ELITISM DRAGONFLY OPTIMIZATION (ELDFO) BASED QoS OPTIMIZATION AND EDGE SYSTEM RECOMMENDATION (ESR) FOR CLOUD COMPUTING

[1*]Ms. N. Menaka, [2]Dr.(Mrs).Jasmine Samraj

Research Scholar (Ph.D.), PG and Research Department of Computer Science, Quaid-E-Millath Government College for Women (Autonomous), Chennai.

menaka.research2021@gmail.com

Associate Professor,PG and Research Department of Computer Science, Quaid-E-Millath Government College for Women (Autonomous), Chennai.

menaka.research2021@gmail.com,dr.jasminesamraj@qmgcw.edu.in

Abstract

Recommender systems were utilised as an effective way to filter out unnecessary data and make recommendations for the helpful products in the face of the exponential growth of data that is accessible through the Web. The widespread use of smart phones, smart wearables, and other Internet of Things (IoT) devices has steadily fuelled the emergence of numerous unique services that are latency-sensitive, computationally intensive, and have improved service quality. In such cases, the data sources have four essential traits: heterogeneity, mobility, volatility, and sparsity. Traditional recommender systems using cloud computing are unable to uncover consumer demand data. In order to deliver more intelligent and individualised service, edge computing is a unique computing paradigm that pushes computation/storage resources from a cloud-based server to the network edge servers. In existing work suggested a hybrid convolutional neural network with long short-term memory (HCNN-LSTM) technique based Edge System Recommendation (ESR) Algorithm for Cloud Service Provider (CSP). However, existing work did not achieve the sufficient effective entity identification accuracy and the QoS requirements in the edge are not satisfied. To overcome the above-mentioned problem in this work introduced a hybrid convolutional neural network with long short-term memory (HCNN-LSTM) technique-based Edge System Recommendation (ESR) Algorithm for Cloud Service Provider (CSP). With this algorithm CSP can easily select Edge Systems. Securing cloud server data uses Elliptical Curve Cryptography (ECC) method to improve the accuracy and satisfy QoS requirements in the edge, an ELitism Dragonfly Optimization (ELDFO) has been introduced for edge.It conducts accurate entity recognition using on the HCNN-LSTM while considering entity feature

data.The outcomesshow the effectiveness of the suggested model intems of CPU utilization, Memory Utilization, Service time, QoS violation and Attack detection.

Keywords: Recommender system, Cloud computing, Edge System, Elliptical Curve Cryptography and ELitism Dragonfly Optimization.

1.Introduction

Currently, Big Data and Artificial Intelligence (AI) are being used to great effect in ubiquitous recommender systems, which are a vital and essential innovation. It can proactively filter out unrelated data in the rapidly expanding amount of data accessible through the Web and make recommendations to users using a recommendation algorithm for the information they would find most helpful (news, services). The capacity of the recommendation algorithm to predict an individual's interests and needs using an analysis of the user's personalised opinions, the qualities of the item, the user/item's prior contact, and other data is one of its important features. These days, recommender systems are starting to gain popularity in a variety of access to data systems and have been effectively implemented in a wide range of web domains, including e-commerce(e.g., Alibaba,Amazon), retrieving data(e.g., Yahoo), social networks(e.g., Twitter), location services(e.g., Waze,Foursquare), and more[1].

Research on recommender systems has gained popularity in both academia and business. By 2020, there will be close to 50 billionInternet of Thing(IoT) devices on the Internet. This will have eventually fuelled a lot novel resulting services like virtual reality (VR), augmented reality (AR), live streaming, electric vehicle (EV), and smart city (SC)[2]. The growth of services is moving in the lines of intelligence, personalisation, and integration to enhance the quality-of-experience (QoE) of users. Such services require a high quality of service (QoS), are latency-critical, and are computationally demanding. On the alternative hand, during the following five years, global mobile data traffic is expected to increase by almost 11 times[3].

By 2021, over 850 ZB of data will be produced annually, but the data centre is only 20.6 ZB, based on Cisco projections. It shows that the arrangement of data sources' physical locations is changing from data centres to a growing variety of edge devices. In this situation, the data streams have the following crucial traits: Heterogeneity, mobility, volatility, and sparsity. Despite the fact that numerous cloud computing-based recommender systems have been put forth in conventional Internet environments, they are progressively unfit to handle

these unique emerging services and massively distributed data in mobile edge networks, and they may fail to predict what users' interests and demands[4].

The recommender systems are mostly dealing with three specific issue as a result of the mentioned difficult characteristics: 1) The cold-start problem, which arises because active client's data sources are typically very sparse and even new or inactive clients lack relevant profiles; 2) The exploration and exploitation problem; and 3) the privacy and security problem, which arises because the data sources are generated by multiple IoT devices and stored on numerous edge platforms, potentially posing a security risk to user data. A privacy issue could arise if all data sources are moved to a central cloud server for management. In order to increase QoS, mobile edge computing (MEC) pushes computation/storage resources from remote cloud servers to network edge servers.

Edge servers can be installed at or close to base stations that are closer to users in order to process computing jobs and services as data sources. MEC is an addition to and complement to traditional cloud computing, rather than being mutually exclusive from it. To address the crucial needs (such as agility, heterogeneous data analysis, and privacy-policy approach) of computation jobs, a variety of machine learning-based intelligent services have been implemented at edge servers. According to the availability of diverse user sources, IoT-based devices and applications can comprehend user profiles thoroughly and to cater to individual user needs. Users' personal data may be protected by data sources processed locally rather than remotely on cloud servers. MEC is therefore anticipated to address the problems with the present recommender systems.In reality, edge computing and recommender systems are being merged progressively[5].

In existing work suggestedhybrid convolutional neural network with long short-term memory (HCNN-LSTM) technique based Edge System Recommendation (ESR) Algorithm for Cloud Service Provider (CSP). However, existing work did not achieve the sufficient effective entity identification accuracy and the QoS requirements in the edge are not satisfied. To overcome the above-mentioned problem in this work introduced a hybrid convolutional neural network with long short-term memory (HCNN-LSTM) technique-based Edge System Recommendation (ESR) Algorithm for Cloud Service Provider (CSP). With this algorithm CSP can easily select Edge Systems. Securing cloud server data uses Elliptical Curve Cryptography (ECC) method to improve the accuracy and satisfy QoS requirements in the edge, an ELitism Dragonfly Optimization (ELDFO) has been introduced for edge. It conducts

accurate entity recognition based on the HCNN-LSTM while taking into account entity feature data.

2.Literature review

Yin, et al [2020][6] proposeda novel matrix factorization (MF) model that incorporates a convolutional neural network (CNN) deep learning model. Joint CNN-MF (JCM) is the name of the envisioned modality. JCM uses the deep latent qualities of neighbours that have been learned to infer the characteristics of a user or a service. The suggested model includes an innovative similarity computing method to increase the accuracy of neighbour's selection. In order to infer the characteristics of the target user or target service, CNN analyses the neighbours' characteristics, creates a feature matrix, and so on. research on a dataset from a real-world service with a variety of data densities to simulate complex invocation scenarios in an edge computing context. The outcomes show that the technique can consistently produce higher QoS prediction outcomes when compared to other techniques.

Xu,et al [2020][7] proposed a cloud-edge computing multi-objective computation offloading approach (MOC) for IoV. The first method is a vehicle-to-vehicle communication-based route obtaining method. Which ECD to submit the computing jobs to is chosen to assure the validity of the IoV data. The compute offloading between ECDs and the cloud is taken into consideration in the scenario where all ECDs are overloaded. In order to achieve the multi-objective optimisation of lowering the load balancing rate, lowering the energy consumption in ECDs, and reducing the processing time for computing jobs, non-dominated sorting genetic algorithm III is also employed. using a straightforward additive weighting system and various criteria decision-making to assess the suggested method's results. The usefulness and efficiency of the suggested method are verified by experiments.

Zhou et al[2019] [8]proposedanadvanced recommendation algorithm called Inverse_CF_Rec. In more concrete terms, for a target user, first look for their adversaries (referred to collectively as "enemies" hereafter); then, indirectly infer the target user's potential friends in accordance with the Social Balance Theory; and finally, suggest the best services to the target user centred on the derived potential friends of the target user. To verify the efficacy and efficiency of concept, experiments are carried out on the real-world dataset WS-DREAM. The outcomes demonstrate the benefits of suggested system in terms of accuracy and effectiveness of recommendations.

Wu et al[2020][9]proposed A collective recommendations system for network content resource discovery employing the data graph and LSTM in edge computing, which can successfully address the issues of data overload and resource trek. The implementation of big data in the packaging business has undergone a thorough system test. The experimental findings demonstrate that the suggested method uses edge computing's knowledge graph and LSTM to recommend network document resources more precisely and further enhance suggestion quality. Consequently, it can more successfully satisfy the user's need for personalised resources.

Yu, et al [2021][10]proposed a strategy for recommending services according to collaborative filtering (CF) and location, considering the characteristics of services at the edge, mobility, and user requests at various times. the multidimensional weighting method to determine the service features of each dimension in various time slices. To address the issue of sparse data, combine the concept of inverse CF recognition with regular CF and forecast the lost quality of service (QoS). At last, a recommendation algorithm using user location and projected QoS is suggested to suggest suitable services to users. The findings demonstrate the recommendation accuracy, the multidimensional inverse similarity recommendation algorithm using time-aware collaborative filtering (MDITCF) surpasses Inverse CF Rec.

Jakhon et al [2022][11]proposed a strategy of agent grouping and edge computing device recommendations to speed up data handling and produce the best outcomes for agent selection in RTS games. The suggested technique, which made use of K-means, influence mapping, and Bayesian probability, was tested in a game setting, which makes it simple to assess how well it handles data. The analysis of the differences among the recommended and random modes reveals that the suggested method has the potential to boost winning percentages by 47%.

Yin, et al [2019][12]proposed a model ensemble that blends neighbourhood-based and model-based CF. This method consists of two phases: learning global features and learning local features. First, an enhanced auto-encoder that deals with sparse inputs by pre-calculating an approximation of the missing QoS values and can get accurate hidden features by preserving the complicated structure of the QoS records is used to address the cold-start issue. In the second stage, a brand-new computing technique using the Euclidean distance is suggested in order to further boost prediction accuracy and address the overestimation issue.In the computation of similarity, propose the two new notions of common activation

factor and activation frequency factor. two separate models and one hybrid model constitute the trio of prediction models. The hybrid model will make use of all neighbours, whereas the two separate models will make use of user- and service-similar neighbours. The tests carried out on a real-world dataset demonstrate that proposed models can produce better prediction outcomes and are not susceptible to parameter changes.

3.Proposed methodology

EoT architecture is introduced for several edge locations and numerous devices. Edge System Recommendation (ESR) is essential concept between the Cloud Server (CS) and Edge System (ES). One user may utilise Edge System differently than another. The Requirements varies based on the cost, speed, reliability, scalability and the capacity of Random Access Memory (RAM) of Edge Server. To analyse the genuiness of the Edge Server, the following process is executed repeatedly. Edge QoS Check by ELitism Dragonfly Optimization (ELDFO). The figure 1. given below describes the workflow of proposed method.

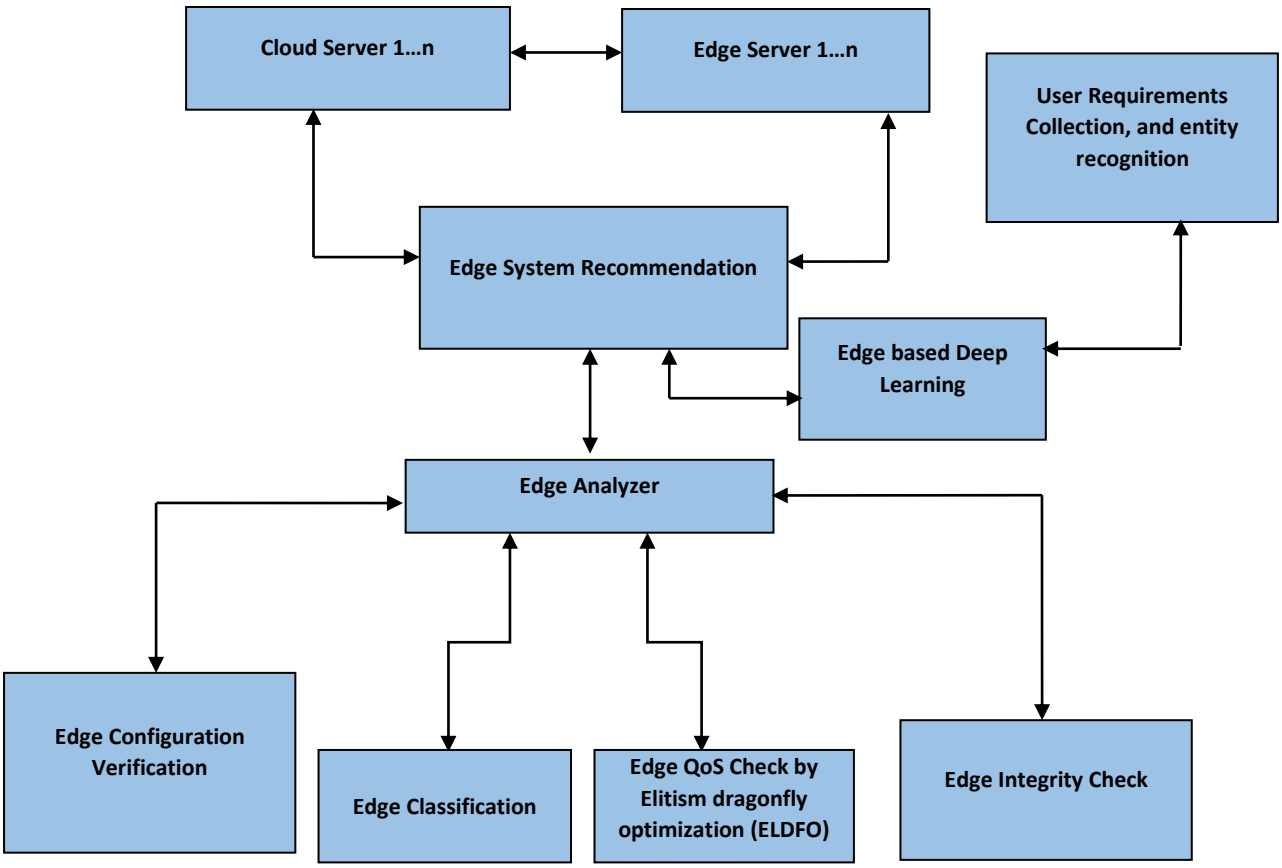

FIGURE 1. PROPOSED EDGE RECOMMENDATION SYSTEM

3.1. Edge Systems

Data communication between user and Cloud System is done through Edge System. Multiple edge systems are engaged for data transaction and processing. Once the edge system is selected alternates are allocated and scheduled for fault tolerance. For edge server is selected by Edge System Recommendation (ESR) algorithm.

3.2. Cloud Server

Individual Data Centre owners as well as individual server owners will apply for Edge System requirements. Either partial or full Servers will be provided by Edge owners. Configuration for shared or dedicated servers will be provided to Cloud Service Providers. Depending upon the system configuration and bandwidth provided price range will be selected. Once the cloud and edge providers are mutually agreed then they will be connected.

3.3. Edge Recommendation System

Edge recommendation system includes of the following steps,

- Edge configuration verification.
- Data integrity check.
- Edge QoS check
- Edge learning
- Feedback collection
- Analyse CSP through feedback

Edge configuration verification: Edge configuration will be verified by getting system information from the edge system in stealth mode. Extracted configuration will be compared with the provided configuration by edge providers. If it is shared server the total configuration is checked for capability. The edge will receive training data and processed for performance check.

Data integrity check: Security program will be installed in Edge System for continuous monitoring. Data given as input to edge system will be tracked for its integrity. The total size of data as input and output are monitored.

Edge QoS check: Quality of service for each edge system is checked. The service provided by Edge Systems is tracked and task completion records are maintained. If any defects or error occurred then it will affect its QoS rating. Fault tolerance rate are monitored based on unfinished task and successful migration.

Edge learning: Learning occurs utilizing the Long Short-Term Memory (LSTM) method. Learning will be done in each check points. Depending upon this data edge system will be classified and rated. Long term learning will be combined with short term memory. Both data will be sorted and merged. With this merged data recommendation will be provided for cloud service providers. Through this learning edge system can improvise itself to attain maximum profit.

Feedback collection: Feedback will be collected from the user, CSP and Edge System. User will provide overall service performance received. Edge will provide feedback about the cloud server provider. Cloud service provider will provide feedback about the Edge Systems.

Feedback analyser: Depending upon feedback from Cloud Server and user Edge Systems is recommended. Depending upon the feedback from Edge System, Cloud Service Provider will be classified and recommended. The feedback will be based on data transaction from CSP and to CSP. Data from CSP will be checked for security issues and other compatibility issues.

3.4. Data Integrity by Elliptical Curve Cryptography (ECC) algorithm

Securing cloud server data uses Elliptical Curve Cryptography (ECC) algorithm which is open-source cryptography method involved centredalgebraic structure with elliptic curves. Key based security used in ECC algorithm with finite fields which secures efficiently. ECC produces keys using the characteristics of an equation for an elliptic curve rather than the more conventional approach of creation as the product a sum of large prime numbers.

3.5. Edge based QoS optimisation by ELitism Dragonfly Optimization (ELDFO) algorithm

ELitism Dragonfly Optimization (ELDFO) has been introduced for optimization of QoS constraints in edge system depending on dragonfly swarming characteristics, both static and dynamic [13,14]. A static swarm's flying actions of sub-swarms over various places is used to mimic the exploration phase. During the exploitation phase, dragonflies fly in larger swarms and in a single direction.

Let us take a cluster of n nodes with p pods. A collection of functions F with various processing capacity needs are requested by cloud users. The cluster's local hosting on an edge server eliminates the necessity to consider internal and external container communication requirements in terms of bandwidth[15].

As logically demonstrated in Equation (5.1), a binary variable $x_{f,i,t} \in \{0,1\}\ i = 1,2,\dots,p \mid p \in P$ is added as a parameter for the instantiation of function f as pod i at time t. Equation (2) additionally introduces a further binary variable y_{ij}, linking pod i to node j at time t. φ^t is the total number of active (running but unscheduled) pods at time t.

$$\varphi^t = \sum_{f\in F}\sum_{i\in P}\sum_{t\in T} x_{f,i,t} \tag{1}$$

$$\sum_{i=1}^{p}\sum_{j=1}^{n} y_{ij} = \varphi^t \tag{2}$$

The subsequent capacity restrictions on the scheduling decisions,

$$\sum_{i=1}^{p} v_{cpu}^{f,t}.\, y_{ij} \le r_{cpu}^{j,t} \tag{3}$$

$$\sum_{i=1}^{p} v_{mem}^{f,t}.\, y_{ij} \le r_{mem}^{j,t} \tag{4}$$

Equations (3) and (4) constitute that the total node capacity is not exceeded by the CPU and memory requests made by the service pods for allotment. The values $r_{cpu}^{j,t}$ and $r_{mem}^{j,t}$ indicate, accordingly, the total CPU and memory capacity of node j at time t.Equations (5-6) can be utilised to calculate the number of unused resources for a certain node j at time t, where $\hat{r}_{cpu}^{j,t}$ and $\hat{r}_{mem}^{j,t}$ are the node's accessible CPU and memory capacity, accordingly

$$\hat{r}_{cpu}^{j,t} = r_{cpu}^{j,t} - \left[\sum_{i=1}^{p}\left(y_{ij}.\, v_{cpu}^{f,t}\right) + \delta_{cpu}\right] \tag{5}$$

$$\hat{r}_{mem}^{j,t} = r_{mem}^{j,t} - \left[\sum_{i=1}^{p}\left(y_{ij}.\, v_{mem}^{f,t}\right) + \delta_{mem}\right] \tag{6}$$

The weighted relationship among CPU and memory usage, as shown by Equation (7), with weight φ_1 is defined as the utilisation U_j^t of node j at time t. Equation (8) shows the memory relationship u^{mem} and the CPU utilisation u^{cpu} as a ratio of the accessible node capacity to the total node capacity.

$$U_j^t = \varphi_1 \times u^{cpu} + (1-\varphi_1) \times u^{mem} \tag{7}$$

$$u^{cpu} = \frac{\hat{r}_{cpu}^{j,t}}{r_{cpu}^{j,t}}; u^{cpu} = \frac{\hat{r}_{mem}^{j,t}}{r_{mem}^{j,t}} \quad (8)$$

Cost is a crucial factor to consider when assessing the viability of a suggested strategy. The next two Equations (9) and (10) express the node's initial deployment is given a fixed value called e_n .

$$C_{node} = \frac{1}{1-u^{cpu}} + \frac{1}{1-u^{mem}} \quad (9)$$

$$C_{total} = e_n \sum_{i=1}^{p} \sum_{j=1}^{n} y_{ij}.C_{node} \quad (10)$$

Three important behavior of swarms for optimization of cost function in edge system are described and quantitatively illustrated as follows [16],

Separation is the process of separating individuals from other QoS restrictions in the vicinity in order to prevent static collision[16],

$$S_i = -\sum_{j=1}^{N} X - X_j \quad (11)$$

where X and $X_j$are the place of the current edge system and place of j^{th} neighbouring edge. N is the number of neighbouring edge depending on the QoS constraints. Alignment is the neighbourhood velocity matching and is represented as follows [16],

$$A_i = \frac{\sum_{j=1}^{N} V_j}{N} \quad (12)$$

where $V_j$is the velocity of j^{th} neighbouring edge system and their QoS parameter for cloud user. Cohesion is the process by which edge systems and their QoS parameters are drawn into the neighborhood's population core [15],

$$C_i = \frac{\sum_{j=1}^{N} X_j}{N} - X \quad (13)$$

where X and $X_j$are the position of the current edge system and position of j^{th}neighbouring edge system. N is the number of neighbouringedge depending on the QoS constraints. Below is a formula that measures attraction to a food source,

$$F_i = X^+ - X \quad (14)$$

Following is a calculation for enemy distraction,

$$E_i = X^- + X \qquad (15)$$

In order to update the position of manufactured dragonflies in a search space and replicate their motions, two vectorsstep (ΔX) and position (X) are taken into account. The following equation(16) is used to update the step vector,

$$\Delta X_{t+1} = (sS_i + aA_i + cC_i + fF_i + eE_i) + w\Delta X_t \qquad (16)$$

where *a*, *s*,*f*,*c*, and *e* are the weight vectors for alignment,separation, food,cohesion, and enemy correspondingly. It has been used to optimal selection of QoS constraints depending on the cloud user for edge server. *w* is the inertial weight and *t* means iteration counter. The position vectors of edge system and cloud user are defined as follows ,

$$X_{t+1} = X_t + \Delta X_{t+1} \qquad (17)$$

By establishing a levy flying using a random walk, the dragonflies' stochastic nature and exploratory behaviour can be enhanced. Continuous

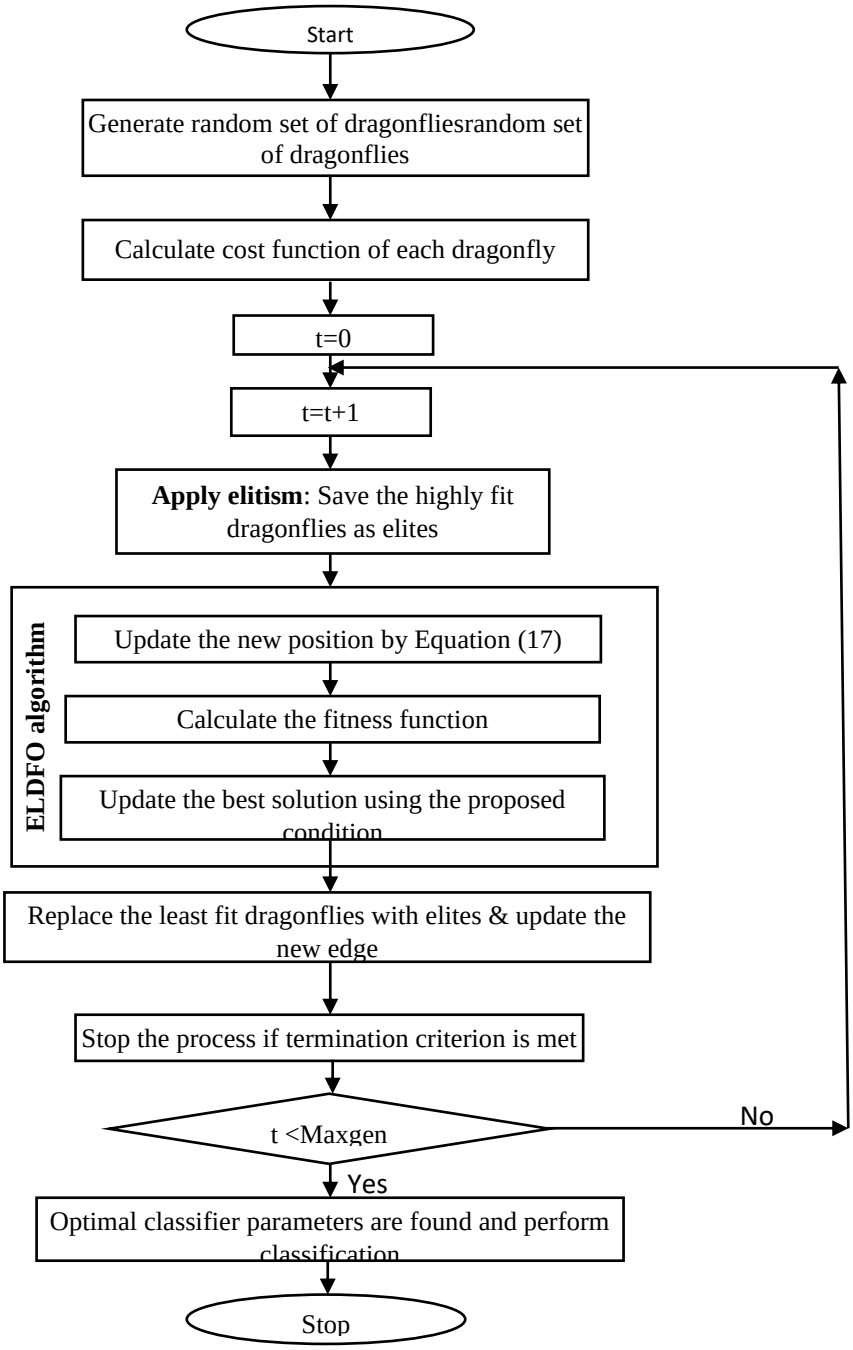

FIGURE 2. FLOWCHART OF THE PROPOSED ELDFO ALGORITHM

A transfer function is used to transform DFO to binary DFO without changing its framework. Using step values as inputs, the v-shaped transfer produces a return value between 0 and 1. It is used to determine the likelihood of a position change for all artificial dragonflies [17,18] ,

$$T(\Delta X) = \left| \frac{\Delta X}{\sqrt{1 + \Delta X^2}} \right| \qquad (18)$$

updating the search agents' positions depending on edge system in binary search spaces is then employed,

$$X_{t+1} = \begin{cases} \sim X_t \; rand < T(\Delta X_{t+1}) \\ X_t \; rand \geq T(\Delta X_{t+1}) \end{cases} \quad (19)$$

The position and velocity (step) for every dragon fly were adjusted after the adversary and food source were modified.

Figure 2 depicts the suggested ELDFO algorithm's flow chart, which demonstrates exactly the algorithm is put into practice. In two circumstances, a best solution and position for an ideal QoS constraint of the edge system and cloud user was updated.If the edge system's QoS performance was better than that of the prior edge system, the present QoS and position were modified. The second circumstance involved an update to the QoS constraint. Later, elitism was used; the two dragonflies with the worst fitness or least fit were substituted with extremely fit dragonflies and their fitness. When a predetermined ending criterion, such as the maximum number of iterations or an ideal QoS limit, is met, the ELDFO algorithm simulation will come to an end. 100 was the predetermined maximum number of iterations.

3.6. HCNN-LSTM based edge learning

CNN[19,20] includes training alternatives that can directly impact feedback connections for extracting entity state information. Long-Short Term Memory (LSTM) algorithm is an Artificial Neural Network (ANN) based deep learning which requires feedback connections with Recurrent Neural Network (RNN). It processes entire sequence of data by predicting data where tasks are segmented and prioritized. Edge System, Cloud Server, Tenant System or Container Bot Recommender System detects in accordance with its data privacy level. Results of CNN[21] and LSTM are combined to produce final feedback connections.

3.6.1.Entity Recognition Method

By storing entity state data on the edge server, it is possible to effectively handle the issue of low accuracy brought on due to the prolonged connection distance. In order to properly categorize entities on the edge side, this study suggests an appropriate entity recognition approach. The edge server runs entity recognition. Edge server users' search history is also kept. Lastly, classify hot and cool entities using multidimensional entity features. The edge server holds hot entity state data with higher access frequency and large temporal fluctuation after categorization.

After extracting entity state data and user access characteristics from the Hybrid Convolutional Neural Network with Long Short-Term Memory (Hybrid CNN-LSTM)

network, K-means clustering is introduced to group similar entity categories [14]. The primary functions of the algorithm, which areentity recognition classificationandentity feature extraction.

Entity information Extraction using Hybrid CNN-LSTM: This research work, CNN has been introduced for entity and their data. CNN includes of two main stages like entity and information about entity where multiple convolution layers are used for entity information extraction and subsequently activation functionsand max-pooling are used for entity recognition.

LSTM: LSTM includes forget,input,output gates as well as a cell activation component. It is representation of a typical LSTM cell. It can employ certain multipliers to control cell activations after receiving activation from various sources. In the LSTM-RNN architecture, LSTM cells take position of the recurrent hidden layer.

Hybrid CNN-LSTM: Where two layers of CNN are used, a suggested HCNN-LSTM technique retrieves entity data. The learnable filters used in the convolution layers allow for the identification of specific properties in the raw data; as a result, a series of activation maps are produced. LSTM further acquire temporal features of entity features. The two fully connected layers are used to conduct the categorization of the entity features based on cloud and edge server.

Entity Recognition and Classification: The set of entity features was created after training with the HCNN-LSTM. In this study, cluster analysis is further performed on the set of characteristics using the K-means clustering technique in order to efficiently differentiate the category of the entity and, storage space required for the entity state information by edge, Tenant, and cloud server. Finally, the algorithm converges. It is the duty of the edge server to locally store information on the hot entity state, while the cloud server is responsible for uploading the information about the cold entity state.

4.RESULTS AND DISCUSSION

In this analysis Bots in Edge System, Cloud Server and Tenant System which is plotted against the number of Bots attacked and handled by Container storage system. The produced model is installed in the Java tool on a computer running Windows 10 with an Intel i3 core processor and 2 GB of RAM.It is tested. 25 edge systems are tested with 3 servers. The development tools are JAVA, and the simulation environment is 64-bit Windows 10 operating system with an Intel (R) Core (TM) i5-3337U processor, 64 GB of RAM, and a

CPU running at 1.80GHz. With KubeEdge version 1.4, a cluster that consists of numerous nodes is created. The worker nodes and 32 logical cores server are setup in addition to the master node's 8 CPU cores and 8 GB of RAM.Performance metrics used in this established model is True Positive Rate (TPR), True Negative Rate (TNR), and testing accuracy. Analyse the three areas of utilisation, service time, and QoS violations in the working system. QoS results are evaluated by existing methods like STaSA (Service–Time-Aware Scheduling Algorithm for multi-node KubeEdge Cluster), Ant Colony Optimization (ACO), and proposed ELDFO algorithm.As a result, the performance of three algorithms has been evaluated; the results are displayed in Figures 5.5–5.8. Figure 5.5 examines CPU utilisation and demonstrates that ELDFO exceeds the other two techniques. As evidenced by the rise in the number of pod instances, all algorithms' utilisation values are somewhat over average, while ELDFO performs better in locating the closest ideal QoS solution mapping for pods to nodes. This means the required level of fitness has been attained. ACO performs poorly while calculating slower jobs, while STaSA has good results in comparison. Also, because all the workers work simultaneously, STaSA can keep the CPU busy owing to a quick upload, increasing resource utilisation. Figure 3 provides an excellent illustration of how CPU utilisation varies amongst nodes in a cluster system. To marginally surpass ACO and STaSA, ELDFO can achieve a greater utilisation with rising pod requests. ELDFO can minimise BoT and provide better outcomes at a point when the pods are around 30 and 35. The proposed ELDFO gives higher CPU utilization of 92.17%, other methods such as ACO, and STaSA gives lesser CPU of 84.51%, and 87.15% for 10 pods.

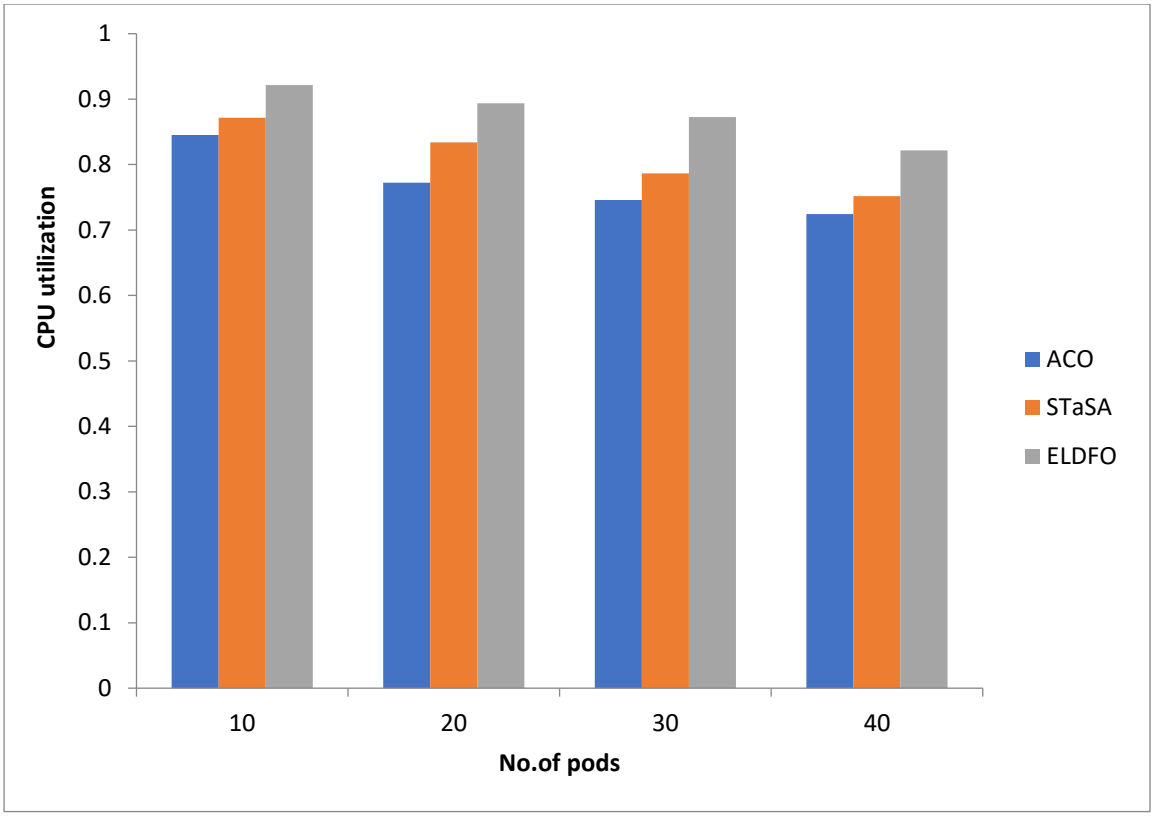

FIGURE 3. CPU UTILIZATION OF QoS OPTIMIZATION METHODS

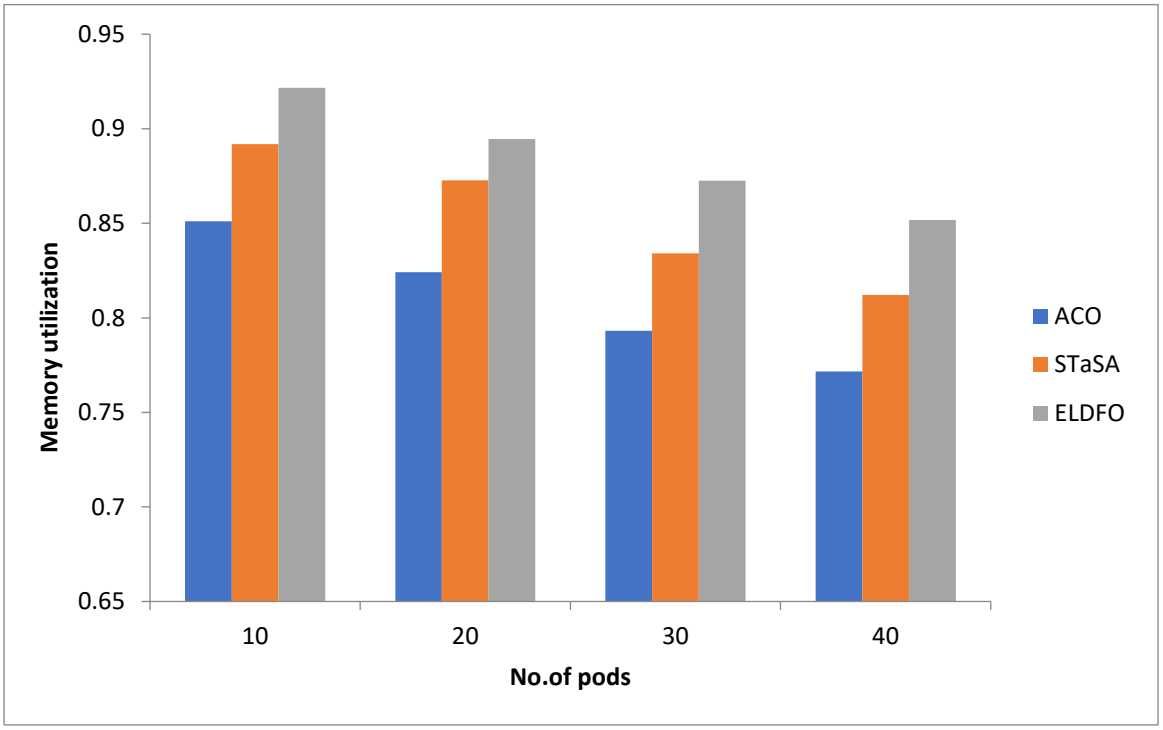

FIGURE 4. MEMORY UTILIZATION OF QoS OPTIMIZATION METHODS

Figure 4 provides an excellent illustration of the way various nodes in a cluster system use memory. ELDFO, yet, is able to achieve a greater utilisation and marginally beats ACO

and STaSA with rising pod requests. The proposed ELDFO gives higher memory utilization of 92.00%, other methods such as ACO, and STaSA gives lesser memory utilization of 84.00%, and 87.00% for 10 pods.

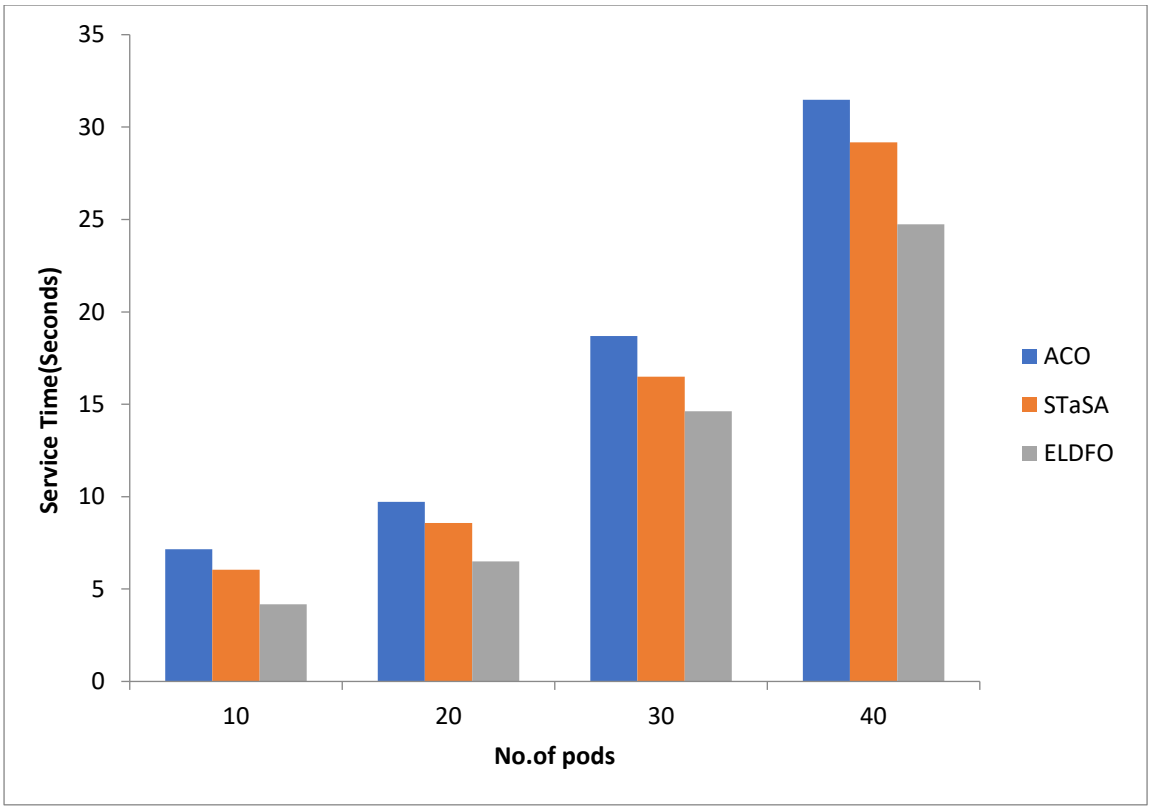

FIGURE 5. SERVICE TIME OF QoS OPTIMIZATION METHODS

Service time comparison of QoS learning methods for edge system is illustrated in the figure 5. By varying the number of pots, proposed system has reduced service time than the other methods, since proposed system quickly optimizes the QoS parameters than the other methods. The proposed ELDFO algorithm has lesser usage of the service time of 24.74 seconds, other methods such as ACO, and STaSA needs more service time of 31.47 seconds, and 29.18 seconds for 40 pods.

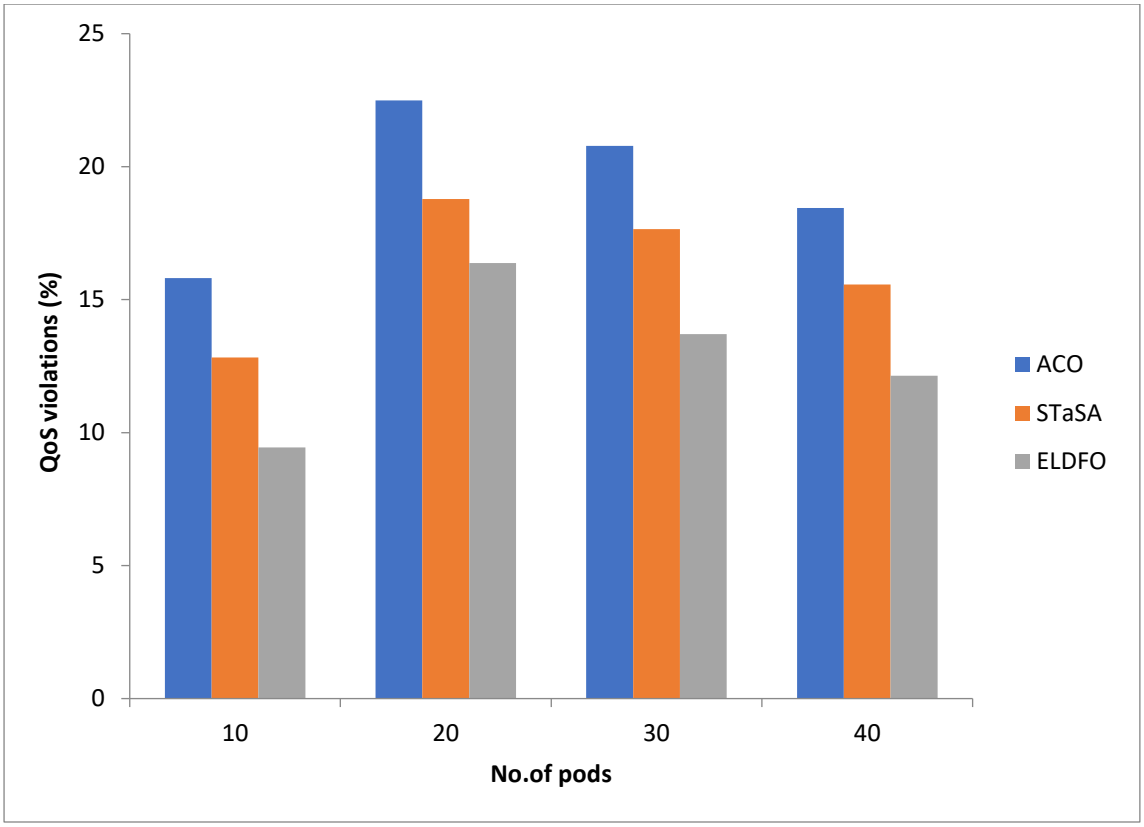

FIGURE 6. QoS VIOLATION OF QoS OPTIMIZATION METHODS

QoS violations results of learning methods for edge system are illustrated in the figure 6. By varying the number of pots, proposed system has lesser violations than the other methods, since proposed system quickly optimizes the QoS parameters by ELDFO than the other methods. The proposed algorithm has lesser QoS violation of 12.14%, other methods such as ACO, and STaSA has more violation rate of 18.45%, and 15.57% for 40 pods.

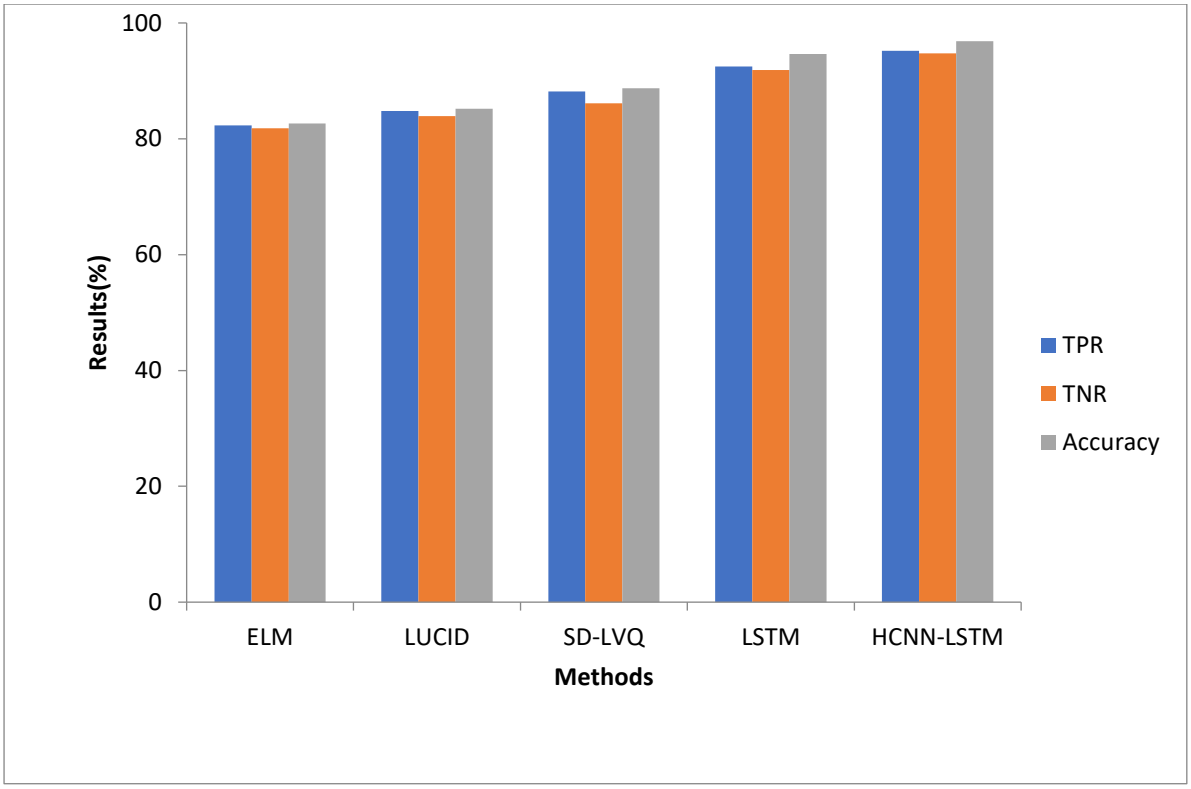

FIGURE 7. PERFORMANCE ANALYSIS OF ATTACK DETECTION METHODS

The figure 7 describes the various numbers of methods for attack detection methods after edge QoS learning. According to the findings, the suggested method has a superior accuracy of 96.85%, whereas other methods such as ELM, LUCID, SD-LVQ, and LSTM gives lesser detection ratio of 82.63%, 85.18%, 88.75%, and 94.62%.

5.CONCLUSION AND FUTURE WORK

In this work, Optimized Edge System Recommendation (ERS) algorithm is introduced for edge recommendation in CSP. ELitism Dragonfly Optimization (ELDFO) has been introduced for optimization of QoS constraints in edge system depending on dragonfly swarming characteristics, both static and dynamic. These behaviours have been performed based on the exploration phase, and exploitation phase. Entity recognition algorithm make uses a Hybrid Convolutional Neural Network- Long Short-Term Memory (HCNN-LSTM) embedded in an edge server to distinguish hot and cold entities for the limited communication capabilities of data. It also provides rating for edge system which will be effective feedback for edge system.From the experimental results it is concluded that the proposed model produces the better performance in terms of attack detection, service time and CPU utilization. Proposed model can be enhanced using ensemble learning to improve the performance and this could be focused in future.

References

1. Łaskawiec, S., Choraś, M., Kozik, R. and Varadarajan, V., 2021. Intelligent operator: Machine learning based decision support and explainer for human operators and service providers in the fog, cloud and edge networks. Journal of Information Security and Applications, 56, p.102685.
2. Queiroz, J., Leitão, P. and Oliveira, E., 2022. A Fuzzy Logic Recommendation System to Support the Design of Cloud-Edge Data Analysis in Cyber-Physical Systems. IEEE Open Journal of the Industrial Electronics Society, 3, pp.174-187.
3. Xu, Y., Li, J., Lu, Z., Wu, J., Hung, P.C. and Alelaiwi, A., 2020. ARVMEC: Adaptive recommendation of virtual machines for IoT in edge–cloud environment. Journal of Parallel and Distributed Computing, 141, pp.23-34.
4. Sun, C., Li, X., Wen, J., Wang, X., Han, Z. and Leung, V.C., 2023. Federated Deep Reinforcement Learning for Recommendation-enabled Edge Caching in Mobile Edge-Cloud Computing Networks. IEEE Journal on Selected Areas in Communications.
5. Yang, J., Wang, H., Lv, Z., Wei, W., Song, H., Erol-Kantarci, M., Kantarci, B. and He, S., 2017. Multimedia recommendation and transmission system based on cloud platform. Future Generation Computer Systems, 70, pp.94-103.
6. Yin, Y., Chen, L., Xu, Y., Wan, J., Zhang, H. and Mai, Z., 2020. QoS prediction for service recommendation with deep feature learning in edge computing environment. Mobile networks and applications, 25, pp.391-401.
7. Xu, X., Gu, R., Dai, F., Qi, L. and Wan, S., 2020. Multi-objective computation offloading for internet of vehicles in cloud-edge computing. Wireless Networks, 26, pp.1611-1629.
8. Zhou, Y., Tang, Z., Qi, L., Zhang, X., Dou, W. and Wan, S., 2019. Intelligent service recommendation for cold-start problems in edge computing. IEEE Access, 7, pp.46637-46645.
9. Wu, Y., Liu, Q., Chen, R., Li, C. and Peng, Z., 2020. A group recommendation system of network document resource based on knowledge graph and LSTM in edge computing. Security and Communication Networks, 2020, pp.1-11.
10. Yu, M., Fan, G., Yu, H. and Chen, L., 2021. Location-based and time-aware service recommendation in mobile edge computing. International Journal of Parallel Programming, 49, pp.715-731.

11. Jakhon, K.S., Guo, H. and Cho, K., 2022. Agent grouping recommendation method in edge computing. Journal of Ambient Intelligence and Humanized Computing, pp.1-11.
12. Yin, Y., Zhang, W., Xu, Y., Zhang, H., Mai, Z. and Yu, L., 2019. QoS prediction for mobile edge service recommendation with auto-encoder. IEEE Access, 7, pp.62312-62324.
13. Alshinwan, M., Abualigah, L., Shehab, M., Elaziz, M.A., Khasawneh, A.M., Alabool, H. and Hamad, H.A., 2021. Dragonfly algorithm: a comprehensive survey of its results, variants, and applications. Multimedia Tools and Applications, 80, pp.14979-15016.
14. Mirjalili, S., 2016. Dragonfly algorithm: a new meta-heuristic optimization technique for solving single-objective, discrete, and multi-objective problems. Neural computing and applications, 27, pp.1053-1073.
15. Singh, H., Sawle, Y., Dixit, S., Malik, H. and Márquez, F.P.G., 2023. Optimization of reactive power using dragonfly algorithm in DG integrated distribution system. Electric Power Systems Research, 220, p.109351.
16. Alshinwan, M., Abualigah, L., Shehab, M., Elaziz, M.A., Khasawneh, A.M., Alabool, H. and Hamad, H.A., 2021. Dragonfly algorithm: a comprehensive survey of its results, variants, and applications. Multimedia Tools and Applications, 80, pp.14979-15016.
17. Shilaja, C. and Arunprasath, T., 2019. Internet of medical things-load optimization of power flow based on hybrid enhanced grey wolf optimization and dragonfly algorithm. Future Generation Computer Systems, 98, pp.319-330.
18. Amroune, M., Bouktir, T. and Musirin, I., 2018. Power system voltage stability assessment using a hybrid approach combining dragonfly optimization algorithm and support vector regression. Arabian Journal for Science and Engineering, 43, pp.3023-3036.
19. Shao, Z., Pan, Y., Diao, C. and Cai, J., 2019. Cloud detection in remote sensing images based on multiscale features-convolutional neural network. IEEE Transactions on Geoscience and Remote Sensing, 57(6), pp.4062-4076.
20. Chai, D., Newsam, S., Zhang, H.K., Qiu, Y. and Huang, J., 2019. Cloud and cloud shadow detection in Landsat imagery based on deep convolutional neural networks. Remote sensing of environment, 225, pp.307-316.

21. Mateo-García, G., Gómez-Chova, L. and Camps-Valls, G., 2017, July. Convolutional neural networks for multispectral image cloud masking. In 2017 IEEE International Geoscience and Remote Sensing Symposium (IGARSS) (pp. 2255-2258). IEEE.