A Variable Structure Learning Automata Approach for the Feature Reduction Problem of Intrusion Detection Systems

Kayvan Asghari[1*], Majid Samadzamini[2], Solmaz Abdollahizad[3]
[1,2,3] Department of Computer Engineering, Sardroud Center, Tabriz Branch, Islamic Azad University, Tabriz, Iran
Email:k.asghari@iau.ac.ir(CorrespondingAuthor)[1*],zamini.m@iau.ac.ir[2],solmaz.abdollahizad@iau.ac.ir[3]

Receive Date: 24 September 2024, Revise Date: 26 October 2024, Accept Date: 03 November 2024

Abstract

A variable structure learning automata based method is proposed in this paperfor solving the feature selection problem in designing the intrusion detection systems.The proposed method can explore the problem's search space using reward and penalty mechanism of learning automata.The target of proposed method is to increase the accuracy rate of the designed intrusion detection system by selecting the most significant features.The UNSW-NB15 intrusion detection dataset is employed for investigating the proposed method. The results of the designed experimentsdemonstrated the performance dominance of the proposed method for most experiments in contrast with some other well-knownmethods.

Keywords: Intrusion detection, Learning automata, Optimization,Feature selection.

1- Introduction

Optimization methods have many applications to find optimal solutions without spending much effort. They also have some weaknesses like finding local optimal solutions. In those cases, the optimization method can not enhance the solutions despite more repetitions.Optimization methods that can produce diverse solutions during the repetitions can improve the local optimum issue. More diversity can cause other problems, such as losing the elite solutions. Thus, managing the appropriate amount of diversity is the most important function of an optimization method. In the first repetitions, the algorithm must generate solutions with a high diversity. This phase, called exploration, provides the opportunity to find promising areas of the search space. In the next phase, the exploitation, the algorithm must focus on the previously found promising solutions.

In the exploitation phase, a local search method is used to search for the optimal solution around the found solutions in the exploration phase.Most of the optimization methods have a bunch of parameters for controlling the balance between exploration and exploitation. These parameters change during the search process, where their values in the first repetitions are tuned so that the

algorithm explores the entire search space of the investigated problem. In contrast, during final repetitions, the parameters' values are tuned to search more focused on the promising solutions found in previous repetitions.

At the end of the game, a proper equipoise among the exploration and exploitation results in the gradual convergence of the optimization method to near-optimal solutions.

The number of computer network cyber attacksis on the rise due to increasing the network-based applications. Therefore, designing accurate intrusion detection systems for computer networks is very important.Intrusion detection systems have different kinds based on the type of detection and analysing the network packets.

One of the classifications is dividing these systems into anomaly-based and abuse-based systems. To build an intrusion detection system, the selection of important features and create a fast and accurate classifier with them are two essential phases. Different methods have been used by researchers for selecting features in intrusion detection systems[1], [2], [3], [4], [5], [6], [7].

An optimization method based on the variable structured learning automatais proposed for feature selection in intrusion detection systems in this paper. A naive Bayesian network as a simple classifier is applied. The learning automata interact with an environment during the iterations, which is the evaluation function for selected features. It tries to find better results by using the reward and penalty mechanism.

2- Previous researches

A concise description of the feature selection problem and optimization algorithms for solving it is provided in this section. In addition, the structure and functionality of the learning automata are presented in the current section.

2-1- The learning automata

A learning automaton is an abstract learning system that is one of the widely used tools in machine learning. In the learning process, the learning automaton tries to recognize the specifications of a random environment, which is the probabilistic relationship between the automaton's actions and the related environment responses. By selecting different actions to interact with the random environment, the learning automata tries to enhance its functionality to find the near-optimal solution.

Learning automata are categorized into fixed and variable structures. The probability of state transition and action change of automata is a fixed value in the first one, where they are updated based on the environment response in the variable structure. The applied automata in this paper are variable structure and state-output type, where each action is related to a unique state[8].

The variable structure learning automaton is defined by$\{\alpha, \beta, p, T\}$, where α is a set of automaton actions ($\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$), β is a set of environmentresponses ($\beta = \{\beta_1, \beta_2, \dots, \beta_m\}$), and $p$is the action probability vector ($p = \{p_1, p_2, \dots, p_r\}$). The learning algorithm is defined by $p(n+1)=T[\alpha(n), \beta(n), p(n)]$.

Each automaton in the learning automata has a finite set of actions which are selected randomly in each iteration according to the action probability vector. In this way, the automaton interacts with a random environment, according to Figure 1.

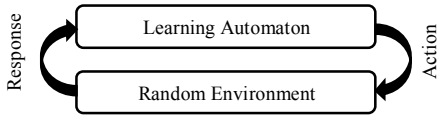

Fig. 1. Interaction of the learning automata and the random environment

The random environment accepts the selected action (α_i) as its input. The environment returns a response (β_i) for each input, which depends on the input action and causes an update of the action probability vector. The environment works with a set of external conditions and their effects on the operation of the learning automata.The response of the environment in moment *n*can be a reward ($\beta(n) = 1$) or a penalty ($\beta(n) = 0$). The learning automaton increases the probability of selected action (α_i) with a reward responseaccording to formula (1) and decreases the probability of selected action with a penalty response according to formula (2).In this way, the sum of action probability for all actions remains a constant value[8].

$$p_i(n+1) = p_i(n) + a(1 - p_i(n))$$
$$p_j(n+1) = (1 - a)p_j(n), \quad (j \neq i) \tag{1}$$

$$p_i(n+1) = (1 - b)p_i(n)$$
$$p_j(n+1) = \frac{b}{r-1} + (1 - b)p_j(n), (j \neq i) \tag{2}$$

In formulas (1) and (2), *a* represents the reward parameter, *b* represents the penalty parameter, and *r* is the number of possible actions. *a* and *b* are numbers between 0 and 1.

Some of the applications of the learning automata tool are solving NP problems, image data compression, job scheduling[9], pattern recognition, network routing[10], neural and Bayesian network structure optimization[11], and database query optimization[12].

3- The Variable Structure learning Automata for Feature Selection

A lot of effort and time is needed for a bunch of network packets with dozens of features to detect a normal event or attack. Some of those features include no new information, and some of them are duplicated. Thus, several feature selection methods have been proposed to construct more effective and quick intrusion detection systems[3].

A novel learning automata-based method is introduced in the current section to tackle the feature selection optimization problem in intrusion detection systems. The introduced algorithm benefits from machine learning techniques to solve the mentioned problem.The selected optimal feature set is employed to construct an intrusion detection event classifier. The optimal feature set represents all features of the network packets. The constructed classifier will investigate each packet with high speed and accuracy and distinguish the normal event from the attack.

The proposed learning automata-based method is evaluated in this paper by optimizing the selected feature set to

construct a classifier. The naive Bayesian network [13]as a simple and easy-to-implement classifier is used in the experiments. Other kinds of classifiers have been employed for intrusion detection systems in previous works. But, as the main focus of this paper is on the feature selection phase, the naive Bayesian classifier has been selected.

The flowchart of applying the learning automata for the feature selection in the intrusion detection system is illustrated in Figure 2.

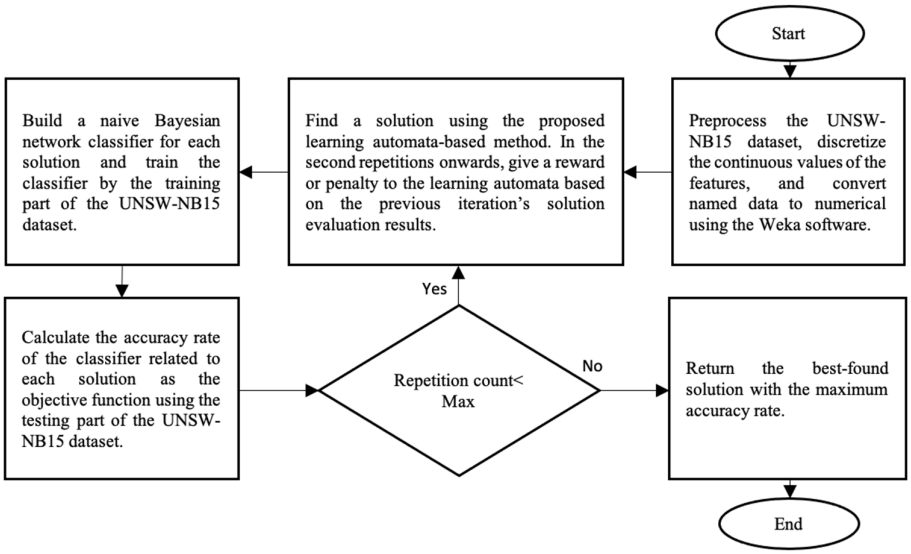

Fig 2. Applying the learning automata for feature selection in the intrusion detection systems

Figure 3 depicts a detailedview of the learning automata's role in the proposed algorithm for feature selection. Each solution includes an array of numbers representing the selected features.To select *n* features, *n* learning automata is employed, where each automaton is responsible for selecting one feature, as Figure 3 illustrates.

The features are considered as actions of the automaton, so there are *n* learning automata, each one with *n* actions. Each automaton selects an action in each iteration and determines one selected feature. To avoid duplicated features, the learning automata repeat the action selection process until the non-repetitive and distinct set of features is selected. All actions of the learning automata have identical selection probability values at the beginning of the algorithm, which changes affected by rewards and penalties during the learning process. A naive Bayesian network is constructed with the selected features and trained using the training dataset after feature selection in each iteration by the proposed and other compared algorithms.

In the next phase, the testing part of the intrusion detection dataset is employed by the constructed classifier for the evaluation of the selected features. If the accuracy rate of the current classifier is improved compared to previous iterations, the learning automata are rewarded. It means that the probability value of the selected action by the automata, which is equivalent to the probability of choosing

the related feature, is increased. In contrast, the probabilities of other actions are decreased. When the accuracy rate of the current classifier is not improved compared to previous iterations, the automata are penalized for changing their actions and consequently, the selected features. For penalizing an action of an automaton, the probability value of the action is decreased, and the probability of other actions is increased.The introduced method is a global search algorithm since it considers both exploration and exploitation phases with its learning mechanisms. Balancing the exploration and exploitation phases is done via the proposed method, which results in finding the near-optimal solution.As expected for an optimization algorithm, the learning automata-based method explores the feature selection problem's search space effectively and finds promising solutions.

Algorithm 1 presents the pseudo-code of the proposed learning automata-based method for feature selection.The proposed method has a low execution time compared to the other algorithms, since it works on a single solution.

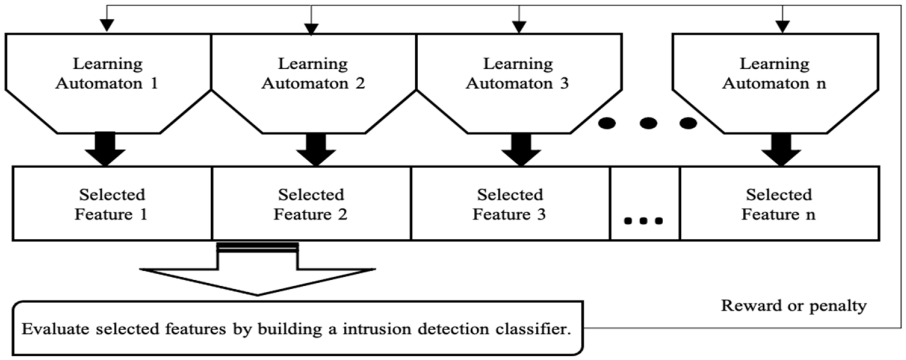

Fig 3.A detailed view of learning automata's rolein the proposed approach for feature selection

Algorithm 1. Pseudo-code for the learning automata-basedmethod for feature selection

```
CreateSelected_Features[i](1≤ i ≤ number_of_features);
Create Learning_Automaton_i , where all actions of each automaton have an equal probability of being selected (1≤i≤number_of_features);
Best_AR=0;
t=1;
While(t<maximum number of iterations)
Begin
For i = 1 To number_of_features Do
Selected_Features[i] = Select an action by Learning_Automaton_i
EndFor
  Create an intrusion detection classifier using Selected_Features and training dataset;
Current_AR= Accuracy rate of classifying the test dataset with created classifier;
IfCurrent_AR>Best_ARThen
Best_AR= Current_AR;
Best_Selected_Features=Selected_Features;
Give Reward to all LA_i(1≤ i ≤ number_of_features) ;
Else
rnd=Generate a random number between 0 and 1;
P = 1-  (t / maximum number of iterations);
ifrnd<P
Give Penalty to all LA_i(1≤ i ≤ number_of_features);
End If;
End If;
  t++;
End While
Return Best_Selected_Features
```

In Algorithm 1, a vector called *Selected_Features* is formed at the beginning, representing the currently selected features. Then, according to the number of selected features, learning automata are created, where the actions of all automata have the same probability of selection. The variable *Best_AR* represents the best value of the accuracy rate found so far and the variable *t* represents the current iteration number of the algorithm. Inside the while loop of the algorithm, in each iteration, the learning automata choose the currently selected features relative to their action selection. Next, using the selected features, a naive Bayesian network classifier is created.

The built classifier is trained with the training records of the intrusion detection dataset and then evaluated with the test records to obtain the accuracy rate. Then, the accuracy rate obtained for the current classifier, built with the currently selected features, is compared with *Best_AR*. If better, all learning automata are rewarded. Otherwise, all learning automata are penalized with a probability controlled by the variable *P*. The job function of variable *P* is to balance the exploration and exploitation operations. The *P* variable is linearly reduced from *1* to *0* during the iterations of the algorithm, which reduces the possibility of penalizing the automata during the iterations. In fact, in the initial iterations of the algorithm, due to more penalties, the automata change their action frequently, and this causes more exploration of the search space. But in the final iterations, the probability of changing the actions by the automata is reduced and they focus on the actions or selected features in the previous steps so that the exploitation operation is carried out by the algorithm.At the end of the algorithm, the best-selected features are returned.

4- Results of the Experiments

To evaluate the proposed variable structure learning automata-based algorithm for feature selection, one of the well-known intrusion detection datasets named UNSW-NB15 [14]has been applied. The Matlab 2022a software has been used to implement the proposed and other existing algorithms.

4-1- The UNSW-NB15 Intrusion Detection Dataset

One of the well-known intrusion detection datasets, which includes newer attacks with nine types, is the UNSW-NB15 dataset[14]. The UNSW-NB15 dataset, which has been used in the experiments of this paper, has records including 42 features. The structure and record types of the UNSW-NB15 dataset are shown in Table 1.

Table 1. The contents of the UNSW-NB15 dataset

Kind of record	Records for train	Records for test
Normal	56000	37000
Analysis	2000	677
DoS	12264	4089
Fuzzers	18184	6062
Shellcode	1133	378
Backdoors	1746	583
Generic	18871	40000
Reconnaissanc	10491	3496
Worms	130	44
Exploits	33393	11132
Records count	175341	82332

4-2- Evaluation measures and solution structure

To evaluate the obtained features by the learning automata and other optimization algorithms, measures like the accuracy rate of detection, false positive rate, and the attack detection rate has been employed to design the intrusion detection system [15]. The confusion table in Table 2, has been used to calculate the mentioned measures.

The first measure to evaluate the proposed feature selection method is the accuracy rate (*AR*) for the detection of network packets that are correctly categorized.The accuracy rate must be high. The accuracy rate can be calculated by formula (3).

Table 2. Confusion table for obtaining the evaluation measures

		Event type as estimated	
		Attack	**Normal**
Event type in real	**Normal**	False Positive (FP)	True Negative(TN)
	Attack	True Positive (TP)	False Negative(FN)

$$AR = \frac{TN + TP}{TP + FN + FP + TN} \quad (3)$$

The second measure to evaluate the proposed feature selection method is the attack detection rate (*DR*). The *DR*, which can be calculated by formula (4), is the rate of correctly categorized network packets.The rate of normal network packets that are incorrectly categorized as network attack is the third measure, named false positive rate (*FPR*). The *FPR* can be calculated by formula (5).

$$DR = \frac{TP}{FN + TP} \quad (4)$$

$$FPR = \frac{FP}{FN + TP} \quad (5)$$

To create a fitness value for the evaluation of selected features by the compared algorithms, the *AR* for categorization of network packets using a classifier is employed. Various classification algorithms can be applied to evaluate the network packets using the selected features in intrusion detection systems, such as Bayesian and neural networks[16]. To have a simple classifier to evaluate the selected features, the naive Bayesian network is used in this paper. The naive Bayesian network is formed by the selected features of the compared and proposed algorithms.

The introduced learning automata-based algorithm of this paper is analogized with the gray wolf optimization [17], secretary bird optimization [18], particle swarm optimization [19], and the genetic algorithm[5] to solve the feature selection problem of intrusion detection systems.

The structure of solution for the proposed method and the implemented genetic algorithm[5]is similar and includes an array of integer values, presenting the selected features numbers. But, for the other compared algorithms, each solution includes 42 real values, which demonstrate the significance of 42 features. The selected features for the algorithms of the second category are distinguished by higher values.

A sample solution for the algorithms of the second category is indicated in formula

(6). This solution includes 10 real values, where 4 of them (first, fifth, eighth, and tenth) are most important ones, as they have the highest values.

[13.1 10 11.2 2 15 1.6 14.2 7.8 5.1 18] (6)

The compared algorithms change the numbers in the solutions' population across the search space of the feature selection problem during the repetitions to find the significant features.These features are employed to construct the intrusion detection classifiers. The classifiers are used to classify the test set of the intrusion detection dataset. The count of selected features is also significant in building a classifier for intrusion detection systems. More features not only increase the processing load but also decrease the accuracy rate of event classification. Thus, a multi-objective algorithm is needed to precisely solve the feature selection problem, which considers the selected feature count as an objective function. However, to simplify the experiments using the proposed and compared single-objective algorithms, the count of features in the experiments of this paper has been decided to be 4, 8, 12, and 18. The accuracy rate of proposed and existing optimization methods for the UNSW-NB15 intrusion detection dataset is depicted in Figure 4 and Table 3.

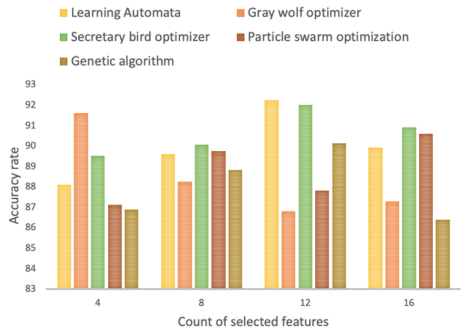

Fig4. Compairing the algorithms with the accuracy rate measurefor 4, 8, 12, and 18 selected features

Table 3. Compairing the algorithms with the accuracy rate measure for 4, 8, 12, and 18 selected features

Count of Selected Features / Method Name	4	8	12	18
Variable structure learning automata	88.11	89.61	**92.25**	89.94
Gray wolf optimizer	**91.62**	88.27	86.81	87.31
Secretary bird optimizer	89.54	**90.09**	92.01	**90.92**
Particle swarm optimization	87.13	89.76	87.82	90.61
Genetic algorithm	86.91	88.83	90.14	86.41

The Figure 4 and Table 3 indicate that the learning automata-based algorithm produces a higher accuracy rate for 12 selected features compared to the other algorithms. But for other selected feature counts, the accuracy rate of the introduced algorithm is lower than the others. However, the highest accuracy rate belongs to the learning automata-based method. The detection rate of compared algorithms for the UNSW-NB15 intrusion detection data set is depicted in Figure 5 and Table 4.

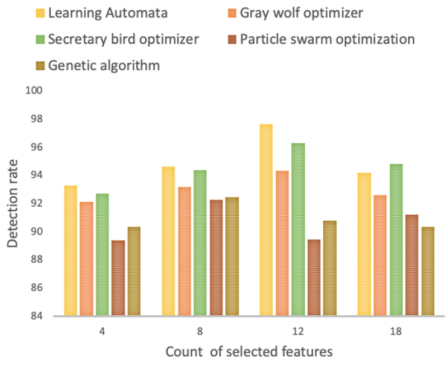

Fig 5. Compairing the algorithms with the detection rate measure for 4, 8, 12, and 18 selected features

Table 4. Compairing the algorithms with the detection rate measure for 4, 8, 12, and 18 selected features

Method Name \ Count of Selected Features	4	8	12	18
Variable structure learning automata	93.28	94.63	97.67	94.21
Gray wolf optimizer	92.13	93.21	94.35	92.62
Secretary bird optimizer	92.72	94.41	96.32	94.84
Particle swarm optimization	89.41	92.29	89.45	91.23
Genetic algorithm	90.35	92.47	90.81	90.37

For the detection rate measure, as Figure 5 and Table 4 show, the learning automata-based algorithm has a higher rate for the number of 4, 8, and 18 feature counts. But for 18 features, the intrusion detection rate of the secretary bird optimizer algorithm is the highest value. The learning automata-based algorithm obtains the second-highest detection rate. The obtained false positive rate measure of compared algorithms for 4, 8, 12, and 18 selected features and the UNSW-NB15 data set is presented inFigure 6 and Table 5.

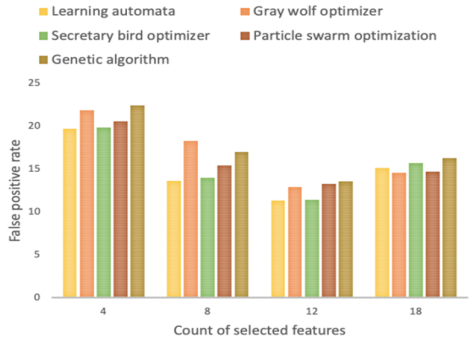

Fig 6. Compairing the algorithms with the false positive rate measure for 4, 8, 12, and 18 selected features

Table 5. Compairing the algorithms with the false positive rate measure for 4, 8, 12, and 18 selected features

Method Name \ Count of Selected Features	4	8	12	18
Variable structure learning automata	19.84	**13.62**	**11.3**	15.14
Gray wolf optimizer	21.86	18.24	12.91	**14.52**
Secretary bird optimizer	**19.7**	14.01	11.44	15.71
Particle swarm optimization	20.56	15.38	13.29	14.68
Genetic algorithm	22.39	16.96	13.54	16.27

A lower value for the false positive rate measure is desirable to have an efficient method for the feature selection. As can be seen in Figure 6 and Table 5, the proposed learning automata-based algorithm performs better than the others for 8 and 12 features. But, the gray wolf optimizer has the lowest value for 18 and the secretary bird optimizer has the lowest false alarm rate for 4 features. However, the lowest value (11.3) for all the cases belongs to the proposed method for 12 selected features.

The outcomes of compared algorithms in the figures and tables indicate that 12 is a proper value for the selected features count to build a classifier for the intrusion detection system. Considering the acquired outcomes, the learning automata-based method can achieve satisfactory solutions in many cases for selecting features to develop the intrusion detection system.

5- Conclusion

The performed experiments indicated that the recommended method provides high efficiency in contrast with the other optimization approaches to solve the feature reduction problem in intrusion detection systems. The variable structure learning automata explore the search space of the problem using the penalty and reward mechanisms and find the near-optimal answers quickly. Applying a hybrid algorithm of learning automata and an optimization method can be a future work for this paper. On the other hand, employing artificial neural networks besides the learning automata can be another future work.

References

[1] 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, Apr. 2015, doi: 10.1016/j.eswa.2014.11.009.

[2] B. Selvakumar and K. Muneeswaran, "Firefly algorithm based feature selection for network intrusion detection," *ComputSecur*, vol. 81, pp. 148–155, Mar. 2019, doi: 10.1016/j.cose.2018.11.005.

[3] T. Khorram and N. A. Baykan, "Feature selection in network intrusion detection using metaheuristic algorithms," *International Journal Of Advance Research, Ideas and Innovations in Technolog*, vol. 4, no. 4, pp. 704–710, 2018.

[4] M. H. Aghdam and P. Kabiri, "Feature Selection for Intrusion Detection System Using Ant Colony Optimization," 2016.

[5] Z. Halim *et al.*, "An effective genetic algorithm-based feature selection method for intrusion detection systems," *ComputSecur*, vol. 110, p. 102448, Nov. 2021, doi: 10.1016/j.cose.2021.102448.

[6] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on Pigeon Inspired Optimizer," *Expert Syst Appl*, vol. 148, p. 113249, Jun. 2020, doi: 10.1016/j.eswa.2020.113249.

[7] T. S. Naseri and F. S. Gharehchopogh, "A Feature Selection Based on the Farmland Fertility Algorithm for Improved Intrusion Detection Systems," *Journal of Network and Systems Management*, vol. 30, no. 3, pp. 1–27, Jul. 2022, doi: 10.1007/s10922-022-09653-9.

[8] "Learning Automata: An Introduction - Kumpati S. Narendra, Mandayam A.L. Thathachar - Google Books." Accessed: Jul. 24, 2024. [Online]. Available: https://books.google.de/books/about/Learning_Automata.html?id=ZwbCAgAAQBAJ&redir_esc=y

[9] S. Sabamoniri, K. Asghari, and M. Javad Hosseini, "Solving Single Machine Total Weighted Tardiness Problem using Variable Structure Learning Automata," *Int J Comput Appl*, vol. 56, no. 1, pp. 37–42, Oct. 2012, doi: 10.5120/8858-2816.

[10] G. I. Papadimitriou, M. S. Obaidat, and A. S. Pomportsis, "On the use of learning automata in the control of broadcast networks: A methodology," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 32, no. 6, pp. 781–790, Dec. 2002, doi: 10.1109/TSMCB.2002.1049612.

[11] K. Asghari, M. Masdari, F. SoleimanianGharehchopogh, and R. Saneifard, "A fixed structure learning automata□based optimization algorithm for structure learning of Bayesian networks," *Expert Syst*, vol. 38, no. 7, Nov. 2021, doi: 10.1111/exsy.12734.

[12] K. Asghari, A. S. Mamaghani, and M. R. Meybodi, "An evolutionary algorithm for query optimization in database," in *Innovative Techniques in Instruction Technology, E-Learning, E-Assessment, and Education*, Kluwer Academic Publishers, 2008, pp. 249–254. doi: 10.1007/978-1-4020-8739-4_44.

[13] S. Mukherjee and N. Sharma, "Intrusion Detection using Naive Bayes Classifier with Feature Reduction," *Procedia Technology*, vol. 4, pp. 119–128, Jan. 2012, doi: 10.1016/j.protcy.2012.05.017.

[14] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion

detection systems (UNSW-NB15 network data set)," in *2015 Military Communications and Information Systems Conference, MilCIS 2015 - Proceedings*, Institute of Electrical and Electronics Engineers Inc., Dec. 2015. doi: 10.1109/MilCIS.2015.7348942.

[15] H. J. Liao, C. H. Richard Lin, Y. C. Lin, and K. Y. Tung, "Intrusion detection system: A comprehensive review," Jan. 01, 2013, *Academic Press*. doi: 10.1016/j.jnca.2012.09.004.

[16] A. Shenfield, D. Day, and A. Ayesh, "Intelligent intrusion detection systems using artificial neural networks," *ICT Express*, vol. 4, no. 2, pp. 95–99, Jun. 2018, doi: 10.1016/j.icte.2018.04.003.

[17] Q. M. Alzubi, M. Anbar, Z. N. M. Alqattan, M. A. Al-Betar, and R. Abdullah, "Intrusion detection system based on a modified binary grey wolf optimisation," *Neural Comput Appl*, vol. 32, no. 10, pp. 6125–6137, May 2020, doi: 10.1007/s00521-019-04103-1.

[18] Y. Fu, D. Liu, J. Chen, and L. He, "Secretary bird optimization algorithm: a new metaheuristic for solving global optimization problems," *ArtifIntell Rev*, vol. 57, no. 5, pp. 1–102, May 2024, doi: 10.1007/S10462-024-10729-Y/FIGURES/4.

[19] A. J. Malik, W. Shahzad, and F. A. Khan, "Network intrusion detection using hybrid binary PSO and random forests algorithm," *Security and Communication Networks*, vol. 8, no. 16, pp. 2646–2660, Nov. 2015, doi: 10.1002/sec.508