SSRG International Journal of Electrical and Electronics Engineering
ISSN: 2348-8379/ https://doi.org/10.14445/23488379/IJEEE-V10I1P117

Volume 10 Issue 1, 176-186, January 2023

Original Article

Brain Storm Optimization with Deep Learning-Based Intrusion Detection System in Vehicular Adhoc Networks

R. Mohan[1], G. Prabakaran[2], T. Priyaradhikadevi[3]

[1,2]*Department of Computer Science and Engineering Annamalai University Annamalai Nagar, Chidambaram.*
[3]*Department of Computer Science and Engineering Mailam Engineering College, Mailam, Villupuram.*

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

Received: 10 December 2022 Revised: 12 January 2023 Accepted: 20 January 2023 Published: 29 January 2023

Abstract *- Vehicular adhoc network (VANET) is an empowering technology in recent transportation systems for offering valuable information and safety, but prone to several attacks, such as active interference and passive eavesdropping. Intrusion detection systems (IDSs) are significant devices that alleviate threats by detecting malicious actions. In addition, the collaborations between vehicles in VANETs could enhance the accuracy level of detection by interacting with their experiences among nodes. So, distributed ML becomes a highly suitable structure for designing scalable and applicable collaborative detection techniques over VANETs. Therefore, this paper proposes a brain storm optimization with a deep learning-based intrusion detection system (BSODL-IDS) for VANET. In the presented BSODL-IDS technique, a primary stage of BSO based feature selection process is involved in it. For intrusion detection, the BSODL-IDS model exploits the long short-term memory recurrent neural network (LSTM-RNN) model. The Adamax optimizer is utilized at the last stage for the hyperparameter tuning of the LSTM-RNN technique. The experimental validation on the benchmark dataset illustrates the BSODL-IDS method's supremacy over other DL approaches.*

Keywords *- Intrusion detection, Security, VANET, Deep learning, Feature selection, Parameter tuning.*

1. Introduction

Due to the tremendous growth of autonomous vehicles and an increasing number of vehicles, road safety has become a prominent problem in recent times [1]. Vehicular adhoc network (VANET) offers a transmission network for disseminating road services, safety-oriented information, navigation, and traffic management. But VANETs are prone to several attacks like active interfering and passive eavesdropping [2]. An adversary could intrude on a particular vehicle, mimic its identity, and deliver false warnings that disturb highway traffic [3]. But the VANET environment can be very dynamic, having quickly changing topologies where the density and speeds of vehicles have been changing, which hampers the continuous interchange of the data between vehicles [4]. Cybercriminals could disturb VANET functions and launch several kinds of attacks resulting in disturbance of the network activities, accidents, and congestion [5]. Thus, security becomes highly important in VANET because of its possible significance to economic activities and people's lives.

Currently, many endeavours have taken place for designing IDSs for VANET. Various techniques of IDS solutions were recommended for VANETs, including hybrid-based, anomaly-based, signature-based, and so on [6]. Diverse IDS structures were devised: collaborative, centralized, decentralized, distributed, cluster, and cooperative IDSs. But because of the cooperative nature of VANET, several newly devised IDSs depend on the association among vehicles for detecting interlopers [7]. Vehicles share their relevant knowledge towards the detection experiences in the cooperative IDS (CIDS) for helping automobiles in the vicinity detect intruders with higher accuracy [8]. For example, researchers discovered distributed machine learning (ML), which can be a suitable scaling technique for collaborative recognition in VANETs and can be utilized to enhance recognition accuracy through the classification of adversarial behaviors utilizing local data and sharing knowledge [9]. Additionally, prevailing CIDS methods depend on the majority win scheme (voting system). Inappropriately, this method can be prone to colluding attacks like a botnet, in which adversaries affiliate to forward misleading data and disturb the IDS system [27].

The study proposes a brainstorm optimization with a deep learning-based intrusion detection system (BSODL-IDS) technique for VANET. In the presented BSODL-IDS technique, a primary stage of the BSO-based feature selection process is involved. The BSODL-IDS model exploits the long short-term memory recurrent neural

network (LSTM-RNN) technique for intrusion detection. The Adamax optimizer is utilized at the last stage for the hyperparameter tuning of the LSTM-RNN technique. The experimental validation on the benchmark dataset illustrates the BSODL-IDS technique's supremacy over other DL approaches.

2. Literature Review

Subba et al. [11] attempt to resolve the intrusion issue by suggesting a multi-layer game theory concept-based ID architecture and a new clustering protocol for VANET. The transmission overhead of the IDS can be decreased based on the set of rules and a light weighted NN-related classification model to detect mischievous vehicles. Zhang and Zhu [28] advise a privacy-preserving ML-based collaborative IDS (PML-CIDS) model for the VANET. Then, differential privacy is employed to capture the presented model's privacy system. The author in [13] proposed a trust-based CIDS (T-BICDS) method. Liang et al. [14] developed a novel IDS used properly in dynamic and wireless networks, such as VANET. It primarily comprises a new feature extracting and classification algorithm related to a better growing hierarchical self-organizing map (I-GHSOM) for IDS in VANET. The suggested method rapidly extracts the feature from vehicle messages for IDS testing and training. In this work, two major characteristics involving the difference in location and traffic flow are extracted.

In[15-17], proposed a distributed collaborative IDS-based invariant named DCDIV to recognize deceived attacks in VANET. Initially, the author developed a CIDS architecture to calculate and store a great deal of information and quickly track and gather information. Then, considering the high-reliability requirement and the strict delay limitation of data communication between vehicles, a reputation-based cooperative transmission system is employed to determine a reliable and stable transmission channel, whereby a new vehicle CH selection model. Zeng et al. [29] proposed a DL-based end-to-end IDS to identify malware traffic automatically for On-Board Unit (OBU). Unlike preceding IDSs, the presented model needs raw traffic rather than private data feature extracted by humans.

3. The Proposed Model

In this paper, a new BSODL-IDS method has been proposed to recognize and classify intrusions for VANET. In the presented BSODL-IDS technique, a primary stage of the BSO-based feature selection process is involved. For intrusion detection, the BSODL-IDS model exploits the LSTM-RNN model. The Adamax optimizer is utilized at the last stage for the hyperparameter tuning of the LSTM-RNN method. Fig. 1 depicts the working process of the BSODL-IDS algorithm.

3.1. Algorithmic Process of BSO-based Feature Selection

Primarily, the presented BSODL-IDS technique undergoes BSO based feature selection process. The BSO is an MH technique that simulates the process of creating innovative ideas from a group of ideas during the discussion of a company meeting [33]. During that meeting, the individual person is clustered into working groups, and the moderator carefully chooses the idea based on predetermined conditions. Similar to other MH algorithms, the BSO initiates by producing a random population that has s a group of N_{pop} Solutions. Next, clustering and generating new solutions are the two major phases used in this algorithm for updating the population [18]:

3.1.1. Clustering Stage

During the clustering phase, the solution of the early population can be clustered into k groups (that characterize the worked groups) with the k-means algorithm. In the updating method, the cluster can be maintained, and the novel solution changes the center of all the clusters and whether that solution has the best fitness function when compared to the center.

3.1.2. Generate a New Solution Stage

During the generating phase, an innovative idea is created based on the individual cluster, whether the probability $P_1 > \delta_1$; or else, it is created from more than one cluster as follows:

$$y_s = \begin{cases} y_i & if\ P_1 > \delta_1 \\ \alpha_1 \times y_{i1} + (1 - \alpha_2) \times y_{i2} & two\ clusters' \end{cases} \quad (1)$$

In Eq. (1), y_s and y_i represents the chosen solution and $i\text{-}th$ solution in the population, correspondingly, α_1, α_2, and δ_1 shows the arbitrary number that belongs to the range [0,1]. Eq (1) is crucial to the BSO approach as its exploration capability is improved once the novel solution is made from more than one cluster. The exploitation capability is also enhanced by constructing a novel solution from a single cluster.

The novel solution (y_n) is produced afterwards, choosing y_s as follow:

$$y_n = y_s + \alpha_3 \times \quad (2)$$

In Eq. (2), α_3 and γ represents an arbitrary integer and a control parameter for the convergence rate correspondingly. γ is upgraded by the following equation [19]:

$$\gamma = \alpha_4 \times logsig\left[\frac{0.5 \times (iter_{\max} - iter_c)}{\tau}\right], \quad (3)$$

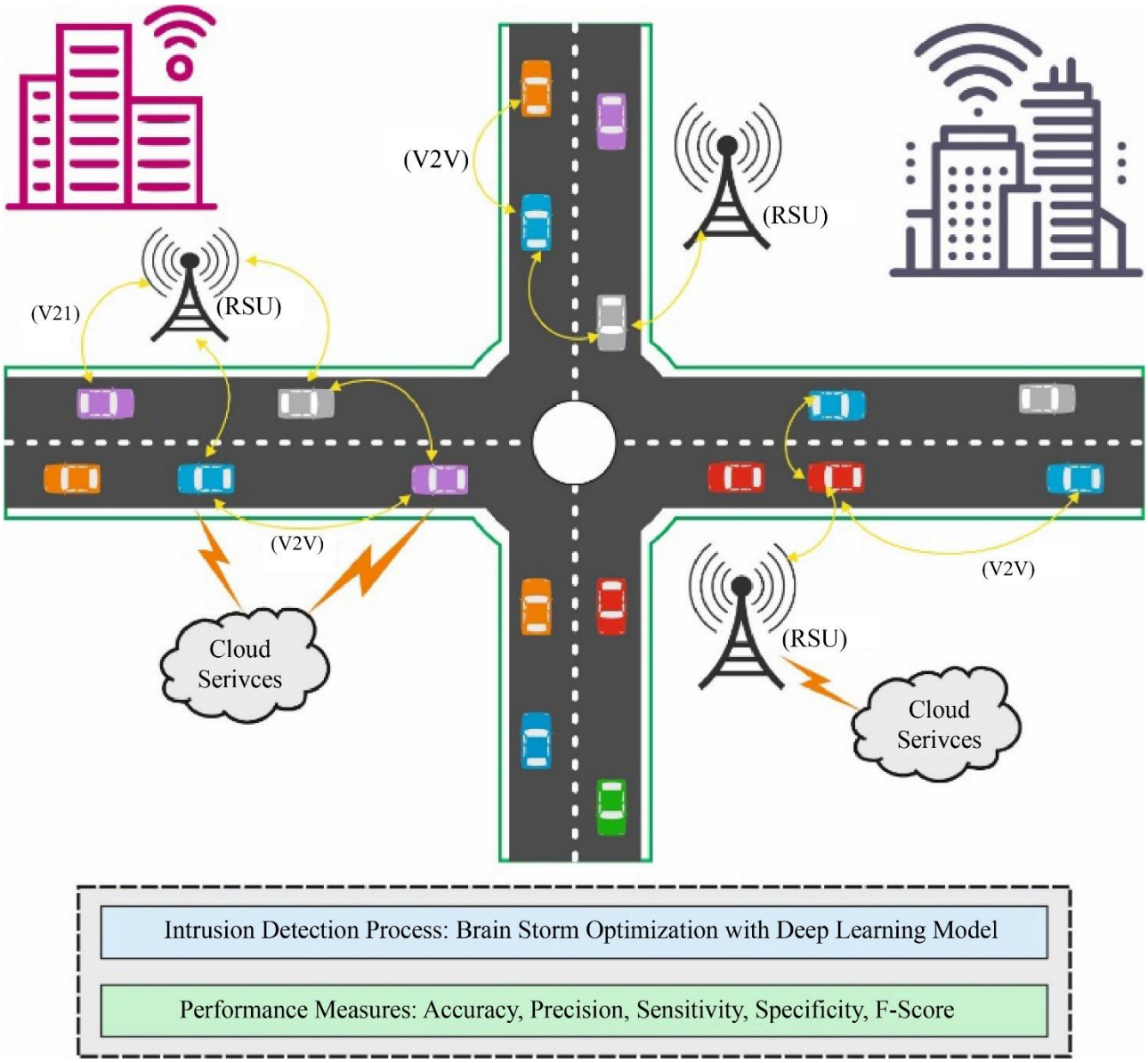

Fig. 1 Overall process of BSODL-IDS technique

Algorithm 1: The major steps of the BSO method.
1: Initialization: Generate a population that comprises a group N_{pop} ideas with D-dimension vector. 2: repeat 3: Clustering phase: Split N_{pop} ideas into m groups through k-means. 4: Generating new solution phase: Select one/two cluster(s) and create a novel idea. 5: Selection: choose the better idea from the newer and the older ideas for updating the population. 6: Evaluation: Evaluate the population by calculating the fitness function for all the ideas. 7: until Stopping, criteria are met

Now, $iter_c$, $iter_{\max}$, and $\alpha_4 \in [0,1]$ represents the existing iteration, the maximal iteration count, and an arbitrary integer, correspondingly. The logsig is a logarithmic sigmoid transfer function that enhances the BSO technique's global and local searching abilities. τ denotes a variable, viz., utilized for changing the slope of the logsig function, which is predetermined. The next step in this stage is to calculate the fitness function for y_n and compared with the fitness function for the cluster center, and the best one is kept as a center. The procedure of the BSO approach is shown in Algorithm 1.

In this work, the fitness function is used to balance between the classification accuracy (maximum) and the number of features selected in each solution (minimum)

acquired with the selected feature; Eq. (4) characterizes the fitness function to calculate a solution.

$$Fitness = \alpha\gamma_R(D) + \beta\frac{|R|}{|C|} \quad (4)$$

Here, $\gamma_R(D)$ signifies the classifier error rate of provided classifier (LSTM RNN classification). $|R|$ indicates the cardinality of the selected set, and $|C|$ denotes the overall feature count, α and β show the two parameters corresponding to the subset length and classification quality. $\in$ [1,0] and $\beta = 1 - \alpha$.

3.2. Intrusion Detection using Optimal LSTM-RNN Model

The BSODL-IDS model exploits the LSTM-RNN model for the IDS technique. LSTM extended the RNN with a memory cell, rather than a recurrent unit, to ease the learning of temporal relationships on the long timescale and store an output dataset [31]. The important novelty of LSTM is its memory cells that basically perform as an accumulator of the state database. LSTM uses the conception of gating-a model based on the element-wise multiplication of the input that determines the behavior of memory cells. LSTM upgrades the cell state based on the activation of the gate. One benefit of utilizing the memory cell and gate for controlling data flow is that the gradient is stocked in the cell and be avoided disappearing, a crucial problem for the vanilla RNN method. The LSTM is fed as input to distinct gates once the process is implemented on the memory cell: reset (forget) gate, write (input) gate, and read (output) gate [21]. The activation of the LSTM unit is evaluated as in the RNN. The calculation of h_t the hidden value of LSTM is upgraded at every time step t. The vector presentation (vector denotes each unit in a layer) of the update of the LSTM layer is represented as the output gate 0_t, input gate i_t, forget gate f_t, hidden state h_t, and memory cell c_t. Fig. 2 demonstrates the stricture of the LSTM-RNN technique.

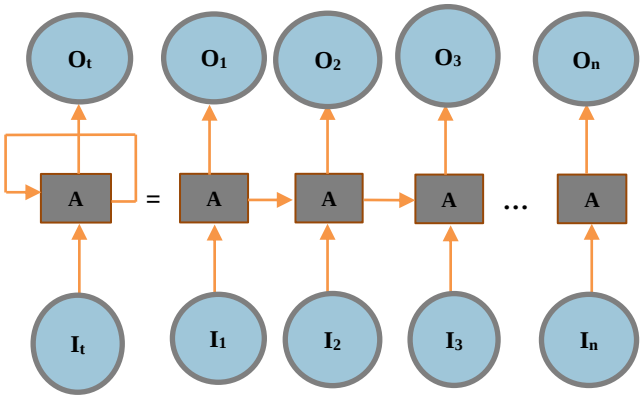

Fig. 2 LSTM-RNN structure

There has been progressing in the study of LSTM; hidden units with different connections within the memory unit were developed. Permitting the hyperbolic tangent nonlinearity squashes input to $[-1;1]$ interval and the sigmoid nonlinearity that squashes the input to $[0;1]$ interval, LSTM upgrade for t time steps provided inputs x_t, h_{t-1}, and c_{t-1} are:

$$\begin{aligned} i_t &= \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \\ f_t &= \sigma(W_{xi}x_t + W_{hf}h_{t-1} + b_f) \\ o_t &= \sigma(W_{xo}x_t + W_{ho}h_{t-1} + b_o) \\ g_t &= \phi(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \\ c_t &= f_t \odot c_t + i_t \odot g_t \\ h_t &= 0_t \odot \phi)(c_t) \end{aligned} \quad (5)$$

Now i, f, 0, c, and g are, correspondingly, the input, forget, output, cell activation, and input modulation gate vectors, and they have a similar size to h vector that describes the hidden value. Terms signify a component-wise application of the sigmoid function. Term x_t, is the input to the layer of memory cells at t time ; W_{xi}, W_{xf}, W_{xo}, W_{xc}, W_{hi}, W_{hf}, W_{ho}, W_{hc} indicates weight matrixes, with subscripts signifying from-to relationship (the hidden-input gate matrixes, input-input gate matrixes, etc.), b_i, b_f, b_o, b_c denotes bias vector; ϕ refers to a component-wise application of $tanh$ operation; $\odot$ signifies component-wise multiplication.

Finally, for the hyperparameter tuning of the LSTM-RNN method, the Adamax optimizer (AO) technique is utilized. Yoshua and Yann introduced the Adam as regarded as a common methodology for the first moment of the gradient-based optimization of stochastic objective function [34]. The prominent benefit of this algorithm is that it can adaptably finetune the learning rate parameter in the training model. The Adam depends on data acquired from the average of the second moment of the gradient. Also, an exponentially decaying average of the past gradient was used in this algorithm. Moreover, the initial set of three hyperparameters was required in this optimizer: the step size α and ($\beta_1 = 0.9$ and $\beta_2 = 0.9999$)the two exponential decay rates. The optimized parameter of a neural computing mechanism is adapted once the gradient of model parameters is evaluated,

$$w_t = w_{t-1} - \alpha \times \frac{\widehat{m}}{\sqrt{v_t} + \mathcal{E}'} \quad (6)$$

Let $\widehat{m_t}$ and $\hat{v}_t$ be the bias-corrected first and second raw moment estimate, correspondingly. The Adamax is an alternative to Adam, whose update rule for model weight scales their gradient inversely proportionate towards a L^p norm of the present and prior gradients. Then, we upgraded the neural network weight using the following equation:

$$w_t = w_{t-1} - \frac{\alpha}{1-\beta_1^t} \times \frac{m_t}{\max(\beta_2 u_{t-1}, |g_t|)}, \quad (7)$$

Now $u_t = 0$ at $t = 0; u_t$ is the biased second raw moment estimate.

4. Results and Discussion

The dataset tests the experimental analysis of the BSODL-IDS method. Fig. 3 exemplifies the confusion matrices formed by the BSODL-IDS technique. On 80% of TRS, the BSODL-IDS technique has recognized 0 instances under U2r class, 2 instances under R2l class, 36204 instances in the DoS class, 53027 instances under the normal class, and 8454 instances under the probe class. Furthermore, on 20% of TSS, the BSODL-ID method has recognized 8891 instances in the DoS class, 0 instances under U2r class, 2 instances under R2l class, and 13459 instances under the normal and 2076 instances under the probe class. Similarly, on 70% of TRS, the BSODL-IDS technique has recognized 30483 instances in the DoS class, 0 instances under U2r class, 2 instances under R2l class, 44706 instances under the normal class, and 7140 instances under the probe class. Also, on 30% of TSS, the BSODL-IDS method has recognized 12926 instances in the DoS class, 0 instances under U2r class, 2 instances under R2l class, 19199 instances under the normal class, and 3115 instances under the probe class.

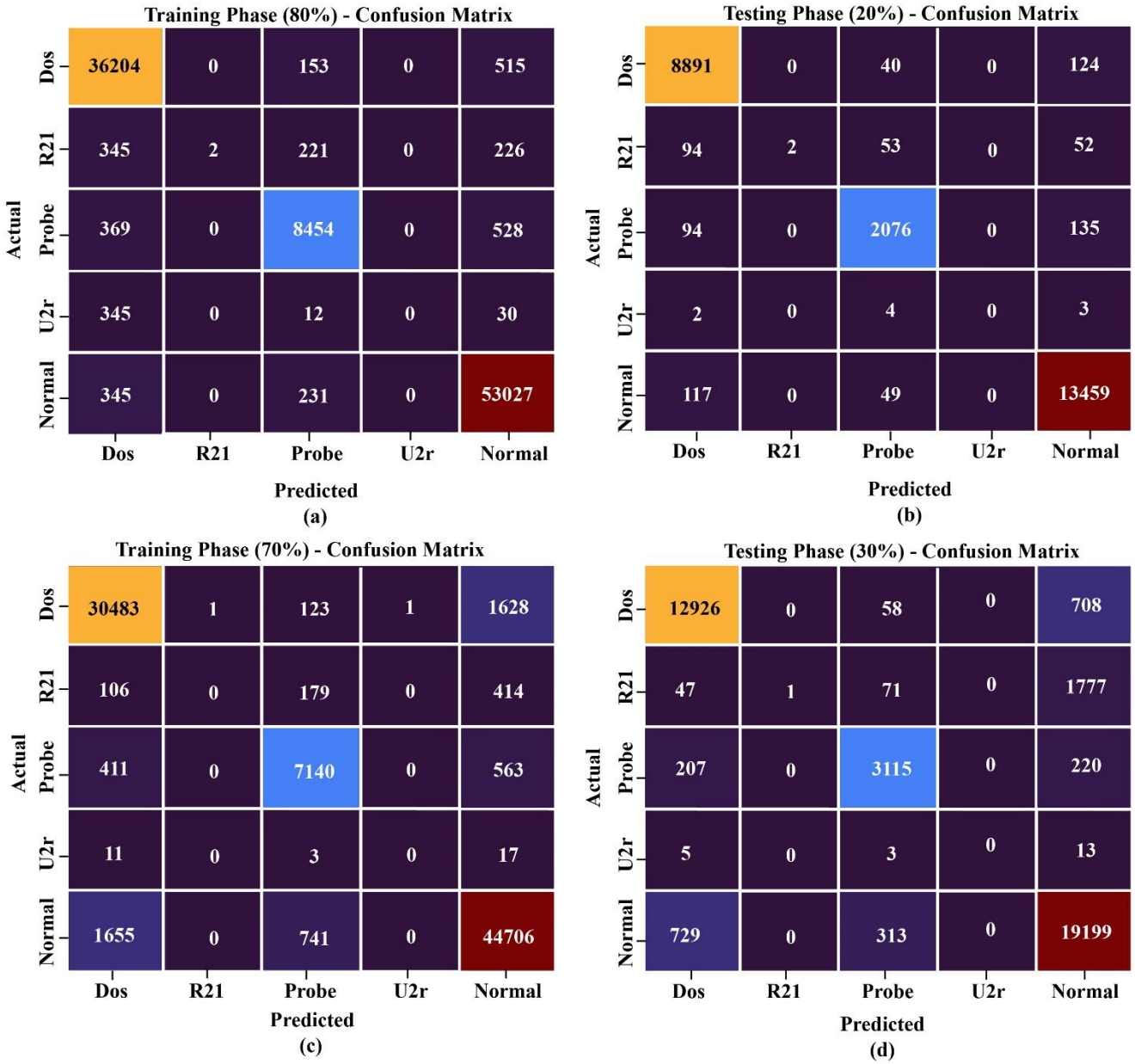

Fig. 3 Confusion matrix BSODL-IDS methodology (a) 80% of TRS, (b) 20% of TSS, (c) 70% of TRS, and (d) 30% of TSS

Table 1 offers the overall IDS outcomes of the BSODL-IDS technique on the TRS/TSS dataset of 80:20. Fig. 4 highlights the classification results given by the BSODL-IDS method on 80% of the TRS dataset. The experimental outcome demonstrates that the BSODL-IDS method has demonstrated superior outcomes under each class label. For

example, in the DoS class, the BSODL-IDS system has obtained $accu_y$ of 98.17%, $prec_n$ of 96.86%, $sens_y$ of 98.19%, $spec_y$ of 98.16%, and F_{score} of 97.52%. In the meantime, in R21 class, the BSODL-IDS technique has gained $accu_y$ of 99.21%, $prec_n$ of 100%, $sens_y$ of 00.25%, $spec_y$ of 100%, and F_{score} of 00.50%. In parallel, on Probe class, the BSODL-IDS algorithm has acquired $accu_y$ of 98.50%, $prec_n$ of 93.20%, $sens_y$ of 90.41%, $spec_y$ of 99.33%, and F_{score} of 91.78%. Finally, on U2r class, the BSODL-IDS algorithm has acquired $accu_y$ of 99.96%, $prec_n$ of 00.00%, $sens_y$ of 00.00%, $spec_y$ of 100%, and F_{score} of 00.00%.

Table 1. Results of the BSODL-IDS method on 80:20 of the TRS/TSS dataset

Training / Testing (80:20)					
Labels	**Acc_y**	**$Prec_n$**	**$Sens_y$**	**$Spec_y$**	**F_{score}**
Training Phase					
Dos	98.17	96.86	98.19	98.16	97.52
R2l	99.21	100.00	00.25	100.00	00.50
Probe	98.50	93.20	90.41	99.33	91.78
U2r	99.96	00.00	00.00	100.00	00.00
Normal	98.03	97.61	98.71	97.24	98.16
Average	98.77	77.53	57.51	98.95	57.59
Testing Phase					
Dos	98.13	96.66	98.19	98.10	97.42
R2l	99.21	100.00	01.00	100.00	01.97
Probe	98.51	93.43	90.07	99.36	91.72
U2r	99.96	00.00	00.00	100.00	00.00
Normal	98.09	97.72	98.78	97.29	98.25
Average	98.78	77.56	57.61	98.95	57.87

Fig. 5 highpoints the classification results given by the BSODL-IDS methodology on 20% of the TSS dataset. The experimental outcome demonstrates the BSODL-IDS methodology has demonstrated superior outcomes under each class label. For example, in the DoS class, the BSODL-IDS algorithm has achieved $accu_y$ of 98.13%, $prec_n$ of 96.66%, $sens_y$ of 98.19%, $spec_y$ of 98.10%, and F_{score} of 97.42%. Finally, in U2r class, the BSODL-IDS method has gained $accu_y$ of 99.96%, $prec_n$ of 00.00%, $sens_y$ of 00.00%, $spec_y$ of 100%, and F_{score} of 00.00%.

Table 2 presents the overall IDS outcomes of the BSODL-IDS technique on the TRS/TSS dataset of 70:30. Fig. 6 represents the classification outcome given by the BSODL-IDS method on 70% of the TRS dataset.

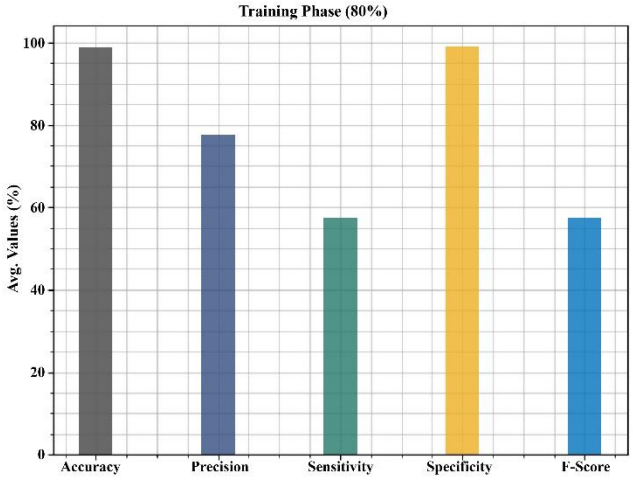

Fig. 4 Average results of the BSODL-IDS method under 80% of the TRS dataset

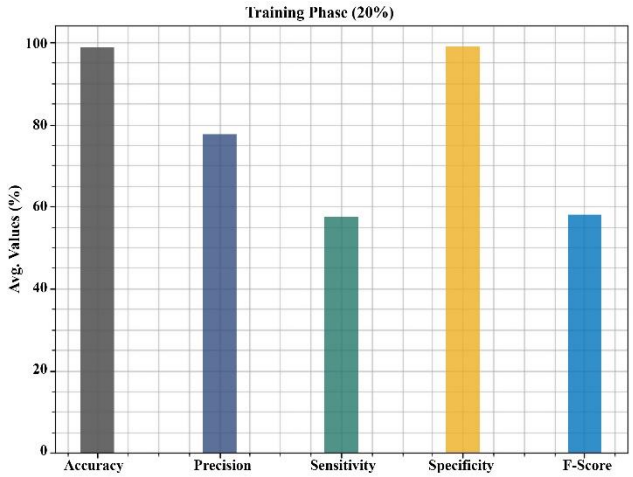

Fig. 5 Average analysis of the BSODL-IDS method under 20% of the TSS dataset

Table 2. Results of BSODL-IDS methodology with different classes on 70:30 of the TRS/TSS dataset

Training / Testing (70:30)					
Labels	**Acc_y**	**$Prec_n$**	**$Sens_y$**	**$Spec_y$**	**F_{score}**
Training Phase					
Dos	95.54	93.32	94.56	96.10	93.94
R2l	99.21	00.00	00.00	100.00	00.00
Probe	97.71	87.22	88.00	98.69	87.61
U2r	99.96	00.00	00.00	100.00	00.00
Normal	94.31	94.46	94.91	93.62	94.69
Average	**97.35**	**55.00**	**55.49**	**97.68**	**55.25**
Testing Phase					
Dos	95.36	92.90	94.41	95.90	93.65
R2l	99.22	100.00	00.34	100.00	00.67
Probe	97.69	87.50	87.94	98.70	87.72
U2r	99.94	00.00	00.00	100.00	00.00
Normal	94.28	94.50	94.85	93.63	94.67
Average	**97.30**	**74.98**	**55.51**	**97.65**	**55.34**

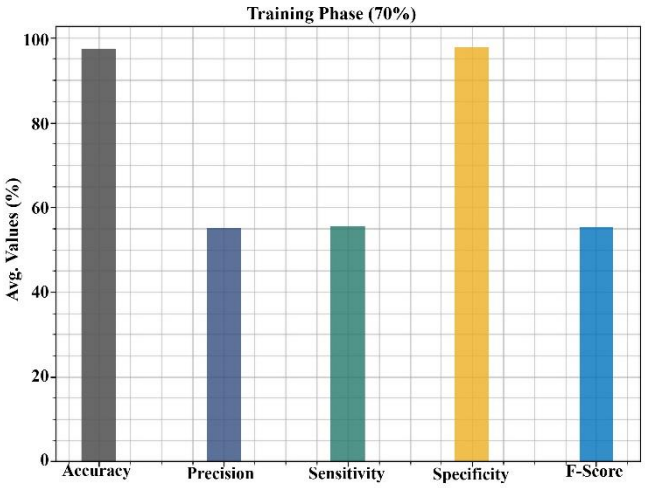

Fig. 6 Average results of the BSODL-IDS method under 70% of the TRS dataset

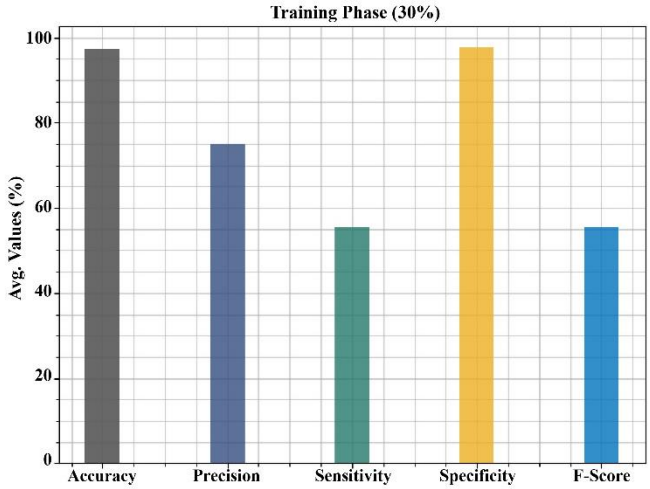

Fig. 7 Average analysis of BSODL-IDS approach under 30% of the TSS dataset

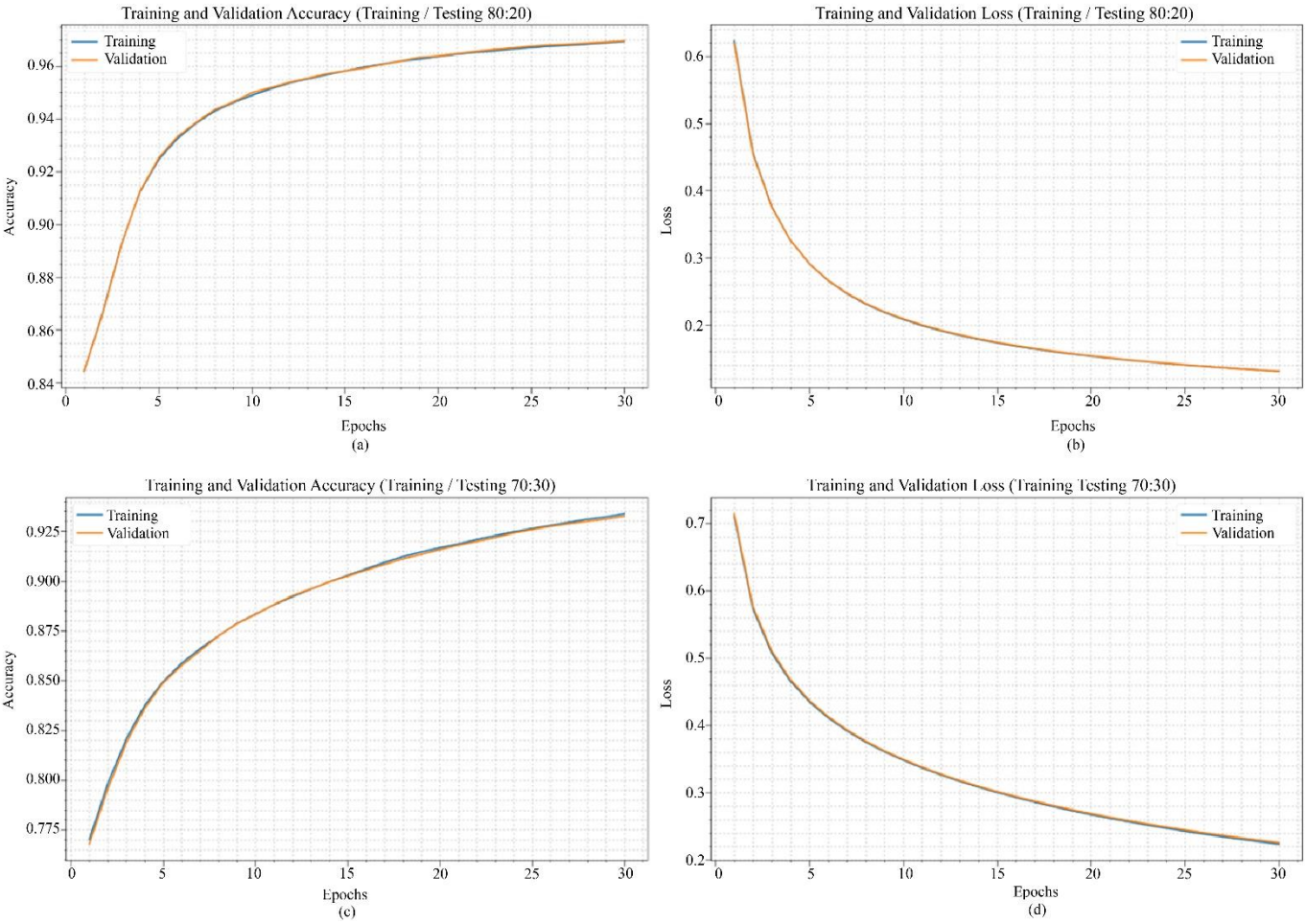

Fig. 8 (a and b) Graph of Accuracy and Loss- 80:20 of TRS/TSS dataset (c and d) Graph of Accuracy and Loss-70:30 of TRS/TSS dataset

The experimental outcomes showed that the BSODL-IDS algorithm demonstrated improved outcomes under each class label. For example, in the DoS class, the BSODL-IDS system has achieved $accu_y$ of 95.54%, $prec_n$ of 93.32%, $sens_y$ of 94.56%, $spec_y$ of 96.10%, and F_{score} of 93.94%. In the meantime, on R21 class, the BSODL-IDS algorithm has

achieved $accu_y$ of 99.21%, $prec_n$ of 00.00%, $sens_y$ of 01.00%, $spec_y$ of 100%, and F_{score} of 00.00%. Simultaneously, in Probe class, the BSODL-IDS approach has reached $accu_y$ of 97.71%, $prec_n$ of 87.22%, $sens_y$ of 88%, $spec_y$ of 98.69%, and F_{score} of 87.61%. Finally, in U2r class, the BSODL-IDS approach has acquired $accu_y$ of 99.96%, $prec_n$ of 00.00%, $sens_y$ of 00.00%, $spec_y$ of 100%, and F_{score} of 00.00%.

Fig. 7 describes the classification outcome given by the BSODL-IDS technique on 30% of the TSS dataset. The results exemplified the BSODL-IDS algorithm has demonstrated superior outcomes under each class label. For example, in the DoS class, the BSODL-IDS system has reached $accu_y$ of 95.36%, $prec_n$ of 92.90%, $sens_y$ of 94.41%, $spec_y$ of 95.90%, and F_{score} of 93.65%. In the meantime, in R21 class, the BSODL-IDS methodology has attained $accu_y$ of 99.22%, $prec_n$ of 100%, $sens_y$ of 00.34%, $spec_y$ of 100%, and F_{score} of 00.67%. Concurrently, in Probe class, the BSODL-IDS technique has gained $accu_y$ of 97.69%, $prec_n$ of 87.50%, $sens_y$ of 87.94%, $spec_y$ of 98.70%, and F_{score} of 87.72%. At last, in U2r class, the BSODL-IDS approach has reached $accu_y$ of 99.94%, $prec_n$ of 00.00%, $sens_y$ of 00.00%, $spec_y$ of 100%, and F_{score} of 00.00%.

Fig. 8 offers the accuracy and loss graph analysis of the BSODL-IDS method under 80:20 and 70:30 of the TRS/TSS dataset. The experimental outcome reveals that the loss value tends to decrease, and the accuracy value tends to rise in epoch count. Noticeably, validation accuracy is maximum, and the training loss is minimum on the test dataset.

Fig. 9 demonstrates the precision-recall and ROC curve examination of the BSODL-IDS model under 80:20 and 70:30 of the TRS/TSS dataset. The figure indicates the BSODL-IDS method has obtained maximum precision-recall and ROC values on the classification distinct class labels.

Table 3 and Fig. 10 exhibit the $accu_y$ outcomes of the BSODL-IDS method with other recent approaches [12,22-26,32]. The experimental outcome shows that the NB Tree, Random Tree, and J48 models have reached the least $accu_y$ of 82.02%, 81.59%, and 81.05% correspondingly.

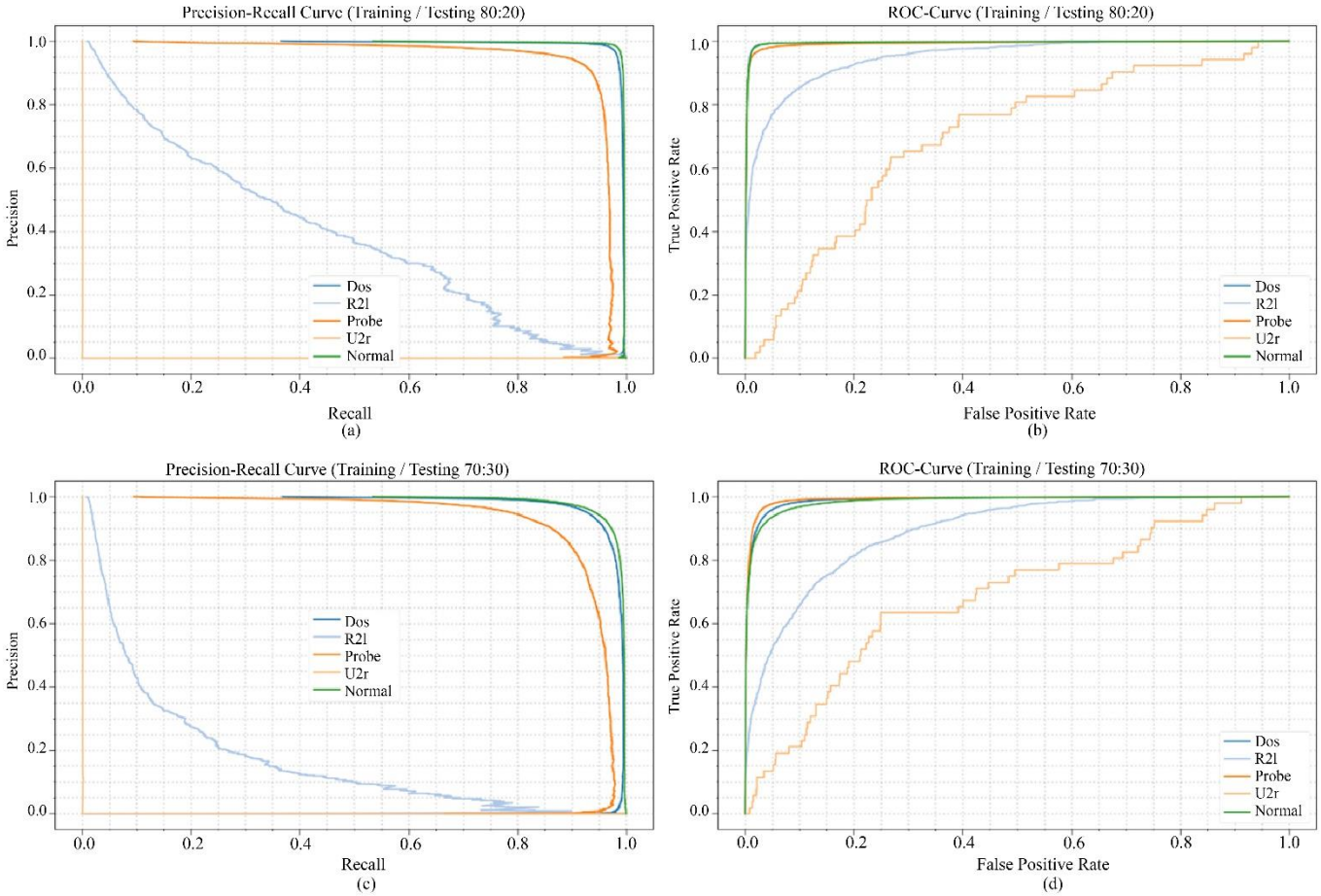

Fig. 9 (a and b) Graph of Precision-recall and ROC-80:20 of TRS/TSS dataset (c and d) Graph of Precision-recall and ROC-70:30 of TRS/TSS dataset

Table 3. Comparison study of the BSODL-IDS model

Method	Accuracy (%)
BSODL-IDS	98.78
NB Tree	82.02
Random Tree	81.59
j48	81.05
Cuckoo Optimization	96.88
DNN-SVM	92.03
DNN	91.40
PCA-DNN	93.80

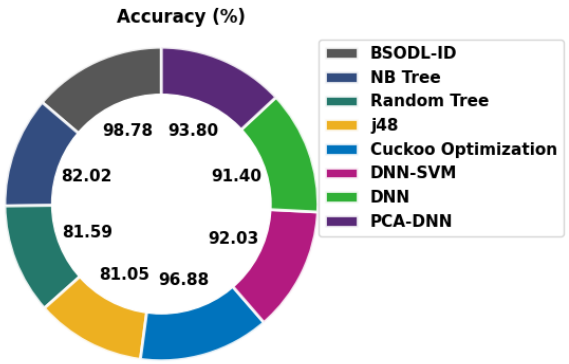

Fig. 10 Comparative analysis of BSODL-IDS approach with recent algorithms

Next, the DNN-SVM, DNN, and PCA-DNN models have resulted in moderately improved $accu_y$ of 92.03%, 91.40%, and 93.80% correspondingly. At the same time, the cuckoo optimization method has accomplished near optimal $accu_y$ of 96.88%. However, the BSODL-IDS model shows higher $accu_y$ of 98.78% and ensuring better performance on VANET security.

5. Conclusion

In this paper, a new BSODL-IDS method has been proposed to recognize and classify intrusions for VANET. In the presented BSODL-IDS technique, a primary stage of the BSO-based feature selection process is involved. For intrusion detection, the BSODL-IDS technique exploits the LSTM-RNN method. The Adamax optimizer is utilized at the last stage for the hyperparameter tuning of the LSTM-RNN model. The experimental validation on the benchmark dataset illustrates the supremacy of the BSODL-IDS method over other DL algorithms. As the comparison result highlights the betterment of the BSODL-IDS technique, it can be effectually utilized for intrusion detection and accomplishing security in VANET. In future, the performance of the BSODL-IDS algorithm can be boosted using feature reduction and clustering approaches in VANET.

References

[1] Ahmed A. Aboelfottoh, and Marianne A. Azer, “Intrusion Detection in VANETs and ACVs using Deep Learning,” *2nd International Mobile, Intelligent, and Ubiquitous Computing Conference (MIUCC)*, pp. 241-245, 2022. *Crossref*, https://doi.org/10.1109/miucc55081.2022.9781691

[2] Hind Bangui, Mouzhi Ge, and Barbora Buhnova, “A Hybrid Machine Learning Model for Intrusion Detection in VANET,” *Computing*, vol. 104, no. 3, pp.503-531, 2022. *Crossref*, https://doi.org/10.1007/S00607-021-01001-0

[3] Fábio Gonçalves et al., “A Systematic Review on Intelligent Intrusion Detection Systems for Vanets,” *2019 11th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, pp. 1-10, 2019. *Crossref*, https://doi.org/10.1109/icumt48472.2019.8970942

[4] Jiangang Shu et al., “Collaborative Intrusion Detection for Vanets: A Deep Learning-Based Distributed SDN Approach,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 7, pp. 4519-4530, 2021. *Crossref*, https://doi.org/10.1109/TITS.2020.3027390

[5] Viacheslav Belenko, Vasiliy Krundyshev, and Maxim Kalinin, “Synthetic Datasets Generation for Intrusion Detection in VANET,” *in Proceedings of the 11th International Conference on Security of Information and Networks*, pp. 1-6, 2018. *Crossref*, https://doi.org/10.1145/3264437.3264479

[6] Ila Naqvi, and Alka Chaudhary, “Intrusion Detection Using Soft Computing Techniques in Vanets,” *2021 9th International Conference on Reliability, Infocom Technologies and Optimization (Trends and Future Directions) (ICRITO)*, pp. 1-4, 2021. *Crossref*, https://doi.org/10.1109/icrito51393.2021.9596468

[7] J. A. Smitha et al., "Optimized Routing on Wireless Body Sensor Network Using Adaptive Lion Optimization Algorithm for IoT," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 12, pp. 189-197, 2022. *Crossref*, https://doi.org/10.14445/23488379/IJEEE-V9I12P117

[8] Farithkhan Abbas Ali, and E. D. Kanmani Ruby, "Clustering Metric Algorithm for Cost-Effective Routing in Flying Ad-Hoc Networks," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 12, pp. 101-108, 2022. *Crossref*, https://doi.org/10.14445/23488379/IJEEE-V9I12P108

[9] Karpaga Priya R et al., "A Novel Spider Swarm Optimized Energy and Security Aware Clustering Protocol for Smart Grid Wireless Sensor Network," *SSRG International Journal of Electrical and Electronics Engineering*, vol. 9, no. 10, pp. 19-26, 2022. *Crossref*, https://doi.org/10.14445/23488379/IJEEE-V9I10P104

[10] S. Gavaskar, E. Ramaraj, and R. Surendiran, "A Compressed Anti IP Spoofing Mechanism Using Cryptography," *IJCSNS International Journal of Computer Science and Network Security*, vol. 12, no. 11, pp.137-140, 2012.

[11] BasantSubba, Santosh Biswas, and SushantaKarmakar, "A Game Theory Based Multi Layered Intrusion Detection Framework for VANET," *Future Generation Computer Systems*, vol. 82, pp. 12-28, 2018. *Crossref*, https://doi.org/10.1016/j.future.2017.12.008

[12] S. Revathi, and A. Malathi, "Network Intrusion Detection Using Hybrid Simplified Swarm Optimization Technique," *International Journal of P2P Network Trends and Technology (IJPTT)*, vol. 3, no. 5, pp. 6-10, 2013.

[13] Tarak Nandy et al., "T-BCIDS: Trust-Based Collaborative Intrusion Detection System for VANET," *2020 National Conference on Emerging Trends on Sustainable Technology and Engineering Applications (NCETSTEA), IEEE*, 2020. *Crossref*, https://doi.org/10.1109/NCETSTEA48365.2020.9119934

[14] 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.

[15] Man Zhou et al., "Distributed Collaborative Intrusion Detection System for Vehicular Ad Hoc Networks Based on Invariant," *Computer Networks*, vol. 172, 2020. *Crossref*, https://doi.org/10.1016/j.comnet.2020.107174

[16] S.Navya Sai, and K. Kishoreraju, "Improved Privacy Preserving Decision Tree Approach for Network Intrusion Detection," *International Journal of Computer & Organization Trends*, vol. 6, no. 1, pp. 55-60, 2016.

[17] Mohammad Dawood Momand, Dr Vikas Thada, and Mr. Utpal Shrivastava, "Intrusion Detection System in IoT Network," *SSRG International Journal of Computer Science and Engineering* , vol. 7, no. 4, pp. 11-15, 2020. *Crossref*, https://doi.org/10.14445/23488387/IJCSE-V7I4P104

[18] Yuhui Shi, "Brain Storm Optimization Algorithm," *International Conference in Swarm Intelligence*, vol. 67285, 2011. *Crossref*, https://doi.org/10.1007/978-3-642-21515-5_36

[19] C. Narmatha et al., "A Hybrid Fuzzy Brain-Storm Optimization Algorithm for the Classification of Brain Tumor MRI Images," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-9, 2020. *Crossref*, https://doi.org/10.1007/s12652-020-02470-5

[20] Gunja Ambica , and Mrs.N.Rajeswari, "Robust Data Clustering Algorithms for Network Intrusion Detection," *International Journal of Computer & Organization Trends*, vol. 2, no. 5, pp. 6-11, 2012.

[21] Salah Bouktif et al., "Multi-Sequence LSTM-RNN Deep Learning and Metaheuristics for Electric Load Forecasting," *Energies*, vol. 13, no. 2, pp. 1-21. *Crossref*, https://doi.org/10.3390/en13020391

[22] Mr. Kulkarni Sagar S, and Prof. Kahate Sandip A., "Review of a Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguous and Dis-Contiguous System Call Patterns," *SSRG International Journal of Computer Science and Engineering* , vol. 2, no. 6, pp. 9-12, 2015. *Crossref*, https://doi.org/10.14445/23488387/IJCSE-V2I6P109

[23] Vulavabeti Raghunath Reddy et al., "Intrusion Detection and Monitoring Using IOT," *SSRG International Journal of Electronics and Communication Engineering*, vol. 5, no. 9, pp. 13-15, 2018. *Crossref*, https://doi.org/10.14445/23488549/IJECE-V5I9P103

[24] Ananth, C. A., et al., "Intrusion Detection System for Energy Efficient Cluster Based Vehicular Adhoc Networks," *International Journal of Advanced Computer Science and Applications*, vol. 12, no. 10, 2021. *Crossref*, https://doi.org/10.14569/IJACSA.2021.0121025

[25] Neelu Khare et al., "SMO-DNN: Spider Monkey Optimization and Deep Neural Network Hybrid Classifier Model for Intrusion Detection," *Electronics*, vol. 9, no. 4, pp. 1-18, 2020. *Crossref*, https://doi.org/10.3390/electronics9040692

[26] S.Kavitha et al., "Detecting Network Intrusion Based on Machine Learning Algorithms," *International Journal of P2P Network Trends and Technology*, vol. 10, no. 3, pp. 1-5, 2020.

[27] Praveensankar Manimaran, and Arun Raj Kumar P.NDNIDS, "An Intrusion Detection System for NDN Based VANET," *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring, IEEE.* pp. 1-5, 2020. *Crossref*, https://doi.org/10.1109/VTC2020-Spring48590.2020.9129365

[28] Tao Zhang, and Quanyan Zhu, "Distributed Privacy-Preserving Collaborative Intrusion Detection Systems for Vanets," *IEEE Transactions on Signal and Information Processing Over Networks*, vol. 4, no. 1, pp. 148-161, 2018. *Crossref*, https://doi.org/10.1109/TSIPN.2018.2801622

[29] Yi Zeng et al., "DeepVCM: A Deep Learning Based Intrusion Detection Method in VANET," *2019 IEEE 5th Intl Conference on Big Data Security on Cloud (Bigdata security), IEEE Intl Conference on High Performance and Smart Computing,(HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS*), pp. 288-293, 2019. *Crossref*, https://doi.org/10.1109/BigDataSecurity-HPSC-IDS.2019.00060

[30] Sanjeevaiah Kuraganti, and Jeevana Jyothi. P, "Privacy Preservation Approach Using K-Anonymity Chinese Remainder Theorem for Intrusion Detection," *International Journal of Computer & Organization Trends*, vol. 4, no. 5, pp. 31-38, 2014.

[31] FeiWang et al., "A Day-Ahead PV Power Forecasting Method Based on LSTM-RNN Model and Time Correlation Modification Under Partial Daily Pattern Prediction Framework," *Energy Conversion and Management*, vol. 212, 2020. *Crossref*, https://doi.org/10.1016/j.enconman.2020.112766

[32] Alzahrani, A.O., and Alenazi, M.J, "Designing a Network Intrusion Detection System Based on Machine Learning for Software Defined Networks, *Future Internet*, vol. 13, no. 5, pp. 1-18, 2021. *Crossref*, https://doi.org/10.3390/fi13050111

[33] Lianbo Ma et al., "Enhancing Learning Efficiency of Brain Storm Optimization via Orthogonal Learning Design," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 11, pp. 6723-6742. *Crossref*, https://doi.org/10.1109/TSMC.2020.2963943

[34] Noam Shazeer, and Mitchell Stern, "Adafactor: Adaptive Learning Rates With Sublinear Memory Cost," *International Conference on Machine Learning*, pp. 4596-4604, 2018.