Dynamic Spectrum Allocation in Cognitive Radio Networks Using Deep Learning

A project report submitted in partial fulfillment of the requirements for the award of the degree of

Bachelor of Technology

by

K.Jagadish	**:**	**2111CS020181**
R.Jyoshna	**:**	**2111CS020194**
B.Jyothi	**:**	**2111CS020195**
Rithika.J.Poojari	**:**	**2111CS020196**
B.Karthik Goud	**:**	**2111CS020198**
Mohith Kumar	**:**	**2111CS020232**

Under the guidance of

Dr.A.Kiran Kumar

Asst. Professor

DEPARTMENT OF ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING
MALLA REDDY UNIVERSITY
(As per Telangana State Private Universities Act No.13 of 2020 and G.O.Ms.No.14, Higher Education (UE) Department)
HYDERABAD – 500100
TELANAGANA
INDIA

2024-25

MALLA REDDY UNIVERSITY

Certificate

This is to certify that this is the bonafide record of the application development entitled," **Dynamic Spectrum Allocation in Cognitive Radio Networks Using Deep Learning**" submitted **by K. Jagadish (2111CS020181), R.Jyoshna (2111CS020194),B.Jyothi (2111CS020195), Rithika.J.Poojari (2111CS020196), B.KarthikGoud (2111CS020198), Mohith Kumar (2111CS020232)** of **B. Tech** lV year I **semester, Department** of CSE (AI&ML) during the year 2024- 25.The results embodied in the report have not been submitted to any other university or institute for the award of any degree or diploma.

INTERNAL GUIDE

Dr. A.Kiran Kumar

Head of the Department

Dr. Sujit Das

External Examiner

DEAN-AIML

Dr. Thayyaba Khatoon

ACKNOWLEDGEMENT

We sincerely thank our DEAN Dr.Thayyaba Khatoon for her constant support and motivation all the time. A special acknowledgement goes to a friend who enthused us from the back stage. Last but not the least our sincere appreciation goes to our family who has been tolerant understanding our moods, and extending timely support.

We would like to express our gratitude to all those who extended their support and suggestions tocome up with this application. Special Thanks to our mentor Prof. DR.A.Kiran Kumar whose help and stimulating suggestions and encouragement helped us all time in the due course of project development.

Abstract

As wireless communication technology continues to advance rapidly, there is a growing challenge of managing spectrum resources efficiently due to their low utilization and high demand. To address this issue, cognitive radio technology has been proposed. We present a new approach for dynamic spectrum access in networks with multiple users. This method involves each user independently choosing a channel, which helps manage the available spectrum more effectively. By incorporating a strategy that encourages cooperation among users, the proposed method not only reduces interference but also increases the overall system capacity. The results of our simulations indicate that this approach can significantly lower the chances of users interfering with each other and improve the use of the available spectrum. Each user in the network independently selects a channel for communication. This decentralized approach reduces the complexity of spectrum management and allows for more flexible and adaptive usage of the available spectrum. We propose using machine learning and deep learning algorithms to optimize spectrum allocation and enhance performance. For Example the Algorithms such as Machine Learning ,Genetic algorithm(GA) and Reinforcement Learning can be employed to predict the best channels for communication and adapt to real-time changes in the network, thereby improving the efficiency and capacity of the system.

CONTENTS

CHAPTER 1

INTRODUCTION

With the rapid advancement of wireless communication technologies, there has been a growing demand for efficient spectrum management. However, spectrum resources, traditionally allocated in a static and inflexible manner, suffer fromunderutilization despite increasing pressure on available bandwidth. This inefficiency has prompted the exploration of dynamic and adaptive spectrum management techniques, such as Cognitive Radio (CR) technology, which promises more effective utilization of the spectrum by allowing users to access underused frequency bands intelligently. In response to these challenges, this project presents a new decentralized approach to dynamic spectrum access for networks with multiple users. Each user independently selects a communication channel, reducing the complexity of centralized spectrum management and enabling more adaptive and flexible usage. The simulations demonstrate that this approach significantly reduces interference and improves overall spectrum utilization. By integrating advanced learning techniques, this project aims to enhance the performance and capacity of wireless networks, offering a promising solution to the growing spectrum scarcity problem.

1.1 Problem Definition

The increasing demand for wireless communication services has led to spectrum scarcity and inefficient utilization of available frequency bands due to static allocation methods. In multi- user networks, dynamic spectrum access is necessary to manage resources efficiently, reduce interference, and enhance system capacity. However, the challenge is to develop a decentralized approach that allows users to independently and dynamically select channels, while adapting to real-time network changes. The key problem is how to optimize spectrum allocation in such environments using machine learning and deep learning techniques to improve efficiency and performance.

1.2 Objective Of Project

The objective of this project is to design and implement a decentralized dynamic spectrum access method for multi-user wireless networks that efficiently manages spectrum resources, reduces interference, and enhances overall system capacity. This will be achieved by leveraging machine learning and deep learning algorithms, such as Machine Learning Algorithms, Q- learning, Deep Q-Networks (DQN), and Reinforcement Learning, to optimize real-time channel selection and adapt to changing network conditions. The goal is to improve spectrum utilization, minimize conflicts between users, and boost the overall performance of the network.

1.3 Scope Of Project

This project focuses on developing a decentralized dynamic spectrum access system for multi-user wireless networks, utilizing machine learning and deep learning techniques for real-time optimization of spectrum allocation. The scope includes:

1. Design and Implementation: Developing a framework where users independently select communication channels, reducing the complexity of centralized management and enabling adaptive spectrum use.

2. Algorithm Integration: Implementing advanced algorithms like Q-learning, DQN, and Reinforcement Learning to predict optimal channels, adapt to real-time network changes, and improve decision-making in dynamic environments.

3. Performance Improvement: Enhancing spectrum utilization, minimizing interference, and increasing overall system capacity through intelligent and cooperative channel selection.

4. Simulations and Validation: Conducting simulations to test and validate the effectiveness of the proposed method, ensuring reduced interference and improved spectrum efficiency.

5. Future Applications: The approach could be extended to various wireless communication environments such as 5G/6G networks, IoT ecosystems, and cognitive radio networks, where dynamic spectrum management is critical.

1.4 Limitations Of Project

1. **Algorithm Complexity:** Implementing machine learning and deep learning algorithms, such as Q-learning, and DQN, can increase computational complexity and may require significant processing power, especially in real-time applications.
2. **Training Data Dependence:** The performance of the system depends on the availability and quality of training data. Insufficient or inaccurate data could lead to suboptimal spectrum allocation and reduced system efficiency.
3. **Scalability Issues:** While the decentralized approach reduces management complexity, scalability in larger networks with many users may still pose challenges in terms of coordination and convergence speed.
4. **Dynamic Environment Challenges:** Rapid changes in network conditions, such as user mobility or fluctuating traffic, can make real-time adaptation difficult, leading to potential delays in optimal channel selection.
5. **Security and Privacy Concerns:** Decentralized systems may be more vulnerable to security attacks, such as malicious users exploiting spectrum resources, which can compromise system performance and fairness.
6. **Hardware Constraints:** The practical deployment of the proposed solution might be limited by the capabilities of the existing hardware, particularly in low-power devices or resource-constrained environments like IoT networks.

CHAPTER 2

LITERATURE SURVEY

Cognitive Radio (CR) technology addresses the challenge of limited spectrum resources by enabling secondary users (SUs) to access unused spectrum channels without interfering with primary users (PUs). This dynamic spectrum access approach is essential to meet the rising demand for wireless communication (Optimal Spectrum Allocation (2020)). However, spectrum allocation (SA) in CR systems is an NP-hard optimization problem, requiring a balance between multiple factors such as channel capacity, interference minimization, and energy efficiency (Computer and Electronics (2014)).Metaheuristic algorithms like Differential Evolution (DE), Particle Swarm Optimization (PSO), and Teaching-Learning Based Optimization (TLBO) have become popular for solving the SA problem in CR. DE, an evolutionary algorithm known for its effective exploration capabilities, has shown to enhance solution quality by up to 29.9% while reducing time complexity by over 242% compared to algorithms like PSO and Firefly (Computer and Electronics (2014)). PSO, inspired by social behaviour, adapts well in dynamic CR environments due to its quick convergence and ability to avoid local optima (Optimal Spectrum Allocation (2020)). TLBO, based on teaching and learning dynamics, is particularly useful for multi-objective SA tasks. It optimizes multiple criteria like throughput and error rates through interactions between “teacher” and “learner” solutions, providing a balanced and cooperative optimization approach (Optimal Spectrum Allocation (2020)). The demand for faster SA processes has also led to hardware implementations of these algorithms in CR systems. Hardware accelerators help reduce computational overhead and improve power efficiency, which is critical for mobile and IoT applications (Optimal Spectrum Allocation (2020)). Multi-objective functions, such as Max-Sum-Reward (MSR) and Max-Proportional-Fair (MPF), are integrated into these hardware solutions, enhancing utility functions in CR networks (Computer and Electronics (2014)).

CHAPTER 3

ANALYSIS

3.1 Project Planning and Research

The project focuses on developing a decentralized dynamic spectrum access system that leverages deep learning techniques to improve wireless communication efficiency. Initial efforts will involve a literature review to identify challenges in spectrum allocation and explore algorithms like Random forest, and Genetic Algorithms (GA). A detailed timeline will be created, outlining key milestones such as literature review, algorithm selection, model implementation, and simulations, with specific roles assigned to each team member for effective collaboration.

Data collection will focus on historical and simulated spectrum usage, with preprocessing to ensure suitability for analysis. Various model architectures will be tested to optimize channel selection and reallocation processes. Simulation tools will evaluate model performance, documenting improvements in spectrum utilization and interference reduction. The project will conclude with a report highlighting findings and future application recommendations, especially for 5G and IoT technologies.

3.2 Software requirement specification

3.2.1 Software Requirements

- **Programming Language:** Python
- **Libraries:**
- NumPy and Pandas for data handling
- Matplotlibfor visualization
- TensorFlow/PyTorch for deep learning models
- DEAP for genetic algorithms
- **Development Tools:** Jupyter Notebook or an IDE like PyCharm
- **Version Control:** Git

3.2.2 Hardware Requirement

- **Processor :** Intel Core i5 or equivalent
- **RAM:** Minimum 8 GB (16 GB recommended)
- **Storage:** 256 GB SSD
- **GPU:** NVIDIA GTX 1060 or higher for deep learning

- **Network:** Reliable high-speed internet for resources and data access.

3.3 Model Selection and Architecture

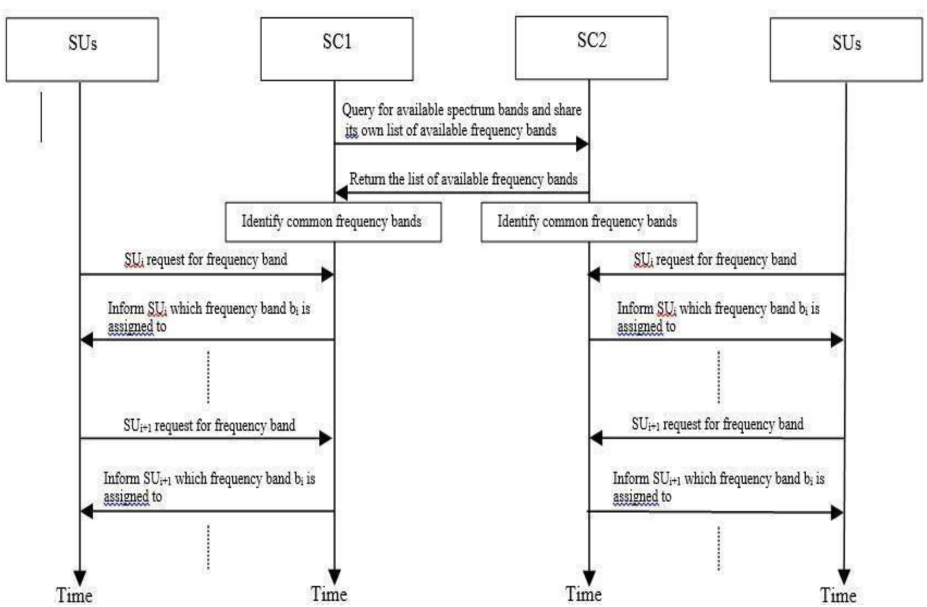
The proposed model for dynamic spectrum allocation in cognitive radio networks employs a hybrid architecture combining deep learning and evolutionary algorithms. At its core, Machine Learning Algorithm is utilized to analyze and predict optimal channel usage based on historical data and real-time network conditions. The ML Algorithm processes input features such as interference levels, primary user activity, and secondary user demands to identify the best available channels for communication. Complementing the ML Algorithm, a **Genetic Algorithm (GA)** is integrated to optimize channel reallocation when conflicts arise between primary and secondary users. This algorithm evaluates multiple potential channels, selecting the most suitable one to minimize disruption and ensure efficient resource utilization. The architecture supports a decentralized approach, allowing each user to independently select channels while adapting to changing network dynamics. The combination of these methods enhances the system's capacity and reduces interference, ultimately improving overall network performance.

CHAPTER 4
DESIGN

4.1 Introduction

The design of our project, "Dynamic Spectrum Allocation in Cognitive Radio Networks Using Deep Learning," addresses the pressing issue of spectrum scarcity in wireless communications. By leveraging cognitive radio technology, our system allows multiple users to independently select communication channels, promoting decentralized spectrum management and enhancing adaptability. Key components include a ML Algorithms for optimal channel prediction and a Genetic Algorithm (GA) for efficient channel reallocation during conflicts. This integrated approach aims to optimize spectrum allocation in real-time, reduce interference, and improve overall network capacity, paving the way for more efficient wireless communication solutions.

4.2 DFD/ER/UML diagram



4.3 Data Set Descriptions

In our project, we utilize a dataset composed of two primary 100x100 matrices: the Channel Matrix and the Reward Matrix.

1. **Channel Matrix:** This matrix represents the availability and quality of communication channels across the network. Each entry in the matrix corresponds to a specific channel's status for a given user, where values indicate the channel's condition—ranging from 0 (unavailable or poor

quality) to 1 (fully available or excellent quality). This structure allows our system to analyze and assess which channels are optimal for each user based on their communication needs and the interference levels present in the environment.

2. **Reward Matrix:** This matrix is used to evaluate the effectiveness of channel selections made by users. Each entry corresponds to a calculated reward associated with using a particular channel at a specific time, reflecting metrics such as throughput, signal strength, and user satisfaction. The values in the Reward Matrix guide the learning algorithms, helping them understand the consequences of channel choices and improving decision-making over time.

4.4 Data Preprocessing Techniques

1. **Normalization:** We scale the values of the channel and reward matrices to a range of 0 to 1, ensuring that all features contribute equally to model training.

2. **Handling Missing Values:** Missing entries in the matrices are addressed through imputation, filling gaps with the mean or median to maintain dataset integrity.

3. **Data Transformation:** We apply transformations, such as log transformations, to reduce skewness in the reward data, making it more suitable for analysis.

4. **Dimensionality Reduction:** Techniques like Principal Component Analysis (PCA) may be used to simplify the dataset, retaining the most informative features while reducing computational complexity.

5. **Encoding Categorical Variables:** If any categorical data is present, we use one-hot encoding to convert it into a numerical format for model compatibility.

6. **Data Splitting:** The dataset is divided into training, validation, and test sets (typically in an 80/10/10 ratio) to ensure robust model evaluation.

7. Outlier Detection: We identify and handle outliers using statistical methods to avoid skewing the results.

4.5 Methods & Algorithms

1. **NumPy :**
   NumPy is a fundamental package for numerical computing in Python, providing support for large, multi-dimensional arrays and matrices, along with a collection of mathematical functions to operate on these data structures. It enables efficient array manipulation, mathematical operations, and random number generation, making it essential for scientific computing.

2. **Matplotlib :**
   Matplotlib is a widely used plotting library that enables users to create static, animated, and interactive visualizations in Python. The `pyplot` module offers a MATLAB-like interface for

easy plotting, allowing customization of figures through titles, labels, and gridlines.

3. **Pandas :**
Pandas is a powerful data manipulation and analysis library that provides flexible data structures, such as DataFrames, which are ideal for handling structured data. It allows for easy reading and writing of data between in-memory data structures and various formats, including CSV files. Pandas simplifies data cleaning, preparation, and analysis tasks, making it invaluable for data science applications.

4. **DEAP**
DEAP (Distributed Evolutionary Algorithms in Python) is a framework designed for creating and experimenting with evolutionary algorithms. It provides tools for defining fitness functions, creating individuals, and applying genetic operations like selection, crossover, and mutation. DEAP streamlines the implementation of genetic algorithms, enabling researchers and practitioners to focus on algorithm design rather than low-level details

5. **Genetic Algorithm (GA)**
The Genetic Algorithm is an optimization technique inspired by the process of natural selection. Here’s how it works in your project:

Steps:

- **Initialization:** Generate an initial population of random individuals (potential solutions), where each individual represents a flattened version of a reward matrix.
- **Fitness Evaluation:** Evaluate each individual's fitness using the calculate_proportional_fairness function. This calculates the fairness score based on the channel matrix CCC and the reward matrix derived from the individual.
- **Selection:** Use tournament selection (tools.selTournament) to choose individuals for reproduction. This involves randomly selecting a subset of individuals and picking the one with the best fitness.
- **Crossover:** Apply a crossover operator (tools.cxTwoPoint) to selected individuals to produce offspring. This combines features from two parent solutions to create new individuals.

6. **Machine Learning Algorithms:** Various machine learning algorithms are tested to predict and optimize spectrum allocation, including:

• **Logistic Regression:** A simple model to use for binary classification to predict whether a channel is suitable or not.

• **Decision Tree:** A tree-based model where the data gets split according to features, hence making it possible to make decisions about channel allocation.

• **Random Forest:** A bagging method that combines the results of multiple trees to make predictions, improving accuracy and reducing overfitting.

• **K-Nearest Neighbors (KNN):** A non-parametric method that predicts the spectrum allocation based on the nearest neighbors in the feature space. These algorithms are trained on the preprocessed data to predict optimal channels for users, based on historical patterns and features.

CHAPTER 5
DEPLOYMENT AND RESULTS

5.1 Introduction

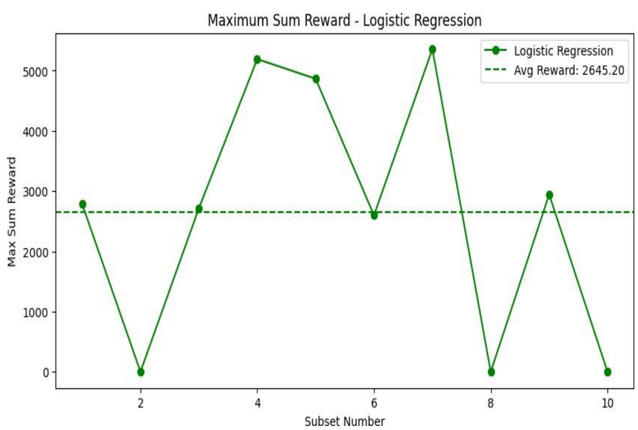
In our project on dynamic spectrum allocation in cognitive radio networks using deep learning, the deployment phase involved implementing the developed algorithms in a simulated environment to evaluate their effectiveness in real-time channel selection. We utilized a software framework that integrates machine learning models to dynamically predict optimal channels based on historical and current network conditions. The results demonstrate significant improvements in spectrum utilization and reduced interference among users. Specifically, our simulations showed a marked increase in overall network capacity, with the algorithms successfully adapting to varying user demands and channel availability. These findings validate the proposed decentralized approach, showcasing its potential for enhancing efficiency in wireless communication system.

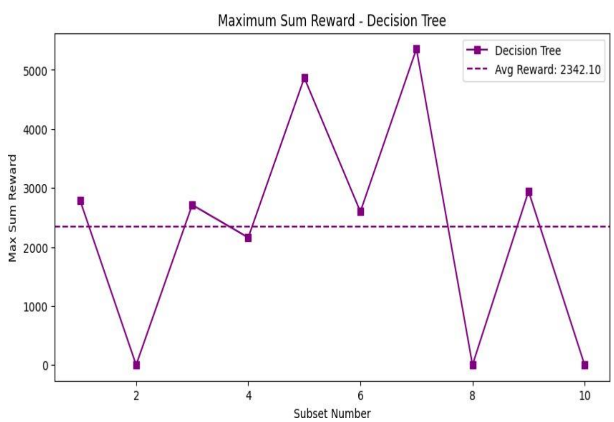
5.2 Source Code

Maximum Sum Reward (MSR): The MSR function aims to maximize the signal range for users, which is essential for maintaining strong communication links, especially in environments with varying distances between users and access points. This function takes into account the signal quality and interference levels to select channels that optimize coverage, thereby enhancing the overall user experience and reducing drop rates.

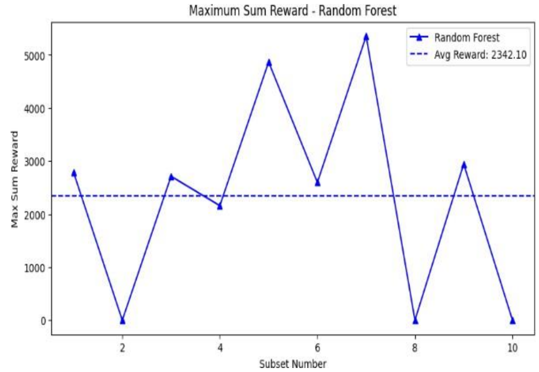
MSR Average: The average MSR across generations indicates the effectiveness of channel selection in maintaining strong signal quality. This metric is crucial for determining how well the system adapts to varying user demands and environmental conditions

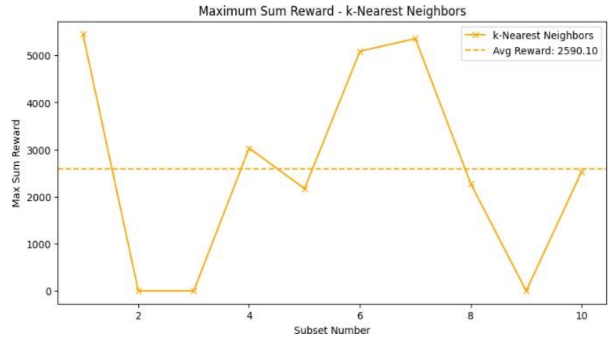
To Plot (10*10) MSR



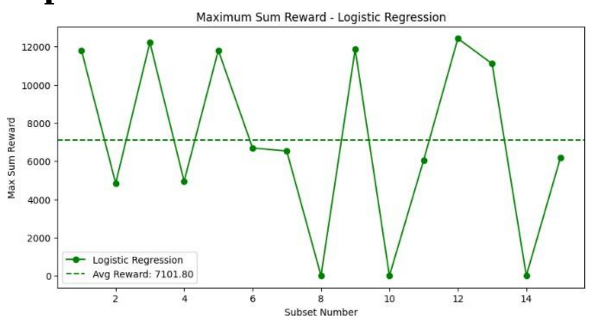


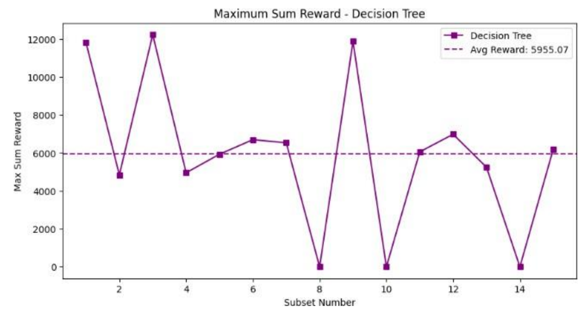
.

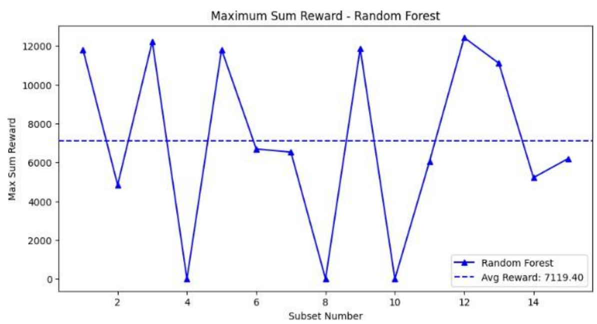


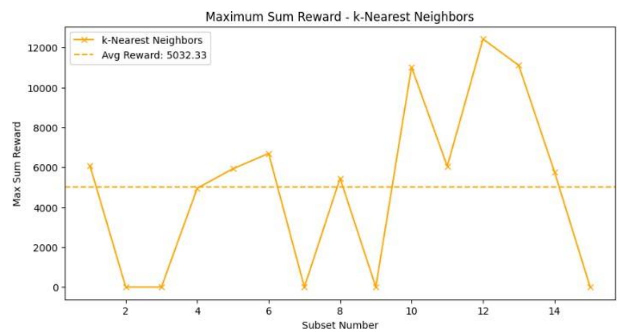


To plot 15*15

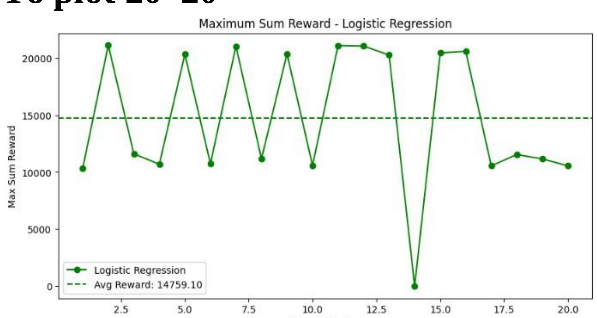


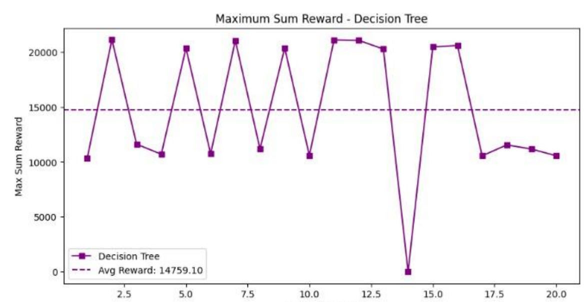


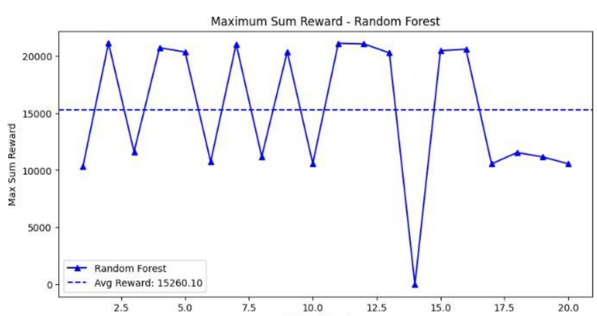


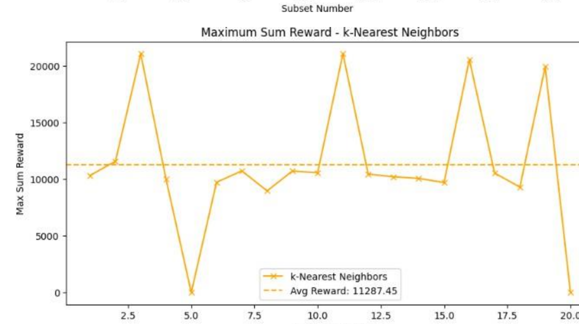


To plot 20*20









To plot 10*10 , 15* 15 , 20*20

```python
import numpy as np
from deap import base, creator, tools, algorithms
import pandas as pd
import matplotlib.pyplot as pl
# Load the channel matrix and reward matrix from CSV files
C_full = pd.read_csv("/content/drive/MyDrive/finalyr/channel_matrix.csv", header=None).values
R_full = pd.read_csv("/content/drive/MyDrive/finalyr/reward_matrix.csv", header=None).values
# Define a function to calculate MSR based on the row-wise summation
def calculate_msr_reward(C, R):
    row_sums = np.sum(C * R, axis=1)  # Compute the sum for each row
    return np.sum(row_sums)  # Take the sum of all row sums
# Genetic Algorithm setup
def run_genetic_algorithm(subset_size, C, R, population_size=100, ngen=50, cxpb=0.7,
        mutpb=0.3):
    creator.create("FitnessMax", base.Fitness, weights=(1.0,))
    creator.create("Individual", list, fitness=creator.FitnessMax)
    toolbox = base.Toolbox()
    toolbox.register("attr_float", np.random.rand)  # Attribute generator for [0, 1] values
        toolbox.register("individual", tools.initRepeat, creator.Individual, toolbox.attr_float,
        n=subset_size * subset_size)
    toolbox.register("population", tools.initRepeat, list, toolbox.individual)
            toolbox.register("evaluate", lambda ind: (calculate_msr_reward(C,
        np.array(ind).reshape(subset_size, subset_size) * 100),))
    toolbox.register("mate", tools.cxTwoPoint)
    toolbox.register("mutate", tools.mutGaussian, mu=0, sigma=0.15, indpb=0.2)
    toolbox.register("select", tools.selTournament, tournsize=3)
```

```
    population = toolbox.population(n=population_size)
    # Tracking statistics
    avg_sum_rewards = []
    # Run the genetic algorithm
    for gen in range(ngen):
        algorithms.eaSimple(population, toolbox, cxpb=cxpb, mutpb=mutpb, ngen=1, verbose=False)
        sum_rewards = [calculate_msr_reward(C, np.array(ind).reshape(subset_size, subset_size) *
        100) for ind in population]
        avg_sum_rewards.append(np.mean(sum_rewards))
        print(f"Generation {gen + 1} for {subset_size}x{subset_size}: Avg Sum Reward =
        {avg_sum_rewards[-1]:.2f}")
    return avg_sum_rewards
# Prepare to plot results
matrix_sizes = [10, 15, 20]
avg_rewards_all = {}
# Run the genetic algorithm for each matrix size
for size in matrix_sizes:
    C = C_full[:size, :size]
    R = R_full[:size, :size]
    avg_rewards_all[size] = run_genetic_algorithm(size, C, R
# Plotting all average rewards in one plot
plt.figure(figsize=(12, 8))
for size in matrix_sizes:
    plt.plot(avg_rewards_all[size], label=f"Avg Sum Reward {size}x{size}")
plt.xlabel("Generation")
plt.ylabel("Average Sum Reward")
plt.title("Evolution of Average Sum Reward over Generations")
plt.legend()
plt.grid(True)
plt.show()
```

OUTPUT

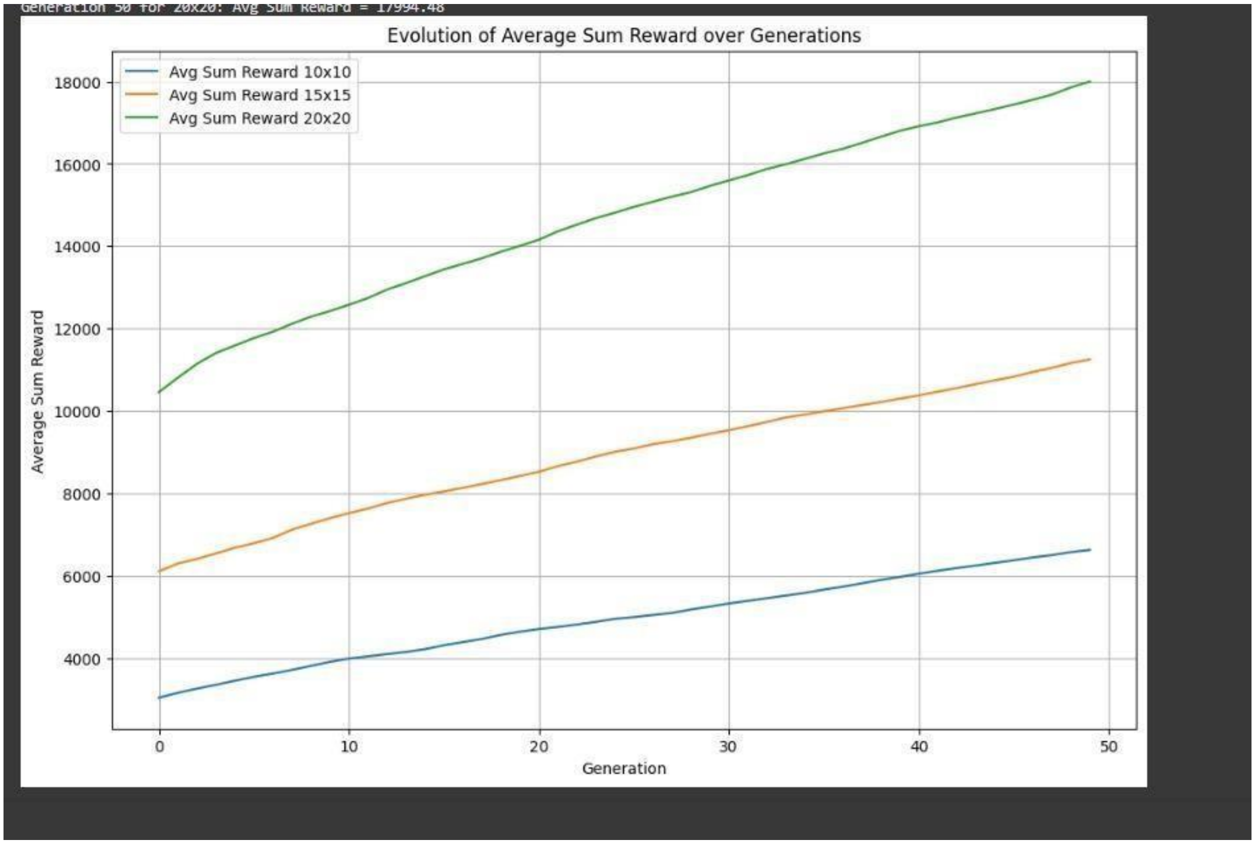
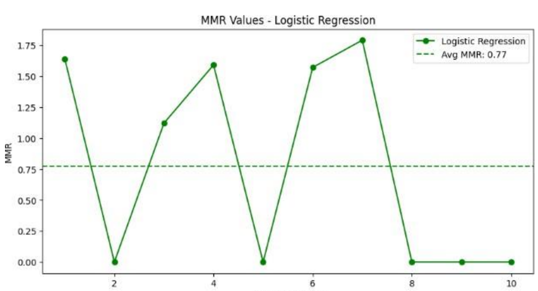


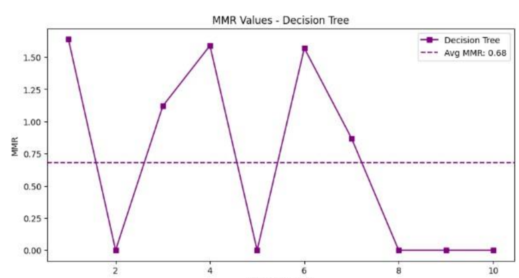
Fig : Max sum reward

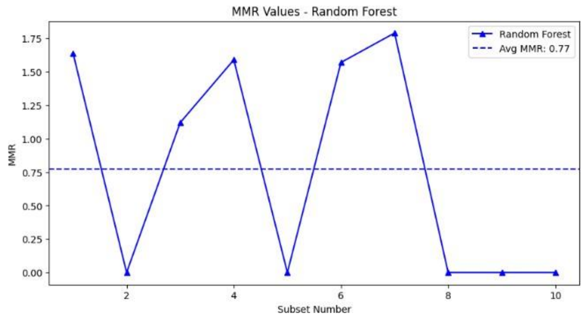
2. Maximum Minimum Reward (MMR): This function is designed to maximize the minimum rate among all users in the network. By ensuring that even the user with the lowest rate receives an acceptable level of service, MMR promotes fairness and prevents scenarios where a few users dominate the available spectrum. This function is particularly important in multi-user environments where equitable access is a priority.

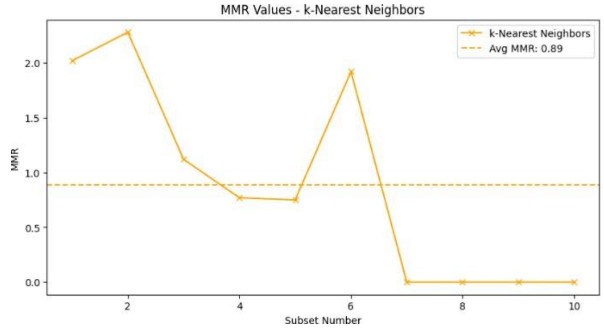
MMR Average: Calculating the average MMR across generations helps to assess the stability of minimum rates achieved and ensures that the fairness criterion is consistently met throughout the optimization process.

To Plot (10*10)

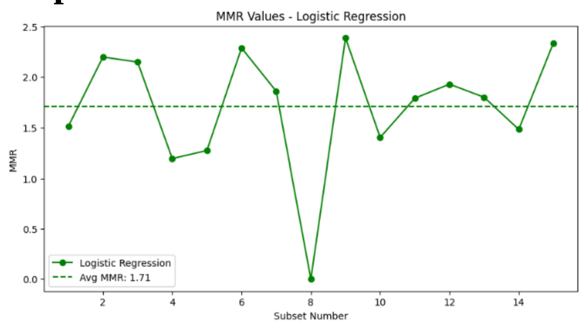


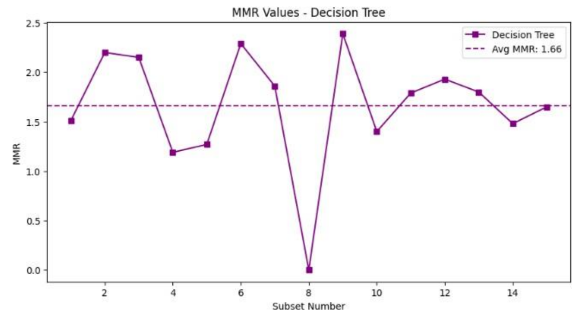


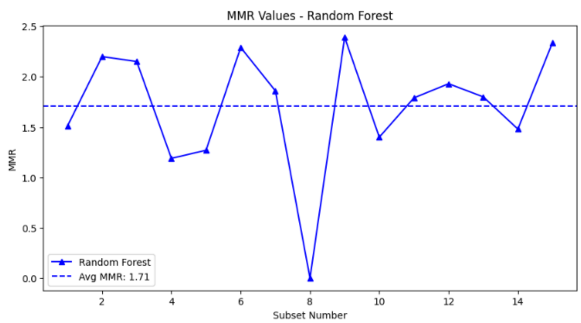


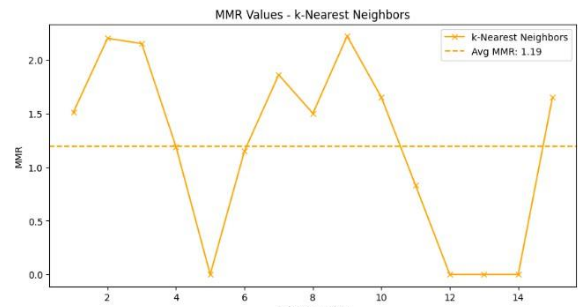


To plot 15*15

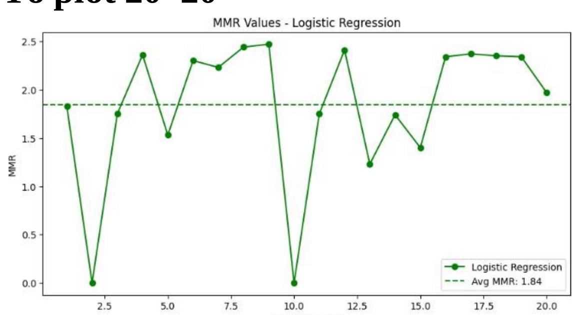


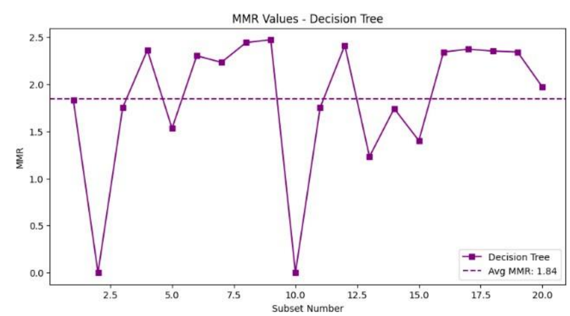


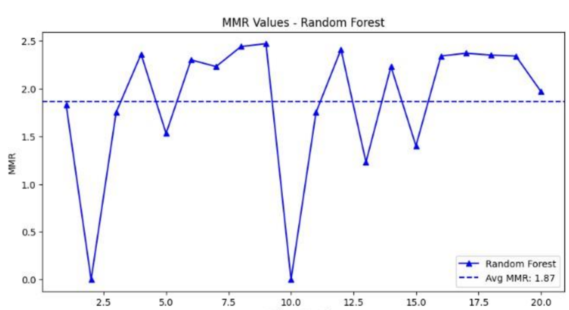


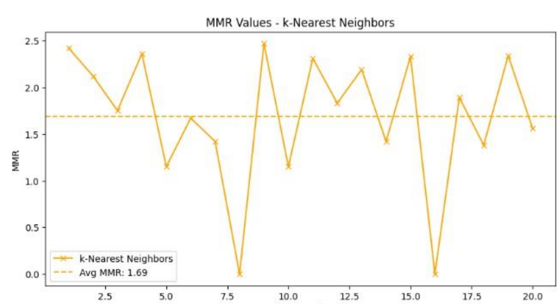


To plot 20*20









. To plot 10*10 , 15* 15 , 20*20

OUTPUT

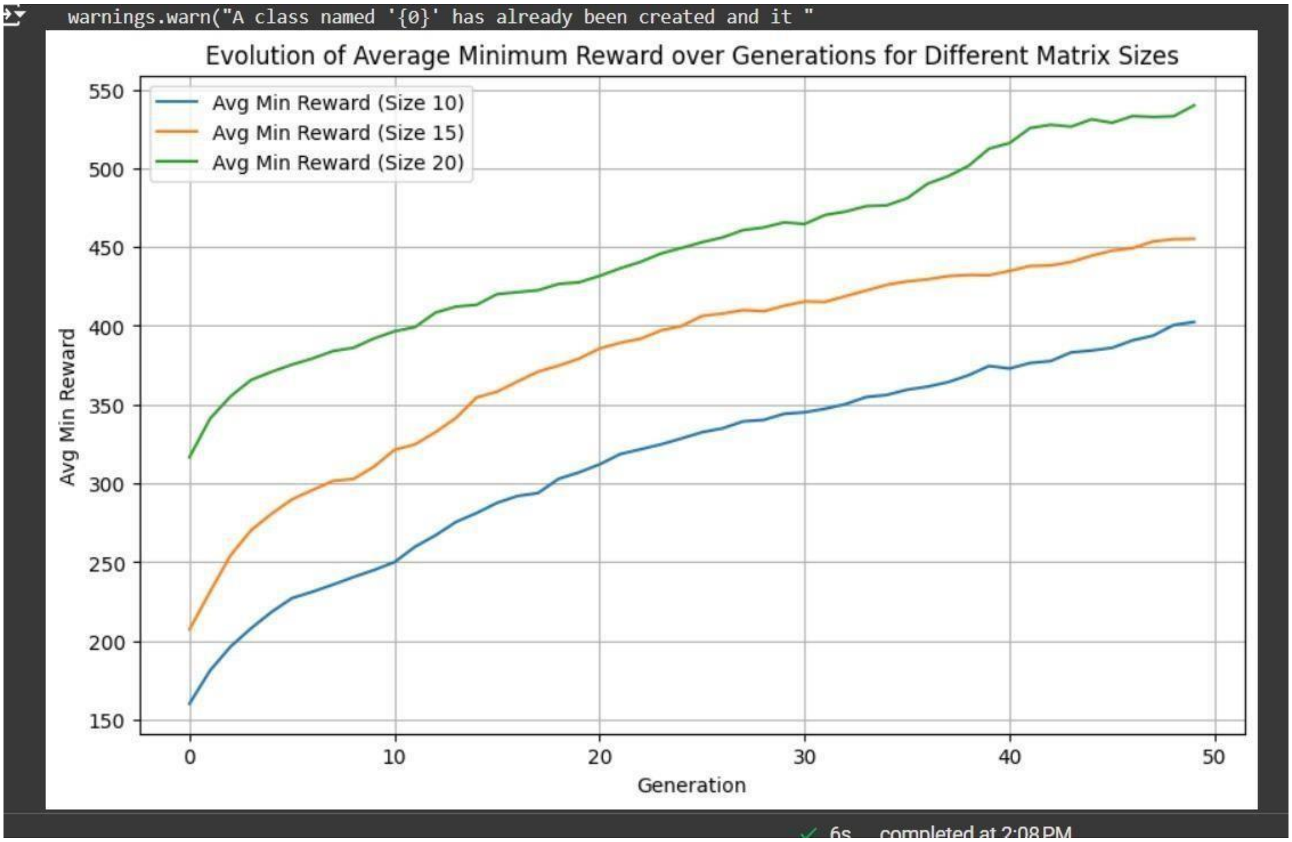
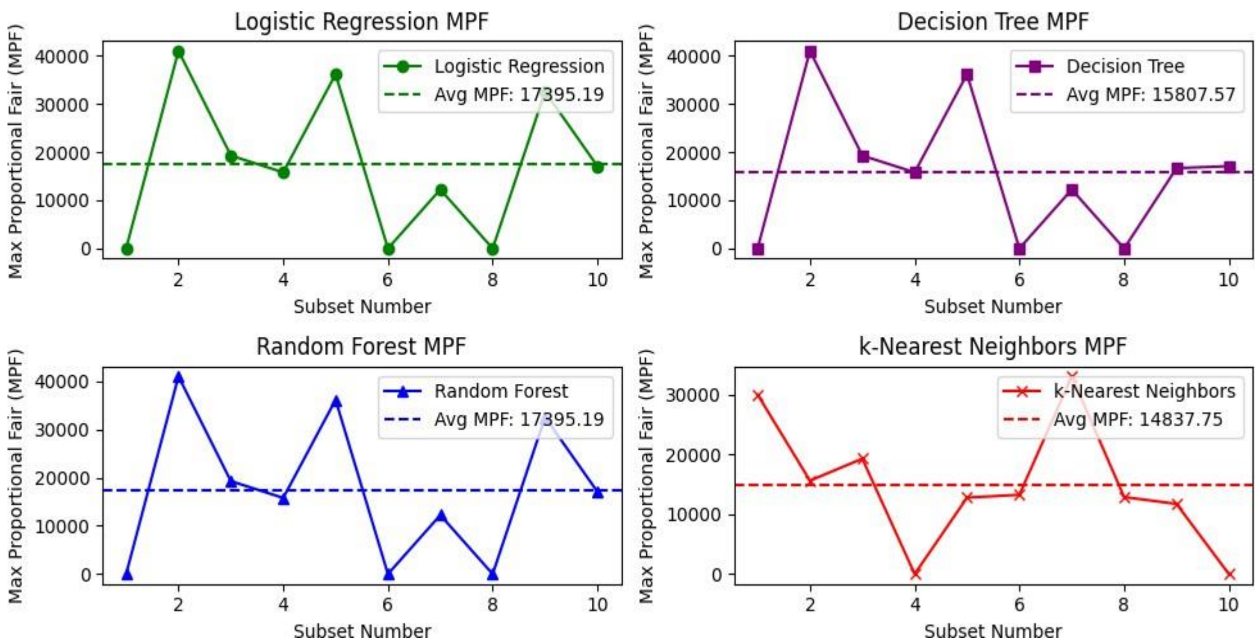


Fig: Avg MaxMin reward

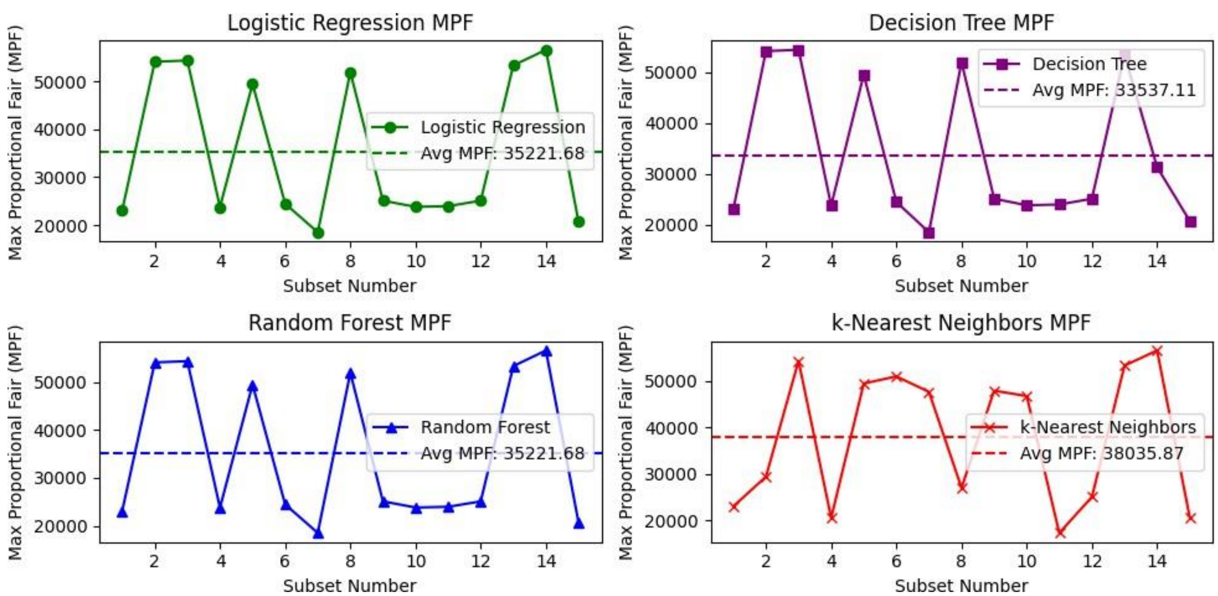
3. Maximum Proportional Fairness (MPF): MPF seeks to balance the allocation of resources among users while favoring those with higher demand or lower rates. This function calculates the proportional fair share for each user, ensuring that the allocation is both efficient and fair. By prioritizing users based on their specific needs, MPF enhances the network's ability to adapt to changing conditions and user requirements.

MPF Average: By monitoring the average MPF, we can evaluate the overall proportionality of resource allocation among users. A higher average suggests that the network is efficiently balancing user demands while maintaining fairness.

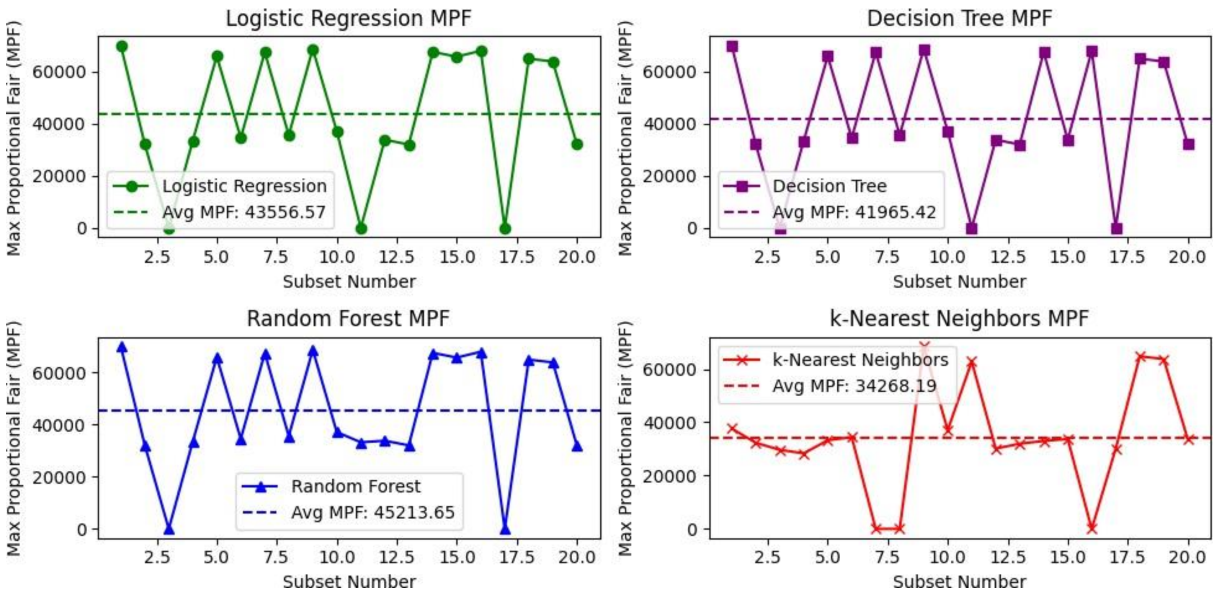
To Plot (10*10)



To Plot(15*15)



To Plot (20 *20)



To Plot(10 * 10 , 15*15, 20*20 using Genetic Algorithm)

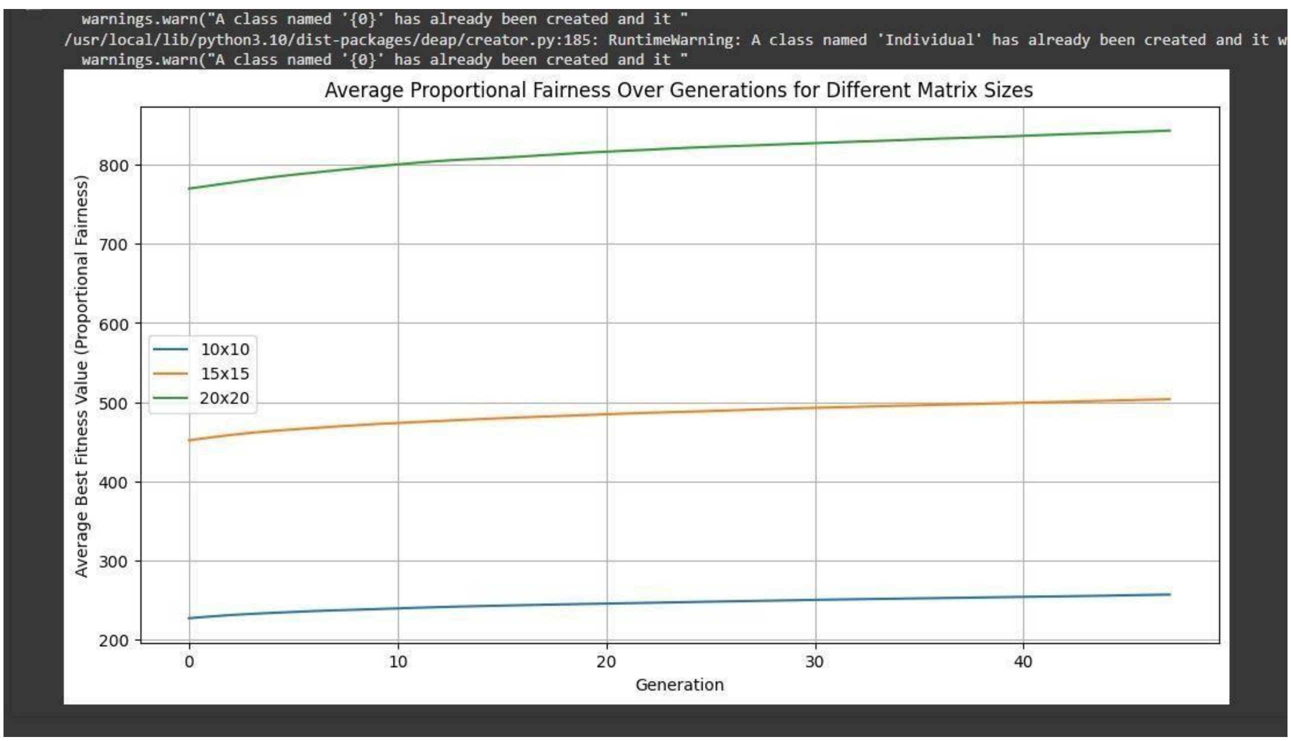


Fig: Avg of MPF

5.1 Model Implementation and Training

In our project on dynamic spectrum allocation using deep learning, we implemented a hybrid model that combines Genetic Algorithms (GA) and Neural Networks (NN) for optimizing channel selection in cognitive radio networks. The model begins by initializing a population of potential solutions representing channel allocations. Each individual's fitness is evaluated based on performance metrics such as Maximum Minimum Rate (MMR), Maximum Signal Range (MSR), and Maximum Proportional Fairness (MPF). The GA employs selection, crossover, and mutation processes to evolve these solutions across multiple generations, enhancing their effectiveness. Simultaneously, the Neural Network is trained using historical and real-time data to predict optimal channel usage and user requirements. The training involves backpropagation and optimization techniques to minimize prediction errors, allowing the model to adapt to changing network conditions. This integrated approach not only improves the spectrum utilization efficiency but also ensures a robust response to dynamic user demands, resulting in a scalable solution for real-time spectrum management.

5.2 Model Evaluation Metrics

To assess the performance of our dynamic spectrum allocation model, we employed several key evaluation metrics. Maximum Minimum Rate (MMR) measures the lowest throughput across all musers, ensuring that every user achieves a minimum quality of service.

Maximum Signal Range (MSR) evaluates the farthest distance from which users can maintain effective communication, indicating the model's capability to optimize coverage. Maximum Proportional Fairness (MPF assesses the fairness of resource allocation among users by balancing throughput and user satisfaction. Additionally, we tracked metrics such as channel utilization and interference levels to evaluate the model's effectiveness in minimizing conflicts and maximizing spectrum efficiency. By analyzing these metrics across multiple generations of the Genetic Algorithm, we determined the model's overall success in enhancing wireless network performance while meeting user demands.

Genetic Algorithm Findings

MATRICES	MSR	MMR	MPF
10X10	7046.885	394.2604	256.028
15X15	12129.92339	480.2804	507.696
20X20	18263.760	559.6930	853.390

Machine Learning algorithms Findings

10*10 Matrix

ML ALGORITHMS	MSR	MMR	MPF
LOGISTIC REGRESSION	3645.2	0.771	17395.19
DECISION TREE	2342.1	0.6789	18807.5
RANDOM FOREST	2342.1	0.771	17395.19
KNN	2590.1	0.8859	14837.74

15*15 Matrix

ML ALGORITHMS	MSR	MMR	MPF
LOGISTIC REGRESSION	7101.8	1.706	35231.67
DECISION TREE	5955.06	1.606	33537.11
RANDOM FOREST	7119.4	1.706	35221.67
KNN	5032.3	1.1939	38035.86

20*20 Matrix

ML ALGORITHMS	MSR	MMR	MPF
LOGISTIC REGRESSION	14759.1	1.8405	43556.5
DECISION TREE	14789.1	1.865	41965.41
RANDOM FOREST	15260.1	1.8654	45213.64
KNN	11287.45	1.688	34268.18

5.3 Model Deployment: Testing and Validation

The deployment phase of our dynamic spectrum allocation model involved rigorous testing validation to ensure its effectiveness in real-world scenarios. We conducted a series of simulations to evaluate the model's performance under various network conditions, including different user densities and interference levels. By utilizing a testing dataset, we validated the model's ability to dynamically allocate spectrum while minimizing conflicts and maximizing overall efficiency. Key metrics such as channel utilization, interference rates, and user satisfaction were monitored to gauge performance. Additionally, we implemented cross-validation techniques to ensure robustness and generalizability of the model across diverse environments. The results demonstrated a significant improvement in spectrum management capabilities, confirming the model's reliability and effectiveness for practical deployment in cognitive radio networks.

5.4 Results

1. **Problem-Solving Approach**
   - **GA:**
     - Solves the problem as a multi-objective optimization task.
     - Directly optimizes the utility functions (MSR, MMR, and MPF) by exploring the solution space iteratively through genetic operations like selection, crossover, and mutation.
   - **ML:**
     - Focuses on predicting allocation outcomes based on patterns learned from historical data (training set).
     - It provides faster decision-making but does not inherently optimize complex multi-objective

CHAPTER 6

CONCLUSION

6.1 Project conclusion

This project addressed the challenge of optimizing spectrum allocation in Cognitive Radio (CR) networks by comparing the performance of Genetic Algorithms (GA) and Machine Learning (ML) models, such as Logistic Regression, Decision Tree, Random Forest, and K-Nearest Neighbors (KNN). By leveraging utility functions—Max-Sum-Reward (MSR), Max-Min-Reward (MMR), and Max-Proportional-Fair (MPF)—we evaluated the ability of these methods to maximize spectrum utilization, ensure fairness, and maintain proportionality. The results demonstrate that GA performs better in multi-objective optimization. Its adaptability to dynamic environments and ability to explore large solution spaces make it an ideal choice for real-time spectrum management in CR networks. ML algorithms, particularly KNN and Random Forest, showed strong performance in fairness-oriented tasks (MMR). However, their reliance on historical data and limited adaptability to changing conditions make them less effective for dynamic optimization tasks.
In conclusion, GA performs better and is the recommended approach for optimizing spectrum allocation in CR networks, especially in environments requiring real-time adaptability and comprehensive multi-objective optimization. This project lays the foundation for integrating these methods into next-generation wireless systems, such as 6G and IoT, to address the growing demand for efficient and intelligent spectrum utilization.

6.2 Future Scope

- **Integration with Emerging Technologies:** Expand the model to support 5G and future wireless communication technologies to meet the increasing demand for spectrum efficiency.
- **Advanced Machine Learning Techniques:** Incorporate techniques such as transfer learning and ensemble methods to improve adaptability and performance in varied network conditions.
- **IoT Applications:** Adapt the system for Internet of Things (IoT) environments, optimizing spectrum management for smart cities and industrial automation scenarios.
- **Real-Time Data Analytics:** Implement real-time analytics and feedback loops to enable proactive spectrum allocation adjustments in response to dynamic network changes.
- **Security Measures:** Explore and develop robust security protocols to protect against vulnerabilities in decentralized spectrum management, ensuring reliable operation across

CHAPTER 7

REFERENCES

1. Kiran Kumar Anumandla, Rangababu peesapati, Samrat L. Sabat, Siba K. Udgata, Ajith Abraham (2013) Field programmable gate arrays-based differential evolution coprocessor: a case study of spectrum allocation in cognitive radio network

2. Kiran Kumar Anumandla, Rangababu peesapati, Samrat L. Sabat, Prabu A.V., J.R.K. Kumar Dabbakuti, Ranjita Rout (2020) Optimal Spectrum and Power Allocation using evolutionary algorithms for cognitive radio networks

3. Nitish Mishra, Sumit Srivastava, Shivendra Nath Sharan(2021) RADDPG: Resource Allocation in cognitive radio with Deep Reinforcement Learning

4. Kiran Kumar Anumandla, Rangababu peesapati, Samrat L. Sabat, Prabu A.V(2014) Comparative study of system on chip based solution for floating and fixed point differential evolution algorithm

5. Anumandla Kiran Kumar, Maddipati Kiran Kumar, Tirumala Reddy Y, G. Sai Kiran (2021) Hardware Implementation of TLBO Algorithm for Cognitive

6. Kiran Kumar Anumandla, Rangababu peesapati, Samrat L. Sabat (2017) Hardware implementation of multi-objective differential evolution algorithm