International Journal of Engineering Trends and Technology
ISSN: 2231–5381 / https://doi.org/10.14445/22315381/IJETT-V72I8P136

Volume 72 Issue 8, 389-399, August 2024

Original Article

Artificial Hummingbird Algorithm with Optimal Deep Learning-Based Intrusion Detection on Vehicular Adhoc Networks

K. Sarathkumar[1], P. Sudhakar[2], A. Clara Kanmani[3]

[1,2] *Department of Computer Science and Engineering, Annamalai University, Chidambaram, India.*
[3]*Computer Science and Engineering, PES University, Bangalore, India.*

[1]*Corresponding Author : sharathsmart3@gmail.com*

Received: 21 February 2024 Revised: 14 May 2024 Accepted: 31 July 2024 Published: 28 August 2024

Abstract *- Vehicular Ad hoc Networks (VANETs) permit transmission between the vehicle infrastructure, improving Intelligent Transportation Systems (ITS). Vehicles connect wirelessly to transmit the data. However, this transmission is susceptible to numerous attacks, mainly in Vehicle-to-Vehicle (V2V) settings. Intrusion Detection (ID) employing Deep Learning (DL) includes training Neural Networks (NNs) to identify the anomalies and patterns in network traffic, allowing the automatic recognition of potential security attacks and unauthorized events. DL methods, like recurrent NNs (RNNs) and convolutional NNs (CNNs), will efficiently analyze intricate and dynamic datasets for improved ID proficiencies. In this article, an artificial hummingbird algorithm with optimal DL-based ID (AHAODL-ID) technique on VANET is developed. The AHAODL-ID technique exploits feature selection with a hyperparameter selection model for detecting intrusions in the VANET. For data preprocessing, Z-score normalization is employed to scale the input data. Next, the AHA-based feature selection approach is executed for choosing an optimal feature subset. Meanwhile, the Bidirectional Long Short-Term Memory (BiLSTM) approach is implemented to identify various kinds of intrusions. Lastly, the hyperparameter election of the BiLSTM approach involves the design of a Manta Ray Foraging Optimization (MRFO) model. The experimental results of the AHAODL-ID technique are assessed using a benchmark IDS dataset. The obtained values underlined the advanced achievement of the AHAODL-ID technique over other existing models.*

Keywords *- VANET, IDS, Manta Ray Foraging Optimization, Intelligent Transportation System, Vehicle-to-Vehicle.*

1. Introduction

Currently, smart vehicles become a backbone of the bright idea of ITS that performs significant roles in improving road safety and enhancing driving capability [1]. For smart cities, road infrastructure is dependent upon Information and Communications Technologies (ICT), which are where smart vehicles are interconnected with alternative vehicles and Roadside Units (RSU). ITS is based on RSU to decrease accidents and increase driving effectiveness [2]. VANETs are susceptible to numerous categories of attacks owing to the environment in which they are exposed to the wireless medium. Inappropriately, such attacks considerably decrease the efficiency of VANET and pose significant challenges for valid drivers [3]. Then, securing VANET's traffic from alteration, intervention, and deletion of the data will be a considerable challenge and among the leading concerns for academic circles and industry. Malicious nodes can alert the communications that should be employed to guide the drivers [4]. Additionally, attackers will transfer false data that accidents may cause. To apply the VANETs in ITS efficiently, the innovative techniques to protect VANETs' traffic must be developed and implemented proficiently. Every VANET's traffic can be categorized for reliability [5]. Each category of attack in VANET will have specified features that create a profile for it. For effective implementation of VANETs in ITS, new techniques to protect VANET traffic must be developed and applied [6]. All the categories of attack in VANETs have a set of features that determines its profiles [7]. Also, Machine Learning (ML) techniques are popularly employed for analyzing large data quantities and obtaining helpful strategies for event identification, prediction, and categorization [8]. ML techniques are utilized in diverse applications like ID, traffic prediction, medicinal diagnosis, speech detection, and endorsement systems. ML is a required solution for detecting intrusions progressively with standard accuracy and speed in VANETs [9]. Several ML models are applied for IDS in VANETs, namely Support Vector Machine (SVM), NNs, and Decision Tree (DT). In general, numerous research workers have developed IDS employing ML and DL techniques [10]. This article develops an artificial hummingbird algorithm with optimal DL-based ID (AHAODL-ID) technique on VANET is developed. The AHAODL-ID technique exploits feature

selection with a hyperparameter selection strategy for detecting intrusions in the VANET. For data preprocessing, Z-score normalization is employed to scale the input data. Next, the AHA-based feature selection approach is executed for choosing an optimal feature subset. Meanwhile, the Bidirectional Long Short-Term Memory (BiLSTM) approach is implemented to identify various kinds of intrusions. Lastly, the hyperparameter election of the BiLSTM approach involves the design of a Manta Ray Foraging Optimization (MRFO) model. The experimental results of the AHAODL-ID technique are assessed using a benchmark IDS dataset.

2. Literature Works

Khalil et al. [11] developed an AI-based IDS model, integrating edge computing and DL methods. The technique utilizes the order preference by similarity to the ideal solution (TOPSIS) and a Bidirectional Generative Adversarial Network (BiGAN) method for IDS. In [12], an innovative ID and Mitigation System (IDMS) was designed that is dependent upon the optimization-enabled DL method. The two main stages are mitigation, attack detection, and feature extraction. Primarily, the features are removed. Subsequently, these removed features have been subjected to the attack detection stage.

Also, an improved particle swarm optimization (IPSO) approach was presented. Similarly, the BAIT-based mitigation method was employed. Chougule et al. [13] projected the multi-branch reconstruction error (MbRE) IDS method. The developed technique encompasses 3 CNN-based reconstruction methods. Ding et al. [14] developed a lightweight, effective, and explainable DL method, DeepSecDrive architecture. This method provides feature-extracting components developed with deformity complexities and a lightweight non-local network (LNLN). DeepSecDrive incorporates Shapley additive exPlanations for constructing the multi-level interpretability elements. Manderna et al. [15] introduced an AI-based NIDS that implements DL techniques.

This method combines the self-attention (SA)-based BiLSTM (SA-BiLSTM) and Cascaded CNN (CCNN) techniques for classifying and learning higher-level factors. The Multi-variant gradient-assisted optimizer (MV-GBO) method was implemented to increase the SA-BiLSTM and CCNN to improve the efficiency. Data was also obtained by employing the MV-GBO-based feature extraction, which could be utilized to enrich feature learning. Reka et al. [16] presented a clustering method. Compact cluster realization was executed using the COA method. The multi-head SA-enabled gated graph CNN (MSA-GCNN) technique with a fusion IDS is also used. In [17], a residual-assisted temporal attention red fox-CNN (RTARF-CNN) model was presented. The system designed the RF and Local Least Squares (DRFLLS) model. The Stacked Contractive AE (St-CAE) technique was employed.

The Residual-based Temporal Attention-CNN (RTA-CNN) and Red Fox Optimizer (RFO) were used. Narayanan and Naresh [18] proposed a Tree Hierarchical Deep CNN (THDCNN) and Identity-assisted Network (THDCNN) model. In [19], an ID and Mitigation System (IDMS) model are introduced. Also, an IPSO model is employed. K-means and Balancing Composite Motion Optimization (BCMO) models are also used. Masood and Zafar [20] present an IDS model implementing the Graph Neural Networks (GNN) technique.

3. The Proposed Model

In this article, an AHAODL-ID technique on VANET is developed. The AHAODL-ID technique exploits feature selection with a hyperparameter selection model for detecting intrusions in the VANET. Figure 1 shows the overall procedure of the presented AHAODL-ID technique.

3.1. Data Preprocessing

For data preprocessing, Z-score normalization is implemented for scaling the input data, also called standardization. It is a statistical technique employed for converting a dataset into a standard normal distribution with a standard deviation of one and a mean of zero [21]. This method supports comparing and analyzing data that can initially have diverse scales. This normalization enables the comparison of data points at various features or variables, making it easy to recognize patterns and outliers and execute statistical evaluations.

3.2. Feature selection using AHA

At this stage, the AHA-based feature selection approach is executed to choose an optimal subset of features. AHA is inspired by the intellectual characteristics of hummingbirds, which analyzes various food sources to select an appropriate food source [22]. Each individual often remembers the particular food source allotted to it. The collection of food sources is stored in the visit table.

3.2.1. Initialization

The procedure has begun by placing hummingbirds on m food sources.

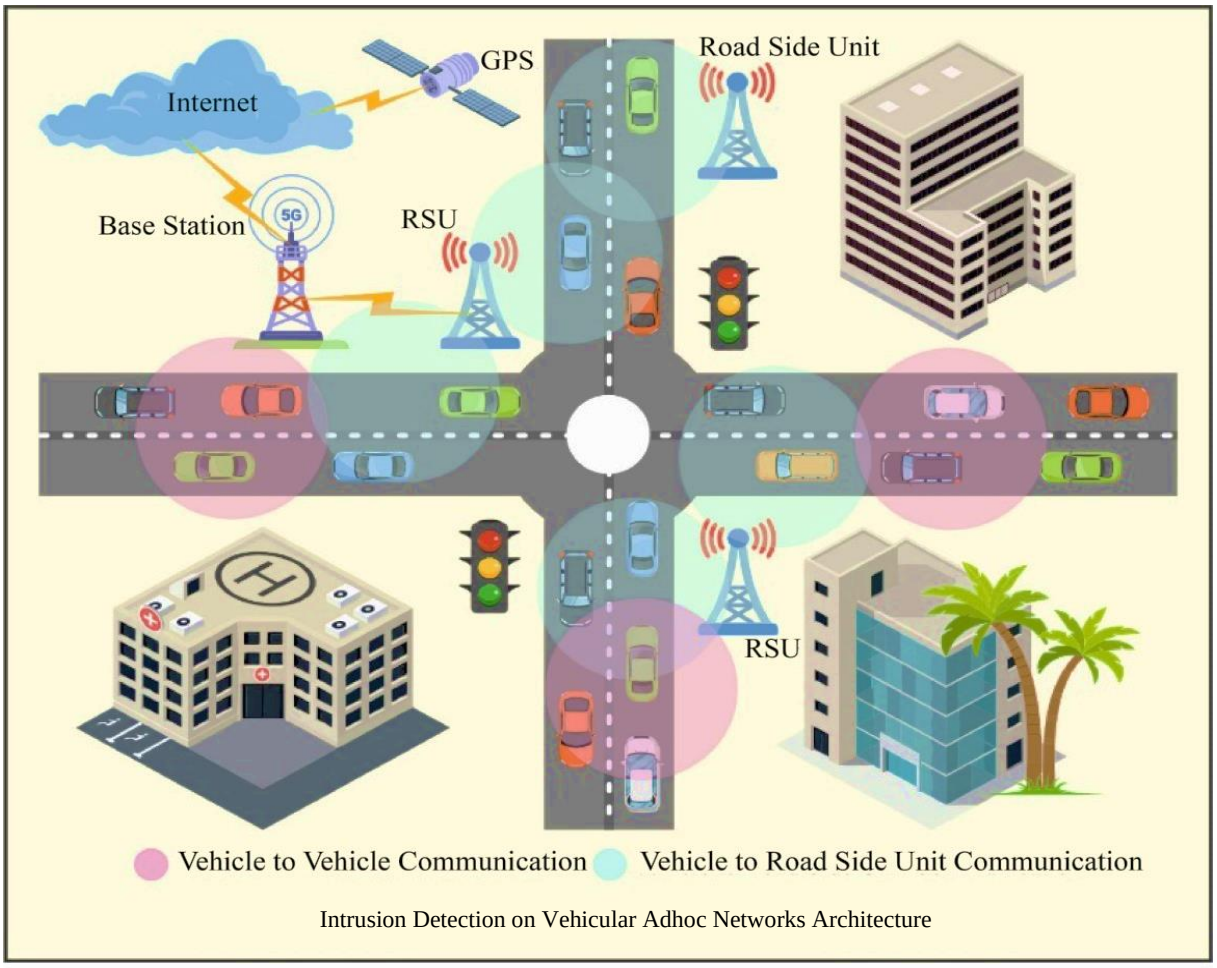
$$c_r = Low + i.(Up - Low) r = 1, \ldots, \quad (1)$$

In Equation (1), c_r indicates the location of r^{th} food supply in the problem space, $and\ Up$ and Low denote the upper and lower boundaries of d-dimensional problems. i indicates a random vector within $[0,1]$.

3.2.2. Guided Foraging

Hummingbirds are talented in finding food sources with the maximum amount of nectar. The three fight skills utilized during foraging are axial, omnidirectional, and diagonal.
The axial fight in $w - W$ space is described by:

$$W^{(r)} = \begin{cases} 1 & if\ r = randi([1, w]) \\ 0 & else \end{cases} r = 1, \ldots, w \quad (2)$$



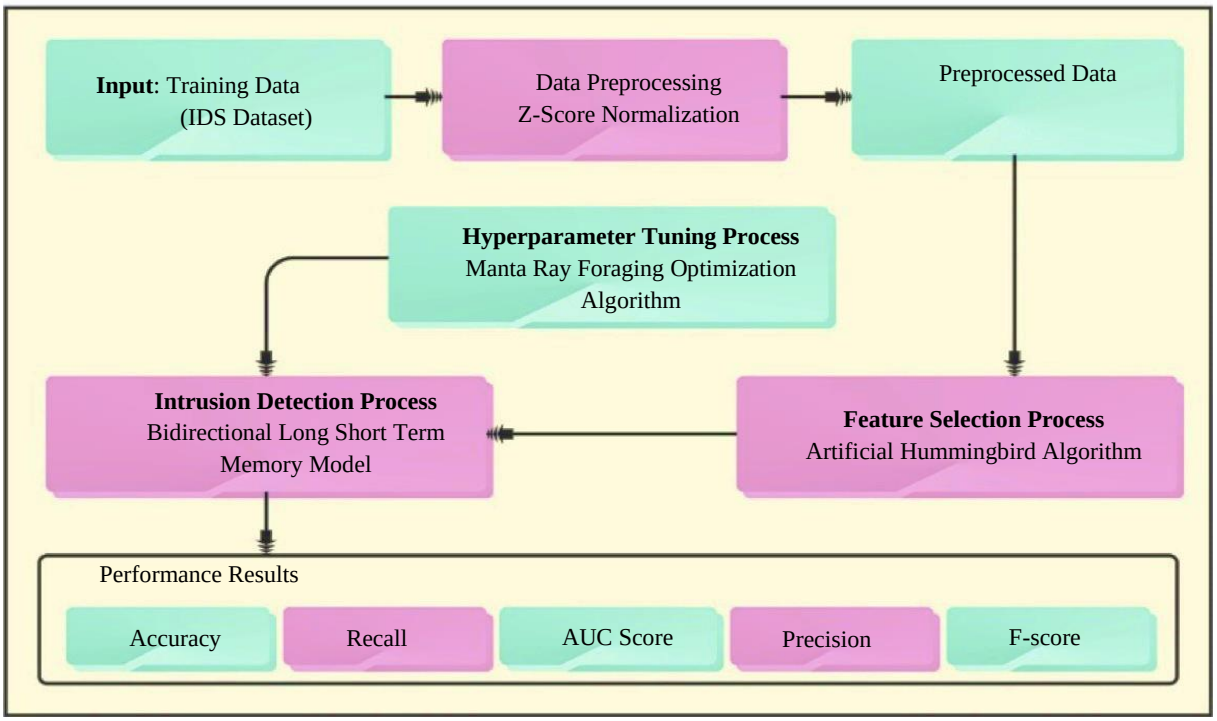


Fig. 1 Overall process of AHAODL-ID technique

The diagonal fight is represented as:

$$W^{(r)} = \begin{cases} 1 \ if \ r = k(q), q \in [1,p], \\ k = randperm(p), \\ p \in [2, \lceil i_1.(w-2) \rceil + 1] \\ 0 \ else \end{cases} \ i = 1, \dots, w \quad (3)$$

The omnidirectional fight is denoted by:

$$W^{(r)} = 1 \, r = 1, \dots, w \quad (4)$$

Here, $randi([1,w])$, and $randperm(p)$ random numbers range between [1,w] and [0,1]:

$$E_r = (S+1) = c_{r,tar}(S) + z.W.\left(c_r(g) - c_{r,tar}(g)\right) \quad (5)$$

$$z \sim M(0,1) \quad (6)$$

Now, $c(g)$ is the location of r^{th} food source at time g, $c_{r,tar}(g)$ indicates the r^{th} food source location where the hummingbird plans to visit, and a is the guided factor that undergoes uniform distribution $M(0,1)$.

$$c_r(g+1) = \begin{cases} c_r(\mathrm{g}) \ u\big(c_r(g)\big) \le u\big(e_r(g+1)\big) \\ e_r(g+1) u\big(c_r(g)\big) > u\big(e_r(g+1)\big) \end{cases} \quad (7)$$

Here, u indicates the function fitness value

3.2.3. Territorial Foraging

The hummingbird searches for new food sources within the territory when the targeted food source is completely eaten, and it is formulated as:

$$e_r(g+1) = c_r(g) + y.W_C(g) \quad (8)$$

$$y \sim M(0,1) \quad (9)$$

Here, y indicates the territorial factor that undergoes uniform distribution $M(0,1)$. The migration from hummingbird to random one with the worst nectar replenishment rate is represented as:

$$C_{worst}(S+1) = LOW + i.(UP — LOW) \quad (10)$$

In Eq. (10), C_{worst} represents the food source with the worst nectar-refilling charge in the population. Guided foraging has a uniform probability of visiting various sources. Therefore, the hummingbird moves toward an equivalent food source as its target after performing a $2m$ repetition in the worst case. To discover the hunting ground and increase the stagnation, the migration approach should be completed at this stage.

$$N = 2m \quad (11)$$

The computational complexity is related to initialization, the hummingbird population size (N_{size}), the health evaluation (x_{eval}), and the measurement of variables (d_{var}) and the maximum iterations ($T_{\max}$),

$$O(AHO) = O(problemdefinition) + O(initialization) + O\big(S(evaluationunction)\big) + O\big(g(guidedforaging)\big) + O(g(teritorialforaging)) + O\big(g(migrationforaging)\big) = O\left(1 + mw + Gxm + \frac{1}{2}Gmw + \frac{1}{2}Gmw\frac{G}{2m}mx\right) \cong O\left(Gxm + Gmw + \frac{Gw}{2}\right) \quad (12)$$

3.2.4. Searching Characteristics

Rosenbrock and Rastrigin are the two test functions that prove the searching characteristics of AHA. $c = (1,1)$ with $u(c) = 0$ is the optimal solution for Rosenbrock function. $c = (O,0)$ with $u(c) = 0$ is its optimal solution for Rastrigin. The fitness function (FF) employed in the AHA method was developed to achieve stability among several chosen factors in every solution (minimum) and classify accuracy (maximum) by employing these chosen factors. Eq. (13) portrays the FF to analyze outcomes.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \quad (13)$$

Now, $\gamma_R(D)$ shows the classifying error rate of a specified classifier. $|R|$ defines the cardinality of the chosen subset, and $|C|$ is the complete dataset feature. α and β describe two factors concerning the prominence of classification quality and length of the subset: $\in [1,0]$ and $\beta = 1 - \alpha$.

3.3. ID utilizing BiLSTM Approach

In this work, the BiLSTM model is used to identify various kinds of intrusions. LSTM is a superior kind of RNN that integrates LSTM unis [23]. Hochreiter and Schmidhuber, German scholars, established these units as a solution to the problem of the gradient disappearing. LSTM is widely used in different applications and now has proven effective over a large number of problem domains. LSTM comprises a memory block with self-connection. This cell retains the temporal state of the network. In addition, LSTM incorporates multiplicative units known as gates, enabling the modification, storage, and retrieval of data. Every cell determines the data to be stored and controls the closing and opening of the gates to reset, write, and read. The gate operates in an open and closed manner. However, the gate is carried out using component-wise multiplication via the sigmoid function, which constricts the value within [0, 1].

This facilitates the backpropagation model and enables differentiability. The gate processes the received signal, selectively blocking or allowing data based on the importance and relevance, which can be defined by the respective weight, and during the learning process, these weights are adjusted, changing the hidden and input states of LSTM. Firstly, the LSTM unit decides what data to be retained. The forget layer, which deals with x_t and h_{t-1}, then outputs a value ranging from zero to one in the c_{t-1} cell state using the sigmoid function. Figure 2 demonstrates the infrastructure of BiLSTM.

$$f_t = \sigma\big(w_f \cdot x_t + u_f \cdot h_{t-1} + b_f\big) \quad (14)$$

Next, the LSTM unit chooses the new data saved in the cell state. The layer of input gate i_t will decide the value to be upgraded by the sigmoid function.

$Tanh$ function constructs a vector of new candidate value $\tilde{c}$ that is supplied to the state.

$$i_t = \sigma(w_i \cdot x_t + u_i \cdot h_{t-1} + b_i) \quad (15)$$

$$\tilde{c}_t = tanh(w_c \cdot x_t + u_c \cdot h_{t-1} + b_c) \quad (16)$$

Next, the old state c_{t-1} is upgraded to c_τ new cell state.

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \tilde{c}_t, \quad (17)$$

At last, the output gate 0_τ defines which memory content is to be yielded to the next hidden state.

$$o_t = \sigma(w_o \cdot x_t + u_o \cdot h_{t-1} + b_o) \quad (18)$$

$$h_t = o_t \cdot tanh(c_t) \quad (19)$$

Where w_0 and u_0 are the weights, $(\cdot)$ denotes the inner product, and b_0 is the bias. BiLSTM is an amendment to the LSTM model, which contains backward and forward hidden states.

$$\overleftarrow{h} = o_t \cdot tanh(c_t) \quad (20)$$

In Equation (20), c_t indicates the cell state and o_t denotes the output gate. Likewise, the $\overleftarrow{h}$ backwards hidden state is evaluated to the forward layer and is merged and given to the following layer. Bi-LSTM incorporates present and previous data for each point. The component-wise addition is employed for concatenating the backwards and forward pass outputs.

$$h_t = \overleftarrow{h_t} \oplus \overrightarrow{h_t} \quad (21)$$

The activation function is utilized for the h_τ hidden state for generating the last output y_t.

3.4. Hyperparameter Tuning utilizing MRFO

Lastly, the hyperparameter election of the BiLSTM method involves designing the MRFO. This section discusses the biologically inspired and mathematical modelling of three MRFO foraging strategies [24].

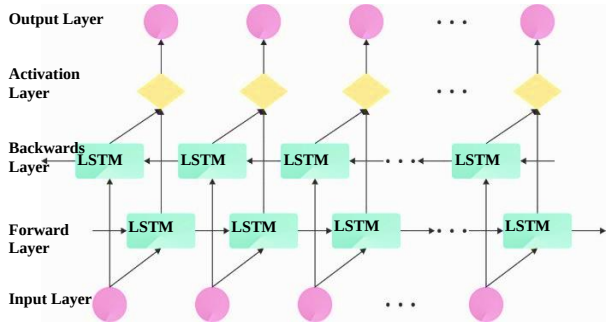


Fig. 2 Structure of BiLSTM

3.4.1. Chain Foraging

In nature, the foraging phase includes the displacement of Manta Rays (MRs) to the high plankton concentration. The MR impacts individual movement and the fittest individual in front of it since the MRs are aligned in chain form during foraging. The mathematical expression of foraging behaviour is given below.

$$x_i^d(t+1) = \begin{cases} x_i^d(t) + r \cdot \left(x_{best}^d(t) + x_i^d(t)\right) + \alpha \cdot \left(x_{best}^d(t) - x_i^d(t)\right); i = 1 \\ x_i^d(t) + r \cdot \left(x_{i-1}^d(t) + x_i^d(t)\right) + \alpha \cdot \left(x_{best}^d(t) - x_i^d(t)\right); i \in [2, NP] i = 1 \end{cases} \quad (22)$$

In Eq. (22), $d \in [1, D]; r$ is a vector having random integers within [0,1]; $x_i^d(t)$ indicates the location of i^{th} manta rays in the d^{th} dimension at t^{th} iterations, weight coefficient $\alpha = 2r\sqrt{|\log(r)|}$; NP shows the overall amount of population size in D-dimensional space and $x_{best}^d(t)$ indicates the present optimum individual.

3.4.2. Cyclone Foraging

The cyclone foraging includes spiral movement towards the food source and the MR movement based on the one in front of it. Thus, exploitation (intensive search) near the present optimum solution can be obtained, and it can be mathematically modelled as follows:

$$x_i^d(t+1) = \begin{cases} x_i^d(t) + r \cdot \left(x_{best}^d(t) + x_i^d(t)\right) + \beta \cdot \left(x_{best}^d(t) - x_i^d(t)\right); i = 1 \\ x_i^d(t) + r \cdot \left(x_{i-1}^d(t) + x_i^d(t)\right) + \beta \cdot \left(x_{best}^d(t) - x_i^d(t)\right); i \in [2, NP] i = 1 \end{cases} \quad (23)$$

In Eq. (10), the weight factor $= 2 \cdot e\left(r_1 \cdot \frac{T_{\max} - t + 1}{T_{\max}}\right) \sin(2\pi r_1)$; maximum iteration count is represented as $T_{\max}$; and r_1 is the random integer within [0,1]. Furthermore, cyclone foraging integrates the movement of MRs according to a random location in the search space to avoid local optima solution $x_{rand} \neq x_{best}$.

$$x_i^d(t+1) = \begin{cases} x_{rand}^d(t) + r \cdot \left(x_{rand}^d(t) + x_i^d(t)\right) + \beta \cdot \left(x_{rand}^d(t) - x_i^d(t)\right); i = 1 \\ x_{rand}^d(t) + r \cdot \left(x_{i-1}^d(t) + x_i^d(t)\right) + \beta \cdot \left(x_{rand}^d(t) - x_i^d(t)\right); i \in [2, NP] i = 1 \end{cases} \quad (24)$$

$$x_{rand}^d = Lb^d + r \cdot (Ub^d - Lb^d) \quad (25)$$

Where Lb^d and Ub^d denote the lower and upper boundaries at the d^{th} dimensional problem. If $(t/T_{\max}) < rand$, $rand$ denotes the randomly generated number within [0, 1]. The critical characteristic of any metaheuristic is the shift from exploratory to exploitative behaviours. When $rand < 0.5$, the foraging of the chain shifts to the cyclone foraing. MRFO is used to provide a statistical occurrence probability of cyclone and chain foraging stages if $rand < p$ switch condition amid the foraging of chain and cyclone is adjusted, where p indicates the user-selectable control parameter.

3.4.3. Somersault Foraging

Taking into account the current best solution (plankton's location) as a pivot, the MRs move forward and back to search

for a new position by somersaulting about the pivot, and it can be mathematically modelled as follows:

$$x_i^d(f+1) = x_i^d(t) + S \cdot \left(r_2 \cdot x_{best}^d(t) - r_3 \cdot x_i^d(t)\right), i = 1,2,\dots,NP \quad (26)$$

In Equation (26), S denotes the somersault factor; r_2 and r_3 are randomly selected numbers within [0,1]. The MRFO method offers an FF for the accomplishment of increased classification effectiveness. It finds a positive numeral signifying the excellent accomplishment of the candidate solutions. The decrease in the classification error rate will be computed as the FF, as specified in Equation (27).

$$fitness(x_i) = ClassifierErrorRate(x_i) = \frac{number\ of\ misclassified\ samples}{Total\ number\ of\ samples} \times 100 \quad (27)$$

Table 1. Dataset specification

Classes	Record Numbers
Backdoor	1000
DDoS	1000
DoS	1000
Injection	1000
MITM	1000
Password	1000
Ransomware	1000
Scanning	1000
XSS	1000
Benign	1000
Total Records	**10000**

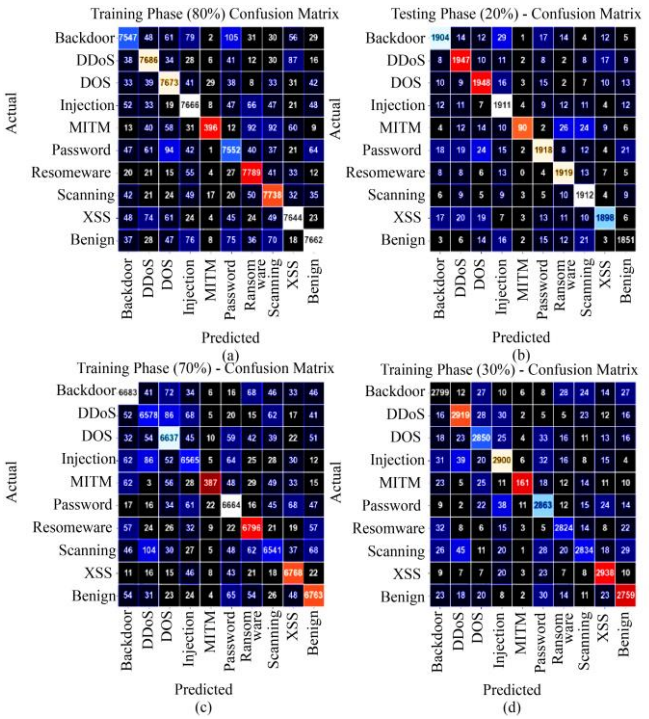


Fig. 3 Confusion matrices of (a-b) 80:20 and (c-d) 70:30 of TRSPH/TESPH

4. Performance Validation

The performance analysis of the AHAODL-ID technique occurs using the TON-IoT dataset [25]. It comprises 10000 instances with ten classes, as shown in Table 1. The AHAODL-ID method has selected 24 feature sets from the available 42 features. Figure 3 portrays the confusion matrices achieved by the AHAODL-ID approach with 80:20 and 70:30 of TRAPH/TESPH. These experimentation outputs portray that the AHAODL-ID approach can be successful in recognition and classification with ten classes precisely. Table 2 and Figure 4 highlight the ID outputs of the AHAODL-ID approach under 80:20 of TRAPH/TESPH. The investigational outputs show that the AHAODL-ID approach correctly recognized the intrusions. With 80% of TRAPH, the AHAODL-ID method exhibits average $accu_y$, $prec_n$, $reca_l$, F_{score}, and AUC_{score} values of 99.05%, 94.15%, 91.13%, 92.20%, and 95.30%, respectively.

Besides, with 20% of TESPH, the AHAODL-ID method gives average $accu_y$, $prec_n$, $reca_l$, F_{score}, and AUC_{score} values of 99.01%, 93.80%, 90.60%, 91.68%, and 95.02%, correspondingly. Table 3 and Figure 5 portray the ID outputs of the AHAODL-ID approach under 70:30 of TRAPH/TESPH. These experimental values depict that the AHAODL-ID approach correctly recognized the intrusions. In 70% of TRAPH, the AHAODL-ID approach provides average $accu_y$, $prec_n$, $reca_l$, F_{score}, and AUC_{score} values of 98.96%, 93.78%, 91.17%, 92.16%, and 95.29%. Also, on 30% of TESPH, the AHAODL-ID method gains average $accu_y$, $prec_n$, $reca_l$, F_{score}, and AUC_{score} values of 98.94%, 93.39%, 91.14%, 92.03%, and 95.27%, correspondingly. The effectiveness of the AHAODL-ID technique under 80:20 of TRAPH/TESPH is demonstrated in Figure 6 under training accuracy (TRAA) and validation accuracy (VALA) curves. The figure portrays the behaviour of the AHAODL-ID technique over changing epochs, illustrating its learning and generalization capacities. The figure exhibits an unintermittent growth in the TRAA/VALA with an epoch surge. It confirms the adaptive behaviour of the AHAODL-ID model in detecting patterns under the TRA/TES data. The increased trends in VALA underline the capacity of the AHAODL-ID model to adjust to the TRA data and also surpass in precise categorization of hidden data, underscoring robust generalization capabilities. Figure 7 depicts an overall depiction of the training loss (TRLA) and validation loss (VALL) outputs of the AHAODL-ID approach under 80:20 of TRAPH/TESPH. The steady decrease in TRLA accentuates the AHAODL-ID approach improving the weights and minimalizing the classifying error on the TRA/TES data. The figure exhibits a precise comprehension of the AHAODL-ID technique associated with the TRA data, underscoring its superiority in capturing patterns inside both datasets. Notably, the AHAODL-ID technique frequently increases its parameters to decrease the discrepancies between the anticipated and real TRA classes.

Table 2. ID outcome of AHAODL-ID model with 80:20 of TRAPH/TESPH

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	AUC_{Score}
TRAPH (80%)					
Backdoor	98.94	95.81	94.48	95.14	96.99
DDoS	99.10	95.47	96.34	95.90	97.89
DoS	99.03	94.89	96.31	95.60	97.84
Injection	98.95	94.75	95.74	95.24	97.54
MITM	99.33	83.37	49.32	61.97	74.60
Password	98.88	94.85	94.89	94.87	97.13
Ransomware	99.19	95.59	97.16	96.37	98.30
Scanning	99.01	94.75	96.39	95.56	97.86
XSS	99.02	95.51	95.60	95.56	97.52
Benign	99.08	96.50	95.10	95.79	97.33
Average	**99.05**	**94.15**	**91.13**	**92.20**	**95.30**
TESPH (20%)					
Backdoor	98.93	95.68	94.63	95.15	97.05
DDoS	98.99	94.74	96.29	95.51	97.81
DoS	98.92	94.61	95.82	95.21	97.57
Injection	98.86	93.81	95.89	94.84	97.55
MITM	99.30	81.82	45.69	58.63	72.79
Password	98.84	95.61	93.97	94.79	96.71
Ransomware	99.12	95.19	96.77	95.97	98.09
Scanning	99.07	94.56	96.96	95.74	98.14
XSS	99.03	96.44	94.71	95.57	97.14
Benign	99.02	95.56	95.27	95.41	97.37
Average	**99.01**	**93.80**	**90.60**	**91.68**	**95.02**

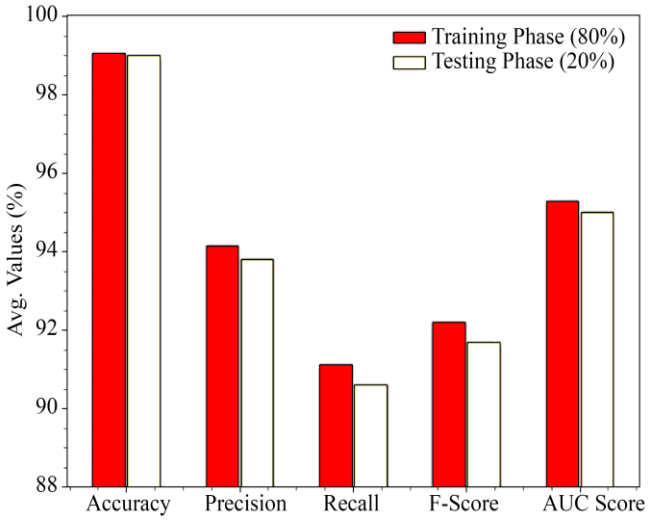


Fig. 4 Average of AHAODL-ID method with 80:20 of TRAPH/TESPH

Table 3. ID outcome of AHAODL-ID model under 70:30 of TRAPH/TESPH

Classes	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}	AUC_{Score}
TRAPH (70%)					
Backdoor	98.81	94.45	94.86	94.65	97.08
DDoS	98.84	94.61	94.73	94.67	97.03
DoS	98.83	94.40	94.94	94.67	97.12
Injection	98.86	94.73	94.75	94.74	97.05
MITM	99.38	83.95	54.51	66.10	77.19
Password	98.88	94.54	95.34	94.94	97.33
Ransomware	99.06	95.34	96.22	95.78	97.82
Scanning	98.81	95.14	93.87	94.50	96.64
XSS	99.20	95.66	97.13	96.39	98.29
Benign	98.92	94.96	95.36	95.16	97.36
Average	**98.96**	**93.78**	**91.17**	**92.16**	**95.29**
TESPH (30%)					
Backdoor	98.74	93.74	94.72	94.23	96.98
DDoS	98.92	94.83	95.52	95.17	97.43
DoS	98.81	94.50	94.72	94.61	97.02
Injection	98.73	94.25	94.43	94.34	96.85
MITM	99.39	80.90	55.52	65.85	77.69
Password	98.79	94.02	95.12	94.57	97.18
Ransomware	99.11	95.60	96.15	95.88	97.81
Scanning	98.81	95.68	93.47	94.56	96.47
XSS	99.15	95.51	96.90	96.20	98.17
Benign	98.91	94.91	94.88	94.89	97.13
Average	**98.94**	**93.39**	**91.14**	**92.03**	**95.27**

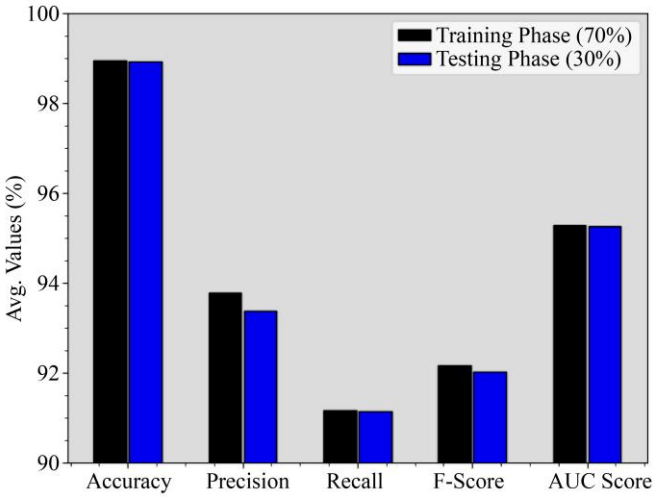


Fig. 5 Average of AHAODL-ID method with 70:30 of TRAPH/TESPH

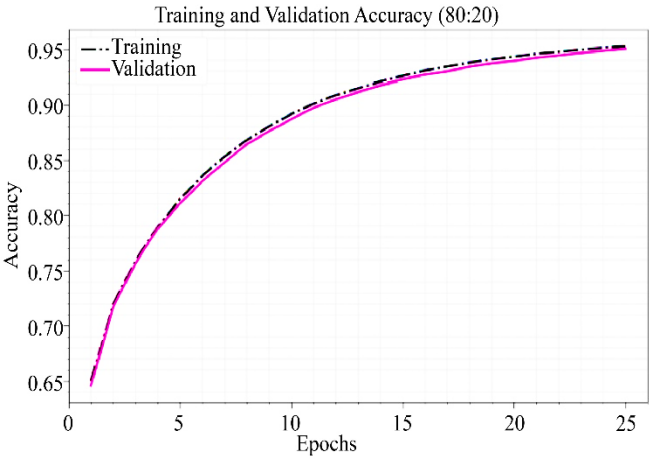


Fig. 6 $Accu_y$ the curve of the AHAODL-ID method with 80:20 of TRAPH/TESPH

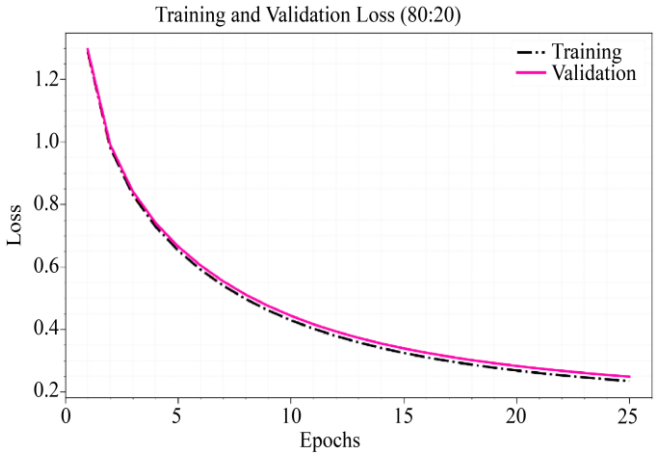


Fig. 7 Loss curve of AHAODL-ID approach under 80:20 of TRAPH/TESPH

In Figure 8, the outputs confirmed that the AHAODL-ID approach with 80:20 of TRAPH/TESPH gradually attained greater PR values in each class. It authenticates the increased capacities of the AHAODL-ID method in the recognition of discrete classes, illustrating superiority in the detection of classes.

Likewise, the ROC curves produced by the AHAODL-ID approach under 80:20 of TRAPH/TESPH are depicted in Figure 9. It outperforms the classification of distinct labels. It also gives an elaborated comprehension of the trade-off between TPR/FRP over diverse detection threshold values and epochs. The figure underscored the improved classification outputs of the AHAODL-ID approach under each class, accentuating the efficiency in addressing many classifying issues.

The results of the AHAODL-ID approach are compared with recent models in Table 4 and Figure 10 [15, 19, 26, 27]. The results indicate that the Logistic Regression (LR), KNN, AdaBoost, SVM, and BC models reported poor performance. While, the CNN-GRU and LSTM-RNN models exhibited improved performance. Also, the AHAODL-ID technique exhibits its supremacy with enhanced $accu_y$, $prec_n$, $reca_l$, and F_{score} of 99.05%, 94.15%, 91.13%, and 92.20%, relatively.

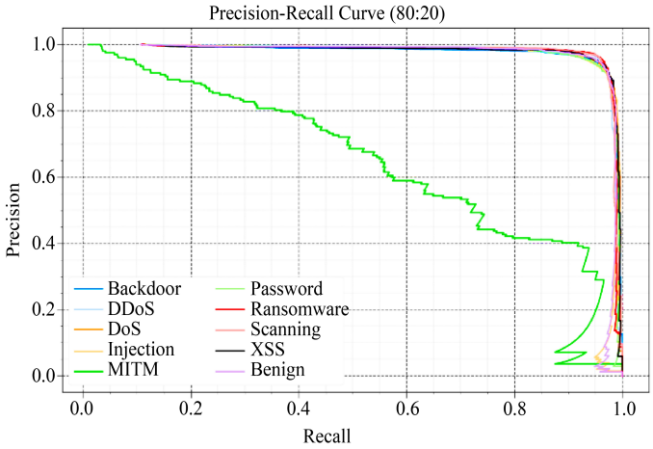


Fig. 8 PR curve of AHAODL-ID approach under 80:20 of TRAPH/TESPH

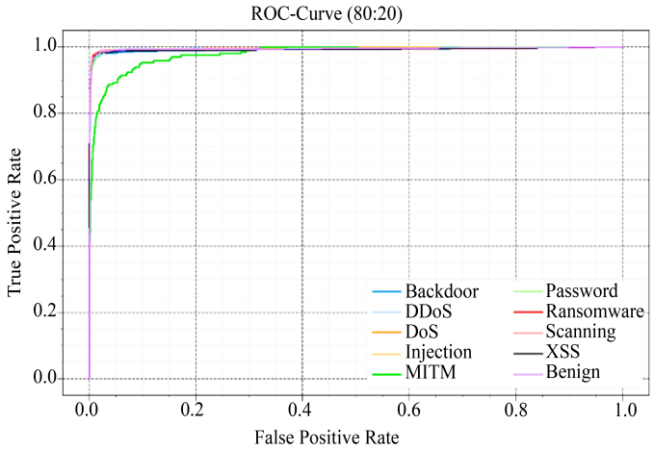


Fig. 9 ROC curve of AHAODL-ID approach under 80:20 of TRAPH/TESPH

Table 4. Comparative evaluation of the AHAODL-ID method with existing models

Techniques	$Accu_y$	$Prec_n$	$Reca_l$	F_{Score}
AHAODL-ID	99.05	94.15	91.13	92.20
LSTM_RNN	92.80	81.50	75.80	80.80
LR	89.20	80.70	73.00	76.80
KNN	88.70	82.00	64.60	79.30
CNN-GRU	96.00	84.00	78.12	80.01
AdaBoost	90.90	82.70	75.90	70.00
SVM	86.80	78.40	75.90	80.20
Bayesian-Coresets	82.40	80.13	76.34	75.59

Table 5. CT output of AHAODL-ID technique with existing models

Methods	CT (sec)
AHAODL-ID	0.71
LSTM_RNN	2.03
LR	2.65
KNN	3.62
CNN-GRU	2.95
AdaBoost	2.18
SVM	2.75
Bayesian-Coresets	3.25

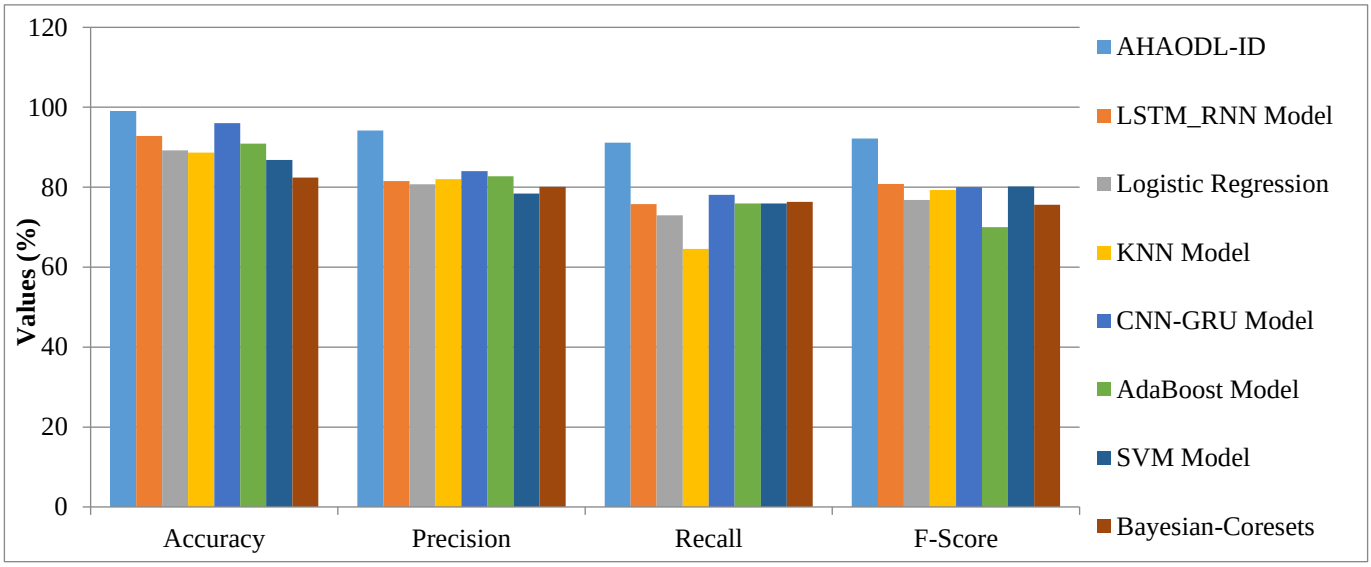


Fig. 10 Comparative evaluation of the AHAODL-ID method with existing models

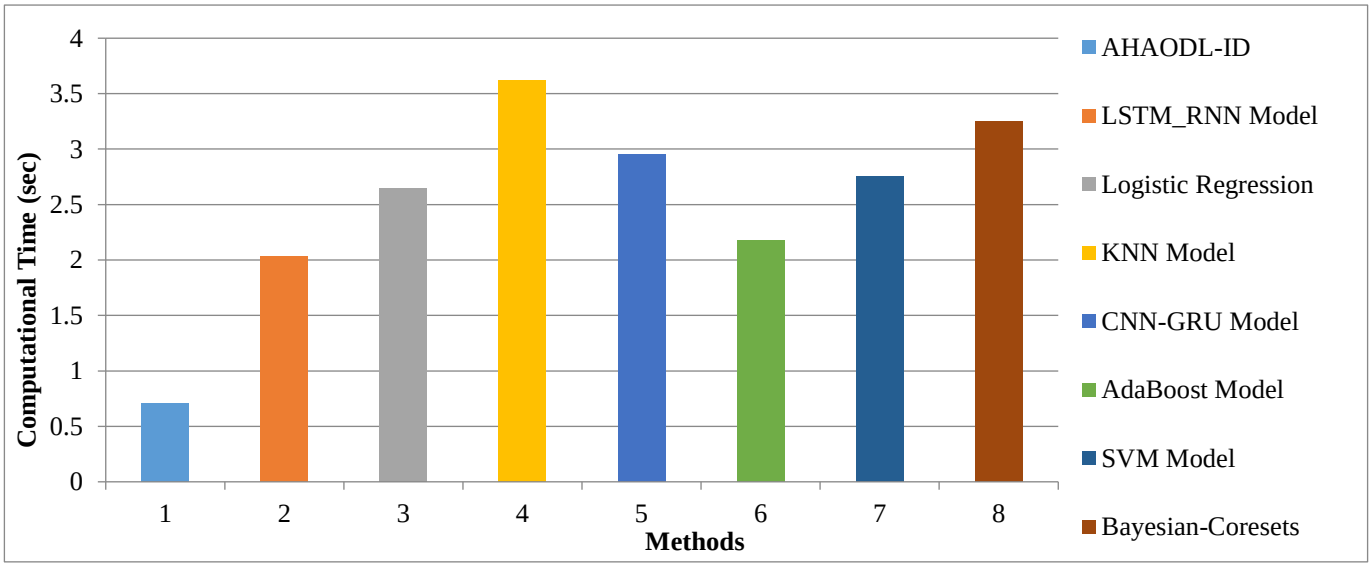


Fig. 11 CT analysis of AHAODL-ID technique with existing models

A wide-ranging Computational Time (CT) outcome of the AHAODL-ID method is compared with other models in Table 5 and Figure 11. These investigational outcomes specify that the LR, KNN, AdaBoost, SVM, and BC methods have described poorer performance. Concurrently, the CNN-GRU model and LSTM-RNN methods gain report considerable accomplishment. Nonetheless, the AHAODL-ID approach depicts its higher achievement with a decreased CT of 0.71s. Thus, the AHAODL-ID approach can be employed to detect intrusions in the VANET efficiently.

5. Conclusion

In this article, an AHAODL-ID technique on VANET is developed.

The AHAODL-ID technique exploits feature selection with a hyperparameter selection model for detecting intrusions in the VANET. For data preprocessing, Z-score normalization is employed to scale the input data. Next, the AHA-based feature selection approach is executed to choose an optimum feature subset.

Meanwhile, the BiLSTM method is used to detect diverse intrusion types. Lastly, the hyperparameter election of the BiLSTM technique involves the design of the MRFO. The experimentation results of the AHAODL-ID method are assessed using a benchmark IDS dataset. The obtained values underlined the advanced achievement of the AHAODL-ID technique over other existing models.

References

[1] Fuad A. Ghaleb et al., "Ensemble-Based Hybrid Context-Aware Misbehavior Detection Model for Vehicular Ad Hoc Network," *Remote Sensing*, vol. 11, no. 23, pp. 1-29, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[2] Junwei Liang et al., "A Novel Intrusion Detection System for Vehicular Ad Hoc Networks (VANETs) Based on Differences of Traffic Flow and Position," *Applied Soft Computing*, vol. 75, pp. 712-727, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[3] Fuad A. Ghaleb et al., "Hybrid and Multifaceted Context-Aware Misbehavior Detection Model for Vehicular Ad Hoc Network," *IEEE Access*, vol. 7, pp. 159119-159140, 2019. [CrossRef] [Google Scholar] [Publisher Link]

[4] Sahrish Khan Tayyaba et al., "5G Vehicular Network Resource Management for Improving Radio Access Through Machine Learning," *IEEE Access*, vol. 8, pp. 6792- 6800, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[5] Dijiang Huang et al., "PACP: An Efficient Pseudonymous Authentication-Based Conditional Privacy Protocol for VANETs," *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 3, pp. 736-746, 2011. [CrossRef] [Google Scholar] [Publisher Link]

[6] Pooja Parameshwarappa, Zhiyuan Chen, and Aryya Gangopadhyay, "Analyzing Attack Strategies Against Rule-Based Intrusion Detection Systems," *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking*, Varanasi, India, pp. 1-4, 2018. [CrossRef] [Google Scholar] [Publisher Link]

[7] Junwei Liang, Maode Ma, and Xu Tan, "GaDQN-IDS: A Novel Self-Adaptive IDS for VANETs Based on Bayesian Game Theory and Deep Reinforcement Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 8, pp. 12724-12737, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[8] Vinoth Kumar Kalimuthu, and Thiruppathi Muthu, "Oppositional Coyote Optimization Based Feature Selection with Deep Learning Model for Intrusion Detection in Fog-Assisted Wireless Sensor Network," *Acta Montanistica Slovaca*, vol. 28, no. 2, pp. 496-508, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[9] Huseyin Polat, Muammer Turkoglu, and Onur Polat, "Deep Network Approach with Stacked Sparse Autoencoders in Detection of DDoS Attacks on SDN-Based VANET," *IET Communications*, vol. 14, no. 22, pp. 4089-4100, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[10] Farkhanda Zafar et al., "Carpooling in Connected and Autonomous Vehicles: Current Solutions and Future Directions," *ACM Computing Surveys*, vol. 54, no. 10s, pp. 1-36, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[11] Abizar Khalil et al., "Artificial Intelligence-Based Intrusion Detection System for V2V Communication in Vehicular Adhoc Networks," *Ain Shams Engineering Journal*, vol. 15, no. 4, pp. 1-9, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[12] Prakash Vijay Sontakke, and Nilkanth B. Chopade, "Optimized Deep Neural Model-Based Intrusion Detection and Mitigation System for Vehicular Ad-Hoc Network," *Cybernetics and Systems*, vol. 54, no. 7, pp. 985-1013, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[13] Amit Chougule et al., "Multibranch Reconstruction Error (MbRE) Intrusion Detection Architecture for Intelligent Edge-Based Policing in Vehicular Ad-Hoc Networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 24, no. 11, pp. 13068-13077, 2022. [CrossRef] [Google Scholar] [Publisher Link]

[14] Weiping Ding et al., "DeepSecDrive: An Explainable Deep Learning Framework for Real-Time Detection of Cyberattack in in-Vehicle Networks," *Information Sciences*, vol. 658, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[15] Ankit Manderna et al., "Vehicular Network Intrusion Detection Using a Cascaded Deep Learning Approach with Multi-Variant Metaheuristic," *Sensors*, vol. 23, no. 21, pp. 1-24, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[16] R. Reka et al., "Multi Head Self-Attention Gated Graph Convolutional Network Based Multi-Attack Intrusion Detection in MANET," *Computers & Security*, vol. 136, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] V. Karthik et al., "Residual Based Temporal Attention Convolutional Neural Network for Detection of Distributed Denial of Service Attacks in Software Defined Network Integrated Vehicular Adhoc Network," *International Journal of Network Management*, vol. 34, no. 3, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] K. Lakshmi Narayanan, and R. Naresh, "Detection and Prevention of Black Hole Attack Using Tree Hierarchical Deep Convolutional Neural Network and Enhanced Identity-Based Encryption in Vehicular Ad Hoc Network," *IEIE Transactions on Smart Processing & Computing*, vol. 13, no. 1, pp. 41-50, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Sara Amaouche et al., "FSCB-IDS: Feature Selection and Minority Class Balancing for Attacks Detection in VANETs," *Applied Sciences*, vol. 13, no. 13, pp. 1-21, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Sehba Masood, and Aasim Zafar, "Deep-Efficient-Guard: Securing Wireless Ad Hoc Networks via Graph Neural Network," *International Journal of Information Technology*, pp. 1-16, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[21] Muhammad Ali Imron, and Budi Prasetyo, "Improving Algorithm Accuracy K-Nearest Neighbor Using Z-Score Normalization and Particle Swarm Optimization to Predict Customer Churn," *Journal of Soft Computing Exploration*, vol. 1, no. 1, pp. 56-62, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[22] Anusha Sanampudi, and S. Srinivasan, "Local Search Enhanced Optimal Inception-ResNet-v2 for Classification of Long-Term Lung Diseases in Post-COVID-19 Patients," *Automatika*, vol. 65, no. 2, pp. 473-482, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[23] Nabila Zrira et al., "Time Series Prediction of Sea Surface Temperature Based on BiLSTM Model with Attention Mechanism," *Journal of Sea Research*, vol. 198, pp. 1-13, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[24] Shibendu Mahata et al., "Reduced Order Infinite Impulse Response System Identification Using Manta Ray Foraging Optimization," *Alexandria Engineering Journal*, vol. 87, pp. 448-477, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[25] Nour Moustafa, "A New Distributed Architecture for Evaluating AI-Based Security Systems at the Edge: Network TON_IoT Datasets," *Sustainable Cities and Society*, vol. 72, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[26] Rabie A. Ramadan et al., "Internet of Drones Intrusion Detection Using Deep Learning," *Electronics*, vol. 10, no. 21, pp. 1-28, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[27] Abdallah R. Gad, Ahmed A. Nashat, and Tamer M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," *IEEE Access*, vol. 9, pp. 142206-142217, 2021. [CrossRef] [Google Scholar] [Publisher Link]