MÄLARDALEN UNIVERSITY
FACULTY OF ENGINEERING AND HEALTH SCIENCES
DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
VÄSTERÅS, SWEDEN

Thesis for the Degree of Bachelor of Science in Computer Science - 15.0 credits

USING LLM TO GENERATE TRAFFIC DATASETS FOR CELLULAR NETWORKS

Zahra Faysal
Sfl23003@student.mdu.se

Examiner: Mohammad Ashjaei
Mälardalen University, Västerås, Sweden

Supervisor: Zenepe Satka
Mälardalen University, Västerås, Sweden

Augusti 20, 2026

Abstract

Access to real-world cellular network traffic data is often limited due to privacy concerns and security restrictions, which creates challenges for research, testing, and evaluation within wireless networking. Large Language Models have shown capabilities beyond traditional natural language processing, including the structured data generation. In this thesis, I investigate whether LLMs can be used to generate synthetic cellular network traffic datasets and how different prompt strategies and LLMs influence the quality of the generated data.

This study evaluates four prompting strategies: Basic, Constraint-Based, Chain-of-Thought, and Few-Shot prompting. The strategies are evaluated using two LLMs, GPT-4.1-mini and Claude Sonnet 5, multiple cellular network traffic datasets are generated containing metrics such as throughput, latency and timestamps. I evaluated the generated datasets using structural quality, descriptive statistics, distributional similarity, quantitative similarity and statistical comparison. Realism and representativeness are evaluated by comparing the generated datasets with measurements from the Glasgow Public and Private 5G Performance Dataset, using statistical measures such as Kolmogorov-Smirnov statistic, Wasserstein distance and quantile-based Mean Absolute Error.

The results show that both LLMs were able to generate structured cellular network traffic datasets, but their ability to follow the requested output structure differed. Constraint-Based and Few-Shot prompting generally produced datasets that were closer to the real-world reference dataset than Basic and Chain-of-Thought prompting. Constraint-Based prompting performed best for GPT-4.1-mini, while Few-Shot prompting performed best for Claude Sonnet 5. GPT-4.1-mini generally showed more consistent structural quality, while Claude Sonnet 5 also produced datasets with close similarity to the reference dataset. This shows that both the choice of LLM and prompting strategy influence the quality of the generated data and its similarity to the reference dataset.

Acknowledgement

I would like to express my sincere gratitude to Zenepe Satka for her continuous support throughout this thesis. Her help was invaluable. I would also like to give thanks to my examiner, Mohammad, for his helpful feedback and support. Finally, I would like to extend my appreciation to my family and support system, for their encouragement throughout this journey.

وما توفيقىٓ إلّا بٱللّه ۚ عليه توكّلت وإليه أنيب

Hud 11:88 ♡

Table of Contents

1. Introduction……4

2. Background……..6
2..1 Large Language Models, Capabilities and Limitations……..……..…..6
2.2 Pre-training, Fine-tuning and Prompt Engineering……6
2.3 Prompt Engineering Techniques…….....7
2.4 Wireless and Cellular network…….....8
2.5 Statistical Evaluation of Synthetic Data.…….....9
2.6 Distributional Similarity……..…….....9
2.7 Quantitative Similarity and Error Metric…….....9
2.8 Statistical Comparison of Generated Datasets…….....10

3. Related Work……....11

4. Problem Formulation……...…….13

5. Methodology.....14
5.1 Research Approach……...…….……14
5.2 Dataset Design……14
5.3 Prompt Engineering……...…….....15
5.4 Implementation……...16
5.5 Evaluation strategy…….20

6. Results.....21
6.1 Dataset Overview……...21
6.1.1 Dataset GitHub Availability…….21
6.2 Descriptive Statistics…...……...……22
6.3 Distributional Similarity…….……23
6.4 Quantitative Similarity …….…….……24
6.5 Comparison of Prompt Engineering……27
6.6 Statistical Comparison.…..…….…..……...……28
6.7 Overall Evaluation…...…….…..…….……29

7. Discussion 31
7.1 RQ1 – Can LLMs generate structured data that reflects important cellular network traffic metrics such as throughput and latency?.....31
7.2 RQ2 – What are the challenges of generating structured data using LLMs in the context of cellular network traffic?.....32
7.3 RQ3 – How similar are the datasets generated by different LLMs and prompting strategies to the real-world cellular traffic reference dataset?.....32
7.4 Practical Use of Methodology……34
7.5 Thesis Limitations…….…….34

8. Conclusions and Future Work ……36

References.....37

Appendix.....40

1. Introduction

Modern cellular networks, such as 4G and 5G systems, rely heavily on large amounts of data to analyze performance and improve overall system efficiency. This data is important for tasks such as network planning, traffic prediction and the evaluation of new algorithms and technologies [14]. However, access to real-world cellular network data is often limited. Network operators typically restrict access due to privacy and security concerns. As a result, researchers and developers face challenges when trying to test, validate, and improve their solutions in realistic environments [1].

To address this limitation, synthetic data generation and simulation-based approaches have become important alternatives in both academia and industry [12], [13]. Traditional methods for generating synthetic network data often rely on statistical models or simulation tools [1]. While these methods have been proven to be useful, they may not fully capture the complexity and variability of real-world network traffic, especially as modern cellular networks continue to become more advanced and diverse.

At the same time, recent improvements in Large Language Models (LLMs) have shown strong capabilities in generating different types of data, including structured and semi-structured data. Although LLMs are mainly designed for natural language processing tasks, their ability to learn patterns from large datasets creates new possibilities for data generation in other domains [1], [12], [13]. Recent research has also investigated the use of LLMs in wireless networking and cellular traffic data generation [2], [4]. However, using LLMs directly in wireless networking might still not be easy. Wireless networks require specific domain knowledge, such as signal propagation, protocol behavior and channel conditions. To generate realistic cellular traffic data, the model must therefore be guided through prompts containing network-related parameters and conditions.

Because of these challenges, prompt engineering can be used as a practical approach for adapting LLMs to this domain. Instead of fully retraining a model for each task, prompt engineering makes it possible to provide the model with network-specific instructions, parameters and traffic scenarios directly through textual inputs [2], [18]. As cellular traffic scenarios can vary a lot, this thesis aims to investigate whether prompt-guided LLMs could provide a more flexible way of generating structured synthetic datasets, without requiring task-specific retraining.

To investigate this, the study considers three aspects of LLM-based cellular network traffic generation. First, it is important to understand whether LLMs can generate structured data that follows the required format and contains important cellular network traffic metrics such as throughput and latency. Second, it is important to identify the challenges that can occur when using different LLMs and prompting strategies for this type of data generation. Finally, comparing the generated datasets with real-world cellular network data makes it possible to evaluate how similar the generated data is to actual network traffic.

Based on these aspects, the study addresses the following three research questions:

1. Can LLMs generate structured data that reflects important cellular network traffic metrics such as throughput and latency?

2. What are the challenges of generating structured data using LLMs in the context of cellular network traffic?
3. How similar are the datasets generated by different LLMs and prompting strategies to the real-world cellular traffic reference dataset?

To answer these research questions, the study evaluates four different prompt engineering strategies using GPT-4.1-mini and Claude Sonnet 5 and compares the generated datasets in terms of structural quality, descriptive statistics, distributional similarity, quantitative similarity, and statistical comparison. The results show differences between the prompting strategies and the two LLMs in terms of structural quality and similarity to the real-world reference dataset.

The remainder of the thesis is structured as follows. Section 2 presents the background and concepts relevant to the study, while Section 3 discusses related work on LLMs and synthetic data generation. Section 4 presents the problem formulation and research questions. Section 5 describes the methodology used to generate and evaluate the synthetic datasets. Section 6 presents the results, and Section 7 discusses the findings in relation to the research questions. Finally, Section 8 concludes the thesis and presents suggestions for future work.

2. Background

2.1 Large Language Models, Capabilities and Limitations

Large Language Models (LLMs) are a type of generative artificial intelligence trained on very large datasets, that are used to understand and generate human-like text [3]. They are designed to guess the next word or token in a sequence. Due to this training process, LLMs can learn patterns and do tasks such as summarizing information, answering questions, and following instructions.

In the last couple of years, LLMs have been used in more areas beyond traditional natural language processing. Their ability to recognize patterns, follow instructions and generate structured outputs has made them useful in different domains [2], [3]. Another capability or strength LLMs have is their commonsense understanding, where LLMs can answer questions and solve tasks based on knowledge that they have learned during training [10], [11]

While LLMs have shown capabilities in many domains, they also have limitations. One common issue they have is hallucination, where the model gives incorrect or misleading information that may seem convincing [20]. Their performance can also rely a lot on the prompt design and available context [3], [19]. LLMs generally perform better on tasks and information that are more represented in their training data, while performance may not be as good in less represented or domain-specific context [10]. Previous research has also shown that synthetic data generated by LLMs may not fully capture the diversity in real-world data distributions, which can affect the realism of the generated outputs [9].

2.2 Pre-training, Fine-tuning and Prompt Engineering

There are different ways of adapting LLMs for domain-specific applications.

The first stage is pre-training, where the model is trained from the beginning on large amounts of data. During this stage, the model learns general capabilities such as reasoning, comprehension and instruction following, this is what later becomes the foundation for many tasks [2]. However, pre-training requires large amounts of computational power, time and energy, which makes it expensive in practice. It also depends on powerful hardware such as Graphic Processing Units (GPUs) or Tensor Processing Units (TPUs), which makes it difficult to train a domain-specific model from zero [2].

Another approach is fine-tuning, where an already pre-trained model is further trained on smaller domain-specific datasets [5], [6], [18]. In wireless networking, this includes things as network logs, 5G datasets, or other traffic-related information. Compared to pre-training, fine-tuning takes less time and computational resources, which makes it a more practical approach for adapting LLMs to specific tasks [21]. Fine-tuning can also improve the model's performance in areas such as network troubleshooting, configuration, and traffic analysis.

A third approach is prompt engineering, which is the focus of this thesis. Instead of retraining or updating the model, prompt engineering guides a pre-trained LLM

through instructions or examples written directly in the prompt in order to generate the wanted outputs [6]. In this thesis, I focus on prompt engineering because it provides a more flexible way of adapting LLMs to different tasks without modifying the model.

Compared to pre-training and fine-tuning, prompt engineering has lower computational cost and can be implemented more quickly. These advantages are among the reasons why prompt engineering was chosen for the synthetic generation of cellular traffic datasets in this thesis. However, despite the flexibility, prompt engineering is still dependent on the limitations of the pre-trained model, especially its context window and reasoning capabilities [2].

2.3 Prompt Engineering Techniques

In this thesis, different prompt engineering techniques are explored to investigate how they affect the structure, consistency of generated cellular traffic datasets and their similarity to a real-world reference dataset.

One technique is in-context learning, where examples are included directly in the prompt to help the model better understand the task [2], [7]. Instead of only giving instructions, the model is also shown examples of the expected output format, which can help it generate more accurate responses for similar tasks.

Example of such a prompt:

```
"You are solving addition problems like:
2 + 2 = 4
5 + 3 = 8
10 + 7 = 17

Now solve:
12 + 6 = ?"
```

In this example, the model learns the expected pattern from the provided examples before generating a new answer. In this this technique is used as inspiration for structuring prompts that guide the model to generate synthetic cellular traffic data, in a more consistent format.

Another technique is Chain-of-Thought (CoT) prompting, where the reasoning process is explained step by step in the prompt [8]. This allows the model to break down problems into smaller reasoning steps before generating a final answer.

Example of such a prompt:

```
"A train travels 60 km in 1 hour. How far will it travel
in 3 hours?
Think step by step before giving the final answer"
```

This technique encourages the model to reason through the task before producing the final output. In this thesis, this approach is investigated to determine whether this type of structured reasoning can also improve the similarity to the reference dataset and consistency of generated traffic datasets. However, CoT prompting may sometimes generate incorrect reasoning steps, which can affect the final output quality [2]. Despite this limitation, it is still useful for tasks that involve planning or multi-step reasoning.

Self-refinement prompting is another technique where the model improves its own output through an iterative process [2]. In this approach, the model first generates an initial response and then refines it step by step based on previous outputs.

Example of such a prompt:
`Write a short explanation of how solar panels work. Then review your answer and improve any unclear parts`

Through the iterative process, LLMs can improve their outputs and generate more consistent results over time. Iterative prompting strategies can also be used to investigate whether they improve the structure and quality of generated traffic datasets**.** However, including previous output in each iteration, can make the prompt longer and increase the computational cost [10].

2.4 Cellular Networks

Cellular networks have developed significantly to support the increasing number of connected devices, larger data volumes, and higher data rate requirements. 5G networks were introduced to address the limitations in previous generations, especially 4G, with three main features: low latency, high-speed data transfer, and improved connectivity [17]. These improvements make it possible to support applications like IoT systems, real-time communication, and other time-sensitive services.

To support these requirements, 5G relies on technologies such as massive Multiple Input Multiple Output (MIMO), device-to-device communication, and cloud-based network functions including Software-Defined Networking (SDN) and Network Function Virtualization (NFV) [16], [17]. Wireless networks also face challenges such as interference, mobility management ,and security, which make monitoring and traffic data analysis important aspects of network design.

A detailed analysis of these technologies and challenges falls beyond the scope of this thesis. Instead, I focus only on the concepts necessary to understand how synthetic cellular traffic datasets can be generated and evaluated using LLMs.

Wireless and cellular networks generate large amounts of data due to the high number of devices and continuous communication between users and base stations. In 5G networks, this traffic is mainly driven by the increased data demand, higher transmission speeds, and massive device connectivity [17].

Key metrics generally used to evaluate wireless network performance include throughput, latency, packet loss, and spectrum usage [14], [16]. Throughput describes how much data the network can handle during a certain amount of time, while latency refers to the delay that occurs when data travels from one point to another, which is especially important in real-time applications [14].

Packet loss means that some data packets do not successfully reach their destination, which can reduce the network reliability. Spectrum usage describes how efficiently the available frequencies are being used [14], [17].

These metrics are important for understanding network behavior and evaluating performance under different conditions. In this thesis, the main focus is therefore on

throughput and latency, since they are widely used indicators of cellular network performance and were the metrics available in the real-world dataset that was later used as a reference for evaluation.

2.5 Statistical Evaluation of Synthetic Data

Synthetic data can be evaluated to see if the generated data has a valid structure, but it can also be evaluated to see if the structural characteristics are similar to real-world data. Descriptive statistics such as mean, median, standard deviation, and interquartile range (IQR) can be used to describe the central values as well as the variation in a dataset [22].

These measures are used in this study to give an overview of the characteristics of a dataset. However, they do not describe how the values are distributed. For example, two datasets can have similar average values but still have different distributions. Therefore, distributional and similarity measures can give more information when evaluating synthetic data against real-world data [23].

2.6 Distributional Similarity

Distributional and similarity measures help to describe how values are spread across a dataset, and comparing distributions can give a better understanding of the similarity between two datasets than only looking at average values and other descriptive statistics.

The ***Kolmogorov-Smirnov (KS) statistic*** measures the largest difference between the cumulative distributions of two datasets. A lower KS value means that the distributions are more similar [24].

The ***Wasserstein distance*** is also a measure that can be used to measure the difference between two distributions[25]. It can be understood as the distance between the values in two distributions. Similarly to the KS statistic, a lower Wasserstein value means that the distributions are closer to each other.

In this study, both measures are used to compare the distributions of generated throughput and latency with the corresponding distributions in the real-world reference dataset.

2.7 Quantitative Similarity and Error Metrics

Quantitative similarity measures can be used to assess how close the values in two datasets are. When comparing generated and real-world datasets, the rows do not always correspond to each other, meaning that a direct row-by-row comparison is not always a good method.

Instead, corresponding ***quantiles*** can be compared between the generated and the real-world reference datasets [26]. This makes it possible to compare different parts of their distributions.

Mean Absolute Error (MAE) can then be used to measure the average absolute difference between the corresponding quantile values [27]. Similarly, a lower MAE means that the difference between the generated and reference data is smaller.

In this thesis, quantile-based MAE is used as an additional measure of similarity for throughput and latency.

2.8 Statistical Comparison of Generated Datasets

Statistical tests can be used to understand if there are significant differences between groups of data. In this thesis, these tests are used to compare datasets generated using different prompting strategies and LLMs, using p-values. The p-value is used indicate the strength of the evidence for a difference between the groups. A lower p-value provides stronger evidence against the assumption that the observed difference is due to random variation.

The ***Kruskal-Wallis test*** is used to compare more than two independent groups [28]. Later in this study, it is used to compare the four prompting strategies within each LLM. A significant result means that at least one strategy differs from the others.

The ***Mann-Whitney U test*** is used to compare two independent groups [29]. In this study, it is used to compare GPT-4.1-mini and Claude Sonnet 5 for the same prompting strategy. When multiple statistical tests are performed, the probability of finding a significant result by chance can increase. To account for this, a Holm correction can be applied to the p-values from multiple comparisons [32]. The p-values are ordered from smallest to largest and adjusted sequentially based on their position in the ordered set [32].

Together, these tests and the Holm correction help to evaluate whether the observed differences between prompting strategies and LLMs are statistically significant.

3. Related Work

Research on synthetic data generation using LLMs has grown a lot in recent years, leading to several studies and survey papers investigating different methods and applications. Synthetic data has been explored as a possible solution to challenges related to privacy concerns and limited accessibility to real-world datasets.

To better understand the current research related to this thesis, previous studies were reviewed from commonly used databases such as IEEE Xplore and Scopus, using combinations of keywords such as “LLM”, “synthetic data generation” and “cellular network”. Although several studies related to synthetic data generation with LLMs came up, there was very limited research specifically focusing on prompt engineering for synthetic cellular network traffic data generation.

Wang et al. [12] present a survey of data synthesis and augmentation techniques for LLMs. The study discusses how synthetic data can be used during different stages of the LLM lifecycle, including pre-training, fine-tuning, and instruction tuning. The authors also highlight challenges related to data quality and evaluation methods in synthetic data generation. This study provides an overview of how LLM-based synthetic data generation is commonly approached and evaluated. However, the main focus of the paper is on natural language processing (NLP), with limited discussion related to cellular network traffic generation.

Another related study by Long et al. [13] reviews previous work on LLM-driven synthetic data generation and evaluation. Their work focuses on how synthetic data can help address limitations in real-world datasets and presents different approaches for evaluating generated data. The paper provides an overview of common workflows and evaluation methods used in synthetic data generation. However, the study mainly focused on general synthetic data generation and does not investigate cellular traffic dataset or the impact of prompt engineering strategies.

More specifically related to cellular traffic synthesis, Duan et al. [4] propose LLMCell, a framework designed for generating synthetic cellular traffic data using LLMs. Their approach combines textual encoding, LLM fine-tuning and synthetic data sampling to generate cellular traffic data. The generated datasets are evaluated based on factors such as fidelity and utility and compared against existing baseline methods. This work is closely related to the topic investigated in this thesis and provides an example of how LLMs can be applied to cellular traffic generation. However, while their work mainly relies on fine-tuning, this thesis focuses on prompt engineering strategies and investigate whether prompt-guided LLMs can generate realistic structured datasets without task specific retraining

Zhou et al. [14] investigate the use of LLM-enabled-in-context learning for wireless network optimization. Their work focuses on how prompting and in-context learning can support optimization tasks in wireless networks without the need of model fine-tuning. Although the study is not related to synthetic data generation, it provides an example of how prompt-based approaches can be applied within wireless and cellular networking domains.

Another paper that was important for this thesis was the Glasgow Public and Private 5G Performance Dataset [15]. Unlike the studies discussed above, this study does not focus on LLMs or synthetic data generation. Instead, it contains real-world 5G network measurements collected across different locations, devices, and network conditions. The dataset includes metrics such as throughput, latency, signal strength, and timestamps. Since access to real cellular network traffic data is limited, this dataset was used as the real-world reference dataset for comparison in this thesis. The Glasgow study contains a large number of real-world network measurements and provides a reference for evaluating the similarity of the generated datasets to real-world cellular network data.

In contrast to previous studies, this thesis focuses specifically on the use of prompt engineering strategies for generating cellular network traffic datasets using LLMs. Rather than focusing on fine-tuning or general synthetic data generation, the study investigates how different prompting approaches can influence the structure, consistency, and similarity of generated cellular traffic datasets to a real-world reference dataset.

4. Problem Formulation

As cellular networks continue to grow, become more advanced and generate more traffic data, the need for realistic traffic datasets for research, testing, and evaluation has also increased. However, the access to the real-world cellular traffic datasets is most of the time limited due to privacy concerns and security restrictions.

Earlier research has shown that LLMs can generate structured data in different domains. LLMs have also been explored for some different tasks in the networking domain, such as network traffic prediction and optimization. However, the use of LLMs for generating synthetic cellular traffic datasets is still relatively unexplored, especially in the context of prompt engineering and structured traffic generation.

Therefore, to address this gap, this thesis is to investigate whether LLMs can be used to generate structured synthetic cellular traffic data that reflects important characteristics of wireless networks. In particular, the thesis investigates how different prompt engineering strategies and LLMs can influence the quality of the generated datasets and their similarity to real-world cellular traffic data. In this thesis, quality refers to the structural and statistical characteristics of the generated datasets. Similarity refers to how closely the generated data resembles the real-world cellular network measurements in the reference dataset, while diversity refers to the variation present in the generated data and its ability to represent different network conditions.

The study focuses on the following research questions:

RQ1. Can LLMs generate structured data that reflects important cellular network traffic metrics such as throughput and latency?

RQ2. What are the challenges of generating structured data using LLMs in the context of cellular network traffic?

RQ3. How similar are the datasets generated by different LLMs and prompting strategies to the real-world cellular traffic reference dataset?

The aim with this thesis is to provide insights into how LLM-based approaches can be applied for synthetic cellular traffic generation and to evaluate whether prompt-guided LLMs can provide a flexible alternative for generating structured network datasets for future research and testing purposes.

5. Methodology

This following chapter presents the methodology used in this thesis, outlining the overall research approach for generating and evaluating synthetic cellular network traffic data using LLMs.

5.1. Research Approach

This thesis follows a prompt-based approach to investigate whether LLMs can be used to generate structured synthetic traffic datasets for cellular networks. Instead of training or fine-tuning a model, the study focuses on prompt engineering techniques to guide two pre-trained LLMs to generate data that reflects expected network behavior.

In this thesis, I use two LLMs, OpenAI GPT-4.1-mini [30] and Claude Sonnet 5 [31], to investigate whether LLMs can generate structured synthetic datasets for cellular networks through different prompting strategies.

The study is exploratory and aims to understand how different prompt design strategies and LLMs affect the quality of the generated data. In particular, the study compares different prompt types and LLMs to examine how they affect aspects such as structural quality, statistical characteristics, and similarity to a real-world reference dataset.

The overall process consists of generating synthetic data using two LLMs and four prompting strategies, converting the output into a structured format, and then evaluating the dataset using structural validation, descriptive statistics, distributional similarity, quantitive similarity and statistical comparison.

5.2. Dataset Design

To simulate cellular network traffic, a set of key features that are commonly used in wireless network analysis is defined [16]. These features are chosen based on standard performance metrics used in cellular networks in order to represent both user behavior and network conditions.

The generated dataset consists of the following attributes:

- **user_id**: Unique identifier for each user
- **timestamp**: The time when the data sample was recorded
- **call_id**: The identifier of the base station that served the user
- **throughputs_mbs**: The downlink data rate experienced by the user, measured in mbps
- **latency_ms**: The time delay in data transmission, measured in milliseconds.

To make the generated data more realistic, expected relationships between different features were included in the prompt design. For example, 5G networks are generally expected to provide higher throughput and lower latency compared to 4G networks, while weaker signal strength is often associated with increased latency and higher packet loss. These relationships were used when designing the prompts and later during the evaluation process.

5.3. Prompt Engineering

Prompt Engineering is the main method used in this study to guide the output generated by the LLM. By designing different types of prompts, the study investigates how variations in prompt design influence the quality and similarity to the real-world reference dataset of the generated cellular network data.

The different prompt engineering strategies that are used in this study:

- **Basic Prompt:**
  A simple instruction where the model is asked to generate cellular network data with minimal guidance. This prompt is used as a baseline for comparison with the other prompting strategies.
- **Constraint-based Prompt:**
  This prompt includes predefined rules that help guide the model to generate more realistic data. For example, it sets limits for values like throughput and latency and makes sure that 5G performs better than 4G.
- **Few-shot Prompt:**
  Example data rows are included in the prompt to show the model what the output and structure should look like. This strategy is used to investigate whether examples can help the model generate more realistic traffic patterns.
- **Chain-of-Thought Prompt:**
  This method encourages the model to think step by step about the relationships between different network features before generating the final dataset.

By comparing these prompting strategies, the study investigates how prompt design affects **the** consistency, structure, and similarity of the generated datasets to the real-world reference dataset**.**

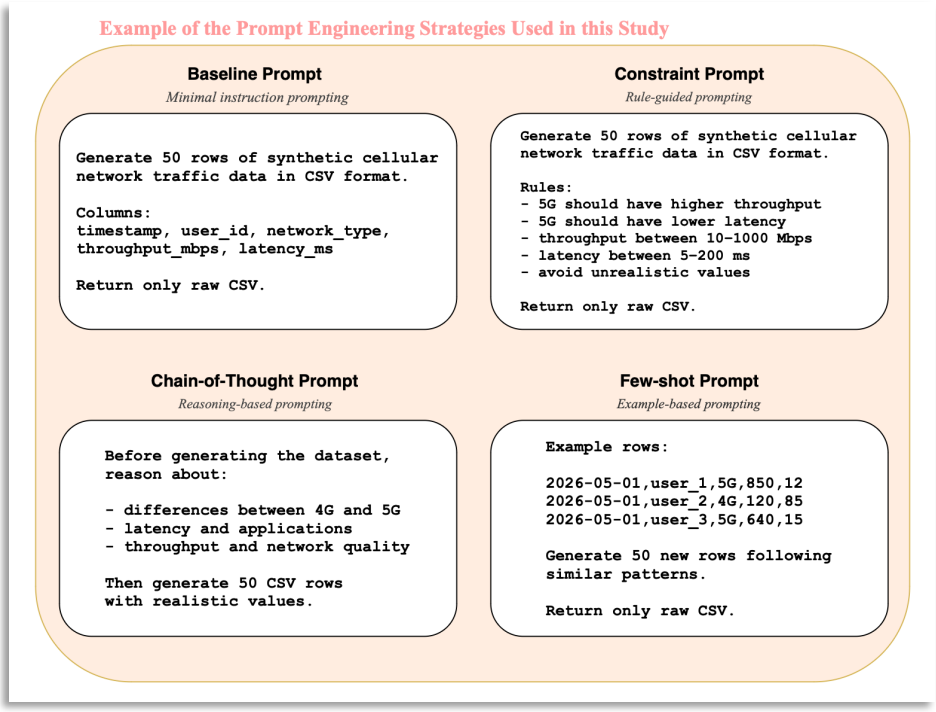

Figure 1: Overview of the prompt engineering strategies used in this study.

Baseline prompt uses simple instructions with minimal guidance, while Constraint-based prompt includes rules and limitations. Few-shot prompt contains example rows, and Chain-of-Thought prompt encourages step-by-step reasoning before generating the dataset.

5.4. Implementation

The implementation is developed in Python and uses the OpenAI and Anthropic APIs together with the GPT-4.1-mini and Claude to interact with pre-trained LLMs. The overall implementation follows a pipeline for generating, processing, and storing synthetic cellular network data.

The process consists of the following steps:

1. A prompt is created based on the selected prompting strategy.
2. The prompt is sent to the LLM through the API.
3. The model generates synthetic cellular network data as text output.
4. The output is then processed in Python and converted into a structured format.
5. The generated data is saved as a CSV file for later analysis and evaluation.

The same processing and evaluation pipeline is used for all prompt strategies and both LLMs, as it makes it easier to compare the generated datasets and analyze how different prompting strategies and LLMs affect the results.

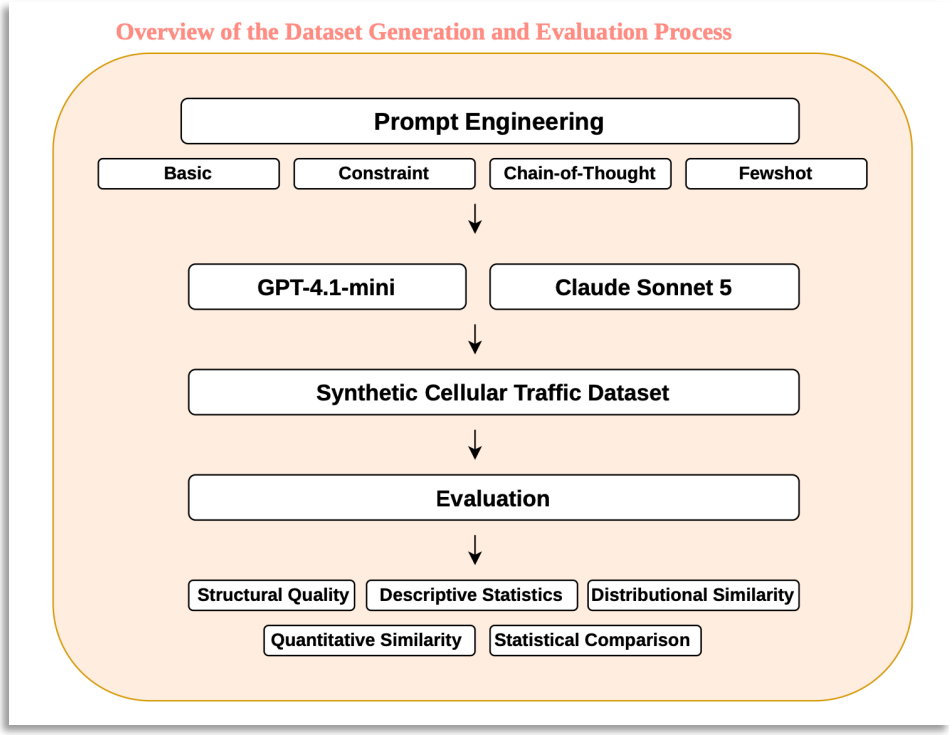

Figure 2: Overview of the dataset generation and evaluation process used in this study.

5.4.1. Prompting Strategies

The exact prompts used for dataset generation are shown in Figures 3–6 down below. The same prompt structures were used for both GPT-4.1-mini and Claude Sonnet 5. The prompts correspond to the four prompting strategies described in Section 5.3.

```
prompt = """
Generate synthetic 5G cellular network traffic data in CSV format.

Columns:
timestamp,user_id,network_type,throughput_mbps,latency_ms

Generate exactly 500 rows.

Return only raw CSV.
Do not use markdown.
"""
```

Figure 3: Exact Basic Prompt used for synthetic cellular traffic data generation.

```
prompt = """
Generate synthetic 5G cellular network traffic data in CSV format.

Columns:
timestamp,user_id,network_type,throughput_mbps,latency_ms

Requirements:

- network_type must always be 5G
- throughput_mbps must be between 10 and 1000
- latency_ms must be between 5 and 200
- values should be realistic for 5G cellular networks
- no missing values

Generate exactly 500 rows.

Return only raw CSV.
Do not use markdown.
"""
```

Figure 4: Exact Constraint-Based Prompt used for synthetic cellular traffic data generation.

```
prompt = """
Generate synthetic 5G cellular network traffic data in CSV format.

Columns:
timestamp,user_id,network_type,throughput_mbps,latency_ms

Before generating the dataset:

1. Consider realistic throughput values in 5G networks.
2. Consider realistic latency values in 5G networks.
3. Consider realistic variation between users.
4. Use this reasoning internally when generating the data.

Generate exactly 5 rows.

Return only raw CSV.
Do not include explanations.
Do not use markdown.
"""
```

Figure 5: Exact Chain-of-Thought Prompt used for synthetic cellular traffic data generation.

```
prompt = """
Generate synthetic 5G cellular network traffic data in CSV format.

Columns:
timestamp,user_id,network_type,throughput_mbps,latency_ms

Example rows:

timestamp,user_id,network_type,throughput_mbps,latency_ms
2025-04-06T08:30:00Z,user_1,5G,907,30
2025-04-06T08:32:00Z,user_2,5G,557,38
2025-04-06T08:33:00Z,user_3,5G,879,15

Generate exactly 5 new rows following similar patterns.

Return only raw CSV.
Do not use markdown.
"""
```

Figure 6: Exact Few-Shot Prompt used for synthetic cellular traffic data generation.

The four prompting strategies were applied to GPT-4.1-mini and Claude to generate synthetic cellular traffic datasets, which were then evaluated using structural validation, descriptive statistics, distributional similarity, quantitative similarity, and statistical comparison.

5.4.2. Python Data Processing and Analysis

After each generation run, the generated output was stored as a CSV file. The generated datasets were organized according to the LLM and prompting strategy, with each generation run stored as a separate CSV file. This made it possible to process each run individually and compare the results across prompting strategies and LLMs.

The generated CSV files were processed using Python. The analysis scripts were implemented using pandas for loading and processing the datasets, NumPy for numerical calculations, SciPy for statistical methods, and Matplotlib for generating distribution plots.

The first step was to check the structural quality of the generated datasets. For each run, the number of returned rows was recorded and compared with the requested 500 rows. The expected columns were checked, and the datasets were examined for missing values, duplicate rows, invalid network types, invalid throughput and latency values, and invalid timestamps.

After the structural checks, the throughput and latency values were processed as numerical data and used for the statistical analyses. Descriptive statistics were calculated for each generated dataset, including the mean, median, standard deviation, minimum, maximum, quartiles, and interquartile range

The generated datasets were then compared with the Glasgow reference dataset. The download speed and ping measurements from the reference dataset were used as the reference values for throughput and latency. The generated and reference data were compared using distributional similarity measures and quantile-based MAE. Distribution plots were also generated to visually compare the distributions.

Finally, the run-level results were used for statistical comparison. The Kruskal-Wallis test was used to compare the four prompting strategies within each LLM, while the Mann-Whitney U test was used to compare GPT-4.1-mini and Claude Sonnet 5 for the same prompting strategy. Holm correction was then applied to the resulting p-values to account for multiple comparisons [32].

The same processing and analysis pipeline was used for all prompting strategies and both LLMs. The results from the different analyses are presented in the Results section.

5.5. Evaluation Strategy

To evaluate the generated datasets, several evaluation methods are used to check their structural quality, statistical characteristics and similarity to real-world cellular traffic data. The generated datasets are compared with the Glasgow Public and Private 5G Performance Dataset [15], which is used as the real-world reference dataset. The evaluation focuses mainly on throughput and latency, since these metrics are available in both the generated and reference dataset.

The evaluation analyses were performed using Python scripts to calculate the different statistical measures and compare the generated datasets with the reference dataset.

The evaluation includes the following parts:

- **Structural Quality Assessment**
  The generated datasets are checked to see if they follow the requested output format and dataset size. This includes checking the number of returned rows, duplicate rows, expected columns, missing values, invalid values, and formatting errors. This is used to evaluate how reliably the LLMs generate structured cellular network traffic data.
- **Descriptive Statistics**
  Descriptive statistics are used in this thesis to summarize the main characteristics and the variation of the generated datasets. Measures such as mean, median, standard deviation, and interquartile range are calculated for throughput and latency. These values are also compared across the generated datasets as well as the reference dataset.
- **Distributional Similarity**
  The distributions of throughput and latency are then compared between the generated datasets and the reference dataset. The *Kolmogorov-Smirnov (KS) statistic* [24] and *Wasserstein distance* [25] are used to measure the differences between the distributions. Lower values mean that the generated and reference distributions are more similar.
- **Quantitative Similarity**
  The quantitative similarity is evaluated by comparing corresponding quantiles of the generated and reference datasets. *Quantile-based Mean Absolute Error (MAE)* [27] is used to measure the difference between these values for throughput and latency. A lower MAE means that the generated data is closer to the reference data.
- **Statistical Comparison**
  Statistical tests are used to compare the generated datasets across prompting strategies and LLMs. The *Kruskal-Wallis* [28] test is used to compare the four prompting strategies within each LLM, while the *Mann-Whitney U* [29] test is used to compare GPT-4.1-mini and Claude for the same prompting strategy. These tests are used to determine whether the differences between the datasets are likely to be due to actual differences rather than random variation.

6. Results

6.1. Dataset Overview

To investigate how different prompt engineering strategies influence the generation of synthetic cellular network traffic data, four prompting approaches were evaluated: Basic Prompting, Constraint-Based Prompting, Chain-of-Thought (CoT) Prompting, and Few-Shot Prompting.

For each prompting strategy and LLM, there were 10 generation runs, with 500 synthetic network traffic rows requested in each run. This resulted in 5,000 requested rows per strategy and a total of 20,000 requested rows per LLM. In total, 80 datasets were generated using two LLMs, GPT-4.1-mini and Claude Sonnet 5. Multiple generation runs were used to examine whether similar patterns appeared across different runs rather than relying on a single generated dataset for each strategy.

To ensure a fair comparison between the four strategies, the generation settings were kept the same within each LLM. The same output format, feature definitions, and generation requirements were used for all prompting strategies. The only factor that changed between experiments within each LLM was the prompt engineering strategy. The same four prompting strategies were then applied to both GPT-4.1-mini and Claude Sonnet 5, which made it possible to compare the results between the two LLMs.

For each generation run, the prompt strategy, number of requested rows, and number of returned rows were documented. This information was used to organize the generated datasets and compare the results across the different strategies and LLMs.

These generated datasets were used in the following sections to evaluate their structural quality, statistical characteristics, similarity to the real-world reference dataset, and differences between prompting strategies and LLMs.

6.1.1 Dataset Availability

The generated datasets, prompt templates, and analysis scripts used in this study have been made publicly available through a GitHub repository:

https://github.com/S4hr4/Thesis-LLM-cellular-traffic-generation.git

The repository contains:

- Generated datasets for all four prompting strategies
- Prompt templates used during dataset generation
- Python scripts used for dataset analysis
- Evaluation results produced during the study

6.2. Structural Quality

When it comes to evaluating the structural quality of the generated datasets, the analysis focused on whether the generated data followed the requested structure and dataset size. This included checking the number of returned rows, duplicate rows, missing values, invalid values, and invalid timestamps. The same validation checks were applied to all generated datasets from both GPT-4.1-mini and Claude Sonnet 5.

For each prompting strategy, 10 generation runs were performed with 500 rows requested in each run. Table 1 shows the structural quality results for the two LLMs and the four prompting strategies**.** The results show differences between the two LLMs in the number of rows returned. GPT-4.1-mini generally followed the requested dataset size more closely than Claude Sonnet 5.

For GPT-4.1-mini, Constraint-Based prompting returned exactly 500 rows in all 10 runs. Few-Shot prompting also returned a number of rows very close to the requested size, with an average of 497.3 rows per run. Basic prompting returned an average of 460 rows, while Chain-of-Thought prompting returned the lowest average with 418 rows per run.

Claude Sonnet 5 returned fewer rows for all four prompting strategies. Table 1 shows that Few-Shot prompting returned the highest average with 372 rows per run, followed by Basic prompting with 341.2 rows and Constraint-Based prompting with 339.1 rows. Chain-of-Thought prompting returned the lowest average with 236.5 rows per run.

The validation checks also show differences between the two LLMs. GPT-4.1-mini did not produce any missing values, duplicate rows, invalid network types, invalid throughput values, invalid latency values, or invalid timestamps in the generated datasets. Claude Sonnet 5, however, produced some missing values, invalid network types, and invalid timestamps in several runs. No duplicate rows or invalid throughput and latency values were found in the Claude datasets either.

Overall, Table 1 shows that both LLMs were able to generate structured cellular network traffic data, but GPT-4.1-mini followed the requested structure and dataset size more consistently. For GPT-4.1-mini, Constraint-Based and Few-Shot prompting returned values closest to the requested number of rows, while Few-Shot prompting returned the highest number of rows for Claude Sonnet 5.

Table 1. Structural quality results for GPT-4.1-mini and Claude Sonnet 5.

LLM	Strategy	Average Rows Returned	Missing Values	Duplicate Rows	Invalid Network Types	Invalid Throughput	Invalid Latency	Invalid Timestamps
GPT	Basic	460.0	0	0	0	0	0	0
GPT	Constraint	500.0	0	0	0	0	0	0
GPT	CoT	418.0	0	0	0	0	0	0
GPT	Few-Shot	497.3	0	0	0	0	0	0
Claude	Basic	341.2	29	0	1679	0	0	6
Claude	Constraint	339.1	31	0	9	0	0	4
Claude	CoT	236.5	31	0	2232	0	0	12
Claude	Few-Shot	372.0	34	0	738	0	0	4

6.3. Descriptive Statistics

To further analyze the generated datasets, descriptive statistics were calculated for throughput and latency. The generated values from GPT-4.1-mini and Claude Sonnet 5 with the Glasgow reference dataset. The analysis included the mean, median, standard deviation, and interquartile range (IQR) to see how the values and their variation differed between the datasets.

Table 2 presents the descriptive statistics for all generated datasets and the Glasgow reference dataset. For GPT-4.1-mini, the results show differences between the four prompting strategies. Constraint-Based and Few-Shot prompting generally produced throughput and latency values that were closer to the Glasgow reference dataset, while Basic and Chain-of-Thought prompting showed larger differences. This was more visible for throughput, where Basic and Chain-of-Thought produced lower values compared to the reference dataset.

The results for Claude Sonnet 5 also show differences between the prompting strategies. As shown in Table 2, Few-Shot and Constraint-Based prompting generally produced values that were closer to the reference values, while Basic and Chain-of-Thought prompting showed larger differences. However, the Claude datasets also showed more variation in some cases, especially for throughput and latency.

The mean values in Table 2, provide an overview of the central values in each dataset, while the standard deviation and IQR show how much the generated values varied. Looking at these measures together makes it possible to examine not only if the average values were close to the reference dataset, but also if the generated data had a similar level of variation.

Overall, Table 2 shows that the prompting strategies produced different statistical characteristics. Constraint-Based and Few-Shot prompting generally produced values that were closer to the Glasgow reference dataset for both LLMs, while Basic and Chain-of-Thought prompting had larger differences. However, descriptive statistics alone do not show the full distribution of the data, which is why the generated and reference distributions were compared further in the next section.

Table 2. Descriptive statistics for the generated datasets and the Glasgow reference

LLM	Strategy	Metric	Mean	Median	SD	IQR
GPT	Basic	Throughput	107.50	107.71	11.00	19.50
GPT	Basic	Latency	12.04	11.85	2.11	3.32
GPT	Constraint	Throughput	546.07	537.89	212.74	355.69
GPT	Constraint	Latency	18.78	17.20	8.20	11.15
GPT	CoT	Throughput	195.93	193.26	47.44	73.69
GPT	CoT	Latency	10.55	10.46	2.44	3.72
GPT	Few-Shot	Throughput	773.98	778.10	92.09	160.65
GPT	Few-Shot	Latency	23.89	23.60	4.17	6.20
Claude	Basic	Throughput	312.81	316.12	236.45	422.08
Claude	Basic	Latency	18.07	12.82	11.30	19.02
Claude	Constraint	Throughput	463.44	454.13	263.39	479.00
Claude	Constraint	Latency	22.23	17.58	14.02	19.00

LLM	Strategy	Metric	Mean	Median	SD	IQR
Claude	CoT	Throughput	339.03	268.07	237.68	335.54
Claude	CoT	Latency	12.99	11.02	6.84	10.10
Claude	Few-Shot	Throughput	556.16	596.90	280.98	425.65
Claude	Few-Shot	Latency	27.29	24.20	14.14	18.15
Glasgow Reference	—	Throughput	694.18	731.84	295.09	494.56
Glasgow Reference	—	Latency	21.36	20.68	9.98	17.87

6.4. Distributional Similarity

To compare the generated datasets with the Glasgow reference dataset, the distributions of throughput and latency were analyzed. The *Kolmogorov-Smirnov (KS) statistic* and *Wasserstein distance* were used to measure the differences between the generated and reference distributions. Lower values indicate that the generated distribution is closer to the reference distribution.

Table 3 presents the KS statistic and Wasserstein distance for throughput and latency for all prompting strategies and both LLMs. For GPT-4.1-mini, Constraint-Based prompting produced the closest throughput distribution to the Glasgow reference dataset, with a KS value of 0.327 and a Wasserstein distance of 166.40. Few-Shot prompting followed with a KS value of 0.398 and a Wasserstein distance of 180.01. Basic and Chain-of-Thought prompting both had larger differences, with KS values of 0.980 and 0.880, respectively.

For Claude Sonnet 5, Table 3 shows that Few-Shot prompting produced the closest throughput distribution, with a KS value of 0.248 and a Wasserstein distance of 138.90. Constraint-Based prompting was the second closest, with a KS value of 0.360 and a Wasserstein distance of 230.98. Basic and Chain-of-Thought prompting showed larger differences from the reference distribution.

The latency distributions show a similar overall pattern, although the differences between some strategies were smaller. As shown in Table 3, GPT-4.1-mini, Constraint-Based and Few-Shot prompting produced distributions that were closer to the reference dataset in comparison to Basic and Chain-of-Thought prompting. For Claude Sonnet 5, Few-Shot and Constraint-Based prompting also produced the closest latency distributions, while Chain-of-Thought showed the largest difference.

The distribution plots in Figures 3−6, also show these same differences visually. The generated distributions from Constraint-Based and Few-Shot prompting generally display more overlap with the Glasgow reference distributions, while Basic and Chain-of-Thought prompting show larger differences. Four examples are included in this section, as for the remaining distribution plots, they are all in the Appendix.

Overall, Table 3 and Figured 3−6 show that Constraint-Based and Few-Shot prompting generally produced distributions that were closer to the real-world reference dataset compared to Basic and Chain-of-Thought prompting. However, the results also show some differences between GPT-4.1-mini and Claude Sonnet 5, especially for throughput.

Table 3. KS statistic and Wasserstein distance between the generated and reference distributions.

LLM	Strategy	Throughput KS	Throughput Wasserstein	Latency KS	Latency Wasserstein
GPT	Basic	0.980	586.99	0.641	9.85
GPT	Constraint	**0.327**	**166.40**	***0.297***	***5.34***
GPT	CoT	0.880	500.39	0.673	10.97
GPT	Few-Shot	**0.398**	**180.01**	0.398	5.78
Claude	Basic	0.625	413.78	0.311	5.36
Claude	Constraint	**0.360**	**230.98**	0.221	5.82
Claude	CoT	0.579	357.71	0.430	8.43
Claude	Few-Shot	**0.248**	**138.90**	***0.209***	***5.93***

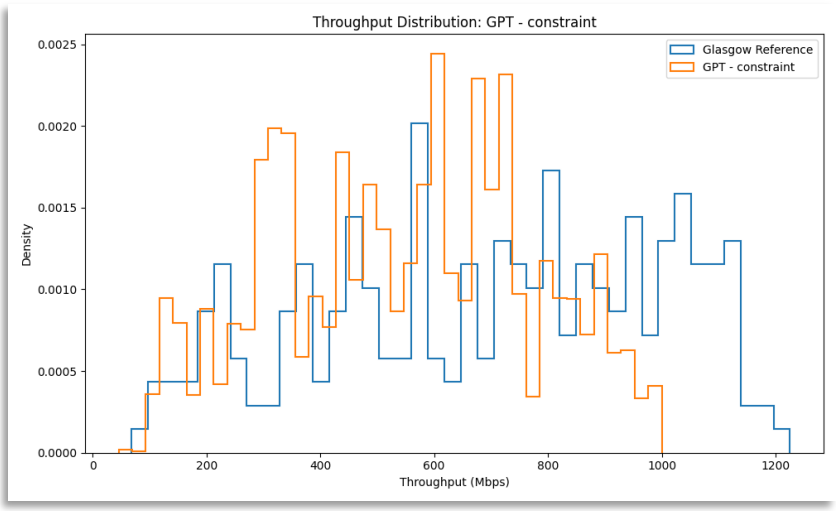

Figure 3: GPT-4.1-mini with Constraint-Based prompting: throughput distribution compared with the Glasgow reference dataset.

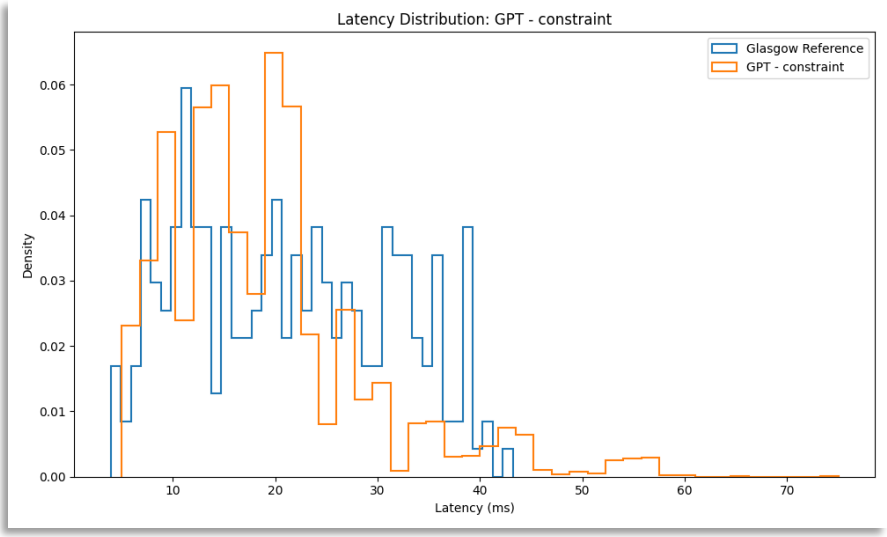

Figure 4: GPT-4.1-mini with Constraint-Based prompting: latency distribution compared with the Glasgow reference dataset.

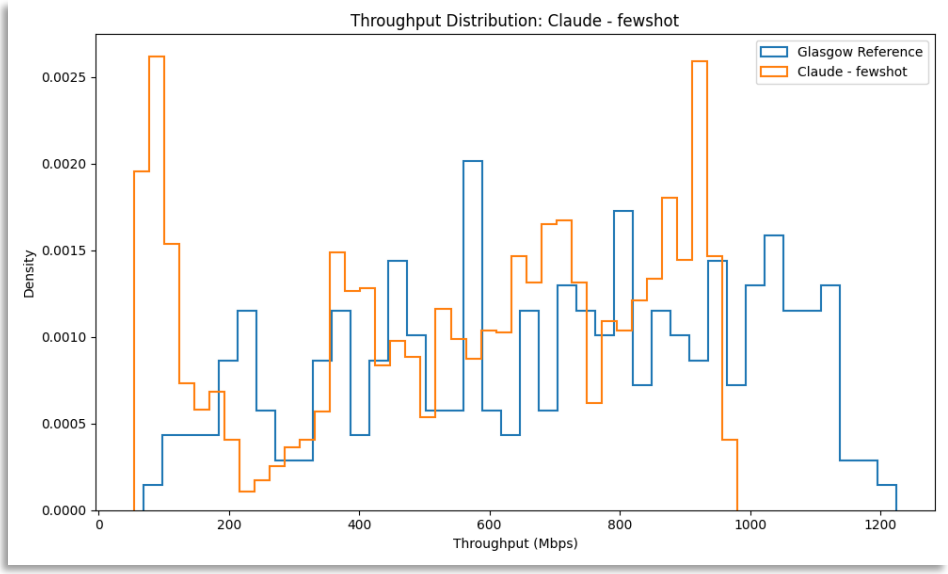

Figure 5: Claude Sonnet 5 with Few-Shot prompting: throughput distribution compared with the Glasgow reference dataset**.**

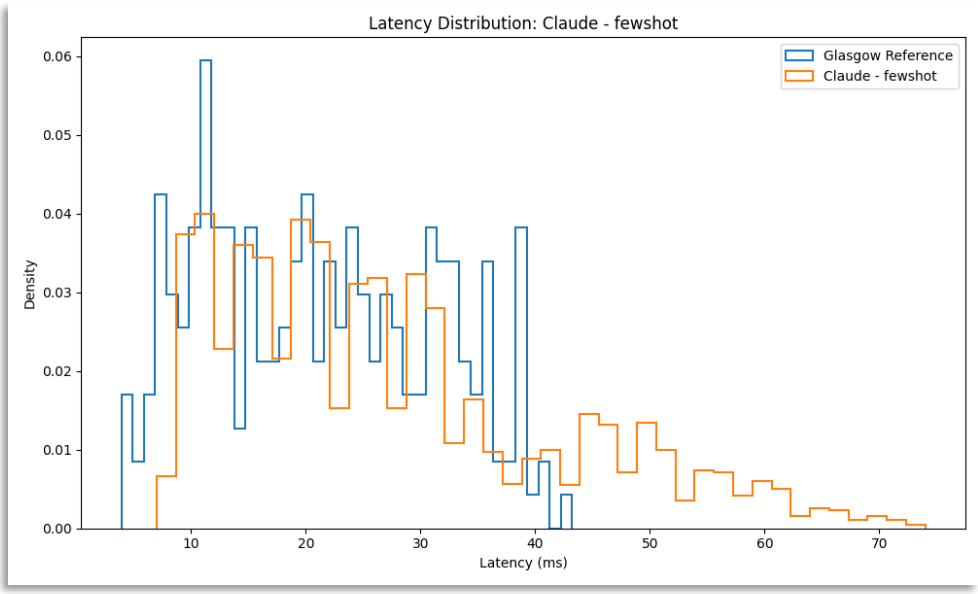

Figure 6: Claude Sonnet 5 with Few-Shot prompting: latency distribution compared with the Glasgow reference dataset

6.5. Quantitative Similarity

When it comes to quantitative similarity and comparing the generated datasets with the Glasgow reference dataset, quantile-based *Mean Absolute Error (MAE)* was used to measure the difference between corresponding quantiles for throughput and latency. Lower MAE values indicate that the generated data is closer to the reference data.

Table 4 presents the quantile-based MAE results for throughput and latency for all prompting strategies and both LLMs. For GPT-4.1-mini, Constraint-Based prompting gave the lowest throughput MAE, with an average value of 166.04 Mbps. Few-Shot prompting followed with 169.65 Mbps, while Chain-of-Thought and Basic prompting had higher values of 502.95 Mbps and 589.49 Mbps, respectively. This shows that Constraint-Based and Few-Shot prompting produced throughput values that were closer to the Glasgow reference dataset across the different quantiles.

For Claude Sonnet 5, Table 4 shows that Few-Shot prompting gave the lowest throughput MAE, with an average value of 138.33 Mbps. Constraint-Based prompting followed with 235.42 Mbps, while Chain-of-Thought and Basic prompting again had higher values of 368.88 Mbps and 419.57 Mbps. This shows a similar pattern to GPT-4.1-mini, where Few-Shot and Constraint-Based prompting performed better than Basic and Chain-of-Thought prompting for throughput.

The latency results were closer between the strategies. According to Table 4, Constraint-Based prompting had the lowest average latency MAE for GPT-4.1-mini, at 5.20 ms, followed by Few-Shot prompting at 5.55 ms. For Claude Sonnet 5, Constraint-Based prompting also had the lowest latency MAE at 5.37 ms, while Few-Shot prompting had a value of 5.61 ms. Chain-of-Thought had the highest value at 8.58 ms.

Overall, Table 4 shows that Constraint-Based and Few-Shot prompting generally produced data that was closer to the Glasgow reference dataset than Basic and Chain-of-Thought prompting. Few-Shot produced the lowest throughput MAE with Claude Sonnet 5, while Constraint-Based prompting produced the lowest throughput MAE with GPT-4.1-mini and for latency with both LLMs.

Table 4. Quantile-based MAE between the generated datasets and the Glasgow reference dataset.

LLM	Strategy	Throughput MAE	Normalized Throughput MAE	Latency MAE	Normalized Latency MAE
GPT	Basic	589.49	0.510	9.65	0.246
GPT	Constraint	**166.04**	**0.144**	***5.20***	***0.132***
GPT	CoT	502.95	0.435	10.88	0.277
GPT	Few-Shot	169.65	0.147	5.55	0.141
Claude	Basic	419.57	0.363	5.49	0.140
Claude	Constraint	235.42	0.204	***5.37***	***0.137***
Claude	CoT	368.88	0.319	8.58	0.218
Claude	Few-Shot	**138.33**	**0.120**	5.61	0.143

6.6. Statistical Comparison

To compare the generated datasets, statistical tests were used to check the differences between the prompting strategies and the two LLMs. The *Kruskal-Wallis* test was used to compare the four prompting strategies within each LLM, while the *Mann-Whitney U* test was used to compare GPT-4.1-mini and Claude Sonnet 5 for the same prompting strategy.

Table 5 presents the statistical comparison results, including the adjusted p-values. The Kruskal-Wallis results show clear differences between the prompting strategies for both throughput and latency for GPT-4.1-mini and Claude Sonnet 5, with p-values below 0.001.

The Mann-Whitney U test also shows differences between GPT-4.1-mini and Claude Sonnet 5 for several prompting strategies. As shown in Table 5, Basic, Chain-of-Thought prompting and Few-Shot prompting showed differences for both throughput and latency. For Constraint-Based prompting, a difference was observed for throughput, while the latency results did not show a statistically significant difference.

Overall, Table 5 shows that the generated datasets differed between prompting strategies and between the two LLMs for several of the comparisons. This indicates that both the prompting strategy and the choice of LLM were associated with differences in the generated data.

Table 5. Statistical comparison between prompting strategies and LLMs.

Comparison	Metric	Test	Adjusted p-value	Result
GPT	Throughput	Kruskal-Wallis	<0.001	Significant
GPT	Latency	Kruskal-Wallis	<0.001	Significant
Claude	Throughput	Kruskal-Wallis	<0.001	Significant
Claude	Latency	Kruskal-Wallis	<0.001	Significant
GPT vs Claude – Basic	Throughput	Mann-Whitney U	0.001461	Significant
GPT vs Claude – Basic	Latency	Mann-Whitney U	0.001979	Significant
GPT vs Claude – Constraint	Throughput	Mann-Whitney U	0.006825	Significant
GPT vs Claude – Constraint	Latency	Mann-Whitney U	0.212294	Not significant
GPT vs Claude – CoT	Throughput	Mann-Whitney U	0.005040	Significant
GPT vs Claude – CoT	Latency	Mann-Whitney U	0.042268	Significant
GPT vs Claude – Few-Shot	Throughput	Mann-Whitney U	0.001461	Significant
GPT vs Claude – Few-Shot	Latency	Mann-Whitney U	0.021854	Significant

6.7. Overall Evalution

To summarize the results from the different evaluation methods, the structural quality and similarity results were compared for all four prompting strategies and both LLMs. The evaluation includes the number of returned rows, distributional similarity using KS and Wasserstein, and quantitative similarity using normalized MAE.

The results presented in Table 6 and 7 show differences between the prompting strategies and the two LLMs. GPT-4.1-mini produced more structurally consistent datasets overall, with Constraint-Based prompting returning exactly 500 rows in all runs and Few-Shot prompting was also close to the reference dataset size, with an average of 497.3 rows per run.

For Claude Sonnet 5, Table 6 shows that Few-Shot prompting had the lowest throughput KS, Wasserstein distance, and normalized MAE values. It also returned the highest average number of rows among the Claude prompting strategies, with 372 rows per run. Constraint-Based prompting also produced relatively close similarity results for throughout and latency.

The latency results in Table 7 show that Constraint-Based prompting had the lowest normalized MAE for both GPT-4.1-mini and Claude Sonnet 5. For GPT-4.1-mini, Constraint-Based prompting also had the lowest latency KS and Wasserstein distance. However, for Claude Sonnet 5, the latency results were more similar between the strategies.

Overall, Table 6 and 7 show that Constraint-Based and Few-Shot prompting generally produced data that was closer to the real-world reference dataset than Basic and Chain-of-Thought prompting. The results also show differences between the two LLMs. GPT-4.1-mini produced more structurally consistent outputs, while Claude Sonnet 5 produced relatively close similarity results for Few-Shot prompting.

Table 6. Overall evaluation of throughput.

LLM	Strategy	Average Rows	Throughput KS	Throughput Wasserstein	Throughput Normalized MAE
GPT	Basic	460.0	0.980	586.99	0.510
GPT	Constraint	**500.0**	**0.327**	**166.40**	**0.144**
GPT	CoT	418.0	0.880	500.39	0.435
GPT	Few-Shot	497.3	0.398	180.01	0.147
Claude	Basic	341.2	0.625	413.78	0.363
Claude	Constraint-Based	339.1	0.360	230.98	0.204
Claude	CoT	236.5	0.579	357.71	0.319
Claude	Few-Shot	**372.0**	**0.248**	**138.90**	**0.120**

Table 7. Overall evaluation of latency.

LLM	Strategy	Average Rows	Latency KS	Latency Wasserstein	Latency Normalized MAE
GPT	Basic	460.0	0.641	9.85	0.246
GPT	Constraint	**500.0**	**0.297**	**5.34**	**0.132**
GPT	CoT	418.0	0.673	10.97	0.277
GPT	Few-Shot	497.3	0.398	5.78	0.141
Claude	Basic	341.2	0.311	5.36	0.140
Claude	Constraint-Based	339.1	0.221	5.82	**0.137**
Claude	CoT	236.5	0.430	8.43	0.218
Claude	Few-Shot	**372.0**	**0.209**	5.93	0.143

7. Discussion

The purpose of this thesis was to investigate whether LLMs can be used to generate structured synthetic cellular network traffic data and how different prompting strategies and LLMs influence the generated datasets. The results show that both GPT-4.1-mini and Claude Sonnet 5 were able to generate structured cellular network traffic data. However, the results also show some differences between the prompting strategies and between the two LLMs.

7.1. RQ1 – Can LLMs generate structured data that reflects important cellular network traffic metrics such as throughput and latency?

The first research question asks if LLMs can generate data that reflects cellular network traffic data. The results show that both GPT-4.1-mini and Claude Sonnet 5 were able to generate datasets containing the requested network metrics and follow the general structure specified in the prompts.

However, the structural quality results also show that being able to generate structured data, does not also mean that the two LLMs followed the requested structure equally well. GPT-4.1-mini generally returned a number of rows that was closer to the requested 500 rows. Constraint-Based prompting returned exactly 500 rows in all runs, while Few-Shot prompting was also close to the requested number. Basic and Chain-of-Thought prompting returned fewer rows.

Claude Sonnet 5 had more difficulty following the requested dataset size. Few-Shot prompting returned the highest average with 372 rows, while Chain-of-Thought returned the lowest average with 236.5 rows. Claude also produced some missing values, invalid network types, and invalid timestamps, while GPT-4.1-mini did not produce these types of structural errors.

Based on these results, GPT-4.1-mini was more consistent than Claude Sonnet 5 when the main requirement was to follow the requested dataset structure and size. This is especially visible with Constraint-Based prompting, which returned exactly the requested 500 rows in all runs. Claude Sonnet 5 was still able to generate structured data, but required more attention to validation and output checking.

The prompting strategy also affected the results. Constraint-Based prompting was particularly effective for GPT-4.1-mini, while Few-Shot prompting produced the highest number of returned rows for Claude Sonnet 5. Chain-of-Thought prompting returned fewer rows for both LLMs. This shows that the ability to generate structured data depends not only on the selected LLM, but also on how the prompt is designed.

7.2. RQ2 – What are the challenges of generating structured data using LLMs in the context of cellular network traffic?

The second question focuses on the challenges of generating structured cellular network traffic data using LLMs. Based on the experiments, one main challenge was getting the LLMs to consistently follow the requested output requirements.

For example, Claude Sonnet 5 had more difficulty following the requested dataset size and also produced some missing values, invalid network types, and invalid timestamps. GPT-4.1-mini did not produce these types of structural errors in the generated datasets. This shows that even when the same dataset structure and requirements are used, different LLMs can handle the generation task differently.

Another challenge was that the prompting strategies did not work in the same way for both LLMs. The results showed differences between Basic, Constraint-Based, Chain-of-Thought, and Few-Shot prompting, and the results also depended on which LLM was used. For example, Constraint-Based prompting generally worked well for GPT-4.1-mini in terms of structural consistency, while Few-Shot prompting produced relatively close results to the reference dataset for Claude Sonnet 5, particularly for throughput. This means that a prompt that works well for one LLM cannot necessarily be expected to work in the same way for another LLM.

The generated data also showed some differences in timestamps and the distribution of the generated values compared to the reference dataset. This makes it difficult to assume that the generated data automatically represents realistic cellular network traffic over longer time periods.

Because of these challenges, the generated datasets need to be checked and evaluated before they can be used. In this study, several evaluation methods were therefore used, including structural quality, descriptive statistics, distributional similarity, quantitative similarity, and statistical comparison. Using several methods made it possible to identify both structural problems and differences between the generated and reference datasets.

Overall, the experiments show that a main challenge of using LLMs for cellular network traffic generation is not only generating structured data, but also generating data that consistently follows the requested requirements and reflects similarity to the reference dataset**.** The results also show that the outcome depends on both the selected LLM and the prompting strategy.

7.3. RQ3 – How similar are the datasets generated by different LLMs and prompting strategies to the real-world reference dataset?

The third research question focuses on how similar the generated datasets are to the Glasgow reference dataset. The results from the different evaluation methods show an overall pattern. Constraint-Based and Few-Shot prompting produced datasets that were closer to the reference dataset in comparison to Basic and Chain-of-Thought prompting.

The results also show differences between the two LLMs. For throughput similarity, Claude Sonnet 5 with Few-Shot prompting produced the closest results to the Glasgow reference dataset. It had the lowest throughput KS statistic, Wasserstein distance, and normalized MAE among all tested combinations. GPT-4.1-mini with Constraint-Based prompting also produced very close throughput results and was the closest GPT-4.1-mini configuration**.**

For structural quality, however, GPT-4.1-mini showed more consistent results than Claude Sonnet 5. GPT-4.1-mini consistently returned a number of rows closer to the requested 500 rows and did not produce the missing values, invalid network types, or invalid timestamps that occurred in the Claude datasets.

The latency results were more mixed. For GPT-4.1-mini, Constraint-Based prompting produced the lowest values for the three latency similarity measures used in the overall comparison, as shown in Table 7. For Claude Sonnet 5, Few-Shot prompting had the lowest KS value, Constraint-Based prompting had the lowest normalized MAE, while Basic prompting had the lowest Wasserstein distance. This shows that no single LLM or prompting strategy produced the lowest values for all latency measures.

The statistical comparison also showed differences between the prompting strategies and between the two LLMs. The Kruskal-Wallis results found differences between the four prompting strategies for both throughput and latency, while the Mann-Whitney U results found differences between GPT-4.1-mini and Claude Sonnet 5 for several of the prompting strategies. These results further indicate that both the choice of LLM and the prompting strategy can affect the generated data.

Based on these results, GPT-4.1-mini showed more consistent structural results when following the requested dataset format was important. Claude Sonnet 5, particularly with Few-Shot prompting, produced closer throughput distributions to the real-world reference dataset. Therefore, the choice between the two LLMs can depend on the purpose and requirements of the generated dataset rather than one LLM being better in all cases.

Previous research discussed earlier in the related work section has shown that LLMs can be used to generate synthetic data in different domains. The results of this thesis show that LLMs can also be used for synthetic cellular network traffic generation. However, the results should not be generalized to all LLMs or cellular network traffic, since only two LLMs and one real-world reference dataset were used.

Overall, the results show that LLMs can be useful for generating synthetic cellular network traffic data, but the generated data still needs to be evaluated against real-world data. GPT-4.1-mini provided more consistent structural outputs, while Claude Sonnet 5 with Few-Shot prompting produced particularly close throughput distributions. The results therefore suggest that the LLM and prompting strategy should be selected based on the requirements of the dataset being generated.

7.4. Practical Use of Methodology

Based on the results of this study, a user who wants to generate a synthetic cellular network traffic dataset can follow the same general process used in this thesis. As shown in Figure 2, the process starts with selecting a prompting strategy and an LLM, followed by data generation and evaluation. The first step is to define the required dataset structure, including the columns, expected value ranges, network types, relationships between the network features, and number of rows. A detailed prompt can then be created using constraints or example rows to guide the LLM. Example of the different prompt types I used in this study are shown in Figure 1.

For the LLMs tested in this study, GPT-4.1-mini and Claude Sonnet 5 can be used with the prompting strategies described in Section 5.3. The results suggest that GPT-4.1-mini is a suitable choice when structural consistency and following the requested dataset size are important. Constraint-Based prompting was particularly effective for this purpose, as it returned exactly 500 rows in all runs. Claude Sonnet 5 with Few-Shot prompting can be considered when similarity in throughput is an important requirement, as this combination produced the closest throughput results to the Glasgow reference dataset.

However, these results do not mean that one LLM should always be selected over the other. The results differed depending on the metric and prompting strategy. Therefore, a user should not assume that a prompt that works well with one LLM will work in the same way with another LLM. Instead, several generation runs should be performed for the selected LLM and prompting strategy, as was done in this study.

After generating the dataset, the output should be checked for structural errors such as missing values or invalid values. The generated data can then be evaluated using the same methods used in this thesis, including structural quality, descriptive statistics, distributional similarity, quantitative similarity, and statistical comparison or other methods. Additionally, the generated data can be compared with a real-world reference dataset using the measures presented in the Results section. This makes it possible to check how close the generated data is to the reference data before using it for further analysis or testing.

If the generated data does not meet the required structure or does not show sufficient similarity to the reference data, the prompt or the selected LLM can be adjusted and the generation process repeated. In this way, the methodology can be used as an iterative process where the dataset requirements are defined, a prompt and LLM are selected, data is generated, and the results are evaluated before the dataset is used.

7.5. Thesis Limitations

A limitation of the dataset generation is that the physical environment was not explicitly considered when generating the synthetic data. The Glasgow reference datasets I used for comparison, consists of real-world public 5G measurements collected outdoors across different urban and suburban areas in Glasgow, and the measurements were collected under varying conditions such as different times of day,

weather, and user loads [15]. The generated datasets in this study, however, were not created for a specific environment such as indoor, outdoor, urban, or rural settings.

Because of this, the generated data does not capture how the physical environment can affect cellular network performance. Factors such as buildings, obstacles, location, and other environmental conditions can influence metrics such as throughput and latency. This means that the generated datasets should not be assumed to represent all types of cellular network environments. A more environment-specific approach could be investigated in future work by including environmental conditions in the dataset structure and prompt design.

8. Conclusion and Future Work

The purpose of this study was to investigate whether LLMs can be used to generate synthetic cellular network traffic datasets and how different prompt engineering strategies and LLMs influence the generated datasets.

Based on the evaluation presented in this thesis, all three research questions could be answered. Both GPT-4.1-mini and Claude Sonnet 5 were able to generate structured cellular network traffic datasets containing important metrics such as throughput and latency. However, the structural quality differed between the LLMs and prompting strategies, showing that generating structured data does not always mean that the requested output is followed equally well.

When comparing the generated datasets with the Glasgow reference dataset, there were differences between the prompting strategies. Constraint-Based and Few-Shot prompting generally generated data that was closer to the reference dataset than Basic and Chain-of-Thought prompting. The results also showed that the strategy producing the closest results could differ between the two LLMs. Overall, this shows that both the choice of LLM and prompting strategy can affect the quality of the generated data and its similarity to the reference data.

Regarding the strengths and limitations of using LLMs for this task, the results showed that large amounts of structured data could be generated without access to real operator datasets or model training. However, the generated data cannot be assumed to fully represent real cellular network traffic. The evaluation was mainly based on throughput and latency and a single real-world reference dataset, meaning that other network characteristics and environments could produce different results. Therefore, the generated data cannot be considered representative of all aspects of real cellular network traffic.

Future work could investigate more LLMs, additional network metrics, and multiple real-world cellular traffic datasets. It could also be useful to test the generated data in network simulations or testing environments and investigate whether combining different prompting strategies or iterative validation can improve the generated datasets.

References

[1] M. Nadas, L. Diosan, and A. Tomescu, "Synthetic Data Generation Using Large Language Models: Advances in Text and Code," *IEEE Access*, vol. 13, pp. 134615–134633, Jul. 2025.

[2] H. Zhou, C. Hu, D. Yuan, Y. Yuan, D. Wu, X. Chen, H. Tabassum, and X. Liu, "Large Language Models for Wireless Networks: An Overview from the Prompt Engineering Perspective," *IEEE Wireless Communications*, Aug. 2025.

[3] Y. Wang and K. Li, "Large Language Models in Operations Research: Methods, Applications, and Challenges," *arXiv preprint arXiv:2509.18180*, Oct. 2025.

[4] S. Duan, F. Lyu, J. Cen, J. Ren, P. Yang, and Y. Zhang, "Flexible and Effective Cellular Traffic Data Synthesis with Large Language Model," in *Proc. IEEE Global Communications Conference (GLOBECOM) Workshops, Selected Areas in Communications: Machine Learning for Communications*, 2024.

[5] J. Wei, M. Bosma, V. Y. Zhao, K. Guu, A. W. Yu, B. Lester, N. Du, A. M. Dai, and Q. V. Le, "Finetuned language models are zero-shot learners," in *Proc. Int. Conf. Learn. Representations (ICLR)*, 2022.

[6] H. K. Chaubey, G. Tripathi, R. Ranjan, and S. K. Gopalaiyengar, "Comparative Analysis of RAG, Fine-Tuning, and Prompt Engineering in Chatbot Development," in *Proc. 2024 Int. Conf. Future Technologies for Smart Society (ICFTSS)*, Aug. 2024.

[7] H. Zhou, C. Hu, D. Yuan, Y. Yuan, D. Wu, X. Liu, and J. Zhang, "Large Language Model (LLM)-enabled In-context Learning for Wireless Network Optimization: A Case Study of Power Control," in *Proc. ICML 2025 Workshop on Machine Learning for Wireless Communications (ML4Wireless)*, 2025.

[8] J. Wei, X. Wang, D. Schuurmans, et al., "Chain-of-thought prompting elicits reasoning in large language models, " in *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

[9] Z. Li, H. Zhu, Z. Lu, and M. Yin, "Synthetic Data Generation with Large Language Models for Text Classification: Potential and Limitations," *arXiv preprint* arXiv:2310.07849, Oct. 2023.

[10] J. Lee, H. Lee, J. Kim, K. Lee, and E. Kim, "Self-Corrective Task Planning by Inverse Prompting with Large Language Models," *arXiv preprint* arXiv:2503.07317, Mar. 2025.

[11] S. Shen, L. Logeswaran, M. Lee, H. Lee, S. Poria, and R. Mihalcea, "Understanding the Capabilities and Limitations of Large Language Models for Cultural Commonsense," *arXiv preprint* arXiv:2405.04655, May 2024.

[12] K. Wang, J. Zhu, M. Ren, Z. Liu, S. Li, Z. Zhang, C. Zhang, X. Wu, Q. Zhan, Q. Liu, and Y. Wang, "A Survey on Data Synthesis and Augmentation for Large Language Models," arXiv preprint arXiv:2410.12896, Oct. 2024.

[13] L. Long, R. Wang, R. Xiao, J. Zhao, X. Ding, G. Chen, and H. Wang, "On LLMs-Driven Synthetic Data Generation, Curation, and Evaluation: A Survey," arXiv preprint arXiv:2406.15126, Jun. 2024.

[14] T. G. Omomule, D. O. Olarinde, C. C. Ugwu, and T. A. Falana, "QoS Performance Metrics for Analyzing Wireless Network Usability," *IRE Journals*, vol. 3, no. 6, Dec. 2019.

[15] A. Hart, H. Sturley, P. Mclean, P. Salva Garcia, and M. Z. Shakir, "Glasgow Public and Private 5G Performance Dataset (2025): Acquisition and Comparative Analysis," *Research Square preprint*, Jan. 2026, doi: 10.21203/rs.3.rs-8148455/v1.

[16] M. I. Rochman, V. Sathya, D. Fernandez, N. Nunez, A. S. Ibrahim, W. Payne, and M. Ghosh, "A comprehensive analysis of the coverage and performance of 4G and 5G deployments," *Computer Networks*, vol. 237, Art. no. 110060, 2023.

[17] N. Panwar, S. Sharma, and A. K. Singh, "A survey on 5G: The next generation of mobile communication," *arXiv preprint arXiv:1511.01643*, 2015.

[18] J. Shin, C. Tang, T. Mohati, M. Nayebi, S. Wang, and H. Hemmati, "Prompt Engineering or Fine Tuning: An Empirical Assessment of LLMs for Code," *arXiv preprint* arXiv:2310.10508, Feb. 2025.

[19] A. Matarazzo and R. Torlone, "A Survey on Large Language Models with Some Insights on Their Capabilities and Limitations," *arXiv preprint* arXiv:2501.04040, Feb. 2025.

[20] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, and T. Liu, "A Survey on Hallucination in Large Language Models: Principles, Taxonomy, Challenges, and Open Questions," *arXiv preprint* arXiv:2311.05232, Nov. 2024.

[21] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, A. Askell, R. Child, P. Dhariwal, S. Agarwal, A. Ramesh, *et al*., "Language Models are Few-Shot Learners," *arXiv preprint* arXiv:2005.14165, Jul. 2020.

[22] N. A. Heckert, J. J. Filliben, C. M. Croarkin, B. Hembree, W. F. Guthrie, P. Tobias, and P. Prinz, *NIST/SEMATECH e-Handbook of Statistical Methods*, NIST Interagency/Internal Report 151, National Institute of Standards and Technology, Gaithersburg, MD, 2002, doi: 10.18434/M32189.

[23] J. Snoke, G. M. Raab, B. Nowok, C. Dibben, and A. Slavkovic, "General and specific utility measures for synthetic data," *Journal of the Royal Statistical Society: Series A (Statistics in Society)*, vol. 181, no. 3, pp. 663–688, 2018, doi: 10.1111/rssa.12358.

[24] National Institute of Standards and Technology, "Kolmogorov-Smirnov Two Sample," *NIST/SEMATECH e-Handbook of Statistical Methods*, National Institute of Standards and Technology. [Online]. Available: https://www.itl.nist.gov/div898/software/dataplot/refman1/auxillar/ks2samp.htm

[25] V. M. Panaretos and Y. Zemel, "Statistical aspects of Wasserstein distances," *Annual Review of Statistics and Its Application*, vol. 6, pp. 405–431, 2019, doi: 10.1146/annurev-statistics-030718 104938

[26] National Institute of Standards and Technology, "Quantile," *NIST/SEMATECH e-Handbook of Statistical Methods*, National Institute of Standards and Technology. [Online]. Available: https://itl.nist.gov/div898/software/dataplot/refman2/auxillar/quantile.htm

[27] National Institute of Standards and Technology, "Mean Absolute Error," *NIST Internal Report 8525*, National Institute of Standards and Technology, 2026, doi: 10.6028/NIST.IR.8525.

[28] National Institute of Standards and Technology, "Kruskal Wallis," *NIST/SEMATECH e-Handbook of Statistical Methods*, National Institute of Standards and Technology. [Online]. Available: https://itl.nist.gov/div898/software/dataplot/refman1/auxillar/kruskwal.htm

[29] National Institute of Standards and Technology, "Mann-Whitney U Test," *NIST/SEMATECH e Handbook of Statistical Methods*, National Institute of Standards and Technology. [Online]. Available: https://www.itl.nist.gov/div898/handbook/prc/section3/prc35.htm

[30] OpenAI, "GPT-4.1 Mini," *OpenAI API Documentation*. [Online]. Available: https://developers.openai.com/api/docs/models/gpt-4.1-mini

[31] Anthropic, “Introducing Claude Sonnet 5,” Anthropic, Jun. 2026. [Online]. Available: https://www.anthropic.com/news/claude-sonnet-5

[32] S. Holm, “A Simple Sequentially Rejective Multiple Test Procedure,” *Scandinavian Journal of Statistics*, vol. 6, no. 2, pp. 65–70, 1979.

Appendix

Here are the remaining plots for the distribution comparisons between the generated datasets and the Glasgow reference dataset.

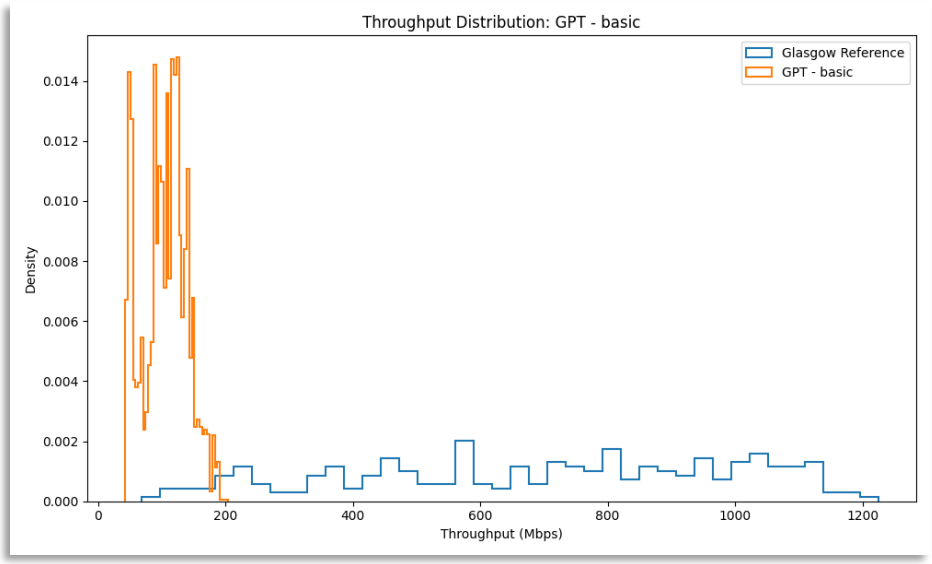

Figure A.1: GPT-4.1-mini with Basic prompting: throughput distribution compared with the Glasgow reference dataset.

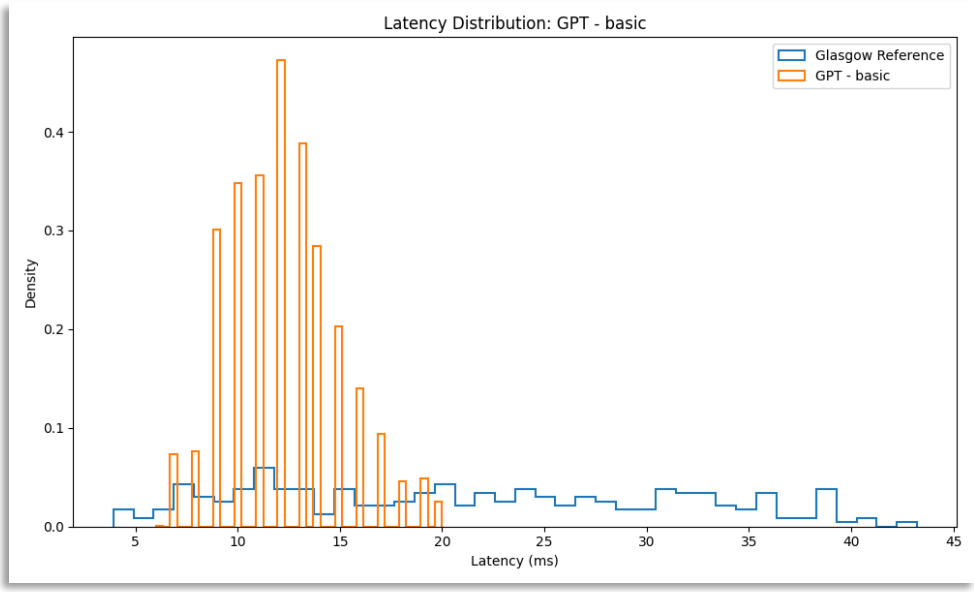

Figure A.2: GPT-4.1-mini with Basic prompting: latency distribution compared with the Glasgow reference dataset.

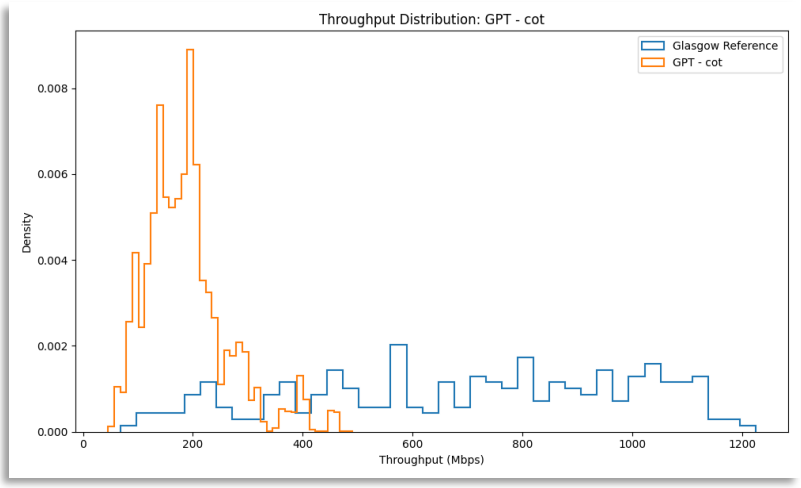

Figure A.3: GPT-4.1-mini with Chain-of-Thought prompting: throughput distribution compared with the Glasgow reference dataset.

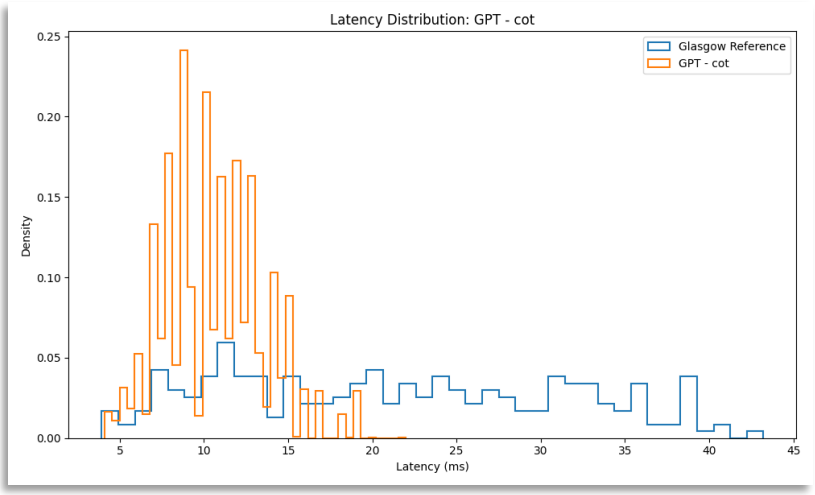

Figure A.4: GPT-4.1-mini with Chain-of-Thought prompting: latency distribution compared with the Glasgow reference dataset.

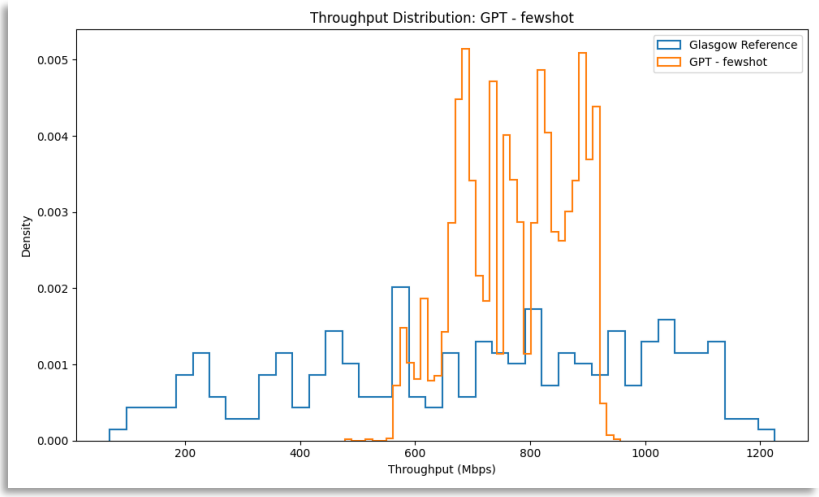

Figure A.5: GPT-4.1-mini with Few-Shot prompting: throughput distribution compared with the Glasgow reference dataset.

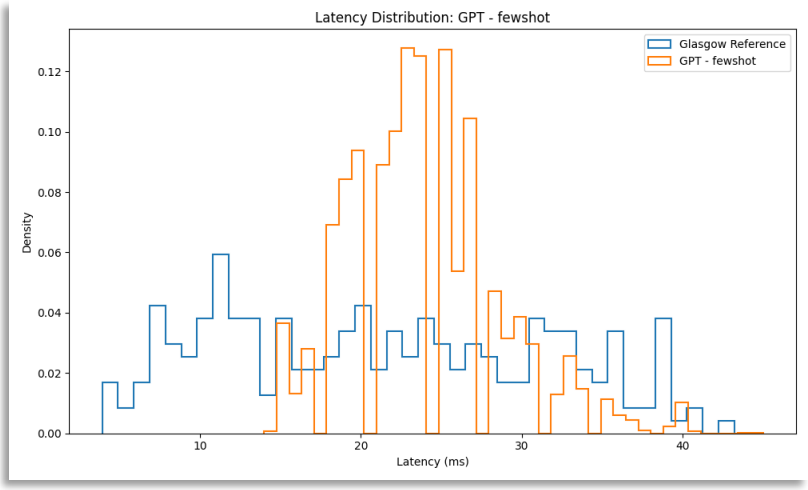

Figure A.6: GPT-4.1-mini with Few-Shot prompting: latency distribution compared with the Glasgow reference dataset.

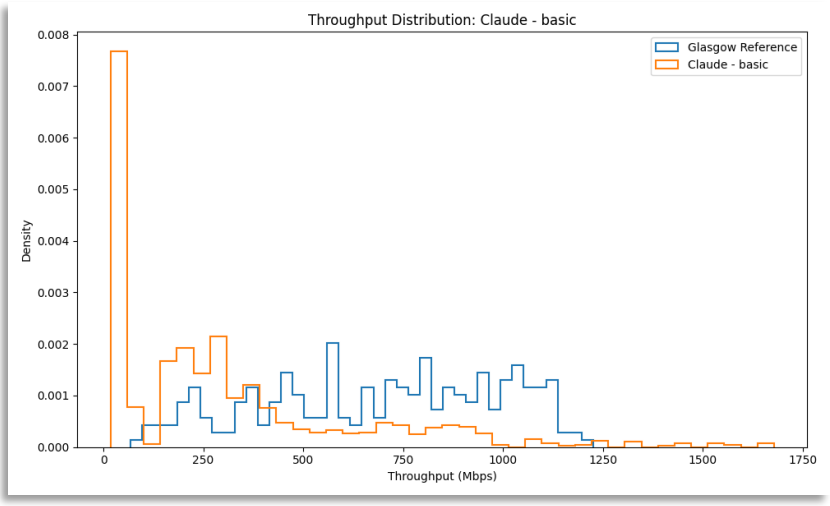

Figure A.7: Claude Sonnet 5 with Basic prompting: throughput distribution compared with the Glasgow reference dataset.

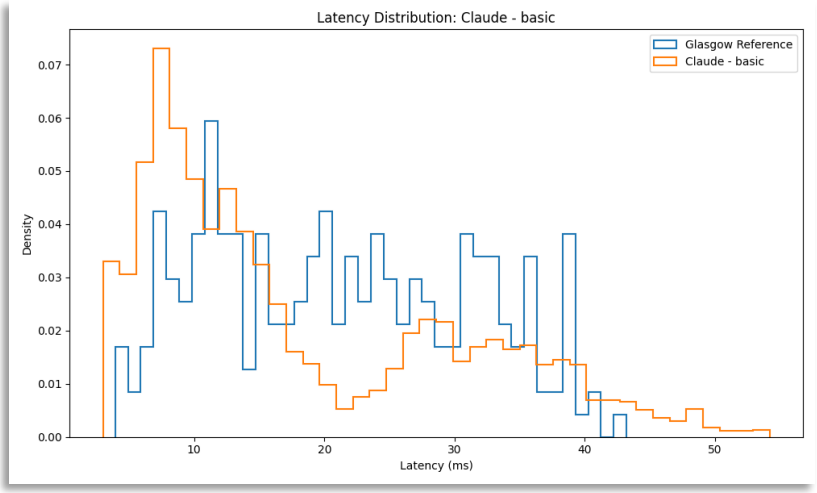

Figure A.8: Claude Sonnet 5 with Basic prompting: latency distribution compared with the Glasgow reference dataset.

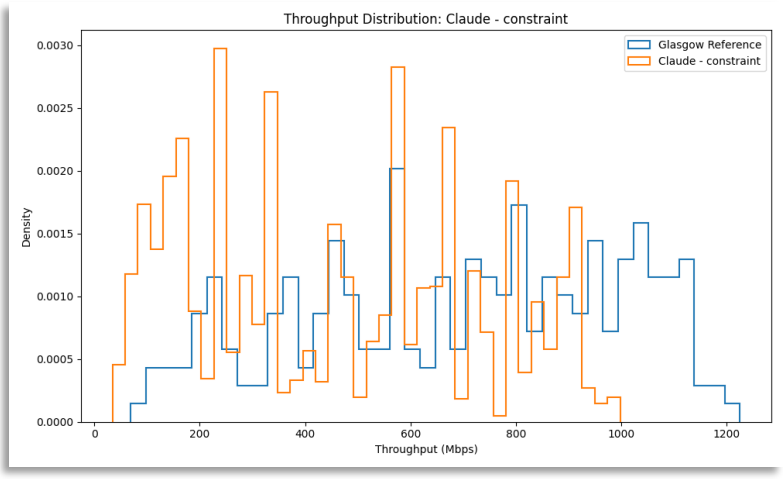

Figure A.9: Claude Sonnet 5 with Constraint-Based prompting: throughput distribution compared with the Glasgow reference dataset.

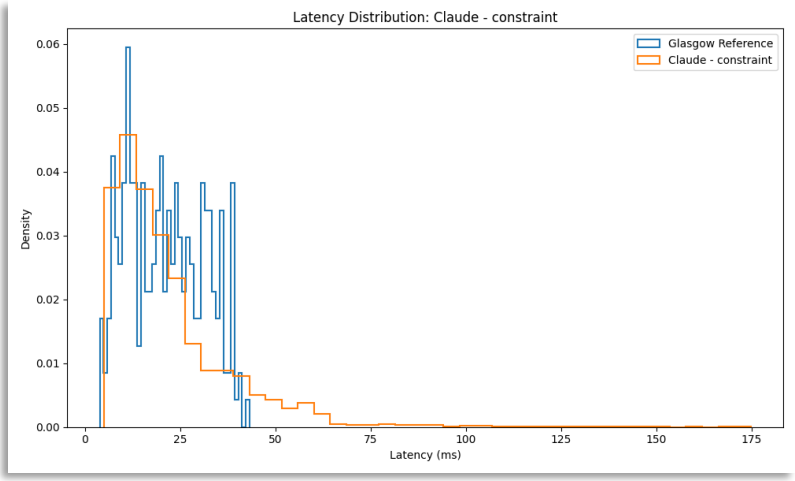

Figure A.10: Claude Sonnet 5 with Constraint-Based prompting: latency distribution compared with the Glasgow reference dataset.

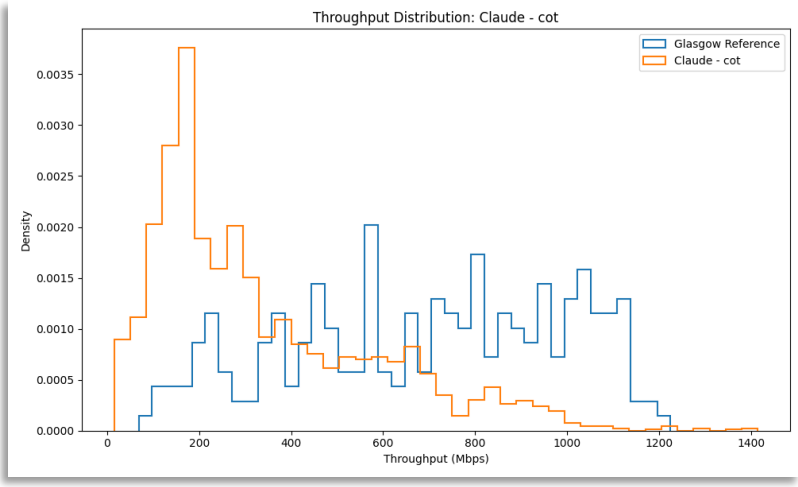

Figure A.11: Claude Sonnet 5 with Chain-of-Thought prompting: throughput distribution compared with the Glasgow reference dataset.

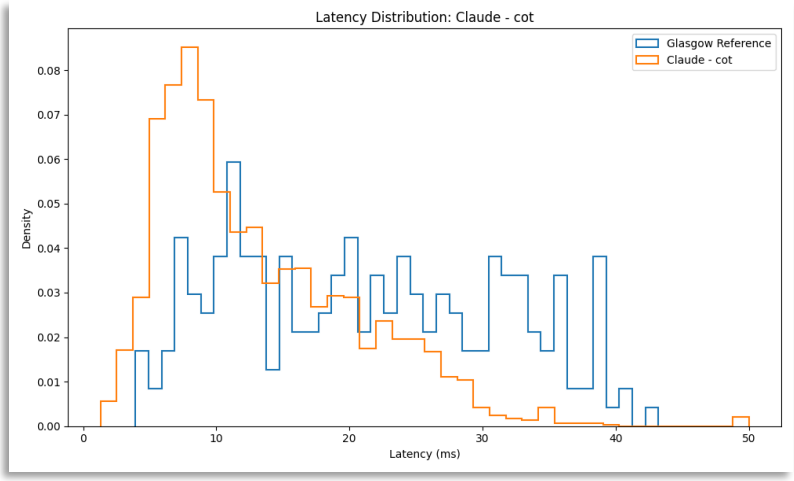

Figure A.12: Claude Sonnet 5 with Chain-of-Thought prompting: latency distribution compared with the Glasgow reference dataset.