ISSN 1330-3651 (Print), ISSN 1848-6339 (Online) https://doi.org/10.17559/TV-20240619001788
Original scientific paper

Enhanced Cyberthreat Detection and Classification with CMCOADL-TDC Using Deep Learning and Optimization Techniques

U. SAKTHIVELU, C. N. S. VINOTH KUMAR*

Abstract: Cyberthreat detection and classification are crucial fields aiming to develop intelligent systems capable of real-time identification and categorization of various cyberthreats such as malware, phishing, social engineering, and ransomware. Detection involves monitoring networks and systems for suspicious activities like unusual traffic patterns, unauthorized access attempts, and abnormal behaviours. Classification entails identifying specific threat types like viruses, Trojans, or worms, requiring a deep understanding of their characteristics and behaviours. This paper introduces a novel approach, the Cauchy-Mutation Coyote Optimization Algorithm with the Deep Learning Enabled Threat Detection and Classification (CMCOADL-TDC) technique. The proposed technique focuses on accurately identifying cyberthreats through pre-processing, feature selection, and classification. The CMCOADL-TDC technique utilizes a feature selection method based on the Cauchy-Mutation Coyote Optimization Algorithm (CMCOA) model to select optimal feature subsets. A bidirectional gated recurrent unit (BiGRU) model is employed for detection and classification. The BiGRU model's Parameter tuning is performed using the Sunflower Optimization (SFO) model. Additionally, network defense mechanisms are enhanced by employing the time-inhomogeneous hidden Bernoulli model (TI-HBM). To demonstrate its efficacy, extensive simulations were performed by comparing the CMCOADL-TDC approach against state-of-the-art models, showing superior performance. The performance validation of the CMCOADL-TDC approach portrayed a superior accuracy value of 95.58% over existing models.

Keywords: BiGRU model; cyberthreat detection; deep learning; feature selection; optimization algorithms

1 INTRODUCTION

Network security encompasses diverse practices and methods to safeguard computer networks against unauthorized access, data breaches, and many other security vulnerabilities [1]. A few instances cause destructive cyber-crimes, such as malicious insider Denial of Service (DOS) and web-based security attacks [2]. All cyber attacks are created over the network connection except for cyber-physical attacks, which are not in the field of application in this research [3]. Internet of Things (IoT) devices are computationally restricted, so applying conventional cyber-security techniques to protect susceptible IoT environments is impracticable. Furthermore, the different multi-dimensional area of susceptibilities requires the IoT platforms to exceed the extent of standard rule-based security techniques [4].

Their use is necessary to ensure the effectiveness of antivirus software, firewalls, Network Intrusion Detection Systems (NIDS), and network security measures [5]. NIDS is commonly employed to detect network intruders by continuous monitoring of network traffic for any indications of malicious or suspicious behaviour. NIDS helps detect network threats comprising worms, viruses, and DDoS attacks [6]. The achievement factors of NIDS include accuracy, detection speed, and reliability. Many research studies have been performed on NIDS; enhancements are still needed to decrease false alarms and raise identification accuracy [7].

ML is a type of Artificial Intelligence (AI) that extracts valuable data from providing information to predict outcomes without explicit programming [8]. This benefit maintains the latest method of distinguishing abnormal and everyday activities, decreasing the false-positive [9]. Many ML methods have been used in anomaly-based IDS, such as decision trees, clustering, DL approaches, nearest neighbours, and association rules. DL is a sub-domain of ML that contains multiple hidden layers, making it easier to work with substantial data issues. DL is classified into two: unsupervised and supervised learning [10].

This paper introduces a novel approach, the Cauchy-Mutation Coyote Optimization Algorithm with the Deep Learning Enabled Threat Detection and Classification (CMCOADL-TDC) technique. The proposed technique focuses on accurately identifying cyberthreats through pre-processing, feature selection, and classification. The CMCOADL-TDC technique utilizes a feature selection method based on the Cauchy-Mutation Coyote Optimization Algorithm (CMCOA) model to select optimal feature subsets. A bidirectional gated recurrent unit (BiGRU) model is employed for detection and classification. The BiGRU model's Parameter tuning is performed using the Sunflower Optimization (SFO) model. Additionally, network defense mechanisms are enhanced by employing the time-inhomogeneous hidden Bernoulli model (TI-HBM).

2 RELATED WORKS

Al Mamun et al. [11] employ Particle Swarm Optimization (PSO) to autonomously develop a classification method, which optimizes weights for each feature in the dataset, which are also used to predict class labels through a weighted sum. Sakthivelu and Kumar [12] utilize a Colonel Blotto game model, where the defense strategically allocates central processing units (CPUs) across storage devices in an organization. Alrehaili et al. [13] present effectual DL models by employing a hybrid approach with Stacked Autoencoder and Long Short-Term Memory (SAE-LSTM) and Convolutional Neural Network (CNN) with LSTM (CNN-LSTM) to detect APT attack indicators. Genge et al. [14] propose E-APTDetect, a methodology for early APT detection that utilizes dynamic attestation and multi-level data fusion. The technique is based on sensitivity analysis and Dempster-Shafer's Theory of Evidence. Chandra, Challa, and Pasupuletti [15] design and implement an onion-layered security system to protect cloud environments from advanced persistent threats, employing diverse defence strategies and techniques comprising social engineering, intrusion

detection, and vulnerability assessments. Hasan, Islam, and Uddin [16] developed an effective method to detect APT attacks to mitigate their impact. Employing ML, it utilizes various boosting-based approaches to predict diverse types of APTs relevant to cybersecurity. Furthermore, Explainable AI (XAI) is incorporated to offer actionable insights for stakeholders and practitioners in the field. Shang et al. [17] propose a novel network flow-based methodology for detecting command-and-control (C&C) channels of unknown APT attacks. The proposed system provides comprehensive protection against Advanced Persistent Threat (APT) attacks through detection and defense mechanisms. Sakthivelu and Kumar [18] employ a modified lateral movement detection model and a Dynamic Deception technique using a belief update approach.

Li et al. [19] aim to develop a novel APT attack-defense technique to quantify and determine optimal defense strategies against APT campaigns. Khalaf et al. [20] propose a novel model that incorporates CNN, Recurrent Neural Networks (RNNs), autoencoders (AEs), and GANs to create a comprehensive strategy for detecting anomalies. Althobaiti and Escorcia-Gutierrez [21] introduce the WSSADL-CTDC technique, integrating a weighted salp swarm algorithm with DL for cyber-threat detection and classification. Albakri et al. [22] presents a Rock Hyrax Swarm Optimization and DL (RHSODL-AMD) model. An RHSO-based feature selection and Adamax-optimized attention recurrent AE are utilized for improved detection. Rasikha and Marikkannu [23] present an intelligent cybersecurity mechanism utilizing benchmark data sources and optimal feature selection via the Modified Puzzle Optimization Algorithm (MPOA). The selected features are input into the Ensemble Serial Cascaded DL with Attention Mechanism (ESCDLAM), incorporating 1DCNNs, RNNs, and Deep Temporal Convolutional Networks (DTCN). Alzubi et al. [24] propose a fusion of DL-based cyberattack detection and classification models for intelligent systems (FDL-CADIS) technique. This method uses MobileNetv2 for feature extraction, black widow optimization (BWO) for hyperparameter tuning, and an ensemble of voting classifiers comprising GRUs and LSTM for efficient malware detection. Kumari et al. [25] present an Artificial Neural Network (ANN) layer optimized with the execution of Spider Monkey Optimization (SMO) to detect attacks or intrusions in the system. Ramaiah et al. [26] introduce an NIDS based on a deep neural network (DNN) that utilizes network data features selected through the bagging and boosting methods.

Saravanan and Mathina [27] introduce a DL-based Echo State Neural Network (ESN) model. The ESN captures temporal fluctuations in network traffic, automatically extracting features to detect complex patterns of potential cyber threats. Eugine Prince et al. [28] propose the Bat Optimization-based Spiking Neural System (BAbSNS) framework. This model utilizes bat-optimal fitness and attack classification by comparing extracted features with trained data for intrusion detection. Almahadeen et al. [29] present a novel approach to enhance financial cybersecurity threat detection using AE-Multilayer Perceptron (AE-MLP) hybrid models. Assiri and Ragab [30] propose the Honey Badger Algorithm with an Optimal Hybrid Deep Belief Network (HBA-OHDBN) methodology. This model employs HBA for optimal feature sets and utilizes the HDBN model for intrusion detection, with hyperparameter tuning via Dung Beetle Optimization (DBO). Furthermore, blockchain technology is employed to enhance network security. Alabduallah et al. [31] propose a Hybrid Salp Swarm Algorithm with DL (HSSADL-CAC) technique for cyber attack classification. This model utilizes HSSA for feature selection and employs a deep extreme learning machine (DELM) for detection, with hyperparameter tuning via beluga whale optimization (BWO). Sangher, Singh, and Pandey [32] present a hybrid DL approach incorporating RNN) and BiLSTM models. Shareef et al. [33] present a model using the Zebra Optimization Algorithm (ZOA). The method also employs a Dual-channel Graph Attention Network (DGAN) to detect IoT anomalies, with enhanced accuracy from hyperparameter tuning via the Sooty Tern Optimization Algorithm (STOA). Santhadevi and Janet [34] developed a model by using DL models.

3 THE PROPOSED MODEL

This paper presents a newly developed CMCOADL-TDC approach that aims to automate and enhance the precision of cyberthreat detection. To achieve this, the CMCOADL-TDC technique encompasses different subprocesses: pre-processing, feature selection, and classification. Fig. 1 illustrates the overall procedure of the CMCOADL-TDC method.

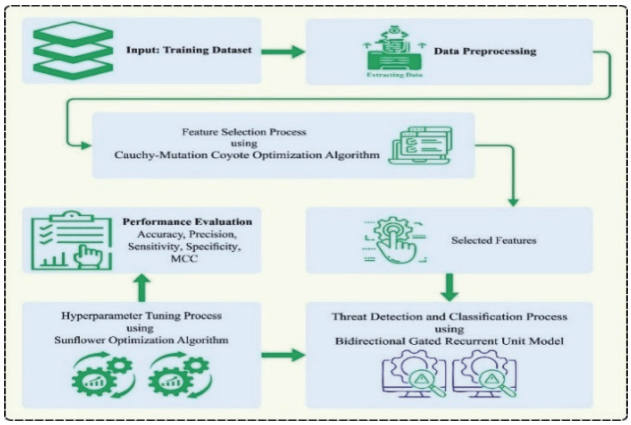

Figure 1 Overall process of CMCOADL-TDC approach

3.1 Data Pre-Processing

At the pre-processing stage, *Z*-score normalization was utilized to transform the input data into a uniform format. *Z*-score normalization, or standard score normalization, is a general approach employed for normalizing or standardizing a database by altering a standard normal distribution. This normalized process is particularly suitable for managing features with distinct scales or measurement units. The *Z*-score of the data point measures several standard deviations (SD) away from the mean of the database. The formula to calculate the *Z*-score of data point, x, provides the mean, μ, and SD σ, of the database:

$$Z = \frac{(x-\mu)}{\sigma} \tag{1}$$

3.2 Feature Selection using CMCOA

The CMCOA technique is used for feature selection [35]. The selection of this model is justified by its capacity to navigate intrinsic search spaces efficiently.

The CMCOA procedure is given below:

Step 1: Initialize the population of coyotes as $G_n = g_1, g_2, \ldots, g_N$ (number of features extracted). Next, split these initialized populations into G_u packs of coyotes of the related size G_v. Now, the size of populations is represented as $G_u \times G_v$. The COA works based on coyotes' social condition (*SCs*), representing the decision variable or $SCs\left(\vec{i}\right)$ of global optimization problems. $\left(SC_u^{v,t}\right)$ of the u^{-th} individual in the v^{-th} pack and at t is iteration formulated by Eq. (2):

$$SC_{bl}^{v,t} = \vec{i} = \left(i_1, i_2, ..., i_L\right) \tag{2}$$

where L denotes the dimensions of the search space.

Step 2: Rank all the new solution's efficacy based on the fitness values of coyotes. The fitness of all the coyotes (features extracted) is calculated by the maximum accuracy of the classification that is formulated as follows:

$$FitCo_u^{v,t} = \max\left(\sum_{n=1}^{N} G_n\left(A'_{acc}\right)\right) f\left(SC_u^{v,t}\right) \tag{3}$$

Step 3: initially, Coyotes are randomly allocated to packs, but coyotes occasionally leave the packs and come to be solitary or join the pack. The eviction from the pack was defined by the number of coyotes within the pack and took place with the probability P_{COA}, that is:

$$P_{COA} = 0.005 \cdot \left(G'_v\right)^2 \tag{4}$$

In Eq. (4), $G'_v \le \sqrt{200}$; hence, the value of P_{COA} is higher than 1. The number of coyotes in all the groups is restricted to 14 to enhance the diversity of the model, viz., cultural interaction among coyotes.

Step 4: Allocate the alpha coyotes in all the packs to the one most familiar with the environments. The alpha (α) was mathematically formulated as follows:

$$\alpha^{v,t} = \left\{SC_u^{v,t} \middle| arg_u = \{1, 2, ..., G'_b\} \min f\left(SC_u^{v,t}\right)\right\} \tag{5}$$

Step 5: Define the characteristic features of the coyote for the cultural transformation using the following expression:

$$CT_x^{v,t} = \begin{cases} Q_{\frac{G'_v+1}{2},x}^{v,t} & G'_v \text{ is odd num} \\ \dfrac{1}{2\left(Q_{\frac{G'_v}{2},x}^{v,t} + Q_{\frac{G'_v}{2}+1,x}^{v,t}\right)} & \text{other wise} \end{cases} \tag{6}$$

In Eq. (6), $G'_v \le \sqrt{200}$ defines the coyotes, social situation ranks for group number v at t time for the parameter x.

Step 6: Generate a new social condition for all the coyotes. Two factors involved, x_1 and x_2 describe the cultural transition between groups that is expressed as follows:

$$x_1 = \alpha^{v,t} - SC_{cs1}^{v,t} \tag{7}$$

$$x_2 = CT^{v,t} - SC_{cs2}^{v,t} \tag{8}$$

In the equation, x_1 denotes the culture difference between the selected coyote (*cs*1) and the leader (alpha), and x_2 shows the cultural difference among the group culture trending as well as the selected coyote (*cs*2). Choose x_1 and x_2 random coyotes using the uniform probability distribution (0 & 1) and later update the newest *SC* using alpha and pack influence:

$$\text{New}\_SC_u^{v,t} = SC_u^{v,t} + w_1 \cdot x_1 + w_2 \cdot x_2 \tag{9}$$

In Eq. (9), w_1 and w_2 indicate the weight values of the alpha and pack influence.

Step 7: Inspect the suitability of the new position and the fitness function of the latest *SCs* using the following equation:

$$\text{New}\_FitCo_u^{v,t} = f\left(SC_u^{v,t}\right) \tag{10}$$

Furthermore, the cognitive capability of the coyotes defines whether the new *SC* is better than the older one, thus:

$$SC_u^{v,t+1} = \text{New}\_SC_u^{v,t} \cdot \left(cauchy\left(0, 1\right) + 1\right) \tag{11}$$

In this case, the CM is used to disrupt the coyote position arbitrarily and is efficient in improving population diversity, which avoids optimum local trapping and enhances the performance of the COA. Thus, the optimum solution (best social condition) at every iteration becomes the set of optimized features (O_s), mathematically formulated as follows:

$$O_s = \left(O_1, 0_2, ..., O_s\right) \tag{12}$$

In Eq. (12), S denotes the optimized features attained from the optimization technique.

3.3 Threat Detection using BiGRU

The BiGRU methodology was implemented during this phase to identify and classify cyberthreats. This bidirectional processing improves the model's understanding of intrinsic patterns in time-series data, BiGRU presents enhanced accuracy and robustness, particularly in dynamic environments where threats evolve rapidly. Its architecture effectually reduces issues related to vanishing gradients, allowing for deeper learning.

Furthermore, BiGRU is computationally efficient, making it appropriate for real-time applications.

The BiGRU, a kind of RNN, has effectively acquired knowledge from the given dataset [36]. The BiGRU layer effectively integrates information from the preceding and current stages to provide accurate predictions. When the unidirectional GRU estimates the prior instruction without moving forward to the next instruction, then the difficulty of data context loss rises. Forward and backward GRU layers are two sequential GRUs that help BiGRU overcome the problems of lost context data.

Reset gate: The reset gate can control the short-term memory network. The reset gate shows that the new input is combined with the formerly evaluated output. Eq. (13) evaluates this number given below:

$$r = \sigma\left(x_t U^r + W^r s_{t-1}\right) \tag{13}$$

Due to the sigmoid function, the r value will oscillate between zero and one. The weight matrix for the reset gate is represented by W^r and U^r.

$$z = \sigma\left(x_t U^u + W^u s_{t-1}\right) \tag{14}$$

GRU goes through a 2-stage procedure to recognize *ht* hidden layer (HL). An initial stage is to be produced that is called candidate HL that is formulated as follows:

$$ht = \tan h\left(x_t U^h + W^h\left(s_{t-1} \cdot r\right)\right) \tag{15}$$

Once the r value equals 1, it implies that each piece of data from the prior s_{t-1} HL is assumed. When the r value is 0, the data in the prior HL is fully ignored.

$$s_t = (1-z)\cdot h + z \cdot s_{t-1} \tag{16}$$

As revealed in Eq. (17), the HL of these two processes is merged into procedure BiGRU.

$$\overleftrightarrow{GRU} = GRU^f \oplus GRU^b \tag{17}$$

3.4 Hyperparameter Tuning using SFO Model

In this study, the SFO technique optimally adjusts the hyperparameter values of the BiGRU model [37]. Its innovative approach provides a significant edge in attaining optimal hyperparameter settings.

One way to put it is that pollination is the backbone of plant reproduction, according to biologists. Next, please define the optimal angle of solar panel orientation towards the sun. During the process of pollination, the purpose is to achieve a value with a low probability using random means. A smaller distance separates flower i and $i + 1$. At the same time, the sunflower develops and creates a single pollen gamete per flower. When the space is doubled, the strength is reduced by one factor.

The quantity of heat Q_i received by all the plants is defined by Eq. (18):

$$Q_i = \frac{P}{4\pi r_i^2} \tag{18}$$

Here, r_i denotes the distance between them, and P shows the power source.

$$\overrightarrow{S_i} = \frac{X^* - X_i}{\left\|X^* - X_i\right\|}, \text{ where } i = 1, 2, 3, ..., n \tag{19}$$

The following expression evaluates the sunflower direction:

$$d_i = \alpha \cdot P_i\left(X_i + X_i - 1\right)\cdot\left\|X_i + X_i - 1\right\| \tag{20}$$

In Eq. (20), α denotes the inertial displacement, and the probability of pollination is $P_i \cdot \left(\left\|X_i + X_i - 1\right\|\right)$ which implies the sunflower "i" pollinates its nearby neighbour $i - 1$ and generates the newest individual in the arbitrary location. The maximal step is formulated by Eq. (21):

$$d_{\max} = \frac{\left\|X_{\max} - X_{\min}\right\|}{2 \cdot N_{pop}} \tag{21}$$

$$\overrightarrow{X_{i+1}} = \overrightarrow{x_i} + d_i \cdot \overrightarrow{s_i} \tag{22}$$

Fitness assessment is a crucial component of the SFO model. Currently, the primary criterion utilized for the design of a feedforward system is the accuracy metric.

$$Fitness = \max\left(P\right) \tag{23}$$

$$P = \frac{TP}{TP + FP} \tag{24}$$

3.5 TI-HBM-based Defender Mechanism

Here, the TI-HBM model is used as a defender mechanism to enhance security in the network.

$$\sum_{i=1}^{N}\sum_{t=1}^{L_{\max}} P(i, t) = 1 \tag{25}$$

$$P(i, t) = 0 \text{ for } t > L_{\max} \tag{26}$$

It develops some valuable parameters in $P(i, t)$ that are required to apply TIHBM in real-time. Let $L_{\max}$ be the maximal length of observation sequence X.

4 EXPERIMENTAL VALIDATIONS

In this section, the simulation outcome of the CMCOADL-TDC approach is validated on the CICIDS2018 database [38]. The dataset comprises 3000 instances with six classes. Among the available 80 features, 28 features are involved in this study.

Fig. 2 exposes the confusion matrices of the CMCOADL-TDC approach under the Threat detection process. The outcomes stated that the CMCOADL-TDC method accurately recognized various kinds of attacks.

Tab. 1 and Fig. 3 show the CMCOADL-TDC method's threat identification results on an 80:20 ratio of TR set/TS set. When applied to 80% of TR sets, the CMCOADL-TDC method yields an average $accu_y$, $prec_n$, $sens_y$, $spec_y$ and MCC of 95.58%, 86.89%, 86.72%, 97.35%, and 84.13%, respectively. At the same time, on 20% of the TS set, the CMCOADL-TDC approach attains average $accu_y$, $prec_n$, $sens_y$, $spec_y$ and MCC of 94.72%, 84.22%, 84.29%, 96.84%, and 81.05% correspondingly.

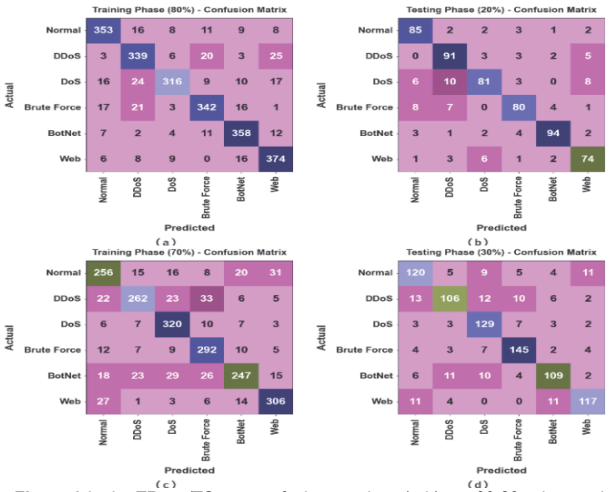

Figure 2 In the TR set/TS set, confusion matrices (a-b) are 80:20, whereas in the C set/TS set, they are 70:30

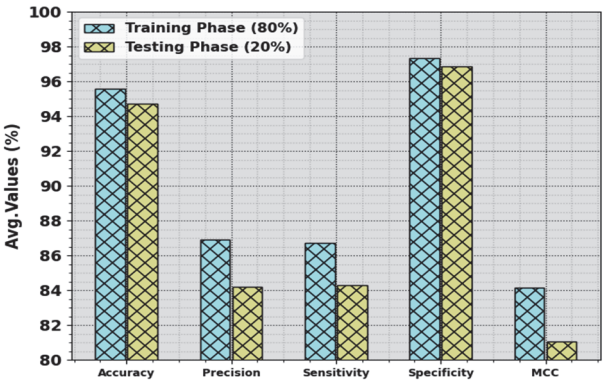

Figure 3 Average outcome of CMCOADL-TDC approach on 80 - 20 of *TR* set/Ts set

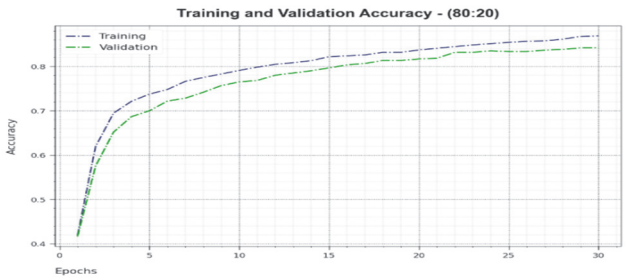

Figure 4 The present study examines the $Accu_y$ outcomes of the CMCOADL-TDC technique on an 80:20 split of the training set and test set

Fig. 4 displays the training accuracy TR_accu_y and validation accuracy VL_accu_y of the CMCOADL-TDC technique on an 80:20 split of the training set TR set/TS set. The TL_accu_y metric is determined by using the CMCOADL-TDC technique on the T*R* dataset.

Fig. 5 examines the *TR*_loss and *VR*_loss for the CMCOADL-TDC approach on an 80:20 split of the training set and test set.

Table 1 Threat detection outcome of CMCOADL-TDC approach on 80:20 of TR set/TS set

Class	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	MCC
Training Phase (80%)					
Normal	95.79	87.81	87.16	97.54	84.96
Ddos	94.67	82.68	85.61	96.46	80.93
DoS	95.58	91.33	80.61	98.51	83.27
Brute Force	95.46	87.02	85.50	97.45	83.54
BotNet	96.25	86.89	90.86	97.31	86.61
Web	95.75	85.58	90.56	96.83	85.47
Average	95.58	86.89	86.72	97.35	84.13
Testing Phase (20%)					
Normal	95.33	82.52	89.47	96.44	83.17
DDos	94.00	79.82	87.50	95.36	79.96
DoS	93.33	86.17	75.00	97.36	76.48
Brute Force	94.33	85.11	80.00	97.20	79.15
BotNet	96.50	91.26	88.68	98.18	87.85
Web	94.83	80.43	85.06	96.49	79.69
Average	94.72	84.22	84.29	96.84	81.05

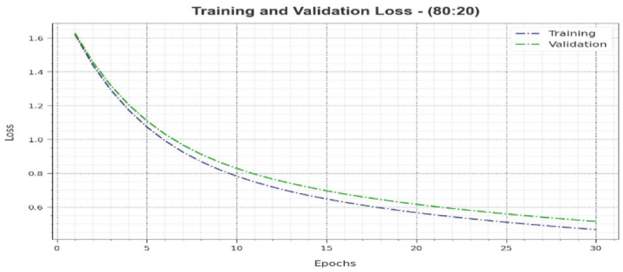

Figure 5 The Loss outcome of CMCOADL-TDC technique on 80:20 of TR set/TS set

Fig. 6 shows a detailed PR analysis of the CMCOADL-TDC technique on 80:20 of the TR set/TS set.

In Fig. 7, an ROC study exposes the CMCOADL-TDC technique at 80:20 of the TR/TS set. The methodology results in higher ROC values. Besides, it is evident that the CMCOADL-TDC methodology can extend maximum ROC values to 6 classes.

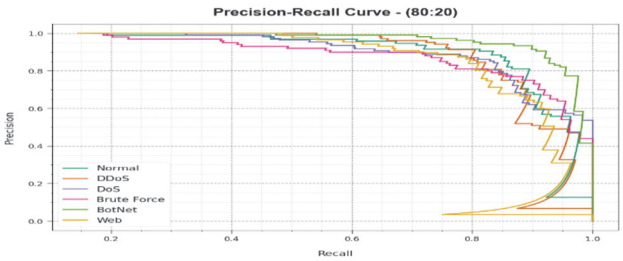

Figure 6 The Loss PR outcome of CMCOADL-TDC technique on 80:20 of TR set/TS set

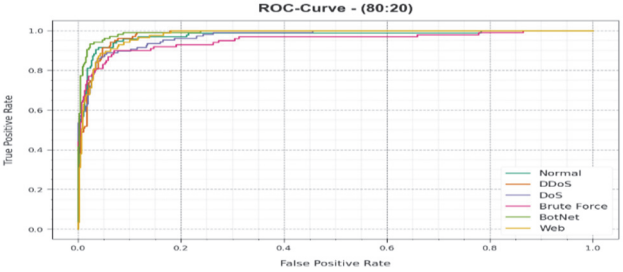

Figure 7 ROC outcome of CMCOADL-TDC technique on 80:20 of TR set/TS set

In Tab. 2 and Fig. 8, the threat detection outcomes of the CMCOADL-TDC model are inspected at 70:30 of the TR set/TS set. Moreover, on 70% of the TR set, the CMCOADL-TDC approach acquires average $accu_y$, $prec_n$, $sens_y$, $spec_y$ and also *MCC* of 93.38%, 80.20%, 80.19%, 96.03%, and 76.17% respectively.

Fig. 9 displays the training accuracy TR_accu_y and VL_accu_y of the CMCOADL-TDC technique at 70:30 of the TR/TS set. The TL_accu_y is determined by estimating the CMCOADL-TDC technique on the TR database, while the VL_accu_y is computed by estimating the performance on an individual testing database.

Table 2 Threat detection outcome of CMCOADL-TDC method on 70:30 of TR set/TS set

Class	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$	*MCC*
Training Phase (70%)					
Normal	91.67	75.07	73.99	95.15	69.55
DDos	93.24	83.17	74.64	96.97	74.83
DoS	94.62	80.00	90.65	95.42	81.97
Brute Force	94.00	77.87	87.16	95.30	78.84
BotNet	92.00	81.25	68.99	96.73	70.24
Web	94.76	83.84	85.71	96.62	81.61
Average	93.38	80.20	80.19	96.03	76.17
Testing Phase (30%)					
Normal	92.11	76.43	77.92	95.04	72.41
Ddos	92.33	80.30	71.14	96.54	71.10
DoS	93.78	77.25	87.76	94.95	78.65
Brute Force	94.89	84.80	87.88	96.46	83.19
BotNet	93.44	80.74	76.76	96.57	74.86
Web	94.78	84.78	81.82	97.23	80.20
Average	93.56	80.72	80.55	96.13	76.74

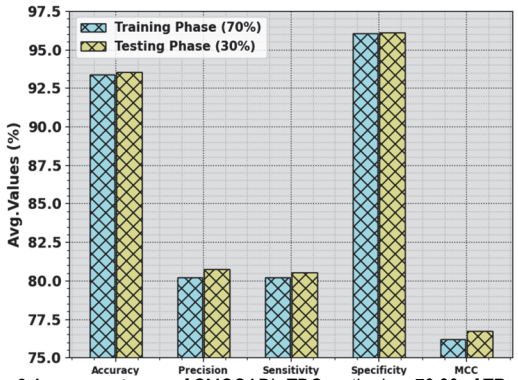

Figure 8 Average outcome of CMCOADL-TDC method on 70:30 of TR set/TS set

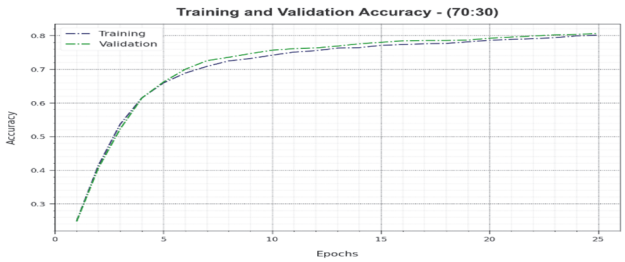

Figure 9 $Accu_y$ outcome of CMCOADL-TDC method on 70:30 of TR set/TS set

Fig. 10 displays the analysis of *TR_loss* and *VR_loss* for the CMCOADL-TDC technique on a 70:30 split of the training set and testing set. The *VR_loss* metric assesses the CMCOADL-TDC method's performance using distinct validation data. The results indicate a decreasing trend in *TR_loss* and *VR_loss* as the number of epochs increases.

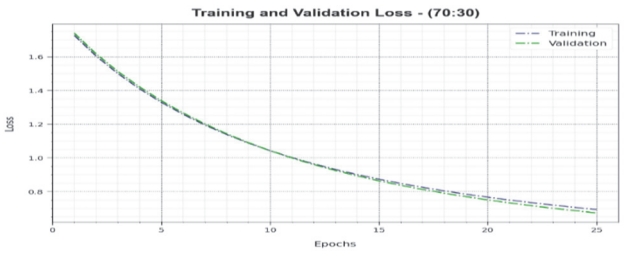

Figure 10 Loss outcome of CMCOADL-TDC method on 70:30 of TR set/TS set

Fig. 11 presents the results of the CMCOADL-TDC technique, showing a comprehensive PR outcome at a ratio of 70:30 for the training set and test set. The results indicate that the CMCOADL-TDC approach yielded the highest PR values. Subsequently, it becomes evident that the CMCOADL-TDC approach can get higher PR values across six distinct classes.

Fig. 12 presents the Receiver Operating Characteristic (ROC) results obtained from the CMCOADL-TDC approach, with a training set to test set ratio of 70:30. Moreover, the CMCOADL-TDC method has superior receiver operating characteristic (ROC) values across six distinct classes.

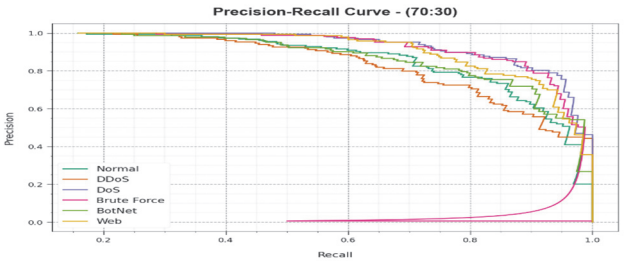

Figure 11 PR outcome of CMCOADL-TDC method on 70:30 of TR set/TS set

The threat detection results of the CMCOADL-TDC method are compared with other DL models in Tab. 3 and Fig. 13.

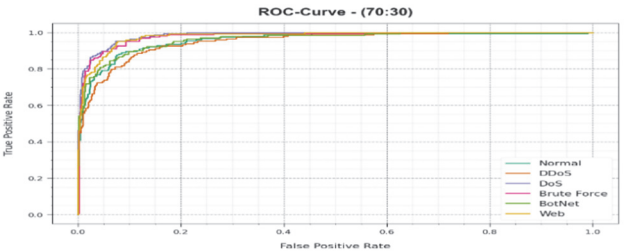

Figure 12 ROC outcome of CMCOADL-TDC method on 70:30 of TR set/TS set

Table 3 Comparative outcome of CMCOADL-TDC methodology with other models

Methods	$Accu_y$	$Prec_n$	$Sens_y$	$Spec_y$
LSTM	85.00	61.00	61.00	90.00
DNN	76.00	40.00	40.00	85.00
CNN	60.00	65.00	60.00	75.00
DRNN	88.00	71.00	71.00	93.00
SVM	77.00	63.00	43.00	86.00
DBN	84.00	60.00	62.00	85.00
DBN+MFO	80.00	60.00	60.00	89.00
DBN+WOA	85.00	62.00	62.00	91.00
CMCOADL-TDC	95.58	86.89	86.72	97.35

In addition, the DNN, SVM, and DBN+MFO models attain slightly boosted results. The DRNN, DBN, LSTM,

and DBN+WOA models also accomplish considerable performance. However, the CMCOADL-TDC technique performs better over other models with maximum $accu_y$ of 95.58%, $prec_n$ of 86.89%, $sens_y$ of 86.72%, and $spec_y$ of 97.35%. These outcomes confirmed the superior threat detection and classification performance of the CMCOADL-TDC methodology.

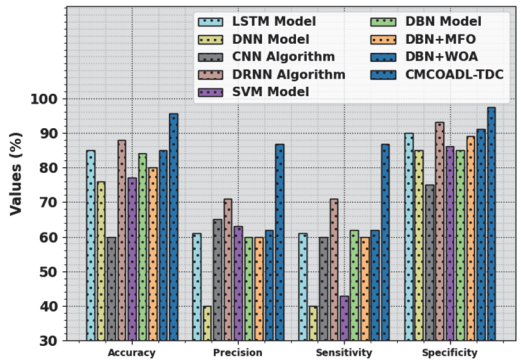

Figure 13 Comparative outcome of CMCOADL-TDC methodology with other models

Tab. 4 and Fig. 14 illustrate the CT analysis of the CMCOADL-TDC technique. The LSTM takes 10.92 seconds, DNN 11.46 seconds, CNN 7.29 seconds, DRNN 8.48 seconds, SVM 9.67 seconds, and DBN 7.88 seconds. Notably, DBN enhanced with MFO has a CT of 9.51 seconds, while DBN with WOA shows a longer CT of 12.14 seconds. The CMCOADL-TDC technique demonstrates the fastest performance at just 4.40 seconds.

Table 4 CT analysis of CMCOADL-TDC methodology with other models

Techniques	CT / sec
LSTM	10.92
DNN	11.46
CNN	7.29
DRNN	8.48
SVM	9.67
DBN	7.88
DBN+MFO	9.51
DBN+WOA	12.14
CMCOADL-TDC	4.40

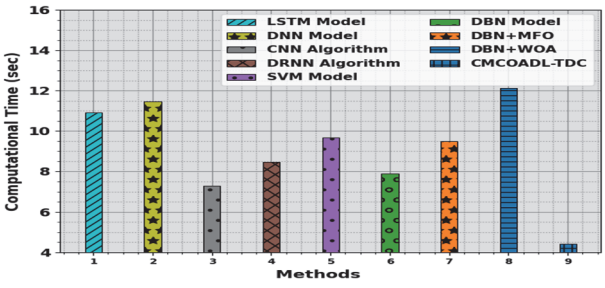

Figure 14 CT analysis of CMCOADL-TDC methodology with other models

5 CONCLUSION

In this article, a novel CMCOADL-TDC method for the automated and accurate identification of cyberthreats is proposed. The CMCOADL-TDC technique primarily focused on the design of feature selection, hyperparameter tuning, and defender mechanism. To achieve this, the CMCOADL-TDC technique encompasses different subprocesses: pre-processing, feature selection, and classification. The CMCOADL-TDC technique employs a CMCOA-based feature selection approach to choose optimal feature subsets. Next, the BiGRU was employed to detect and classify cyberthreats. Eventually, the SFO model will be used to parameterize the BiGRU model. Moreover, the network defender mechanism uses TI-HBM. An extensive ranging simulation is performed to demonstrate the greater outcome of the CMCOADL-TDC method. The performance validation of the CMCOADL-TDC approach portrayed a superior accuracy value of 95.58% over existing models. The limitations of the CMCOADL-TDC approach comprise the dependence on specific datasets that may not fully represent the diversity of real-world scenarios, potentially affecting the generalizability of the outcomes. Moreover, the computational complexity of the proposed methods may limit the scalability of massive datasets. Future work should consider validating the framework across varied datasets to improve robustness and generalization. Exploring hybrid approaches that incorporate various optimization techniques could enhance efficiency. Examining real-time implementation and monitoring systems would strengthen practical applications in cybersecurity. Lastly, improving model interpretability will significantly gain user trust and safeguard effectual deployment in real-world environments.

6 REFERENCES

[1] Alhan, B., Gönen, S., Karacayilmaz, G., Bariskan, M. A., & Yilmaz, E. N. (2022). Real-Time Cyber Attack Detection OverHoneyPi Using Machine Learning. *Technical Gazette, 29*(4), 1394-1401. https://doi.org/10.17559/TV-20210523121614

[2] Kumar, K. V. & Rajakani, V. (2024). Modeling of Intrusion Detection System Using Double Adaptive Weighting Arithmetic Optimization Algorithm with Deep Learning on Internet of Things Environment. *Brazilian Archives of Biology and Technology, 67*. https://doi.org/10.1590/1678-4324-2024231010

[3] Alsanad, A. & Altuwaijri, S. (2022). Advanced Persistent Threat Attack Detection using Clustering Algorithms. *International Journal of Advanced Computer Science and Applications, 13*(9). https://doi.org/10.14569/IJACSA.2022.0130976

[4] Ghafir, I., Hammoudeh, M., & Prenosil, V. (2018). Defending against the advanced persistent threat: Detection of disguised executable files. *PeerJ Preprints*. https://doi.org/10.7287/peerj.preprints.2998v2

[5] Kavitha, S., Uma Maheswari, N., & Venkatesh, R. (2023). Intelligent Intrusion Detection System using Enhanced Arithmetic Optimization Algorithm with Deep Learning Model. *Tehnički vjesnik, 30*(4), 1217-1224 https://doi.org/10.17559/TV-20221128071759

[6] Abdullayeva, F. J. (2021). Advanced Persistent Threat attack detection method in cloud computing based on autoencoder and softmax regression algorithm. *Array, 10*, 100067. https://doi.org/10.1016/j.array.2021.100067

[7] Zou, Q., Sun, X., Liu, P., & Singhal, A. (2020). An Approach for Detection of Advanced Persistent Threat Attacks. *Computer, 53*(12), 92-96. https://doi.org/10.1109/mc.2020.3021548

[8] Rajakani, V. & Kumar, K. V. (2023). Barnacles Mating Optimizer with Hopfield Neural Network-Based Intrusion

Detection in Internet of Things Environment. *Tehnički vjesnik - Technical Gazette, 30*(6). https://doi.org/10.17559/TV-2021121611563

[9] Alqazzaz, A. & Alrashdi, I. (2024). An efficient intrusion detection model based on neutrosophic logic for optimal response from the arranged response set. *International Journal of Neutrosophic Science,* (3), 233-33. https://doi.org/10.54216/IJNS.230320

[10] Ebrahim, M., Nunamaker Jr., J. F., & Chen, H. (2020). Semi-Supervised cyber threat identification in dark net markets: A transductive and deep learning approach. *Journal of Management Information Systems, 37*(3), 694-722. https://doi.org/10.1080/07421222.2020.1788797

[11] Al Mamun, A., Al-Sahaf, H., Welch, I., & Camtepe, S. (2022). Advanced Persistent Threat Detection: A Particle Swarm Optimization Approach. *32nd International Telecommunication Networks and Applications Conference (ITNAC)*. https://doi.org/10.1109/itnac55475.2022.9998358

[12] Sakthivelu, U. & Kumar, C. N. S. V. (2023). An Adaptive Defensive Mechanism for DQN Storage Resources Allocation from Advanced Persistent Threats. *International Conference on Innovative Computing, Intelligent Communication and Smart Electrical Systems (ICSES)*, 1-8. https://doi.org/10.1109/ICSES60034.2023.10465545

[13] Alrehaili, M., Alshamrani, A., & Eshmawi, A. (2021). A Hybrid Deep Learning Approach for Advanced Persistent Threat Attack Detection. *The 5th International Conference on Future Networks & Distributed Systems*, 78-86. https://doi.org/10.1145/3508072.3508085

[14] Genge, B., Haller, P., & Roman, A. S. (2023). E-APTDetect: Early Advanced Persistent Threat Detection in Critical Infrastructures with Dynamic Attestation. *Applied Sciences, 13*(6), 3409. https://doi.org/10.3390/app13063409

[15] Chandra, J. V., Challa, N., & Pasupuletti, S. K. (2017). A Defense Approach from Advanced Persistent Threat through Defense in Depth Mechanism for Cloud Security. *Advanced Science and Technology Letters*. https://doi.org/10.14257/astl.2017.147.37

[16] Hasan, M. M., Islam, M. U., & Uddin, J. (2023). Advanced Persistent Threat Identification with Boosting and Explainable AI. *SN Computer Science, 4*(3). https://doi.org/10.1007/s42979-023-01744-x

[17] Shang, L., Guo, D., Ji, Y., & Li, Q. (2021). Discovering unknown advanced persistent threat using shared features mined by neural networks. *Computer Networks, 189,* 107937. https://doi.org/10.1016/j.comnet.2021.107937

[18] Sakthivelu, U. & Kumar, C. N. S. V. (2023). Advanced Persistent Threat Detection and Mitigation Using Machine Learning Model. *Intelligent Automation & Soft Computing, 36*(3). https://doi.org/10.32604/iasc.2023.036946

[19] Li, P., Yang, X., Xiong, Q., Wen, J., & Tang, Y. Y. (2018). Defending against the Advanced Persistent Threat: An Optimal Control Approach. *Security and Communication Networks, 2018*, 2975376. https://doi.org/10.1155/2018/2975376

[20] Khalaf, L. I., Alhamadani, B., Ismael, O. A., Radhi, A. A., Ahmed, S. R., & Algburi, S. (2024). Deep Learning-Based Anomaly Detection in Network Traffic for Cyber Threat Identification. *Proceedings of the Cognitive Models and Artificial Intelligence Conference*, 303-309. https://doi.org/10.1145/3660853.3660932

[21] Althobaiti, M. M. & Escorcia-Gutierrez, J. (2024). Weighted salp swarm algorithm with deep learning-powered cyber-threat detection for robust network security. *AIMS Mathematics, 9*(7), 17676-17695. https://doi.org/10.3934/math.2024859

[22] Albakri, A., Alhayan, F., Alturki, N., Ahamed, S., & Shamsudheen, S. (2023). Metaheuristics with deep learning model for cybersecurity and Android malware detection and classification. *Applied Sciences, 13*(4), 2172. https://doi.org/10.3390/app13042172

[23] Rasikha, V. & Marikkannu, P. (2024). An ensemble deep learning-based cyber attack detection system using optimization strategy. *Knowledge-Based Systems, 301*, 112211. https://doi.org/10.1016/j.knosys.2024.112211

[24] Alzubi, O. A., Qiqieh, I., & Alzubi, J. A. (2023). Fusion of deep learning based cyberattack detection and classification model for intelligent systems. *Cluster Computing, 26*(2), 1363-1374. https://doi.org/10.1007/s10586-022-03686-0

[25] Kumari, D., Sinha, A., Dutta, S., & Pranav, P. (2024). Optimizing neural networks using spider monkey optimization algorithm for intrusion detection system. *Scientific Reports, 14*(1), 17196. https://doi.org/10.1038/s41598-024-68342-6

[26] Ramaiah, M., Vanmathi, C., Khan, M. Z., Vanitha, M., & Deepa, M. (2023). An Efficient Intrusion Detection System to Combat Cyber Threats using a Deep Neural Network Model. *Journal of ICT Research & Applications, 17*(3). https://doi.org/10.5614/itbj.ict.res.appl.2023.17.3.2

[27] Saravanan, N. & Mathina, P. A. (2024). SDN-Enabled Networks: A Deeper Learning-Based Echo State Neural Network for Cyber Threat Detection. *7th International Conference on Circuit Power and Computing Technologies (ICCPCT), 1*, 1356-1360. https://doi.org/10.1109/ICCPCT61902.2024.10673080

[28] Eugine Prince, M., Josephin Shermila, P., Sajithra Varun, S., Anna Devi, E., Sujatha Therese, P., Ahilan, A., & Jasmine Gnana Malar, A. (2023). An Optimized Deep Learning Algorithm for Cyber-Attack Detection. *International Conference on Frontiers of Intelligent Computing: Theory and Applications,* 465-472. https://doi.org/10.1007/978-981-99-6706-3_39

[29] Almahadeen, L., Mahadin, G. A., Santosh, K., Aarif, M., Deb, P., Syamala, M., & Bala, B. K. (2024). Enhancing Threat Detection in Financial Cyber Security Through Auto Encoder-MLP Hybrid Models. *International Journal of Advanced Computer Science & Applications, 15*(4). https://doi.org/10.14569/IJACSA.2024.0150495

[30] Assiri, F. Y. & Ragab, M. (2023). Optimal Deep-Learning-Based Cyberattack Detection in a Blockchain-Assisted IoT Environment. *Mathematics, 11*(19). https://doi.org/10.3390/math11194080

[31] Alabduallah, B., Maray, M., Alruwais, N., Alabdan, R., Darem, A. A., Alallah, F. S., Alsini, R., & Yafoz, A. (2024). Class Imbalanced Data Handling with Cyberattack Classification Using Hybrid Salp Swarm Algorithm with Deep Learning Approach. *Alexandria Engineering Journal, 106*, 654-663. https://doi.org/10.1016/j.aej.2024.08.061

[32] Sangher, K. S., Singh, A., & Pandey, H. M. (2023). Cyber Threat Intelligence Traffic Through Black Widow Optimisation by Applying RNN-BiLSTM Recognition Model. International *Journal of Computer Science and Network Security, 23*(11), 100.

[33] Shareef, S. K., Chaitanya, R. K., Chennupalli, S., Chokkakula, D., Kiran, K. V. D., Pamula, U., & Vatambeti, R. (2024). Enhanced Botnet Detection in IoT Networks Using Zebra Optimization and Dual-Channel GAN Classification. *Scientific Reports, 14*(1), 17148. https://doi.org/10.1038/s41598-024-67865-2

[34] Santhadevi, D. & Janet, B. (2023). Stacked Deep Learning Framework for Edge-Based Intelligent Threat Detection in IoT Network. *The Journal of Supercomputing, 79*(11), 12622-12655. https://doi.org/10.1007/s11227-023-05153-y

[35] Eke, H. N. & Petrovski, A. (2023). Advanced Persistent Threats Detection Based on Deep Learning Approach. *IEEE 6th International Conference on Industrial Cyber-Physical Systems (ICPS)*. https://doi.org/10.1109/icps58381.2023.10128062

[36] Rahman, Z., Yi, X., & Khalil, I. (2023). Blockchain-Based AI-Enabled Industry 4.0 CPS Protection against Advanced Persistent Threat. *IEEE Internet of Things Journal, 10*(8), 6769-6778. https://doi.org/10.1109/JIOT.2022.3147186

[37] Sun, H., Yang, X., Yang, L. X., Huang, K., & Li, G. (2023). Impulsive Artificial Defense against Advanced Persistent Threat. *IEEE Transactions on Information Forensics and Security, 18*, 3506-3516. https://doi.org/10.1109/TIFS.2023.3284564

[38] Sakthivelu, U. & Kumar, C. N. S. V. (2024). A Multi-Step APT Attack Detection Using Hidden Markov Models by Molecular Magnetic Sensors. *Optical and Quantum Electronics, 56*(3), 282. https://doi.org/10.1007/s11082-023-05905-3

Contact information:

U. SAKTHIVELU, Assistant Professor
Department of Computer science and Engineering,
Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology,
Avadi, Chennai, 600062, Tamilnadu, India
E-mail: sakthiveluudayakumar@gmail.com

C. N. S. VINOTH KUMAR
(Corresponding Author)
Department of Networking and Communications,
College of Engineering and Technology (CET),
SRM Institute of Science and Technology, Kattankulathur-603203, Chennai,
Tamil Nadu, India
E-mail: vinothks1@srmist.edu.in