National College of Ireland

Improvement of Intelligent Task Prediction and Computation Offloading towards mobile-edge cloud computing.

MSc Research Project
Cloud Computing

Pravin Ravindran Shanmugam

Student id: x23186607

School of Computing
National College of
Ireland

Supervisor: Shivani Jaswal

National College of Ireland Project Submission Sheet School of Computing

Student Name:	Pravin Ravindran Shanmugam
Student ID:	X23186607
Programme:	Masters in Cloud Computing
Year:	2024
Module:	MSc Research Project
Supervisor:	Shivani Jaswal
Submission Due Date:	29/01/2025
Project Title:	Improvement of Intelligent Task Prediction and Computation Offloading towards mobile-edge cloud computing.
Word Count:	10032
Page Count:	24

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.

ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature:	Pravin Ravindran Shanmugam
Date:	29th January 2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST:

Attach a completed copy of this sheet to each project (including multiple copies).	Q
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	Q
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keepa copy on computer.	Q

Assignments that are submitted to the Programme Coordinator office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Improvement of Intelligent Task Prediction and Computation Offloading towards mobile-edge cloud Computing

Pravin Ravindran Shanmugam

X23186607

MSCCLOUDB – MSc Research Project

National College of Ireland, Dublin

Abstract

The research focuses on the improvements in the field of edge computing towards the ability to predict the tasks and offload the tasks according to the requirement. There are lots of issues faced during the computation as the mobile devices are insufficient to handle highly intensive tasks which may be led to loss of battery power, bad performance and high execution time. To overcome these drawbacks instead of having fixed decision-making ability, dynamic decision-making ability to act upon sudden spikes in the behavior of the components. Because fixed decision-making ability will not be suitable for the real time tasks. Machine learning model can be applied to predict the task based on the complexity and take decisions to perform the tasks locally or offload them to the edge server based on the length of the complexity. The decision – making ability depends on the factors such as memory availability, task complexity, network latency and CPU usage. The trained random forest model has shown the improved precision in predictions towards the offloading decisions. The observed results indicates that the random forest has the highest accuracy that is 53.3% followed by the XGBoost model with 52.5% and then the Logistic Regression with the 49.1%. The research leads to enhanced accuracy in the process of making decisions in the real time by the utilization of optimized machine learning model in the AWS SageMaker and iFogSim Environment.

Keywords: Edge computing, Machine learning model, offloading decisions, AWS SageMaker and iFogSim Environment.

1. Introduction

In the current trend of the world, most of the important and necessary things which are essential to make lifestyle easy and fast are available in the hands through the applications available in the mobile devices like maps and navigation, social networking, e-shopping, virtual reality, face recognition and entertainment. Nevertheless, running such a highly intensive apps could bring a lot of processing workload to the devices. Processing the high complex tasks of the applications within the mobile devices would result in poor performance and high execution time. However, with the continuous updates coming on the working system of the applications will require more computational power to complete the requirements of the updates which could led to the difficulty of achieving this because of the computation capability in available in the local device as well as due to the limited battery power of the mobile devices. This was the motivation behind to choose the topic of mobile edge computing, providing the ability to offload the tasks based on the complexity to edge with this enhancement the above faced issues might be solved and led to fast and smooth computation and processing. However, the move from cloud computing to edge computing as it is not suitable for real time applications which needs a low latency requirement.

To address these issues efficient resource allocation and proper decision-making ability is crucial. This is where Mobile-edge cloud computing (MEC) comes to play to provide solutions for the above issues by having cloud and edge computing to process the resources close to source. By utilizing this there would be less latency in returning responses of the application and provides better performance of the application which is been accessed. The two main components of the MEC are the computation offloading and intelligent task prediction in which these can predict the future processing demands based on the observed user's application needs, present networking environment, transaction and behavior patterns. Then determining the decision of offloading the computational tasks to close by edge or cloud server based on the observed complexity. By employing this strategy, the amount of energy and execution time as well as the weight of load consumption on the mobile devices would be reduced leading to a better performance.

The importance of the computation offloading, and intelligent task prediction leads out for achieving a success parameter in terms of the support with the MEC environment. MEC system employs intelligent method to predict the future needs and use machine learning techniques to allocate and manage the available resources in the places of the real time needs and requests which would lead to high productivity in usability and the mobile application's response time.

1.1 Research Question and Objective

Research Question: what may be the impact of implementation of compute offloading and intelligent task prediction by combining more modern approaches to AI combined with more realistic datasets to significantly enhance the system performance and maximize resource efficiency in terms of mobile edge cloud computing?

This Proposed project utilizes the machine learning techniques to predict whether the task needs to be processed internally in the local device or to be offloaded to either edge or cloud based on the complexity. The process of this decision-making ability depends on features like memory availability, network latency, task complexity, user preferences and CPU usage.

The structure of the document is as follows: Section 2 explains the literature review which entails the work of earlier research related to the research question. Section 3 holds the Research methodology specifications which entails the approach, design implementation and research resources for the implementation. Section 4 describes the results which will clearly specify the outcome of the project in detail. Finally, last section holds the Conclusion which will draw the overall outcome of the learnings observed in the process of the implementation.

2. Literature Review

In this review process the insights of the working functionality and the technical knowledge related to the computation and intelligent task prediction are described in detail with the previous research done related to the specified field. In the Mobile-edge computing, the important factors that can be considered to increase the user experience and the efficiency of the system in the different applications sector are the task prediction and computation offloading.

2.1 Mechanisms of Machine Learning for Task Prediction

In the task prediction the ML mechanisms provide a vital impact in the field MEC as they offer trustable remedies for dynamic forecasts for the prediction requests. Mainly utilized models are neural networks, deep learning models and a collective technique for the purpose of historical data and user behaviour analyses. The prediction precision will be improved by these models as they could adapt to quick changes and act according to the situation. In the Fuzzy Q-learning, the auto-scaling models require accurate computations by predicting the future workloads, LA-based offloading decision works well for understanding different offloading requests (Nikougoftar et al., 2023). While improving the accuracy prediction in complex and dynamic MEC environment, the ensemble technique is used to reduce the biases by combining various ML algorithms. The above approaches will be useful in terms of allocating the tasks to the appropriate resources which will lead to reduce in the execution time as well energy these parameters are important to the improvement of MEC.

2.2 Offloading Schemes for Mixed Assumption Computation

In the edge computing paradigm and in the cloud for highly efficient task allocation happens through strategies in the model of hybrid computation offloading includes the methods like static, dynamic and heuristic based methods for the process. In the stable network load and conditions, the static approach will work fine as it offloads power according to the predefined set of rules. Improvement in the performance as well as with the solutions are achievable with the Real-time sentinel approach as it has the ability to adapt according to the changes on the network and load (Tan et al., 2024). For creating a probabilistic option regarding the decisions of resource utilization and latency, the suitable way of handling this is by rule-based computation achieved by the heuristic approach. The above strategies are very much useful in the mobile conditions which can sometime undergo huge or unexpected processing, the efficient resource and energy allocation might upgrade the user experience by decreasing the execution time (Maleki, Mashayekhy & Nabavinejad, 2021).

2.3 Task Offloading Optimization through Swarm Intelligence

In MEC platforms, to increase the offloading of workloads Heuristic algorithms such as particle swarm optimization and grey wolf optimization are employed to achieve this. In terms of finding the best or nearly best solutions, this approach acts based on the strategies in the natural system. For example, to decide on the decision of offloading the workloads with the grey wolf optimization technique they will utilize the network state, energy quality, power consumption and computation resources to search for the vacant spaces (Mahenge et al., 2022). By frequently modifying the coordinates of the swarm particles according to the evaluated values based on the fitness, particle swarm optimization improves the strategies of offloading and aims to increase the system performance and scalability. By the utilization of the swarm intelligence in the field of the MEC systems, it is feasible to achieve ability to act on the sudden fluctuation on the workload and the environment to maximize the response time and the resource utilization.

2.4 Applied Strategies for Offloading Computation

Here there was a wide range of research has been taken to examine techniques such as static, combination of static and dynamic, dynamic models for the offloading process these strategies will be very much helpful in efficient allocation of resources. In this part we must analyze the abilities of the working system such as average of energy consumed, number of the tasks can be processed, availability of the assets and the resources. Now it will be very simple for implementing the offloading of tasks as the above scenarios have already been analyzed as the system will be ready to face any kind of requirements under various situations leading to make the more flexible system. By following these above approaches, it will provide a way to offload the computation tasks based on the severity to either edge or cloud computing environments that possess the required resources to complete the tasks. On the basis of the above approaches, it can lead high execution time that might cause the lag, causing negative impact towards the customers. Anyways this paper provides to handle this issue by offering multi-task parallel scheduling as solution for the above issue. Genetic and greedy algorithms have been proposed to result in a solution for the related issues in terms of scheme evaluation algorithm for DNN tasks scheduling. When comparing the performance between the genetic and greedy algorithms, the genetic algorithm will perform much better requires a little more time (Chen, 2020).

2.5 Heterogeneous Cloud Computing Dynamic Resource Management

The incorporation of the collaborative cloud edge with the frameworks for adaptive resource allocation to allow the system for flexibility in the scheduling of the jobs based on the availability of the resources. To be a future ahead in determining the amount of the upcoming tasks to be performed and find an effective ways to distribute these workloads according to the severity to either edge and cloud, these frameworks depends on the machine learning algorithms and predictive algorithms (Guo et al., 2024).One of the important key features towards the real time applications will be the adaptive resource management, cause this grant the ability to allocate the resources according to the requirements to be necessary for achieving the performance of the application by allowing periodic updates on the system's parameters such as device capacity, latency of the network and availability of the bandwidth. Thus, this strategy helps in lowering the operating costs and the resource usages in the MEC settings which could lead to the adaptable system growth and quick responses to the user. Moreover, with the integration of the more advanced computational intelligence specifically with the algorithms of the machine learning it allows the system to increase the adaptability of the application domain and the efficiency of the system which could lead to the advancement in the functionality of the resource optimization (Heidari et al., 2023). To improve the prediction standards and the model's ability to adjust and act according to the new different types of the MEC environments was the main goal aimed to be achieved with the adaptive resource allocation.

2.6 Contextual Approaches for Task Offloading

To enhance the decisions for offloading by taking into the account of the application as well as user context and network behaviour, the current CT has produced many numbers of CA-TO policies for achieving this. These approaches will be useful for distributing the workloads between the cloud and edge which would navigate to the way of enhancing the resource utilization and the latency based on the models of prediction and the machine learning algorithm (Huang et al., 2024). According to the context, the possibility and option of offloading will be possible by certain factors such as application features, mobility of user and network conditions. It can be modified dynamically to yield the better possibilities of outcomes for the user and the system (Liu et al., 2024). As a result of this, the approach had shown improvement in the MEC systems proving that they can adapt to sudden changes in the environment and workload of the systems, that is very much helpful in the scenarios where there need to be an accurate time completion of the data processing and the tasks. In the wide range of IoT application domains including mobile multimedia services and the smart city environments, the context aware task offloading solutions are necessary for the correct and appropriate choice for management of the available resources in the real time applications to support its focused goal towards the required business**.**

2.7 Optimizing Offloading tasks based on Meta-heuristic Approach

Based on this area, an intelligent offloading mechanism was constructed by the ideas based on the meta-heuristics method related to the field of mobile edge computing for taking care of the offloading of tasks for the system (Abbas, Raza, Aadil, and Maqsood, 2021). Based on this study, it is possible to predict the clear and efficient way of choosing tasks to offload that will enhance the performance, appropriate resource utilization and reduce the execution period. This method is well suggested to be used with techniques like genetic algorithms and simulated annealing. The system can make the decisions automatically which will be very useful in the place when the tasks to be offloaded and to where to be offloaded depending on the factors such as consumption of energy, availability of tasks, current network conditions and complexity of the tasks. The important highlight of the meta-heuristics approach is to predict more effective and evident direction to reach out to find the possible solution for the problem. By utilizing the meta-heuristic algorithm, enhanced performance and minimized execution time can be achieved by giving them the enhanced efficiency by effectively offloading the required tasks (Abbas et al. 2021).

2.8 Ethical Concerns and their future

In terms of achieving more progress and socially acceptable answers, it is much needed to observe the future challenges and moral considerations towards the computation offloading of the intelligent tasks and prediction models in the MEC settings. Among these include features such as security, privacy, and equity or justice in the allocation of the resources (Yang 2022). Holding very strong corporate governance policy, anti-fraud polices, and the standard regulations is very useful in protecting the user data and safeguard the transparency in the process of the decision making which is essential to tackle these issues. To achieve the implementation of offloading technique with the collaboration of artificial intelligence in many platforms as much as achievable, so the future research agendas should focus on implementing the ethical considerations (Farahbakhsh 2023). Furthermore, expanding our understanding towards the components of predictive analysis and machine learning which might led to increased accuracy of forecasts, promoting the enhancement of the MEC networks which will be able to adapt to the social and the technical developments. To progress in the required advancements and more advancements towards the AI technologies with the benefits of the humans, MEC systems must be able to investigate the ethical concerns and the future problems by itself that will be very much helpful in the future study.

2.9 Techniques for mobile edge computing task offloading

Large number of applications have been developed by usage of 5G and 6G technologies, while deploying these complex and sensitive applications it might lead to bandwidth issue and delay in the internet. To overcome these issues Mobile Edge Computing (MEC) was utilized as they have ability to overcome these issues. In this paper it splits the decisions about offloading into five categories such as maintaining energy consumption and latency, reduced energy consumption, minimized delay, initiating high computing offloading and handling various application states. After that it evaluates and contrast the previous studies in that specific area. Many numerous applications have used cloud computing but that is not enough for the real time Internet of things applications as they need low latency. Parallelly, they need for the device-based computation have been increased as the growth of the IoT technology and the massive utilization of the smart technology. To overcome these obstacles the emerging technological shift has moved to distributed edge computing to address all the problems. In this paper, the main focus is making the decision of unloading of tasks for mobile edge computing. They provide a detailed information regarding the concept of the mobile edge computing with the associated technologies and the models of the unloading (Dong 2024).

2.10 Mobility-aware task migration and computation for edge computing

Mobility aware devices are considered to be the essential elements of Industrial Internet of Things (IIoT). Even though they have some drawbacks towards the computation power and the battery ability which make them unsuitable in the scenarios like where it requires larger bandwidth and powerful processing capacity for completing the complex intensive tasks. Although offloading can maximize the device processing ability, incorrect offloading decisions is not suitable for the current mobile edge computing and mobility of the devices, also it has the limited capacity to adjust to different environment conditions. In order to tackle these issues, in this study mobility-aware computation offloading and task migration approach has been proposed that reduces the system energy consumption and the task turnaround time this proposed system is based on the resource prediction and trajectory. At the same time, it reduces the task migration rates by continuously enhancing the decision rates. This approach utilizes the Long Short-Term Memory (LSTM) to monitor the time varying features in the IIoT and lagrange interpolation algorithms to predict the trajectory of the mobile devices. Deep Deterministic Policy Gradient (DDPG) will uphold the prediction results to help with the process of making decision on allocation of resources, online computation offloading and task migration. Presently cloud computing will be mostly used for processing data intensive jobs cause the mobile devices to have limited battery capacity. Anyways the mobile cloud computing (MCC) strains the core network and does not have ability to satisfy the latency requirements. To solve this issue, Mobile edge computing (MEC) which fills up the gaps that lags in the cloud computing. It is necessary to keep an eye on the resource fluctuations in the mobile devices and the possible ways of finding the task execution failures due to server dysconnectivity or variations in the device locations that will lead to the computation resource and the energy wastage during the study of the mobility-aware computation offloading problem. To tackle the issue, the agent must move the work from one edge device to another edge device which appears to be closer. The device which is closer to the source will return the results to the device (Qin 2024).

3. Research Methodology

In this area it highlights the phases of the research, the design of the project and the grounded approach towards the mobile edge computing towards the envision of the prediction scheme, offloading scheme and the assessment metrics. In this section, the study includes various stages that are very essential for creating and checking the intelligent framework for predicting and offloading the tasks in the MEC Settings.

3.1 Data Collection and Ethical Declaration

The dataset utilized in this study was obtained from Kaggle and known as "Mobile Edge Task Prediction." In mobile edge computing system, it offers wide range characteristics related to task offloading. This dataset's main aim is to coordinate the building the models of machine learning which provides the ability of predicting the correct place to execute the computational activities, whether it is to be offloaded to mobile device or to be edge server. This dataset holds several characteristics which includes task complexity, network latency, energy cost and device processing capacity, that influence on the task offloading.

By including these features, this dataset reflects the trade-offs and backlogs that edge servers and mobile devices face during the actual world while performing computational tasks. With this utilization of the dataset training the model can maximize the efficiency of the mobile edge computing. These models must be trained to test the features of the job and decide the best execution place depending on the elements like network conditions and device capabilities. Resource intensive applications such as virtual reality, augmented reality and real time analytics rely on mobile applications to maintain the energy efficiency and response time. This dataset's license agreement solves the ethical issues related to it. The dataset motivates the fair use, allow for change, allocation and utilization in the research as long it has been acknowledged by the original user, in compliance with the MIT License that is the Kaggle license.

3.2 AWS SageMaker

A cloud-based platform which offers effective, scalable and providing a secure environment for developing, deploying and training the machine learning models by implementing task offloading by machine learning models. With this configuration they have the addon benefits with the Amazon Web Services capacity, flexibility and providing the confirmation of the model with the ability to handle the large datasets and ability to handle complex computations for the prediction of task offloading. The initial step begins with the process of uploading the "Mobile Edge Task Prediction" to Amazon S3 bucket that serves main storage system for cloud based. The dataset may be quickly processed and fetched from

S3 it helps with training, assessment and planning. Furthermore, with the collaboration of S3 with the SageMaker makes it easy to automatically upload the data into the SageMaker notebook instances that leads to speed up the data transfers and processing. Data preprocessing is to be carried out by SageMaker after that have been saved from the S3 bucket. In the way to make the dataset ready for training it needs to be cleaned that involves removing the null values and changing the features as required. SageMaker has some tools like SageMaker Processing and Data Wrangler to make the best use out of it for preforming the preprocessing. With these tools they make the data best suitable for training which will lead to precise and effective predictions.

Furthermore, by optimizing and determining the model hyperparameters for the data, SageMaker improves the model performance by the hyperparameter tuning. After the training the model is deployed in the SageMaker for accurate prediction. One of the best features of the SageMaker deployment is has the capacity to automatically monitor the model accuracy which helps the model to be highly stable and availability. By predicting the outcomes based on the data that is received and the model performance, SageMaker integration with other AWS services like lambda and CloudWatch enable real time prediction in the prod environment.

Data Preprocessing: To confirm the model can learn effectively form the data preprocessing is required that includes like encoding categorical features, number values and managing the missing values which all required cleaning the dataset.

Model Training: Numerous ML models like random forest model, logistic regression model and XGBoost model have been assessed. For this research purpose we have implemented Random Forest Model for its ease of use and effective solution in addressing the categorization problems. To ensure that the model is trained to act according to different circumstances the dataset should be spitted into training and testing based on the respective ratio.

Analysis: Performance metrics like F1-score, accuracy and precision will be used to examine the trained model performance after the training to evaluate the prediction accuracy for the models based on the score.

Deployment: Followed by the training and validation the model will be deployed in the SageMaker and get it available as the endpoint. This makes it possible to integrate with the iFogSim Simulator to permit predictions in the real time.

3.3 iFogSim Simulation for Task Offloading

The simulation environment utilized here is the iFogSim, that supplies independent platform for developing the model and assessing the architecture of mobile edge cloud computing. Researchers could be able to test the effectiveness of the various task offloading schemes, management of resources and setup of network in the virtual by the utilization of the iFogSim that is responsible for simulation of fog and edge computing. With the utilization of the iFogSim they could handle very intensive real time edge computing situations in which they have tied a collaboration with the cloud, edge and mobile to process the intensive tasks. Various components of the mobile system have been integrated with this iFogSim environment that can closely be a clone like mobile edge settings. They hold they are following components like edge, cloud, network infrastructure and mobile devices that is responsible to unite them. In this simulation environment every element present hold a separate entity with the responsible for independent computation ability and network. With the above features ability, it makes it possible to completely evaluate the resource requirements and latency for the task execution schemes.

The main aim of this simulation framework is to optimize the offloading options depending on the predictions possible by the machine learning that is been deployed in the AWS SageMaker. Based on the above the predictions it helps to determine whether the tasks should be processed locally at the mobile device or cloud or edge server. Because of the iFogSim's adaptability they can be easily integrated with the models of predictions to decision making processing, leading to the offloading process based on the response time, demand of energy and utilization of the resource. The iFogSim will be able to simulate various conditions of the network that reflects in the task offloading options like varying latency and bandwidth. These characteristics is very important for getting to know how the machine learning model will function depending on the different conditions of the network like restricted, low and fast latency environments or high latency options common in the setting of the mobile.

To evaluate the machine learning model availability and flexibility in varying conditions, scenarios in high demand, the simulation require the extra things to consider like varying workload on the edge and cloud server. Additionally, the iFogSim has a feature to keep a track on the performance metrics in each offloading scenario that offers valuable insights about the processing between the local and edge node. This research can forecast for the effective setup for the management of real time task by designing different offloading schemes and evaluating how the affect the energy and efficiency consumption. They assist in evaluating the machine learning's ability to control offloading and then fine tune for the optimal performance.

3.4 SageMaker Endpoint Integration in iFogSim Simulation

To make the effort for coordinating the real time decision making for the offloading of tasks in a mobile edge cloud environment, in this area it describes the process of how the SageMaker endpoint has been integrated with the iFogSim simulator environment. With this integration it leads to enhance the quickness and effectiveness in the process of offloading of tasks depending on the predictions by direct connection between the SageMaker and iFogSim which therefore increasing the performance of the mobile edge environment.

Integrating iFogSim with the SageMaker: To establish the connection between the SageMaker endpoint and the iFogSim in the secure way the AWS SDK for java have been utilized for safe connection. To make sure the data transfer in the real time is achieved by the iFogSim capacity to communicate in a programmatical way with the connection SageMaker API. iFogSim can interact with the SageMaker endpoint when the jobs are been initiated in the collaborating with the simulation of the SDK. This provides a path to send data's regarding the features and get to know the predictions of the real time scenarios about offloading options. To provide a guaranteed way of achieving the authorization is allowing only the authorized instances of the iFogSim that might have the access the endpoint of the SageMaker, the collaboration is necessary way to achieve the permission and authentication process. The reason is the SDK coordinates the process safely this process helps in the communication with the iFogSim and the SageMaker with the secured way and able to input the requests. With the utilization the SDK, the iFogSim will be able to enhance the communication with the simulation intense maximizes, providing a trust through smooth flow of the data when there is a heavy load of tasks.

Predictions of task offloading in real time: Features such as complexity of the task, available bandwidth, memory usage and network latency have been organized and have been send to the endpoint of the SageMaker as task have been initiated in the iFogSim simulation environment. Based on the inputs provided to the endpoint of the machine learning model it provides a way to determine whether the tasks to be performed by the edge or cloud server or can be performed by the local device. iFogSim can react to the model's option in real time instantly crediting to its real time prediction iteration that establishes a mechanism that have feedback allowing the jobs to de either performed locally or to offloaded to the right server based on the resource and network conditions at times. As the system has the capacity to adjust dynamically to the changes of the observed availability of the resources and the conditions of the network, this is very useful for the applications that are needed to perform quickly and concentrate with the latency. Additionally, the continuous vision of the predictions from the SageMaker providing a trust that iFogSim that provide options based on the data driven which can enhance the efficiency based on the terms that include energy, cost and the speed. The simulation framework may be closely replicating the real time mobile edge system behaviour by providing the feedback to the predictions of the real time and offering a high vision by showing a function of the model in the live deployment.

4. Design Specifications

4.1 Utilized tools and frameworks:

This project's study complete creation and implementation relied on the collection of the tools and the frameworks made to handle data implement the machine learning model and provide a real time simulation environment. With the combination of the technologies, they offer a detailed approach to assess the task offloading tactics in the mobile edge cloud environment. The following are the lists of the frameworks and tools that have utilized in the research.

Amazon SageMaker: AWS SageMaker is the completely supervised machine learning solution the purpose was intended for helping the developers and the research scientists in the field of data in the process of the effectively creating, training and implementing the machine learning model. It has the detailed suit of the tools that can simplify the intricacies of the machine learning system. With the advantage of the SageMaker the users have a plus point on getting better performance on a high demand computing by the utilizing the preconfigured

Jupyter notebook instances for the ease of exploring of the data and preprocessing. SageMaker is very much flexible as it enables ease of the custom code, frameworks and the libraries with this the overall addition of its inbuild algorithms that are enhanced for the speed and the scalability. After the models have been trained the SageMaker will be utilized to deploy them easily to the environment as it has the features such as the auto scaling, split testing and the monitoring of the model available options to trust for the continuous accuracy.

Amazon S3: Huge datasets utilized for the machine learning and other applications can be mostly stored and securely managed by the utilization of Amazon S3(Simple Storage Service) which is a secure and scalable choice for the cloud storage. As the S3 is versatile for handling different workflow of the machine learning as it allows the users to be able to store the processed and raw data or output of the models in the different forms. The AWS S3 serves as the primary storage system of data when collaborating with the sagemaker which making it ease of entry to the training and testing of the data that can get directly. Even with the complex intensive datasets the collaboration will be very smooth in terms of managing the data the main reason behind this is S3 that promotes the fast access and retrieval of the data.

iFogSim: The iFogSim is a toolkit for simulation especially designed for the Internet of things (IoT) and fog computing environment. It provides a way for the developers and the researchers to construct how the cloud infrastructure, fog devices and the components of the network observe and communicate and process of testing completely and analyzing of the dispersed applications. Features of the simulating various parts of the fog computing that involves the allocation of the resources, offloading of tasks, latency of network and usage of energy accessible by the iFogSim. Above mentioned features will help the users to analyze and enhance the task management schemes like picking when and where to offload work for achieving enhanced efficiency and effectiveness of the cost.

4.2 Algorithms and Strategies:

In his research there were three machine learning algorithms have been chosen to solve the classification problem and for deciding which set of tasks to offloaded to edge or processed locally in the system. The utilized algorithms have discussed below.

Random Forest Algorithm: It is a machine learning method that utilizes many decisions trees response rather than relying upon one tree. Based on each tree's response it provides us more accurate information about the predictions. Each tree has a unique response whether to offload or process it locally but combining all the tree's response it provides an accurate accuracy in terms of to offload the task or not, thus this method is more reliable. This model trains all the trees with the different set of data to cover the hidden patterns of the data so that it can cover all the scenarios, so it won't be a surprise that it had not covered a scenario and by this it avoids overfitting.

Logistic Regression Algorithm: It is simple algorithm that resolve the problems with the results indicating yes or no. It looks the dataset and identifies the key features such as network speed, latency, task size and decides to come to decision whether to process it locally indicating no and indicating yes if to be offloaded. It utilizes the math formula to predict the chance of each possibility results. It decides to offload the task if the possibility percentage is above the half ratio else it can be processed locally. Concluding this model with the simplest and efficient model to be used as it can make decision based on the features, compared to other models it is the simplest model with the efficient results. It uses the logistic function which holds the mathematical calculation that helps to determine the task to processed locally or to be offloaded. This is feed with the dataset that holds the key features that are very important as they impact the decisions to make.

XGBoost Algorithm: This algorithm is the most potent machine learning algorithm that is purposely developed to provide more extremely correct predictions, and most important factor is it will perform operations quickly compared to other algorithms. This is considered to be the powerful cause it builds a sequence of decision tree to enhance the predictions where each of the tree will be able to learn from the tree to improve the predictions this is very simple and efficient technique for the classification problems. It processes very quickly and efficiently with the huge datasets with the possibility of combining advanced methods. It has been observed as the best model for the classification problems compared with other models to handle huge dataset. XGBoost constructs a one by one a sequence of decision tree in which that one tree will work in a way correcting the mistakes made from the other tree that makes it to predict correctly. It begins the process with the initial decision tree and try to predict which might not give accurate results. And then model follows up with the second tree that identifies the mistakes that was caused by the first and then solves that issue and proceeds further.

4.3 System Architecture:

The Research utilizes the combination of the mobile-edge computing and machine learning approach to make accurate decisions on task offloading in the real time environment. It analyses the task features by utilizing the prediction algorithm to decide whether the job to be offloaded or to be processed locally in the system. Resource usage has been optimized and the system responsiveness has also been enhanced through the above integration.

Model Training using SageMaker: In this research to provide comfortable access and confirming the scalability to achieve this the storing of data is handled by the Amazon S3. The provided model is to be trained with the holding of data present in the AWS SageMaker to provide a confirmation on whether to offload the jobs or to perform it locally. Once the model is trained, they have been deployed as SageMaker endpoint once the SageMaker can efficiently control the computational demands. This API will be acting like a live API that offers help in the predictions of the decisions to be taken in terms of offloading in the real time. The deployed API is abstracted in the pickle file and been utilized in the iFogSim. So, this iFogSim holds the data that has been send to the deployed API that decides the offloading decisions in the response.

iFogSim Environment: A simulation suite known as iFogSim is responsible for building a virtual world that exactly provides a clone of the cloud architecture, edge server and mobile devices. It provides the ability to observe the behavior of the system by simulating various parameters such as varying device capability, size of the task and the speed of the network. Through the real-time decision-making process this tool will be very helpful in terms of deciding and testing the ability of the offloading model performance. This gives the help to understand the measurement of the important parameters that included features such as resource consumption, latency and response of the system. By following these approaches, they would help us improve the performance before deploying it.

SageMaker Endpoint Real Time Predictions: The tasks that have been generated by the simulation environment that needs to be passed to the machine learning model that are holed by the Amazon SageMaker. This model has been deployed the API which is then analyzed based on the data provided and with that data it provides the accurate predictions of decision in the real time. Based on the above predictions on the data provided with the key features it should determine in terms of the provided tasks to be processed in the local system or to be offloaded to higher environment like cloud servers. This iFogSim simulation environment gets the results of the predictions and then utilizes them to offloading procedures. This configuration is very helpful in making the right decision related to offloading as it provides quick and trusted solution even in unpredictable situation like the real-world situations.

Method of Making Decisions: Based on the predictions results of the model the place where the task needed to preform has been decided. The task can be performed locally in the system if the prediction results indicate that it is efficient for this job to preformed locally. If the prediction results lead to information that the job to offloaded, then it should be offloaded to either edge or cloud server. By effectively distributing the resource with the devices and the servers it results in providing an effective decision-making process. Overall, by increasing the effective usage of the resources and reducing the time required for performing the tasks it enhances the performance ability of the mobile edge environment.

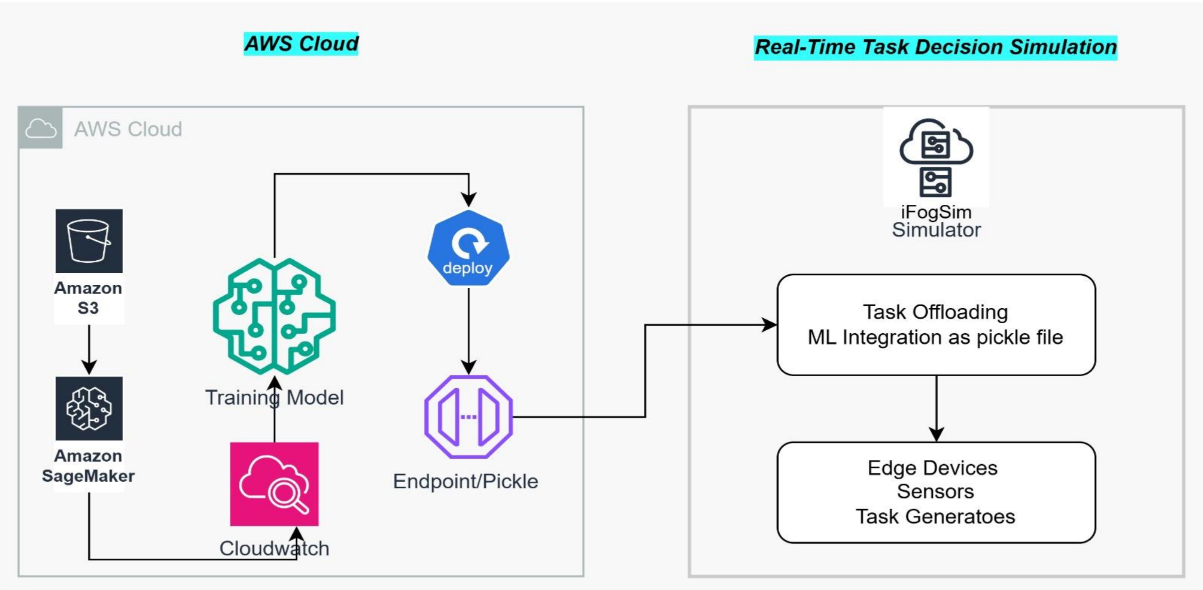


Fig. 1: Architecture Diagram

4.4 iFogSim Configuration:

Development of Application Modules and Fog Devices: In iFogSim we develop a device that are virtual that are known as fog devices which is necessary to mimic the real time systems such as mobile and edge or cloud servers. To replicate the constraints of the hardware that created each fog devices that can be configured with the key features like processing power of the task, available memory and the bandwidth. For an instance the processing ability of cloud server is far better than the local device because it has more processing capability and memory availability. These setups can also aid in the results of the simulation related to the performance issue and the resource constraints that are seen in the actual environment.

Integration of ML Model Pickle File: This project connects the iFogSim simulation with the trained machine learning model that have been implemented in the AWS Sagemaker that utilizes the SDK for Java. With the help of this integration the iFogSim can simply send the data easily to the Sagemaker model. The iFogSim interacts with the deployed model's endpoint through the SDK with the tasks that have been generated through the simulation. After the analysis and processing of the generated tasks the endpoint helps in providing the accurate decisions in terms of that tasks to be offloaded or to be performed locally in the system. This configuration setup enables the seamless communication with the machine learning model and the iFogSim Simulation. It aids in assessing the system's ability to respond fast, precise decision-making ability in the real time situations.

5. Implementation:

In this study the main aim is to predict the real-time decision-making ability towards the task offloading in the mobile edge environment, this is achieved through the integration of the machine learning model that have been trained on the SageMaker with the iFogSim simulation platform. The process of the implementation has been discussed below in the subsections.

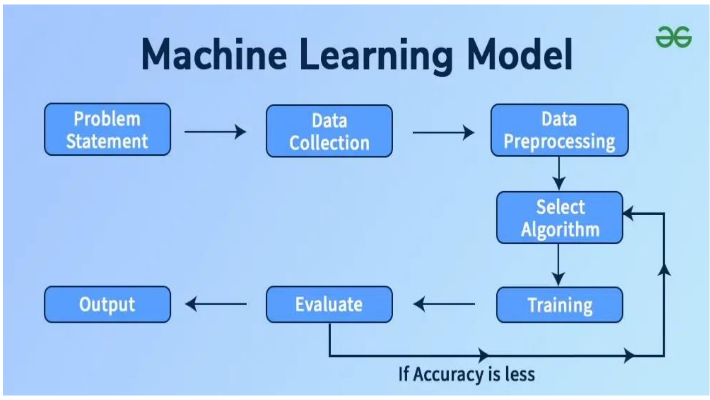


Fig.2: ML Model Task Prediction

5.1 SageMaker model deployment

The key role in this solution was the setting up the machine learning model on the SageMaker for implementation. Due to the random forest robustness, interpretable and performance efficiently on the datasets, they will be utilized to train the model to decide the task offloading decisions. The major process that are involved in the process are entailed in detail below.

Selection of Algorithm: For model training random forest model was chosen as it can handle the high dimensional data, and it delivers precise predictions without any overfitting in the data. This machine learning model is the most suitable for job offloading problem as shown in figure1, which directly impacts the decision that rely on the key features present in the dataset such as CPU Utilization, availability of memory, task size and network environment. While performing the local testing in the environment has been observed that the random forest model provides more accuracy than the other models that we have chosen which is logistic regression and the XGBoost. Through this results it is been observed that the prediction results received are worthy.

Model Training: The selected model will be trained with the dataset that have been labelled that have been kept on the S3 storage bucket with the key characteristics such as Network_Type, CPU_Usage (%), Battery_Level (%) and Memory_Usage (%). During the training it involves hyperparameter tweaking that was utilized to enhance the parameters like number of trees and the tree depth. With all the cross validation observed it is seen that random forest shows excellent predictions and validation that provides a stabilization between the recall and precision that are important factors helps in the reducing the delay in the simulation and utilization of energy.

Deployment of Endpoint: Followed by the training the trained model is to be deployed in the SageMaker as the endpoint that allows to coordinate the real time communication. iFogSim was able to send the task that will be used for the predictions, and it makes it accessible through the endpoint with the RESTful API. Due to the scalability offered by the SageMaker it is made possible by the endpoint that can efficiently manage the changing the workload that is most suitable for the real time applications.

5.2 iFogSim Simulation

The intelligent job offloading system in this research have been implemented and evaluated by utilization of the iFogSim simulation framework as a base. To enhance the existing functionality of the fog computing, this simulation was configured in a way that it can be incorporated with the trained model for the real time predictions.

Configuration of fog and edge devices: As a part of this simulation create the hierarchical fog devices that are the servers like cloud and edge devices. Metrics for the energy consumed, uplink, downlink and the bandwidth capacities and available computing resources such as storage, CPU and RAM must be described for each device. This setup with the fog computing is used in the real time systems. The keys features in that are the cloud and edge devices. In cloud devices that job that require the large processing power have been carried out by this centralized and high-performance computing resources. In edge devices the tasks that needs to executed fast with low latency that is very close to the data source.

Simulation of Task Offloading and Integration of Datasets: The required task data have been feed into the simulation from the pre-processed dataset that involves key characteristics like CPU utilization, memory needs and network latency. The smooth incorporation of the key characteristics during the runtime that ensured through the provided machine learning model that has been trained as shown in Figure 2.

Flow of Task execution: Through the simulation that deployed sensors produces the job that individual represents the computational demand that are required for each individual set of resources. The jobs that have been generated have be incorporated with the machine learning model that is used to predict the decision of the tasks whether to perform the tasks locally on edge that involves reduced latency or other offload to the cloud.

Making decisions using models: After the training the random forest model have been serialized into a pickle file that was incorporated into the simulation environment. With the utilization of the process builder the designed python scripts can run in the environment with java to help in the predictions.

Execution of Simulation - Functionality of Sensors and Actuators: To mimic the real time situations the sensors can produce the data tuples with the different transmission break that have been modelled by the uniform distribution. The fog computing continues to provide the results that can be responded through the actuators.

5.2 ML Model integration with the iFogSim for the Real time Simulation

The first and important step towards the process of enabling the offloading predictions in real time is with the integration of the trained machine learning model with the iFogSim. After the model training and implementation in the SageMaker the developed random forest model has been extracted and downloaded as a pickle file that have been passed into the simulation environment present in eclipse. Decision making process was made very easy and smooth through this integration that leads to enabling the iFogSim to utilize the developed model without relying on any other endpoints.

Serialization and Model Extraction: The trained the random forest model have been serialized into a pickle file into the simulation environment during the runtime. Python scripts have been utilized throughout the integration for the deserialization and the inference. The process builder has been utilized to ensure that the smooth transaction between the java based iFogSim environment and the python-based model training. The serialized model and the dataset have been given as the input to the python script. As a result, this will return the prediction about the decisions regarding the task offloading.

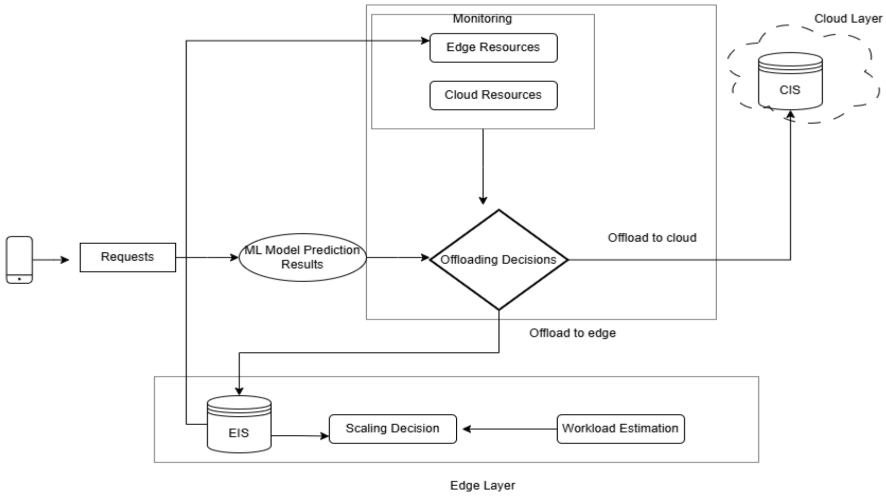


Fig. 3: iFogSim Task Offloading

Deserialization of the Model: With the utilization of the process builder, it helps to run the developed python scripts in the environment of java with the pickle file that was model.pkl that have been deserialized during the simulation. This method made it possible for the seamless communication between the iFogSim simulation environment and the python capabilities of the machine learning.

Real Time Predictions: The key input characteristics like CPU Utilization, availability of memory, latency of network and complexity of task that have been generated for the tasks during the simulation that was then pre-processed and then sent to the random forest that have been deserialized for the prediction. The developed model will return results in the binary choice that is 0 and 1. The response 0 indicates that the task should be performed locally in the edge device and the response 1 indicates that the task to be offloaded to the cloud for its processing and execution. Based on the predictions results each task will be assigned a specific place for the execution with the efficient management of the computational resources.

Integration of Workflow: The developed random forest model during the training will be able to predict the key characteristics in the dataset of the simulation that matches their respective features. This process involves choosing the appropriate features, encoding the categorical format values and ensures that not irrespective or missing values present in the dataset. To maintain the consistency and prevent any mistakes happening during the runtime, any specific features that are not required for decision making process have been eliminated during the preprocessing.

Implementing Decisions: The edge device that is edge-1 that will perform the assigned task that will be needed to perform locally which will result in the reduced latency with the proper utilization of the available resources. Rest of the tasks that can be offloaded to the cloud that has the higher processing capacity which will perform the tasks that requires additional network latency.

5.3 Simulation Results and the Output of the Deployed Endpoint on AWS

The output of the simulation has been displayed in the following metrics which will be discussed below that includes Decisions regarding task offloading it provides a detailed log of events related to the task and their related to decide the offloading decisions and the respective location of execution device that can be either cloud or edge. Execution Time is the amount of overall time that required for task during the simulation. Energy Consumption is the amount of energy consumed by assigned devices during the execution of task. Network Usage is the total requirement of the available bandwidth for the task offloading.

6. Results and Discussion:

6.1 AWS Sage Maker Model Evaluation

Model Results:

Metric	Logistic Regression	Random Forest	XGBoost
Accuracy	49.17%	53.33%	52.50%
Precision	53%	55%	55%
F1-Score	47%	46%	46%
Recall	52%	52%	52%

Challenges: All the trained models were providing low recall value for the class 1 for the tasks that needed to be offloaded which leads to difficulty in predictions of task offloading. Based on the results it has been observed that there should enhancement in the scaling of features, hyperparameter tuning that might lead to the increased prediction results.

Deployment: After the model training and analysis of the performance the model have been deployed in the SageMaker endpoints that are responsible for providing the real-time decision-making ability in the iFogSim environment. For the job offloading settings this deployment acts as a basic for continuous improvement and testing.

EDA (Exploratory Data Analysis): The Correlation matrix provides a weak dependency between the key features as shown in the Figure 4 that shows its independence. The offloading decision present from this based on the key features that rely on this correlation between the Available Memory and the data size, this relation values are too less in terms of predictions. Need to improve the correlation by the advancement of engineering towards the features.

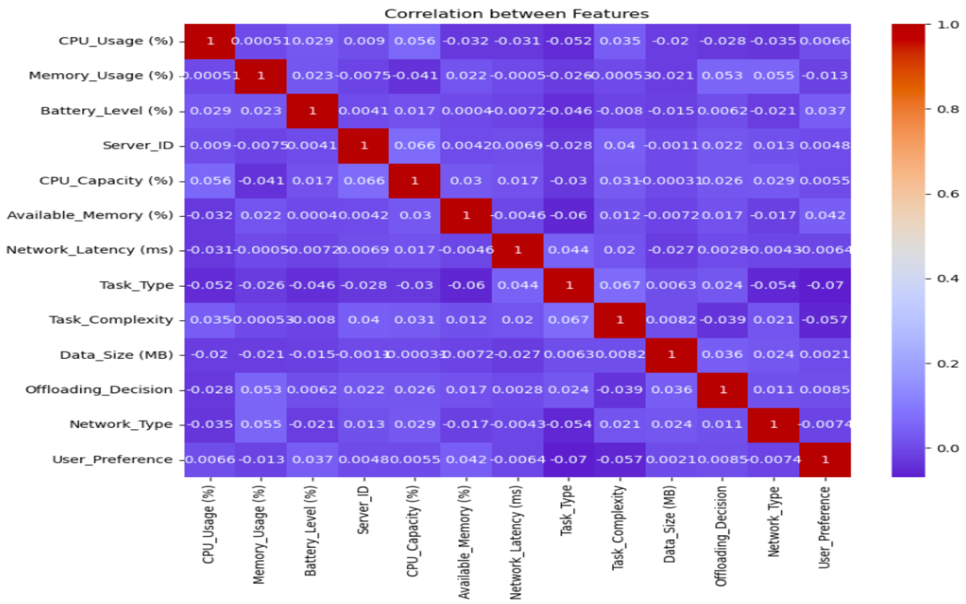


Fig. 4: Correlation Matrix

Confusion Matrix: The purpose of the confusion matrix is to provide a visual depiction with the comparison of the actual and predicted values of the performance of model. It holds the true positive and negative and false positive and negative values of each model. The figure 5 is for the logistic regression, figure 6 is for the random forest and figure 7 is for the XGBoost.

ROC Curve: ROC Curve is used to plot the trye positive rate versus the false positive rate that will indicate how well the model can perform according to different classification criteria. The ROC value achieved for all the model has been of 0.53 as shown in the figure 8.

Fig. 5: Logistic Regression

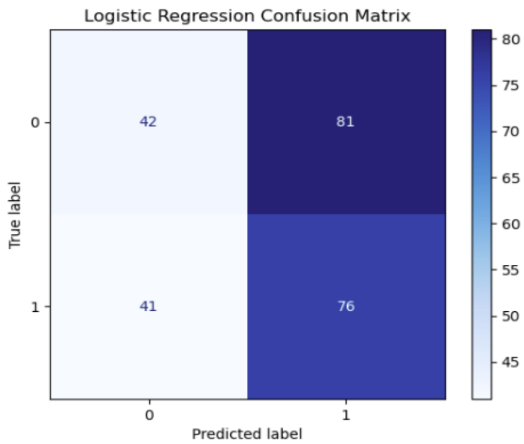
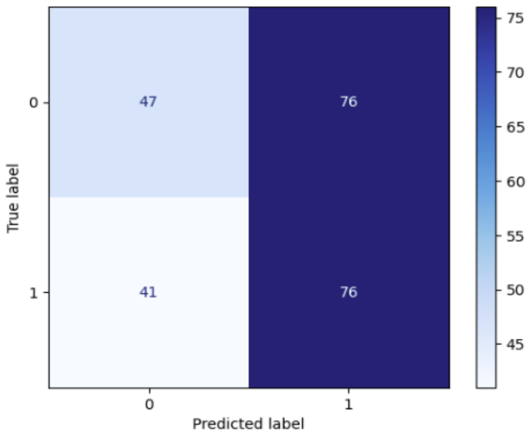


Fig. 6: Random Forest



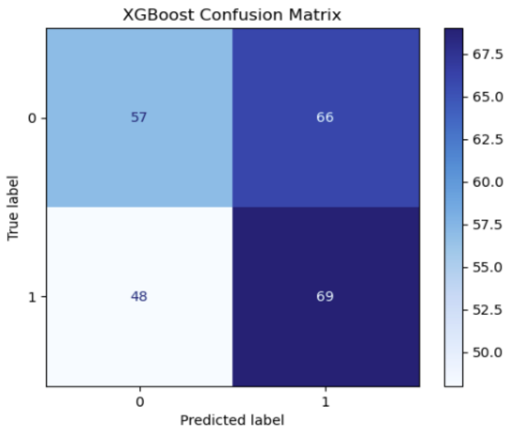


Fig. 7: XGBoost

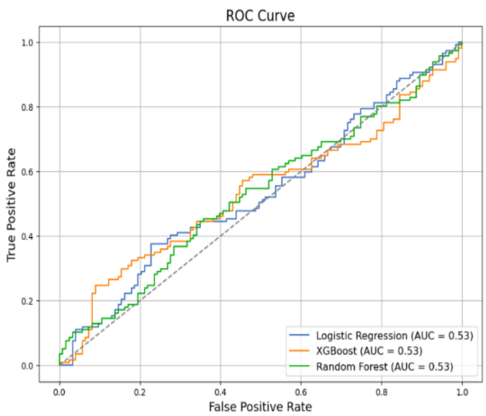
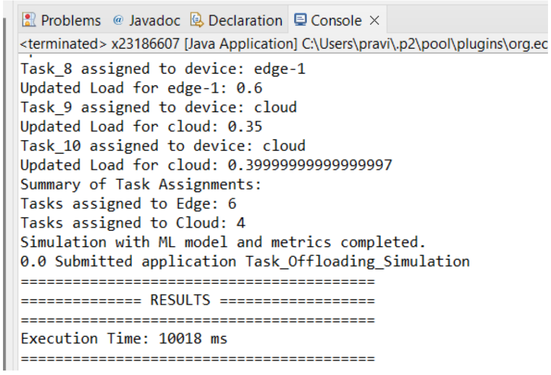


Fig. 8: ROC Curve

6.2 Experiment 2: iFogSim Simulation Results

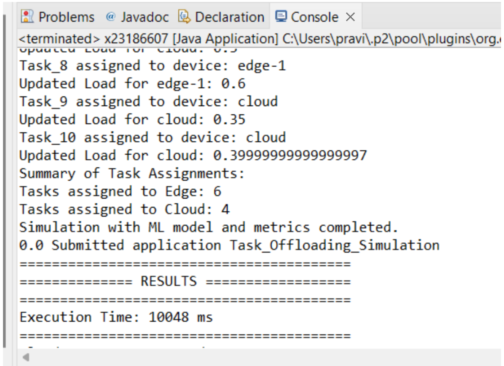
Results and Observations: The application includes the EEG data that have been used to process the sensor processor with the edge devices. Towards the data management and decision making the data have been routed towards the decision_maker and the central_data_node that process through the cloud services. The actuators are responsible for the task execution with the received signals. Based on the prediction results the task offloaded to cloud are 4 and the tasks offloaded to edge are 6. The load distribution of the tasks has been distributed based on edge to have 30 % utilized and the cloud to have 20 % utilized.

Module Placement with the cloud and Edge: The execution of the tasks happened in the cloud and the edge with the setting of the cloud equals to false and true and the results are being indicated with the execution time. When it is edge the controller will consumes these fogDevices, sensors, actuators, application and moduleMapping and the execution time is 10,018 ms as shown in fig 9. When the execution is on cloud the controller will consume these fogDevices, application and moduleMapping and the execution time is 10,048 ms as shown in fig 10.



```
Problems  Javadoc  Declaration  Console
<terminated> x23186607 [Java Application] C:\Users\pravi\.p2\pool\plugins\org.ec
Task_8 assigned to device: edge-1
Updated Load for edge-1: 0.6
Task_9 assigned to device: cloud
Updated Load for cloud: 0.35
Task_10 assigned to device: cloud
Updated Load for cloud: 0.39999999999999997
Summary of Task Assignments:
Tasks assigned to Edge: 6
Tasks assigned to Cloud: 4
Simulation with ML model and metrics completed.
0.0 Submitted application Task_Offloading_Simulation
========================================
============= RESULTS =================
========================================
Execution Time: 10018 ms
========================================
```

Fig. 9: Simulation Execution on edge



```
Problems  Javadoc  Declaration  Console
<terminated> x23186607 [Java Application] C:\Users\pravi\.p2\pool\plugins\org.
Task_8 assigned to device: edge-1
Updated Load for edge-1: 0.6
Task_9 assigned to device: cloud
Updated Load for cloud: 0.35
Task_10 assigned to device: cloud
Updated Load for cloud: 0.39999999999999997
Summary of Task Assignments:
Tasks assigned to Edge: 6
Tasks assigned to Cloud: 4
Simulation with ML model and metrics completed.
0.0 Submitted application Task_Offloading_Simulation
========================================
============= RESULTS =================
========================================
Execution Time: 10048 ms
========================================
```

Fig. 10: Simulation Execution on cloud

7. Conclusion and Future Work:

In this study it effectively describes how well the mobile edge cloud computing, and the machine learning can be combined to enhance the job offloading scenarios in various and resource limited environment. Real time task predictions on the offloading decisions have been achieved by the utilizing of the best machine learning methods like random forest, logistic regression and XGBoost by following with the further deployment of the trained model in the SageMaker. The trained machine learning model is then combined into the iFogSim Simulation environment to execute various tasks based on the system and features that could effectively simulates the real time scenarios. This suggested approach focus is to tackle the main issues faced in the MEC like excessive latency, wastage of resources and short duration of the mobile battery life. The results in this showed the enhancement in the system performance that includes the maximized resource utilization, low latency and good user experience. A scalable and effective task offloading approach have been introduced in this research that might help in the advancements in the field of MEC. The future work towards this research is employing more sophisticated machine learning techniques such as reinforcement learning and deep learning that has advance features by applying this, we can expect more accurate prediction results towards the task offloading.

References

Alqahtani, F., Al-Maitah, M., & Elshakankiry, O. (2022). A proactive caching and offloading technique using machine learning for mobile edge computing users. Computer Communications, 181,224-235.

Alsadie, D. (2024). Efficient Task Offloading Strategy for Energy-Constrained Edge Computing Environments: A Hybrid Optimization Approach. IEEE Access, 12, 85089-85102.

Bi, S., Huang, L., & Zhang, Y. J. A. (2020). Joint optimization of service caching placement and computation offloading in mobile edge computing systems. IEEE Transactions on Wireless Communications, 19(7), 4947-4963.

Guo, Y., Xu, X., & Xiao, F. (2024). MADRLOM: A Computation offloading mechanism for software defined cloud-edge computing power network. Computer Networks, 245, 110352.

Abbas, A., Raza, A., Aadil, F., & Maqsood, M. (2021). Meta-heuristic-based offloading task optimization in mobile edge computing. International Journal of Distributed Sensor Networks, 17(6),15501477211023021.

Liu, G., Dai, F., Huang, B., Qiang, Z., Wang, S. and Li, L., (2022). Dependency-aware task offloading for vehicular edge computing with end-edge-cloud collaborative computing.

Hao, Y., Jiang, Y., Chen, T., Cao, D., & Chen, M. (2019). iTaskOffloading: intelligent task offloading for a cloud-edge collaborative system. IEEE Network, 33(5), 82-88

Heidari, A., Navimipour, N. J., Jamali, M. A. J., & Akbarpour, S. (2023). A green, secure, and deep intelligent method for dynamic IoT-edge-cloud offloading scenarios. Sustainable Computing: Informatics and Systems, 38, 100859.

Huang, H., Zhan, W., Min, G., Duan, Z., & Peng, K. (2024). Mobility-aware computation offloading with load balancing in smart city networks using MEC federation. IEEE Transactions on Mobile Computing.

Huda, S. A., & Moh, S. (2022). Survey on computation offloading in UAV-Enabled mobile edge computing. Journal of Network and Computer Applications, 201, 103341.

Liu, J., Mi, Y., Zhang, X., & Li, X. (2024). Task graph offloading via deep reinforcement learning in mobile edge computing. Future Generation Computer Systems, 158, 545-555.

Nikougoftar, E. and Ghobaei-Arani, M., (2023). A Fuzzy Q-learning-based Approach for Autoscaling and Computation Offloading in Edge/Cloud Computing.

Mahenge, M.P.J., Li, C. and Sanga, C.A., (2022). Energy-efficient task offloading strategy in mobile edge computing for resource-intensive mobile applications. Digital Communications and Networks, 8(6), pp.1048-1058.

Peng, H., Zhang, X., Li, H., Xu, L., & Wang, X. (2023). An ai-enhanced strategy of service offloading for iov in mobile edge computing. Electronics, 12(12), 2719.

Saleh, H., Saber, W., & Rizk, R. (2022, April). Mobile computation offloading in mobile edge computing based on artificial intelligence approach: a review and future directions. In International Conference on Advanced Machine Learning Technologies and Applications (pp. 593-603). Cham: Springer International Publishing.

Tan, X., Zhao, D., Wang, M., Wang, X., Wang, X., Liu, W., & Ghobaei-Arani, M. (2024). A decision-making mechanism for task offloading using learning automata and deep learning in mobile edge networks. Heliyon, 10(1).

Maleki, E.F., Mashayekhy, L. and Nabavinejad, S.M., (2021). Mobility-aware computation offloading in edge computing using machine learning. IEEE Transactions on Mobile Computing, 22(1), pp.328-340.

Wang, P., Di, B., Song, L., & Jennings, N. R. (2022). Multi-layer computation offloading in distributed heterogeneous mobile edge computing networks. IEEE Transactions on Cognitive Communications and Networking, 8(2), 1301-1315.

Yang, S., Lee, G., & Huang, L. (2022). Deep learning-based dynamic computation task offloading for mobile edge computing networks. Sensors, 22(11), 4088.

Zeng, C., Wang, X., Zeng, R., Li, Y., Shi, J., & Huang, M. (2024). Joint optimization of multidimensional resource allocation and task offloading for QoE enhancement in Cloud-Edge-End collaboration. Future Generation Computer Systems, 155, 121-131.

Zhou, H., Zhang, Z., Wu, Y., Dong, M., & Leung, V. C. (2022). Energy efficient joint computation offloading and service caching for mobile edge computing: A deep reinforcement learning approach. IEEE Transactions on Green Communications and Networking, 7(2), 950-961.

Zhu, X., Jia, Z., Pang, X., & Zhao, S. (2024). Joint Optimization of Task Caching and Computation Offloading for Multiuser Multitasking in Mobile Edge Computing. Electronics, 13(2), 389.

Farahbakhsh, F., Shahidinejad, A., & Ghobaei-Arani, M. (2023). Context-aware computation offloading for mobile edge computing. Journal of Ambient Intelligence and Humanized Computing,14(5), 5123-5135.

Chen, Z., Hu, J., Chen, X., Hu, J., Zheng, X., & Min, G. (2020). Computation offloading and task scheduling for DNN-based applications in cloud-edge computing. IEEE Access, 8, 115537-115547.

Zhang, S., Yi, N. and Ma, Y., 2024. A survey of computation offloading with task types. IEEE Transactions on Intelligent Transportation Systems.

Dong, S., Tang, J., Abbas, K., Hou, R., Kamruzzaman, J., Rutkowski, L. and Buyya, R., 2024. Task offloading strategies for mobile edge computing: A survey. Computer Networks, p.110791.

Qin, W., Chen, H., Wang, L., Xia, Y., Nascita, A. and Pescapè, A., 2024. MCOTM: Mobility-aware computation offloading and task migration for edge computing in industrial IoT. Future Generation Computer Systems, 151, pp.232-241.

Yin, L., Guo, S. and Jiang, Q., 2024. Joint Task Allocation and Computation Offloading in Mobile Edge Computing with Energy Harvesting. IEEE Internet of Things Journal.