Assignment 6.1: Pre-Trained Generative Model for IoT Applications

Daily RAN Network Health Report Generator

Student: Marco Gonzalez
Course: AAI-530 - Introduction to IoT
Date: February 2026

Use Case Description

IoT Application: Automated RAN Network Health Reporting

This application demonstrates how generative AI can transform raw telemetry data from 5G Radio Access Network (RAN) IoT sensors into human-readable daily health reports for network operations teams.

Problem: Network operations centers monitor thousands of base stations generating millions of metrics daily. Human operators struggle to quickly identify critical issues and trends from raw numerical data.

Solution: An automated system that:

1. Collects 24-hour aggregated metrics from RAN IoT sensors (cell towers)
2. Structures the data in JSON format
3. Sends the data to a generative AI model via API
4. Receives a natural language summary highlighting critical issues, trends, and recommended actions

Benefits:

- Reduces time to identify network issues from hours to seconds
- Makes technical metrics accessible to non-technical stakeholders
- Provides actionable recommendations based on industry best practices
- Enables early warning of capacity or performance problems

This type of AI-powered reporting is increasingly used in IoT telemetry applications across industries including telecommunications, industrial IoT, smart cities, and energy management systems.

Setup and Dependencies

In [ ]:
```
# Install required package (uncomment if needed)
# !pip install openai

import json
import os
from datetime import datetime, timedelta
from openai import OpenAI
```

API Connection Setup

We use the OpenAI API to access GPT-4 for natural language generation. The API key should be stored as an environment variable for security.

Note: For this demo, you can get a free API key at https://platform.openai.com/api-keys

To set your API key:

export OPENAI_API_KEY='your-api-key-here'

In [ ]:
```
# Initialize OpenAI client
client = OpenAI(
    api_key=os.environ.get("OPENAI_API_KEY")
```

```
)

print("✓ OpenAI API client initialized successfully")
```

Sample RAN Telemetry Data

This JSON structure represents the type of aggregated daily metrics that would be produced by an IoT monitoring system collecting data from 5G base stations.

Data collected by IoT sensors:

- Cell site performance metrics (throughput, latency, packet loss)
- Resource utilization (CPU, memory, bandwidth)
- Quality of Service indicators (handover success rate, call drops)
- Alarm and alert counts
- User traffic patterns

In [ ]:
```
# Sample daily telemetry data from RAN IoT monitoring system
ran_telemetry_data = {
  "report_date": "2026-02-17",
  "network_id": "Metro-West-Cluster-01",
  "total_cells": 247,
  "summary_metrics": {
    "avg_throughput_mbps": 485.3,
    "avg_latency_ms": 12.4,
    "avg_packet_loss_pct": 0.18,
    "avg_cpu_utilization_pct": 67.2,
    "avg_resource_block_utilization_pct": 73.8,
    "total_active_users": 142380,
    "peak_concurrent_users": 18560
  },
  "quality_indicators": {
    "handover_success_rate_pct": 97.2,
    "call_drop_rate_pct": 0.45,
    "coverage_availability_pct": 99.87,
    "avg_signal_strength_rsrp_dbm": -82.3
  },
  "alerts": {
    "critical": 3,
    "major": 12,
    "minor": 28,
    "warnings": 56
  },
  "top_issues": [
    {
      "cell_id": "CELL-089",
      "issue": "High CPU utilization",
      "severity": "critical",
      "value": "94.2% avg",
      "duration_hours": 8.3
    },
    {
      "cell_id": "CELL-156",
      "issue": "Elevated packet loss",
      "severity": "major",
      "value": "2.1% loss rate",
      "duration_hours": 4.7
    },
    {
      "cell_id": "CELL-223",
      "issue": "Handover failures",
      "severity": "major",
      "value": "12% failure rate",
      "duration_hours": 2.1
    }
  ],
  "capacity_trends": {
    "resource_utilization_trend": "increasing",
    "week_over_week_growth_pct": 3.2,
    "cells_above_80pct_utilization": 34,
    "estimated_days_to_capacity": 45
  }
}
```

```
# Display the data structure
print("Sample RAN Telemetry Data (JSON):")
print(json.dumps(ran_telemetry_data, indent=2))
```

Generative AI Prompt Engineering

The prompt instructs the AI model how to interpret the telemetry data and generate a useful report. Key elements:

1. **Role definition**: Act as a network operations analyst
2. **Task specification**: Generate a daily health report
3. **Output format**: Structured sections with executive summary, critical issues, trends, and recommendations
4. **Context**: Include industry-specific knowledge about RAN performance thresholds
5. **Tone**: Professional but accessible to non-technical stakeholders

In [ ]:
```
def generate_network_report(telemetry_data):
    """
    Generate a daily network health report using OpenAI GPT-4.

    Args:
        telemetry_data: Dictionary containing RAN telemetry metrics

    Returns:
        String containing the generated report
    """

    # Convert telemetry data to JSON string
    telemetry_json = json.dumps(telemetry_data, indent=2)

    # Create the system prompt
    system_prompt = """You are an expert network operations analyst for a telecommunications company.
Your role is to analyze daily telemetry data from 5G Radio Access Network (RAN) infrastructure and
generate clear, actionable daily health reports for network operations teams.

Use the following performance thresholds as reference:
- Good throughput: >400 Mbps
- Good latency: <15 ms
- Acceptable packet loss: <0.5%
- Healthy CPU utilization: <75%
- Good handover success: >98%
- Acceptable call drop rate: <0.5%

Generate reports that are professional, concise, and highlight actionable insights."""

    # Create the user prompt with the telemetry data
    user_prompt = f"""Please analyze the following 24-hour RAN network telemetry data and generate a
comprehensive daily health report.

The report should include:
1. Executive Summary (2-3 sentences on overall network health)
2. Key Performance Metrics (highlight metrics that are outside normal ranges)
3. Critical Issues (list of urgent problems requiring immediate attention)
4. Capacity and Trends (analysis of utilization trends and capacity planning)
5. Recommendations (specific actions for the operations team)

Telemetry Data:
{telemetry_json}

Format the report in markdown with clear sections and bullet points where appropriate."""

    # Call OpenAI API
    print("Calling OpenAI API to generate network health report...\n")

    response = client.chat.completions.create(
        model="gpt-4",  # Using GPT-4 for better analytical capabilities
        messages=[
            {"role": "system", "content": system_prompt},
            {"role": "user", "content": user_prompt}
        ],
        temperature=0.3,  # Lower temperature for more factual, consistent output
        max_tokens=1500
    )
```

```
    # Extract the generated report
    report = response.choices[0].message.content

    return report
```

Generate the Daily Network Health Report

In [ ]:
```
# Generate the report
daily_report = generate_network_report(ran_telemetry_data)

# Display the generated report
print("="*80)
print("GENERATED DAILY NETWORK HEALTH REPORT")
print("="*80)
print(daily_report)
print("="*80)
```

Example Output

Since the actual API call requires a valid API key and incurs costs, here's an example of what the generated report would look like:

Daily Network Health Report
Network: Metro-West-Cluster-01
Date: February 17, 2026
Total Cells Monitored: 247

Executive Summary

The Metro-West-Cluster-01 network is operating at satisfactory levels overall, with 99.87% coverage availability and strong average throughput of 485.3 Mbps. However, 3 critical and 12 major alerts require immediate attention, primarily related to resource utilization approaching capacity thresholds. Capacity planning should be prioritized as 34 cells are operating above 80% resource utilization.

Key Performance Metrics

Performing Well:
- Throughput: 485.3 Mbps average (exceeds 400 Mbps target)
- Latency: 12.4 ms average (within 15 ms target)
- Packet Loss: 0.18% average (below 0.5% threshold)
- Coverage Availability: 99.87%

Requires Attention:
- CPU Utilization: 67.2% average approaching 75% threshold (34 cells above 80%)
- Resource Block Utilization: 73.8% indicating high demand
- Handover Success Rate: 97.2% (below 98% target)
- Call Drop Rate: 0.45% (acceptable but near 0.5% limit)

Critical Issues

CRITICAL - Immediate Action Required:

1. CELL-089: Sustained High CPU Utilization
   - Issue: CPU utilization at 94.2% for 8.3 hours
   - Impact: Risk of service degradation, dropped connections
   - Action: Immediate investigation; consider load balancing or traffic offload to neighboring cells

MAJOR Issues:

2. CELL-156: Elevated Packet Loss
   - Issue: 2.1% packet loss rate for 4.7 hours
   - Impact: Degraded user experience, potential video buffering, VoIP quality issues
   - Action: Check backhaul connection integrity; investigate interference sources

3. CELL-223: Handover Failures
   - Issue: 12% handover failure rate for 2.1 hours
   - Impact: Dropped calls during mobility, poor user experience in cell boundaries
   - Action: Verify neighbor cell configuration; check antenna alignment and coverage overlap

Capacity and Trends

Capacity Status:
- 34 cells (13.8% of network) operating above 80% resource utilization
- Week-over-week growth: +3.2% in resource utilization
- Trend: Increasing demand (sustained upward trajectory)
- Estimated time to capacity: Approximately 45 days at current growth rate

Analysis: The consistent 3.2% weekly growth indicates strong subscriber growth or increased per-user data consumption. Without intervention, capacity constraints will begin affecting service quality within 6 weeks.

Recommendations

Immediate (Next 24 Hours):
1. Investigate and resolve CELL-089 CPU overload condition
2. Deploy field technician to CELL-156 to diagnose packet loss source
3. Reconfigure handover parameters for CELL-223 and neighbor cells

Short-term (Next 7 Days):
4. Review alarm escalation procedures - 99 total alerts (3 critical, 12 major) suggest potential systemic issues
5. Implement automated load balancing for cells exceeding 80% utilization
6. Schedule proactive maintenance for cells showing degraded handover performance

Strategic (Next 30-45 Days):
7. Capacity Expansion Required: Begin planning for additional cell sites or carrier upgrades to address 45-day capacity timeline
8. Consider implementing dynamic spectrum sharing or carrier aggregation in high-demand areas
9. Evaluate deploying small cells in hotspot areas to offload traffic from macro cells

---

Next Report: February 18, 2026
On-Call Engineer: Contact NOC for escalations
This report demonstrates how generative AI transforms raw IoT telemetry data into actionable business intelligence for network operations teams.

Alternative Use Case: Energy Consumption Summary

To demonstrate versatility, here's how the same approach could be applied to smart home energy monitoring (related to our Module 5 dashboard):

In [ ]:
```
# Sample smart home energy data
energy_data = {
  "date": "2026-02-17",
  "household_id": "HOME-4472",
  "daily_summary": {
    "total_consumption_kwh": 28.4,
    "avg_power_kw": 1.18,
    "peak_power_kw": 5.2,
    "peak_time": "19:30",
    "cost_estimate_usd": 3.98
  },
  "circuit_breakdown": {
    "kitchen": {"kwh": 8.3, "percent": 29.2},
    "hvac": {"kwh": 12.1, "percent": 42.6},
    "laundry": {"kwh": 4.8, "percent": 16.9},
    "other": {"kwh": 3.2, "percent": 11.3}
  },
  "alerts": [
    "High power usage detected at 19:30 (5.2 kW - likely electric oven + dryer)",
    "HVAC consumption 15% higher than typical February average"
  ],
  "comparison": {
    "vs_yesterday_pct": -8.2,
    "vs_last_week_avg_pct": +12.4,
    "vs_similar_homes_pct": +18.7
  }
}

print("Smart Home Energy Data:")
print(json.dumps(energy_data, indent=2))

# This could be sent to the same generative AI with a different prompt
# to produce a user-friendly daily energy summary email
```

Conclusion

This demonstration shows how pre-trained generative AI models can be integrated into IoT applications to:

1. **Transform raw sensor data** into human-readable insights
2. **Reduce cognitive load** for operators monitoring complex systems
3. **Provide actionable recommendations** based on domain knowledge
4. **Enable non-technical stakeholders** to understand IoT data
5. **Scale to thousands of devices** with minimal manual effort

The same pattern can be applied across IoT domains:

- **Industrial IoT**: Equipment health summaries from vibration sensors
- **Smart Cities**: Traffic pattern analysis from connected infrastructure
- **Healthcare IoT**: Patient vital sign summaries from wearables
- **Agriculture**: Crop health reports from soil and weather sensors

As IoT systems generate increasingly large volumes of data, generative AI will become essential for extracting value and enabling rapid decision-making.