AIMS Mathematics, 9(3): 7131–7151.
DOI: 10.3934/math.2024348
Received: 01 December 2023
Revised: 05 January 2024
Accepted: 18 January 2024
Published: 19 February 2024

https://www.aimspress.com/journal/Math

Research article

Hybrid arithmetic optimization algorithm with deep learning model for secure Unmanned Aerial Vehicle networks

Sultanah M. Alshammari[1,2], Nofe A. Alganmi[1], Mohammed H. Ba-Aoum[3,4], Sami Saeed Binyamin[5], Abdullah AL-Malaise AL-Ghamdi[2,6,7] and Mahmoud Ragab[2,8,*]

[1] Department of Computer Science, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[2] Center of Excellence in Smart Environment Research, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[3] Industrial and Systems Engineering Department, Virginia Polytechnic Institute and State University, Blacksburg 24061, VA, USA
[4] Systems Engineering Department, King Fahd University of Petroleum and Minerals, Dhahran 31261, Saudi Arabia
[5] Computer and Information Technology Department, The Applied College, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[6] Information Systems Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia
[7] Information Systems Department, School of Engineering, Computing and Design, Dar Al-Hekma University, Jeddah 22246, Saudi Arabia
[8] Information Technology Department, Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah 21589, Saudi Arabia

* **Correspondence:** Email: mragab@kau.edu.sa.

Abstract: Securing Unmanned Aerial Vehicle (UAV) systems is vital to safeguard the processes involved in operating the drones. This involves the execution of robust communication encryption processes to defend the data exchanged between the UAVs and ground control stations. Intrusion detection, powered by Deep Learning (DL) techniques such as Convolutional Neural Networks (CNN), allows the classification and identification of potential attacks or illegal objects in the operational region of the drone, thus distinguishing them from the routine basics. The current research work offers a new Hybrid Arithmetic Optimizer Algorithm with DL method for Secure Unmanned Aerial Vehicle

Network (HAOADL-UAVN) model. The purpose of the proposed HAOADL-UAVN technique is to secure the communication that occurs in UAV networks via threat detection. At the primary level, the network data is normalized through min-max normalization approach in order to scale the input dataset into a useful format. The HAOA is used to select a set of optimal features. Next, the security is attained via Deep Belief Network Autoencoder (DBN-AE)-based threat detection. At last, the hyperparameter choice of the DBN-AE method is implemented using the Seagull Optimization Algorithm (SOA). A huge array of simulations was conducted using the benchmark datasets to demonstrate the improved performance of the proposed HAOADL-UAVN algorithm. The comprehensive results underline the supremacy of the HAOADL-UAVN methodology under distinct evaluation metrics.

Keywords: Unmanned Aerial Vehicle; arithmetic optimization algorithm; feature selection; autoencoder; deep learning
Mathematics Subject Classification: 68M11, 68M25, 68T07, 68W1

1. Introduction

Unmanned Aerial Vehicle (UAV) networks provide distinct benefits in smart-city infrastructure, which makes it a crucial element for expensive programs [1]. These vehicles help in real-time monitoring and surveillance abilities, thus allowing the authorities to collect vital information from the targeted sources and locations. Further, the UAVs have confirmed advantages in some sort of situations such as monitoring the visitors in a live location, ecological analysis as well as disaster responses for protecting the public [2]. While the UAVs are progressively getting established in smart cities, the privacy and safety concerns also increase in parallel [3]. Privacy problems followed by intrusion detection and prevention have become essential, since the UAVs collect and transmit a massive quantity of data. The extensive incorporation of the UAVs in smart cities has become an innovative dimension for both public protection as well as urban management [4]. These multipurpose aircrafts possess real-time information-gathering abilities that can be leveraged in different fields from traffic monitoring to disaster response [5]. However, the development of the UAVs still poses a serious challenge i.e., to assure the privacy and security of the gathered and transferred data.

Intrusion is the most significant problem that raise suspicions upon the UAV networks since it has deleterious effects on data integrity and public security [6]. When the UAVs are developed targeting crucial and significant applications, then it becomes predominant to protect their functions against possible attacks. Furthermore, the sensitive features of the data demand strong privacy-preserving methods [7]. In the studies conducted earlier, the researchers have made significant developments in the domain of UAV intrusion detection. A few data-driven approaches have been employed in domains with higher security coefficients such as industrial and military sectors since these sectors need higher efficiency and accuracy. The Machine Learning (ML) techniques can accomplish superior effectiveness compared to non-ML algorithms because of the former's learning and training components [8]. Therefore, a cost-effective and an efficient IDS is extremely required to ensure the security of drone networks. Numerous researchers have developed Artificial Intelligence (AI) methods, comprising ML, Deep Learning (DL), and other approaches [9]. Both DL and ML models remain the most commonly used methodologies in network security, since they are capable of learning the valuable features in network traffic and identify both abnormal and normal activities depending on its

learning ability. The ML approaches learn valuable insights from network traffic data by primarily relying upon feature engineering [10]. Simultaneously, the DL methods are also based on feature engineering and are precise in automatically learning the intricate features from complete data thanks to its deep architecture.

The current research article designs an innovative Hybrid Arithmetic Optimization Algorithm with DL Model for Secure Unmanned Aerial Vehicle Networks (HAOADL-UAVN) technique. The purpose of the proposed HAOADL-UAVN technique is to secure the communication in UAV networks via threat detection. At the primary level, the network data is normalized through min-max normalization approach so as to scale the input data into a useful design. The HAOA is used to select an optimal set of features. Next, the security is attained via Deep Belief Network Autoencoder (DBN-AE)-based threat detection. At last, the selection of the hyperparameters for the DBN-AE method is accomplished using the Seagull Optimization Algorithm (SOA). A huge array of simulations was conducted using the benchmark datasets in order to demonstrate the improved performance of the HAOADL-UAVN model. The key contributions of the current study are listed herewith.

- A new HAOADL-UAVN technique is developed to safeguard the communication that occurs in UAV systems by addressing the communication tasks and applying robust encryption processes.
- The HAOADL-UAVN model incorporates the HAOA for optimum feature selection, thus improving the efficacy and significance of the dataset for threat recognition.
- The DBN-AE model is used for threat detection in which DL models like CNNs are leveraged to recognize and categorize the potential attacks or illegal objects in UAV operational region.
- The SOA is incorporated for hyperparameter tuning of the DBN-AE model, thus refining the complete performance and flexibility of the threat detection method.

2. Related works

Kateb and Ragab [11] developed the Archimedes Optimizer with DL-based Aerial Image Classification and Intrusion Detection (AODL-AICID) method that had two main processes. This technique was inclusive of Backpropagation Neural Network (BPNN)-based classifier, Archimedes Optimizer Algorithm (AOA)-based hyperparameter optimizer, and MobileNetv2 feature extraction. Furthermore, the AODLAICID algorithm implemented a stacked bi-directional-LSTM (SBi-LSTM) architecture. In the last stage, Nadam optimization technique was employed for hyperparameter tuning. In the study conducted earlier [12], an architecture was proposed to protect the UAVs against malicious attackers and improve the rogue UAVs. The model, introduced in the study, implemented a dynamic theoretical grid-based layout in real geographical utilization. Public key cryptographic techniques were utilized to secure the communication connection. Neural Network (NN)-based forecasting was employed in this study to recognize the abnormalities. Principal Component Analysis (PCA) based on multi-variable statistical evaluation was performed to recognize the outliers in aerial network infrastructure. The authors [13] developed the Deep Reinforcement Learning (DRL) method. This method discussed about the possible applications and architecture of the UAVs. Further, intrusion attacks were considered in UAV aerial computing networks. Then, the DRL-empowered intrusion detection process was executed to safeguard the security facilities. At last, the method arrived at the decision through numerous valuable analytical methods.

Masadeh et al. [14] designed an autonomous UAV model by redeveloping the optimization issues through Markov-decision technique with the deployment of RL techniques. Afterwards, the RL-based method was applied to resolve the maximization issue of the developed value. Particularly, various model variations of the RL-based technique were employed in this study to have various exploration methods and temporal variance approaches. In literature [15], a Secured Privacy-Preserving Framework (SP2F) architecture was designing including two key engines. In this two-stage privacy engine, BC and smart contract-based enhanced Proof of Work (ePoW) helped in alleviating the data poisoning attacks which can further be developed for data authentication purpose. Sparse-AE (SAE) was implemented in this study to convert the data into novel encryption formats so as to avoid the intrusion attacks. In anomaly detection engine, the SLSTM was utilized for both evaluation and training the outcomes. Lastly, a comparative analysis was conducted. The authors [16] introduced the Quantum Dwarf Mongoose Optimizer with an Ensemble-DL-based ID (QDMO-EDLID) system in the CPS infrastructure. For the purpose of subset Feature Selection (FS), the QDMO-EDLID method utilized the QDMO technique. Additionally, Deep-AE (DAE), Deep Belief Networks (DBN), and an ensemble of Convolution Residual Networks (CRN) algorithms were also incorporated in this study for intrusion classification method.

In the study conducted earlier [17], the authors intended to overcome the security shortage and suggested an experience-based DL method to provide the DDoS, DoS and other distinct types of attacks cover the ping-of-death attacks. The presented method employed the concept of IDS. Additionally, a nature-inspired control routing protocol called AntHocNet was evaluated with the rest of the algorithms for dependable communication. The authors [18] suggested an autonomous IDS that could proficiently identify the malicious attacks which invade the UAVs, with the help of the DCNN (UAV-IDS-DCNN) approach. In particular, this developed model deliberated the encoded Wi-Fi traffic data records with three categories of frequently-utilized UAVs namely, DBPower UDI UAVs, DJI Spark UAVs and Parrot Bebop UAVs.

In literature, the authors [19] developed the new optimum Squeezenet model with Deep Neural Network (OSQNDNNs) technique for aerial image classification in the UAV systems. The presented OSQN-DNN method originally allowed the UAVs to seize the images using inbuilt imaging sensors. Also, the OSQN method was used as a feature extractor to originate a beneficial set of feature vectors. On the other hand, the Coyote Optimizer Algorithm (COA) was used to optimally pick the hyperparameters, tangled in the traditional SqueezeNet method. In the study conducted earlier [20], the issue of adversarial attacks upon the DL-based UAVs was discussed and dual adversarial attack models were developed against the regression methods in the UAVs. Sangeetha Francelin et al. [21] planned a new approach for introducing protected communication in the UAV Network (UAVN). At first, the UAVN was simulated following which the data transmission was executed among the nodes utilizing a routing track; hence, an optimum routing path was identified using the newly-invented Tunicate Swarm Political Optimization (TSPO) system. In literature [22], the authors developed a method to appreciate the effectual and secure transmission of mobile users' data, when organizing Mobile Edge Computing (MEC) servers with AI so as to help in processing the data on UAVs.

3. The proposed method

In the current study, the design and development of the HAOADL-UAVN technique are focused. The purpose of the proposed HAOADL-UAVN technique is to secure the communication in the UAV

networks via threat detection. The HAOADL-UAVN approach comprises of four main processes namely min-max normalization approach, HAOA-based FS process, DBN-AE-based classification, and SOA-based hyperparameter tuning. Fig. 1 represents the entire process involved in the proposed HAOADL-UAVN method.

3.1. Data normalization

Initially, the network data is normalized through min-max normalization approach in which the input data is scaled up into a useful set-up. Normalization is an important stage that compares the data from different fields [23]. The process of normalization upgrades the data from a given domain to a range of 0 and 1.

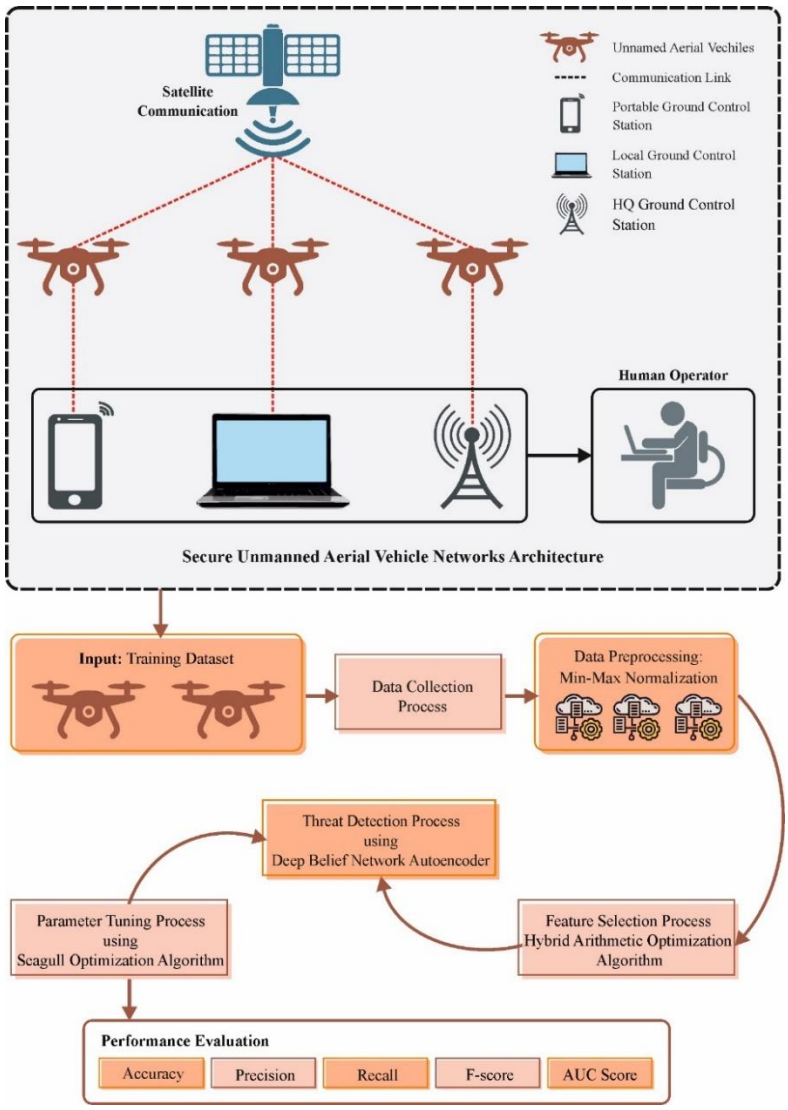

Figure 1. Overall process of the HAOADL-UAVN technique.

Norm normalization Z-score, decimal scaling, median-mad, min-max, and mean-mad techniques are generally used in the normalization of data. In the current study, the min-max normalization

technique is used to rescale the properties from its domain to achieve novel values in the range of [0,1]:

$$X_{normalized} = \frac{X - X_{min}}{X_{max} - X_{min}}, \tag{1}$$

where X implies the input feature value and $X_{normalized}$ signifies the normalized features. The higher and lower sets of the input feature are denoted by X_{max} and X_{min} correspondingly.

3.2. HAOA-based feature selection

For the feature selection process, the HAOA has been employed in this study. Each meta-heuristic method shares a common trait whereas the optimization process is separated into two phases (which are together named as the 'searching step') such as the exploitation and exploration phases [24]. Between these, the exploration step is required to seek the uncharted portions of the searching space whereas the exploitation step is in control of engaging the regions that have been already observed.

AOA starts the optimization procedure using an arbitrarily-created solution space (as matrix X, but x_{ij} is a single performance, $1 \le i \le N, 1 \le j \le n$) whereas a better candidate outcome, previous best performer, from all the iterations is assumed. Addition, division, multiplication, and subtraction processes finalize the places with near-optimal performance besides the iteration trajectories. Primarily, the AOA elects the searching step by evaluating a co-efficient named Math Optimizer Accelerated (MOA) function, which is utilized in both exploitation as well as exploration steps:

$$MOA(C_{Iter}) = \text{Min} + C_{Iter} \times \left(\frac{\text{Max} - \text{Min}}{M\_Iter}\right). \tag{2}$$

Here, $MOA(C\_Iter)$ represents the function values at $t^{he\,tth}$ iteration, C_Iter denotes the existing iteration spanning between one and the maximal iteration count, M_Iter. The minimal and maximal outcomes of the enhanced function are represented by Min and Max correspondingly.

Division (D) and Multiplication (M) operators were utilized from the exploration step. However, it arbitrarily surveyed the searching region from various sectors with a purpose to attain the optimum performance. These approaches are defined in Eq (3) and the searching stage is forced by the MOA function with a random integer, $r1 > MOA$. In this step, the alternative operator (M)cannot be considered until the 1st operator (D) finishes the existing task, controlled by a random integer, $r2 < 0.5$. Then, the 2nd operator (M) is utilized from the location D to complete the existing task.

$$x_{i,j}\,(C_{Iter} + 1) = \begin{cases} best(x_j) \div (MOP + \varepsilon) \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & r2 < 0.5 \\ best(x_j) \times MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), & otherwise \end{cases}. \tag{3}$$

Here, ε refers to any smaller integer, μ denotes the set control parameter, $x_{i,j}(C\_Iter + 1)$ corresponds to the i^{th} performance in the next iteration, $x_{i,j}(C\_Iter)$ defines the j^{th} position of the i^{th} performance in the existing iteration and $best(x_j)$ denotes the j^{th} position of the existing best outcome. LB_j and UB_j imply the lower and upper boundaries of the j^{th} position, correspondingly.

$$MOP(C_{Iter}) = 1 - \frac{C_{Iter}^{\frac{1}{\alpha}}}{M\_Iter^{\frac{1}{\alpha}}} \tag{4}$$

Here, Math Optimizer Probability MOP (C_Iter) implies the function values during the t^{th} iteration, C_Iter indicates the existing iteration, M_Iter stands for the maximal iteration count and α defines the set parameter that is well known to track the exploitation accuracy with iterations.

Then, the search space is utilized by leading a deep search using Subtraction (S) and Addition (A) searching strategies. In this study, the alternative operator (A) cannot be reserved for consideration until the time, the 1st operator (S) finishes its existing task (1st rule in Eq (5)), controlled by the arbitrary integer, $r3 < 0.5$. Otherwise, the 2nd operator (A) is utilized from the location S to complete the existing task.

$$x_{i,j}(C_{Iter}+1) = \begin{cases} best\left(x_j\right) - MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), r3 < 0.5 \\ best\left(x_j\right) + MOP \times \left(\left(UB_j - LB_j\right) \times \mu + LB_j\right), otherwise \end{cases}. \tag{5}$$

To reiterate, once $r1 > MOA$, the purpose of the candidate's outcomes differs from the near-optimal outcome. Once the $r1 < MOA$, it inclines to converge at the near-optimal performance. The parameter MOA heavily increases from 0.2–0.9 so as to promote the exploitation and exploration phases. It is also important to note that the computation complexity of AOA stands at $O\left(N \times (ML+1)\right)$.

The current study involves the hybridization of AOA and ABC techniques since the latter can overcome the inadequate exploration ability of the former. In case of ABC algorithm, if the utilized bee cannot enhance the food source, then the source must be abandoned and the bee is converted into a scout. This outcome can be obtained using the limited control variable that defines a solution to be eliminated. Based on this particular process, the ABC algorithm accomplishes strong exploration whereas the exploration strength of the AOA technique gets enhanced on a similar mechanism that replaces the depleted solutions. Each solution that remained the same, even after so many rounds of iteration, is substituted by the pseudo-random solution. The HAOA approach exploits a similar limit control variable. Each performance in the hybrid mechanism is prolonged using a single feature named trial. These trial values are incremented, if the solution does not get improve during a certain round.

Fitness Function (FF) assumes the number of features elected and the classification outcomes. It reduces the fixed dimension of the designated features and improves the classification accuracy. Thus, the FF is deployed in this study to estimate the specific solutions as given below.

$$Fitness = \alpha \times ErrorRate + (1-\alpha) \times \frac{\#SF}{\#All\_F}. \tag{6}$$

In Eq (6), $ErrorRate$ denotes the classifier error rate based on the selected features. $ErrorRate$ is estimated as a percentage of improper classification to the number of classifications within [0,1]. $\#SF$ represents the number of features chosen whereas $\#All\_F$ shows the overall number of features from the original dataset. α controls the import of classification quality and subset length, where the value for α is fixed to be 0.9.

3.3. Threat detection using classification model

In this stage, security is attained via DBN-AE-based threat detection process. DBN is a combination of unsupervised networks namely, Restricted Boltzmann Machines (RBM) that perform a visible layer of the next layer as well as the Hidden Layer (HL) of all the sub-networks [25]. DBN architecture has an effective layer-wise process that defines the dependence of the variable in the above layer.

The DBN model performs logistic regression for classification using various hidden and visible RBM layers. At the beginning, the dissimilar feature spaces of the vector are mapped after which the RBM layer, trained in an unsupervised manner is retained with the feature dataset. Next, a fine adjustment is carried on. At last, the resultant feature vector of the RBM layer is considered as the input feature vector for the subsequent RBM layer.

AE is a simple and three-layered, unsupervised NN that is used for representative learning such as size reduction or FS and helps in rebuilding the input pattern in the outcome layer.

Like the symmetrical structure, the input and output layers of the AE model are also similar in size. Compared to the visible layer, the hidden layer has a fewer number of neurons in the network model. Using less number of neurons, attempts are made to represent or encode the input in its compact form, which captures the relevant features of the input vector.

Here, the AE can be trained using a BP model as in FFNN-based technique on MSE loss function. The training model includes coding and decoding stages. During the coding stage, the input is encoded in a hidden depiction based on the principles of lower half layer. During the decoding stage, a similar input is used as an attempt to reconstruct from coding representation by applying the weight conditions of the upper half layer.

Assume X denotes the data with n samples and m number of features. Y refers to the encoder. The accurate representation of the encoding and decoding layers for AE is shown in Eqs (7) and (8), correspondingly.

$$Y = f(wX + b), \tag{7}$$

$$\acute{X} = g(\acute{w}Y + b). \tag{8}$$

Here w and b denote the adjustable parameters, the activation functions are f and g, the recreated input vector in the output layer is denoted by $\acute{X}$, and the transposed weight $(\acute{X})$ is denoted by W.

AE training process is inclusive of finding the w and b parameters that reduce the error between the input data X and the reconstruction data, $\acute{X}$.

The DBN-AE structure has two most important parts while the AE can be utilized as a DL algorithm for feature extraction process.

The encoding part of the AE is used to extract the feature, which signifies the characteristics of the input data. The extracted feature is then fed into the DBN for detection. Before training the DBN detection model, the AE is separately trained to attain the weight matrices. The decoding part of the AE is utilized to verify the features extracted so as to recreate the new data. Next, the attained weight matrices for AE are integrated with the DBN mechanism and are lastly trained using the input datasets used for detection. Figure 2 illustrates the infrastructure of the DBN-AE model.

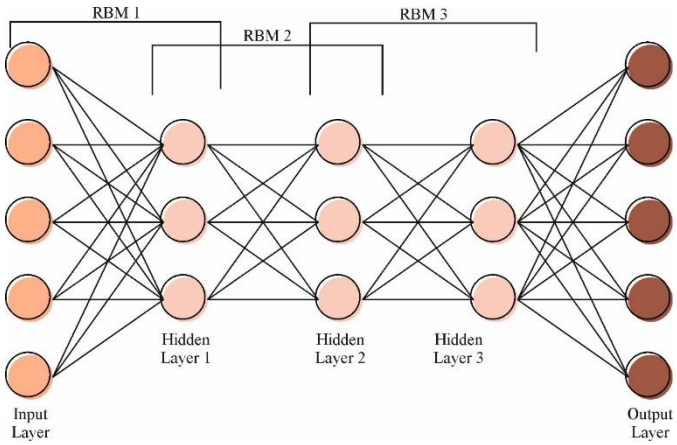

Figure 2. Architecture of the DBN-AE model.

3.4. Hyperparameter tuning

Eventually, the hyperparameters for the DBN-AE algorithm are selected with the help of SOA. The hyperparameters chosen by the SOA include learning rate, batch size, and the number of epochs. SOA is a novel nature-inspired optimization algorithm that gains its inspiration from the aggressive migration behavior of the seagulls [26]. SOA has better search performance, simple in nature and easy to implement compared to the rest of the conventional techniques namely GA and PSO. Every individual seagull exemplifies the individual search entity in the search range and so the location of the seagull denotes a promising outcome to the optimization problems. The implementation of the SOA method is as follows.

Migration behavior (global search): The seagulls move around from one location to another during the seagull migration period. This migration behavior must fulfill three conditions as given below:

Avoiding collision. In order to prevent collision among the neighboring seagulls, an additional parameter A is established so as to update the seagull's position during the iterative computation:

$$\vec{N}_s = A \cdot \overrightarrow{P_s}(i). \tag{9}$$

In Eq (9), the new position is represented by $\vec{N}$; the existing position of the seagull is indicated by $\overrightarrow{P_s}(i)$; the existing iteration count is i and the seagull's movement in the given search range is A. The computation process is given below.

$$A = f_c - i \cdot \left(\frac{f_c}{\mathrm{Max}_{iteration}}\right). \tag{10}$$

In Eq (10), the control frequency of the parameter A is f_c that lies in the range of $[0, f_c]$. Usually, f_c is fixed as 2; $i = 0,1,2 \cdots \mathrm{Max}_{iteration}$ where i implies the existing iteration count and $\mathrm{Max}_{iteration}$ indicates the maximal iteration count.

Moving towards the best neighbor: The seagulls move to a better neighbor thus avoid any collisions between the neighboring seagulls.

$$\vec{B}_S = B \cdot \left(\vec{P}_{gS}(i) - \vec{P_s}(i)\right). \tag{11}$$

In Eq (11), $\vec{B_S}$ denotes the direction in which the seagull moves from $\vec{P_s}(i)$ original location to $\vec{P}_{gS}(i)$, the best neighboring location. B is a random variable that balances the global and local search as given below.

$$B = 2 \cdot A^2 \cdot rd. \tag{12}$$

Here, rd is a randomly generated number within [0,1].

Moving towards the optimum location. The seagull updates its location, according to the optimal location.

$$\vec{D}_s = \left|\vec{N}_s(i) + \vec{B_s}(i)\right|. \tag{13}$$

In Eq (13), the distance between the existing and the global optimal location is represented by $\vec{D}_s$.

Aggressive behavior (local search): Once the seagull attacks its prey during flight, then it develops a spiral formation from the air. Such behavior is detailed through the Eqs (14)–(17).

$$x = r \cdot \sin k, \tag{14}$$

$$y = r \cdot \cos k, \tag{15}$$

$$z = r \cdot k, \tag{16}$$

$$r = u \cdot e^{k \cdot v}, \tag{17}$$

Here, the spiral circle designed by the seagull swarm is r; a random integer in the interval $[0,2\pi]$ is k; the constants v and u define the spiral shape; and the base of the natural logarithm is represented by e.

$$\vec{P_s}(i) = x \cdot y \cdot z \cdot \vec{D}_s + \vec{P}_{gS}(i). \tag{18}$$

In Eq (18), the attack location of the seagull is denoted by $\vec{P_s}(i)$ which represents the finally updated location.

The SOA approach derives the FF to obtain the effective classification outcomes. It explains a positive integer to describe the high accuracy of the resultant candidate. At this point, the reduction in classifier error rates is specified as the FF.

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{No.of\ misclassified\ samples}{Total\ No.of\ samples} \times 100 \tag{19}$$

4. Performance validation

The performance of the proposed HAOADL-UAVN system was examined using dual datasets [27] i.e., NSL-KDD and TON-IoT datasets. The NSL-KDD dataset has different classes such as Unauthorized Access to Root (U2R), Unauthorized Access from a Remote Machine (R2L), Surveillance and Probing (Probe), Denial of Service (DoS), and normal. Next, the TON_IoT dataset

encompasses several classes such as backdoor, Distributed Denial of Service (DDoS), DoS, injection attack, Man-in-the-Middle (MITM), Password, Ransomware, Scanning, Cross-Site Scripting (XSS), and benign.

Figure 3 showcases the classification outcomes of the HAOADL-UAVN technique, when using the NSL-KDD database. Figures 3a and b describe the confusion matrices delivered by the HAOADL-UAVN algorithm on 80:20 of the TRPH/TSPH. The outcomes infer that the HAOADL-UAVN approach identified and categorized all the five class labels with high accuracy. Equally, Figure 3c showcases the PR examination outcomes achieved by the HAOADL-UAVN technique. The experimental values indicate that the HAOADL-UAVN approach yielded better PR values for all the classes. Moreover, Figure 3d exhibits the ROC examination results attained by the HAOADL-UAVN approach. The simulation outcomes characterize that the HAOADL-UAVN methodology produced adept results with the highest ROC outcomes for diverse classes.

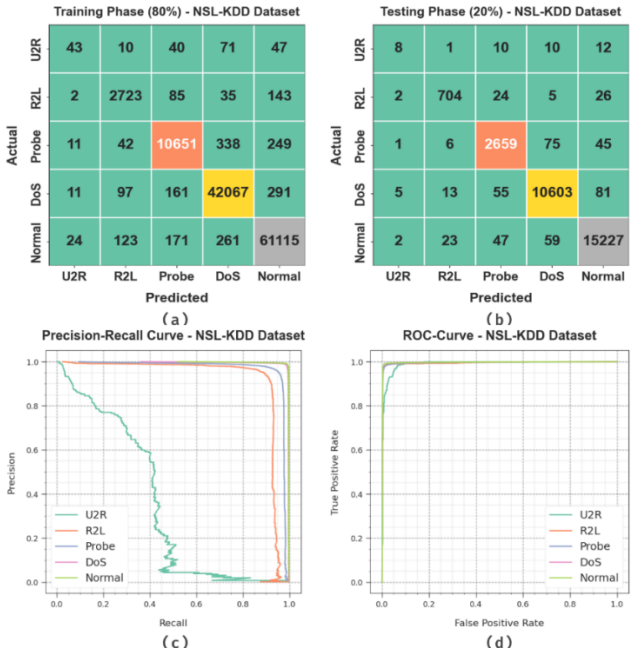

Figure 3. (a and b) Confusion matrices and (c and d) PR and ROC curves of the NSL-KDD database.

The intrusion detection analysis was conducted upon HAOADL-UAVN algorithm using the NSL-KDD database and the results are shown in Table 1 and Figure 4. The simulation outcomes imply that the HAOADL-UAVN system properly detected the intrusions. With 80% TRPH, the HAOADL-UAVN methodology achieved an average $accu_y$ of 99.26%, $prec_n$ of 86.25%, $reca_l$ of 80.72%, F_{score} of 82.41% and an AUC_{score} of 90.07%. Additionally, with 20% TSPH, the HAOADL-UAVN model delivered an average $accu_y$ of 99.32%, $prec_n$ of 86.27%, $reca_l$ of 81.04%, F_{score} of 82.68% and an AUC_{score} of 90.26%, respectively.

Table 1. Intrusion detection outcomes of the HAOADL-UAVN algorithm on NSL-KDD database.

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	AUC_{Score}
TRPH (80%)					
U2R	99.82	47.25	20.38	28.48	60.17
R2L	99.55	90.92	91.13	91.02	95.45
Probe	99.08	95.89	94.33	95.10	96.95
DoS	98.94	98.35	98.69	98.52	98.88
Normal	98.90	98.82	99.06	98.94	98.89
Average	99.26	86.25	80.72	82.41	90.07
TSPH (20%)					
U2R	99.86	44.44	19.51	27.12	59.74
R2L	99.66	94.24	92.51	93.37	96.18
Probe	99.11	95.13	95.44	95.29	97.47
DoS	98.98	98.61	98.57	98.59	98.89
Normal	99.01	98.93	99.15	99.04	99.00
Average	99.32	86.27	81.04	82.68	90.26

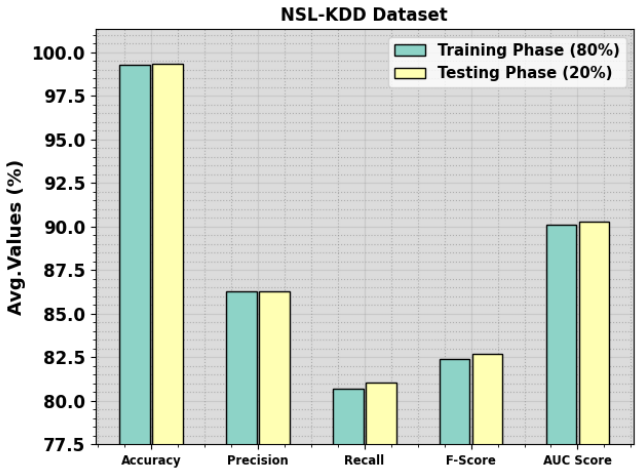

Figure 4. Average values of the HAOADL-UAVN algorithm on NSL-KDD database.

Figure 5 shows the $accu_y$ curves for training (TR) and validation (VL) datasets, plotted by the HAOADL-UAVN algorithm upon the NSL-KDD database. The figure provides valuable insights about the effectiveness of the method under several epochs. Mainly, this outcome denotes a reliable development of the TR and TS $accu_y$ values based on the increasing number of epochs, thus establishing the capability of the model to absorb and recognize the designs from both TR and TS datasets. The increasing trends in TS $accu_y$ underline the flexibility of the model upon the TR database, its capability to make precise estimates on the hidden data, the ability to emphasize robust generalization.

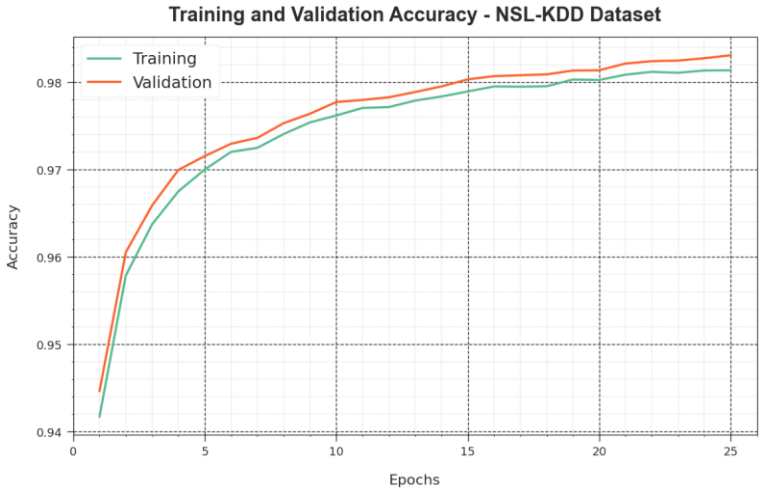

Figure 5. $Accu_y$ curve of the HAOADL-UAVN algorithm on NSL-KDD database.

Figure 6 illustrates the wide-ranging results of TR and TS loss values attained by the HAOADL-UAVN algorithm upon the NSL-KDD database across many epochs. The TR loss got gradually declined as the model improved its ability to minimize the classification faults in the database. The loss curves reveal the model's arrangement with the TR data, thus establishing its capability to capture the designs in an efficient manner. It is significant to note that the parameters in the HAOADL-UAVN algorithm got continuously altered in order to minimize the differences between the estimates and definite TR labels.

Figure 6. Loss curve of the HAOADL-UAVN algorithm on the NSL-KDD database.

The comparison study of the HAOADL-UAVN methodology and other models was conducted upon the NSL-KDD database and the results are portrayed in Table 2 and Figure 7 [28,29]. The results indicate the ineffectual performance of the LR model. Next to that, the KNN, CNN, and DCA models exhibited a moderate performance. Although the LSTM_RNN model accomplished reasonable results, the HAOADL-UAVN technique achieved the maximum outcomes with an $accu_y$ of 99.32%, $prec_n$ of 86.27%, $reca_l$ of 81.04%, and an F_{score} of 82.68%.

Table 2. Comparative analysis outcomes of the HAOADL-UAVN method and other models upon the NSL-KDD database.

NSL-KDD Database				
Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
HAOADL-UAVN	99.32	86.27	81.04	82.68
LSTM_RNN Model	95.91	83.37	76.75	76.54
Logistic Regression	78.60	85.54	77.99	76.72
KNN Model	82.18	73.84	79.12	76.60
CNN Model	86.54	84.04	78.03	76.91
DCA Model	89.77	85.2	79.87	78.37

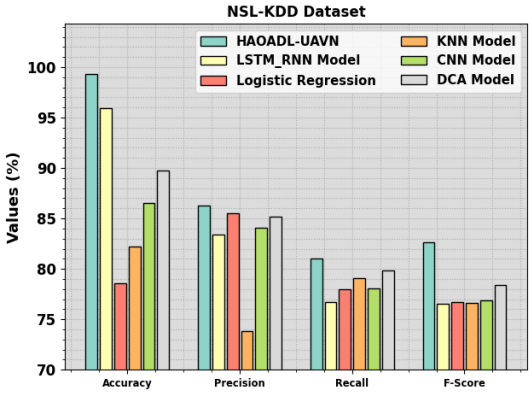

Figure 7. Comparative analysis outcomes of the HAOADL-UAVN methodology under NSL-KDD database.

Figure 8 shows the classification outcomes of the HAOADL-UAVN approach for the TON_IoT dataset. Figures 8a and b show the confusion matrices obtained by the HAOADL-UAVN technique on 80:20 TRPH/TSPH. The experimental values infer that the HAOADL-UAVN technique recognized and characterized all the ten classes. Likewise, Figure 8c shows the PR study outcomes achieved by the HAOADL-UAVN technique. The result is definite that the HAOADL-UAVN method produced the maximum PR values for all the classes. Furthermore, Figure 8d displays the ROC examination results of the HAOADL-UAVN methodology. The results reveal that the HAOADL-UAVN method caused adept consequences with peak ROC values under separate classes.

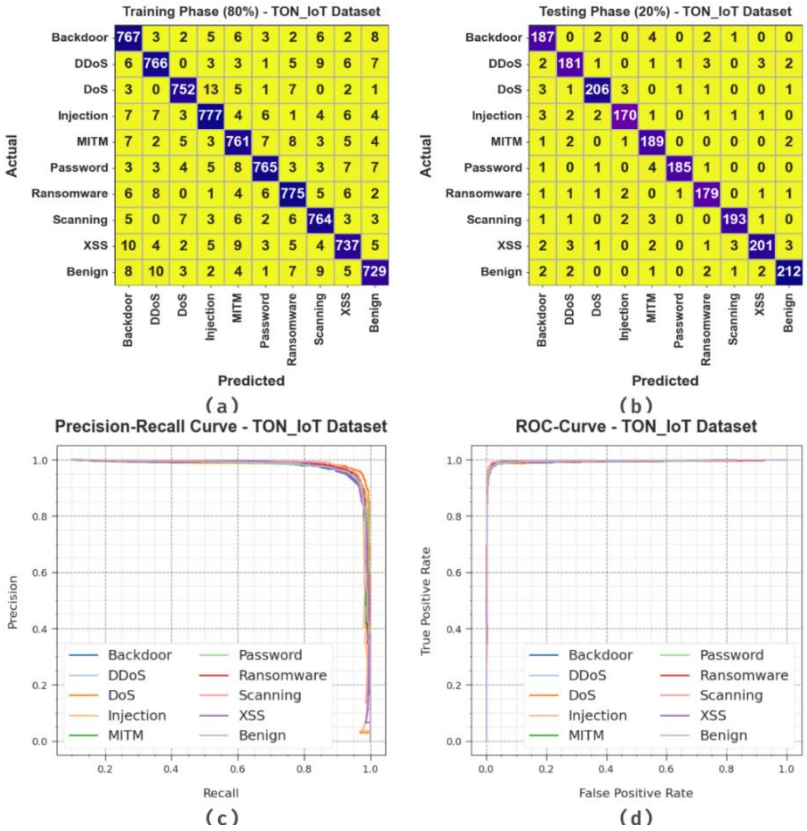

Figure 8. (a and b) Confusion matrices and (c and d) PR and ROC curves of the TON_IoT dataset.

The intrusion detection outcomes of the HAOADL-UAVN method upon the TON_IoT database are portrayed in Table 3 and Figure 9. The outcomes specify that the HAOADL-UAVN system correctly identified the intrusions. With 80% TRPH, the HAOADL-UAVN technique delivered an average $accu_y$ of 98.98%, $prec_n$ of 94.92%, $reca_l$ of 94.91%, F_{score} of 94.91% and an AUC_{score} of 97.17%. Moreover, with 20% TSPH, the HAOADL-UAVN method produced an average $accu_y$ of 99.03%, $prec_n$ of 95.16%, $reca_l$ of 95.16%, F_{score} of 95.14% and an AUC_{score} of 97.31%.

Table 3. Intrusion detection outcomes of the HAOADL-UAVN algorithm using the TON_IoT dataset.

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	AUC_{Score}
TRPH (80%)					
Backdoor	98.85	93.31	95.40	94.34	97.32
DDoS	99.04	95.39	95.04	95.21	97.26
DoS	99.28	96.66	95.92	96.29	97.78
Injection	98.98	95.10	94.87	94.99	97.16
MITM	98.84	93.95	94.53	94.24	96.93
Password	99.09	96.23	94.68	95.45	97.13
Ransomware	98.98	94.63	95.33	94.98	97.36

Continued on next page

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	AUC_{Score}
TRPH (80%)					
Scanning	99.02	94.67	95.62	95.14	97.51
XSS	98.89	94.61	94.01	94.31	96.71
Benign	98.88	94.68	93.70	94.19	96.57
Average	98.98	94.92	94.91	94.91	97.17
TSPH (20%)					
Backdoor	98.75	92.12	95.41	93.73	97.26
DDoS	98.75	93.78	93.30	93.54	96.32
DoS	99.10	96.26	95.37	95.81	97.46
Injection	99.05	95.51	93.92	94.71	96.74
MITM	98.90	92.20	96.92	94.50	98.02
Password	99.50	98.40	96.35	97.37	98.09
Ransomware	99.05	94.21	95.72	94.96	97.56
Scanning	99.30	96.98	96.02	96.50	97.84
XSS	98.85	96.17	93.06	94.59	96.30
Benign	99.05	95.93	95.50	95.71	97.49
Average	99.03	95.16	95.16	95.14	97.31

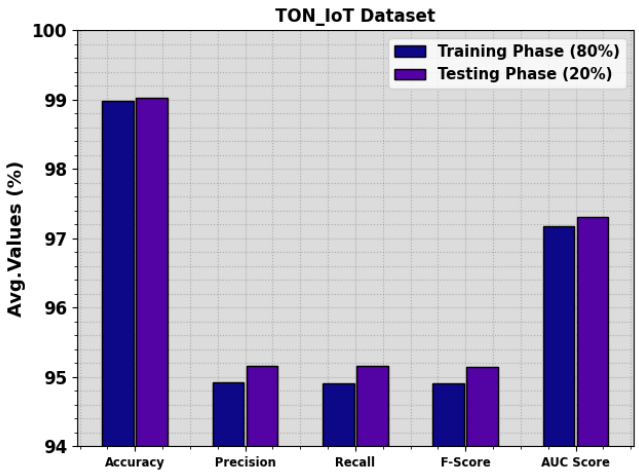

Figure 9. Average values of the HAOADL-UAVN algorithm on the TON_IoT database.

The $accu_y$ curves for TR and VL datasets are shown in Figure 10 for the HAOADL-UAVN algorithm using the TON_IoT dataset. The outcomes deliver valuable insights about its effectiveness with different count of epochs. Mainly, the proposed model achieved reliable improvement in both TR and TS $accu_y$ values with an increase in the number of periods. This phenomenon signifies the ability of the model to learn and recognize the patterns in TR and TS datasets. The rising trends in TS $accu_y$ underline the flexibility of the model to TR dataset, its aptitude for producing precise forecasts on unnoticed data and underscoring its strong generalization competencies.

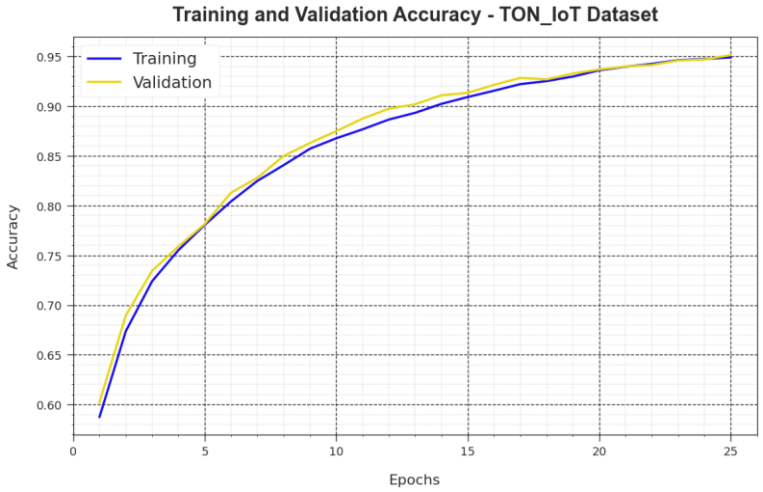

Figure 10. $Accu_y$ curve of the HAOADL-UAVN algorithm on the TON_IoT dataset.

Figure 11 shows an extensive overview of the TR and TS loss values of the HAOADL-UAVN algorithm upon the TON_IoT dataset across dissimilar number of epochs. The TR loss values progressively decrease as the model expands its abilities to reduce the classification errors under the datasets. The loss curves exemplify the configuration of the model with TR data, thus emphasizing its capacity to capture the outlines successfully. It is significant to observe that the HAOADL-UAVN algorithm constantly modified the limits in order to minimize the differences between the forecasted values and the actual TR labels.

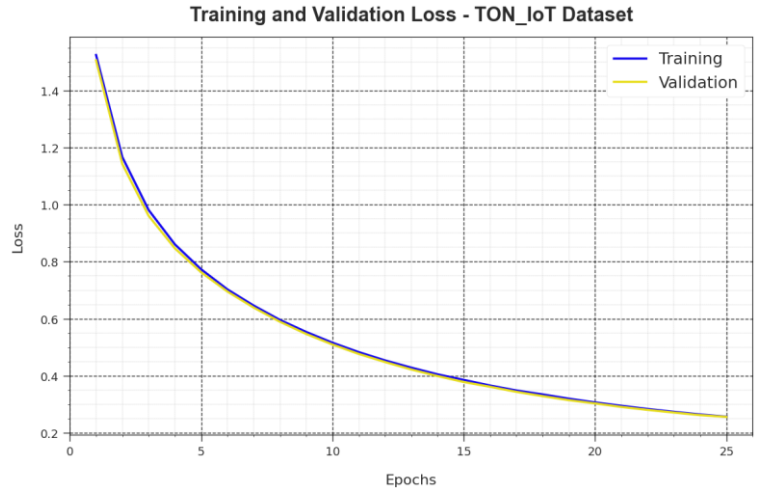

Figure 11. Loss curve of the HAOADL-UAVN algorithm on the TON_IOT dataset.

In Table 4 and Figure 12, the comparative analysis outcomes achieved by the HAOADL-UAVN model and other models using the TON_IoT database are depicted. The outcomes designate the incompetent performance of the LR. Next to that, the KNN, CNN, and DCA approaches displayed the modest outcomes. Although the LSTM_RNN technique achieved reasonable results, the HAOADL-UAVN method displayed the maximum results with an $accu_y$ of 99.03%, $prec_n$ of 95.16%, $reca_l$ of 95.16%, and F_{score} of 95.14%. Therefore, the HAOADL-UAVN technique has been proved that it

can be employed for secure UAV networks.

Table 4. Comparative analysis results of the HAOADL-UAVN method with other models using the TON_IoT dataset.

TON_IoT Dataset				
Methods	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
HAOADL-UAVN	99.03	95.16	95.16	95.14
LSTM_RNN Model	98.80	81.50	93.80	88.80
Logistic Regression	89.20	90.70	93.00	93.80
KNN Model	95.70	90.00	94.60	92.30
CNN Model	96.79	94.64	94.09	94.54
DCA Model	96.65	94.05	92.96	93.24

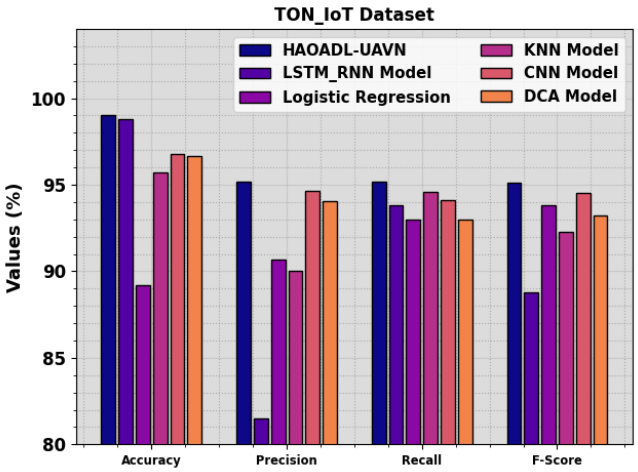

Figure 12. Comparative outcomes of the HAOADL-UAVN methodology under the TON_IoT dataset.

5. Conclusions

In the current study, the authors focused on designing and developing the HAOADL-UAVN technique. The purpose of the HAOADL-UAVN technique is to secure the communication in the UAV networks via threat detection. The HAOADL-UAVN approach comprises of four main processes namely, min-max normalization, HAOA-based FS, DBN-AE-based classification, and SOA-based hyperparameter tuning. Initially, the normalization of the network data is conducted through min-max normalization approach in order to scale the input data into a useful set-up. The HAOA is used to select an optimal set of features. Next, the security is attained via DBN-AE-based threat detection. Eventually, the hyperparameters of the DBN-AE algorithm are selected with the help of the SOA. A huge array of simulations was conducted upon the benchmark databases to demonstrate the improved performance of the HAOADL-UAVN model. The comprehensive results establish the supremacy of the HAOADL-UAVN methodology under distinct evaluation metrics. In the future, the HAOADL-UAVN system can be protracted to handle dynamic and developing threat landscapes, thus improving its adaptability. Furthermore, the integration of the model with real-time data streams and innovative anomaly

detection models would strengthen its skills in safeguarding the UAV systems. Moreover, identifying the applicability of the HAOADL-UAVN model in various operational atmospheres and scaling its efficacy for superior UAV fleets are potential avenues for future studies.

Use of AI tools declaration

The authors declare they have not used Artificial Intelligence (AI) tools in the creation of this article.

Acknowledgments

This research work was funded by Institutional Fund Projects under grant no. (IFPHI: 020-612-2020). Therefore, the authors gratefully acknowledge technical and financial support provided by the Ministry of Education and Deanship of Scientific Research (DSR) at King Abdulaziz University (KAU), Jeddah, Saudi Arabia.

Conflict of interest

The authors declare that they have no conflict of interest. The manuscript was written through contributions of all authors. All authors have given approval to the final version of the manuscript

References

1. T. Alladi, V. Kohli, V. Chamola, F. R. Yu, A deep learning-based misbehavior classification scheme for intrusion detection in cooperative intelligent transportation systems, *Digit. Commun. Netw.*, **9** (2023), 1113–1122. https://doi.org/10.1016/j.dcan.2022.06.018
2. K. H. Park, E. Park, H. K. Kim, Unsupervised intrusion detection system for unmanned aerial vehicle with less labeling effort. In: *Information security applications: 21st international conference*, Springer, Cham. 2020, 45–58. https://doi.org/10.1007/978-3-030-65299-9_4
3. S. Jeong, E. Park, K. U. Seo, J. Do Yoo, H. K. Kim, MUVIDS: false MAVLink injection attack detection in communication for unmanned vehicles. In: *Workshop on automotive and autonomous vehicle security (AutoSec)*, 2021. https://doi.org/10.14722/autosec.2021.23036
4. D. Basavaraj, S. Tayeb, Towards a lightweight intrusion detection framework for in-vehicle networks, *J. Sensor Actuator Netw.*, **11** (2022), 6. https://doi.org/10.3390/jsan11010006
5. P. Mansourian, N. Zhang, A. Jaekel, M. Kneppers, Deep learning-based anomaly detection for connected autonomous vehicles using spatiotemporal information, *IEEE T. Intell. Transp. Syst.*, **24** (2023), 16006–16017. https://doi.org/10.1109/TITS.2023.3286611
6. L. Yang, A. Moubayed, A. Shami, MTH-IDS: A multitiered hybrid intrusion detection system for internet of vehicles, *IEEE Internet Things J.*, **9** (2022), 616–632. https://doi.org/10.1109/JIOT.2021.3084796
7. N. Vanitha, P. Ganapathi, Traffic analysis of UAV networks using enhanced deep feed forward neural networks (EDFFNN). In: *Handbook of research on machine and deep learning applications for cyber security*. 2020. https://doi.org/10.4018/978-1-5225-9611-0.ch011

8. S. Aziz, M. T. Faiz, A. M. Adeniyi, K. H. Loo, K. N. Hasan, L. Xu, et al., Anomaly detection in the internet of vehicular networks using explainable neural networks (xnn), *Mathematics*, **10** (2022), 1267. https://doi.org/10.3390/math10081267
9. S. Anbalagan, G. Raja, S. Gurumoorthy, R. D. Suresh, K. Dev, IIDS: Intelligent intrusion detection system for sustainable development in autonomous vehicles, *IEEE T. Intell. Transp. Syst.*, **24** (2023), 15866–15875. https://doi.org/10.1109/TITS.2023.3271768
10. R. Fotohi, E. Nazemi, F. S. Aliee, An agent-based self-protective method to secure communication between UAVs in unmanned aerial vehicle networks, *Veh. Commun.*, **26** (2020), 100267. https://doi.org/10.1016/j.vehcom.2020.100267
11. F. Kateb, M. Ragab, Archimedes optimization with deep learning based aerial image classification for cybersecurity enabled UAV networks, *Comput. Syst. Sci. Eng.*, **47** (2023), 2171–2185. https://doi.org/10.32604/csse.2023.039931
12. M. A. Sayeed, R. Kumar, V. Sharma, Safeguarding unmanned aerial systems: An approach for identifying malicious aerial nodes, *IET Commun.*, **14** (2020), 3000–3012. https://doi.org/10.1049/iet-com.2020.0073
13. J. Tao, T. Han, R. Li, Deep-reinforcement-learning-based intrusion detection in aerial computing networks, *IEEE Netw.*, **35** (2021), 66–72. https://doi.org/10.1109/MNET.011.2100068
14. A. Masadeh, M. Alhafnawi, H. A. B. Salameh, A. Musa, Y. Jararweh, Reinforcement learning-based security/safety uav system for intrusion detection under dynamic and uncertain target movement, *IEEE T. Eng. Manage.*, 2022, 1–11. https://doi.org/10.1109/TEM.2022.3165375
15. R. Kumar, P. Kumar, R. Tripathi, G. P. Gupta, T. R. Gadekallu, G. Srivastava, SP2F: A secured privacy-preserving framework for smart agricultural Unmanned Aerial Vehicles, *Comput. Netw.*, **187** (2021), 107819. https://doi.org/10.1016/j.comnet.2021.107819
16. L. Almutairi, R. Daniel, S. Khasimbee, E. L. Lydia, S. Acharya, H. Kim, Quantum dwarf mongoose optimization with ensemble deep learning based intrusion detection in cyber-physical systems, I*EEE Access*, **11** (2023), 66828–66837. https://doi.org/10.1109/ACCESS.2023.3287896
17. I. U. Khan, A. Abdollahi, M. A. Khan, I. Uddin, I. Ullah, Securing against DoS/DDoS attacks in internet of flying things using experience-based deep learning algorithm, 2021. (Preprint) https://doi.org/10.21203/rs.3.rs-271920/v1
18. Q. Abu Al-Haija, A. Al Badawi, High-performance intrusion detection system for networked UAVs via deep learning, *Neural Comput. Appl.*, **34** (2022), 10885–10900. https://doi.org/10.1007/s00521-022-07015-9
19. M. S. Minu, R. Aroul Canessane, S. S. Subashka Ramesh, Optimal squeeze net with deep neural network-based arial image classification model in Unmanned Aerial Vehicles, *Traitement du Signal*, **39** (2022), 275–281. https://doi.org/10.18280/ts.390128
20. J. Tian, B Wang, R. Guo, Z. Wang, K. Cao, X. Wang, Adversarial attacks and defenses for deep-learning-based unmanned aerial vehicles, *IEEE Internet Things J.*, **9** (2021), 22399–22409. https://doi.org/10.1109/JIOT.2021.3111024
21. V. F. S. Francelin, J. Daniel, S. Velliangiri, Intelligent agent and optimization-based deep residual network to secure communication in UAV network, *Int. J. Intell. Syst.*, **37** (2022), 5508–5529. https://doi.org/10.1002/int.22800
22. H. Zhang, J. Xue, Q. Wang, Y. Li, A security optimization scheme for data security transmission in UAV-assisted edge networks based on federal learning, *Ad Hoc Netw*., 150 (2023), 103277. https://doi.org/10.1016/j.adhoc.2023.103277

23. A. M. Elshewey, M. Y. Shams, N. El-Rashidy, A. M. Elhady, S. M. Shohieb, Z. Tarek, Bayesian optimization with support vector machine model for Parkinson disease classification, *Sensors*, **23** (2023), 2085. https://doi.org/10.3390/s23042085
24. M. Stankovic, J. Gavrilovic, D. Jovanovic, M. Zivkovic, M. Antonijevic, N. Bacanin, et al., Tuning multi-layer perceptron by hybridized arithmetic optimization algorithm for healthcare 4.0, *Procedia Comput. Sci.*, **215** (2022), 51–60. https://doi.org/10.1016/j.procs.2022.12.006
25. H. Al-Khazraji, A. R. Nasser, A. M. Hasan, A. K. Al Mhdawi, H. Al-Raweshidy, A. J. Humaidi, Aircraft engines remaining useful life prediction based on a hybrid model of autoencoder and deep belief network, *IEEE Access*, **10** (2022), 82156–82163. https://doi.org/10.1109/ACCESS.2022.3188681
26. P. Krishnadoss, V. K. Poornachary, P. Krishnamoorthy, L. Shanmugam, Improvised seagull optimization algorithm for scheduling tasks in heterogeneous cloud environment, *Comput. Mater. Con.*, **74** (2023), 2461–2478. https://doi.org/10.32604/cmc.2023.031614
27. N. Moustafa, A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets, *Sustain. Cities Soc.*, **72** (2021), 102994. https://doi.org/10.1016/j.scs.2021.102994
28. R. A. Ramadan, A. H. Emara, M. Al-Sarem, M. Elhamahmy, Internet of drones intrusion detection using deep learning, *Electronics*, **10** (2021), 2633. https://doi.org/10.3390/electronics10212633
29. L. Kou, S. Ding, T. Wu, W. Dong, Y. Yin, An intrusion detection model for drone communication network in SDN environment, *Drones*, **6** (2022), 342. https://doi.org/10.3390/drones6110342