Journal of Theoretical and Applied Information Technology
15th March 2025. Vol.103. No.5

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

A NOVEL FUZZY RULE BASED DECISION-MAKING APPROACH FOR TASK OFFLOADING ON MULTI-TIER MOBILE-EDGE CLOUD COMPUTING SYSTEMS

R. AISHWARYA[1*] and G. MATHIVANAN[2]
[1,2]Department of Computer Science and Engineering
[1,2]Sathyabama Institute of Science and Technology
Jeppiaar Nagar, SH 49A
Chennai, Tamil Nadu 600119
[1]aishwaryarajkumar20@outlook.com
[2]mathivanan.it@sathyabama.ac.in
*Corresponding Author: R. Aishwarya

ABSTRACT

Mobile Edge Cloud Computing (MECC) is a developing distributed processing method, which delivers access to CC services at the network's edge and closer mobile operators. In a speedily varying and dynamic environment, it is very cruel to discover the optimum target server for managing unloaded tasks since we don't recognize the end user's demand further. By offloading tasks at the network's edge rather than transmitting them to a remote cloud, MECC can recognize flexibility and actual handling. In the present study, the varying desires of Internet of Things (IoT) applications at dissimilar phases are often neglected in the context of computation offloading. This study introduces fuzzy rule-based decision-making for task offloading approach (FRBDM-TOLA) technique on multi-tier MECC systems. Initially, the presented FRBDM-TOLA technique uses fuzzy clustering algorithms like Fuzzy C-Means (FCM) to categorize tasks into fuzzy subcategories based on their attributes, intensity levels, and temporal aspects. Moreover, the developed method employs the Hybrid Fuzzy-Neural Network (HFNN) approach to select the most suitable target node for the offloading of tasks depending on latency sensitivity function, server capacity, and the state of the network. The HFNN is a hybrid of the FL and the NN used for the rule generation with reference to the intensity level and traffic flow. FL can claim linguistic comments and uncertainties and neural networks can take up complex patterns from the data offered to them. With the aim to improve the performance of the HFNN method the FRBDM-TOLA technique employs Spotted Hyena Optimizer (SHO) for hyperparameter optimization. The developed models optimally employ processes of classification and clustering so as to increase classification accuracy of evaluating network traffic data.

Keywords: *Fuzzy Logic, Internet of Things, Mobile-Edge Cloud Computing, Spotted Hyena Optimizer, Task Offloading.*

1. INTRODUCTION

In the past years, the rise and progress of Internet of Things (IoT) applications used in many fields namely Virtual Reality (VR), Online Gaming, Augmented Reality (AR), and Multimedia Streaming as directed to the growth of communication technologies and the online-based distributing computed outlook [1]. IoT devices are considered to be the key platform, and they undergo small storage, battery life, and computing resources to execute delay-sensitivity IoT uses [2]. To find out the limits stated above, edge computing was uplifted to extend the cloud computing techniques to bring the resources abilities close to the end-user at the margin of the networks for attaining small delay and live admission to the network service areas. To achieve this, it needs IoT applications to be off-loaded and it has been performed through edge servers instead of attending by distant cloud servers [3]. Computation off-loading is achieved between cloud servers, IoT devices, and edge servers, somewhere it is pleased based on several Quality of Service (QoS) desires in IoT applications such as privacy, load balancing energy management, and data security [4]. Even though a few earlier studies have utilized machine learning (ML) techniques to contract with the computation off-loading in the Mobile Edge Cloud Computing (MECC) surroundings, additional efforts are required to help IoT uses in the MECC eco-system powerfully. Considering edge servers, application performance is larger or has fewer black boxes, which makes it tough in design time to describe ideal regulations [5].

Still, in more situations, application designers have little bits of knowledge of the edge framework. Alternatively, due to active alterations for using and accessing the IoT applications in time, which requires that the edge servers auto-scaling plan contracts with the work-loaded variation of IoT use for providing the preferred presentation at implementation time difficult tasks to be considered [6]. In the end, they studied the united computation off-loading and auto-scaling device for helping IoT applications in the MECC environments.

Earlier off-loading jobs to the edge or cloud, it is necessary to cautiously consider the open resources and requirements [7]. Still, edge-cloud computing are active and resource-restricted atmosphere. Hence, to make the best result for tasks off-loading based on the obtainable resources is a serious problem. Task off-loading is the method of transporting tasks or workloads from nearby devices to distant devices, like servers or cloud resources, to increase the presentation and efficiency of the partial devices. Off-loading tasks may outcome in improved delay and energy intake while edge servers can have only a small computing ability, which can directly improve computational response time. It is essential to examine this trade-off before concluding. 5G networks using great masses can also have experienced advanced communication delay [8]. Co-operative job off-loading is a method applied in edge-cloud systems to increase the presentation of circulated systems. In this distributing method, tasks are divided between devices in the networks namely cloud servers and edge devices, to enhance resource utilization and to decrease the amount of work on separate devices [9]. Several researches have been directed on the subject of computational off-loading in edge-cloud systems. Still, owing to the different necessities of end devices and the small data only accessible around wireless stations, bandwidth, and computational resources, it is thought-provoking to project an ideal off-loading approach [10]. Figure 1 portrays the structure of a multi-tier MECC system.

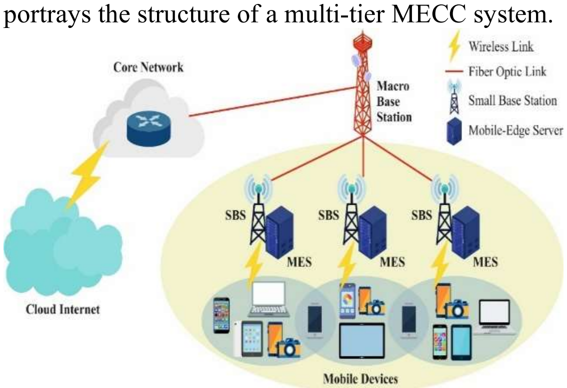

Figure 1: Architecture of Multi-Tier MECC Systems

This study introduces fuzzy rule-based decision-making for task offloading approach (FRBDM-TOLA) technique on multi-tier MECC systems. Initially, the presented FRBDM-TOLA technique uses fuzzy clustering algorithms like Fuzzy C-Means (FCM) to categorize tasks into fuzzy subcategories based on their attributes, intensity levels, and temporal aspects. Also, the developed method utilizes the Hybrid Fuzzy-Neural Network (HFNN) technique to choose the best target node for task offloading based on latency sensitivity, server capability, and the conditions of the network. The HFNN is a hybrid method, which unites fuzzy logic (FL) and neural networks for rule generation, examining intensity levels and traffic patterns. To enhance the performance of the HFNN method, the FRBDM-TOLA technique uses a spotted hyena optimizer (SHO) algorithm for hyperparameter tuning. The mathematical outcome shows that the FRBDM-TOLA model has superior performance and scalability for dissimilar processes than the present advanced models.

2. REVIEW OF LITERATURE

Tolba et al. [11] introduce a joint user association, task offloading strategy, and service caching to decrease delay and increase users' QoS from multi-tier communication and multi-tier edge computing heterogeneous networks (HetNets) method. The respected method contains multi-users by various service data sizes and tasks that communicate HetNets from one macro base station (BS) many multiple-input multiple-output (M-MIMO) and a few small BSs. In [12], computation offloading and resource allocation (RA) are assumed for multi-user multi-UAV-enabled MECC methods. Primarily, an effective computation offloading and RA method is presented for multi-user multi-UAV-enabled MECC methods. Additionally, the network uses multi-level MEC technology to produce computational abilities at the border of RAN. In [13], a novel compression safety and energy-aware task offloading method was presented. Especially, the first method presents an effective layer of compression to decrease the communication data smartly on the channel. Moreover, the security problem is addressed by a novel layer of security that depends on the AES cryptographic method offered to defend offloaded and sensitive data from various vulnerabilities. Afterward, compression of data, security, and task offloading are mutually expressed as mixed integer issues whose aim is to decrease the total energy of the method over latency restraints.

In [14], a multi-access edge computing network method containing numerous IoT and BS devices is created. Combined optimization of BS pricing, IoT device BS election, and task offloading approaches are aimed to increase BS income and IoT device services. Limin and Ke [15] presented an isolated multi-tier computing network that looks similar to real-world circumstances when related to other research. DQN is a reinforcement learning (RL) approach that leverages DNN to elevate decision-making in consecutive decision tasks. In [16], a mobile CC can be offered as a suggested solution. Later, the edge-CC model is presented and widely helps to reduce the problems. However, the existing task offloading methods allow UAVs to implement their intense tasks at the combined edge server, causing too many loads owing to enormous UAV counts increasing the delay.

In [17], a heterogeneous structure is an advanced method to improve the energy efficiency of mobile phones through analyzing parameters namely non-task and task offloading, radio access networks remote cloud servers, and local cloudlets. This work presents a task offloading structure that utilizes a new approach, the Hybrid Red Fox Flow Direction-based Ensemble SVM Forest method, parameter methods and scheduling tasks are improved in offloading CC circumstances. In [18], a fuzzy-based method with an optimum inference method has been recommended to create an appropriate offloading decision. The suggested method uses a Regression Tree (CART) and Classification method at the inference engine by decreased time complexity, conventional fuzzy-based offloading methods were demonstrated to be more effective.

3. METHODOLOGY

In this study, we have introduced a FRBDM-TOLA technique on multi-tier MECC systems. The main purposes of the FRBDM-TOLA technique comprise three distinct processes FCM-based task categorization, HFNN-based task offloading, and SHO-based hyperparameter tuning processes. Figure 2 represents the entire flow of the FRBDM-TOLA technique.

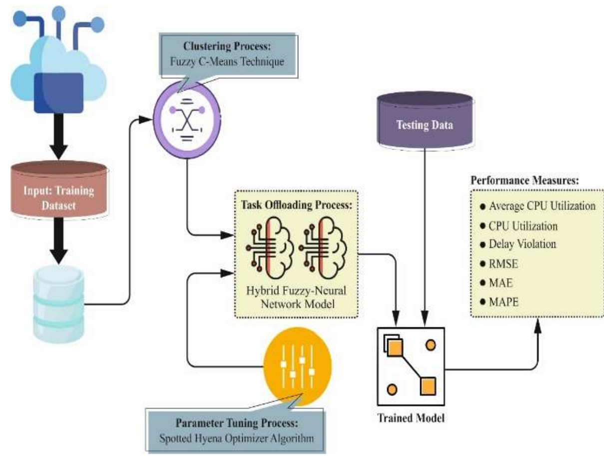

Figure 2: Overall flow of FRBDM-TOLA technique

3.1 Problem Statement

In MEC-enabled systems, the main challenging issue is task offloading due to the restricted computing resources and delay restriction [19]. Furthermore, congestion is initiated by offloading manifold tasks from numerous consumers to similar edge servers. A huge number of consumers' processing tasks on the MEC server are currently queued and waiting for execution. As an outcome, for every task, the processing delay is extensive due to overload. Hence, it is not a superior choice always to remove the task of computing to the nearby edge server. Already, the edge node l is overloaded owing to user desires. For processing, the overload tasks were furthered to the remote cloud. On the other hand, edge node 2 is slightly overloaded and many sources are obtainable to progress the computing task. This node can certainly overwhelm this issue for edge node l without distributing the tasks to the remote cloud. In the current $5G$ system, several edge servers have been used near user devices in a sort of mobile communication. So, consumers have numerous choices to offload tasks to the closest edge servers for getting services. Numerous edge servers are accessible in the MEC network, so it will be a challenging task to choose which best edge server for task offloading. Therefore, the project of an effective task-offloading device was significant since QoS differs in the task-offloading choices. The major challenges met when offloading tasks in the MEC network are given below:

1. Must remote or edge servers be utilized to offload the tasks of computing?
2. What is the ideal edge server for offloading the task?

To noticeably recognize the offloading issue. This system contains $M = \{1,2,3, \dots, M\}$ small base station (SBS), and one server of MEC was used in every SBS. $T = \{1,2,3,\ldots,T\}$ independent tasks and $N = \{1,2,3,\ldots,N\}$ user devices from every user are accessible. We signify the computing ability as $\mathfrak{r}^{mec}$, and the server collects its mobile capacity from users $N, \varphi_1, \varphi_2, \ldots, \varphi_n$. Depending upon the user device's ability, few tasks have been implemented near the device, and the excess tasks were offloaded to a MEC local server. When $(\Sigma\varphi > \mathfrak{r}^{mec})$, then it is cruel to perform additional tasks on this server. So, task 2 fails owing to the extreme workload. To discover the nearby SBS and remote cloud, we detected the following:

1. To overwhelm the MEC local server overload issue and use the nearby MEC server with the remote cloud, we insert a layer of orchestrator organization for effective task offloading amongst MEC servers in the cloud.

2. To choose whether task offloading is highly effective by a local server of MEC, a remote cloud, or a neighboring server depends upon the sensitivity of delay, size of the task, and network condition.

3. The rate of effectively implemented tasks will increase, and the task end time is much decreased by task offloading together between the remote cloud and MEC server.

3.2 FCM-based Task Categorization

Initially, the presented FRBDM-TOLA technique uses fuzzy clustering algorithms like FCM to categorize tasks into fuzzy subcategories based on their attributes, intensity levels, and temporal aspects. The generally utilized fuzzy clustering algorithm is the FCM-based algorithm [20]. It is created on reduction of the subsequent objective function, with esteem to ∪, a fuzzy c-partition of the dataset, to V, a K set prototypes:

$$J_m(U,V) = \sum_{j=1}^{n}\sum_{i=1}^{c} u_{ij}^{m} \|X_i - V_i\|^2, 1 < m < \infty \quad (1)$$

Here, m denotes a real value bigger than 1; Uij represents the degree of membership Xj in the j cluster, Xj represents the jth of d-dimension evaluated input data, Vi refers to the center of the group, and $||*||$ refers to any norm stated the resemblance among any dignified data and center. A fuzzy partition is implemented over an iterative optimizer of Eq. (1) with the upgrade of membership Uij and the cluster center Vi by:

$$U_{ij} = \frac{1}{\sum_{k=1}^{c}\left(\frac{a_{ij}}{d_{ij}}\right)^{\frac{2}{m-1}}} \quad (2)$$

$$v_i = \frac{\sum_{j=1}^{n} U_{ij}^{m} x_j}{\sum_{j=1}^{n} U_{ij}^{m}} \quad (3)$$

In this iteration, the criteria will end if maxij $|Uij - \widehat{U}ij| < \varepsilon$, while ε denotes a stop criterion among 0 and 1, similarly the highest number of iteration cycles can be employed as a stop criterion.

3.3 HFNN-based Optimal Target Node Selection for Task Offloading

Next, the developed method utilizes the HFNN technique to choose the best target node for task offloading. Fuzzy rule-based methods are built on FS theory, whose main objective is to utilize fuzzy IF-THEN rules to signify the human expert's knowledge [21]. Future, other options were proposed to make the fuzzy rule-based method without the involvement of human professionals. These techniques are mechanically generated using learning models. Many proposals are accessible to accept this learning task like space partition-based models, gradient descendent learning models, heuristic procedures, and neural-fuzzy methods among others. Along with a range of learning models to be used, the fuzzy reasoning constitution must be chosen. Depending on the kinds of fuzzy IF-THEN rules and fuzzy reasoning, many neuro-fuzzy inference models are divided into 3 kinds such as Takagi-Sugeno, Mamdani, and Tsukamoto. In this work, we utilize the Mamdani method, whose foremost benefits are interpretability, linguistic variables, and identification capacity. In the Mamdani method, fuzzy IF-THEN rules contain below given method:

$$If\,(X_1\ is\ A_1\ and \cdots and\ X_n\ is\ A_n)\ then\ Y\ is\ B, \quad (4)$$

Here, X_i and Y denote the input and output of linguistic variables, respectively; A_i and B represent the linguistic values.

The classical structure of the Mamdani method has 4 components knowledge base, fuzzification, defuzzifier, and inference engine. The fuzzification module converts the input into linguistic value. The knowledge base contains a dataset with the definition of FS and a rule base covering the fuzzy IF-THEN rules. The knowledge-based system accepts the learning procedure by utilizing the input data and fuzzy rule. Lastly, the defuzzifier decrypts the values of linguistics into an output.

Fuzzy neural networks unite artificial neural networks (ANN) with fuzzy rule-based methods. It is set upon the formation of a neural network and the learning model is employed to adjust the parameters in the fuzzy method. Between the dissimilar models accessible to unite neural networks and fuzzy systems, in this study, we will utilize a hybrid neural fuzzy inference system (HyFIS). HyFIS is a multi-layer neural networks-based fuzzy method with 5 layers. The node function and semantic meaning in this network are described. In layer1, the nodes are input, which signifies an input linguistic variable. In the layer, every node is only linked to layer2 nodes that signify the linguistic value of an equivalent linguistic variable. In layer2, the nodes perform as membership functions to signify particular linguistic variables. Then, the input values are served to layer2 which computes the membership degree. This was executed utilizing the Gaussian membership function with dual parameters such as width or variance (σ^2) and center or mean (c).

$$\mu A(x) = Gaussian(x; c, \sigma) = e^{\frac{(x-c)^2}{\sigma^2}} \tag{5}$$

At first, in this layer, a connection weight is the membership and unity functions, which are spread out similarly done the weight space. In this node, the output function is the degree to an assumed membership function:

$$y_i^{II} = e^{\frac{(x-c)^2}{\sigma^2}}, \tag{6}$$

Whereas, σ^2 and c denote the parameters. As these parameters modify, the bell-shaped function differs, therefore displaying numerous methods of membership function on the linguistic label. In layer3, every node signifies a probable IF-portion of a fuzzy rule. The weights have been fixed to unity. In this layer, the nodes achieve the AND process. So, every node custom a fuzzy rule base. The mathematical formulation is mentioned below:

$$y_j^m = \min_{i \in I_k} (y_i^{II}), \tag{7}$$

Here, I_j denotes the set of indices in layer2, which are linked to node j in layer3, and y_i^{II} refers to the output of node i.

In layer 4, a node signifies a probable THEN-part of a fuzzy rule and every node executes the fuzzy OR process for incorporating the area rules foremost to the similar output linguistic variable. The node activation signifies the grade to which this function of membership is reinforced by every fuzzy rule collectively. The early connection weights among layer3 and layer4 have been chosen at random in the range of $[-1, +1]$. The functions of this layer are stated below:

$$y_{\mathrm{k}}^{\mathrm{IV}} = \max_{j \in I_k} (y_j^{III} w_{kj}^2), \tag{8}$$

whereas I_k denotes the set of node indices in layer3, which are related to the node k in layer4. Every rule is initiated to a definite degree signified by the sharpened weight value. Layer5 signifies the output variable. These links and nodes enclosed to them perform as defuzzifier. A node calculates a hard output signal utilizing the Centre of Gravity model:

$$y_l^V = \frac{\sum_{K \in I_l} y_k^{IV} \sigma_{lk} c_{lk}}{\sum_{k \in I_k} y_k^{IV} \sigma_k}, \tag{9}$$

Here, I_t denotes the set of indices in layer4 are linked to the node l in layer5; c_{lk} and s_{lk} represent the centroid and width, respectively; k signifies the output linguistic value. The weights from the nodes in layer5 to layer4 are unity. Hence, learnable weights are $w_{kj}s$ among layers3 and layer4.

The learning method in HyFIS contains dual stages. The initial stage is the structure learning utilizing the knowledge acquisition unit. Parameter learning is the 2nd stage for tuning the membership function to attain a chosen level of efficiency. Simplicity is the main benefit of adapting the fuzzy rule and novel data turn into accessible.

In the stage of rule-finding, a set of fuzzy rules from the chosen input or output sets has been produced. Then these fuzzy rubrics are employed for defining the formation of the neuro-fuzzy method in the HyFIS. Therefore, the initial stage contains separating the input and output space into fuzzy areas. Next, they are produced from assumed data sets. Lastly, a grade is given to every regulation.

In the stage of parameter learning, once the fuzzy rules have been recognized, the entire structure of the network is definite. Next, the system arrives at the 2nd learning stage to ideally alter the parameter of the membership function. Also, the gradient learning algorithm was used to minimize error function E:

$$E = \frac{1}{2}\sum_x \sum_{l=1}^{q} (d_l - y_l^V)^2, \quad (10)$$

Whereas, q denotes the numerous nodes in layer5; d_l and y_l^V represent the outputs of the target and actual node l for an input X. Figure 3 depicts the infrastructure of HFNN.

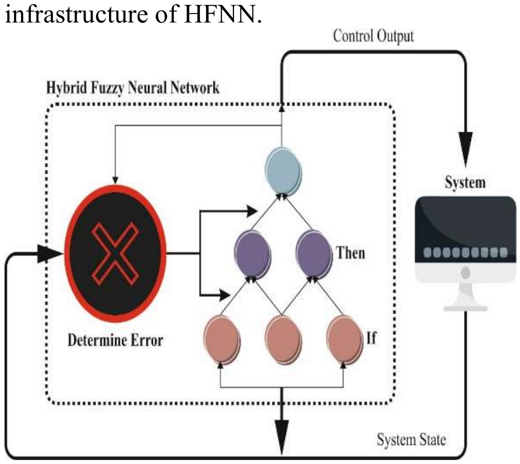

Figure 3: Structure of HFNN

3.4 Hyperparameter Tuning using SHO Algorithm

Eventually, the FRBDM-TOLA model utilizes the SHO technique for hyperparameter tuning to enhance the performance of the HFNN method. The SHOA is a novel optimizer algorithm based on the natural cooperative and adaptive behaviors ascertained in spotted hyenas [22]. Leveraging the collaborative, adaptable, and persistent hunting strategies of spotted hyenas, SHOA functions on a population-based technique, which simulates the collective quest for a solution comparable to a group of hyenas searching for food. SHOA incorporating global and local search techniques, combines adaptable operators and parameters to optimize efficiency across different fields of optimization such as machine learning, engineering design, and image processing.

Prey encircling

The strategy includes position adjustment of the spotted hyena according to the search factor for effectively locating the optimum solution. These behaviors are mathematically characterized as follows:

$$F_{hp} = \left| b * u_p(k) - u_h(k) \right| \quad (11)$$

$$u_h(k+1) = u_p(k) - c * F_{hp} \quad (12)$$

Where F_{hp} refers to the distance between the spotted hyena and the target prey. u_p and u_h are the location vectors of the prey and the spotted hyenas. k is a variable representing the existing iteration, and b and c are the coefficient vectors.

$$b = 2u_{r1} \quad (13)$$

$$c = 2m * u_{r2} - m \quad (14)$$

$$m = 5 - \left(Iter * \left(\frac{5}{Iter^{max}} \right) \right) \quad (15)$$

Where $= 0,1,2, \dots \dots \dots . Iter^{max}$. u_{r2} and u_{r2} are random vectors in[0, 1]. m is a variable that decreases linearly from 5 to 0

Hunting

The hunting strategies include assessing the total count of spotted hyenas and evaluating the spotted hyena's position about its optimum spot of the prey. This can be mathematically stated in the following:

$$F_{hp} = \left| b * u_{pfinest}(k) - u_{hfinest}(k) \right| \quad (16)$$

$$u_{hfinesc} = u_{pfinest} - c * F_{hp} \quad (17)$$

$$Dp_h = u_{h,finest} + \mathcal{U}_{h,finest} + 1 + \cdots \dots \dots \dots \mathcal{U}_{hfinest} + n_h \quad (18)$$

In this case, the optimum location of the spotted hyena about its prey is represented as $u_{pfines\tau}$, and an alternative location for the hyena is represented as $u_{hfinesc}$. n_h is the overall amount of spotted hyenas.

$$n_h = e_n(u_{p,finest)} u_{p,finest+1)} u_{p,finest+2}, \ldots \ldots \ldots \ldots (e_{p,finest+g}) \quad (19)$$

In the above equation, g denotes a random vector within the range [0.5, 1], n shows the overall count of responses, and e_n aggregates the n_h optimum solutions.

i) Exploitation (Prey attacking)

The prey assaulting procedure can be mathematically expressed as follows:

$$u_h(z+1) = e_n / n_h \quad (20)$$

This updates the location of $u_h(z+1)$ to grip the best solution and adjust the locations of other components according to the best search site.

ii) Exploration (Prey finding)

In the exploration stage, the search for a proper solution considers two probabilities for the c value: either less than or greater than 1. Another important factor assisting exploration is the usage of *the* b vector. Vector b contains a random number that allocates weight to the prey. In such cases, the vector $b > 1$ takes the superiority over the vector $b < 1$, which emphasizes the significance of distance and the random nature of SHOA.

The following steps are used for the SHOA implementation:

1. Determine optimization parameters (maximal iteration count, population size, repetition, variable restrictions, and number of variables).
2. Generate the initial population matrix, with all rows and columns representing a member and bus number for the installation of the capacitor.
3. Consider restrictions, assess the objective function for the member population, and select the hyena illustrative with the minimum MOF.
4. Upgrade the members of the population at the following location using continuous variables.
5. Round each member value to the nearest value for a distinct search range, reassess the objective function, and switch the previous member if the updated value is greater.
6. Define the ending criteria and stop if the iteration is obtained; or else, go to step 4.

4. EXPERIMENTAL RESULT AND ANALYSIS

In this section, the performance evaluation of the FRBDM-TOLA technique with recent models is given. In Table 1, the comparative average CPU utilization (ACPUU) results of the FRBDM-TOLA technique under varying workloads [23].

Table 1: Average CPU utilization of FRBDM-TOLA technique with recent models under various time intervals

Average CPU Utilization (%)				
Real				
Time Interval	**LAQ Model**	**LAF Model**	**LAFA3C Model**	**FRBDM-TOLA**
0	66.00	66.69	68.09	70.17
50	34.02	40.27	57.66	69.48
100	48.62	53.48	73.65	80.60
150	45.84	57.66	70.87	80.60
200	39.58	50.01	63.22	82.69
250	34.02	46.53	57.66	75.04
300	25.67	41.66	50.01	70.17
350	22.20	31.93	46.53	67.39
400	17.33	32.62	40.27	64.61
Smooth				
Time Interval	**LAQ Model**	**LAF Model**	**LAFA3C Model**	**FRBDM-TOLA**
0	6.23	5.53	5.53	9.02
50	74.04	71.24	62.85	84.53
100	69.85	46.77	64.25	81.03
150	52.37	67.75	73.34	85.23
200	62.85	60.76	68.45	81.73
250	67.05	67.75	77.54	83.13
300	54.47	57.96	66.35	78.93
350	62.16	42.58	58.66	71.94
400	53.07	39.08	69.85	78.93
Bursty				
Time Interval	**LAQ Model**	**LAF Model**	**LAFA3C Model**	**FRBDM-TOLA**
0	15.43	15.43	17.14	20.55
50	49.55	62.35	69.18	83.68
100	22.25	24.81	54.67	75.15
150	25.67	22.25	47.85	70.88
200	15.43	7.75	62.35	84.53
250	45.29	34.20	53.82	65.76

300	41.88	55.53	63.20	74.29
350	79.41	87.95	91.36	97.33
400	52.97	45.29	63.20	70.03

Figure 4 inspects a comparative ACPUU results of the FRBDM-TOLA method under a real workload. The results indicate that the FRBDM-TOLA technique reaches enhanced ACPUU values over other models. With a time interval of 50, the FRBDM-TOLA technique obtains a higher ACPUU of 69.48% while the LAQ, LAF, and LAFA3C models attain lower ACPUU of 34.02%, 40.27%, and 57.66%, respectively. Besides, with a time interval of 100, the FRBDM-TOLA method gets greater ACPUU of 80.60% while the LAQ, LAF, and LAFA3C models attain lower ACPUU of 48.62%, 53.48%, and 73.65%, respectively.

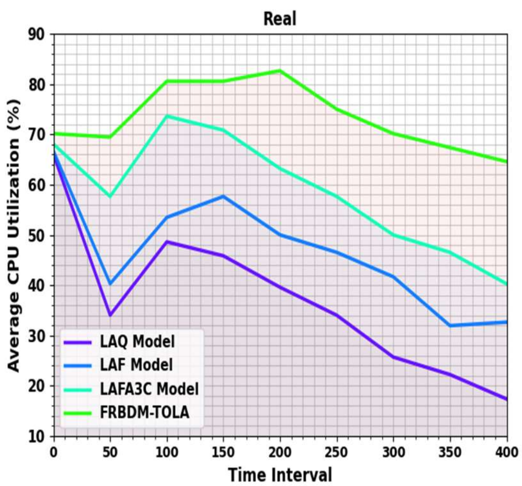

Figure 4: ACPUU outcome of FRBDM-TOLA technique under real workload

Figure 5 inspects a comparative ACPUU results of the FRBDM-TOLA method under smooth workload. The outcomes specify that the FRBDM-TOLA method attains improved ACPUU values over other techniques. With a time interval of 50, the FRBDM-TOLA technique obtains a higher ACPUU of 84.53% while the LAQ, LAF, and LAFA3C models attain lower ACPUU of 74.04%, 71.24%, and 62.85%, respectively. Besides, with a time interval of 100, the FRBDM-TOLA method gets a greater ACPUU of 81.03% while the LAQ, LAF, and LAFA3C models attain lower ACPUU of 69.85%, 46.77%, and 64.25%, respectively.

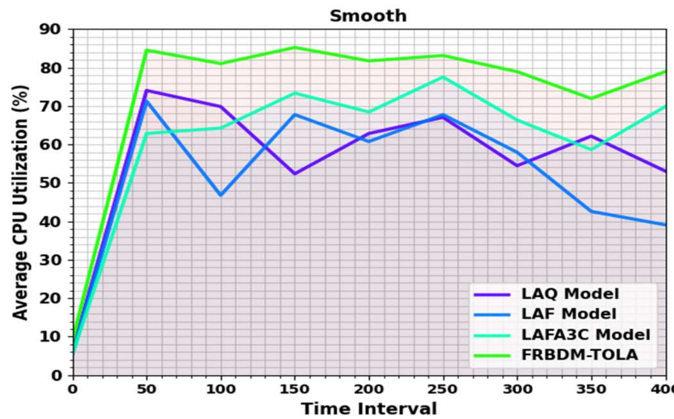

Figure 5: ACPUU outcome of FRBDM-TOLA technique under smooth workload

Figure 6 inspects a comparative ACPUU results of the FRBDM-TOLA method under bursty workload. The outcomes specify that the FRBDM-TOLA method achieves enhanced ACPUU values over other approaches. With a time interval of 50, the FRBDM-TOLA system gets a higher ACPUU of 83.68% whereas the LAQ, LAF, and LAFA3C techniques obtain lesser ACPUU of 49.55%, 62.35%, and 69.18%, correspondingly. Also, with a time interval of 100, the FRBDM-TOLA technique achieves a larger ACPUU of 75.15% whereas the LAQ, LAF, and LAFA3C approaches reach lesser ACPUU of 22.25%, 24.81%, and 54.67%, respectively.

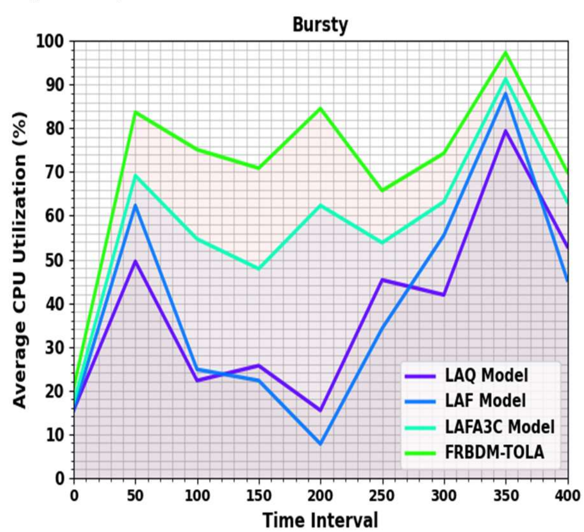

Figure 6: ACPUU outcome of FRBDM-TOLA technique under bursty workload

In Table 2 and Figure 7, the comparative delay violation (DV) results of the FRBDM-TOLA technique under varying time intervals. With bursty workload, the FRBDM-TOLA technique offers reduced DV of 102.49 whereas the LAQ, LAF, and LAFA3C models have attained increased DV of 129.80, 143.90, and 117.10, respectively. Also, with real workload, the FRBDM-TOLA model provides decreased DV of 148.37 while the LAQ, LAF, and LAFA3C techniques have got enlarged DV of 186.50, 196.50, and 163.60, respectively. Besides, with a smooth workload, the FRBDM-TOLA system achieves reduced DV of 75.22 whereas the LAQ, LAF, and LAFA3C models have attained increased DV of 124.40, 112.20, and 89.30, respectively.

Table 2: Delay Violation of FRBDM-TOLA technique with recent models under various workloads

Delay Violation			
Workload	**Bursty**	**Real**	**Smooth**
LAQ Model	129.80	186.50	124.40
LAF Model	143.90	196.50	112.20
LAFA3C Model	117.10	163.60	89.30
FRBDM-TOLA	102.49	148.37	75.22

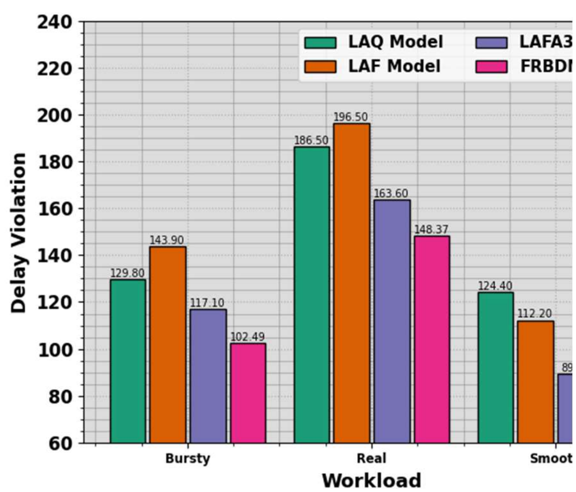

Figure 7: DV outcome of FRBDM-TOLA technique under varying workloads

The performance evaluation of the FRBDM-TOLA method with recent methods is given. Table 3 demonstrates the comparative CPU utilization outcomes of the FRBDM-TOLA system under various requests.

Figure 8 inspects the comparative CPU utilization results of the FRBDM-TOLA technique. The outcomes specify that the FRBDM-TOLA approach gets enhanced CPU utilization values over other techniques. With 5 requests, the FRBDM-TOLA system gets higher CPU utilization of 26.37% while the LAQ, LAF, and LAFA3C models attain lower CPUs of 23.41%, 22.27%, and 25.46%, respectively. Besides, with 10 requests, the FRBDM-TOLA approach attains greater CPU utilization of 37.54% whereas the LAQ, LAF, and LAFA3C models attain lower CPU utilization of 31.16%, 30.25%, and 32.98%, respectively.

Table 3: CPU utilization of FRBDM-TOLA system under recent models

CPU Utilization (%)				
Number of Requests	**FRBDM-TOLA**	**LAFA3C Model**	**LAQ Model**	**LAF Model**
5	26.37	25.46	23.41	22.27
10	37.54	32.98	31.16	30.25
15	61.47	57.60	48.48	52.58
20	71.50	68.08	58.51	62.84
25	78.80	76.29	67.40	71.50
30	87.46	84.04	76.06	80.62
35	96.80	95.21	83.13	89.28

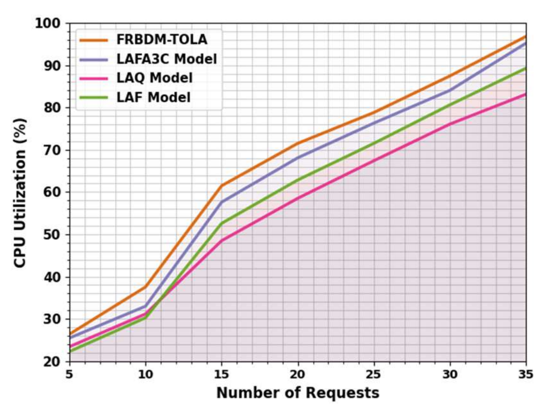

Figure 8: CPU utilization outcome of FRBDM-TOLA method under various requests

In Table 4 and Figure 9, the Execution time results of the FRBDM-TOLA system under varying requests. With 5 requests, the FRBDM-TOLA system delivers reduced Execution time of 0.08sec whereas the LAQ, LAF, and LAFA3C techniques have attained enlarged Execution time of 0.28sec, 0.32sec, and 0.18sec, respectively. Also, with 10 requests, the FRBDM-TOLA technique gets a decreased Execution Time of 0.18sec whereas the LAQ, LAF, and LAFA3C models have enlarged

Execution time of 0.32sec, 0.52sec, and 0.34sec, correspondingly. Besides, with 15 requests, the FRBDM-TOLA technique gets a reduced Execution Time of 0.28sec where the LAQ, LAF, and LAFA3C models have attain increased Execution Time of 0.74sec, 0.90sec, and 0.88sec, respectively.

Table 4: Execution Time of FRBDM-TOLA technique with recent models under various number of requests

Execution Time (sec)				
Number of Requests	**LAQ Model**	**LAF Model**	**LAFA3C Model**	**FRBDM-TOLA**
5	0.28	0.32	0.18	0.08
10	0.32	0.52	0.34	0.18
15	0.74	0.90	0.88	0.28
20	1.38	1.68	1.60	0.50
25	2.43	2.93	2.75	0.96
30	3.85	4.73	4.27	1.66
35	5.90	6.92	6.64	3.93

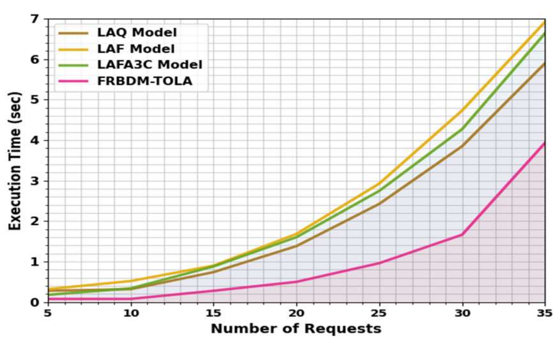

Figure 9: Execution Time outcome of FRBDM-TOLA technique under various requests

Table 5 demonstrates the comparative analysis of the FRBDM-TOLA technique in terms of MAE, RMSE, and MAPE with existing methods under varying workloads. Figure 10 illustrates the MAE outcome of the FRBDM-TOLA technique under various workloads. With real workload, the FRBDM-TOLA system provides a decreased MAE of 0.91 while the ARMA, ARIMA, and LSTM techniques have got enlarged MAE of 1.72, 1.56, and 1.43, respectively. Also, with a smooth workload, the FRBDM-TOLA model provides a decreased MAE of 1.01 while the ARMA, ARIMA, and LSTM techniques have got enlarged MAE of 1.49, 1.32, and 1.22, respectively. Besides, with a smooth workload, the FRBDM-TOLA system achieves a reduced MAE of 3.68 whereas the ARMA, ARIMA, and LSTM models have attained increased MAE of 8.23, 7.91, and 6.86, respectively.

Table 5: Comparative analysis of FRBDM-TOLA approach in terms of MAE, RMSE, and MAPE

MAE			
Techniques	**Real**	**Smooth**	**Bursty**
ARMA Model	1.72	1.49	8.23
ARIMA Model	1.56	1.32	7.91
LSTM Classifier	1.43	1.22	6.86
FRBDM-TOLA	0.91	1.01	3.68
RMSE			
ARMA Model	2.08	1.83	8.67
ARIMA Model	1.82	1.73	8.21
LSTM Classifier	1.78	1.64	7.21
FRBDM-TOLA	1.29	1.09	7.09
MAPE (%)			
ARMA Model	4.38	3.92	17.93
ARIMA Model	3.89	3.71	16.83
LSTM Classifier	3.66	3.41	15.21
FRBDM-TOLA	2.99	2.87	14.04

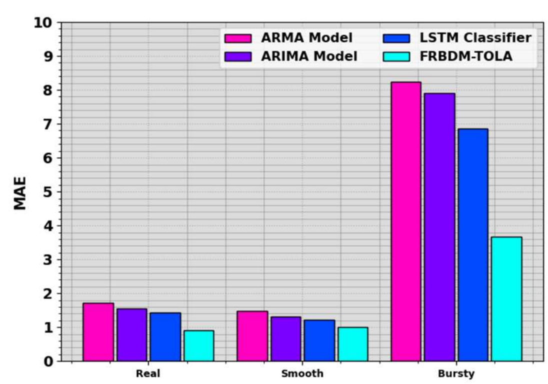

Figure 10: MAE outcome of FRBDM-TOLA approach under various workloads

Figure 11 illustrates the RMSE outcome of the FRBDM-TOLA technique under various workloads.

With real workload, the FRBDM-TOLA system provides decreased RMSE of 1.29 while the ARMA, ARIMA, and LSTM techniques have got enlarged RMSE of 2.08, 1.82, and 1.78, respectively. Also, with a smooth workload, the FRBDM-TOLA model provides a decreased RMSE of 1.09 while the ARMA, ARIMA, and LSTM techniques have got enlarged RMSE of 1.83, 1.73, and 1.64, respectively. Besides, with a smooth workload, the FRBDM-TOLA system achieves a reduced RMSE of 7.09 where the ARMA, ARIMA, and LSTM models have attained increased RMSE of 8.67, 8.21, and 7.21, respectively.

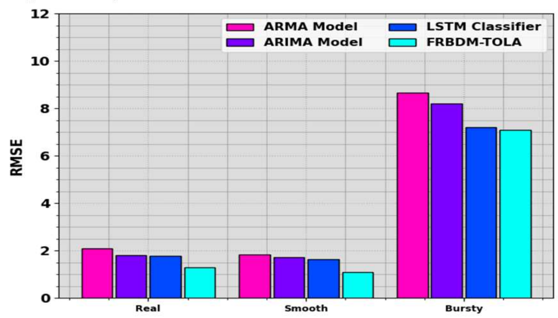

Figure 11: RMSE outcome of FRBDM-TOLA approach under various workloads

Figure 12 illustrates the MAPE outcome of the FRBDM-TOLA technique under various workloads. With real workload, the FRBDM-TOLA system provides a decreased MAPE of 2.99 while the ARMA, ARIMA, and LSTM techniques have got enlarged MAPE of 4.38, 3.89, and 3.66, respectively. Also, with a smooth workload, the FRBDM-TOLA model provides a decreased MAPE of 2.87 while the ARMA, ARIMA, and LSTM techniques have got enlarged MAPE of 3.92, 3.71, and 3.41, respectively. Besides, with a smooth workload, the FRBDM-TOLA system achieves a reduced MAPE of 14.04 whereas the ARMA, ARIMA, and LSTM models have attained increased MAPE of 17.93, 16.83, and 15.21, respectively.

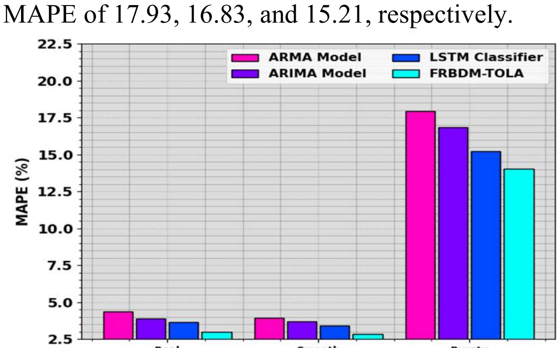

Figure 12: MAPE outcome of FRBDM-TOLA approach under various workloads

Thus, the FRBDM-TOLA technique can be applied for effectual task-offloading process in the MECC environment.

5. CONCLUSION

In this paper, we defined a FRBDM-TOLA model on multi-tier MECC systems. Initially, the presented FRBDM-TOLA technique uses fuzzy clustering algorithms like FCM to categorize tasks into fuzzy subcategories based on their attributes, intensity levels, and temporal aspects. This allows for a comprehensive representation of task characteristics considering various dimensions. Also, the developed method utilizes the HFNN technique to choose the best target node for task offloading based on latency sensitivity, server capability, and the conditions of the network. The HFNN is a hybrid method, which unites FL and neural networks for rule generation, examining intensity levels and traffic patterns. FL can occupy linguistic comments and uncertainties, while neural networks can absorb complex patterns from data. To enhance the performance of the HFNN method, the FRBDM-TOLA technique uses the SHO algorithm for hyperparameter tuning. The developed models efficiently use classification and clustering processes to enhance classification accuracy in evaluating network traffic data. The efficiency of the FRBDM-TOLA model was verified over wide experimentations utilizing the iFogSim simulator. The mathematical outcome shows that the FRBDM-TOLA model has superior performance and scalability to dissimilar processes than the present advanced models.

DECLARATION

Data Availability: No data were used to support the findings of this study.

Funding Statement: The authors received no specific funding for this study.

Conflicts of Interest: The authors declare that they have no conflicts of interest to report regarding the present study.

Ethical Approval: The declaration is Not Applicable.

REFERENCE

[1]. H.A. Alharbi, M. Aldossary, J. Almutairi, and I.A. Elgendy, (2023). “Energy-Aware and Secure Task Offloading for Multi-Tier Edge-Cloud Computing Systems,” *Sensors*, vol. *23*, no. 6, pp. 3254. https://doi.org/10.3390/s23063254

[2]. S. Alhelaly, A. Muthanna, and I. A. Elgendy, (2022). "Optimizing task offloading energy in multi-user multi-UAV-enabled mobile edge-cloud computing systems," *Applied Sciences*, vol. *12*, no. 13, pp.6566. https://doi.org/10.3390/app12136566

[3]. J. Almutairi, M. Aldossary, H. A. Alharbi, B.A. Yosuf, and J.M. Elmirghani, (2022). "Delay-optimal task offloading for UAV-enabled edge-cloud computing systems," *IEEE Access*, vol. *10*, pp.51575-51586. https://doi.org/10.1109/ACCESS.2022.3174127

[4]. S.R. Behera, N. Panigrahi, S.K. Bhoi, M. Bilal, K.S. Sahoo, and D. Kwak, (2023). "A distributed fuzzy optimal decision-making strategy for task offloading in an edge computing environment," *IEEE Access*, vol. 11, pp.33189-33204. https://doi.org/10.1109/ACCESS.2023.3262611

[5]. M. Etemadi, M. Ghobaei-Arani, and A. Shahidinejad, (2021). "A cost-efficient auto-scaling mechanism for IoT applications in fog computing environment: a deep learningbased approach," *Cluster Comput*, vol. *24*, no. 4, pp. 3277–3292. https://doi.org/10.1007/s10586-021-03307-2

[6]. F. García, F. Guijarro, J. Oliver, and R. Tamošiūnienė, (2018). "Hybrid fuzzy neural network to predict price direction in the German DAX-30 index," *Technological and Economic Development of Economy*, vol. *24*, no. 6, pp.2161-2178.https://doi.org/10.3846/tede.2018.6394

[7]. M.D. Hossain, T. Sultana, M.A. Hossain, M.I. Hossain, L.N. Huynh, J. Park, and E.N. Huh, (2021). "Fuzzy decision-based efficient task offloading management scheme in multi-tier MEC-enabled networks," *Sensors*, vol. 21, no. 4, p.1484. https://doi.org/10.3390/s21041484

[8]. X. Huang, Y. Cui, Q. Chen, and J. Zhang, (2020). "Joint task offloading and QoS-aware resource allocation in fog-enabled Internet-of-Things networks," *IEEE Internet Things,* vol. 7, no. 8, pp. 7194–7206. https://doi.org/10.1109/JIOT.2020.2982670

[9]. M. M. T. Jawhar, and M. Mehrotra, (2010). "Design network intrusion detection system using a hybrid fuzzy-neural network," *International Journal of Computer Science and Security*, vol. 4, no. 3, pp.285-294.

[10]. P. Khatri, B. Tongli, P. Kumar, A.J. Hamidovich, T.V. Lakshmi, and M.W. Bhatt, (2024). "Efficient computation for task offloading in 6G mobile computing systems," *International Journal of Data Science and Analytics*, pp.1-19. https://doi.org/10.1007/s41060-024-00506-z

[11]. N. Kiran, C. Pan, S. Wang, and C. Yin, (2019). "Joint resource allocation and computation offloading in mobile edge computing for SDN-based wireless networks," *J. Commun. Network*, vol. 22, no. 1, pp. 1–11. https://doi.org/10.1109/JCN.2019.000046

[12]. J. Limin, and Z. Ke, (December 2023). "Enhanced DQN in Task Offloading Across Multi-Tier Computing Networks," *In 2023 20th International Computer Conference on Wavelet Active Media Technology and Information Processing (ICCWAMTIP)*, (pp. 1-6). IEEE. https://doi.org/10.1109/ICCWAMTIP60502.2023.10387060

[13]. P. Mach, P. Becvar, (2017). "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Commun. Surv. Tutor*, vol. 19, pp. 1628–1656. DOI: 10.1109/COMST.2017.2682318

[14]. Q. Pham, F. Fang, V.N. Ha, M.J. Piran, M. Le, L.B. Le, W. Hwang, and Z. A. Ding, (2020). "Survey of Multi-Access Edge Computing in 5G and Beyond: Fundamentals, Technology Integration, and State-of-the-Art," *IEEE Access*, 8, pp. 116974–117017. DOI: 10.1109/ACCESS.2020.3001277

[15]. Q.V. Pham, T. Leanh, N. H. Tran, B.J. Park, and C.S. Hong, (2018). "Decentralized computation offloading and resource allocation for mobile-edge computing: a matching game approach," *IEEE Access*, 6, pp. 75868–75885.
DOI: 10.1109/ACCESS.2018.2882800

[16]. S. Sheng, P. Chen, Z. Chen, L. Wu, and Y. Yao, (2022). "Deep reinforcement learning-based task scheduling in IoT edge computing*," Sens*, vol. 21, no. 5, pp. 1666, https://doi.org/10.3390/s21051666

[17]. E.V.D. Subramaniam, and V. Krishnasamy, (2024). "Hybrid Optimal Ensemble SVM Forest Classifier for Task Offloading in Mobile Cloud Computing." *The Computer Journal*, vol. 67, no. 4, pp.1286-1297. https://doi.org/10.1093/comjnl/bxad059

[18]. X. Tan, D. Zhao, M. Wang, X. Wang, X. Wang, W. Liu, and M. Ghobaei-Arani, (2024). "A decision-making mechanism for task offloading using learning automata and deep learning in mobile edge

networks," *Heliyon*, vol. 10, no. 1. https://doi.org/10.1016/j.heliyon.2023.e23651

[19]. B. Tolba, M. Abo-Zahhad, M. Elsabrouty, A. Uchiyama, and A.H. Abd El-Malek, (2024). "Joint user association, service caching, and task offloading in multi-tier communication/multi-tier edge computing heterogeneous networks," *Ad Hoc Networks*, vol. 160, p.103500. https://doi.org/10.1016/j.adhoc.2024.103500

[20]. J. Wang, J. Pan, F. Esposito, P. Calyam, Z. Yang, P. Mohapatra, (2019). "Edge cloud offloading algorithms: issues, methods, and perspectives," *ACM Comput. Surv*, vol. 52, no. 1, pp. 1–23. https://doi.org/10.1145/3284387

[21]. S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. A. Wang, (2017). "Survey on Mobile Edge Networks: Convergence of Computing Caching and Communications," *IEEE Access*, *5*, pp. 6757–6779. DOI: 10.1109/ACCESS.2017.2685434

[22]. Y. Yazid, and I.E. Zazi, (2021). "UAV-enabled mobile edge-computing for IoT based on AI: A comprehensive review," *Dron*, vol. 5, no. 4, pp. 148. https://doi.org/10.3390/drones5040148

[23]. T. Yuvaraj, S. Arun, T. D. Suresh, and M. Thirumalai, (2024). "Minimizing the impact of electric vehicle charging station with distributed generation and distribution static synchronous compensator using PSR index and spotted hyena optimizer algorithm on the radial distribution system," e-*Prime-Advances in Electrical Engineering, Electronics and Energy*, vol. 8, p.100587. https://doi.org/10.1016/j.prime.2024.100587