DOI: 10.6977/IJoSI.202504_9(2).0002
S.W. Kareem, H.Q. Awla, etc./Int. J. Systematic Innovation, 9(2), 14-25 (2025)

Structure learning of Bayesian networks using sparrow optimization algorithm

Shahab Wahhab Kareem[1,2]*, Hoshang Qasim Awla[3], Amin Salih Mohammed[4]

[1]Department of Technical Information Systems Engineering, Erbil Technical Engineering College, Erbil Polytechnic University, Erbil, Iraq

[2]Department of Computer Technical Engineering, Al-Qalam University College, Kirkuk, Iraq

[3]Department of Computer Science, Faculty of Science, Soran University, Soran, Iraq

[4]Department of Software and Informatics, Salahaddin University, Erbil, Iraq

*Corresponding author E-mail: Shahab.kareem@epu.edu.iq

(Received 12 June 2024; Final version received 10 December 2024; Accepted 19 December 2024)

Abstract

Bayesian networks are powerful analytical models in machine learning, used to represent probabilistic relationships among variables and create learning structures. These networks are made up of parameters that show conditional probabilities and a structure that shows how random variables interact with each other. The structure is shown by a directed acyclic graph. Despite the NP-hard nature of learning Bayesian network structures, there has been significant progress in improving the accuracy of approximation solutions. The main focus is on score-based search strategies, which make use of functions to evaluate network models and identify structures with high scores. This study is significantly focused on structure learning Bayesian networks using the Bayesian Dirichlet equivalent uniform scoring function and metaheuristic search strategies. To this end, this paper presents the sparrow optimization algorithm (SOA), a new metaheuristic algorithm derived from the foraging behavior of sparrows. SOA performs a concurrent optimization in the solution space by simultaneously performing a local and global search that leads to the discovery of near-optimal structures. The results from our experiments on several benchmark datasets show that SOA yields overall better performance than SA and greedy search algorithms. In particular, it is claimed that by applying the proposed approach of SOA, the convergence speed is significantly higher compared with the existing ones; F1 score is 0.35 and 0.05 for the Hamming distance with better results. Given these results, signed operators prove to be very efficient in SOA's Bayesian network structure learning as a concept, especially for real-world use.

Keywords: Search and Score, Global and Local Search, Bayesian Network, Sparrow Search Optimization Algorithm, Structure Learning

1. Introduction

The Bayesian network is widely regarded as a widely used analytical model in machine learning for constructing the probabilistic framework of knowledge (Ji et al., 2012). It is feasible to employ knowledge design, reasoning, and inference systematically (Fortier et al., 2013). A directed acyclic graph (DAG) is utilized to illustrate the structure of a Bayesian network, consisting of two fundamental elements: the network's parameters and its structure. The structure is used to express dependencies on variability, whereas the parameters are employed to describe conditional probabilities. The task of addressing the learning structure of a Bayesian network might be challenging in the absence of a well-defined search plan. However, significant efforts have been made to develop approximation algorithms for acquiring knowledge about the network structure. Achieving the appropriate NP-hard class is necessary to overcome the challenges associated with learning the structure of a Bayesian network from a dataset (Li & Chen, 2014). The main components of structural learning in Bayesian networks are two distinct processes. While the second strategy employs a combination of score and search tactics, the first approach is focused on constraints (Margaritis, 2003). Until the desired metric value is reached, the scoring and search approach is used to methodically evaluate each potential network structure and explore the range of Bayesian network configurations. Using a function to evaluate the network and the provided data to maximize the score – the intended result – is the basis of score-based approaches (Fast, 2010). The Bayesian score and the information-theoretic score are the two primary criteria used to generate the score function approach. Bayesian networks are valuable tools in decision-making processes because they can discover connections between variables and make predictions utilizing uncertain data. The search and scoring component performs an essential part of the Bayesian network structure learning (BNSL) process. The process involves examining several potential network structures and assessing their suitability by utilizing scoring criteria that determine their level of compatibility. The complexity increases with the rise in the number of variables, leading to a significant expansion in the number of possible DAGs that can represent the relationships between variables. As the number of variables grows, a typical issue that arises is the NP-hard problem, which occurs in the majority of search spaces. The NP-hard problem of BNSL relates to the difficulty of determining the optimal network structure for a given dataset. This problem becomes more difficult when using search and scoring methods, which require analyzing a wide range of possible structures of networks and evaluating their sufficiency. The NP-hardness arises from the exponential growth of the search space with an increasing number of variables. This makes it practically impossible to thoroughly search over all possible options, particularly for huge datasets.

The application of the information-theoretic score involves the utilization of several techniques, such as mutual information tests, minimum description length, normalized minimum likelihood, log-likelihood, Akaike information criterion, and Bayesian information criterion. The Bayesian score is utilized in several approaches, such as BDe (Bayesian Dirichlet, where "e" represents likelihood-equivalency), BDeu (Bayesian Dirichlet equivalent uniform, where "u" denotes uniform joint distribution), and K2 (Cooper & Herskovits, 1992).

The complexity of structure learning can be enhanced through the utilization of diverse search strategy approaches. The literature includes references to several algorithms, including Bee Colony (Li & Chen, 2014), Swarm Intelligence (Cowie et al., 2007), Ant Colony (Salama & Freitas, 2012), Hybrid Algorithm (He & Gao, 2018; Kareem & Okur, 2018; Li & Wang, 2017), Simulated Annealing Algorithm (Hesar, 2013), Bacterial Foraging Optimization (Yang et al., 2016), and Genetic Algorithms (Larrañaga & Poza, 1996). Numerous algorithms have been examined in the existing body of literature (Djan-Sampson & Sahin, 2004; Fan et al., 2014; Orphanou et al., 2018; Yuan et al., 2011; Rahier et al., 2019). Several algorithms have been proposed in the literature, such as the Breeding Swarm Algorithm (Khanteymoori et al., 2018), Binary Encoding Water Cycle (Wang & Liu, 2018), Pigeon Inspired Optimization (Kareem & Okur, 2019), Cuckoo Optimization Algorithm (Askari & Ahsaee, 2018), and Minimum Spanning Tree Algorithm (Sencer et al., 2013). The utilization of swallow optimization, a cutting-edge metaheuristic approach, is prevalent in the field of structure learning within Bayesian networks. This study presents a novel approach to address the difficulty of acquiring knowledge about the architecture of Bayesian networks. This study provides a comparative assessment of the technique mentioned earlier. This paper presents the sparrow optimization algorithm (SOA), a new metaheuristic strategy based on sparrows' foraging habits in enhancing the structure learning of Bayesian networks. SOA does the concurrent running of both local and global searches, thereby improving the discovery of almost optimal structures. In this paper, benchmark datasets are used to establish the superiority of SOA over conventional algorithms such as SA and greedy search, especially in terms of convergence rate and accuracy. The proposed algorithm can be considered very efficient – it does not take a long time to produce results and seems perfectly capable of dealing with big data. This work emphasizes that SOA can greatly enhance the speed of the BNSL while yielding much better results than other similar approaches.

The subsequent sections of this study are organized in the following manner: Section 2 provides an explanation of the principles underlying structure learning in Bayesian networks. Section 3 provides a concise summary of the SOA. Section 4 delves into the approach extensively and presents the experimental findings. The conclusions are presented in Section 5.

2. Structure Learning of Bayesian Networks

The composition of a Bayesian network has two distinct components, namely G and P. A DAG is the main category, consisting of a finite set of vertices (or nodes), V, that are connected by specified edges (or links), E. The representation of it is denoted by the symbol G(V; E). The equation P = P (Xi | Pa (Xi)) describes the collection of conditional probabilistic distributions that are unique for each variable Xi, which corresponds to the vertices in a graph. In addition, it should be noted that the function Pa(Xi) represents the collection of parents of node Xi inside graph G (Cowie et al., 2007). This model enables the depiction of a basic probabilistic combination for a (G; P) network in the following way:

$$P\,(X_i, \ldots X_n)=\Pi_{i=1}\,P(X_i|\,Pa(X_i)) \quad (1)$$

On the other side, the scoring system is dependent on several criteria, such as the minimal length of description, information and entropy, and Bayesian approaches (Campos, 2006). The posterior likelihood of the Bayesian network can be represented according to the principles of Bayesian inference as follows:

$$P(G/D) = P(D/G).P(G)/\Sigma_G P(D/G') \quad (2)$$

The marginal likelihood P(D|G) in Eq. (2) is defined as follows using the normality constant P(D):

$$P(G/D) = \int P(D/G, \theta)\, P(\theta/G)\, d\theta \quad (3)$$

The prevailing belief is that P(D) does not constitute a component of the structure of G's Bayesian network. The variable represents the parameter of the model, whereas the prior probability is given as P(G'). Hence, it is possible to calculate the next distribution of the network structure, provided that the marginal probability of all potential topologies has been determined (Zhang & Liu, 2008). Structure learning approaches utilize score-based tactics by integrating the current and historical scores of the structure. The eventual representation of the score is (Heckerman et al., 1995):

$$Score\,(G, D)=\Sigma\, Score(X_i, Pa(X_i), D(X_i, pa(X_i))) \quad (4)$$

3. SOA

Metaheuristics refer to methods that draw inspiration from nature and are used to find possible solutions to complex computational optimization problems. Animals such as fireflies-BAT (Reddy & Khare, 2016), cuckoos (Gadekallu & Khare, 2017), ants, pigeons, fish, bees, and others have utilized their swarming behaviors in metaheuristics (Gandomi et al., 2013). The metaheuristics possess several fundamental qualities, including uniformity, flexibility, illation-free instruments, and the capability to ignore local optima (Mirjalili et al., 2014). The metaheuristic algorithm proposed by Segundo et al. (2019) is derived from the sparrow routine, which is used for hunting food. The SOA method is a reliable and resilient method designed for addressing stochastic population-based problems that require complex configurations involving several parameters and operating in three stages.

The recommended process was impacted by the way sparrows forage, that is, how they look for food when they are in the air. The sparrow is a solitary creature that adapts its hunting strategy based on its needs. Nevertheless, distinct strategies emerge, and impressive models adhere to the essential principles of flight and navigation in a secure area, as supported by many research findings (Tucker, 1998; 2000). The objectives are assessed for the maximum level of flying accomplishment during different stages of delicate searching or hunting (Hedenström et al., 1999). The implementation methodology of flying in the framework involves the computation of the mechanical power required for navigation, determining the average speeds throughout the flight, and adapting to wind conditions (Hedenström et al., 1999). The sparrow is one of the fastest creatures in the world. The main hunting or searching activity takes place throughout the day, including in the morning. The prime source of nutrition is derived from minor to medium-sized prey and occasionally includes insects (Dekker, 2009).

Based on the above description of sparrows, the authors are able to formulate a mathematical model for developing the sparrow search algorithm. To simplify matters, they conceptualized the subsequent actions of the sparrows and devised related principles.

(i) Producers often possess abundant energy reserves and offer browsing locations or instructions for all scavengers. Its primary function is to detect and locate regions abundant in healthy food resources. The energy reserves are contingent upon the individuals' fitness values being evaluated.

(ii) When the sparrow recognizes its attacker, it starts chirping as an alerting signal. If the signal value outstrips the safety threshold, the producers must direct all those searching for resources to the designated safe region.

(iii) Every sparrow has the probability to suit a producer by seeking out improved food sources, but the ratio of production to scroungers remains constant among the entire population.

(iv) Sparrows with greater energy levels would function as producers. A group of famished scavengers will be more inclined to migrate to diverse positions in search of meals to acquire additional energy.

(v) The scavengers trail behind the producer that can offer the highest quality nourishment to forage for food. Meanwhile, certain opportunistic individuals may continuously surveil the producers and engage in food competition to enhance their predation ratio.

(vi) When sparrows on the outside of the group become aware of danger, they promptly move to a secure region to achieve a more advantageous situation. Conversely, sparrows situated in the center of the group exhibit unpredictable movement patterns to maintain proximity to their peers. During the simulation experiment, the authors must use virtual sparrows to find sustenance. A specific matrix depicts the spatial distribution of sparrows:

$$X = \begin{bmatrix} X_{1,1} & \cdots & X_{1,d} \\ \vdots & \ddots & \vdots \\ X_{1,n} & \cdots & X_{n,d} \end{bmatrix} \tag{5}$$

$$F_X = \begin{pmatrix} f(X_{1,1}, X_{1,2}, \ldots X_{1,d} \\ f(X_{12,1}, X_{2,2}, \ldots X_{2,d} \\ \vdots \\ f(X_{n,1}, X_{n,2}, \ldots X_{n,d} \end{pmatrix} \tag{6}$$

Each item in the "FX" array represents the fitness value of a single sparrow, whereas the variable "n" shows the total number of sparrows. During the search phase in the SOA, food is prioritized for those with higher fitness values. Furthermore, producers also take on the responsibility of acquiring food supplies and directing population movement as a whole. Because of this, the producers are able to look at a wider variety of resources for food than the scavengers. The producer's location is changed at each iteration in the following ways, per rules (i) and (ii):

$$V_{i,j}^{t+1} = \begin{cases} V_{i,j}^{t}.\exp\left(\dfrac{-i}{\alpha.iter_{max}}\right) & if\ R_2 < ST \\ V_{i,j}^{t} + Q.L & if\ R_2 > ST \end{cases} \tag{7}$$

In this context, the variable t shows the current iteration, while j ranges from 1 to d. The notation V_(i,j) t represents the value of the jth dimension of the ith sparrow at iteration t. The term "itermax" is a constant that denotes the upper limit of iterations. Let α denote a random number that falls within the interval (0, 1). The variable R2, which ranges from 0 to 1, denotes the alert value. ST, where ST ranges from 0.5 to 1.0, denotes the safety level. The variable Q exhibits stochasticity and adheres to a normal distribution. The matrix L is a vector space with dimensions 1 × d, where each element is equal to 1. In instances where the resource-to-search ratio (R2) falls below the search threshold (ST), signifying the lack of predators, the producer commences the wide search mode.

If the value of R2 is greater than or equal to ST, it indicates that certain sparrows have become aware of the predator's presence, and each sparrow needs to rapidly reposition to alternative secure locations. Regarding the individuals who scrounge, it is necessary to implement and uphold regulations (iv) and (v). As previously said, certain scavengers monitor the producers with greater frequency. Upon discovering that the producer has located high-quality sustenance, they promptly abandon their present location to vie for nourishment. If they emerge victorious, they will promptly obtain the producer's food. Otherwise, they will persist according to the guidelines (v). The formula for updating the position of the scrounger is described as follows:

$$X_{i,j}^{t+1} = \begin{cases} Q.\exp\left(\dfrac{X_{worst}^{t} - X_{ij}^{t}}{j^2}\right) & if\ i > n/2 \\ X_p^{t+1} + \left|X_{ij}^{t} - X_P^{t+1}\right|.A^{+}.L & otherwise \end{cases} \tag{8}$$

The parameter "XP" shows the optimal position set by the manufacturer. The abbreviation Xworst describes the current world position that is usually viewed as the most unfavorable. The matrix A is a 1 ò d matrix in which each element is assigned a random value of either 1 or Ω1. A+ is the result of multiplying the transposition of A by the inverse of the product of A and its transpose. If the value of i exceeds n/2, it means that the ith scrounger with the lowest fitness rating is highly probable to be facing starvation. In the simulated experiment, it is assumed that a subset of the sparrow population, totaling around 10–20% of the total, possessed knowledge of the potential risk. The origins of these sparrows are created in a random manner inside the population. The mathematical model can be denoted in accordance with rule (vi) as follows:

$$X_{i,j}^{t+1} = \begin{cases} X_{best}^{t} + \beta.\left|X_{i,j}^{t} - X_{best}^{t}\right| & if\ f_i > f_g \\ X_{i,j}^{t} + K.\left(\dfrac{\left|X_{i,j}^{t} - X_{worst}^{t}\right|}{(f_i - f_w) + \varepsilon}\right) & if\ f_i = f_g \end{cases} \tag{9}$$

The notation "Xbest" indicates the current global optimal condition. The step size parameter, represented by β, is distributed normally, with a variance of 1 and a mean of 0. The variable K is restricted and denotes a stochastic number inside the interval [−1, 1]. The variable fi signifies the current sparrow's fitness value. fw represents the current global lowest fitness value, whereas fg represents the current global maximum fitness value. The constant ε is the minimum value needed to avoid division by zero errors. To make things easier to grasp, the sparrow is close to the group's edge when fi > fg. Xbest reliably illustrates the location of the population center and ensures its security in the surrounding area. The equation fi = fg indicates that the sparrows, which are in the interior of the population, are aware of the threat and are forced to fly toward the other members of the group. The variable K represents the path of the sparrow's drive as well as the coefficient that controls the step's size. The pseudo-code algorithm, which is generated from the conception and viability of the previously described model, may be used to define the basic operations of the service-oriented architecture (SOA).

4. SOA For Bayesian Network Structure Learning

The approach that is proposed makes use of the SOA paradigm as a search tool to explore Bayesian network architecture. The architecture of a Bayesian network is evaluated using a measure called BDeu. The SOA algorithm is an iterative procedure that considers a population of sparrows and assigns a prospective position and velocity to each bird within a predetermined area. The search zone is defined as this area. The suggested approach makes use of several strategies. The first method uses Eq. (8) to investigate the necessary process if (R2 < ST). Equation VII is used in the alternate method if this requirement is not satisfied. Proceed with the required procedure if the value of i exceeds n divided by 2. To get the ideal location given in Eqs. (8) and (9), compare the BDeu score functions of the two phases. The pseudocode for this method is shown in Fig. 1. In the process of building the SOA algorithm's answer, several neighborhoods in the search space are used. We formulate the solution for learning the structure of Bayesian networks for each prospective DAG. Every sparrow is a DAG with empty arcs that symbolize a possible solution. After that, a sparrow searches the exploration zone for the best or almost the best solution, also known as the BDe score. The BDeu score, which performs as the optimization procedure's objective function, is calculated by Eq. (4). The investigation's goal is to raise the network structure's BDeu score. All first solutions are produced by iterative actions. Arcs are added consecutively to an empty graph (G0), provided that they are not already included in the graph solution. Only carry out the append operation in the event when the new solution's score function is higher than the previous score and it conforms with the DAG limitation. Until the predefined number of arcs is reached, this process is repeated. Allocating a population to each operator in the model and choosing the solution with the greatest score function is the first step in the procedure. The Sparrow algorithm iterates indefinitely, either till the all-out number of iterations is touched or until the BDeu score stops increasing.

The operations conducted in this particular domain frequently involve the substitution of a solitary edge from a rival solution, resulting in a cumulative count of four substitutions. Incorporating a relatively restricted region near the solution allows for better integration. Every movement operation induces modifications to the set of parents of the existing edges, leading to a substantial adjustment

Algorithm: The Bayesian network structure is derived from the Sparrow search optimization method.

INPUT: - benchmarks
Population size, NS
- Maximum iteration, MaxIter
- Discovery rate, Pa
- Awareness probability, Pb
- Learning probability, Pl
- Maximum velocity, Vmax
- Initial position bounds, Xmin, Xmax
OUTPUT:
- Bayesian Network Structure
OUTPUT: - learning Bayesian Network
Algorithm:
1. Initialize Sparrows randomly within the search space:
a. Randomly generate initial positions for NS sparrows.
b. Initialize velocities for each sparrow randomly within [-Vmax, Vmax].
2. Evaluate the fitness (BDe score) of each sparrow based on the Bayesian Network Structure.
3. Set the best solution as the sparrow with the highest fitness.
4. For each iteration (iter) up to MaxIter:
a. For each sparrow (i) in the population:
i. Generate random values R1 and R2.
ii. Update velocity and position using the Sparrow Search Algorithm equations:
*- Velocity update: V_i = W * V_i + Pa * R1 * (P_best - X_i) + Pb * R2 * (G_best - X_i)*
- Position update: X_i = X_i + V_i
(where W is the inertia weight, P_best is the greatest position, G_best is the global best position)
iii. Apply a random selection mechanism with probability Pl to update part of the position.
- If rand() < Pl, update a portion of the position randomly.
iv. Clamp the position within the search space [Xmin, Xmax].
v. Evaluate the fitness of the new position.
vi. If the fitness is better than the current best fitness, update the best position.
b. Update the global best position.

5. Return the Bayesian Network Structure corresponding to the best position found.

Fig. 1. SOA for structure learning Bayesian network

to the current solution. Furthermore, if the solution stays unchanged after the application of fundamental operators, the move operator possesses the capability to improve it. The frequency of escape as a sparrow approaches the intended solution exhibits an upward trend in the context of local optimization. As the sparrow swiftly moves from one solution to another in its search for a superior one, the utilization of fly directions, which entails alternating between numerous local optimization operators, becomes increasingly common. As an outcome, the present velocity is altered by employing either the optimal global or local solution of the sparrow, which is decided by the R2 value and some optimization techniques such as deletion, addition, reversion, and movement.

The fundamental idea of DP is covered in the first three operations. The SOA updates its velocity based on the sparrow's current optimum position inside the search space. Conversely, the optimal

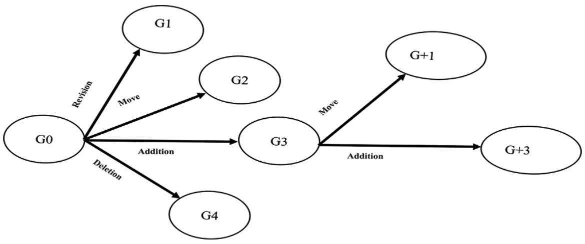


Fig. 2. Searching steps for one Sparrow (Li & Wang, 2017)

choice for sparrows inside a search zone next to a perfect location determines the speed. Fig. 2 depicts the activities of a sparrow G0, which is a model of an arc-based DAG. The sparrow tries reversal, change, addition, and omission to get new solutions G1, G2, G3, and G4. Since G3 has the finest score, the sparrow will now choose a similar strategy to go on to G+3. In the event that the BDe score of G+3 exceeds that of G+1, the appropriate operator will be performed. The iterative processes will continue until the iteration loop achieves its maximum value or the BDe score reaches a stable condition. Throughout the whole process, the sparrow selects orientations using the cognitive processes of Deletion, Addition, Movement, and Reversion.

5. Experimental Evaluation

One often-used evaluation approach for evaluating the effectiveness of SOA involves using probabilistic samples that are taken from well-established Bayesian network standards. The experimental configuration consists of a personal computer with the following characteristics: one of the system's current configurations is a Core i5 CPU running at 2.1 GHz. With an operating system of Ubuntu 14.04, the gadget has 4GB RAM. The algorithm is implemented using Java. The details for the dataset used in the experimental results are shown in Table 1.

In addition, the authors considered a few more complex datasets, such as Sonsor, Meta, Bands, Voting, Zoo, Horse, and Soybean, which include over a thousand variables (Dekker, 2009). Accuracy is defined as the number of correctly identified directed edges divided by the total number of edges in the predicted Bayesian network. The F1-score is known as the harmonic average of precision and recall. Precision measures the proportion of correctly identified directed edges out of all the edges predicted, while recall measures the proportion of correctly identified directed edges out of the total number of edges in the actual Bayesian network. The current investigation is based on the supposition that the data are stable and that the datasets used for training are stationary. Before assessing the efficacy of the SOA algorithm just on stationary collections of data, it is imperative to thoroughly assess the challenging effort of extending its applicability to encompass collections or other forms of online flow data sets. The research's authors used simulated annealing, pigeon-inspired optimization (PIO), and greedy search to compare the outcomes (Kareem & Okur, 2019). They used appropriate measurements for the datasets. We defined the parameters of the SOA algorithm and then used the same parameters to evaluate each approach. For the experiment in the field of service-oriented architecture, we utilized the numbers tmax = 1000 and N = 100 as fixed parameters for the optimization process in SOA. The parameters for the service-oriented architecture are as follows: the proportion of producers is set to 20%, and the proportion of SD accounts is set to 10%, with ST being equal to 0.8. The simulated annealing algorithm has the following parameters: the reannealing temperature is set at 500°, with a cooling factor of 0.8 and an initial temperature of 1000°. The following are the parameters for a greedy search: three thousand networks is the recommended bare

Table 1. Specification of the dataset used

Dataset name	Number of arcs/variables	Number of instances
Andes	338 arcs, 223 variables	500
Lucap02	143 variables	10,000
Win95pts	112 arcs, 76 variables	574
Hepar	123 arcs, 70 variables	350
Hailfinder	66 arcs, 56 variables	2,656
Alarm	46 arcs, 37 variables	10,000
Soybean	35 variables	307
Hepatitis	35 variables	137
Static Banjo	33 variables	320
Water	66 arcs, 32 variables	10,083
Epigenetics	30 variables	72,228
Insurance	52 arcs, 27 variables	3,000
Sensors	25 variables	5,456
Mushroom	23 variables	1,000
Parkinsons	23 variables	195
Heart	22 variables	267
Imports	22 variables	205
Child	25 arcs, 20 variables	230
Letter	17 variables	20,000
Adult	16 variables	30,162
Lucas01	10 variables	10,000
WDBC	9 variables	1,000
Asia	8 arcs, 8 variables	3,000

Table 2. Hyperparameters tuning for all methods

Algorithm	Hyperparameter	Value/Range	Description
Simulated Annealing (SA)	Reannealing temperature	(300, 700) degrees	Initial reannealing temperature range.
	Cooling factor	(0.7, 0.9)	The range for the factor by which temperature decreases.
	Initial temperature	(800, 1500) degrees	Starting temperature range for the annealing process.
Greedy Search (GS)	Minimum networks before restart	(2000, 4000) networks	Range for the minimum networks before restart.
	Minimum networks after best score	(800, 1500) networks	Range for minimum networks after obtaining best score.
	Maximum networks before restart	(4000, 6000) networks	Range for maximum networks before restart.
	Maximum parent count	(3, 7) parents	Range for maximum parent count during search.
	Restart method	Random network restart enabled	Fixed value (as randomization inherently ensures range).
	Execution time	2, 5, 10, 60 min	Multiple execution timeframes tested.
Pigeon-Inspired Optimization (PIO)	Search space dimension	D ∈ (10, 30)	Dimension of the search space.
	Population size	NP ∈ (200, 500)	Range for the number of pigeons.
	Maximum iterations (map/compass)	Nc1 max ∈ (3000, 7000)	Iteration range for map and compass operation.
	Map and compass factor	P ∈ (0.2, 0.5)	Factor range for map and compass operation.
	Maximum iterations (landmark)	Nc2 max ∈ (8000, 12000)	Iteration range for landmark operation.
FOA Algorithm	Population size	N ∈ (80, 150)	Population size range for the FOA experiments.
	AP	AP ∈ (0.25, 0.35)	Range of AP values.
	Maximum iterations	tmax ∈ (800, 1500)	Maximum iterations range.
	Sc	Sc ∈ (2.5, 4.0)	Range for Sc.
	Cc	Cc ∈ (1.5, 3.5)	Range for Cc.
	Fc	Fc ∈ (3, 5)	Range for Fc.
	Random value range	t ∈ (−1.5, 1.5)	Expanded range for random value initialization.
	Vmax	Vmax ∈ (0.08, 0.15) ub	Velocity upper boundary range.
	DP	DP ∈ (0.8, 0.9)	Range for DP.
Sparrow Optimization Algorithm (SOA)	Population size	N ∈ (80, 150)	Population size range for SOA experiments.
	Maximum iterations	tmax ∈ (800, 1500)	Maximum iterations range.
	Proportion of producers	(15%, 25%)	Range for the proportion of producers in SOA.
	Proportion of SD accounts	(8%, 12%)	Range for the proportion of SD accounts.
	ST	ST ∈ (0.75, 0.85)	Fixed range for ST value.

minimum before restarting. 1000 is the minimum number of networks that is advised once the best score has been obtained. Before restarting, a maximum of 5000 networks are advised. There is no information

DOI: 10.6977/IJoSI.202504_9(2).0002
S.W. Kareem, H.Q. Awla, etc./Int. J. Systematic Innovation, 9(2), 14-25 (2025)

Table 3. BDeu score for FOA, simulated annealing, and greedy was calculated for execution times of 2, 5, and 60 min

	Sparrow	PIO	Simulated Annealing	Greedy	Sparrow	PIO	Simulated Annealing	Greedy	Sparrow	PIO	Simulated Annealing	Greedy
Dataset	2Minutes	2Minutes	2Minutes	2Minutes	5 Minutes	5 Minutes	5 Minutes	5 Minutes	60 Minutes	60 Minutes	60 Minutes	60 Minutes
Hepatitis	-1330.4645	-1327.73	-1330.4645	-1350.16	-1170.7418	-1327.73	-1330.46	-1350.16	-1494.7584	-1327.73	-1330.46	-1350.16
Parkinsons	-1599.45	-1598.91	-1601.2968	-1732.76	-934.7074	-1598.91	-1601.3	-1721.16	-745.6308	-1598.91	-1601.3	-1700.36
Imports	-1828.9059	-1811.99	-1828.9059	-1994.15	-1186.0609	-1811.99	-1828.91	-2012.21	-1812.11	-1811.99	-1828.91	-1995.76
Heart	-2432.1878	-2423.8	-2432.1878	-2576.93	-2208.4751	-2423.8	-2423.8	-2560.43	-3019.6663	-2423.8	-2432.19	-2527.44
Mashroom	-3375.3104	-3372.51	-3375.3104	-3734.22	-1146.0213	-3372.51	-3375.31	-3706.66	-923.4354	-3372.51	-3375.31	-3588.69
WDBC	-6682.7161	-6666.04	-6682.7161	-8089.41	-5589.4145	-6666.04	-6682.72	-7954.65	-5205.4104	-6666.04	-6682.72	-7841.35
win95pts	-47085.0996	-46779.5	-47085.0996	-83749.3	-35054.3887	-46779.5	-47085.1	-83150.7	-33287.8815	-46779.5	-47085.1	-81779.5
Sensors	-60710.4985	-60343.3	-60710.4985	-69200.3	-54509.9623	-60343.3	-60710.5	-69150	-44605.6925	-60343.3	-60710.5	-68364
Hepar	-160437	-160095	-161086.4216	-169497	-160265	-160095	-161086	-169881	-160188	-160095	-161086	-168871
Letter	-178562.216	-175200	-178562.2167	-184307	-162061.888	-175200	-178562	-184916	-156726.166	-175200	-178562	-184118
Epigenetics	-179300.214	-176657	-179910.3328	-225346	-143327.197	-176657	-179300	-224172	50750.7629	-176657	-179300	-217246
Adult	-211677.716	-207809	-211677.7164	-211844	-157037.947	-207809	-211678	-211781	-201422	-207809	-211678	-211762

on the maximum number of parents for surgeries. Table 2 shows the hyperparameters for all methods. After 5 min, the system will automatically restart. Moreover, there is always the chance of a random network restart. Three different execution times for the algorithms have been tested: 2, 5, and 60. The data in Table 3 display the scores achieved by every technique in the specified datasets, along with the related time values. Upon examining the data, it is evident that the suggested method outperforms the predefined greedy search and simulated annealing algorithms in all scenarios, yielding superior score values. This demonstrates that the SOA is able to achieve the highest score in the least amount of time. We calculated the confusion matrix for each dataset and its corresponding described network structure to assess the effectiveness of structure identification. Each network has been computed with the metrics: True Negative, True Positive, False Negative, and False Positive using different algorithms.

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{11}$$

$$F1\ Score = \frac{2 * TP}{2TP + FP + FN} \tag{12}$$

$$Precision = \frac{TP}{TP + FP} \tag{13}$$

The Recall findings for FOA searching, PIO searching, Simulated Annealing, and Greedy searching are shown in Fig. 3. In most datasets, the proposed technique performs better than the PIO, Simulated Annealing, and Greedy algorithms. Similarly, Fig. 4 demonstrates that the recommended method outperforms the PIO, simulated, and greedy methods in terms of accuracy across most datasets. The proposed SOA learning algorithm has exceptional efficacy in accurately determining the appropriate structure. The iterative SOA algorithm demonstrates superior prediction accuracy compared to other algorithms across the majority of datasets.

Furthermore, in terms of construction times, the SOA technique performs better than the other approaches. F1 and the highest score from the Bayesian findings were used as criteria to estimate the model's accuracy. The F1-score, metric of precision, and recall are both used to calculate the effectiveness of the recommended strategy. In this context, accuracy is defined as the ratio of properly identified directed edges to the total number of edges in the proposed Bayesian network. By dividing the total number of edges in the network by the number of focused edges that were successfully recognized, one may determine the recall of a Bayesian network. It is widely accepted that the F1 statistic represents the harmonic mean of accuracy and recall. Fig. 5 compares the simulated annealing, PIO search, greedy search, and SOA searching. The suggested approaches are more effective than the PIO, greedy search, and simulated annealing procedures. Furthermore, accuracy is a reliable measure of the model's efficacy because its primary objective is to provide a meaningful approximation of the actual domain. In terms of Hamming distances, the suggested technique outperforms the DAG space and regularly yields values that are substantially less.

The precision measure is among the most frequently utilized metrics that should provide

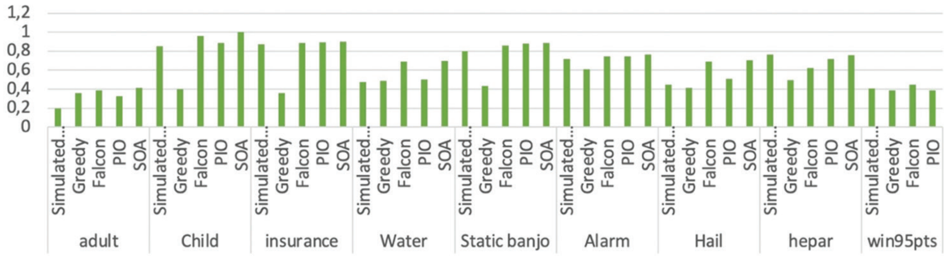


Fig. 3. Recall for SA, Greedy, Falcon, PIO, and SOA

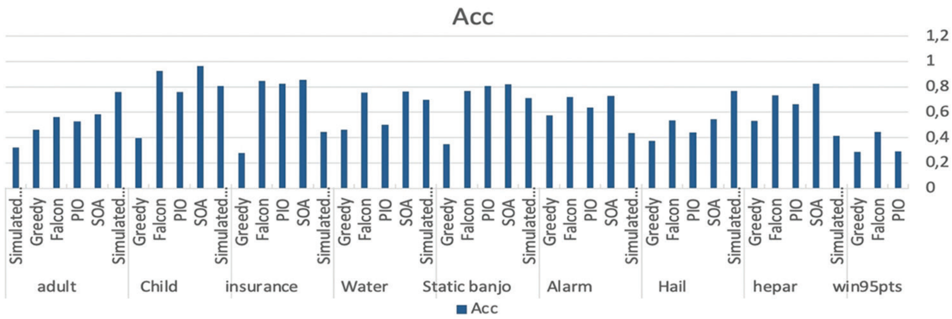


Fig. 4. Accuracy for SA, Greedy, Falcon, PIO, and SOA

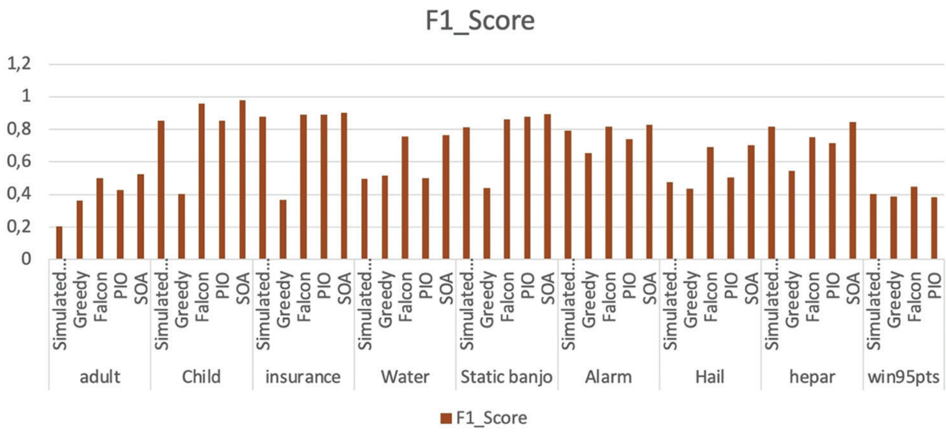


Fig. 5. F1_Score for SA, Greedy, Falcon, PIO, and SOA

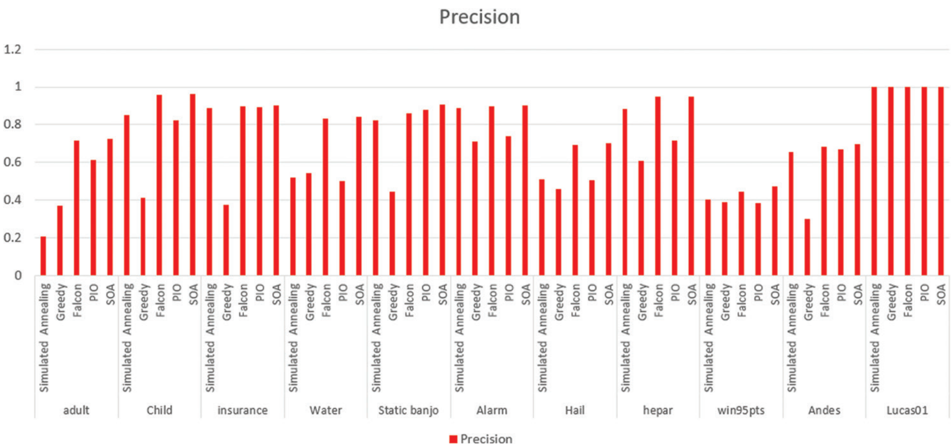


Fig. 6. Precision for SA, Greedy, Falcon, PIO, and SOA

information about the quality of learned Bayesian network structures. Comparison of different structure learning algorithms is always very clear in presented studies, with the ultimate focus on the tradeoff between exploration and exploitation of the existing models. For instance, Fig. 6 depicts the mean precision of different algorithms, which points to the fact that the proposed methodology yields better precisions in comparison to other methodologies. However, one must appreciate that it is quite difficult to get accurate measures of the actual network structure, Given the realities of real data are often complex and noisy. Nevertheless, the above improvements are currently under observation and there are continuous attempts to overcome the above-mentioned drawbacks. The standard learning algorithms are fine, but they are not without their shortcomings: they work with previously defined models and do not lend themselves well to optimizing certain kinds of probabilistic dependencies. In the future, a more detailed analysis of the presented approach can be made when incorporating other optimization methods to improve both the quality of learning the Bayesian network and its computational complexity. Further, using these algorithms for time series data that deal with real-time data could enhance the decision-making potential of the model. Another area for the work extension in the future is the use of domain-specific semantic knowledge to enhance the model and make it more precise in dependency identification. Therefore, the need to achieve deeper theoretical analysis of the convergence properties and longevity of accuracy-controlled approaches to learning seems to entail more extensive research furthering the notion of potential future development of related methods.

6. Conclusion

The authors utilized the SOA technique to address the issue of learning Bayesian network architectures. This study employs a scoring and search methodology, utilizing SOA as a search mechanism and using the BDeu metric as a scoring function. SOA is a stochastic search technique that draws inspiration from the navigational behaviors of sparrows. SOA is a flexible approach for examining distinct solution spaces that may be adjusted to different areas of application. The concentration control in SOA enables faster convergence to global optima by directing birds along logarithmic spirals toward the most favorable regions of the solution space. The suggested method showcases improved search capabilities, resulting in superior structural solutions, larger score function values, and accurate approximations of network structures. In addition, the algorithms improve the overall efficiency of global searches, resulting in rapid convergence. Subsequent research will involve assessing the extra characteristics of SOA, such as runtime analysis, resource consumption, and overall efficiency, by employing varied datasets and experimental configurations.

References

Askari, M.B.A., & Ahsaee, M.G. (2018). Bayesian Network Structure Learning Based on Cuckoo Search Algorithm. In: *6th Iranian Joint Congress on Fuzzy and Intelligent Systems* (*CFIS*). Kerman, Iran.

Campos, L.M. (2006). A scoring function for learning Bayesian networks based on mutual information and conditional independence tests. *Journal of Machine Learning Research*, 7, 2149–2187.

Cooper, G.F., & Herskovits, E. (1992). A Bayesian method for the induction of probabilistic networks from data. *Machine Learning*, 9, 309–347. https://doi.org/10.1007/BF00994110

Cowie, J., Oteniya, L., & Coles, R. (2007). *Particle Swarm Optimisation for Learning Bayesian Networks*. Engineering and Physical Sciences Research Council, United Kingdom.

Dekker, D. (2009). *Hunting Tactics of Peregrines and Other Falcons*. Wageningen University, Wageningen, The Netherlands.

Djan-Sampson, P.O., & Sahin, F. (2004). Structural Learning of Bayesian Networks from Complete Data Using the Scatter Search Documents. In: *IEEE International Conference on Systems, Man and Cybernetics*. IEEE, The Hague, Netherlands. https://doi.org/10.1109/ICSMC.2004.1400904

Fan, X., Yuan, C., & Malone, B. (2014). *Tightening Bounds for Bayesian Network Structure Learning*. Association for the Advancement of Artificial Intelligence, Washington, DC.

Fast, A.S. (2010). *Learning the Structure of Bayesian Networks with Constraint Satisfaction*. Ph.D. Thesis, Department of Computer Science, University of Massachusetts.

Fortier, N., Sheppard, J., & Pillai, K.G. (2013). Bayesian Abductive Inference Using Overlapping Swarm Intelligence. In: *IEEE Symposium on Swarm Intelligence*. IEEE, Singapore. https://doi.org/10.1109/SIS.2013.6615188

Gadekallu, T.R., & Khare, N. (2017). Cuckoo search optimized reduction and fuzzy logic classifier for heart disease and diabetes prediction. *International Journal of Fuzzy System Applications*, 6, 25–42. https://doi.org/10.4018/IJFSA.2017040102

Gandomi, A.H., Yang, X.S., Talatahari, S., & Alavi, A.H. (2013). *Metaheuristic Applications in Structures and Infrastructures*. Elsevier, USA.

He, C., & Gao, X. (2018). Structure Learning of

Bayesian Networks Based on the LARS-MMPC Ordering Search Method. In: *2018 37th Chinese Control Conference* (*CCC*). IEEE, Wuhan, China.

Heckerman, D., Geiger, D., & Chickering, D.M. (1995). Learning Bayesian networks: The combination of knowledge and statistical data. *Machine Learning*, 30, 197–243. https://doi.org/10.1023/A:1022623210503

Hedenström, A., Rosén, M., Åkesson, S., & Spina, F. (1999). Flight performance during hunting excursions in Eleonora's Falcon Falco Eleonorae. *Journal of Experimental Biology*, 202, 2029–2039. https://doi.org/10.1242/jeb.202.15.2029

Hesar, A.S. (2013). Structure learning of Bayesian belief networks using simulated annealing algorithm. *Middle-East Journal of Scientific Research*, 18, 1343–1348. https://doi.org/10.5829/idosi.mejsr.2013.18.9.12375

Ji, J., Wei, H., & Liu, C. (2012). *An Artificial Bee Colony Algorithm for Learning Bayesian Networks*. Springer-Verlag, Berlin, Heidelberg.

Kareem, S.W., & Okur, M.C. (2019). Bayesian network structure learning based on pigeon-inspired optimization. *International Journal of Advanced Trends in Computer Science and Engineering*, 8(1.2), 131–137. https://doi.org/10.30534/ijatcse/2019/2281.22019

Kareem, S.W., & Okur, M.C. (2018). Bayesian Network Structure Learning Using Hybrid Bee Optimization and Greedy Search. In: *3rd International Mediterranean Science and Engineering Congress*. Adana, Turkey.

Khanteymoori, A., Olyaee, M.H., Abbaszadeh, O., & Valian, M. (2018). A novel method for Bayesian networks structure learning based on breeding swarm algorithm. *Soft Computing*, 9, 1–12. https://doi.org/10.1007/s00500-017-2557-z

Larrañaga, P., & Poza, M. (1996). *Structure Learning of Bayesian Networks by Genetic Algorithms*. Springer-Verlag, Berlin Heidelberg.

Li, J., & Chen, J. (2014). A hybrid optimization algorithm for Bayesian network structure learning based on database. *Journal of Computers*, 9.

Li, S., & Wang, B. (2017). A Method for Hybrid Bayesian Network Structure Learning from Massive Data Using MapReduce. In: *2017 IEEE 3rd International Conference on Big Data Security on Cloud* (*Bigdatasecurity*)*, IEEE International Conference on High Performance and Smart Computing* (*HPSC*)*, and IEEE International Conference on Intelligent Data and Security* (*IDS*). IEEE, Beijing, China. https://doi.org/10.1109/BigDataSecurity.2017.42

Margaritis, D. (2003). *Learning Bayesian Network Model Structure from Data*. (Technical Report CMU). Carnegie-Mellon University, Pittsburgh, PA.

Mirjalili, S., Mirjalili, S.M., & Lewis, A. (2014). A grey wolf optimizer. *Advances in Engineering Software*, 69, 46–61. https://doi.org/10.1016/j.advengsoft.2013.12.007

Nagarajan, R., Scutari, M., & Lèbre, S. (2013). *Bayesian Networks in R with Applications in Systems Biology*. Springer, New York.

Orphanou, K., Thierens, D., & Bosman, P.A.N. (2018). Learning Bayesian Network Structures with GOMEA. In: *GECCO 2018 - Proceedings of the 2018 Genetic and Evolutionary Computation Conference*. Kyoto, Japan. https://doi.org/10.1145/3205455.3205502

Rahier, T., Marie, S., Girard, S., & Forbes, F. (2019). Fast Bayesian network structure learning using quasi-determinism screening. *HAL*, 2, 14–24.

Reddy, G.T., & Khare, N. (2016). FFBAT-optimized rule-based fuzzy logic classifier for diabetes. *International Journal of Engineering Research in Africa*, 24, 137–152. https://doi.org/10.4028/www.scientific.net/JERA.24.137

Salama, K.M., & Freitas, A.A. (2012). ABC-Miner: An ant-based Bayesian classification algorithm. Swarm Intelligence. In: *ANTS 2012. Lecture Notes in Computer Science.* Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-32650-9_2

Segundo, E.H.V., Mariani, V.C., & Coelho, L.S. (2019). Design of heat exchangers using Falcon Optimization Algorithm. *Applied Thermal Engineering*, 156, 119–144. https://doi.org/10.1016/j.applthermaleng.2019.04.038

Sencer, S., Oztemel, E., Taskin, H., & Torkul, O. (2013). Bayesian Structural Learning with Minimum Spanning Tree Algorithm. In: *The World Congress in Computer Science, Computer Engineering, and Applied Computing*.

Tucker, V.A. (1998). Gliding flight: Speed and acceleration of ideal falcons during diving and pull out. *Journal of Experimental Biology*, 201, 403–414. https://doi.org/10.1242/jeb.201.3.403

Tucker, V.A. (2000). Gliding flight: Drag and torque of a hawk and a falcon with straight and turned heads, and a lower value for the parasite drag coefficient. *Journal of Experimental Biology*, 203, 3733–3744. https://doi.org/10.1242/jeb.203.24.3733

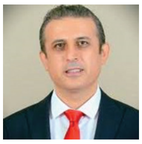
Wang, J., & Liu, S. (2018). Novel binary encoding water cycle algorithm for solving Bayesian network structures learning problem. *Knowledge-Based Systems*, 150, 95–110. https://doi.org/10.1016/j.knosys.2018.03.007

Yang, C., Ji, J., Liu, J., Liu, J., & Yin, B. (2016). Structural learning of Bayesian networks by bacterial foraging optimization. *International Journal of Approximate Reasoning*, 69, 147–167. https://doi.org/10.1016/j.ijar.2015.11.003

Yuan, C., Malone, B., & Wu, X. (2011). Learning optimal Bayesian networks using A* search. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence* (*IJCAI*). Barcelona.

Zhang, S.Z., & Liu, L. (2008). MCMC Samples Selecting for Online Bayesian Network Structure Learning. In: *International Conference on Machine Learning and Cybernetics*. IEEE, Kunming. https://doi.org/10.1109/ICMLC.2008.4620690

AUTHOR BIOGRAPHIES



Amin Salih Mohammed, Professor in the Department of Software and Informatics Engineering at Salahaddin University-Erbil, College of Engineering, is a distinguished academic with over 17 years of teaching and research experience. He is an IEEE Senior Member and an active researcher, frequently serving as a resource person for workshops and faculty development programs organized by international institutions. Amin completed his B.Sc., M.Sc., and Ph.D. in Computer Engineering from Kharkiv National University of Radio Electronics, Kharkiv, Ukraine. His research focuses on computer networks, wireless networks, and cloud computing. A prolific author, he has published more than 50 research articles in reputed international journals indexed in SCI and Scopus databases, as well as in prestigious IEEE conferences. His expertise spans software engineering, artificial intelligence, machine learning, data science, and cybersecurity, with a strong emphasis on creating secure and intelligent systems. As an educator, Professor Mohammed has mentored numerous students, integrating theoretical and practical knowledge to prepare them for successful careers.

Shahab Wahhab Kareem, his BSc in Control and Computer Engineering from the University of Technology Baghdad in 2001, MSc in Software Engineering from Salahadeen University in 2009, and Ph.D. in Yasar University Izmir, Turkey in 2020. My research interests include Machine learning and BIG DATA. I'm a Lecturer at the Information System Eng. (ISE) Department (2011-till now), Erbil Polytechnic University, Iraq (shahab.kareem@epu.edu.iq). His expertise spans multiple domains, including cybersecurity, big data, IoT security, and artificial intelligence. He has made significant contributions to advancing research in these areas, focusing on innovative approaches to solving real-world challenges in network security and data analytics. Shahab's research interests encompass developing cutting-edge solutions for intrusion detection, botnet detection, and secure data management. His work leverages advanced techniques such as ensemble learning, federated learning, deep learning, and big data analytics. He has a particular focus on integrating privacy preserving technologies with scalable frameworks to enhance IoT security and improve detection systems in large-scale distributed environments.

Hoshang Qasim Awla earned his B.Sc. in Computer Science from Soran University, Iraq, in 2012 and his M.Sc. in Computer Engineering from Hasan Kalyoncu University, Turkey, in 2016. He recently completed his Ph.D. in Data Science at Soran University in 2024. With a strong academic background and a passion for research, Hoshang has made significant contributions to the fields of data science and computer engineering. His research interests include data analytics, machine learning, artificial intelligence, and big data systems. He has published numerous high-impact research articles in renowned SCI-indexed journals and IEEE conferences, establishing himself as a prolific researcher in his field. Hoshang is also actively involved in academic and professional development initiatives, participating in international workshops and conferences to disseminate his findings and collaborate with experts worldwide.