IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

MODELING OF OPTIMAL SIAMESE NEURAL NETWORK BASED INTRUSION DETECTION SYSTEM FOR VEHICULAR ADHOC NETWORKS

[1]Pavithra T, [2]Nagabhushana B S

[1]VTU Research Scholar, [2]Dean Academic

[1]Department of Electronics and Communication Engineering,

[1]BMS College of Engineering, Bengaluru, India

Abstract: Vehicular Ad hoc Networks (VANETs) are an emerging network type that supports Intelligent Transportation Systems (ITS) and establishes structures that allow communication between road entity and enhance the advancement of novel services and applications focused on improving driving experience and rising road safety. Demanding features make it problematic to apply security systems, causing vulnerability simply discovered by aggressors. Such complexities are solved by utilizing deep-learning and machine-learning method. This paper presents an Optimal Siamese Neural Network-based IDS (OSNN-IDS) model for VANET. The major goal of this model is the recognition and classification of intrusions into distinct classes. To accomplish this, the presented model follows three major processes namely pre-processing, classification, and parameter tuning. To demonstrate the better performance of the OSNN-IDS method, an extensive range of simulations were carried out on CIC_IDS_2017 dataset and the comparative study has indicated the better performance of our algorithm compared to other models.

Index Terms **- intrusion detection system, Siamese neural networks, accuracy, vehicular Adhoc networks**

I. INTRODUCTION

The Vehicle ad-hoc networks (VANETs) aim at reducing transportation issues, like traffic congestion, car accidents, and pollution and it is becoming one of the trends in smart city implementation [1]. VANETs forms an effective collaboration among vehicles which permits both drivers and road operators to be intimated regarding the current traffic incident information through V2I and V2V communication [2]. It necessitates effectual traffic coordination which has 2 kinds of communications one is communication between vehicles and the infrastructure (V2I) and another one is communication among vehicles (V2V) [3]. Through V2I and V2V, the vehicles could show traffic management instructions and warning messages for increased road safety and enhancing the driving comfort, especially in urban area.

The VANET was not just utilized for improving road safety, but also used to ensure the demand to maintain the huge volume of sensing data produced from smart cities. Integrating VANET into mobile applications serves an important part in smart city, like distributing cloud applications thereby minimizing latency and offering high bandwidth. Vehicles serve as cloud node and devote computing sources to unite the data analysis with cloud computing (CC). In addition, VANET provides several cloud services (i.e., computing and storage) in the geographical location of vehicles, that diminishes latency and ignore congestion in the network [5]. To conquer security vulnerability, multiple intrusion detection systems (IDSs) were devised to secure UAV and VANET interactions [6]. In many prevailing detection solutions, adapted and advanced ML techniques were included for analysing traffic data, finding the attack type, and after differentiating the malicious nodes and benign nodes. To overcome security vulnerabilities, various IDSs were devised to secure VANET communications [7].

There exist 2 major kinds of cyber analytics from supporting of IDSs: 1) anomaly based and 2) misuse based. Misuse based technologies just detect familiar types of intrusions with a large necessity of storage [8]. Misuse-based techniques are outdated as novel kinds of intrusions were appearing every second. Formerly, many anomaly- related technologies were related to Machine Learning (ML) that necessitates private information for feature selection, denote the communication duration [9]. But many private information can be disguised with privacy protocols in recent times. Additionally, OBUs in a VANET is resource limited, that is ML-Based and misuse-based approaches were not preferred and implemented in vehicles [10]. A light-weight ID no need for private information is an urgent need for the VANET.

This paper presents an optimal Siamese neural network-based IDS (OSNN-IDS) model for VANET. The presented OSNN-IDS model focuses on the identification and classification of intrusions from the VANET environment. Initially, the OSNN-IDS model pre-processes the input data to make it compatible for further processes. Next, the OSNN-IDS model applies SNN and LSTM model for intrusion detection process. At last, Root Mean Square propagation (RMSprop) optimizer can be exploited for the parameter tuning of the SNN method, which helps in accomplishing enhanced classification performance and reduced classification

error rate. To demonstrate the better performance of the OSNN-IDS method, an extensive range of simulations were executed on CIC_IDS_2017 dataset and the results are assessed in terms of many evaluation measures. In short, the paper contribution is given as follows.

- An intelligent OSNN-IDS model technique comprising of pre-processing, SNN learning, LSTM classification, RMSProp hyperparameter tuning is presented for IDS in VANET. To the best of our knowledge, the OSNN-IDS model has never presented in the literature.
- Hyperparameter optimization of the SNN model using RMSProp helps to boost the predictive outcome of the OSNN-IDS model for unseen data.
- Validate the performance of the OSNN-IDS model on CICIDS_2017 dataset from Kaggle repository.

The rest of the paper is organized as follows. Section 2 renders a detailed review of existing IDS approaches for VANET. Next, section 3 introduces the presented OSNN-IDS method and section 4 provides a detailed experimental validation. At last, section 5 concludes the study with key findings and possible future enhancements.

II. LITERATURE SURVEY

Bangui et al. [11] devised a hybrid ML methodology to modify the IDSs performance by handling the explosive progression in computational power and the necessity to detect malicious incidents promptly. The presented technique mostly leverages the benefits of Random Forest (RF) for detecting well-known network intrusions. Shu et al. [12] used DL with Generative Adversarial Network and explores distributed SDN for designing a collaborative IDS (T-BICDS) for VANETs that assists many Software Defined Network (SDN) controllers jointly training a global ID method for whole network without directly interchanging its sub-network flows. In [13], a trust-related collaborative IDS system was modelled. In T-BICDS, every car will maintain score tables of other cars in network for identifying earlier paradigms of network performance. Furthermore, the vehicles collect realistic network traffic analysis related to local IDS agents that were equipped with k-Nearest Neighbour (k-NN) non-linear classifier. Also, a vehicle could make a collaboration with other adjacent vehicles and upgrade the score tables for identifying intruders in real-time and utilize the table for future prediction.

Gonçalves et al. [14] modelled an Intelligent Hierarchical IDS, which splits the network into four levels, and each one of them into multiple clusters, permitting the utility of various ML related detection approaches. Therefore, every level might utilize a technique that highly suits its requirements. Ercan et al. [15] presented an ML method which employs three new features that were mostly based on the sender position enabling to improve the IDS performance for position falsification assaults. Additionally, it offers a comparison of 2 distinct ML approaches for classifying purposes, i.e., k-NN and RF that were employed for detecting malicious vehicles with the help of these features. At last, Ensemble Learning (EL) that complies various ML approaches has been employed to enrich the detection performances.

Alsarhan et al. [16] applied support vector machine (SVM) for ID in VANET. The SVM structure has several computation benefits, like special direction at a finite sample and irrelevancy among the difficulty of method and sample dimension. ID in VANET becomes combinatorial and nonconvex issue. Therefore, three optimization approaches were employed to optimize the accuracy value of SVM method. In [17], Deep Q-learning Network-based IDS and Bayesian Game theory can be projected for VANETs, named GaDQN-IDS. The communications between attackers and an IDS are framed as a dynamic ID game, where the IDS determines either to modify trade off among efficiency and accuracy or retrained totally when its detection capability is decreased. The Nash Equilibria (NE) of the game can be extracted for revealing the optimum decision of IDS based on the road conditions and detection performances. In [18], the Stacked Sparse AutoEncoder (SSAE) and Softmax classification deep network algorithm can be modelled for detecting the Distributed Denial of Service (DDoS) attacks targeting SDN-related VANETs.

III. THE PROPOSED MODEL

In this study, a new OSNN-IDS algorithm has been devised to detect intrusions and accomplish security in VANETs. The presented OSNN-IDS model encompasses a three-stage process. At the initial stage, the network data is pre-processed in three distinct ways. Initially, the OSNN-IDS model performs data pre-processing to convert the input data into a compatible format for further processing. In this work, data pre-processing is carried out in different ways as given below.

- Dropping Null Values,
- Dropping Duplicate values,
- Replacing Categorical Columns to Numeral, and
- Merging Multiple Data.

As an example, Data will have null values and categorical values before data processing as shown in the table below. The corresponding row with NaN and Infinity has been dropped.

Table 1. An example to show data with Nan and categorical values

Destination Port	Flow Duration	Fwd Packet Length Mean	Fwd Packet Length Std	Bwd Packet Length Max	Bwd Packet Length Min	Bwd Packet Length Mean	Bwd Packet Length Std	Flow Bytes/s	Flow Packets/s	Flow IAT Mean
49578	4	0	0	0	0	0	0	0	500000	4
49578	0	0	0	0	0	0	0	NaN	Infinity	0
49577	3	0	0	0	0	0	0	0	666666.7	3

Next, the pre-processed data is passed into the SNN algorithm for conducting the learning process and the LSTM model is employed for classification process. At last, the hyperparameters related to the SNN model are optimally chosen by the design of RMSProp optimizer, which in turn helps in accomplishing enhanced classification performance. Fig. 1 shows the overall system architecture of the OSNN-IDS algorithm.

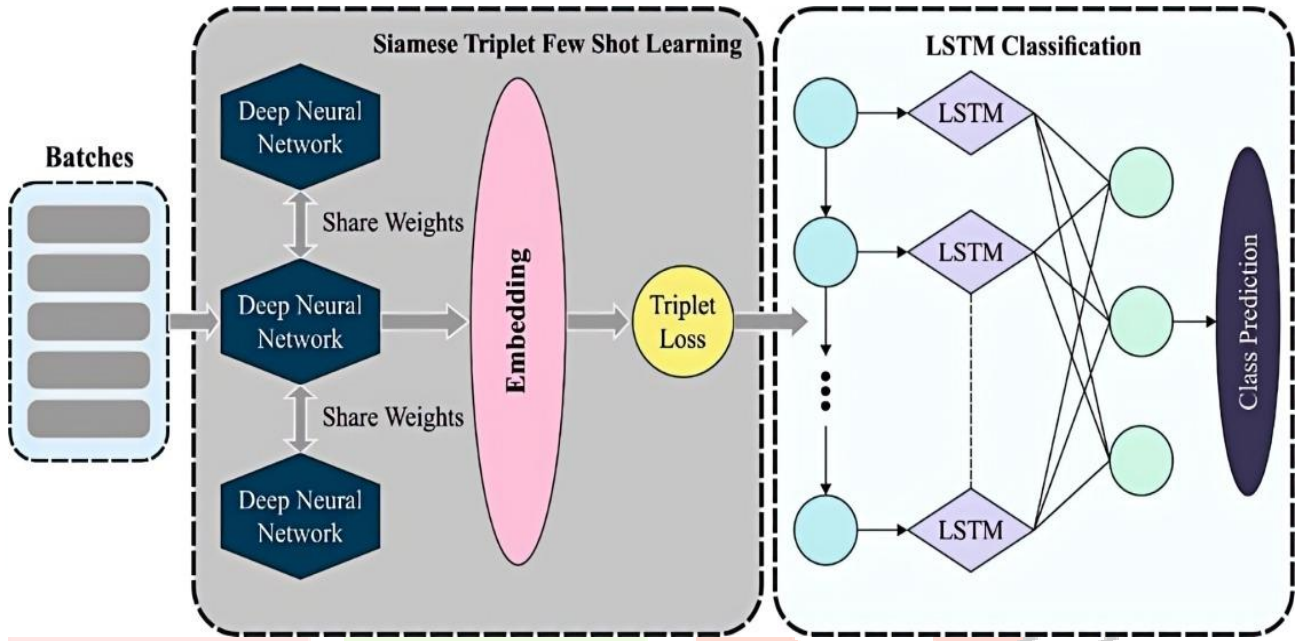

Fig. 1. System architecture

3.1. Siamese Triplet Few Shot Learning Process

In this work, the pre-processed data is passed into the SNN model to classify the distinct kinds of intrusions in the network. The SNN comprises at least two identical, parallel, Convolution Neural networks. This parallel CNN structure allows to learn similarity that is utilized rather than a direct classification. Every parallel CNN which forms a part of the SNN is intended to reduce dimensional representation, or produce an embedding of the input and they are used to improve a Ranking Loss, and test time taken for generating a similarity score.

The parallel CNN could, theoretically, take any form. However, one significant point is that they need to share a similar structure, they have to be completely identical; have the same hyperparameters; share similar initial and updated weights. This consistency permits the model to compare the input it receives, generally one for all the CNN branches.

Fig. 2 shows the structure of SNN. The proposed structure aims at learning the matching functions between 2 dissimilar objects and has extensively been used for multitask learning. We choose a node pair (i, j) as an input, and afterward the primary conversion, the attribute data is provided to the shared hidden layer for learning the latent description [19]. Especially, we adopted a technique of linearly transforming the node attribute data and later fed to the SNN to exploit the complicated relationship among all the node pairs. Inception layer, firstly we adopt the linear conversion to transform the new attribute information and fed as input to the second hidden neuron as follows:

$$h_0 = W_0^T x_i, \tag{1}$$

In Eq. (1), x_i indicates the attribute vector of v_i node and W_0 denotes the linear transformation matrix.

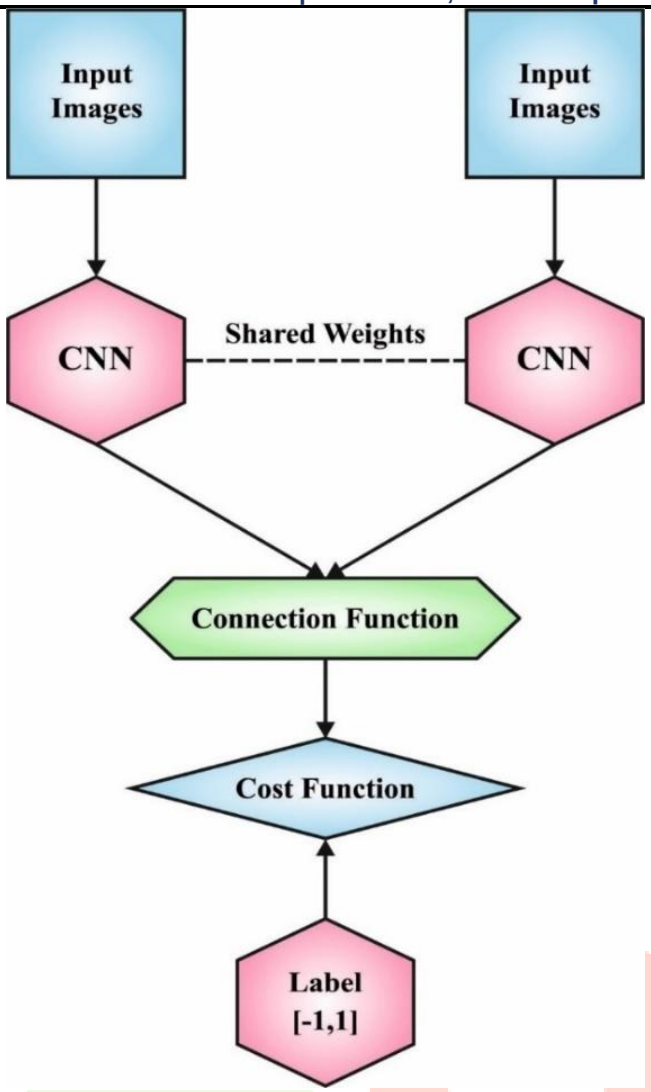

Fig. 2. Structure of SNN

Shared hidden layer: The concatenated vector is processed by a Multi-Layer Perceptron (MLP) function to produce the low-dimension node vector. The hidden representation is represented as:

$$h_1 = \delta_1(W_1^T h_0 + b_1)$$

$$h_2 = \delta_2(W_2^T h_1 + b_2) \quad (2)$$

$$h_k = \delta_k\left(W_k^T h_{k-1} + b_k\right)$$

$W_1, W_2, \ldots, W_K$ and b_1, b_2, b_K denotes the weight matrixes and bias vector, δ_k represents the activation function in $k\text{-}th$ hidden layers. In our model, we employed ReLU activation function. Fig. 3 demonstrates the layers in classifier model.

We would perform the following steps using a Ranking Loss:

- Extract feature from the input.
- Embed the extracted feature onto a d dimension hyperspace.
- Measure the distance between the embedding (Euclidean distance) to be applied as a similarity measure.

The commonest type of Ranking Loss used for SNN is Triplet Ranking Loss. SNN with loss function named Triplet Network as if they are their own thing but they are simply an SNN with 3 branches. As the name suggests, Triplet Ranking Loss encompasses 3 inputs that have been termed a triplet. All the data-points in the triplet have their own task [20]. The Anchor is data of certain class C determines that class the triplet training the model. The Negative is a data-point of certain class that is not C. The Positive is another example of class C. During the training time, every triplet unit is fed to its own CNN branch to be embedded and it is passed to Triplet Loss Function as follows:

$$L = \max(0, D(A,P) - D(A,N) + margin) \quad (3)$$

Now D (A, P) refers to the embedded distance between Positive and Anchor, D (A, N) shows the embedded distance between the Negative and the Anchor. Also, we determine some margin - a frequently applied initial value for this is 0.2, the margin utilized in FaceNet. Fig. 4 showcases the process of triplet loss function [21, 22].

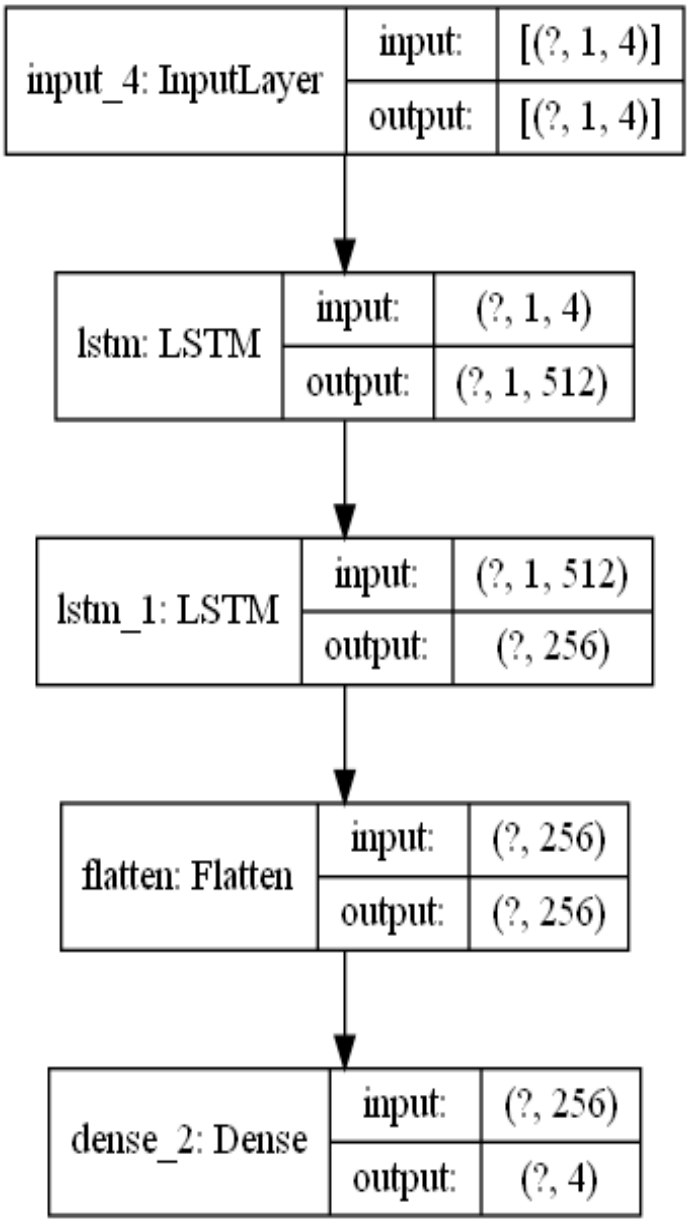

Fig. 3. Layers in Classifier Model

The objective was to minimize the distance between the Positive and the Anchor while maximizing the distance between the Negative and the Anchor. The triplets can be derived using anchor sample x_a, positive sample x_p and negative sample x_n. The intention was that the distance among the anchor sample and the negative sample representation $d(r_a, r_n)$ is higher (and bigger than a margin m) than the distance between the anchor and positive representation $d(r_a, r_p)$. With the same notation, the loss function can be written as follows:

$$L(r_a, r_p, r_n) = \max\left(0, m + d(r_a, r_p) - d(r_a, r_n)\right) \quad (4)$$

Three possible situations of the loss exist and are given as follows.

Easy Triplets: $d(r_a, r_n) > d(r_a, r_p) + m$). The negative samples are sufficiently detached from the anchor samples in relation to the positive samples in the embedded space. The loss was 0 and the net parameters were not upgraded.

Hard Triplets: $d(r_a, r_n) < d(r_a, r_p)$. The negative samples are closer to the anchor compared to the positive. The loss was positive (and more than m).

Semi-Hard Triplets: $d(r_a, r_p) < d(r_a, r_n) < d(r_a, r_p) + m$. The negative samples are highly distant to the anchor compared with the positive, however, the distance was not more than the margin, therefore the loss remains positive (and lesser than m).

Due to continuous deepening of the model, the number of hyperparameters related to the DL model also increased rapidly that leading to model overfitting. Meanwhile, various hyperparameters have an important effect on the CNN methods efficacy. As the trial-and-error approach for hyperparameter tuning becomes an erroneous and tedious procedure, RMSProp optimizer is utilized in this study.

The RMSprop optimizer [23] restricts the oscillation in a vertical direction. As a result, we increased the learning rate and the process might take large steps in a horizontal direction converging fast. The RMSprop calculation is given as follows. The momentum value can be represented as beta and is generally fixed as 0.9.

$$vdw = \beta . vdw + (1 - \beta)\, dw^2 \quad (5)$$

$$vdb = \beta . vdb + (1-\beta) . db^2 \quad (6)$$

$$W = W - \alpha \frac{dw}{\sqrt{vdw} + z} \quad (7)$$

$$b = b - \alpha . \frac{db}{\sqrt{vdb} + \varepsilon} \quad (8)$$

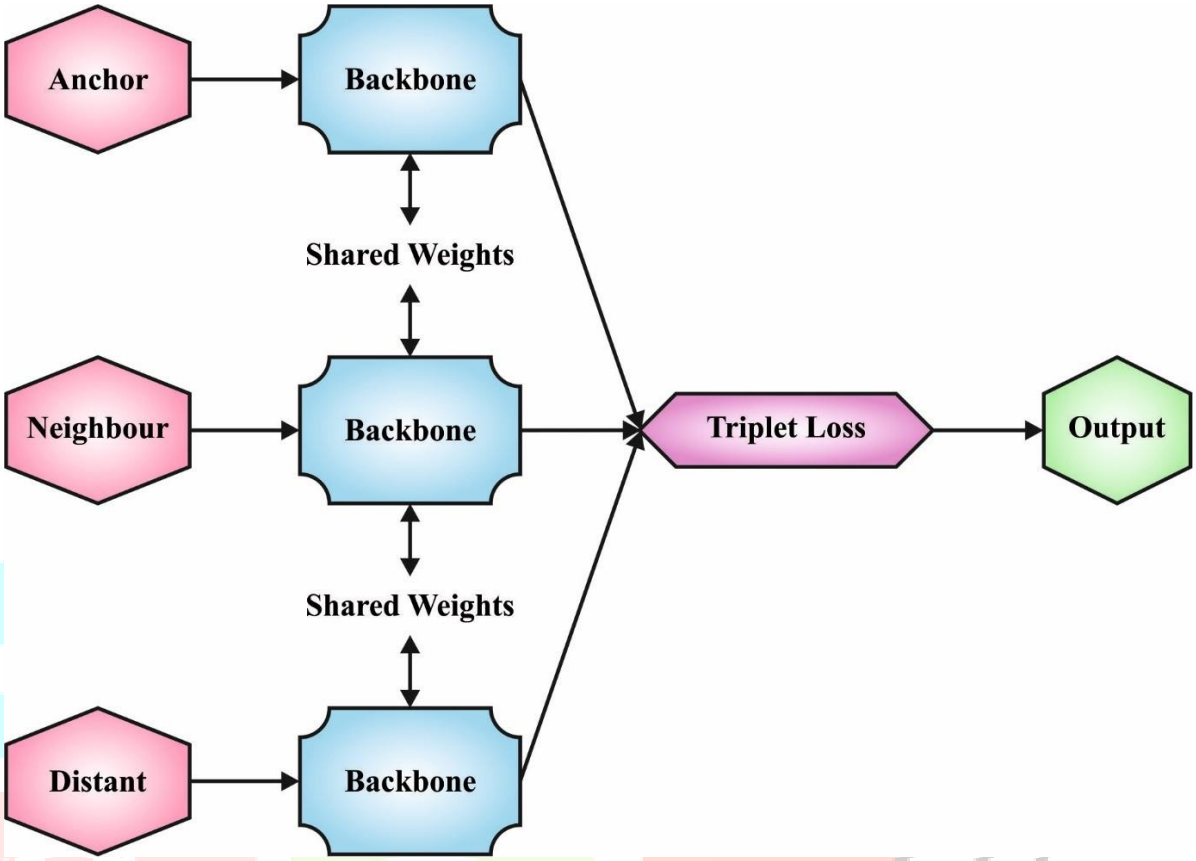

Fig. 4. Triplet loss function

In backpropagation, we apply dW and db for updating W and b parameters as:

$$W = W - learning\ rate \cdot dw \quad (9)$$

$$b = b - learning\ rate \cdot db \quad (10)$$

In RMSprop, before using dW and db individually for all the epochs, we take exponential weighting average of the square of dW and db.

$$S_{dw} = \beta' S_{dw} + (1-\beta) \cdot dW^2 \quad (11)$$

$$S_{db} = \beta' S_{db} + (1-\beta) \cdot db^2 \quad (12)$$

Here, β' beta is an additional hyperparameter and takes value from zero to one. The new weighting average is generated by weight, average of prior and current value square. Afterward the calculation of exponential weighting average and upgrade the parameter as follows.

$$W = W - learning\ rate \cdot \frac{dW}{sqrt(S)} \quad (13)$$

$$b = b - learning\ rate \cdot \frac{db}{sqrt(S)} \quad (14)$$

is reasonably lesser such that we are dividing it by dW. S_{db} is reasonably larger such that we are dividing db with a comparatively large number to decelerate the update on vertical dimension.

3.2. LSTM Based Classification Process

The LSTM learn the dependency that ranges from arbitrary time interval. LSTM attempts to overcome the gradient vanishing problem by switching better neuron with poor one as the LSTM units. During the training phase, the input gate undergoes training within the CEC dataset. Similarly, the input gate is assigned through the measure of zero. As well, the final gate learns a time of permitting flow of dataset from CEC. When the gate is closed, the activation method can be defined within the memory cells. It

permits the error signal to flow through the problem of vanishing gradient. It is the considerable unit of LSTM as follows:

Input: Here, the LSTM exploits the existing input vector represented as x_{tm} and outcome retrieved from preceding time step denoted as $h_{tm} - 1$. Therefore, weighted input is passed and summarized by utilizing $tanh$ activation that resulting in z_{tm}.

Input gate: It is employed for reading x_{tm}, which determines the weighted amount, and apply sigmoid activation. Therefore, the desirable outcomes are optimized by utilizing z_{tm} to give input flow to a memory cell.

Forget gate: here, the LSTM is trained for resetting memory dataset and deliberated to be prior and non-relevant. The forget gate read x_{tm} & h_{tm-1} and exploits sigmoid function for weighted input. The outcomes, f_{tm} is maximized using a cell state in the preceding time step indicates s_{tm-1} activate the memory dataset.

Memory cell: It involves recurrent edge, unit weight, and CEC. An existing cell state S_t can be determined by eliminating unnecessary information from the preceding time step to guarantee the correlated dataset from the recent input.

Output gate: It is exploited by the weighted amount of x_{tm} and h_{tm-1} and exploits sigmoid activation for balancing the flow of data from LSTM.

Output: The outcomes of LSTM units, h_{tm}, is defined by the cell state c_{tm} with $\tan_h$ and optimizes the last gate, o_{tm}. The LSTM performance can be demonstrated below:

$$i_{tm} = \sigma\left(W_i \cdot \left[w_{tm}, h_{tm} - 1 \right] + b_i \right) \tag{15}$$

$$f_{tm} = \sigma\left(W_f \cdot \left[w_{tm}, h_{tm} - 1 \right] + b_f \right) \tag{16}$$

$$o_{tm} = \sigma\left(W_o \cdot \left[w_{tm}, h_{tm} - 1 \right] + b_o \right) \tag{17}$$

$$c_{tm} = f_{tm} \Theta c_{tm} + i_{tm} \Theta \tanh\left(W_c \cdot \left[w_{tm}, h_{tm-1} + b_c \right] \right) \tag{18}$$

$$h_{tm} = o_{tm} \Theta \tanh\left(C_{tm} \right) \tag{19}$$

From the expression, o, i, and f respectively indicates output, input, and forget gates, σ indicates the sigmoid activation function used to handle external and internal details, and $\{W_i, W_f, W_o, W_c, b_i, b_f, b_o, b_c\}$ represent the variable to be learned in the training process.

IV. PERFORMANCE VALIDATION

The experimental validation of the OSNN-IDS approach is tested utilizing the CICIDS_2017 dataset [24] from Kaggle repository. The proposed model is simulated using Python tool. The CICIDS2017 dataset comprises benign and the most up-to-date general attack that resembles the true real-world data (PCAPs). It also contains the outcomes of network traffic examination utilizing CICFlowMeter with labelled flow dependent upon the source, time stamp, and destination IPs, source and destination ports, protocols, and attack (CSV files). During this work, it contains utilized 40000 samples under with 10000 samples in every class such as DDoS, Benign, PortScan, and DoS Hulk.

Fig. 5 exhibits the confusion matrix generated by the OSNN-IDS model on CICIDS_2017 dataset.

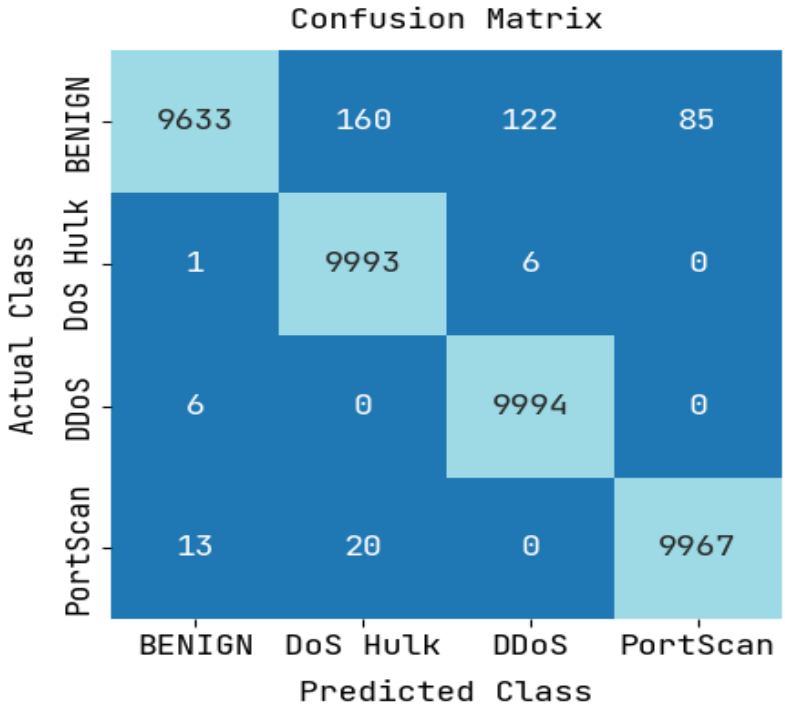

Fig. 5. Confusion matrix of OSNN-IDS approach under CICIDS_2017 dataset

Table 2 ad Fig 6. Shows overall intrusion detection outcomes of the OSNN-IDS system on the applied dataset.

To understand the result better, we define each of the term defined in Table 2

.

i. Accuracy

It is the percentage of predictions our model has got right out of all predictions. It is defined as

$$Accuracy = \frac{(TP + TN)}{(TP + TN + FP + FN)} \tag{20}$$

ii. Precision

It is used to calculate the model's ability to classify positive values correctly.

$$Precision = \frac{(True\ Positive)}{(Predicted\ Positive)} = \frac{TP}{(TP + FP)} \tag{21}$$

iii. Recall or Sensitivity

It is used to calculate the model's ability to predict the positive values

$$Recall = \frac{True\ Positive}{Actual\ Positive} = \frac{TP}{(TP + FN)} \tag{22}$$

iv. F1 Score

It combines precision and recall into one metric by calculating the harmonic mean between these two parameters

v. ROC (Receiver Operating Characteristic curve)

It is the graph showing the performance of a classification model at all classification thresholds.

Table 2 Overall result analysis of OSNN-IDS approach with distinct measures

Metrics	Values
Accuracy	98.97
Precision	98.98
Recall	98.97
F1-Score	98.96
ROC Score	99.98

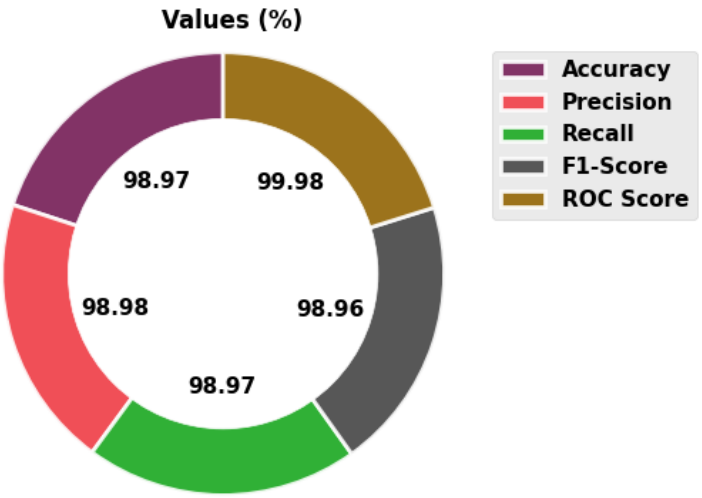

Fig. 6. Overall result analysis of OSNN-IDS approach with distinct measures

An obvious precision-recall investigation of the OSNN-IDS algorithm on test dataset is represented in Fig. 7. The figure exposed that the OSNN-IDS system has resulted to higher values of precision-recall values under all classes.

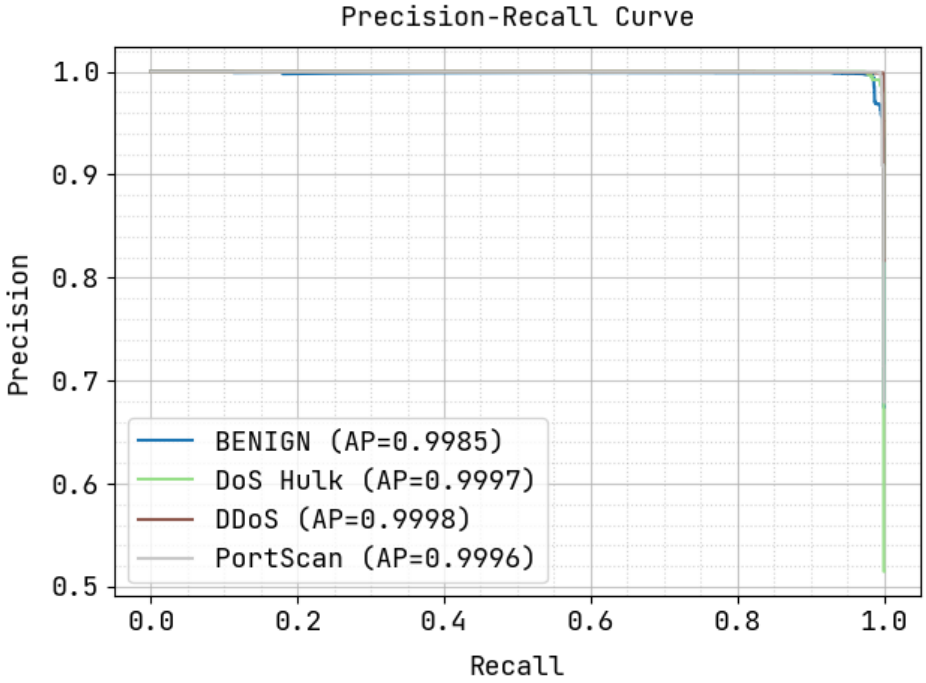

Fig. 7. Precision-recall analysis of OSNN-IDS methodology

A detailed ROC examination of the OSNN-IDS approach on test dataset is represented in Fig. 8. The outcomes signified the OSNN-IDS approach has outperformed their ability in categorizing distinct classes on test dataset.

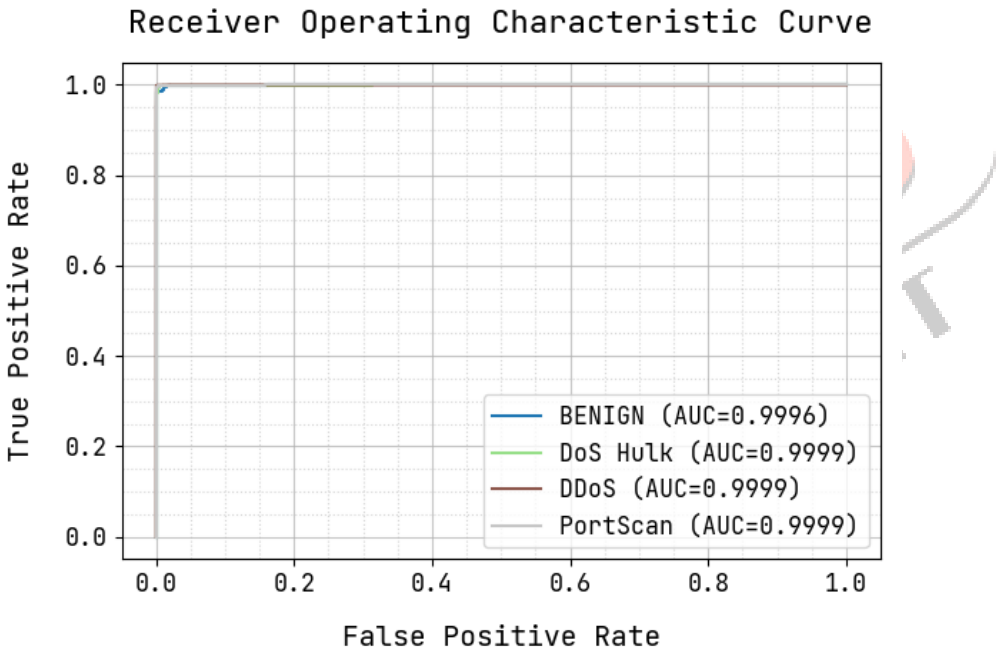

Fig 8. ROC Characteristic curve of OSNN-IDS Methodology

The training accuracy (TRA) and validation accuracy (VLA) acquired by the OSNN-IDS algorithm on test dataset is depicted in Fig. 9. The experimental result exposed that the OSNN-IDS system has gained enhanced values of TRA and VLA. Commonly, the VLA looked that superior to TRA.

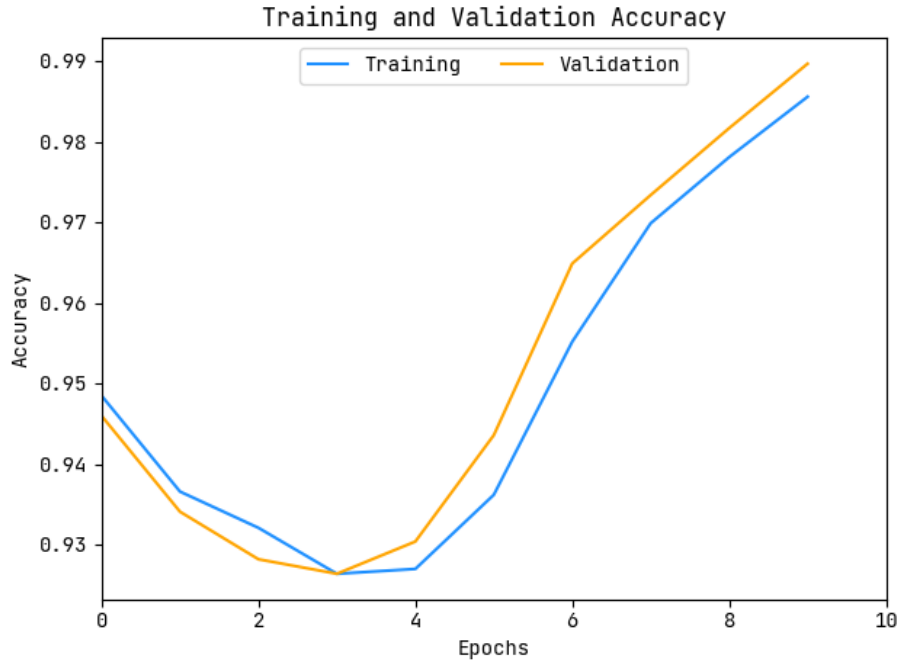

Fig. 9. TRA and VLA analysis of OSNN-IDS methodology

The training loss (TRL) and validation loss (VLL) realized by the OSNN-IDS system on test dataset are depicted in Fig. 10. The experimental result stated that the OSNN-IDS technique has achieved minimal values of TRL and VLL. In certain, the VLL is lesser than TRL.

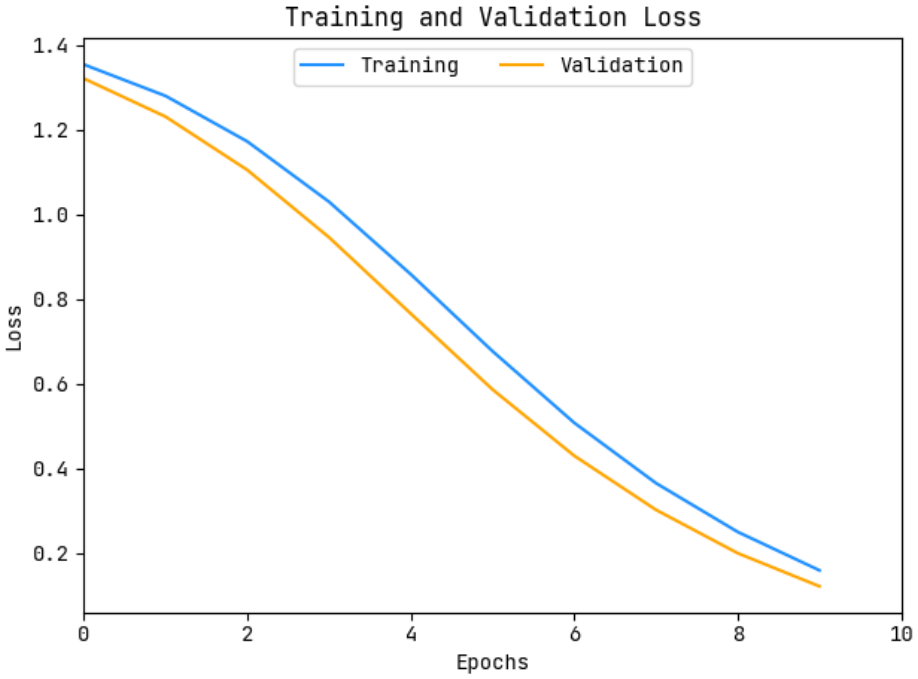

Fig. 10. TRL and VLL analysis of OSNN-IDS methodology

To demonstrate the betterment of the OSNN-IDS approach, a comparative analysis is made with other ML models in Table 3 [23]. The results demonstrated that the OSNN-IDS system has shown enhanced results over other models.

For instance, with respect to $prec_n$, the OSNN-IDS model has offered higher $prec_n$ of 98.98% and a superior $reca_l$ of 98.97%. Table 2 implied that the MLP and SMO techniques have demonstrated worse performance that lower $accu_y$ values of 95.13% and 95.08% respectively. Next, the RF, HT, and PART Models have accomplished moderately closer classification performance with $accu_y$ of 96.19%, 96.94%, and 96.50% correspondingly. But the OSNN-IDS model has outperformed the other models with maximum $accu_y$ of 98.97%.

Table 3 Comparative analysis of OSNN-IDS approach with existing methodologies

Methods	Accuracy	Precision	Recall	F1-Score	ROC Score
OSNN-IDS	98.97	98.98	98.97	98.96	99.98
Random Forest	96.19	96.10	97.27	96.74	99.12
MLP	95.13	97.94	95.57	97.25	98.78
Hoeffding Tree	96.94	98.13	97.28	97.18	99.01
SMO	95.08	98.21	98.29	96.64	98.96
PART	96.50	97.64	98.15	95.45	98.95

Fig. 11 illustrates a comparison analysis of the OSNN-IDS methodology with other ML techniques in terms of $F1_{score}$ and ROC_{score}. The outcomes portrayed that the OSNN-IDS technique has illustrated higher outcomes over other techniques. For sample, with regard to $F1_{score}$, the OSNN-IDS system has obtainable higher $F1_{score}$ of 98.96% whereas the RF, MLP, HT, SMO, and PART approaches have accomplished lesser $F1_{score}$ of 96.74%, 97.25%, 97.18%, 96.64%, and 95.45% correspondingly. Besides, in terms of ROC_{score}, the OSNN-IDS technique has obtainable higher ROC_{score} of 99.98% whereas the RF, MLP, HT, SMO, and PART models have accomplished minimal ROC_{score} of 99.12%, 98.78%, 99.01%, 98.96%, and 98.95% correspondingly.

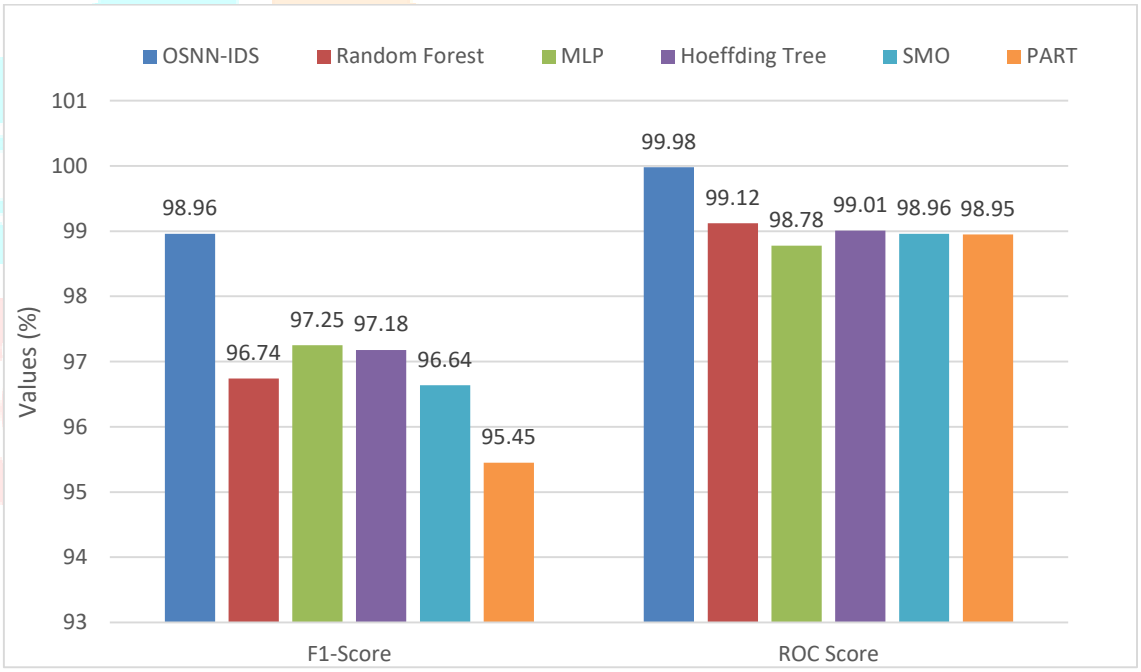

Fig. 11. $F1_{score}$ and ROC_{score} analysis of OSNN-IDS approach with existing methodologies

Finally, a comprehensive $accu_y$ analysis of the OSNN-IDS model with recent models [4, 6, 20-22] are performed in Table 4.

Table .4 Accuracy analysis of OSNN-IDS model with other developed networks

Methods	Accuracy (%)
OSNN-IDS	98.97
Leveraging Siamese Networks	82.93
Siamese Capsule Networks	95.56
FSL-SCNN	96.80
Few Shot Learning	92.34

The results inferred that the Leveraging Siamese Networks has demonstrated lower $accu_y$ of 82.93%. Next, few shot learning and Siamese Capsule Networks (SCN) models have obtained slightly increased $accu_y$ of 92.34% and 95.56% respectively. Though the FSL-SCNN model has shown reasonable $accu_y$ of 96.80%, the OSNN-IDS model has outperformed the other models with maximum $accu_y$ of 98.97%. From the detailed results and discussion, it can be assured that the OSNN-IDS technique has achieved

improved intrusion detection performance over other models.

V. CONCLUSION

In this study, a novel OSNN-IDS methodology was formulated to detect intrusions and accomplish security in VANET. The presented OSNN-IDS model encompasses a three-stage process. At the initial stage, the network data is pre-processed in three distinct ways. Next, the pre-processed data is passed into the SNN algorithm for performing the learning process. At last, the hyperparameters related to the SNN model are optimally chosen by the design of RMSProp optimizer, which in turn helps in accomplishing enhanced classification performance. At final stage, the LSTM model is applied for classification process. For assuring the better performance of the OSNN-IDS method, a wide range of simulations were carried out on CIC_IDS_2017 dataset. A wide-ranging experimental analysis demonstrates the promising performances of the OSNN-IDS method compared to other existing models. Therefore, the presented OSNN-IDS model can be employed to accomplish maximum security in VANET. In future, feature selection and outlier detection approaches can be developed for enhancing the detection efficiency of the OSNN-IDS algorithm.

REFERENCES

[1] Bangui, H. and Buhnova, B., "Recent advances in machine-learning driven intrusion detection in transportation: survey". Procedia Computer Science 184, pp.877-886, 2021

[2] Gonçalves, Fábio et al. "A Systematic Review on Intelligent Intrusion Detection Systems for VANETs." 11th International Congress on Ultra-Modern Telecommunications and Control Systems and Workshops (ICUMT) (2019): 1-10.

[3] Y. Zeng, M. Qiu, D. Zhu, Z. Xue, J. Xiong and M. Liu, "DeepVCM: A Deep Learning Based Intrusion Detection Method in VANET," IEEE 5th Intl Conference on Big Data Security on Cloud (BigDataSecurity), IEEE Intl Conference on High Performance and Smart Computing, (HPSC) and IEEE Intl Conference on Intelligent Data and Security (IDS), 2019, pp. 288-293, doi: 10.1109/BigDataSecurity-HPSC-IDS.2019.00060

[4] Iliyasu, A.S.; Abdurrahman, U.A.; Zheng, L. "Few-Shot Network Intrusion Detection Using Discriminative Representation Learning with Supervised Autoencoder" Applied Sciences, vol. 12, no. 5, pp 2351, 2022, https://doi.org/10.3390/app12052351

[5] A. R. Gad, A. A. Nashat and T. M. Barkat, "Intrusion Detection System Using Machine Learning for Vehicular Ad Hoc Networks Based on ToN-IoT Dataset," in IEEE Access, vol. 9, pp. 142206-142217, 2021, doi: 10.1109/ACCESS.2021.3120626.

[6] Y. Yu and N. Bian, "An Intrusion Detection Method Using Few-Shot Learning," in IEEE Access, vol. 8, pp. 49730-49740, 2020, doi: 10.1109/ACCESS.2020.2980136.

[7] A. Ghaleb, F., Saeed, F., Al-Sarem, M., Ali Saleh Al-rimy, B., Boulila, W., Eljialy, A.E.M., Aloufi, K. and Alazab, M., "Misbehavior-aware on-demand collaborative intrusion detection system using distributed ensemble learning for VANET". Electronics, 9(9), p.1411.

[8] A. M. Alrehan and F. A. Alhaidari, "Machine Learning Techniques to Detect DDoS Attacks on VANET System: A Survey," 2nd International Conference on Computer Applications & Information Security (ICCAIS), 2019, pp. 1-6, doi: 10.1109/CAIS.2019.8769454.

[9] Sara Ftaimi and Tomader Mazri., "A comparative study of Machine learning algorithms for VANET networks", Proceedings of the 3rd International Conference on Networking, Information Systems & Security (NISS2020). Association for Computing Machinery, New York, NY, USA, Article 10, 1–8. https://doi.org/10.1145/3386723.3387829

[10] F. Gonçalves, J. Macedo and A. Santos, "Evaluation of VANET Datasets in Context of an Intrusion Detection System," 2021 International Conference on Software, Telecommunications and Computer Networks (SoftCOM), 2021, pp. 1-6, doi: 10.23919/SoftCOM52868.2021.9559058.

[11] Bangui, H., Ge, M. and Buhnova, B., "hybrid data-driven model for intrusion detection in VANET." The 12th International Conference on Ambient Systems, Networks and Technologies (ANT)., Procedia Computer Science, 184, pp.516-523

[12] J. Shu, L. Zhou, W. Zhang, X. Du and M. Guizani, "Collaborative Intrusion Detection for VANETs: A Deep Learning-Based Distributed SDN Approach," in IEEE Transactions on Intelligent Transportation Systems, vol. 22, no. 7, pp. 4519-4530, July 2021, doi: 10.1109/TITS.2020.3027390.

[13] T. Nandy, R. M. Noor, M. Yamani Idna Bin Idris and S. Bhattacharyya, "T-BCIDS: Trust-Based Collaborative Intrusion Detection System for VANET," 2020 National Conference on Emerging Trends on Sustainable Technology and Engineering Applications (NCETSTEA), 2020, pp. 1-5, doi: 10.1109/NCETSTEA48365.2020.9119934.

[14] Gonçalves, F.; Macedo, J.; Santos, A. "An Intelligent Hierarchical Security Framework for VANETs". 2021, vol. 12, no. 11 article no. 455. https://doi.org/10.3390/info12110455.

[15] Ercan, S., Ayaida, M. and Messai, N., "Misbehavior detection for position falsification attacks in VANETs using machine learning", IEEE Access, vol. 10, pp.1893-1904, Digital Object Identifier 10.1109/ACCESS.2021.3136706.

[16] Alsarhan, A., Alauthman, M., Alshdaifat, E. et al. "Machine Learning-driven optimization for SVM-based intrusion detection system in vehicular ad hoc networks", Journal of Ambient Intelligence and Humanized Computing , Feb 2021. https://doi.org/10.1007/s12652-021-02963-x

[17] Liang J, Ma M, Tan X (2021) "Gadqn-ids: A novel self-adaptive ids for vanets based on bayesian game theory and deep reinforcement learning". IEEE Transactions on Intelligent Transportation Systems pp 1–14

[18] Polat, H., Turkoglu, M. and Polat, O.,"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.

[19] J. Wang, N. Gao, J. Peng and J. Mo, "Attributed Network Embedding via a Siamese Neural Network," 2019 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computing, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI), 2019, pp. 1101-1108, doi: 10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00209.

[20] Hindy, H., Tachtatzis, C., Atkinson, R., Brosset, D., Bures, M., Andonovic, I., Michie, C. and Bellekens, X.,"Leveraging siamese networks for one-shot intrusion detection model", arXiv, doi 10.48550/ARXIV.2006.15343

[21] Wang, Z.M., Tian, J.Y., Qin, J., Fang, H. and Chen, L.M., "A few-shot learning-based Siamese capsule network for intrusion detection with imbalanced training data. Computational intelligence and neuroscience", Computational Intelligence and Neuroscience(2021), vol. 2021, https://doi.org/10.1155/2021/7126913

[22] X. Zhou, W. Liang, S. Shimizu, J. Ma and Q. Jin, "Siamese Neural Network Based Few-Shot Learning for Anomaly Detection in Industrial Cyber-Physical Systems," in IEEE Transactions on Industrial Informatics, vol. 17, no. 8, pp. 5790-5798, Aug. 2021, doi: 10.1109/TII.2020.3047675.

[23] Babu, D.V., Karthikeyan, C. and Kumar, A., "Performance analysis of cost and accuracy for whale swarm and rmsprop optimizer", IOP Conference Series: Materials Science and Engineering, vol. 993, No. 1, p. 012080, doi:10.1088/1757-899X/993/1/012080

[24] "Intrusion Detection Evaluation Dataset", https://www.kaggle.com/datasets/cicdataset/cicids2017

[25] Gonçalves, Fábio, Joaquim Macedo, and Alexandre Santos. 2021. "An Intelligent Hierarchical Security Framework for VANETs" *Information* vol. 12, no. 11: 455. https://doi.org/10.3390/info12110455