Mirjalili, Shahrzad Saremi, Hossam Faris, and Seyed Mohammad Mirjalili. Salp swarm algorithm: A bio-inspired optimizer for engineering design problems. *Advances in Engineering Software*, 114:163–191, 2017.

[19] Eid Emary, Hossam M Zawbaa, and Aboul Ella Hassanien. Binary grey wolf optimization approaches for feature selection. *Neurocomputing*, 172:371–381, 2016.

[20] Majdi Mafarja and Seyedali Mirjalili. Whale optimization approaches for wrapper feature selection. *Applied Soft Computing*, 62:441–453, 2018.

[21] Hamidreza Rashidy Kanan and Karim Faez. An improved feature selection method based on ant colony optimization (aco) evaluated on face recognition system. *Applied Mathematics and Computation*, 205(2):716–725, 2008.

[22] Yuanning Liu, Gang Wang, Huiling Chen, Hao Dong, Xiaodong Zhu, and Sujing Wang. An improved particle swarm optimization for feature selection. *Journal of Bionic Engineering*, 8(2):191–200, 2011.

[23] Sahoo, Kshira Sagar, et al. "Toward secure software-defined networks against distributed denial of service attack." The Journal of Supercomputing 75.8 (2019): 4829-4874.

[24] P Ravi Kiran Varma, V Valli Kumari, and S Srinivas Kumar. A survey of feature selection techniques in intrusion detection system: A soft computing perspective. In *Progress in Computing, Analytics and Networking*, pages 785–793. Springer, 2018.

[25] Eid Emary, Hossam M Zawbaa, and Aboul Ella Hassanien. Binary ant lion approaches for feature selection. *Neurocomputing*, 213:54–65, 2016.

[26] Majdi M Mafarja and Seyedali Mirjalili. Hybrid whale optimization algorithm with simulated annealing for feature selection. *Neurocomputing*, 260:302–312, 2017.

[27] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95:51–67, 2016.

[28] Sahoo, Kshira Sagar, et al. "ESMLB: Efficient Switch Migration-based Load Balancing for Multi-Controller SDN in IoT." IEEE Internet of Things Journal (2019).

[29] Vaishali, R., Sasikala, R., Ramasubbareddy, S., Remya, S., & Nalluri, S. (2017, October). Genetic algorithm based feature selection and MOE Fuzzy classification algorithm on Pima Indians Diabetes dataset. In 2017 International Conference on Computing Networking and Informatics (ICCNI) (pp. 1-5). IEEE.

[30] Basu, S., Kannayaram, G., Ramasubbareddy, S., & Venkatasubbaiah, C. (2019). Improved Genetic Algorithm for Monitoring of Virtual Machines in Cloud Environment. In Smart Intelligent Computing and Applications (pp. 319-326). Springer, Singapore.

[31] Somula, R., Anilkumar, C., Venkatesh, B., Karrothu, A., Kumar, C. P., & Sasikala, R. (2019). Cloudlet services for healthcare applications in mobile cloud computing. In Proceedings of the 2nd International Conference on Data Engineering and Communication Technology (pp. 535-543). Springer, Singapore.

[32] Ramasubbareddy, S., Vedavasu, G., Krishna, G., & Savithri, A. (2019). PIOCM: Properly Identifying Optimized Cloudlet in Mobile Cloud Computing. Journal of Computational and Theoretical Nanoscience, 16(5-6), 1967-1971.

[33] Sahoo, K. S., Tiwary, M., Mishra, P., Reddy, S. R. S., Balusamy, B., & Gandomi, A. H. (2019). Improving End-Users Utility in Software-Defined Wide Area Network Systems. IEEE Transactions on Network and Service Management.