

Assignment 6.1: Pre-Trained Generative Model for IoT Applications

Daily NB-IoT Core Network Health Report Generator

Student: Marco Gonzalez

Course: AAI-530 - Introduction to IoT

Date: February 2026

Use Case Description

IoT Application: Automated NB-IoT Core Network Health Reporting

This application demonstrates how generative AI can transform raw telemetry data from Narrowband IoT (NB-IoT) Core network infrastructure into human-readable daily health reports for network operations teams.

Problem: Network operations centers monitor NB-IoT Core networks supporting millions of low-power IoT devices (smart meters, sensors, trackers). Core network elements generate massive volumes of telemetry data covering device attach procedures, session management, data throughput, and power optimization features. Human operators struggle to quickly identify issues affecting IoT device connectivity and performance from raw numerical metrics.

Solution: An automated system that:

1. Collects 24-hour aggregated metrics from NB-IoT Core network elements (MME, S-GW, P-GW, HSS)
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 IoT connectivity issues from hours to seconds
- Makes technical Core network metrics accessible to non-technical stakeholders
- Provides actionable recommendations for IoT device performance optimization
- Enables early detection of capacity constraints or service degradation
- Monitors power-saving features (PSM, eDRX) critical for battery-powered IoT devices

This type of AI-powered reporting is increasingly used in IoT Core network telemetry applications across industries including smart cities, utilities, agriculture, asset tracking, and industrial monitoring.

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 NB-IoT Core Telemetry Data

This JSON structure represents the type of aggregated daily metrics that would be produced by an IoT monitoring system collecting data from NB-IoT Core network elements.

Data collected from Core network IoT monitoring:

- Device attach and detach procedures (success rates, failures)
- PDN session establishment and teardown metrics
- Uplink/downlink data volumes per device category
- Power saving mode (PSM) and extended discontinuous reception (eDRX) usage
- MME (Mobility Management Entity) performance metrics
- Gateway (S-GW/P-GW) throughput and session counts
- HSS (Home Subscriber Server) authentication metrics
- Coverage enhancement level distribution

```
In [ ]: # Sample daily telemetry data from NB-IoT Core monitoring system
nbiot_core_telemetry_data = {
    "report_date": "2026-02-17",
    "network_id": "Metro-NB-IoT-Core-01",
    "core_network_elements": {
        "mme_count": 4,
        "sgw_count": 6,
        "pgw_count": 4,
        "hss_count": 2
    },
    "device_metrics": {
        "total_registered_devices": 2847630,
        "active_devices_24h": 1923450,
        "new_registrations_24h": 12340,
        "deregistrations_24h": 8920,
        "devices_using_psm": 2156780,
        "devices_using_edrx": 1834920
    },
    "attach_procedures": {
        "total_attach_attempts": 487650,
        "successful_attaches": 478320,
        "attach_success_rate_pct": 98.09,
        "failed_attaches": 9330,
        "avg_attach_duration_ms": 3450,
        "timeout_failures": 2140,
        "authentication_failures": 890,
        "network_reject_failures": 6300
    },
    "session_management": {
        "pdn_connection_attempts": 523470,
        "successful_pdn_connections": 518960,
        "pdn_success_rate_pct": 99.14,
        "active_pdn_sessions": 1867230,
        "avg_session_setup_time_ms": 1870,
        "session_drop_rate_pct": 0.08
    },
    "data_traffic": {
        "total_uplink_gb": 8945.6,
        "total_downlink_gb": 3287.4,
        "avg_uplink_per_device_kb": 4.8,
        "avg_downlink_per_device_kb": 1.7,
        "peak_concurrent_sessions": 892340
    },
    "mme_performance": {
        "avg_cpu_utilization_pct": 42.3,
        "avg_memory_utilization_pct": 58.7,
        "control_plane_messages_per_sec": 1840,
        "s1_interface_utilization_pct": 34.6
    }
},
```

```

"gateway_performance": {
    "sgw_avg_throughput_mbps": 2340.5,
    "pgw_avg_throughput_mbps": 2156.8,
    "s5_interface_utilization_pct": 38.2,
    "sgi_interface_utilization_pct": 41.7
},
"power_optimization": {
    "avg_psm_timer_value_hours": 24,
    "avg_edrx_cycle_seconds": 163.84,
    "devices_in_idle_mode_pct": 87.3,
    "avg_battery_saving_mode_duration_hours": 18.6
},
"alerts": {
    "critical": 2,
    "major": 7,
    "minor": 18,
    "warnings": 34
},
"top_issues": [
    {
        "component": "MME-03",
        "issue": "High authentication failure rate",
        "severity": "critical",
        "value": "8.2% failure rate",
        "affected_devices": 4830,
        "duration_hours": 6.4
    },
    {
        "component": "PGW-02",
        "issue": "Elevated SGi interface utilization",
        "severity": "major",
        "value": "78% utilization",
        "impact": "Potential throughput degradation",
        "duration_hours": 3.2
    },
    {
        "component": "S-GW-05",
        "issue": "Session setup delays",
        "severity": "major",
        "value": "4.2s avg setup time",
        "affected_sessions": 12450,
        "duration_hours": 2.8
    }
],
"capacity_trends": {
    "device_growth_trend": "increasing",
    "week_over_week_device_growth_pct": 2.8,
    "mme_capacity_utilization_pct": 42,
    "gateway_capacity_utilization_pct": 45,
    "estimated_days_to_capacity": 120
},
"coverage_enhancement": {
    "ce_level_0_devices_pct": 68.4,
    "ce_level_1_devices_pct": 24.3,
    "ce_level_2_devices_pct": 7.3
}
}

# Display the data structure
print("Sample NB-IoT Core Telemetry Data (JSON):")
print(json.dumps(nbiot_core_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 Core network operations analyst specializing in NB-IoT
2. **Task specification:** Generate a daily health report for IoT Core infrastructure
3. **Output format:** Structured sections with executive summary, critical issues, trends, and recommendations
4. **Context:** Include industry-specific knowledge about NB-IoT Core performance thresholds
5. **Tone:** Professional but accessible to non-technical stakeholders

In []: `def generate_network_report(telemetry_data):`

```

"""
Generate a daily NB-IoT Core network health report using OpenAI GPT-4.

Args:
    telemetry_data: Dictionary containing NB-IoT Core 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 Core network operations analyst for a telecommunications company specializing in Narrowband IoT (NB-IoT) infrastructure. Your role is to analyze daily telemetry data from NB-IoT Core network elements (MME, S-GW, P-GW, HSS) and generate clear, actionable daily health reports for network operations teams managing millions of IoT devices.

Use the following performance thresholds as reference:
- Good attach success rate: >98%
- Good PDN connection success rate: >99%
- Acceptable session drop rate: <0.1%
- Healthy MME CPU utilization: <60%
- Good attach duration: <4 seconds
- Acceptable session setup time: <2 seconds
- Good authentication failure rate: <2%


Generate reports that are professional, concise, and highlight actionable insights for IoT device connectivity and Core network performance."""
"""

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

The report should include:
1. Executive Summary (2-3 sentences on overall Core network and IoT device connectivity health)
2. Key Performance Metrics (highlight metrics that are outside normal ranges)
3. Critical Issues (list of urgent problems requiring immediate attention)
4. IoT Device Trends (analysis of device growth, power optimization usage, 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 NB-IoT Core 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(nbilot_core_telemetry_data)

# Display the generated report
print("*" * 80)
print("GENERATED DAILY NB-IoT CORE 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 NB-IoT Core Network Health Report
Network: Metro-NB-IoT-Core-01
Date: February 17, 2026
Registered IoT Devices: 2,847,630
```

Executive Summary

The NB-IoT Core network is operating at satisfactory levels, successfully serving 2.8 million registered IoT devices with 1.9 million active devices in the last 24 hours. However, 2 critical and 7 major alerts require immediate attention, primarily related to authentication failures on MME-03 and interface utilization on PGW-02. Power optimization features (PSM, eDRX) show strong adoption at 75% and 64% respectively, indicating effective battery life extension for IoT devices.

Key Performance Metrics

Performing Well:

- PDN Connection Success Rate: 99.14% (exceeds 99% target)
- Session Drop Rate: 0.08% (well below 0.1% threshold)
- Average Attach Duration: 3.45 seconds (within 4 second target)
- MME CPU Utilization: 42.3% average (healthy, below 60% threshold)
- Device Growth: 2.8% week-over-week (sustainable growth)

Requires Attention:

- Attach Success Rate: 98.09% (below 98% target, marginal)
- Authentication Failures: 890 failures (needs investigation)
- S-GW-05 Session Setup Time: 4.2s average (exceeds 2s target)
- PGW-02 SGi Interface: 78% utilization (approaching saturation)

Critical Issues

CRITICAL - Immediate Action Required:

1. MME-03: High Authentication Failure Rate

- Issue: 8.2% authentication failure rate affecting 4,830 devices over 6.4 hours
- Impact: IoT devices unable to attach to network, service interruption
- Root Cause: Likely HSS synchronization issue or incorrect device credentials
- Action: Immediately investigate HSS-MME communication; verify authentication vectors; check for credential provisioning errors

MAJOR Issues:

2. PGW-02: Elevated SGi Interface Utilization

- Issue: 78% SGi interface utilization for 3.2 hours
- Impact: Potential throughput degradation for IoT data transmission to external networks
- Action: Monitor for continued growth; prepare to activate additional PGW capacity or redistribute traffic load

3. S-GW-05: Session Setup Delays

- Issue: Average 4.2 second session setup time, affecting 12,450 sessions over 2.8 hours
- Impact: Delayed IoT device connectivity, increased battery drain during connection attempts
- Action: Investigate S5 interface latency; check S-GW-05 resource utilization; review recent configuration changes

IoT Device Trends

Device Population:

- Total Registered Devices: 2,847,630
- Active Devices (24h): 1,923,450 (67.5% activity rate)
- New Registrations: 12,340 devices
- Deregistrations: 8,920 devices
- Net Growth: +3,420 devices daily

Power Optimization Adoption:

- PSM (Power Saving Mode): 2,156,780 devices (75.7% adoption)
- eDRX (Extended Discontinuous Reception): 1,834,920 devices (64.4% adoption)

- Devices in Idle Mode: 87.3% (excellent power efficiency)
- Average PSM Timer: 24 hours (optimal for infrequent reporting applications)
- Average eDRX Cycle: 163.84 seconds (balanced between latency and power saving)

Coverage Enhancement Distribution:

- CE Level 0 (Good Coverage): 68.4% of devices
- CE Level 1 (Moderate Coverage): 24.3% of devices
- CE Level 2 (Deep Coverage): 7.3% of devices

Analysis: Strong power optimization adoption indicates successful battery life extension for IoT devices. The 7.3% of devices in deep coverage (CE Level 2) should be monitored for potential coverage improvements through additional base stations or antenna optimization.

Capacity Status:

- MME Capacity Utilization: 42% (ample headroom)
- Gateway Capacity Utilization: 45% (ample headroom)
- Estimated Days to Capacity: 120 days at current 2.8% weekly growth rate

Recommendations

Immediate (Next 24 Hours):

- 1.** Escalate MME-03 authentication failure investigation to Core engineering team
- 2.** Analyze HSS logs for authentication vector generation issues
- 3.** Verify device credential provisioning for affected IMSI ranges
- 4.** Implement temporary traffic offload from PGW-02 to PGW-01/03

Short-term (Next 7 Days):

- 5.** Conduct detailed performance analysis of S-GW-05 session setup latency
- 6.** Review recent MME-03 software updates or configuration changes that may correlate with authentication failures
- 7.** Implement automated alerting for authentication failure rate exceeding 3%
- 8.** Schedule capacity planning review for PGW SGi interfaces

Strategic (Next 30-60 Days):

- 9.** Plan PGW capacity expansion to address growing SGi utilization (add PGW-05)
- 10.** Evaluate HSS redundancy and failover mechanisms to prevent single-point authentication issues
- 11.** Analyze CE Level 2 device locations for potential coverage enhancement opportunities
- 12.** Continue monitoring device growth trend; current 120-day capacity runway provides adequate planning time for MME/Gateway expansion

Next Report: February 18, 2026

On-Call Engineer: Contact IoT Core NOC for escalations

This report demonstrates how generative AI transforms raw NB-IoT Core telemetry data into actionable business intelligence for network operations teams managing millions of IoT devices.

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 millions of IoT devices** with minimal manual effort

The same pattern can be applied across IoT domains:

- **NB-IoT Core Networks:** Device connectivity health summaries (as demonstrated)
- **Smart Utilities:** Smart meter data aggregation and anomaly detection
- **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 from millions of low-power devices, generative AI will become essential for extracting value and enabling rapid decision-making in Core network operations and device management.