INDEX

PREFACE

Edge analytics form the backbone of modern computing, enabling efficient solutions to complex problems in areas like optimization, networks, and data processing. This lab manual for M.Tech CSE I Year II Semester equips students with hands-on skills to implement and analyze key techniques—from brute-force enumeration to sophisticated string matching and flow algorithms—fostering deep insight into performance trade-offs.

The experiments align with core texts like S. Sridhar's *Design and Analysis of Algorithms* and references such as Cormen et al.'s *Introduction to Algorithms*, emphasizing practical coding in C/C++ alongside theoretical rigor. Through these 10 structured labs, students will not only code solutions but also evaluate time/space complexities, preparing them for real-world applications in IT infrastructure and systems design.

ACKNOWLEDGEMENT

It has been a rewarding experience working on the Edge analytics Laboratory Manual. We would like to express our sincere gratitude to **Dr.K.Abdul Basith**, Professor and Head of the Department of Computer Science and Engineering, Marri Laxman Reddy Institute of Technology & Management, for his constant encouragement, valuable guidance, and continuous support in the preparation of this manual.

We are deeply indebted and gratefully acknowledge the support and motivation provided by the **Dr.P.Sridhar** Director, Marri Laxman Reddy Institute of Technology & Management, for granting us the opportunity and necessary resources to develop this laboratory manual.

Our heartfelt thanks to **Dr. R. Murali Prasad**, Principal, Marri Laxman Reddy Institute of Technology & Management, for his insightful suggestions, timely feedback, and academic guidance throughout the preparation of this document.

Finally, we extend our sincere appreciation to all the faculty members of the CSE Department whose encouragement, cooperation, and constructive inputs have played a significant role in helping us accomplish this work successfully.

COMPUTER SCIENCE AND ENGINEERING

EDGE ANALYTICS LAB MANUAL

GENERAL INSTRUCTIONS

- Students are instructed to attend the laboratory on time. Late comers will not be entertained in the lab.
- Students must be punctual. Experiments conducted during the session will not be repeated for those who are absent or late.
- Students are expected to come prepared with the theory and procedure of the experiment scheduled for the day.
- The use of mobile phones inside the lab is strictly prohibited.
- Any damage or loss of system components such as keyboard, mouse, or other peripherals during the lab session will be the responsibility of the student, and a fine or penalty will be imposed accordingly.
- Students must update their lab records and observation books session-wise. Before leaving the lab, the student should get their observation book signed by the concerned faculty member.
- Lab records must be submitted in the next lab session to the respective faculty members in the staffroom for correction and return.
- Students should not move around or disturb others during the lab session.
- In case of any emergency, students must obtain written permission from the concerned faculty member before leaving the laboratory.
- Faculty members reserve the right to suspend any student from the lab session on disciplinary grounds.

COMPUTER SCIENCE AND ENGINEERING

EDGE ANALYTICS LAB MANUAL

INSTITUTE VISION AND MISSION

Vision:

To be as an ideal academic institution by graduating talented engineers to be ethically strong, competent with quality research and technologies.

Mission:

- Utilize rigorous educational experiences to produce talented engineers
- Create an atmosphere that facilitates the success of students
- Programs that integrate global awareness, communication skills and Leadership qualities
- Education and Research partnership with institutions and industries to prepare the students for interdisciplinary research

COMPUTER SCIENCE AND ENGINEERING

EDGE ANALYTICS LAB MANUAL

DEPARTMENT VISION AND MISSION

Vision:

To empower the students to be technologically adept, innovative, self-motivated and responsible global citizen possessing human values and contribute significantly towards high quality technical education with ever changing world.

Mission:

- To offer high-quality education in the computing fields by providing an environment where the knowledge is gained and applied to participate in research, for both students and faculty.
- To develop the problem solving skills in the students to be ready to deal with cutting edge technologies of the industry.
- To make the students and faculty excel in their professional fields by inculcating the communication skills, leadership skills, team building skills with the organization of various co-curricular and extra- curricular programmes.
- To provide the students with theoretical and applied knowledge, and adopt an education approach that promotes lifelong learning and ethical growth.

COMPUTER SCIENCE AND ENGINEERING

EDGE ANALYTICS LAB MANUAL

PROGRAM OUTCOMES

PO No.	NBA Statement / Vital Features	No. of Vital Features
PO1.	**An ability to independently carry out research /investigation and development work to solve practical problems.** 1. Research problems in Computer Science and Engineering are clearly identified and defined. 2. Literature review highlights research gaps and suitable methods. 3. Experiments or simulations are conducted using appropriate tools. 4. Data collection, analyses, and interpretation systematically. 5. Innovative approaches are applied to engineering problem-solving. **6.** Results are validated against established theories and standards.	**6**
PO 2.	**An ability to write and present a substantial technical report/document** 1. Technical reports, dissertations, and papers are well-structured. 2. Referencing and academic integrity practices are properly maintained. 3. Content is presented with clarity, precision, and logical flow. 4. Oral communication and presentation skills are effectively demonstrated. 5. Digital tools are used for documentation and visualization. **6.** Research findings are communicated to both technical and non-technical audience.	**6**
PO 3.	**Students should be able to demonstrate advanced proficiency in Computer Science and allied emerging areas of Engineering.** i. In-Depth Technical Knowledge ii. Advanced Problem-Solving and Algorithmic Skills. iii. Hands-On Practical Expertise iv. Research and Innovation Aptitude **v.** Interdisciplinary Integration	**5**

PO 4.	**Students should be able to identify, analyze, and effectively solve complex real-world problems by applying advanced computing concepts, while considering solutions from a global perspective** i. Ability to break down multifaceted problems into manageable parts and develop effective solutions using advanced computing techniques. ii. Analytical Thinking and Critical Reasoning iii. Advanced Computing Knowledge. iv. Capability to understand and model complex systems, considering interdependencies and dynamic behaviors within global contexts. v. Global and Societal Awareness. vi. Aptitude for developing novel approaches and innovative solutions to address complex challenges. vii. Ability to work effectively in diverse teams and integrate knowledge from multiple disciplines to solve global problems. viii. Skill in clearly presenting problem analyses, solutions, and their implications to technical and non-technical global audience.	**8**
PO 5.	**An ability to acquire and apply advanced technical knowledge, professional skills, and modern computing tools to develop sustainable solutions** i. Ability to continuously learn and apply cutting-edge technical knowledge in computing and related fields. ii. Proficiency with Modern Computing Tools. iii. Capability to design and implement solutions that are environmentally, and economically feasible. iv. Understanding of professional ethics and responsibility in creating technology solutions with long-term positive impact. v. Integration of Multidisciplinary Knowledge. vi. Adaptability to sustainable practices.	**6**
PO 6.	**An Ability to recognize the significance of lifelong learning and actively pursue continuous professional development by adapting technologies in emerging areas.** i. Self-Directed Learning. ii. Skill in quickly learning and integrating new technologies and tools as they emerge in the field. iii. Capability to assess the relevance and impact of emerging technologies and decide on their applicability. iv. Continuous Professional Development Planning. v. Ability to engage with professional communities, attend workshops, and collaborate to stay updated. vi. Habit of regularly reflecting on personal growth, learning experiences, and professional competencies to improve continuously.	**6**

MARRI LAXMAN REDDY
INSTITUTE OF TECHNOLOGY AND MANAGEMENT
(AN AUTONOMOUS INSTITUTION)
(Approved by AICTE, New Delhi & Affiliated to JNTUH, Hyderabad)
Accredited by NBA and NAAC with 'A' Grade & Recognized Under Section2(f) & 12(B)of the UGC act,1956

COMPUTER SCIENCE AND ENGINEERING

EDGE ANALYTICS LAB MANUAL

PROGRAM EDUCATIONAL OBJECTIVES

Sl. No.	PEOs Name	Program Education Objective Statements
1	PEO – 1	Graduates will achieve professional excellence and success in the field of Computer Science and Engineering by applying strong technical foundations and problem-solving

		skills to contribute effectively to industry, academia, and entrepreneurship.
2	**PEO – 2**	Graduates will demonstrate a commitment to lifelong learning by continuously enhancing their knowledge and skills through professional development and self-directed learning to effectively adapt to evolving global challenges.
3	**PEO – 3**	Graduates of the Computer Science and Engineering program will actively pursue advanced research, contributing to the development of solutions for complex problems and the generation of new knowledge to effectively address real-world challenges.
4	**PEO – 4**	Graduates will exhibit professionalism, effective communication, leadership skills, and ethical responsibility while working in multidisciplinary teams to deliver computing solutions that address societal needs and contribute to sustainable development.

COMPUTER SCIENCE AND ENGINEERING

EDGE ANALYTICS LAB MANUAL

PROGRAM SPECIFIC OUTCOMES

PSO1: Applications of Computing: Ability to use knowledge in various domains to provide solution to new ideas and innovations.

PSO2: Programming Skills: Identify required data structures, design suitable algorithms, develop and maintain software for real world problems.

COMPUTER SCIENCE AND ENGINEERING

EDGE ANALYTICS LAB MANUAL

Course Structure

Edge analytics Lab will have a continuous evaluation during II semester for 40 sessional marks and 60 end semester examination marks.Out of the 40 marks for internal evaluation, day-to-day work in the laboratory shall be evaluated for 20 marks and internal practical examination shall be evaluated for 10 marks conducted by the laboratory teacher concerned.

The end semester examination shall be conducted with an external examiner and internal examiner. The external examiner shall be appointed by the principal / Chief Controller of examinations

COMPUTER SCIENCE AND ENGINEERING

EDGE ANALYTICS LAB MANUAL

OBJECTIVE:

- Understand the concept of edge computing
- Understand the Edge computing Architecture
- Implement the edge computing in IOT
- Understand the concept of multi-access edge computing
- Implement edge computing in MEC

OUTCOMES:

Upon the completion of Operating Systems practical course, the student will be able to:

- Identify the benefits of edge computing
- Develop the microservices in iofog
- Develop user defined services in the edge
- Create use cases in IOT with edge computing
- Develop services in MEC
- Implement use cases in MEC

2425838: EDGE ANALYTICS LAB (Professional Elective - III Lab)

M.Tech CSE I Year II Sem.

L	T	P	C
0	0	4	2

Course Objectives:

1. Understand the concept of edge computing
2. Understand the Edge computing Architecture
3. Implement the edge computing in IOT
4. Understand the concept of multi-access edge computing
5. Implement edge computing in MEC

Course Outcomes:

1. Identify the benefits of edge computing
2. Develop the microservices in iofog
3. Develop user defined services in the edge
4. Create use cases in IOT with edge computing
5. Develop services in MEC
6. Implement use cases in MEC

List of Experiments:

1. Set up the Arduino IDE for ESP8266-12 module and program it to blink a LED light.
2. Installation tools to create and manage ECN's
3. Deploy micro services and writing your own microservices
4. Setup the Communication Parameters
5. Implement any two Communications protocols
6. Deploy modules to a Windows IoT Edge device
7. Create an IoT hub.
8. Register an IoT Edge device to your IoT hub.
9. Install and start the IoT Edge for Linux on Windows runtime on your device.
10. Remotely deploy a module to an IoT Edge device and send telemetry.
11. Python based basic programs using Raspberry Pi.
12. Deploy a module Manage your Azure IoT Edge device from the cloud to deploy a module that sends telemetry data to IoT Hub.
13. Publishing Data using HTTP.
14. Sensor Interfacing and Logging using MQTT.
15. File IO Example - # Example code to demonstrate writing and reading data to/from files
16. write code to turn on one of the LEDs on the board (Breadboard)

Additional Exercises on IOT Edge Analytics Applications

17. Temperature Logger
18. Home Automation

TEXT BOOKS:

1. Hands-On Edge Analytics with Azure IoT: Design and develop IoT applications with edge analytical solutions including Azure IoT Edge by Colin Dow
2. MicroPython for the Internet of Things A Beginner's Guide to Programming with Python on Microcontroller, Charles Bell, A Press.

REFERENCE BOOKS:

1. Learn Edge Analytics - Fundamentals of Edge Analytics: Automated analytics at source using Microsoft Azure by Ashish Mahajan
2. Peter Waher, "Mastering Internet of Things: Design and create your own IoT applications using Raspberry Pi 3", First Edition, Packt Publishing, 2018
3. John C. Shovic, "Raspberry Pi IoT Projects: Prototyping Experiments for Makers", Packt Publishing, 2016
4. Python for Microcontrollers: Getting Started with MicroPython Paperback – 16 December 2016, by Donald Norris, McGraw-Hill Education TAB
5. Programming with MicroPython: Embedded Programming with Microcontrollers and Python, by Nicholas H. Tollervey, O′Reilly
6. R. Buyya, S.N. Srirama (2019), Fog and Edge Computing: Principles and Paradigms, Wiley-Blackwell, 2019.

Experiment-1

Set up the Arduino IDE for ESP8266-12 module and program it to blink a LED light.

Steps:

1. **Install Arduino IDE:** Download and install the Arduino IDE.
2. **Add ESP8266 Board:**
   - Open Arduino IDE.
   - Go to File > Preferences.
   - In the "Additional Boards Manager URLs" field, add:

http://arduino.esp8266.com/stable/package_esp8266com_index.json

   - Click OK.
3. **Install ESP8266 Board:**
   - Navigate to Tools > Board > Boards Manager.
   - Search for "ESP8266" and click Install.
4. **Select ESP8266 Board:**
   - Go to Tools > Board and select your ESP8266 model (e.g., NodeMCU 1.0).
5. **Connect LED:**
   - Connect an LED to the ESP8266, typically to GPIO2 (D4).Microsoft Learn
6. **Upload Blink Code:**
   - Use the following code to blink the LED:

```
void setup() {
pinMode(D4, OUTPUT);
}
void loop() {
digitalWrite(D4, HIGH);
delay(1000);
digitalWrite(D4, LOW);
delay(1000);
}
```

   - Click the Upload button to program the ESP8266.

Experiment-2

Installation tools to create and manage ECN's.

An ECN (Electronic Communications Network) broker facilitates direct trading between participants in equity and currency markets through electronic networks. To manage ECNs, tools like MetaTrader, cTrader, or proprietary trading platforms are commonly used. These platforms allow for real-time order book access, trade data analysis, and execution of trades with reduced latency.

Experiment-3

Deploy micro services and writing your own microservices.

Microservices architecture involves developing small, independent services that communicate over a network.

Steps:

1. **Develop Microservice:**
   - Use frameworks like Flask (Python) or Spring Boot (Java) to create microservices.
   - Ensure each service has its own database and is independently deployable.
2. **Containerize Microservice:**
   - Create a Dockerfile to containerize the microservice.
   - Build the Docker image:

```
docker build -t my-microservice.
```

3. **Deploy Microservice:**
   - Use orchestration tools like Kubernetes or Docker Compose to deploy and manage microservices.

Experiment-4

Setup the Communication Parameters.

For IoT devices, communication parameters include:

- Protocol: MQTT, HTTP, CoAP
- Port: e.g., 1883 for MQTT, 80 for HTTP
- Security: TLS/SSL encryption
- Authentication: API keys, certificates

Configure these parameters in your device firmware and cloud platform settings to ensure secure and reliable communication.

Experiment-5

Implement any two Communications protocols.

1.MQTT (Message Queuing Telemetry Transport)

- Lightweight protocol suitable for low-bandwidth, high-latency networks.
- Use libraries like paho-mqtt in Python or PubSubClient in Arduino.

2. HTTP (Hypertext Transfer Protocol)

- Standard protocol for web communication.
- Use libraries like requests in Python or ESP8266HTTPClient in Arduino.

Experiment-6

Deploy modules to a Windows IoT Edge device.

To deploy modules to a Windows IoT Edge device:

1. **Install IoT Edge Runtime:**
   - Follow the official guide to install the IoT Edge runtime on your Windows device.
2. **Deploy Modules:**
   - Use Azure IoT Hub to create a deployment manifest.
   - Deploy the manifest to your IoT Edge device.

Experiment-7

Create an IoT hub.

To create an IoT Hub on Azure:

1. Sign in to the Azure portal.
2. Navigate to Create a resource > Internet of Things > IoT Hub.
3. Fill in the required details:
    - Subscription: Select your Azure subscription.
    - Resource Group: Create a new or select an existing resource group.
    - Region: Choose the region closest to you.
    - IoT Hub Name: Enter a globally unique name for your IoT Hub.
4. Click Review + Create and then Create.

Experiment-8

8 Register an IoT Edge Device to Your IoT Hub

To register an IoT Edge device:

1. In the Azure portal, navigate to your IoT Hub.
2. Select IoT Edge under Device Management.
3. Click + Add to add a new device.
4. Provide a Device ID and select **Authentication Type

Experiment-9

Install and start the IoT Edge for Linux on Windows runtime on your device.

To install and start the Azure IoT Edge for Linux on Windows (EFLOW) runtime:

1. **Install EFLOW:**
   - Follow the official guide to install EFLOW on your Windows device.
2. **Start the EFLOW VM:**
   - Use the Windows Admin Center or PowerShell to start the EFLOW virtual machine. Microsoft Tech Community+1Azure+1
3. **Verify Installation:**
   - Run iotedge version in PowerShell to verify that the IoT Edge runtime is installed and running.

Experiment-10

Remotely Deploy a Module to an IoT Edge Device and Send Telemetry

To deploy a module and send telemetry:

1. **Create a Deployment Manifest**:
   - In the Azure portal, navigate to your IoT Hub.
   - Under **Device Management**, select **IoT Edge**.
   - Choose your device and click **Set Modules**.
   - Add the desired module (e.g., a simulated temperature sensor).
   - Specify routes for telemetry data.DEV CommunityMicrosoft Learn
2. **Deploy the Manifest**:
   - Click **Review + Create** and then **Create** to deploy the module to your IoT Edge device.
3. **Monitor Telemetry**:
   - Use Azure Monitor or IoT Hub metrics to monitor the telemetry data sent by the module.

Experiment-11

Python-Based Basic Programs Using Raspberry Pi.

To write basic Python programs on a Raspberry Pi:Tom's Hardware

1. **Install Python**:
   - Python is pre-installed on Raspberry Pi OS.
2. **Write a Simple Program**:
   - Open a terminal and create a new Python file:

     nano hello.py
   - Write the following code:Raspberry Pi Stack Exchange

     print("Hello, Raspberry Pi!")
   - Save and run the program:

     python3 hello.py

Experiment-12

12) Deploy a Module to Manage Your Azure IoT Edge Device from the Cloud.

To deploy a module that sends telemetry data to IoT Hub:

1. **Create a Custom Module**:
   - Develop a Python module that collects and sends telemetry data.
2. **Containerize the Module**:
   - Create a Docker file and build the container image.
3. **Push to Azure Container Registry**:
   - Push the container image to Azure Container Registry.DEV Community+8Microsoft Learn+8Microsoft Learn+8
4. **Deploy via Azure Portal**:
   - Follow the steps in Deploy modules from Azure portal to deploy the custom module to your IoT Edge device.Microsoft Learn.

Experiment-13

13) Publishing Data Using HTTP.

To publish data using HTTP:

1. **Set Up an HTTP Server**:
    - Use Flask (Python) or Express (Node.js) to set up an HTTP server.
2. **Send Data**:
    - Use requests (Python) or axios (Node.js) to send data to the server.
3. **Handle Data**:
    - Process the incoming data on the server and send appropriate responses.

Experiment-14

14) Sensor Interfacing and Logging Using MQTT.

To interface a sensor and log data using MQTT:

1. **Connect Sensor**:
   - Connect a sensor (e.g., DHT11) to the Raspberry Pi.
2. **Install MQTT Client**:
   - Install an MQTT client library:

pip install paho-mqtt

3. **Write Code**:
   - Write Python code to read sensor data and publish it to an MQTT broker.
4. **Log Data**:
   - Log the received data to a file or database for later analysis.

Experiment-15

File I/O Example.

To demonstrate writing and reading data to/from files:

1. **Write to a File**:
   - Use the following Python code to write data:

```
with open("data.txt", "w") as file:
file.write("Hello, File I/O!")
```

2. **Read from a File**:
   - Use the following Python code to read data:

```
with open("data.txt", "r") as file:
print(file.read())
```


Experiment-16

16) Write Code to Turn On One of the LEDs on the Board (Breadboard).

To control an LED connected to the Raspberry Pi:scribles.net+1Raspberry Pi Stack Exchange+1

1. **Connect LED**:
   - Connect the LED to a GPIO pin (e.g., GPIO17).
2. **Write Python Code**:
   - Use the following Python code to turn on the LED:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setup(17, GPIO.OUT)
GPIO.output(17, GPIO.HIGH)
time.sleep(1)
GPIO.output(17, GPIO.LOW)


GPIO.cleanup()
```