Paper No: 25PESGM0612

Large Language Model for Power System Scenario Generation Considering Weather Condition

Xiyuan Zhou[a], Yuheng Cheng[b,*], Yuan Zhong[c], Huan Zhao[d,*], Yifan Zhu[b], Xinlei Cai[e], Yan Xu[a,*], Junhua Zhao[b,*]

[a] School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798, Singapore.
[b] School of Science and Engineering, The Chinese University of Hong Kong (Shenzhen), Shenzhen, 518100, China.
[c] College of Arts & Science, New York University, New York, NY 10003, United States.
[d] Department of Building Environment and Energy Engineering, The Hong Kong Polytechnic University, Hong Kong, China.
[e] Guangdong Power Grid Corporation, Guangzhou, 510600, China.
xiyuan002@e.ntu.edu.sg, yuhengcheng@link.cuhk.edu.cn, yz7628@nyu.edu, huan-paul.zhao@polyu.edu.hk, yifanzhu2@link.cuhk.edu.cn, 517665114@qq.com, xuyan@ntu.edu.sg, zhaojunhua@cuhk.edu.cn

Introduction

Background

- High **renewable penetration** increases uncertainty in power systems.
- Uncertainty is strongly influenced by complex weather dynamics.
- Existing methods **lack flexibility** to model both stochasticity and weather impacts.
- **LLMs** offer reasoning and context awareness for **scenario generation.**

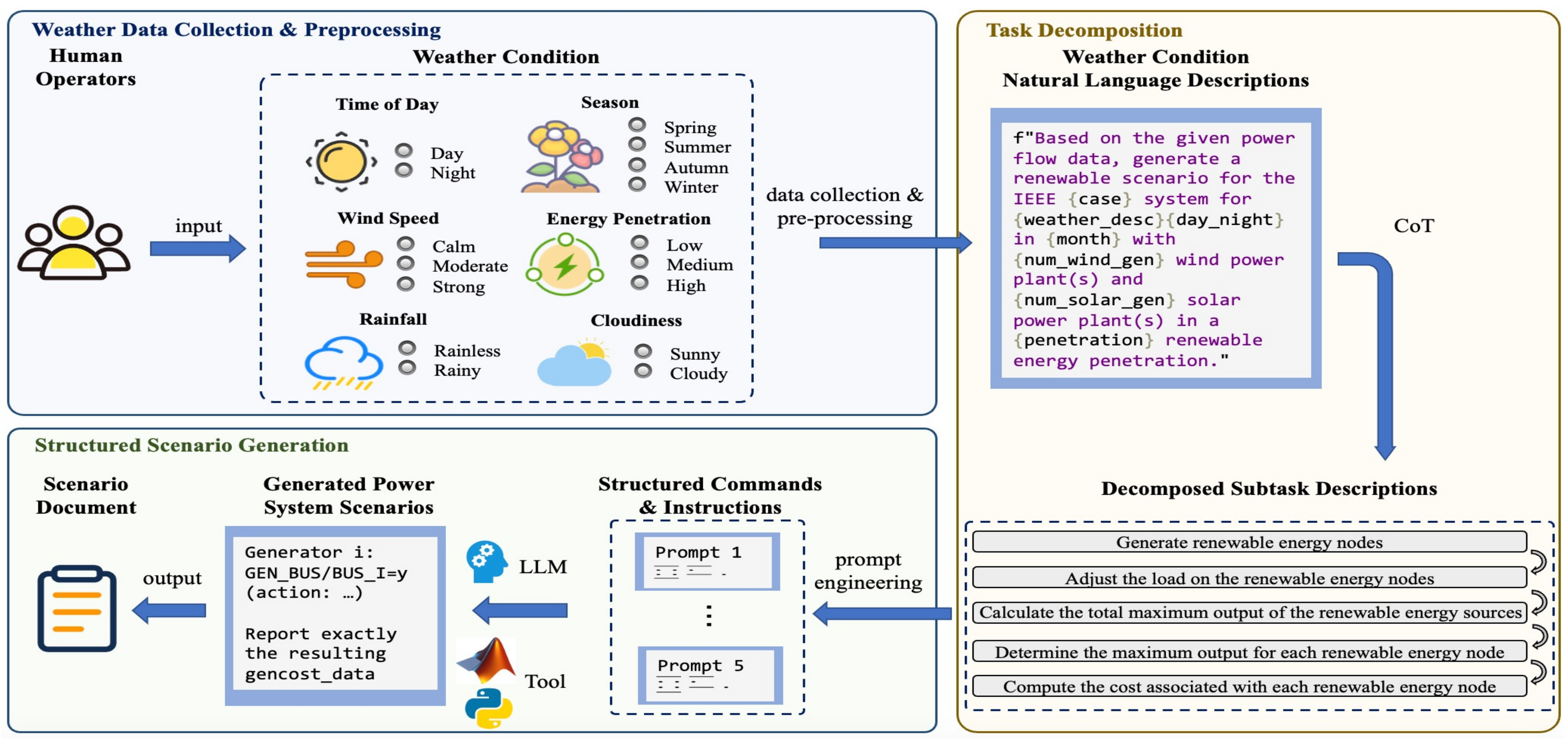
Contributions

This is the first application of LLMs to weather-aware, stochastic power scenario generation.

- Propose an **LLM-based** framework for weather-aware power system **scenario generation** with natural language interaction.
- Integrate **CoT** reasoning and **prompt engineering** to enhance scenario quality.
- Evaluate and compare multiple LLMs, demonstrating their potential for scenario generation.

Methodology

Overall Framework



A framework integrating an **LLM, CoT reasoning, and prompt engineering** is proposed to generate power system scenarios under weather conditions.

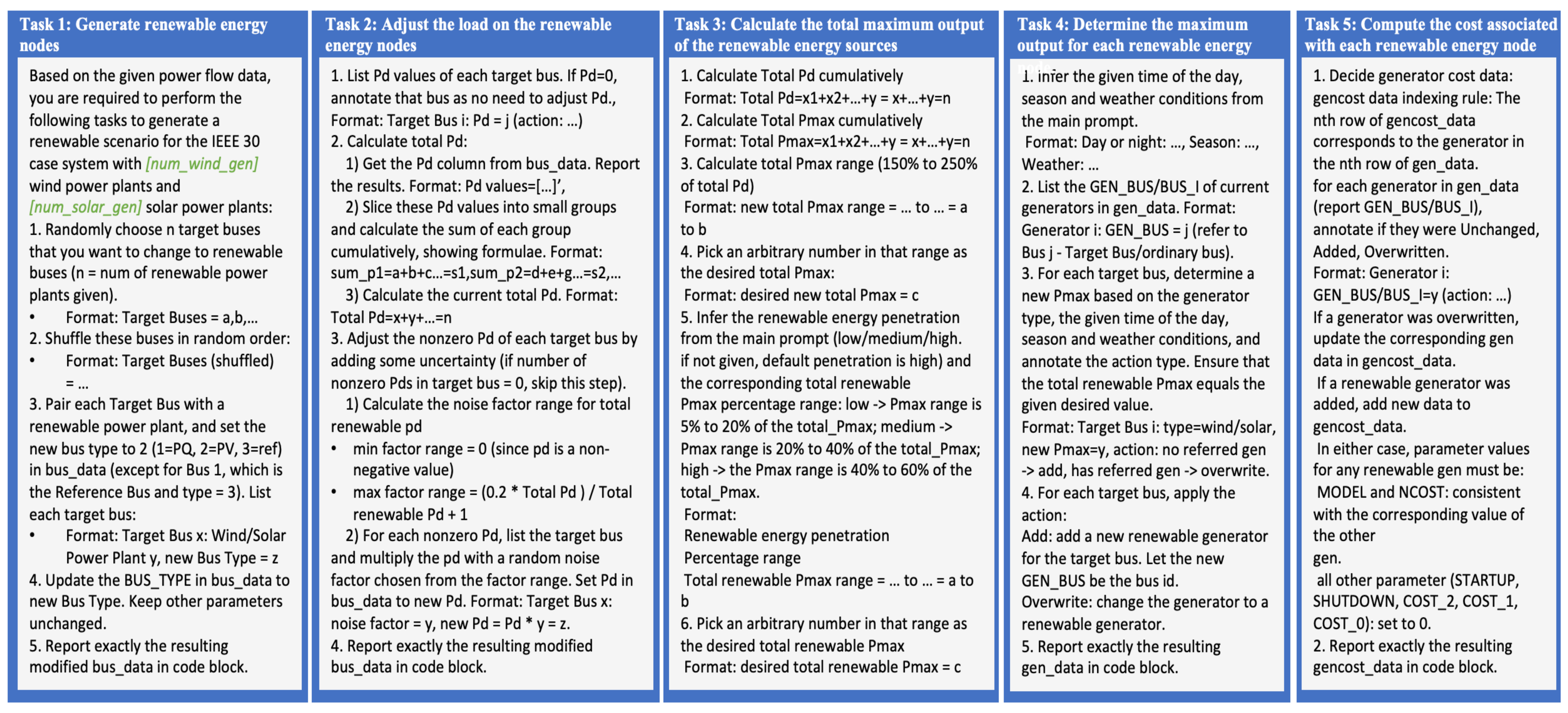
- **Weather Input:** Weather conditions are standardized and described in natural language.
- **Task Decomposition:** Scenario generation is broken into logical subtasks for step-by-step reasoning.
- **Scenario Generation:** Structured prompts guide the LLM to produce feasible and cost-effective scenarios.

Chain-of-Thought Reasoning

- Decomposes complex scenario generation into logical subtasks.
- Enhances feasibility, diversity, and interpretability of outputs.
- Enables step-by-step reasoning aligned with domain constraints.

Prompt Engineering

- Translates weather and system parameters into structured model instructions.
- Ensures outputs are technically feasible and economically reasonable.
- Guides the LLM to generate controlled, high-quality scenarios.



Test Scenario Construction

The IEEE **30-bus** system is used to construct scenarios under **228 weather combinations**. Six representative weather conditions are selected, and ten scenarios per condition are generated, considering **season, time of day, renewable generator type and quantity, and penetration level.**

Test Criteria

- **Success Rate:** Measures the proportion of scenarios that are feasible under OPF constraints.
- **Typicality:** Evaluates how representative and diverse a scenario is within and across clusters.
- **Distribution Accuracy:** Assesses how closely the generated distributions match the actual solar and wind power distributions.

Testing Results

Comparison of Different Methods

TABLE I. COMPARISON OF DIFFERENT METHODS.

	GPT-3.5+CoT	LLaMA2 13b+CoT	LLaMA2 70b+CoT
Success Rate	**65.75%**	10.25%	38.33%
Typicality	**0.5423**	0.4570	0.4816
Distribution Accuracy	**0.001223**	0.001220	0.001221

- **GPT-3.5+CoT** generates more **feasible and representative** scenarios than LLaMA2-based models, showing stronger reasoning and generalization in complex tasks.
- Model **architecture** and **pretraining quality** significantly impact scenario generation performance.

Comparison of Chain of Thought Method

TABLE II. PERFORMANCE COMPARISON USING COT.

	GPT-3.5	GPT-3.5+CoT	LLaA2 13b	LLaMA2 13b+CoT
Success Rate	0	**65.75%**	0	**10.25%**
Typicality	0	**0.5423**	0	**0.4570**
Distribution Accuracy	0	**0.001223**	0	**0.001220**

- CoT is **essential** for task success: without CoT, both models fail completely.
- CoT **improves** feasibility, diversity, and interpretability by guiding LLMs through logical subtasks.
- CoT **enhances robustness** of LLMs under domain constraints.

Case Study Example

Response to "Compute the cost associated with each renewable energy node."

```
# Decide generator cost data
Generator 1: GEN_BUS = 1 (action: Unchanged)
Generator 2: GEN_BUS = 2 (action: Unchanged)
Generator 22: GEN_BUS = 22 (action: Unchanged)
Generator 9: GEN_BUS = 9 (action: Added)
Generator 13: GEN_BUS = 13 (action: Added)
Since generators 9 and 13 were added as renewable generators, we need to update gencost_data for these generators while keeping MODEL and NCOST consistent with the corresponding values of the other generators. All other parameters (STARTUP, SHUTDOWN, COST_2, COST_1, COST_0) should be set to 0.
# Report exactly the resulting gencost_data
gencost_data = [[2, 0, 0, 3, 0.02, 2, 0],       // Unchanged
            [2, 0, 0, 3, 0.0175, 1.75, 0],      // Unchanged
            [2, 0, 0, 3, 0.0625, 1, 0],         // Unchanged
            [2, 0, 0, 3, 0.00834, 3.25, 0],     // Unchanged
            [2, 0, 0, 3, 0.025, 3, 0],          // Unchanged
            [2, 0, 0, 3, 0.025, 3, 0],          // Unchanged
            [2, 0, 0, 3, 0, 0, 0],              // Overwritten
            [2, 0, 0, 3, 0, 0, 0]]              // Overwritten
The gencost_data has been updated to reflect the changes for the added renewable generators (Generator 9 and Generator 13).
```

- The framework generates realistic, cost-effective scenarios that support weather-aware power system planning.