

Large Language Models in Power Systems: Enhancing Control and Decision-Making

Alen Bernadić¹, Goran Kujundžić¹, Ivana Primorac²

¹Faculty of Mechanical Engineering, Computing and Electrical Engineering University of Mostar, Bosnia and Herzegovina

²University of Rijeka, Faculty of Economics and Business, Croatia

Abstract

This paper explores the application of Large Language Models (LLMs) in power systems, focusing on their potential to revolutionize control, optimization, and decision-making processes. We present a comprehensive review of current research and applications, highlighting the challenges and opportunities in this emerging field. A practical example is provided, demonstrating the implementation of an LLM agent for power system control using Python. The power system is modeled using Pandapower, while the LLM agent is based on Llama 3, executed through Ollama. Our findings suggest that LLMs can significantly enhance the efficiency and reliability of power system operations, paving the way for more intelligent and adaptive energy management systems.

Article history

Received: 14. 11. 2024.

Revised: 10. 01. 2025.

Accepted: 12. 01. 2025.

Keywords

Power Systems, Large Language Models, Control

1 Introduction

Integrating artificial intelligence (AI) and machine learning (ML) technologies into power systems has gained significant attention in recent years [1]-[2]. As power grids become more complex and dynamic, traditional control and management approaches struggle to manage vast amounts of data and rapidly changing conditions. Large Language Models (LLMs), a subset of AI known for their impressive capabilities in natural language processing and general problem-solving, offer a promising solution to these challenges [3].

LLMs, such as GPT (Generative Pre-trained Transformer) models, have demonstrated their ability to understand and generate human-like text, perform complex reasoning tasks, and even write code. These capabilities make them potentially valuable tools for power system applications, where they could assist in tasks ranging from fault diagnosis and predictive maintenance to real-time control and optimization of grid operations.

This paper explores the current state of research and applications of LLMs in power systems and presents a practical example of how an LLM agent can be implemented for power system control. We begin with a comprehensive background on power systems and LLMs, followed by a review of existing literature on the intersection of these two fields. We then present our methodology for implementing an LLM agent using Llama 3, a state-of-the-art language model, and integrating it with a power system model developed in Pandapower, a widely used open-source tool for power system modeling and analysis.

Our practical examples illustrate how an LLM agent can be used to monitor system conditions, make decisions, and control various aspects of a power system. We discuss the challenges encountered during implementation, the performance of the LLM agent, and the potential implications for future power system operations.

Our aim is to contribute to the growing body of knowledge on AI-driven power systems and inspire further research and development in this exciting field by bridging the gap between theoretical studies and practical applications. Integrating LLMs into

Contact Alen Bernadić, alen.bernadac@fsre.sum.ba, Faculty of Mechanical Engineering, Computing and Electrical Engineering University of Mostar

©2025 by the Author(s). Licensee IJISE by Faculty of Mechanical Engineering, Computing and Electrical Engineering, University of Mostar. This article is an open-access and distributed under the terms and conditions of the CC BY 4.0 (<https://creativecommons.org/licenses/by/4.0/>)

power systems can enhance grid resilience, improve energy efficiency, and support the transition to a more sustainable and intelligent energy infrastructure.

1.1 Power Systems

Power systems are complex networks designed to generate, transmit, and distribute electrical energy to consumers. These systems consist of various components, including power plants, transformers, transmission lines, substations, and distribution networks. The primary objective of a power system is to ensure a reliable and efficient supply of electricity while maintaining system stability and power quality. Key aspects of power systems include Generation, Transmission, and Distribution. Each part of the system includes control and protection mechanisms, load management, and techniques to balance electricity supply and demand, such as demand response and load forecasting. These measures contribute to grid stability, ensuring the system remains stable under various operating conditions and disturbances. The complexity of power systems has increased significantly in recent years due to factors such as integrating renewable energy sources, distributed generation, electric vehicles, and the need for greater energy efficiency. These challenges have led to a growing interest in applying advanced computational techniques, including AI and ML, to improve power system management and control.

1.2 Large Language Models

Large Language Models are a class of artificial intelligence models designed to understand, generate, and manipulate human language [4]. These models are typically based on deep learning architectures, particularly transformer models, and are trained on vast amounts of text data. Key characteristics of LLMs include:

- a. Scale: LLMs often have billions of parameters, allowing them to capture complex patterns and relationships in language;
- b. Self-attention mechanism: This enables the model to assess the importance of different parts of the input when processing language;

- c. Transfer Learning: LLMs can be fine-tuned for specific tasks after being pre-trained on general language data;
- d. Few-shot Learning: The ability to perform new tasks with minimal task-specific training examples;
- e. Multi-task Capability: LLMs are capable of performing a wide range of language-related tasks, including translation, summarization, and question-answering.

Notable examples of LLMs include GPT (Generative Pre-trained Transformer) series by OpenAI, BERT (Bidirectional Encoder Representations from Transformers) by Google, and, more recently, open-source models like Llama by Meta AI. The capabilities of LLMs extend beyond simple language processing. They have demonstrated proficiency in reasoning, problem-solving, and even code generation tasks. This versatility makes them potentially powerful tools for complex domains like power systems, where they could assist in interpreting data, making decisions, and even generating control strategies.

2 Introduction LLMs in Power Systems: Current Applications and Research

The application of LLMs in power systems is an emerging field with growing interest from both academia and industry [5]. While traditional machine learning techniques have been widely used in power systems for tasks such as load forecasting and fault detection, LLMs offer new possibilities due to their ability to process and generate human-like text, understand context, and perform complex reasoning. Various processes within power systems could be driven by LLMs [6]-[7], opening up significant research and engineering opportunities. Current applications and research areas of LLMs in power systems include anomaly detection and fault diagnosis, predictive maintenance, knowledge management, decision support, and more. Power systems generate vast amounts of technical documentation, standards, and operational procedures. LLMs can be used to create intelligent knowledge management systems that efficiently retrieve relevant information, answer technical questions, and provide decision support to operators

and engineers. LLMs can assist in generating and analyzing various scenarios for grid planning and expansion. By considering multiple factors such as load growth, renewable integration, and regulatory changes, LLMs can help planners explore different options and their potential impacts. As power systems become increasingly digitized, cybersecurity is also a growing concern. LLMs can be employed to analyze network traffic patterns, log files, and threat intelligence reports to identify potential security threats and suggest mitigation strategies. The intermittent nature of renewable energy sources poses challenges to grid stability. LLMs can be used to process diverse data sources, including weather forecasts, historical generation patterns, and grid conditions, to optimize the integration of renewable energy into the grid. LLMs can assist in interpreting and applying complex regulatory requirements. They can be trained on regulatory documents and standards to help ensure that power system operations comply with evolving regulations. AI also shows great potential in areas such as capacity optimization, grid mapping, asset management, and predictive maintenance.

While the application of LLMs in power systems shows great promise, several challenges need to be addressed in future applications:

Data Privacy and Security: Power system data is often sensitive, so the use of LLMs must ensure data privacy and security are protected.

Interpretability and Explainability: The decision-making processes of LLMs can be opaque, which may be problematic in critical infrastructure like power systems where decisions need to be explainable and auditable.

Real-time Performance: Many power system applications require real-time or near-real-time responses. The computational requirements of large LLMs may pose challenges for such applications.

Integration with Existing Systems: Implementing LLMs in power systems requires careful integration with existing SCADA systems, energy management systems, and other control infrastructure.

As research in this field progresses, we can expect to see more sophisticated applications of LLMs in power systems, potentially leading to more intelligent, efficient, and resilient grid operations.

3 Methodology

Our methodology for implementing an LLM agent for power system control involves several key steps, as shown in the following subsections.

3.1 Power System Modeling

We use Pandapower [8], an open-source Python library for power system modeling and analysis. Pandapower enables the creation of a realistic power system model, including generators, loads, transmission lines, and transformers. The model is easily manipulated and analyzed, providing an ideal environment for testing our LLM agent.

3.2 LLM Selection and Implementation

In this work, we used the Llama 3 model, a state-of-the-art language model developed by Meta AI [9]. Llama 3 offers an excellent balance of performance and efficiency, making it well-suited for our power system control application. For run the selected model locally on PC, we use Ollama [10], an open-source tool for running Llama models locally, which provides flexibility and control over the model's deployment. Ollama is a platform designed to simplify the process of running and deploying large language models (LLMs) locally. Ollama allows users to download and run various open-source LLM models on their local machines, providing a more flexible and secure environment for interacting with AI models. It primarily focuses on enabling AI models to be used directly on personal or enterprise systems without the need for continuous cloud connectivity. Key features of Ollama include:

Local Model Execution: One of the features of Ollama is its ability to run AI models locally on user devices.

Multi-Model Support: Ollama supports various types of large language models (LLMs) that can handle tasks such as text generation, summarization, and more. Its flexible architecture enables users to easily integrate and switch between different models.

Cost-Effective: Running LLM models locally can lead to significant cost savings compared to cloud-based solutions or commercial solutions, which often involve expensive fees for computation and storage.

Offline Capability: For applications where internet connectivity is not guaranteed, such as in remote areas or high-security environments (which is often the case for large power plants), Ollama's ability to run AI models offline is highly valuable.

Privacy and Security: With growing concerns over data privacy, many organizations hesitate to use cloud-based AI services due to the potential risks of exposing sensitive information. Ollama addresses this by keeping all data and model interactions local, ensuring no data is sent to the cloud.

3.3 Agent Design

Our LLM agent is designed to perform the following tasks:

- Monitor system conditions by interpreting Pandapower simulation outputs;
- Make decisions based on predefined criteria and learned patterns;
- Generate control actions to maintain system stability and efficiency;
- Provide explanations for its suggestions to the operational staff in natural language.

3.4 Integration

We developed a Python script that integrates the Pandapower model with the Llama 3 agent and the Ollama interface. This script handles the flow of information between the power system simulation and the LLM, as well as the execution of control actions. All modules are part of the Python ecosystem and run within a single script. Python modules are installed using Anaconda in a conda-managed environment [11]. The only requirement is that the Ollama server is installed and running on <https://11434> local IP address [10].

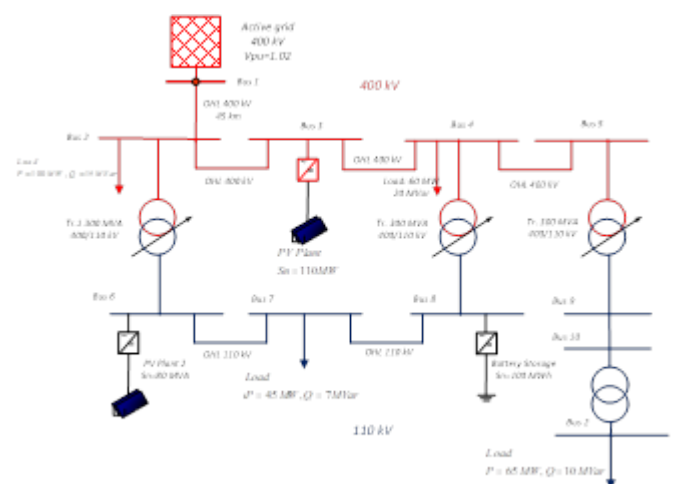
4 Practical Example 1: LLM Agent for Power System Analysis and Control

Experiments on integrating LLMs with power system control are progressing intensively [12]. Researchers are working on power system simulation cases involving LLM-based control and surveillance logic. Our work focuses on optimizing power system operations, as presented in [13]. This section provides a practical example of using Python to implement an LLM agent for power system control. We will break this down into three main parts: power system modeling with Pandapower, LLM agent implementation using Llama 3 and Ollama, and the integration of these components.

4.1 Power System Modeling with Pandapower

The complex configuration of the 400/110 kV power system network is selected for a practical case, represented by a single-line diagram in Figure 1. Using Pandapower, the network is modeled with large-capacity, controllable PV plants and battery storage added at several nodes to increase complexity.

Pandapower implementation of 400/110 kV power network with additional PV plants and battery storages is on Figure 2. Network is designed with additional PV plants and battery storages for ability of injecting active and reactive power. Power system transformers are equipped with tap-changers for local voltage regulations. Loads are also positioned on buses and parametrized with reactive and active



components. Initial conditions of power system is intentionally made for large overvoltages as input state for LLM agent, which has task of stabilizing voltages and the power system in overall. The initial is 8 – 9 buses with voltages out of the desired range of 1.00-1.05 per unit (p.u.), which is a complicated task to stabilize with a set of simultaneous actions of tap-changers and active and reactive injections of PV plants inverters and battery storages which is LLM model task.

```
def create_network():
    net = pp.create_empty_network()
    buses = {f'bus{i}': pp.create_bus(net, vn_kv=400 if i <= 6 else 110, name=f"Bus
    {i}")
            for i in range(1, 13)}

    pp.create_ext_grid(net, bus=buses['bus1'], vm_pu=1.02, name="Grid Connection") #
    Changed vm_pu to 1.02
    # Create lines
    line_data = [
        ('bus1', 'bus2', 80), ('bus2', 'bus3', 100), ('bus3', 'bus4', 70),
        ('bus4', 'bus5', 80), ('bus5', 'bus6', 90), ('bus7', 'bus8', 40),
        ('bus8', 'bus9', 50), ('bus9', 'bus10', 60), ('bus10', 'bus11', 45),
        ('bus11', 'bus12', 55)
    ]
    for from_bus, to_bus, length in line_data:
        pp.create_line(net, from_bus=buses[from_bus], to_bus=buses[to_bus],
        length_km=length,
                        std_type=f"490-AL1/64-ST1A {'380.0' if int(from_bus[3:]) <= 6 else
        '110.0'}")
    # Create transformers
    for hv_bus, lv_bus in [('bus2', 'bus7'), ('bus4', 'bus9'), ('bus6', 'bus12')]:
        pp.create_transformer_from_parameters(
            net, hv_bus=buses[hv_bus], lv_bus=buses[lv_bus], sn_mva=300, vn_hv_kv=380,
            vn_lv_kv=110,
            vkr_percent=0.25, vk_percent=12, pfe_kw=100, i0_percent=0.1, shift_degree=0,
            tap_side="hv", tap_neutral=0, tap_min=9, tap_max=9, tap_step_percent=1.5,
            tap_pos=0)
    # Create loads
    load_data = [('bus3', 135, 27), ('bus4', 115, 23), ('bus8', 85, 17), ('bus10', 65,
    13), ('bus11', 185, 21)]
    for bus, p, q in load_data:
        pp.create_load(net, bus=buses[bus], p_mw=p, q_mvar=q)
    # Create PV plants (increased capacity)
    pv_data = [('bus2', 105, 55), ('bus5', 85, 45), ('bus7', 95, 50), ('bus9', 100, 52),
    ('bus11', 90, 47)]
    for i, (bus, p, q) in enumerate(pv_data, 1):
        pp.create_sgen(net, bus=buses[bus], p_mw=p, q_mvar=0, name=f"PV{i}",
            controllable=True, max_q_mvar=q, min_q_mvar=-q, max_p_mw=p,
            min_p_mw=0)
    # Create battery storage systems (increased capacity)
    battery_data = [('bus3', 130, 65), ('bus6', 110, 55), ('bus8', 90, 45), ('bus10',
    100, 50), ('bus12', 120, 60)]
    for i, (bus, e, p) in enumerate(battery_data, 1):
        pp.create_storage(net, bus=buses[bus], p_mw=0, max_e_mwh=e, soc_percent=50,
            name=f"Battery{i}", max_p_mw=p, min_p_mw=-p)

    return net

import pandapower as pp
import pandapower.networks as pn
import matplotlib.pyplot as plt
import networkx as nx
import numpy as np
from ollama import Client
import re

def create_network():
    net = pn.case14()
    pp.create_sgen(net, 2, p_mw=50, q_mvar=0, name="PV_1")
    pp.create_sgen(net, 5, p_mw=40, q_mvar=0, name="PV_2")
    pp.create_storage(net, 3, p_mw=0, q_mvar=0, max_e_mwh=100, soc_percent=50,
    name="Battery_1")
    pp.create_storage(net, 6, p_mw=0, q_mvar=0, max_e_mwh=80, soc_percent=50,
    name="Battery_2")
    return net
```

Figure 2 Pandapower implementation of 400/110 kV system with PV plants and battery storages

4.2 LLM agent implementation

The 400/110 kV power system network model, shown in the single-line diagram in Figure 1, is created in Pandapower for further integration into the overall Python ecosystem. An AI agent based on the Llama 3 open-source LLM model is implemented via the Ollama platform to analyze and control the power network. The Llama 3-based LLM agent, by

controlling the tap-changer and injecting active and reactive power, aims to stabilize the network and reduce overvoltages.

4.2.1 Presumptions for model design

In the literature [14]-[16], transformer tap-changer operations and the injection of active and reactive power from PV plant inverters and battery storage systems have been shown to stabilize voltages within the desired operational range. These are the physical and engineering assumptions used to guide the design of the Llama 3 LLM agent. The agent aims to stabilize bus voltages through a set of synchronized actions, including the manipulation of transformer tap-changers and the injection of active and reactive power from PV plant inverters.

4.2.2 Prompt design

While Llama 3 is pre-trained on a large corpus of text, it does not possess specific knowledge about voltage stabilization. In this initial model, we provide the context through a detailed prompt, without fine-tuning or using RAG on power system operational data, technical documentation, or example scenarios (Figure 3). This specific prompt, as input to the LLM,

```
prompt = f"""Given the following voltage profile in a power network:
Voltages: {voltages}
Overvoltage buses (>1.051 p.u.): {overvoltage}
Undervoltage buses (<0.998 p.u.): {undervoltage}

You are a power system control agent. Your goal is to bring all voltages within
0.9985-1.0515 p.u. in 5 actions or less.
Use all known power system theory to achieve goals.

Pandapower Network information:
Number of transformers: {network_info['num_transformers']}
Transformer indices: {network_info['transformer_indices']}
Number of PV plants: {network_info['num_pv_plants']}
PV plant indices: {network_info['pv_plant_indices']}
Number of batteries: {network_info['num_batteries']}
Battery indices: {network_info['battery_indices']}

Suggest actions to correct these violations using tap changers, active and reactive power
control of PV plants, and active power control of batteries.

Provide your response as a Python dictionary with the following structure:
{{
    'tap_changes': {{trafo_index: tap_change, ...}},
    'q_changes': {{pv_index: q_change, ...}},
    'p_changes': {{pv_index: p_change, ...}},
    'battery_changes': {{battery_index: p_change, ...}}
}}

where indices are integers within the provided ranges, tap_change is an integer (+1 for
up, -1 for down),
q_change and p_change are floats representing the change in Mvar and MW for PV plants,
respectively,
and p_change for batteries is a float representing the change in MW (positive for
discharging, negative for charging).
Make aggressive changes to correct violations quickly. Limit changes to ±1 taps, ±12 Mvar
and ±10 MW for PV, and ±8 MW for batteries per step.
Remember:
1. Injection of +Mvar raises voltage in the bus, injection of -Mvar decreases voltage.
2. Increasing tap changer raises voltages, decreasing tap changer decreases voltages in
nodes where transformer is connected.
3. Injecting active power of PV plants and batteries raises voltages.

All voltages after 5 steps of actions must be in range 0.99 - 1.052 p.u. as soon as
possible (0.99 <= v <= 1.052), within 5 taken actions of llama agent.
In nodes with overvoltages inject -Mvar or - MW
In nodes with undervoltages inject +Mvar or +MW
After every step consider overall network state and consider where to inject active or
reactive power

ONLY return the Python dictionary. Do not include any explanations or additional text.
```

Figure 3 Prompt designed for llama3 agent context of power system control

helps the model understand the context and terminology specific to power systems.

4.2.3 Model execution

The Pandapower 400/110 kV transmission and distribution network is simulated with overvoltages and loaded into a Python environment, where the Llama 3 model is run via the Ollama server. Initially, nine buses in the system have voltages outside the desired range of 1.00-1.05 per unit (p.u.).

Visualization is created using the NetworkX [17] Python library, with nodes representing buses, and overvoltage nodes are highlighted in red, as shown in Figure 4. The Llama 3 agent optimizes the Pandapower 400/110 kV network parameters in the model's five steps. After each step, the updated network state is loaded into the LLM prompt for processing, the network parameters are modified, and power flow is recalculated to assess overall network stability. After the five-step execution, the network state is satisfactory, with only a small pair of

overvoltages remaining, which can be further addressed through actions on lower voltage levels. [16].

It is evident from Figures 4 and Figure 5 that with each action set of the Llama 3-based LLM agent, the power system's voltage profile stabilizes within the desired range of 1.00-1.05 p.u. After five sets of actions on the tap-changers of transformers and the injection of active and reactive power from PV plants and battery storage systems, the efficiency and stability of the observed network are significantly improved, along with valuable comments and recommendations generated as LLM output.

4.2.4 Result analysis

For 20 model executions, we obtained a distribution of results, as shown in Figure 6. Over 70% of executions produced very good results, with 0-2 overvoltages after the five-step model execution. Considering that only prompt contexting was used for the LLM model, without fine-tuning or the use of the

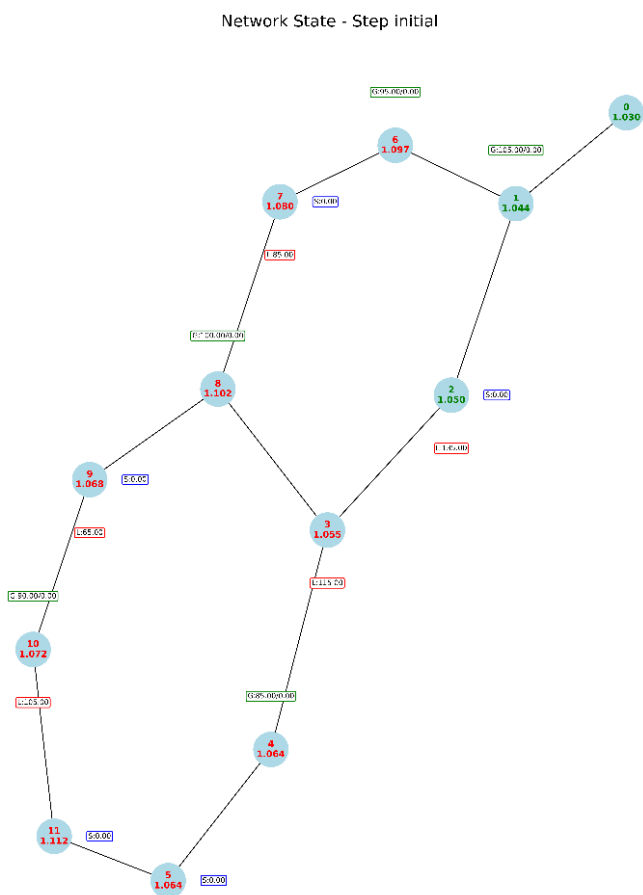


Figure 5 Initial state of the power network with overvoltages

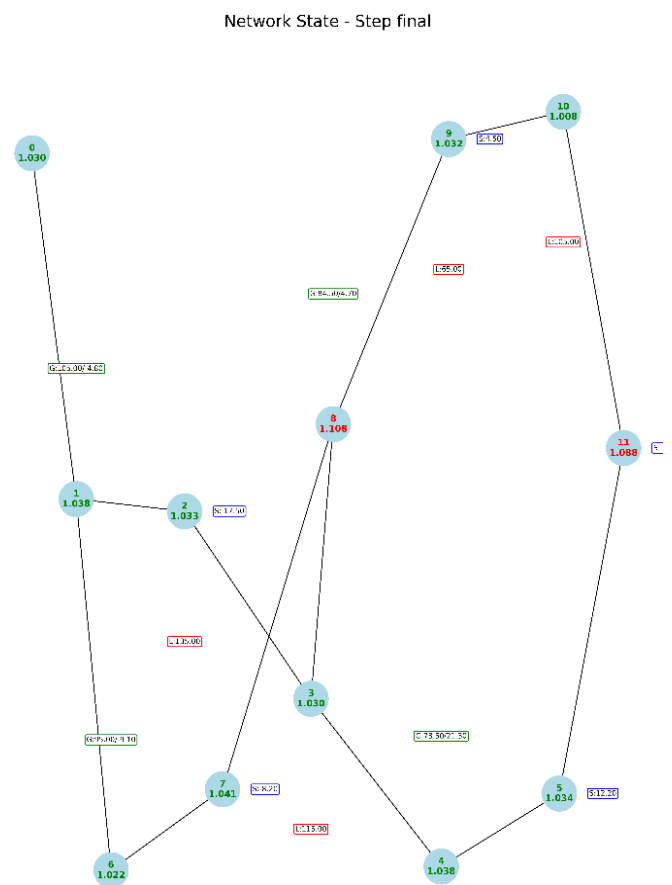


Figure 4 Power network voltage profile after execution of five action sets of llama3 agent

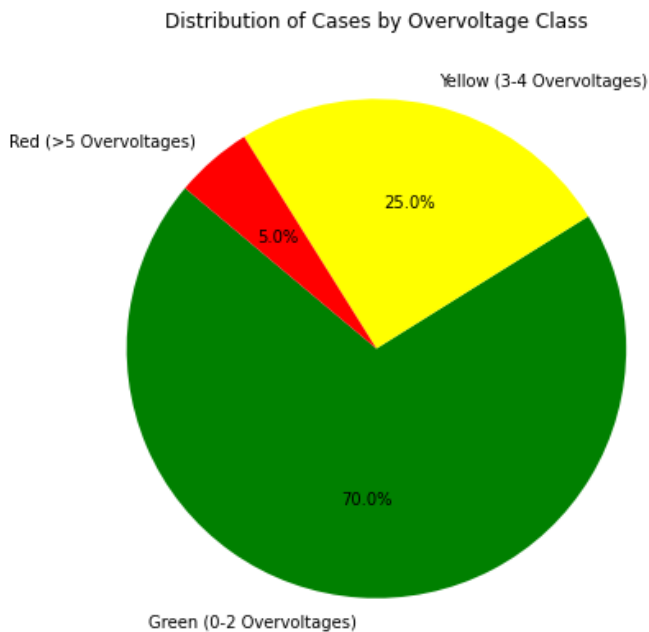


Figure 6 Analysis of model executions

Retrieval Augmented Generation (RAG) mechanism, the results are more than satisfactory for the Llama 3 open-source model.

5 Conclusion and Future Work

This study has demonstrated the potential of Large Language Models in enhancing power system control and decision-making processes. The LLM agent showed remarkable capabilities in analyzing complex system states, providing context-aware recommendations, and generating multi-step control strategies.

Integrating natural language processing with power system control opens up new possibilities for intuitive human-machine interfaces in control rooms, automated report generation, and knowledge management in power utilities. The ability of LLMs to process and synthesize information from diverse sources could prove invaluable in managing the increasingly complex and data-rich environment of modern power grids.

However, the challenges identified in this study, particularly those related to numerical precision, real-time performance, and explainability, highlight the need for further research and development. Future work should focus on:

1. Developing hybrid systems that combine the strengths of LLMs with traditional optimization and control algorithms;
2. Investigating the use of reinforcement learning techniques to allow the LLM agent to improve its decision-making over time-based on system outcomes;
3. Developing the Retrieval Augmented Generation (RAG) mechanism to improve model accuracy;
4. Developing a multi-agent system for effective collaboration to achieve shared goals;
5. Exploring the application of LLMs in other areas of power system operation, such as long-term planning, renewable energy integration, and cybersecurity;
6. Addressing the explainability and transparency of LLM-based decision-making systems, which is crucial for their adoption in critical infrastructure;
7. Investigating the potential of multi-modal LLMs that can process not only textual data but also visual and time-series data, which are common in power system monitoring.

Special attention will be given to the security issues associated with LLM applications in power systems [18].

In conclusion, while there are challenges to overcome, the application of Large Language Models in power systems represents a promising frontier in the ongoing digital transformation of the energy sector [19]. As these technologies continue to evolve, they have the potential to play a significant role in creating more intelligent, efficient, and resilient power grids of the future.

Conflicts of Interest: The authors report there are no competing interests to declare;

References

- [1] S. Azad and M. Nazari-Heris, Eds., *Artificial Intelligence in the Operation and Control of Digitalized Power Systems*. in *Power Systems*. Cham: Springer Nature Switzerland, 2024. doi: 10.1007/978-3-031-69358-8.
- [2] Q. A. Al-Haija, O. Mohamed, and W. A. Elhaija, *Advances in AI for Simulation and Optimization of Energy Systems*, 1st ed. Boca Raton: CRC Press, 2025. doi: 10.1201/9781003520498.
- [3] Y. Zhang, "Application of Large Language Models in Power System Operation and Control", *JCEIM*, vol. 15, no. 3, pp. 79–83, Dec. 2024, doi: 10.54097/sb9qdz28.
- [4] P. Martra, *Large Language Models Projects: Apply and Implement Strategies for Large Language Models*. Berkeley, CA: Apress, 2024. doi: 10.1007/979-8-8688-0515-8.
- [5] K. Satterlee, "Large Language Models May Revolutionize the Electric Energy Sector". Accessed: Jan. 15, 2025. [Online]. Available: <https://engineering.tamu.edu/news/2024/06/large-language-models-may-revolutionize-the-electric-energy-sector.html>
- [6] R. Li et al., "LLM-Based Frameworks for Power Engineering from Routine to Novel Tasks", 2024. doi: 10.2139/ssrn.4741095.
- [7] C. Huang, S. Li, R. Liu, H. Wang, and Y. Chen, "Large Foundation Models for Power Systems", Dec. 12, 2023, arXiv: arXiv:2312.07044. doi: 10.48550/arXiv.2312.07044.
- [8] L. Thurner et al., "Pandapower—An Open-Source Python Tool for Convenient Modeling, Analysis, and Optimization of Electric Power Systems", *IEEE Trans. Power Syst.*, vol. 33, no. 6, pp. 6510–6521, Nov. 2018, doi: 10.1109/TPWRS.2018.2829021.
- [9] Meta, "Introducing Meta Llama 3: The most capable openly available LLM to date", ai.meta.com, Apr. 18, 2024. <https://ai.meta.com/blog/meta-llama-3/>
- [10] "Ollama". Accessed: Jan. 15, 2025. [Online]. Available: <https://ollama.com>
- [11] "Anaconda | The Operating System for AI", Anaconda. Accessed: Jan. 15, 2025. [Online]. Available: <https://www.anaconda.com/>
- [12] M. Jia, Z. Cui, and G. Hug, "Enabling Large Language Models to Perform Power System Simulations with Previously Unseen Tools: A Case of Daline", Nov. 19, 2024, arXiv: arXiv:2406.17215. doi: 10.48550/arXiv.2406.17215.
- [13] K. V. Arnautov and D. A. Akimov, "Application of Large Language Models for Optimization of Electric Power System States", in *2024 Conference of Young Researchers in Electrical and Electronic Engineering (ElCon)*, Saint Petersburg, Russian Federation: IEEE, Jan. 2024, pp. 314–317. doi: 10.1109/ElCon61730.2024.10468377.
- [14] A. Bernadić, G. Kujundžić, and I. Primorac, "Primjena algoritama podržanog učenja u upravljanju elektroenergetskog sustava", 3. *Savjetovanje BH CIRED*, Mostar, 2022.
- [15] A. Bernadić, G. Kujundžić, and I. Primorac, "Reinforcement Learning in Power System Control and Optimization", *B&H Electrical Engineering*, vol. 17, no. 1, pp. 26–34, Jul. 2023, doi: 10.2478/bhee-2023-0004.
- [16] A. Bernadić, et al. "Primjena podržanog učenja u regulaciji napona EES-a upravljanjem reaktivnih snaga fotonaponskih elektrana", 4. *Savjetovanje BH Cired*, Mostar, 2024.
- [17] "NetworkX — NetworkX documentation". Accessed: Jan. 15, 2025. [Online]. Available: <https://networkx.org/>
- [18] Ruan et al., "Applying Large Language Models to Power Systems: Potential Security Threats", *IEEE Trans. Smart Grid*, vol. 15, no. 3, pp. 3333–3336, May 2024, doi: 10.1109/TSG.2024.3373256.
- [19] S. Majumder et al., "Exploring the capabilities and limitations of large language models in the electric energy sector", *Joule*, vol. 8, no. 6, pp. 1544–1549, Jun. 2024, doi: 10.1016/j.joule.2024.05.009.