2020 International Conference on Data Analytics for Business and Industry: Way Towards a Sustainable Economy (ICDABI) | 978-1-7281-9675-6/20/$31.00 ©2020 IEEE | DOI: 10.1109/ICDABI51230.2020.9325669

Hybrid Intrusion Detection System Based on Deep Learning

Ali Azawii Abdul Lateef
College of Computer Science and IT
University of Anbar
Ramadi, Iraq
aliazawii@uoanbar.edu.iq

Sufyan T. Faraj Al-Janabi
College of Computer Science and IT
University of Anbar
Ramadi, Iraq
orcid.org/0000-0002-2805-5738

Belal Al-Khateeb
College of Computer Science and IT
University of Anbar
Ramadi, Iraq
belal-alkhateeb@uoanbar.edu.iq

***Abstract*— Every day, there are new types of cyber-attacks faced by systems and networks of official and non-official organizations, e-commerce, and even people around the world. Since Deep Learning (DL) can derive better representations from the data and construct better models, this work proposes an Intrusion Detection System (IDS) based on DL techniques by using the Recurrent Neural Network (RNN) algorithm. Hence, this paper presents the design and implementation of the binary class IDS based on RNNs. The Crow Swarm Optimization (CSO) algorithm has been used to reduce the dataset features. This is necessary as reducing the features means dealing with less data, which reflects positively on the system's accuracy and the implementation time. Using the KDD 99 dataset for benchmarking, the experimental results have illustrated that RNN is very suitable for solving the intrusion detection problem in binary classification methods. Indeed, the obtained results have shown the CSO algorithm's superiority for features selection and reduction, where it produced three selected features with an accuracy rate of (98.34%).**

Keywords—*intrusion detection systems, recurrent neural network, deep learning, deep neural network, crow swarm optimization, KDDCup 99*

I. Introduction

Intrusion Detection Systems (IDSs) have become a standard security measure in computer networks. An IDS is a computer or software program that checks for unauthorized behavior or protocol breaches on a network or system. Usually, any behavior or intrusion observed is reported either to an administrator or centrally gathered using a scheme of security information and event management (SIEM). A SIEM framework incorporates multi-source outputs and employs alert detection approaches to separate suspicious behavior from false alerts [1].

Deep learning (DL), which is also referred to as the Deep Neural Network or Deep Neural Learning, is a sub-set of Machine Learning (ML) in the area of Artificial Intelligence (AI). DL deals with networks that can learn in both supervised and unsupervised manners from labeled and unlabeled data. DL is an AI function that simulates the working of the human brain in how it processes data and creates patterns for use in decision-making [2].

Complex, large, and high-dimensional data is generated by the vast and rapid proliferation of computing devices using Wi-Fi networks, which is confounding when collecting attack assets. Feature learning serves as an important tool for optimizing a DL model's learning process. It consists of the building, extraction, and selection functions. Feature design extends the original features to boost their expressiveness, while feature extraction turns the original features into a different shape and unwanted features are replaced by feature selection. Feature learning is a key aspect of improving the performance of existing machine-learning-based IDSs [3]. Therefore, deep abstractions are leveraged by building a DL model to improve and evaluate features selection tasks.

RNNs are Neural Networks (NNs) that utilize recurrence, and they use information from a preceding forward pass over the NN. All RNNs can be viewed as a relationship of recurrence. RNNs are suitable and have had considerable success when applied to issues in which the input data on which the predictions need to be done is in the form of a sequence (series of entities in which order is of importance) [4]. Figure 1 represents a general structure of RNN, where h_k denotes the input at time step k, while X_k denotes the output [5].

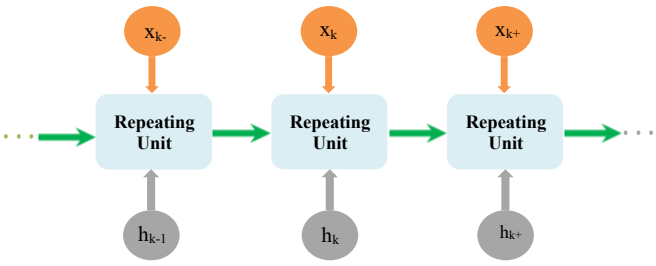

Fig. 1. A general structure of RNN [7].

The remaining of this paper is organized as follows: Section 2 presents a literature review for the most recent related work, while Section 3 is a simple explanation for the KDD-99 dataset which is used in the proposed system. Section 4 briefly explains the CSO algorithm, which is used in our work. Then, in sections 5 and 6, the proposed system design is presented and the results are explained and discussed. Finally, the paper is concluded in Section 7.

II. Related Work

In this section, a literature review is conducted on the most important related research that has used the RNN algorithm in intrusion detection in the last years to summarize the findings of this research.

J. Kim et al. applied RNN to IDS with the Hess free optimization. They utilized the DARPA dataset for the sake of training and testing their model of intrusion detection. They extended the architecture of Long Short-Term Memory (LSTM) to an RNN and trained their IDS using the dataset KDDCup-99. For the training, they created a dataset by comparing it with other IDS classifiers by extracting samples from the KDDCup-99 dataset [6]. Besides, a Gated Recurrent Unit Recurrent Neural Network (GRU-RNN) was proposed by Tuan Tang et al., which allowed IDSs for SDNs. The method was evaluated using the KDDCup-99 dataset and obtained an accuracy equal to 89 percent, with just six specific features [7].

Regarding features selection in IDS models, different stochastic algorithms have been applied for the hybrid-based feature selection and deep learning components. An IDS model that connects Particle Swarm Optimization (PSO) and Support Vector Machine (SVM) was proposed by J. Wang et al. Their twist consists of using Binary PSO for selecting features and Standard PSO for selecting SVM classifier input parameters [8]. Also, H. Li et al. proposed a hybrid feature selection model based on random forest and PSO, which uses the learning algorithm to evaluate feature subsets. They utilized the KDD1999 dataset to evaluate e proposed model. The proposed model's experimental evaluation selected six features was about 98.0 Accuracy Rate (AR) [9].

PSO, Ant Colony Optimization algorithm (ACO), Artificial Bee Colony algorithm (ABC), K-Nearest Neighbors (KNN) algorithm, and SVM were used by T. Khorram et al. to choose the most important feature set for the detection of network attacks. To test the efficiency of these feature selection algorithms, KNN and SVM algorithms were used as classifiers. This research used the standard NSL-KDD dataset for training and testing [10].

Hui Xu et al. introduced the PSO strategy to update and select the features mechanism. In order to achieve better dimensional reduction, the Improved Binary Whale Optimization Algorithm (IBWOA) was then proposed based on ensuring precision for intrusion detection feature selection. For testing of feature selection in intrusion detection, the KDD CUP 99 was used. The experimental findings showed that the IBWOA enhanced the accuracy of classification and reduction of dimensionality in features [11].

A filter and wrapper-based approach with a firefly algorithm was implemented in the wrapper by B. Selvakumar et al. to select the features. C4.5 and the Bayesian Networks (BN) based classifier with KDD CUP 99 dataset is subjected to the resulting features [12]. Moreover, S. Mohammadi et al. proposed an IDS based on feature selection and clustering algorithm using feature grouping based on the linear correlation coefficient (FGLCC) algorithm and Cuttlefish Algorithm (CFA), respectively. In their proposed system, the decision tree was used as the classifier. The proposed system was applied to KDD Cup 99 for performance verification [13].

III. The KDD'99 Dataset

The KDD'99 dataset has been used in this work to test the output of the subsets chosen by running the proposed method. The KDD'99 dataset is widely used for the evaluation of the performance of IDSs. It is made up of the test and train data sets.

The original KDD dataset consists of 4,898,431 data points. Each data point represents a session between two hosts in a network. Each vector has an attribute named 'label.' which denotes if it is normal or the type of attack if it is malicious. A total of 972781 data points were labeled as normal, and the rest 2952839 data points were labeled as a type of attack [14]. Table 1 shows the names of the features and their sequence in the dataset.

TABLE I. : The Sequence of KDD Dataset 41 Features [15], [16].

#	Features Names	#	Features Names
1	duration	23	count
2	protocol_type	24	srv_count
3	service	25	serror_rate
5	flag	26	srv_serror_rate
6	src_bytes	27	rerror_rate
7	dst_bytes	28	srv_rerror_rate
8	land	29	same_srv_rate
9	wrong_fragment	30	diff_srv_rate
10	urgent	31	srv_diff_host_rate
11	hot	32	dst_host_count
12	num_failed_logins	33	dst_host_srv_count
13	logged_in	34	dst_host_same_srv_rate
14	num_compromised	35	dst_host_diff_srv_rate
15	root_shell	36	dst_host_same_src_port_rate
16	su_attempted	37	dst_host_srv_diff_host_rate
17	num_root	38	dst_host_serror_rate
18	num_file_creations	39	dst_host_srv_serror_rate
19	num_shells	40	dst_host_rerror_rate
20	num_access_files	41	dst_host_srv_rerror_rate
21	num_outbound_cmds		

IV. Crow Swarm Optimization Algorithm (CSO)

The American Crow (CorvusBrachyrhynchos) is an example of a species that has evolved complicated social behaviors. They are crows, conjointly called the common crows. They live in northwestern cities. The crows are divided into teams and the typical cluster size of American Crows that outbound their last hunt site of the day is 237 ± forty-three. Daytime hunt aggregations of crows throughout the nonbreeding season area unit sometimes composed of various family teams. At night, crows reside in communal roosts, which will vary from 100 to 2 million crows. However, these hunt sites of area unit usually are not among the visibility of their roosting sites [17].

The following three predictions will facilitate supporting the required hypothesis [18]:

- American Crows can leave their communal search sites in giant teams, and singly in some critical situations.

- American Crows can depart their communal forage web sites when the sunset and fly along towards their communal roosting site to own the longest forage opportunities day by day.
- American Crows can fly within the direction of their communal roosting site once outward their last communal search site.

The CSO algorithm is presented, as shown in Algorithm 1 [18],[19].

ALGORITHM I. CROW SWARM OPTIMIZATION [18].

Input: $Population_{size.}$, $Group_{size}$
Output: ***BestSolution***
Start
Step 1: Initialization
$C_{speed} \leftarrow RandomSpeed()$, $C_{position} \leftarrow RandomPosition()$
$C_{\theta} \leftarrow RandomAngle()$, *Distribute C into G.*
Step 2: Evaluate (fitness)
Step 3: Select Best Solution.
Step 4: While (StopCondition())
$C_{speed} \leftarrow$ UpdateSpeed
$C_{position} \leftarrow$ UpdatePosition
Evaluate (fitness).
Select Best Solution.
Update fitness [If(NewFitness<CurrentFitiness)
CurrentFitness ← NewFitness]
$C_{\theta} \leftarrow$ Updateθ
Stop Condition
Step 5: Return Best Solution.
End

List of Acronyms:

- C_{speed} *Speed of Crows*
- $C_{position}$ *position of Crows*
- C_{θ} *Angle of Crows*
- C *Crows.*
- G *Groups.*

V. THE PROPOSED SYSTEM DESIGN

As presented in the literature review, no metaheuristic algorithm is perfect for features selection problems. This leaves the research open to define more algorithms to reach optimal solutions as much as possible. Therefore, we decided to apply the CSO algorithm as features selection algorithm to obtain the lowest number of features with the highest accuracy ratio.

In general, the system's overall structure consists of two main parts: The first is the implementation of a DL algorithm (RNN) to the KDD-99 dataset, which is an IDS. This leads us to obtain high accuracy of the results and choose the most appropriate architecture used in the second part of the proposed system.

The second part is implementing the CSO algorithm as features selection method at which the RNN algorithm (obtained from the first part) is used as a fitness evaluator. Indeed, the proposed fitness evaluator consisted of two parts, the first one is the accuracy obtained from RNN, while the second part is the number of the extracted features.

Some processes must be performed between the first and second parts before entering the core of the work, including the dataset preprocessing and dataset mapping. The proposed system architecture is shown in Figure 2.

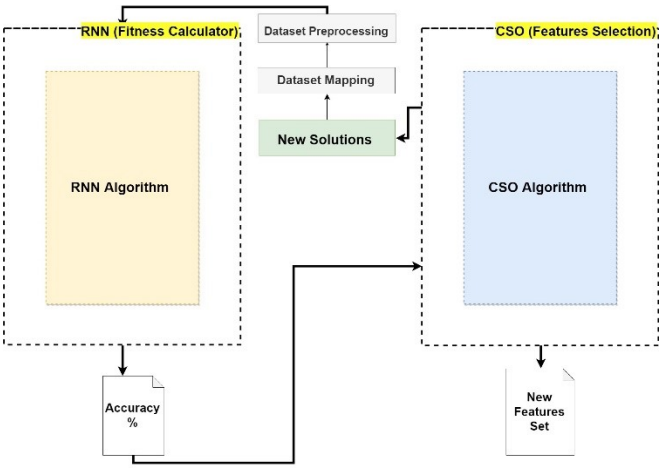

Fig. 2. The proposed system diagram.

The basic steps needed to implement the CSO algorithm in this work are the features selection algorithm, problem representation, fitness calculation process for the problem, and the stopping condition to obtain the final results. Figure 3 below generally describes the main CSO algorithm steps.

Fig. 3. The main steps of the CSO algorithm implementation.

VI. EXPERIMENTAL RESULTS AND DISCUSSION

The CSO algorithm has been applied to select as few features as possible to be used in our proposed system, where the integrated CSO algorithm with the RNN algorithm has been used to extract these features. Figure 4 and Table 2 show the results of experiments to obtain these features per each test.

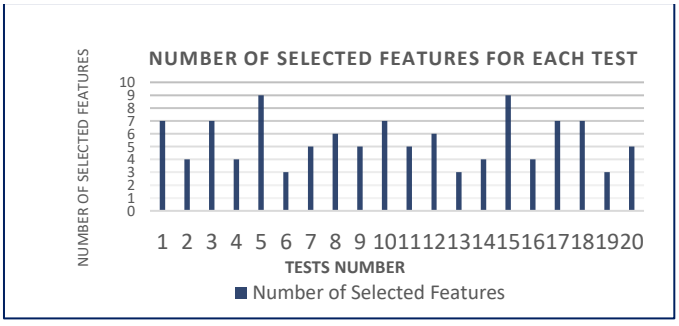

Fig. 4. The number of selected features for each test.

TABLE II. PERFORMANCE MEASURES OF CSO-RNN USING KDD-99 DATASET.

Test No.	20 Max Iterations and 10 Solutions			
	Selected Features	*Number of Selected Features*	*CSO Accuracy*	*Selected Features Accuracy Using RNN*
1	[9, 18, 22, 26, 32, 33, 34]	7	92.4920	98.73%
2	[24, 27, 32, 33]	4	95.4335	98.58%
3	[3, 18, 23, 25, 26, 31, 35]	7	92.4560	98.64 %
4	[4, 12, 22, 39]	4	95.2960	98.24 %
5	[1, 4, 10, 11, 14, 25, 34, 35, 37]	9	90.1825	98.62 %
6	[22, 32, 36]	3	96.2720	98.34%
7	[3, 7, 8, 18, 26]	5	94.4601	98.65%
8	[0, 5, 7, 15, 30, 40]	6	92.7259	96.81%
9	[2, 7, 27, 35, 38]	5	94.6800	99.20%
10	[11, 15, 23, 24, 28, 30, 32]	7	92.3920	98.48%
11	[1, 5, 7, 14, 16]	5	93.6120	96.53%
12	[4, 14, 21, 23, 24, 34]	6	93.3073	98.61%
13	[4, 5, 32]	3	96.2560	98.12 %
14	[5, 9, 29, 30]	4	94.3835	96.77%
15	[7, 9, 11, 18, 20, 25, 29, 33, 38]	9	88.8920	94.73
16	[0, 3, 11, 22]	4	95.3442	98.69%
17	[11, 15, 23, 24, 28, 30, 32]	7	92.3920	98.48%
18	[2, 13, 14, 19, 33, 37, 39]	7	92.4640	98.66%
19	[1, 3, 10]	3	96.0600	97.65%
20	[5, 6, 7, 20, 39]	5	94.5942	98.99%
1	[9, 18, 22, 26, 32, 33, 34]	7	92.4920	98.73%

As shown in the table and figure above, it is noted that the experiments have been repeated for 20 experiments. Each experiment produces a different number of features. These feature numbers are shown in Table 2. The obtained results show that the three features (22, 32, 36) with an accuracy of (98.34%) are the best available option for the proposed IDS. This is because the accuracy of the other features is slightly different. Therefore, using fewer features means using less data, which increases the classifying speed and reduces its time. Hence, it turns out that the use of a smaller number of features leads to work improvement of the proposed system in terms of time and speed of implementation while maintaining the high accuracy. Figure 5 shows the number of selected features by the CSO algorithm and its accuracy.

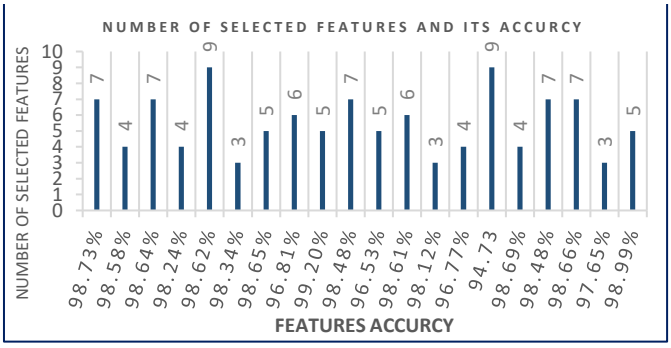

Fig. 5. Selected features and their accuracy.

As mentioned in the literature review presented at the beginning of the paper, some previous related works used different methods and algorithms to minimize the features. Thus, it is fruitful to compare our results with some earlier works. Table 3 shows some related references and their findings compared to this proposed work. More theoretical insights of this study can be found in [20].

TABLE III. COMPARISON WITH THE PREVIOUS RELATED WORKS.

	Literature Reference	Method Name	Number of Features	Accuracy
1	[21]	PSO + AUC	12	94.49
2	[22]	SSO + RS	6	93.60
3	[23]	CFA+ DT	10	92.05
4	[24]	ACO	8	98.90
5	[25]	ACO + SVM	14	98.00
6	[9]	PSO + RF	6	98.00
7	[10]	ABC + KNN	7	98.90
8	[11]	IBWOA	5	97.89
9	[12]	Firefly + BN	10	99.95
10	[13]	FGLCC + CFA + DT	10	95.03
11	Our work	CSO +RNN	3	98.34

In Table 3, the obtained results from the previous researches and the proposed system are presented. The choice should be made according to the system requirements. For example, if the system requires high accuracy regardless of the number of features, the selected method is the method that produces the highest accuracy. However, if the number of features is required, it is very logical to integrate CSO and RNN. The integration of the RNN algorithm with the CSO algorithm has led to new and very good results compared to the other methods' results. The features derived by the two algorithms are used in our proposed system with a good time and prediction accuracy.

VII. CONCLUSION

One of the most important conclusions of this work is that DL techniques such as the RNN algorithm are very suitable for dealing with binary classification of intrusion detection problems because of the high performance resulting from them when using RNN with KDD-99 dataset. Indeed, there is no fixed architecture for the RNN to deal with the problem of intrusion detection. Therefore, experiments have been done to choose the most appropriate architecture to deal with this problem.

Reducing the model layers increases the evaluation model's performance because it reduces the number of parameters used and execution time. Also, increasing the number of epochs does not significantly affect the accuracy of the model. As a feature selection, the CSO algorithm effectively deals with the intrusion detection problem, especially in dealing with the KDD-99 dataset. In this respect, the next step is to extend the work from the binary classification case (where an output of 'zero' means it is a normal behavior and an output of 'one' means it is an attack) to the multiclassification case. Another possible future work direction is to apply the CSO algorithm as a feature selection to solve the intrusion detection problem using some other intrusion detection datasets such as ADFA and ISCX IDS

2012 datasets. This can help generalizing the proposed approach.

REFERENCES

[1] Z. M. A. Osman, "An Intrusion Detection System Using Artificial Neural Network Based on Intrusion Behavior," M.Sc. Thesis, Sudan University of Science and Technology, 2018.

[2] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature* 521 (7553): 436," *Google Sch.*, 2015.

[3] H. Motoda and H. Liu, "Feature selection, extraction and construction," *Commun. IICM (Institute Inf. Comput. Mach. Taiwan) Vol*, vol. 5, no. 67–72, p. 2, 2002.

[4] S. Pouyanfar *et al.*, "A Survey on Deep Learning: Algorithms, Techniques, and Applications," *ACM Comput. Surv.*, vol. 51, no. 5, p. 92, 2018.

[5] A. Khan and F. Zhang, "Using recurrent neural networks (RNNs) as planners for bio-inspired robotic motion," in *Control Technology and Applications (CCTA), 2017 IEEE Conference on*, 2017, pp. 1025–1030.

[6] J. Kim and H. Kim, "Applying recurrent neural network to intrusion detection with hessian free optimization," in *International Workshop on Information Security Applications*, 2015, pp. 357–369.

[7] T. A. Tang, S. Ali, R. Zaidi, D. Mclernon, L. Mhamdi, and M. Ghogho, "Deep Recurrent Neural Network for Intrusion Detection in SDN-based Networks," in *2018 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 25–29, [Online]. Available: http://eprints.whiterose.ac.uk/129091/.

[8] J. Wang, X. Hong, R. Ren, and T. Li, "A real-time intrusion detection system based on PSO-SVM," in *Proceedings. The 2009 International Workshop on Information Security and Application (IWISA 2009)*, 2009, p. 319.

[9] H. Li, W. Guo, G. Wu, and Y. Li, "An RF-PSO Based Hybrid Feature Selection Model in Intrusion Detection System," in *2018 IEEE Third International Conference on Data Science in Cyberspace (DSC)*, 2018, pp. 795–802.

[10] T. Khorram and N. A. Baykan, "Feature selection in network intrusion detection using metaheuristic algorithms," *Int. J. Adv. Res. Ideas Innov. Technol.*, vol. 4, no. 4, pp. 704–710, 2018.

[11] H. Xu, Y. Fu, C. Fang, Q. Cao, J. Su, and S. Wei, "An Improved Binary Whale Optimization Algorithm for Feature Selection of Network Intrusion Detection," in *2018 IEEE 4th International Symposium on Wireless Systems within the International Conferences on Intelligent Data Acquisition and Advanced Computing Systems (IDAACS-SWS)*, 2018, pp. 10–15.

[12] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," *Comput. Secur.*, vol. 81, pp. 148–155, 2019.

[13] S. Mohammadi, H. Mirvaziri, M. Ghazizadeh-Ahsaee, and H. Karimipour, "Cyber intrusion detection by combined feature selection algorithm," *J. Inf. Secur. Appl.*, vol. 44, pp. 80–88, 2019.

[14] A. Divekar, M. Parekh, V. Savla, R. Mishra, and M. Shirole, "Benchmarking datasets for Anomaly-based Network Intrusion Detection: KDD CUP 99 alternatives," in *2018 IEEE 3rd International Conference on Computing, Communication and Security (ICCCS)*, 2018, pp. 1–8.

[15] H. M. Harb, A. A. Zaghrot, M. A. Gomaa, and A. S. Desuky, "Selecting optimal subset of features for intrusion detection systems," 2011.

[16] D. D. Protić, "Review of KDD Cup'99, NSL-KDD and Kyoto 2006+ datasets," *Vojnoteh. Glas.*, vol. 66, no. 3, pp. 580–596, 2018.

[17] F. F. Moghaddam, R. F. Moghaddam, and M. Cheriet, "Curved Space Optimization : A Random Search based on General Relativity Theory," *IEEE*, pp. 1–16, 2012.

[18] M. Yousif and B. Al-Khateeb, "A Novel Metaheuristic Algorithm for Multiple Traveling Salesman Problem," *J. Adv. Res. Dyn. Control Syst. (SCOPUS Q4)*, vol. 10, no. 13, pp. 2113–2121, 2018.

[19] Mohammed Yousif Ibraheem, "A Proposed Crow Swarm Optimization Algorithm," M.Sc. Thesis, College of Computer Science and IT, University of Anbar, 2018.

[20] A. A. Abdul Lateef, S. T. F. Al-Janabi, and B. Al-Khateeb, "Survey on Intrusion Detection Systems based on Deep Learning," *Period. Eng. Nat. Sci.*, vol. 7, no. 3, pp. 1074–1095, 2019.

[21] J. Tian and H. Gu, "Anomaly detection combining one-class SVMs and particle swarm optimization algorithms," *Nonlinear Dyn.*, vol. 61, no. 1–2, pp. 303–310, 2010.

[22] Y. Y. Chung and N. Wahid, "A hybrid network intrusion detection system using simplified swarm optimization (SSO)," *Appl. Soft Comput.*, vol. 12, no. 9, pp. 3014–3022, 2012.

[23] A. S. Eesa, Z. Orman, and A. M. A. Brifcani, "A novel feature-selection approach based on the cuttlefish optimization algorithm for intrusion detection systems," *Expert Syst. Appl.*, vol. 42, no. 5, pp. 2670–2679, 2015.

[24] M. H. Aghdam and P. Kabiri, "Feature Selection for Intrusion Detection System Using Ant Colony Optimization.," *IJ Netw. Secur.*, vol. 18, no. 3, pp. 420–432, 2016.

[25] T. Mehmood and H. B. M. Rais, "Ant colony optimization and feature selection for intrusion detection," in *Advances in machine learning and signal processing*, Springer, 2016, pp. 305–312.