I. J. Computer Network and Information Security, **2024, 3, 144-158**
Published Online on June 8, 2024 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2024.03.10

Optimized Extreme Gradient Boosting with Remora Algorithm for Congestion Prediction in Transport Layer

Ajay Kumar*
Department of CSE, JECRC University, Jaipur, Rajasthan, India
E-mail: ajaychhotu@gmail.com
ORCID iD: https://orcid.org/0000-0002-3313-4860
*Corresponding author

Naveen Hemrajani
Department of CSE, JECRC University, Jaipur, Rajasthan, India
E-mail: naven_h@yahoo.com
ORCID iD: https://orcid.org/0009-0006-3655-9211

Received: 15 June 2023; Revised: 26 August 2023; Accepted: 01 December 2023; Published: 08 June 2024

Abstract: Transmission control protocol (TCP) is the most common protocol found in recent networks to maintain reliable communication. The most popular transport protocol in use today is TCP that cannot fully utilize the ability of the network because of the constraints of its conservative congestion control algorithm and favor's reliability over timeliness. Despite congestion is the most frequent cause of lost packets, transmission defects can also result in packet loss. In response to packet loss, end-to-end congestion control mechanism in TCP limits the amount of remarkable, unacknowledged data segments that are permitted in the network. To overcome the drawback, Optimized Extreme Gradient Boosting Algorithm is proposed to predict the congestion. Initially, the data is collected and given to data preprocessing to improve the data quality. Min-Max normalization is used to normalize the data in the particular range and KNN-based missing value imputation is used to replace the missing values in the original data in the preprocessing section. Then the preprocessed data is fed into the Optimized Extreme Gradient Boosting Algorithm to predict the congestion. Remora optimization is used in the designed model for optimally selecting the learning rate to minimize the error for enhancing the prediction accuracy in machine learning. For validating the proposed model, the performance metrics attained by the proposed and existing model are compared. Accuracy, precision, recall and error values for the proposed methods are 96%, 97%, 96% and 3% values are obtained. Thus, the proposed optimized extreme gradient boosting with the remora algorithm for congestion prediction in the transport layer method is the best method than the existing algorithm.

Index Terms: TCP, Min-Max Normalization, KNN-based Missing Value Imputation, Extreme Gradient Boosting Algorithm (XGBOST), Remora Optimization (ROA).

1. Introduction

During the 1980s, protocols that compose the present TCP/IP stack were initially established and communication systems that enable Internet access and backhaul connectivity have undergone a significant transformation [1]. By 2021, mobile traffic is anticipated to exceed server and desktop traffic [2]. The demand for online and multimedia traffic is rising, which is causing both mobile and fixed networks to evolve quickly [3]. New communication technologies have become accessible for a variety of networks and devices, enhancing the Internet throughout time. To gain full advantage of modern communication technologies, the community of researchers on the Internet has continued to work to expand the protocols at the transport layer are needed to function with modern network capabilities [4, 5]. The transport layer manages network congestion and offers reliable end-to-end transmission. Transmission Control Protocol (TCP), one of the most well-liked transport layer protocols, is heavily employed to accomplish this aim. Each device in the network exchanges information from one to another device anywhere in the network. As a result, Internet congestion also grows due to the bulk information sharing in the network. For wired and wireless networks, TCP is a dependable transport layer protocol that regulates the transmission rate in accordance with network congestion. It allows the global interconnection

This work is open access and licensed under the Creative Commons CC BY 4.0 License.

of all conventional computing devices that implement the TCP/IP architecture [6]. Congestion window referred as CWND to predict congestion in the last decades, but it is not applicable in today's world due to the dynamic and complicated network.

Traditional rule based approaches are primarily heuristic was not solve the complex issues in congestion control leading to poor performance [7]. Such that machine learning algorithms such as random forest, decision tree, k-nearest neighbor etc. had been utilized for predicting congestion in the network. However, accurate congestion prediction is not achieved using the existing approach and sometimes it can make the wrong prediction. In networks, a heavy traffic volume causes additional communication delays, low network throughput, and bandwidth and processing resources to be wasted [8]. As a result, it is important to classify and predict congestion and implement the required preventive measures in order to ensure optimal communication within the network [9, 10]. Some of the existing techniques for predicting the congestion will predict inaccurate due to the improper learning of the algorithm that can be occurred because of the over fitting issue in the existing machine learning algorithms. There have been numerous attempts to increase the effectiveness of the transport layer by utilizing various existing technologies, however, those approaches do not accurately estimate the congestion. In order to overcome this problem, a machine learning based optimized Extreme Gradient Boosting (XGB) algorithm has been created. The proposed methods primary contributions are as follows:

- Optimized Extreme Gradient Boosting with Remora Optimization algorithm is used to predict congestion in the transport layer at TCP protocol.
- NS2 based network simulator is used for generating the data to evaluate congestion between the nodes and also finding the performance of the designed model.
- Min-Max normalization and KNN based missing value imputation is used for linear transformation and to find the mean value for replacing the missing values.
- Remora optimization algorithm is used for selecting the tuned learning rate of the classifier to minimize the error rate.
- Extreme Gradient Boosting (XGBoost) classifier is used to predict the congestion of data in the transport layer network.

The forthcoming sections are organized as several articles that have been reviewed and that deal with the congestion control in the transport layer is discussed in section 2. Brief explanation and the experimental results of the proposed model is illustrated in section 3 and 4. Section 5 summarizes the entire research article.

2. Literature Survey

Internet is growing day by day due to a lot of users or populations in the internet and communication between users, information sharing between machine to machine, human to machine data transfer is increased in network can cause congestion in network. So various studies on prediction and control of congestion have been conducted in order to address these issues.

Kumar et al. [11] designed a multipath packet scheduling based on buffer acknowledgement for congestion control. Multipath Packet Scheduling (MP-PS) method in the model to effectively stream the data packets across several pathways. It comprises of a Multi-Path Packet Scheduler (MP-PS) to carry out the route scheduling based on the probability of the path scheduling delay and a Multipath Congestion Control (MCC) to control the rate of transmission based on packet delivery rate and buffer availability.

Kanagarathinam et al. [12] modelled a Dynamic TCP (D-TCP) congestion control mechanism for mm-Wave NR and LTE-A networks in the research. By calculating the channel bandwidth that are available, the designed D-TCP method are able to handle mm-Wave channel fluctuations. To calculate the congestion control factor N, the estimation of bandwidth was utilized. Based on the estimated management factor for congestion, the congestion window is raised or decreased in an adaptive manner. It assessed D-performance TCPs in comparison to heritage and existing TCP algorithms in terms of throughput, fairness and congestion window growth and demonstrated that the effect of underutilization in mobility gets reduced by D-TCP using simulations of mm-Wave NR during LOS -> NLOS transitions.

Verma and Kumar [13] developed an IoT based new congestion control algorithm in two ways were reactive and proactive mode. In order to quickly adjust the transmission rate whenever the available bandwidth and latency change, a new congestion management policy was introduced. The suggested method keeps a constant state to lower packet drop and produce higher throughput. This research also suggests flexible methods for preserving fairness when using TCP Cubic, which are commonly used. The experimental results showed that the suggested TCP performs better in terms of throughput and inter-protocol fairness.

Makarem et al. [14] designed a CoAP was a simple congestion control system based on binary exponential timeouts and subsequent retransmissions. In the model, improved mechanisms for CoAP were suggested. Although some thought about concentrated on enhancing the process of retransmission, others enhancing estimation of retransmission timeout. Examine recent and major ideas in-depth to expose their flaws. After that, consider two algorithms for congestion control are MBC-CoAP and IDC-CoAP, which enhance the estimation of a retransmission timeout for congestion prediction and suitably maintain the simplicity needed by restricted devices while using a rate-based strategy for congestion control.

Wei et al. [15] developed and introduced a novel shared bottleneck based congestion control (SB-CC) strategy that uses the ECN (Explicit Congestion Notification). To determine the level of congestion for every sub flow and to discover shared bottlenecks on sub flows, employ the ECN method. Then, using the level of congestion, the loads are evenly distributed over all sub flows via SB-CC, which also evens out variations in the congestion window. Additionally, designed a Forward Prediction packet Scheduling method based on Shared Bottlenecks (SB-FPS) to avoid throughput reduction caused by out-of-order packets. Because SB-FPS distributes data in accordance with how each sub flows window size varies, it could schedule data in shared bottleneck scenarios more precisely.

Akhtar et al. [16] implemented the congestion can be prevented by using the Bandwidth Aware Routing Scheme (BARS) by keeping track of the remaining bandwidth capacity in network paths and the amount of queue space that can be used to cache data. Before sending messages, the available and utilized bandwidth as well as the remaining cache must be calculated. The BARS make use of the feedback mechanism to inform the traffic source and alter the data rate in accordance with the bandwidth and queue availability in the routing path.

Polese et al. [17] analyzed the three primary research fields, outlining the major novelties in transport protocols that were recently presented: (i) the development of congestion control algorithms, with the aim of achieving optimum efficiency under difficult conditions, particularly through the use of machine learning methods; (ii) the implementation of new user-space protocols for transport that are alternatives to Transmission Control Protocol (TCP); and (iii) introduce multi-path functionality at the transport layer.

Though the above mentioned techniques performs better for congestion prediction and transmission of data, certain drawbacks are still a major issue for the reviewed articles such as buffer queue with data loss, delay of data transmission [11], high mobility leads to random packet loss [12], an issue like a congestion collapse arises when a protocol gets resources unevenly, slow convergence [13], packet mismatch due to non-verification of received data [14], degradation in quality of service [15], bandwidth issues in network path [16] and RTT error sensitivity [17]. Thus, to overcome these issues the proposed model based on Extreme Gradient Boosting with remora optimization algorithm was designed.

3. Proposed Methodology

TCP transmission is used by most modern online web services, which directly affects user satisfaction and business profitability. Most modern communications networks are constrained by the layered protocol design, which prevents individual components from being aware of how the network is being used by other components. It launches a particular protocol for the transmission of data and makes use of a transport layer to send data. One of the most critical events of the transport layer protocol is congestion prediction. A machine learning-based optimized Extreme Gradient Boosting (XGB) approach has been developed to predict congestion issues.

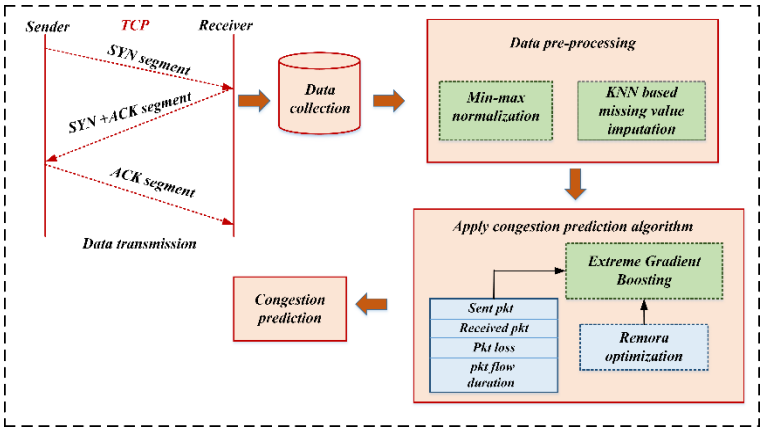


Fig.1. Architecture of proposed methodology

Figure 1 illustrates the architecture of the proposed methodology to predict the congestion in the transport layer. Data are initially transferred through the Transmission Control protocol. Then the collected raw data are pre-processed by utilizing min-max normalization and KNN based missing value imputation in order to improve the quality of raw data. Min-Max normalization is to transform the data linearly to normalize the data to the range of (0, 1) and data distribution remains unchanged. Furthermore, KNN based missing value imputation is used to replace the missing values in dataset by mean value of k neighbors. After preprocessing of dataset, it given to extreme gradient boosting (XGBoost) algorithm to classify the congestion in transport layer based on four main parameters are sent packet, received packet, packet loss, and duration of packet flow and also it contains some hyper parameters such as number of sub tress, maximum tree depth, learning rate, regularization, complexity control and minimum child weight. To classify the congestion by XGBoost method, optimization is required to improve the performance. For optimization, remora optimization technique is implemented to tune the crucial parameter as learning rate at the range between 0 and 1 for minimizing the error.

3.1. Transport Layer

Initially, data are transmitted via the Transmission Control Protocol (TCP) in the transport layer. The transport layer's primary responsibility is to provide end-to-end communication between the source and destination in the network. It plays a significant role in the protocol hierarchy that offers efficient and reliable, cost-effective data transmission services from the source to the destination. It provides services such as flow control, connection oriented services, congestion control and reliable transmission that improve the performance and quality of the communication network. The protocol stack is composed of seven layers. Each of them performs a certain function and follows a particular set of protocols that are operate in each layer. From that seven layer, only the transport layer is performed to predict the congestion. Transport layer is the fourth layer which is next to the network layer that performs data transfer services between nodes in the network. Various types of protocols are used in the transport layer for internet applications, here the protocol is considered as Transmission Control Protocol (TCP).

Transmission Control Protocol (TCP)

TCP is a transport layer protocol also known as a reliable protocol used to transfer data from sender to receiver. The primary function of TCP is to retrieve data from the application layer. The data is then divided into a number of packets, given a unique number, and transmitted to the destination. The messages are reassembled by the TCP and sent to the application layer from the other side. Since TCP is a connection-oriented protocol, the connection will remain active as long as the sender and receiver are still in communication.

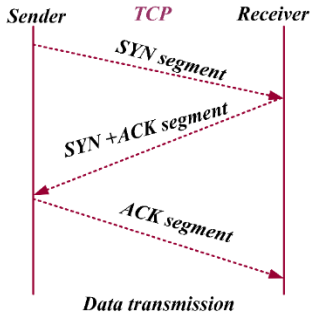


Fig.2. Transmission control protocol

Figure 2 demonstrates the TCP protocol. This implies, the logical link between the source and destination must be established first (setup phase). Data packets are transmitted following the establishment of the logical link (transmission phase). The packets of data are numbered and delivered in the correct sequence during transmission. Before delivering them to the application layer, TCP at the destination must reorder them if they are not received in the proper sequence. The logical connection is terminated after the transmission phase is finished. Additionally, TCP offers flow management, which synchronizes the source's data transmission rate with the destination's data receiving rate. The goal of flow management is to prevent the source node from sending too many data packets to the destination node. Additionally, TCP offers congestion management to prevent packet congestion in the network. The TCP source slows down packet transmission when it gets multiple acknowledgements for a sent packet or no acknowledgement at all. This recognize there is a network congestion. Moreover, TCP offers a reliable connection. Thus, packets that are corrupted at the destination are lost, and the TCP transmitter side must send them again. In a same manner, duplicate packets are rejected and missed packets must be transmitted again. Data transmission in transport layer is taken, where these collected data are given to data preprocessing.

3.2. Data Preprocessing

Data preprocessing is the fundamental step in process of data mining [18]. The methods used to collect data are typically lightly regulated, which leads to irregularities, incorrect data combinations, missing data, etc. As a result, before beginning any analysis, the representation and quality of the data must be considered. A successful data prediction by a machine learning may depend on a number of aspects, including reliability, quality, and availability. Knowledge discovery during the training process becomes particularly challenging when there is irrelevant information available or noisy and inaccurate data. So appropriate preprocessing step is frequently performed in order to minimize the impact of data disturbances on the execution of further processing processes. Collected data are taken for preprocessing, where data preprocessing techniques used as Min-Max normalization and KNN based missing value imputation.

A. KNN based Missing Value Imputation

The simplest method for imputing missing values is K-nearest neighbors (KNN) and this algorithm used to compute the missing values in the dataset. [19] The KNN method selects samples as a subset with high similarities to the gene containing incomplete data to impute missing values. Based on a certain set of data that is nearest to the object, KNN is a technique used to categorize objects. In order to perform classification, KNN measures the Euclidian distance between

data that is new (testing data) and data with previously learned classes (training data). K-nearest neighbors (KNN) defines a collection of nearest neighbors for a sample and replaces missing data by averaging its neighbors' non-missing values. This method is used to impute a value for a variable. When using KNN to handle missing data, the first step is to identify the number of closest neighbors, or nearest observations, denoted by *K*. Next, determine the shortest distance from each observation that is lacking no data. The following are the steps for using the KNN algorithm to impute missing data:

Define the parameter as K, where K represent the number of closest neighbors or nearest observations to be considered.

Determine the distance using additional j variables that are equivalent to the Euclidian distance formula between observations with incomplete data and complete observations on the i-th variable that do not have any missing data, specifically,

$$d(z_a, z_b) = \sqrt{\sum_{i=1}^{m}(z_{ai} - z_{bi})^2} \tag{1}$$

Where $d(z_a, z_b)$ is the distance between missing data observations and non-missing data observations, is the *i-th* variable value in each observation with missing data, i = 1,2,…,m, z_{bi} the value of the additional variables in each observation without a missing value with = 1,2,…,m.

According to the observation with the highest distance value to the observation with the lowest distance value, order the observations based on distance.

Using the least distance value, identify the K observations that are the closest.

Calculate the weight mean estimation value for missing data by using the following formula to the closest K observations that do not have missing data values:

$$\bar{z}_i = \frac{\sum_{k=1}^{K} w_k c_k}{\sum_{k=1}^{K} w_k} \tag{2}$$

where$\bar{z}_i$ is a weighted average that is estimated, *K* is the number of closest observations utilized, *K* is the observed value of *k*, c_k represent the value of the entire data on the variable including data that is missing based on observations from *k* and w_k is the weight of the *K-th* closest neighbor observation using the formula namely,

$$w_k = 1/d(z_{ak}, z_{bk})^2 \tag{3}$$

where $d(z_a, z_b)$ is the observation distance K.

B. Min-Max Normalization

Normalization is a scaling or mapping techniques, in which can find a new range from an old one. In order to preserve the wide difference in prediction and forecasting, the Normalization procedure is needed to bring them closer. By normalizing the data, data transformation was accomplished. The process of scaling attribute values of data such that they can be placed in a specific range is known as normalization. Min-Mix Normalization is a method that performs linear transformation to the original set of data. The difference between the maximum and minimum values in several dataset variables, such as Flow Duration, Packet Length, Flow Packets and Flow has a very wide range. The term "Min-Mix Normalization" refers to a method that preserves the relationships between the original data. A simplistic technique called min-max normalization that allows data to be fit precisely within a predefined boundary with a predefined boundary. Data is normalized in the range of (0, 1) and no changes in the data distribution. Its normalization a value of R to $\acute{w}$ in the range [new-$min_r, new - max_r$] by evaluating as

$$\acute{w} = \left(\frac{w - min_r}{max_r - min_r}\right) * (new - max_r) + new - max_r \tag{4}$$

where w is the range of original data.

If the data preprocessing methods are Min-Max normalization is normalized the range as (0, 1) and KNN based missing value imputation performs to replace the missing values in dataset by k-neighbors. These preprocessed data is given to congestion prediction algorithm to detect the congestion.

3.3. Extreme Gradient Boosting Algorithm

It originated as an enhanced version of the gradient boosting methodology and is a supervised machine-learning strategy based on the ensemble method. [20] Utilizing additive methods, the XGBoost algorithm combines the predictions of weak learners to create a robust learning model. Along with to being quick and extremely efficient, the XGBoost classifier addresses the overfitting problem and maximizes the use of computational resources. Such advantages result from the objective functions being simplified to enable parallel execution during training and including the use of regularization and predictive terms. Using the XGBoost algorithm, the first learner is fitted to the entire set of data. The errors of the first learners are then transformed into the second learner. The final prediction model is then produced by

adding the predictions of all learners, and this process is continued until a stopping condition is satisfied.

This extreme gradient boosting classifier consist of certain hyperparamter which estimates the classification accuracy. And it contains some hyper parameters such as number of sub tress, learning rate, regularization, complexity control and minimum child weight. Tuning these hyper parameter helps to improve the classification performance. Figure 3 shows the flow diagram of Optimized Extreme Gradient Boosting Algorithm.

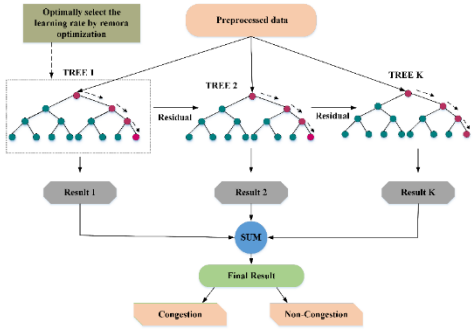


Fig.3. Diagram representation of ROA-XGB

Consider a data set with the following expression:

$$D = \{(p_i, q_i)\} where P_i \in R^c, q_i \in R \tag{5}$$

Where *c* indicates the features dimension P_i and q_i represented sample response *i*. In a situation where $|.|$ signifies the cardinality of a set, indicate by m the number of samples such that $|D| = m$. Compute the entry predicted value of *i*, $\widehat{q_i}$ as,

$$\widehat{q_i} = \sum_{k=1}^{K} g_k(p_i),\ g_k \in G, \tag{6}$$

Predicted score provided by the *k-th* tree *i-th* sample is referred to as $g_k(p_i)$ .where g_k denotes an independent tree in G, a set of regression trees. The XGBoost objective function, designated by $\mathcal{L}$, is stated below:

$$\mathcal{L} = \sum_{i=1}^{n} l(q_i, \widehat{q_i}) + \sum_{k=1}^{K} \Omega(g_k) \tag{7}$$

The $g_k$functions of the regression tree model can be learned by minimizing the objective function$\mathcal{L}$. The difference between actual value q_i and prediction $\widehat{q_i}$ is determined by the training loss function $l(q_i, \widehat{q_i})$. Here, the term Ω is implemented in order to avoid the overfitting issue by penalizing the complexity of the model in the following ways,

$$\Omega(g_k) = S\gamma + \frac{1}{2}\lambda\|w\|^2 \tag{8}$$

Where the regularization parameters are γ and λ, the scores on each leaf and the numbers of leaves are denoted by *w* and *S*. The objective function can be approximately described by a second degree Taylor series. Let I_j be an instance set of leaf *j* with the fixed structure *x(q)*, where $I_j = \{i|x(q_i) = j\}$. The following equations can be used to determine the optimal values and weights w_j* for leaf j:

$$w_j * = -\frac{f_j}{h_j + \lambda} \ and \mathcal{L} * = -\frac{1}{2} \sum_{j=1}^{S} \frac{\left(\sum_{i \in I_j} f_i\right)^2}{\sum_{i \in I_j} h_i + \lambda} + \lambda S \tag{9}$$

Where the loss function $\mathcal{L}$ first and the second gradient orders are denoted by f_i and h_i respectively. The tree structure *q* quality score can be calculated using the loss function$\mathcal{L}$. The model is more accurate if the score is lower. A greedy approach can resolve the issue by beginning with a single leaf and repeatedly adding branches to the tree because it is impossible to list all tree structures. Consider the instance sets of the right and left nodes after splitting as I_r andI_l. Assuming $I = I_r \cup I_l$, then, after the split, the loss reduction will be shown as follows:

$$\mathcal{L}_{split} = \frac{1}{2}\left[\frac{\left(\sum_{i \in I_l} f_i\right)^2}{\sum_{i \in I_l} h_i + \lambda} + \frac{\left(\sum_{i \in I_r} f_i\right)^2}{\sum_{i \in I_r} h_i + \lambda} - \frac{\left(\sum_{i \in I} f_i\right)^2}{\sum_{i \in I} h_i + \lambda}\right] - \gamma \tag{10}$$

In practice, this formula is generally used to evaluate the split candidates. The leaf nodes are scored during splitting in the XGBoost model, which employs several basic trees. The left, right, and original leaf scores are represented,

respectively, by the first three terms of the equation. Additionally, the term regularization" γ" on the additional leaf will be used during the training phase.

Generally, the learning rate is a crucial hyperparameter which governs the entire training process. If the value is too little, the training process may take a long time and become stuck, whereas if the value is too large, the training process may become unstable or learn an unsatisfactory set of weights quickly. So, optimal selection of learning rate is essential to improve training accuracy.

3.4. Remora Optimization

Remora can swim mounted on whales, which uses less energy, and also protected from dangerous from the adversary. [21] The remora separates from its host (the whale) in a circumstance where the sea is filled with food, once the food has been consumed and digested, it gets placed back on another level and moved to a different area of the sea. In this research, the learning rate is tuned using remora optimization algorithm. The learning rate must range between 0 and 1. The fitness function provided for this optimization algorithm is to minimize error and optimally select the parameter. The flow diagram of Remora Optimization algorithm is illustrated in figure 4.

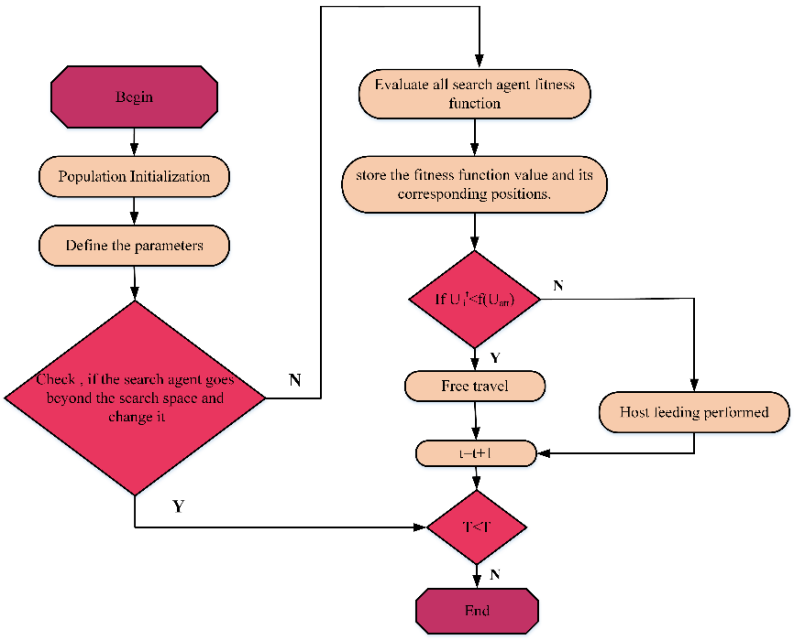


Fig.4. Flow diagram of remora optimization algorithm

A. Preparation (Initialization)

The candidate's response is treated as a remora in the ROA, and the position (U) of the problem variable in the search space has been chosen. As it reaches the top of the one-dimensional space, the remora's position changes. The position of remoras in current state as: $U_{i1} = (U_{i1}, U_{i2}, \ldots, U_{id})$, where i refer to the remora number and d denotes its dimension, respectively.

To put in another way, $U_{best} = (U_1^*, U_2^*, \ldots, U_d^*)$ describes the remora's biological behavior in relation to food (target), and denotes the best solution to the ROA. Each solution has a competency fitness, which is described as $f(U_i) = f(U_{i1}, U_{i2}, \ldots, U_{id})$. The highest deserve amount that is compatible with the best position of remora as $f(U_{best}) = f(U_1^*, U_2^*, \ldots, U_d^*)$.

B. Free Travel (Exploration)

The Swordfish optimization (SFO) approach, which is based on the elite method employed in the swordfish algorithm, is used by the ROA to execute the global search. The following formula can be used to update a position:

$$U_i^{t+1} = U_{best}^t - (rand(0,1) \times \left(\frac{U_{best}^t + U_{rand}^t}{2}\right) - U_{rand}^t) \quad (11)$$

where T indicates the maximum number of iterations, t denotes the current iteration, and U_{rand} represents a random position chosen. Exploring the search space is another requirement for the random selection of remora. Whether or not the host has consumed the prey affects the remora's decision about which host to choose. In addition, compared to the previous generation, the eligibility rate at the moment is better. As a result, the history of the attack is used to determine the present worth of competency.

Attack experience: The remora must continuously take short steps close to the host to determine whether to switch the host or not based on fitness level. The following is a model for this behavior.

$$U_{att} = U_i^t + (U_i^t - U_{pre}) \times randn \tag{12}$$

where U_{att} and U_{pre}stand for the test step and the position of the previous generation. Similarly, randn indicates short global step done at random by the remora. The host's status is then evaluated at random by the remora. In other words, the fitness value of the current response f(U_{it}) and tested response f(U_{att}) are compared. When a test's fitness value is smaller than a response's current competency value, the following happens:

$$f(U_i^t) > f(U_{att})$$

For local optimization in this situation, the remora chooses one of the feeding techniques. The remora chooses the host if the tested value of fitness is higher than the value of fitness in current response:

$$f(U_i^t) < f(U_{att})$$

C. Eat Thoughtfully (Exploitation)

Remora can also attach itself to humpback whales when searching for food. Remora will consequently move similarly to humpback whales. The ROA algorithm uses the WOA (Whale Optimization Algorithm) technique to carry out the local search. The updating formulas for modified position are as:

$$U_{i+1} = D \times e^{\alpha} \times \cos(2\pi\alpha)\, U_i \tag{13}$$

$$\alpha = rand \times (c - 1) + 1 \tag{14}$$

$$c = -(1 + \frac{t}{T}) \tag{15}$$

$$D = |U_{best} - U_i| \tag{16}$$

If Remora is on a whale in the larger solution space, their locations can be regarded as being the same. The distance between the hunter and the prey is called *D*, α is a value chosen at random from [−1, 1], and *c* is a number whose value falls exponentially between [−2, −1].

Host feeding: The process of exploitation includes host feeding as another division. The solution space can now be reduced to the host's location space. Small steps can be considered to be moving on or around the host, which can be expressed as follows:

$$U_i^t = U_i^t + A \tag{17}$$

$$A = B \times (U_i^t - C \times U_{best}) \tag{18}$$

$$B = 2 \times V \times rand(0,1) - V \tag{19}$$

$$V = 2 \times \left(1 - \frac{t}{\max\_iter}\right) \tag{20}$$

Thus, the letter A was utilized to signify a slight movement that was connected to the host and remora's volume space. A remora factor C was employed to reduce the position of the remora in order to distinguish between the host and remora's locations in the solution space. If the host has the volume 1, then the volume of the remora is about a portion of that of the host. Thus, the performance is improved in this extreme gradient boosting method using Remora optimization algorithm for congestion prediction in transport layer.

3.5. Optimal Feature Selection for Proposed Methodology

Some of the steps evaluated for selecting optimal features using Remora optimization are initiation, fitness function, updating, and termination.

Step 1: Initialization

Initialization of remora population is replaced with the tuning parameters of the XGboost algorithm such as eta, max_depth, gamma and alpha. These gathered parameters are considered as population $U_1, U_2, U_3 \ldots U_n$.

$$F = (U_1, U_2, U_3 \ldots U_n) \tag{21}$$

Step 2: Fitness Function

The fitness function for tuning the classifier parameter is evaluated by maximizing the coefficient of validation based on the Monte Carlo technique.

$$fitness = \{ maximize(MCCV) \tag{22}$$

$$MCCV = \frac{1}{Nn_v}\sum_{i=0}^{N}\left\|A_{R_c(i)} - \hat{A}_{R_v(i)}\right\|^2 \tag{23}$$

Monte Carlo cross validation is simple and effective method that randomly split the samples into two parts. The first part (calibration set), denoted as R_c , contains n_c samples for fitting the models. The second part (validation set), denoted as R_v , contains $n_v = n - n_c$ samples for validating the model.

Step 3: Updating

The modified position updating formulas are as follows based on WOA (Whale Optimization Algorithm) strategy follows in equation (13).

Step 4: Termination

After attaining, the best tuning parameters for the XGboost classifier algorithm. The entire optimization process will get terminated.

Algorithm 1: Pseudo Code For Optimized Extreme Gradient Boosting With Remora Algorithm For Congestion Prediction In Transport Layer

```
#Node deployment
N= N1,N2,N3….Nn
SA= Node deployment (N)
FD= Fetching data (SA)
#Preprocessing
K= KNN (FD) // KNN based missing value replacement is determined by using the equation (3)
MM= Min-max (K) // Min-max normalization algorithm is determined by using the equation (4)
# Remora Optimization
Initialize(L) = LR1, LR2, LR3….LRN // Learning rate parameters of XG-Boost are initialized
for x in L
    clc=XGboost (MM, L(x))
    CV=MCCV (clc) // Monte Carlo validation is determined for the learning rate using the equation (23)
    print(CV)
end
OL= maximize (CV) // Optimized learning rate is selected based on the maximized cross validation accuracy.
#Classification
X=XGBoost (OL) // Prediction of Congestion using the equation (7)
    If (X=0)
    Print(Congestion)
    Else
    Print (Non-congestion)
End
```

Output: Prediction of Congestion in transport layer

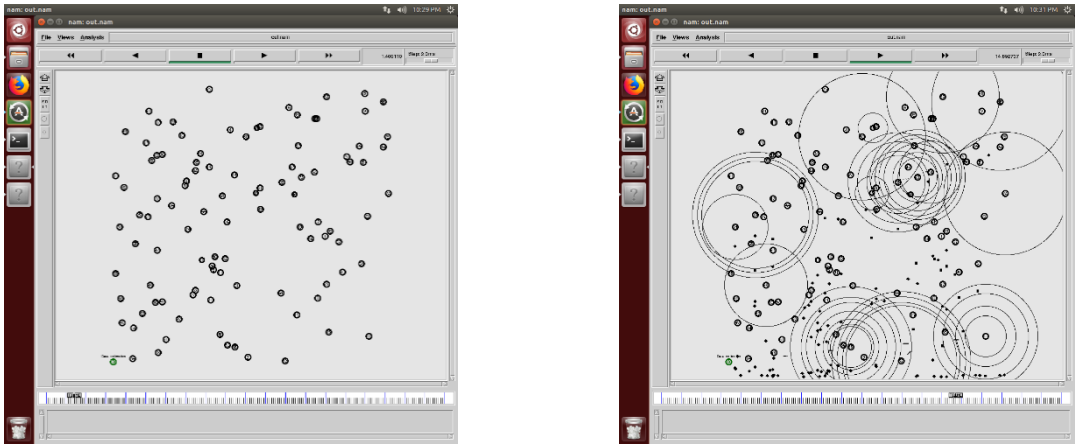
4. Results and Discussion

Based on Optimized Extreme Gradient Boosting with Remora Algorithm, the proposed method is used for Congestion Prediction in Transport Layer. To test the designed model, the Python integrated with spyder IDE (python 3.8) and NS2 simulator of version 2.35 is employed with Intel i5-10th processor working at 2.50GHz, 32GB RAM and a 64-bit operating system. In this developed methodology, at first, data passes through using the Transmission Control protocol. The obtained raw data are then pre-processed using KNN-based missing value imputation and min-max normalization to enhance the precision. After the dataset has been preprocessed, the extreme gradient boosting (XGBoost) technique is used to predict the congestion in the transport layer. For accurate prediction of congestion, remora optimization is used for select the learning rate optimally in this designed model.

4.1. Simulation Results

Figure 5 (a) and (b) illustrates the node deployment and transmission of packet in the network. Simulation refers to the process of creating a computerized representation of a system in the real world for behavioral testing. Before establishing a complicated network, simulation can be used to examine the network's performance and behavior. In this research, a well-known network simulator named NS2 is used to run the complete simulation. In this simulation, 100 nodes are had been distributed in a 1000 m × 1000 m region in the network. Nodes in the network transmit the packets

from one node to another. Sender node send the packet to receiver node where connection establishment takes place. If the receiver node gets the SYN packet from the sender node, it reacts by sending an ACK (Acknowledgement Sequence Number) packet or SYN/ACK packet as a confirmation receipt. The sender node acknowledges the receiver response in the final section, and then they both create a secure connection to begin the actual data transmission. Thus, the data transmission occurs. If any congestion happens, then the packet loss happens in the transmission. The maximum number of packets delivered and received, with the lowest possible packet loss and packet flow duration is the congestion free packets. Suppose the packet transmitted from sender to receiver with highest possibility of packet loss and duration of packet flow causes congestion in the transport layer. By using this simulation, nodes are collected and makes the dataset for further processing.

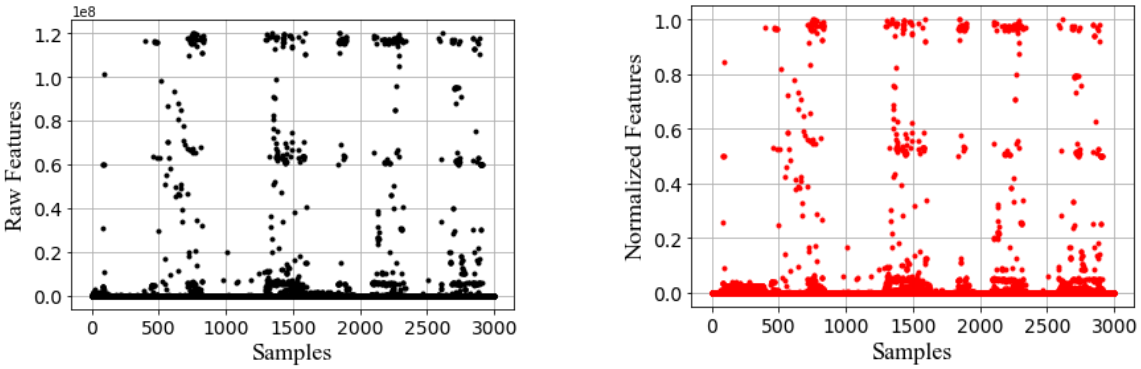


(a) (b)

Fig.5. (a) Node deployment and (b) Packet transmission between 100 nodes

4.2. Normalization

Raw features of proposed method are shown in figure 6 (a) which shows the data are scattered in the original collected data. Figure 6 (b) depicts the normalized features using Min-Max normalization method to normalize the data within the range of 0 to 1. In the raw features data are spread from o to 1.2 after applying the min-max normalization the data are scaled from 0 to 1. Within the boundary scattered data are located and easy to analyze the data in the proposed machine learning.



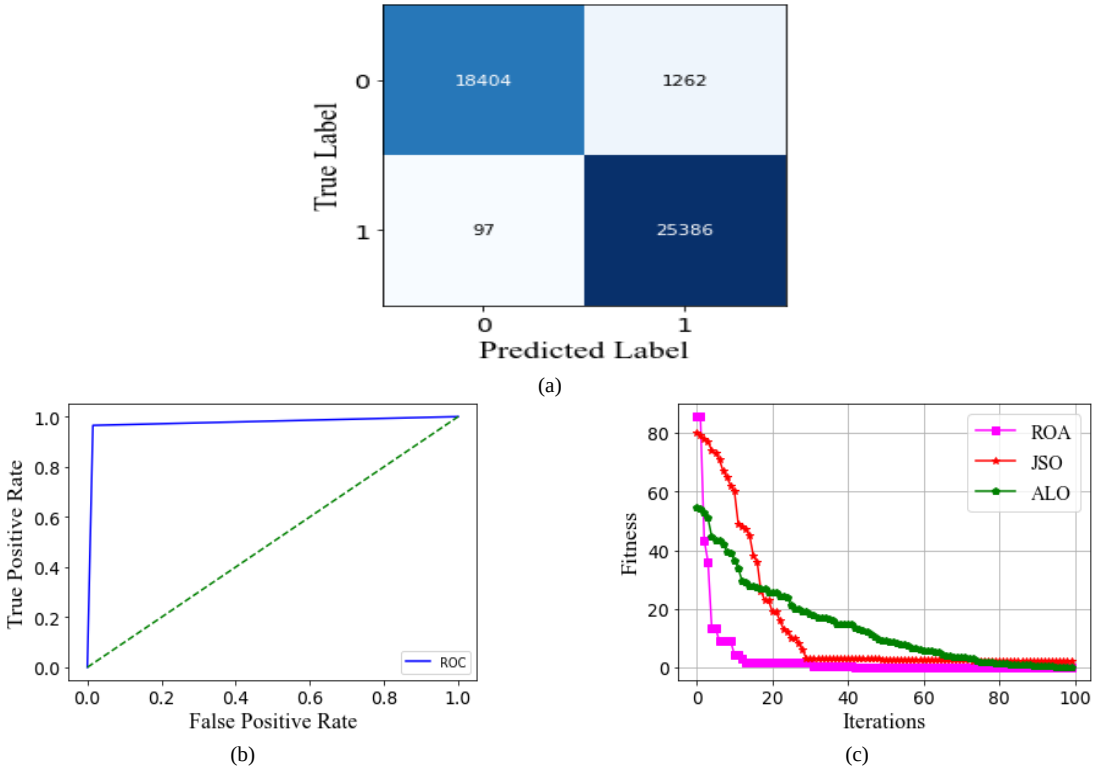
(a) (b)

Fig.6. (a) Raw features of proposed method and (b) Normalized features of proposed method

Dataset Confusion Metrics

The confusion metrics of the proposed model are shown in Figure 7 (a). It is used to determine a classification based on machine learning algorithm for efficiency to differentiate between correct and wrong predictions. A classifiers entire predicted and actual values are plotted in a table. The total number of data used for testing is 45149, in which the predicted data for the 0 and 1 classes are 18404 and 25386, respectively that determines 943 are incorrectly predicted and 43790 are predicted based on the actual class. The ROC curve plot for the proposed model is illustrated in Figure 7 (b). The receiver operating characteristic curve, or ROC graph, shows how well the classification model performs across all thresholds. Optimization fitness value is compared for the proposed Remora Optimization Algorithm (ROA) and the existing optimizations are Jellyfish Search Optimizer (JSO) and Ant Lion Optimizer Algorithm (ALO) are illustrated in figure 7 (c). Fitness value is a type of objective function designed to discover the optimal solution for a certain problem by including all of the parameters. The main parameters taken as learning rate to optimally select the best solution for the

machine learning algorithm to predict the congestion accurate. From graph represent that the Remora Optimization is the best optimization to predict the value based on desired output when compared to other existing algorithms.



(a) (b) (c)

Fig.7. (a) Confusion metrics of proposed method, (b) Receiver Operating Characteristic curve for proposed method and (c) Optimization Fitness value comparison of proposed method and existing optimization model

4.3. Evaluation of Performance Metrics for XGBOOST

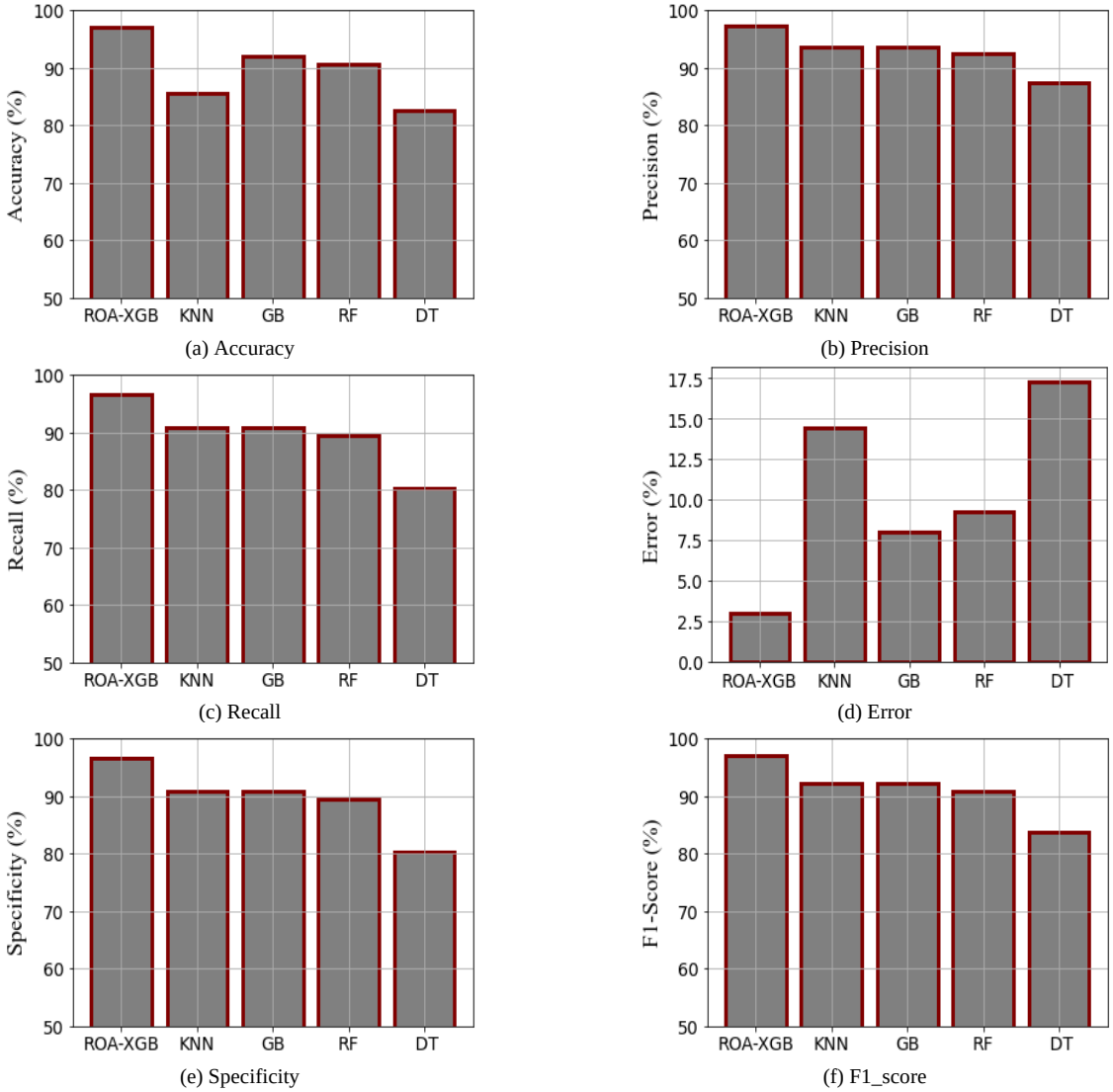
Performance metrics for the designed ROA-XGB model are evaluated involving accuracy, precision, recall, specificity, F1 score, error, False Negative Rate (FNR), Negative Predictive Value (NPV), False Positive Rate (FPR), False Discovery Rate (FDR), False Omission Rate (FOR), kappa, and Mathew's correlation coefficient (MCC) is compared with existing machine learning such as K-Nearest Neighbor (KNN), Gradient Boosting (GB), Random Forest (RF), Decision Tree (DT) algorithm.

Figure 8 (a) compares the proposed ROA-XGB and existing techniques in terms of accuracy. This examination compares the accuracy of proposed model ROA-XGB with that of existing methods such as KNN, GB, RF and DT machine learning algorithms. The accuracy values attained for the proposed and existing models are 96.9, 85.6, 92.02, 90.7, and 82.7 respectively. Accuracy for proposed method of extreme gradient boosting algorithm with remora optimization is 96%. The proposed model has higher accuracy and serves more accurate for predicting the congestion, when compared to earlier methods. Figure 8 (b) depicts the precision outcome of the proposed ROA-XGB versus the existing techniques. Comparative analysis is done between the proposed methodology and existing machine learning methods as KNN, GB, RF and DT. Values of precision for both proposed and existing models are 97.3, 93.6, 93.6, 92.4, and 87.4. When compared to existing techniques, the precision rate is greater. The recall results of the proposed ROA-XGB compared to existing methods are shown in Figure 8 (c). The algorithms like KNN, GB, RF and DT models have achieved recall values of 96.6, 90.8, 90.8, 89.5, and 80.2 respectively. This result reveals that the proposed ROA-XGB model has a better recall compared to existing model. The error comparison between the performances of the optimized proposed method is illustrated at Figure 8 (d). The proposed ROA-XGB and existing machine learning methods such as KNN, GB, RF and DT obtained error values are 3.1, 14.4, 8, 9.3 and 17.3. The ROA-XGB proposed model works better than the other model, determined by the obtained error values.

Figure 8 (e) shows the comparison of specificity for proposed and existing machine learning algorithms. The obtained specificity values for the proposed ROA-XGB and existing techniques are 96.6, 90.8, 90.8, 89.5 and 80.2. The proposed ROA-XGB model achieves a greater level of specificity when compared to existing approaches. Evaluation of proposed and existing machine learning methods for F1_score is demonstrated in figure 8 (f). Precision and Recall are combined to form the F1_Score. The obtained F1_Score values for the following models are 96.9, 92.2, 92.2, 90.9, and 83.7 for ROA-XGB, KNN, GB, RF and DT. Figure 8 (g) shows the comparison of both proposed and existing models for negative predictive value. The values for Negative Predictive Value of proposed ROA-XGB and existing machine learning algorithms likes KNN, GB, RF and DT are 97.3, 93.6, 93.6, 92.4, and 87.4. As a result, the proposed method ROA-XGB has a greater F1_Score and NPV than the existing model. Figure 8 (h) shows the validation of FNR for

proposed ROA-XGB and the existing approaches. The false negative rate values are 3.4, 9.2, 9.2, 10.5 and 19.8 for the proposed ROA-XGB and existing models such as KNN, GB, RF and DT for comparison. Determination of False Positive Rate (FPR) for proposed and existing model is depicted in figure 8 (i). The values attained for proposed ROA-XGB method and existing models are 3.4, 9.2, 9.2, 10.5 and 19.8. According to the obtained FNR and FPR values of the proposed and existing model, the proposed method performs better.

Figure 8 (j) illustrates the FOR for proposed ROA-XGB and existing models. The occurrence of false-negative values to total negative values predicted as false and true is represented as the false omission rate. Values for false omission rate for proposed ROA-XGB method and existing machine learning algorithms are 2.7, 3.4, 3.4, 7.6 and 12.6 respectively. Figure 8 (k) represent the validation of FDR for the proposed ROA-XGB and existing approaches. Proposed method FDR value is 2.7 and the existing KNN, GB, RF and DT machine learning values are 3.4, 3.4, 7.6 and 12.6. Thus, the FOR and FDR value of the proposed method is less compared to the existing machine learning algorithm. The validation of MCC and kappa metrics for proposed ROA-XGB and existing approaches are illustrated at figure 8 (l) and (m). The attained kappa values of proposed ROA-XGB and existing models are 93.8, 69.6, 83.4, 80.7, and 63.26. MCC is a measurement of the test precision. The attained MCC values of ROA-XGB, KNN, GB, RF and DT are 93.9, 72.5, 84.5, 81.9 and 67.3. As a result, the proposed methods kappa and MCC values are higher compared with each of the existing approaches.



(a) Accuracy

(b) Precision

(c) Recall

(d) Error

(e) Specificity

(f) F1_score

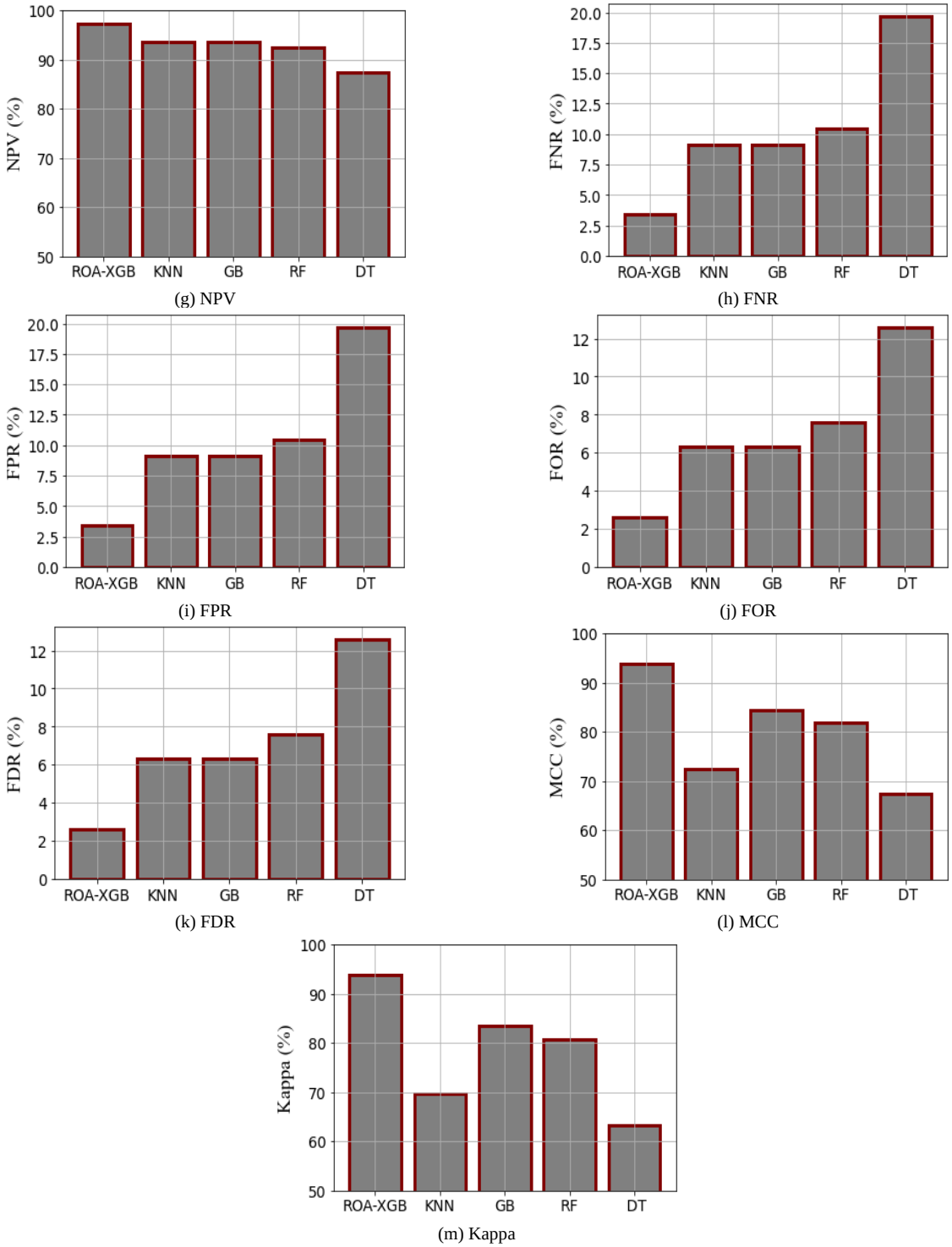


Fig.8. (a) – (m) Comparison of performance metrics for proposed and existing model

5. Conclusions

Current network has become more and more complicated due to population leads to congestion and it is critical to employ Machine learning techniques to create effective congestion control algorithms. Implementing the Optimized Extreme Gradient Boosting ML algorithm that is created for congestion prediction in the proposed method. In this method, Data are collected from the simulation results of the nodes and are pre-processed using the min-max normalization and K-means missing value replacement. After preprocessing the data, optimized XGBOOST machine learning algorithm based on remora optimization is used for classifying the congestion prediction on the lowest probability of packet loss and highest probability of data transmission. The proposed methods performance metrics such as accuracy, precision, recall, and error are compared to the results from existing models. Values for performance metrics of proposed method are 96.9, 97.3, 96.6 and 3.1. These estimated values are compared to the outcomes of other machine learning approaches

such as DT, GB, RF and KNN. The performance metric value is higher when compared to other existing techniques and it represent the optimized Extreme Gradient Boosting method has higher accuracy in this designed model. In future, the congestion attained in a network can be predicted based on this model and can be upgradable to a strategic avoidance method for establishing a congestion-free network in the transport layer for many sectors such as military, health care, real time traffic scenario, industrial companies for loss less and fast communication of data.

Acknowledgment

Funding: The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Conflict of Interest: The authors declared that they have no conflicts of interest to this work. We declare that we do not have any commercial or associative interest that represents a conflict of interest in connection with the work submitted.

Availability of data and material: Not applicable

Code availability: Not applicable

Author contributions: The corresponding author claims the major contribution of the paper including formulation, analysis and editing. The co-author provides guidance to verify the analysis result and manuscript editing.

Compliance with ethical standards: This article is a completely original work of its authors; it has not been published before and will not be sent to other publications until the journal's editorial board decides not to accept it for publication.

References

[1] K. Wang, Y. Liu, X. Liu, Y. Jing, and S. Zhang, "Adaptive fuzzy funnel congestion control for TCP/AQM network," *ISA transactions*, vol. 95, pp. 11-17, 2019.

[2] SJSA. Fathima, T Lalitha, F. Ahmad, and S. Karthick, "Unital Design Based Location Service for Subterranean Network Using Long Range Topology," *Wireless Personal Communications*, vol. 124, pp. 1815-1839, 2022.

[3] J. Huang, S. Li, R. Han, and J. Wang, "Receiver-driven fair congestion control for TCP outcast in data center networks," *Journal of Network and Computer Applications*, vol. 131, pp. 75-88, 2019.

[4] Y. Bai, and Y. Jing, "Event-triggered network congestion control of TCP/AWM systems," *Neural Computing and Applications*, vol. 33, pp. 15877-15886, 2021.

[5] O. Lamrabet, N. El Fezazi, F. El Haoussi, E. H. Tissir, and T. Alvarez, "Congestion control in TCP/IP routers based on sampled-data systems theory," *Journal of Control, Automation and Electrical Systems*, vol. 31, pp. 588-596, 2020.

[6] Z. Xu, J. Tang, C. Yin, Y. Wang, and G. Xue, "Experience-driven congestion control: When multi-path TCP meets deep reinforcement learning," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1325-1336, 2019.

[7] S. Karthick, SP. Sankar, and YPA. Teen, "Trust-Distrust Protocol for Secure Routing in Self-Organizing Networks," In *2018 International Conference on Emerging Trends and Innovations in Engineering and Technological Research (ICETIETR)*, pp. 1-8, 2018.

[8] R. Al-Saadi, G. Armitage, J. But, and P. Branch, "A survey of delay-based and hybrid TCP congestion control algorithms," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3609-3638, 2019.

[9] B. Jaeger, D. Scholz, D. Raumer, F. Geyer, and G. Carle, "Reproducible measurements of TCP BBR congestion control," *Computer Communications*, vol. 144, pp. 31-43, 2019.

[10] M. Swarna, and T. Godhavari, "Enhancement of CoAP based congestion control in IoT network-a novel approach," *Materials Today: Proceedings*, vol. 37, pp. 775-784, 2021.

[11] A. Kumar, P. V. Srinivas, and A. Govardhan, "A multipath packet scheduling approach based on buffer acknowledgement for congestion control," *Procedia Computer Science*, vol. 171, pp. 2137-2146, 2020.

[12] M. R. Kanagarathinam, S. Singh, I. Sandeep, H. Kim, M. K. Maheshwari, J. Hwang, A. Roy, and N. Saxena, "NexGen D-TCP: Next generation dynamic TCP congestion control algorithm," *IEEE Access*, vol. 8, pp. 164482-164496, 2020.

[13] L. P. Verma, and M. Kumar, "An IoT based congestion control algorithm," *Internet of Things*, vol. 9, pp. 100157, 2020.

[14] N. Makarem, W. B. Diab, I. Mougharbel, and N. Malouch, "On the design of efficient congestion control for the Constrained Application Protocol in IoT," *Computer Networks*, vol. 207, pp. 108824, 2022.

[15] W. Wei, K. Xue, J. Han, D. S. Wei, and P. Hong, "Shared bottleneck-based congestion control and packet scheduling for multipath TCP," *IEEE/ACM Transactions on Networking*, vol. 28, no. 2, pp. 653-666, 2020.

[16] N. Akhtar, M. A. Khan, A. Ullah, and M. Y. Javed, "Congestion avoidance for smart devices by caching information in MANETS and IoT," *IEEE Access*, vol. 7, pp. 71459-71471, 2019.

[17] M. Polese, F. Chiariotti, E. Bonetto, F. Rigotto, A. Zanella, and M. Zorzi, "A survey on recent advances in transport layer protocols," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 4, pp. 3584-3608, 2019.

[18] Z. Zou, Y. Yang, Z. Fan, H. Tang, M. Zou, X. Hu, C. Xiong, and J. Ma, "Suitability of data preprocessing methods for landslide displacement forecasting," *Stochastic Environmental Research and Risk Assessment*, vol. 34, pp. 1105-1119, 2020.

[19] W. C. Lin, and C. F. Tsai, "Missing value imputation: a review and analysis of the literature (2006–2017)," *Artificial Intelligence Review*, vol. 53, pp. 1487-1509, 2020.

[20] H. A. Alamri, and V. Thayananthan, "Bandwidth control mechanism and extreme gradient boosting algorithm for protecting software-defined networks against DDoS attacks," *IEEE Access*, vol. 8, pp. 194269-194288, 2020.

[21] H. Jia, X. Peng, and C. Lang, "Remora optimization algorithm," *Expert Systems with Applications*, vol. 185, pp. 115665 2021.

Authors' Profiles

Ajay Kumar received an M.Tech. (CSE) from Rajasthan Technical University, Kota in 2009 and B.Tech. (CSE) from Dr. B.R.A. University, Agra in 2001. He is currently working as an Assistant Professor at JECRC University and is pursuing a Ph.D (CSE) from JECRC University, Jaipur. His research interests include mobile communications, wireless networks and machine learning.

Prof. (Dr.) Naveen Hemrajani, Dean School of Engineering & Technology, JECRC University is BE, M.Tech. and PhD Computer Science & Engineering has more than 31 years of Teaching and Research Experience. His research interests include Network Security, MANET, Software Engineering, Cloud Computing, Machine learning and Data Science. He has published three books and many research papers in International and National Journals of repute.

How to cite this paper: Ajay Kumar, Naveen Hemrajani, "Optimized Extreme Gradient Boosting with Remora Algorithm for Congestion Prediction in Transport Layer", International Journal of Computer Network and Information Security(IJCNIS), Vol.16, No.3, pp.144-158 2024. DOI:10.5815/ijcnis.2024.03.10