

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**.

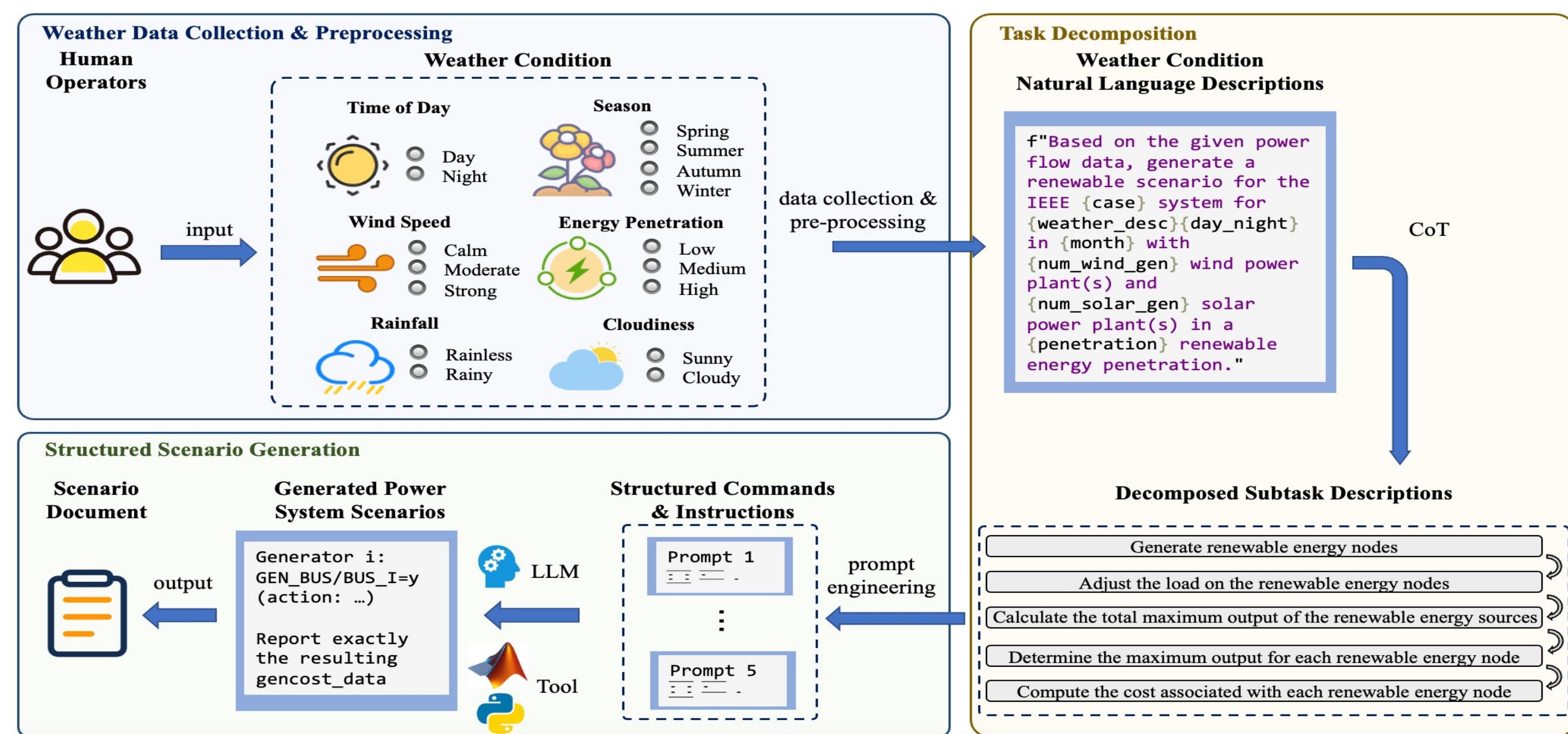
Contributions

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.

Task 1: Generate renewable energy nodes	Task 2: Adjust the load on the renewable energy nodes	Task 3: Calculate the total maximum output of the renewable energy sources	Task 4: Determine the maximum output for each renewable energy source	Task 5: Compute the cost associated with each renewable energy node
<p>Based on the given power flow data, you are required to generate a renewable scenario for the IEEE 30 case system with [num_wind_gen] wind power plants and [num_solar_gen] solar power plants: 1. Randomly choose a target bus that you want to change to renewable bus (n = num of renewable power plants given). • Format: Target Bus = a... 2. Shuffle these buses in random order: • Format: Target Buses (shuffled) = ... 3. Pair each Target Bus with a renewable power plant, and set the new bus type to 2 (1=PV, 2=EV) in bus_data (except for Bus 1, which is the Reference Bus and type = 3). List each target bus: • Format: Target Bus x: Wind/Solar Power Plant y, new Bus Type = z new Bus Type. Keep other parameters unchanged. 5. Report exactly the resulting modified bus_data in code block.</p>	<p>1. Set Pd values of each target bus. If Pd=0, annotate that bus as no need to adjust Pd. Format: Target Bus = Pd + action: ... 2. Calculate total Pd cumulatively. Format: Total Pmax*x1*x2...*y1...*y2... 3. Calculate total Pmax range (150% to 250% of total Pd) Format: new total Pmax range = ... to ... + a to b 4. Pick an arbitrary number in that range as the new Pd values. Format: Target Pd = ... 5. Infer the renewable energy penetration from the main prompt (low/medium/high, if not given, default penetration is high) and the corresponding total renewable Pmax percentage range: low > Pmax range is 5% to 20% of the total_Pmax; medium > Pmax range is 20% to 40% of the total_Pmax; high > the Pmax range is 40% to 60% of the total_Pmax. Format: Renewable energy penetration Percentage range 6. Pick an arbitrary number in that range as the desired total renewable Pmax. Format: desired total renewable Pmax = c 7. For each target bus, apply the action: Add: add a new renewable generator for the target bus. Let the new Bus_ID be the bus id. Overwrite: change the generator to a renewable generator. 5. Report exactly the resulting genconst_data in code block.</p>	<p>1. Calculate Total Pd cumulatively. Format: Total Pd=x1*x2...*y1...*y2... 2. Calculate Total Pmax cumulatively. Format: Total Pmax*x1*x2...*y1...*y2... 3. Calculate total Pmax range (150% to 250% of total Pd) Format: new total Pmax range = ... to ... + a to b 4. Pick an arbitrary number in that range as the new Pd values into small groups and calculate the sum of each group. Format: Pmax = x1*x2...*y1...*y2... 5. Infer the renewable energy penetration from the main prompt (low/medium/high, if not given, default penetration is high) and the corresponding total renewable Pmax percentage range: low > Pmax range is 5% to 20% of the total_Pmax; medium > Pmax range is 20% to 40% of the total_Pmax; high > the Pmax range is 40% to 60% of the total_Pmax. Format: Renewable energy penetration Percentage range 6. Pick an arbitrary number in that range as the desired total renewable Pmax. Format: desired total renewable Pmax = c 7. For each target bus, apply the action: Add: add a new renewable generator for the target bus. Let the new Bus_ID be the bus id. Overwrite: change the generator to a renewable generator. 5. Report exactly the resulting genconst_data in code block.</p>	<p>1. Calculate Total Pd cumulatively. Format: Total Pd=x1*x2...*y1...*y2... 2. Calculate Total Pmax cumulatively. Format: Total Pmax*x1*x2...*y1...*y2... 3. Calculate total Pmax range (150% to 250% of total Pd) Format: new total Pmax range = ... to ... + a to b 4. Pick an arbitrary number in that range as the new Pd values. Format: Target Pd = ... 5. Infer the renewable energy penetration from the main prompt (low/medium/high, if not given, default penetration is high) and the corresponding total renewable Pmax percentage range: low > Pmax range is 5% to 20% of the total_Pmax; medium > Pmax range is 20% to 40% of the total_Pmax; high > the Pmax range is 40% to 60% of the total_Pmax. Format: Renewable energy penetration Percentage range 6. Pick an arbitrary number in that range as the desired total renewable Pmax. Format: desired total renewable Pmax = c 7. For each target bus, apply the action: Add: add a new renewable generator for the target bus. Let the new Bus_ID be the bus id. Overwrite: change the generator to a renewable generator. 5. Report exactly the resulting genconst_data in code block.</p>	<p>1. Decide generator cost data: genconst_data indicating rule: The nth row of genconst_data corresponds to the generator in the nth row of gen_data. for each generator in gen_data (report GEN_BUS/BUS), annotate if they were Unchanged, Added, or Deleted. Format: Generator i: GEN_BUS/BUS (action: ...) If a generator was overwritten, update the corresponding gen data in genconst_data. If a renewable generator was added, add new data to genconst_data. In either case, parameter values for any renewable gen must be: MODEL and NCOST consistent with the corresponding value of the other gen. All other parameter (STARTUP, SHUTDOWN, COST_2, COST_1, COST_0) set to 0. 2. Report exactly the resulting genconst_data in code block.</p>

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	LLaMA2 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

```
# 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 genconst_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 genconst_data
genconst_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, 0.0225, 3, 0], // Unchanged
[2, 0, 0, 3, 0.0225, 3, 0], // Unchanged
[2, 0, 0, 3, 0, 0, 0], // Overwritten
[2, 0, 0, 3, 0, 0, 0]] // Overwritten
The genconst_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.