LEVERAGING LARGE LANGUAGE MODELS FOR A RESILIENT ELECTRICITY SUPPLY SYSTEM

Abdullah Binjos[2,+], Grégory Mermoud[2,+], Baoling Guo[1,+], Jérémie Vianin[2], Julien Pouget[1], David Wannier[2]*

[1]*Institute of Energy and Environment (IEE), HES-SO Valais-Wallis, Sion, Switzerland*
[2]*Institute of Informatics (II), HES-SO Valais-Wallis, Sierre, Switzerland*

[+] *These authors contributed equally. *Corresponding author: david.wannier@hevs.ch*

Keywords: LARGE LANGUAGE MODEL (LLM), AI AGENTS, TEXT-TO-SQL, RETRIEVAL AUGMENTED GENERATION (RAG), MICROGRID RESILIENCE

Abstract

This work introduces Large Language Model (LLM)-based agent that interacts with a resilient microgrid platform, enabling users to interact via intuitive text-based queries. The system provides both expert knowledge, scenario testing, and decision-making support related to resilient electricity systems while ensuring data privacy through local hosting and GDPR compliance. Users can explore strategies such as leveraging local energy production and storage units, shedding less critical loads to mitigate blackouts' impact. Enhanced with Geographic Information System (GIS) capabilities, the platform, developed during the European project OpenGIS4ET [1], offers actionable insights and visualizations to support microgrid resilience. This framework, tested with the use case and dataset relevant to the wastewater treatment plant (STEP) in Neuchâtel, is designed to be scalable and replicable across different regions in a next step.

1. Introduction

Modern power systems have been increasingly challenged by complex interdependencies among components and the heightened frequency of extreme weather events caused by climate change. These issues pose significant risks, such as power interruptions or blackouts, with severe consequences for society and the economy. These risks are outlined in the national risk analysis report by Swiss Federal Office for Civil Protection (FOCP) [2], which further emphasize the critical need for resilience strategies within critical infrastructures. A project mandated by FOCP and coordinated by HES-SO in collaboration with local distribution system operator (DSO) Viteos in Neuchâtel is therefore proposed to study how to enhance resilience of electricity supply systems to maximize the survivability of critical consumers during the blackouts.

The increasing integration of distributed energy sources like photovoltaics (PV), electric vehicles (EVs) through Vehicle to Grid (V2G), and storage units, facilitates the creation of microgrids that can operate autonomously, providing both economic and contingency benefits during emergencies, from hours to several days [2]. The project investigates microgrid solutions to improve electricity supply resilience by leveraging local renewable energy sources and diverse storage units. The project combines interdisciplinary research and stakeholder collaboration, blending technical and social science perspectives. The paper investigates the integration of an LLM-powered agent that allows end users to interact with a microgrid platform in natural language. The final goal is to help improve the resilience of the electricity supply system within critical infrastructures by answering technical questions related to the underlying microgrid, as well as general energy-related questions. The data used in this work pertains to the wastewater treatment plant (STEP) in Neuchâtel, identified as a critical infrastructure.

Prior works include ChatGrid™, developed by the Pacific Northwest National Laboratory (PNNL) [3], whose focus is however on the broader U.S. electrical transmission grid and their OpenAI's models are incompatible with the confidentiality requirements of our project. The authors in [4] review the use of large language models (LLM) in the electric energy sector, such as power flow analysis, diagnostics, and forecasting within traditional power systems. In contrast, our paper has a specific focus on microgrids solutions for resilience, and the specific regulatory constraints associated with representative critical infrastructures defined in the national risk report [2].

The rest of this paper is organized as follows: Section 2 describes the dataset imported in the LLM for use case test; Section 3 describes the methodology, the LLM model, and the evaluation benchmarks; Section 4 will describe and discuss the experimental results. Finally, conclusions and perspectives are covered in Section 5.

2. Testbed and Datasets

The local STEP authorises us access to historical imported and exported energy data from the STEP zone in 2023. Data access agreements are signed with all relevant partners. The energy data are measured separately by two metering devices, recording power flows in opposite directions at the transformer's output, then recorded in energy format [kWh] in

15-minute intervals. Note that the imported energy represents the net energy purchased from the grid, namely the total consumption minus local production. The exported energy refers to the surplus production that is sold back to the grid.

The STEP has some production by roof-installed PV panels, and more PVs will be installed in 2025. Additionally, the STEP recycles the organic matter extracted from sludge to produce biogas. The gas produced is stored in two 1000 Nm³ tanks. This biogas is then fed into a heat and power coupling (HPC) consisting of a combustion engine and a generator that produce electricity and heat. The surplus energy is transferred to electricity grid and district heating system, respectively. Local production and storage capacity strongly enhance its self-consumption efficiency for both electricity and heat during the backouts. At STEP site, the PV power is produced in the daytime, and two biomass-powered generator units operate during the night. Figure 1 presents the daily, weekly, and monthly energy data recorded at the STEP site over the entire year of 2023. The energy profiles vary significantly depending on the season. In summer, the energy purchased from the grid is almost negligible for the STEP, making it fully self-sufficient. In winter, however, the STEP still relies heavily on the electricity supplied by the DSO.

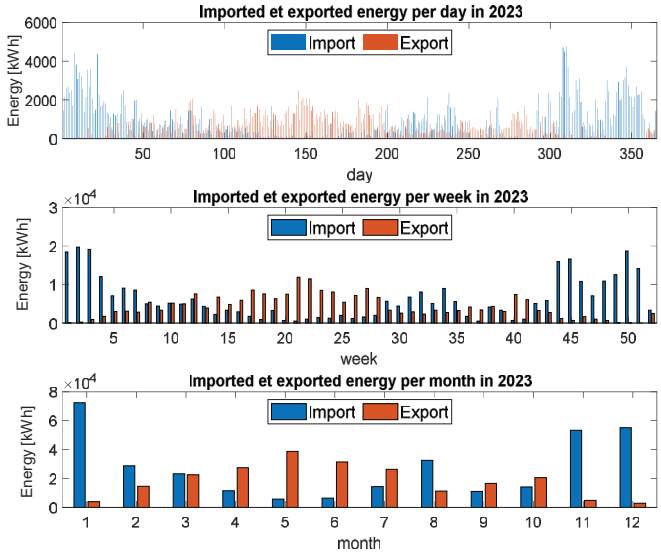

Figure 1: The imported and exported energy historical data from/to the grid per day, week, and month in 2023. The agent is using these data to answer questions.

This basic version dataset is integrated into the LLM-powered agent described in Section 3.2 to test its functionalities and verify the precision of the LLM model used. In the next step, the dataset will be advanced by incorporating the grid topology data, the order of critical loads and shedding rules during blackout co-defined with the electrical engineer from the STEP. Cases study results of microgrids solutions will be also integrated to answer more complex questions.

Besides the measured energy data for specific microgrid sites, to answer energy-related technical questions, the dataset is made up of scientific articles, reviews, and conference *proceedings*. The PDFs used are all non-confidential and allow the LLM to interact when the user asks general questions about energy. The next section describes in further details how the system leverages such data.

3. Methods

To build a system capable of answering both general and system-specific questions, we adopt an *agentic architecture*, which has multiple components as shown in Figure 2. Section 3.1 describes the high-level workflow whereas Section 3.2 describes further the system architecture and the tools available to the agent. Finally, the LLMs used in the system are described in Section 3.3.

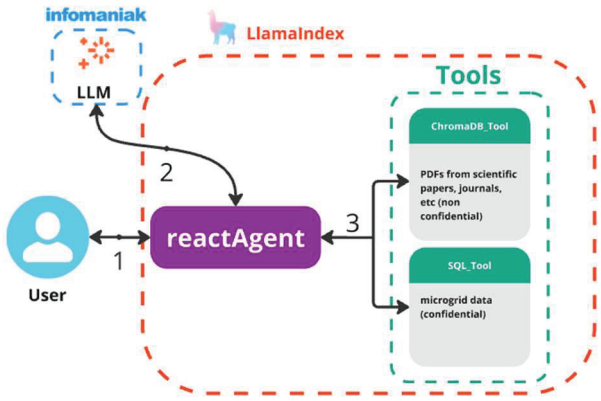

Figure 2: Block diagram of the system architecture, which includes a ReAct agent endowed with two distinct tools (`SQL_Tool` and `Chroma_Tool`). Note how the agent is at the core of the system and responsible for interacting with (1) the end user, (2) the tools, and (3) the LLM itself.

3.1. High-level Workflow

The user first submits a question using an interface hosted on the project's web platform. This question is then transmitted to the agent, which operates iteratively. At every step, the agent prompts an LLM to determine the *action* to be performed. Actions often consist in retrieving further information from external sources, which are accessible via *tools* (see Section 3.2). A rough sketch of the agent's workflow is as follows:

1) The LLM is prompted with the user's request. Its first task is to determine which tool is pertinent to fulfill this request.
2) Once the tool has been defined, the agent prompts again the LLM with fixed, tool-specific instructions (e.g., the database schema) and the variable user's request and contextual information (e.g., previous interactions or tool usages).
3) Based on the LLM's output, the tool is then executed by the agent (e.g., a SQL query is executed against the database), and the LLM is prompted again with the outcome of the tool (e.g., the result of the SQL query).
4) If the LLM has enough information, it may finally produce an answer, which is returned to the end user. If not, it goes back to Step 1, except that the LLM is now

provided with contextual information about previous steps it performed.

The above workflow is implemented using an open-source software framework called LlamaIndex [6].

3.2. ReAct Agent and Tools

As briefly described above, the agent is responsible for: (1) deciding which tool is the most appropriate to carry out the user's request, (2) generating the appropriate input of the tool, and (3) interpreting the tool's output to provide a natural language response to the user. We leverage the ReAct strategy [7], which is an improvement over the 'Chain of Thoughts' (CoT) prompting scheme, and has been shown to reduce hallucinations and error propagation by enforcing a deliberate reasoning step (Thought) prior to generating a tool use (Action). The LLM continuously evaluates the intermediate results that it obtains, to determine if they are sufficient to answer the user's request or if further queries are necessary to provide a suitable answer.

Two tools are used by the agent to answer questions: `SQL_Tool` and `Chroma_Tool`. The `SQL_Tool` allows the LLM to access an SQL database that contains the microgrid data by generating SQL queries, which works as follows:

- When a question is asked, for example 'What is the average consumption of the microgrid in January?', the agent analyses the question and identifies the appropriate tool to answer it. When the question requires specific data from the microgrid, it picks the `SQL_Tool` tool.
- The tool injects the database (DB) schema into the prompt, allowing the agent to examine the relevant columns. In this example, the `consumption` column would be selected.
- The agent generates an SQL query to retrieve the necessary data, such as the monthly consumption values for January.
- If the data returned are not sufficient to answer the question (e.g., no average is calculated), the agent performs an additional 'chain of thoughts' to produce a complete answer.

To avoid errors linked to the model (such as some spurious escaping characters produced by `mixtral-8x22B`), specific guidelines may be included: 'Never escape the underscore character in field, table name, and values'.

The description of the tool, which is a key element enabling the agent to select the appropriate tool, is as follows: 'This tool processes SQL queries about the actual microgrid. Provide a valid SQL query as input to retrieve data from the table `energyData`.'

`Chroma_Tool` is an implementation of Retrieval Augmented Generation (RAG) that allows the agent to access non-confidential PDF files, such as scientific papers, technical documentation, etc. This tool should be used when the questions concern energy or microgrids in the broadest sense. All relevant text sources (PDFs, manuals) are parsed, chunked, and embedded using a transformer-based embedding model, producing vector representations that capture semantic meaning. All embeddings are produced using the `all-MiniLM-L6-v2` model, and are then stored locally in the open-source database Chroma [8], which we host locally.

The tool is described to the LLM as follows: 'Chroma retrieval augmented generation (RAG) tool for microgrid.' When a user's request pertains to energy (e.g., "Explain the implications of voltage drop in rural areas"), the LLM may select the `Chroma_Tool` to the most contextually relevant passages based on similarity scores.

3.3. Large Language Models

To orchestrate the interaction between both tools, we conducted extensive evaluations with language models of various sizes:

- `llama3.2-3B` (3B parameters): this smaller model is part of the LLaMa-3 family [9] is able to handle simple RAG requests effectively. However, it struggles to manage the complexity of the entire infrastructure, prompting the move to larger models like `llama3.3-70B`.
- `mixtral-8x22B` (mixture of 8 x 22B experts): this model is larger than `llama3.2-3B` and uses a Mixture of Expert (MoE) architecture [10]. However, it cannot achieve the same level of understanding and integration capabilities as `llama3.3-70B`.
- `llama3.3-70B` (70B parameters): due to the limitations observed with the smaller model of the LLaMa-3 family, we transition to the 70 billion parameter model `llama3.3-70B` [9], hosted by Infomaniak in Switzerland to be compliant with nFADP (New Federal Act on Data Protection) and to keep dataset within Swiss infrastructure. As shown in Section 4.1, this significantly larger model demonstrates much superior capabilities to handle instructions and effectively utilize tools.

3.4. Hosting and Privacy Considerations

Due to the confidential nature of the microgrid data, we could not utilize OpenAI's GPT-4 for this specific project. However, for public or non-confidential datasets, such models should be considered, along with other flagship commercial models. Instead, we explore two solutions for the inference of our models: (i) the Swiss provider Infomaniak and (ii) a self-hosting platform:

Infomaniak: we use this provider for the hosting of the `llama3.3-70B` and `mistral-8x22B` models. Being Swiss-based and ISO 27001 certified, it fulfils nearly all requirements of security, data privacy, compliance with the Swiss New Federal Act on Data Protection (nFADP) and industry-specific confidentiality requirements.

Ollama: we use Ollama [11] as a self-hosting solution for `llama3.2-3B`. It provides benefits such as reduced latency and

the highest standard of data privacy, being a fully local solution.

3.5. Benchmarks and Performance Metrics

The performance of the system is assessed using two distinct benchmarks:

Benchmark 1 (text-to-SQL) assesses the ability to answer microgrid-specific questions using the `SQL_Tool` based 20 questions that require the model to access the SQL database and extract specific information about the underlying microgrid. These questions have varying levels of difficulty. Examples of such questions include:

- "When was the top 10 highest spikes in energy demand (bought_kwh)?"
- "What is the avg per day of bought kWh for March?"

Benchmark 2 (tool selection) assesses the ability of the model to select the appropriate tool based on the user's query. It consists of 10 questions: 5 specific questions that require the `SQL_Tool` and 5 general questions aimed at `RAG_Tool`. The questions are shuffled randomly and presented to the agent for response generation. Sample questions are shown here:

- `SQL_Tool`: If a blackout occurs during the 5th week of the year at the microgrid, how much energy needs to be shed to ensure its safe operation for one week?
- `RAG_Tool`: How does adaptive inertia suppress harmonic interactions between renewable energy sources and conventional generators in high-voltage grid regions?

3.6. Performance Metrics

For both benchmarks, we compute an overall accuracy, i.e., the fraction of correct answers across the whole benchmarks. Furthermore, for Benchmark 1, we also analyze failure modes of the system. We classify errors into one of the following three categories: **Type 1:** bad query (query does not address the question); **Type 2:** no query (improper query formatting or re-prompting); **Type 3:** bad interpretation of results (correct query, but misinterpreted output).

4. Results

4.1. Benchmark 1: Assessing Text-to-SQL Capabilities

Table 1 summarizes the results of Benchmark 1. The `llama3.2-3B` model achieves an accuracy of 0%, being unable to even perform any tool selection. Such small models seem to be unable to tackle the complexity associated with the ReAct framework. The larger model `mistral-8x22B` can answer some of the simpler questions, but it still produces incorrect queries quite often. Meta's flagship model `llama3.3-70B` achieves the best performance, even for non-trivial questions as shown in Table 2. All failures observed (20% of all questions) are of type 3, i.e., the agent is unable to interpret the output of an otherwise correct query. This issue originates from large query results in overwhelming the model's context window. Consequently, the agent occasionally enters an infinite loop of self-prompting.

Table 1: Performance metric for Benchmark 1 for two models.

Metric/Model	`mistral-8x22B`	`llama3.3-70B`
Accuracy	25%	80%
Bad query	60%	0%
No query	0%	0%
Bad interpretation	15%	20%

Table 2: Question on the microgrid using SQL_Tool

Step	**Content**
Question	If a blackout occurs during the 5th week of the year at the microgrid, how much energy needs to be shed to ensure its safe operation for one week?
Tool selection	`SQL_Tool`
LLM query	`SELECT SUM(bought_kwh) AS total_bought_kwh, SUM(sold_kwh) AS total_sold_kwh FROM cireddata WHERE WEEK(timestamp) = 5;`
Tool output	`total_bought_kwh = 7098.0, total_sold_kwh = 3842.75`
Final answer	The amount of energy to shed is 3255.25 kWh.

Table 2 illustrates the successful execution of the agent using `llama3.3-70B`. The model has correctly inferred that a safe operation is required to compare the total amount of bought and sold energy during that week. However, the query does not compute this difference directly, but the LLM did it on its own, which LLMs are notoriously poor at arithmetic [12].

4.2. Benchmark 2: Assessing Tool Selection Capabilities

Both `llama3.3-70B` and `mistral-8x22B` can consistently select the correct tool with perfect accuracy, ensuring a strong foundation for subsequent query processing. This result is interesting, as it suggests that models have a sufficiently deep understanding of the user's request to appropriately determine whether it requires specific data or general knowledge. On the other hand, the smaller `llama3.2-3B` model is unable to process basic chain of thought prompts, failing to generate correct responses for all 10 test questions.

5. Conclusion and future work

This paper introduces a proof-of-concept of LLM-GIS interactions using ReAct-style agents based on `llama3.3-70B`. The results reported in this paper indicate that such a system is effective at handling microgrid-specific tasks, achieving high accuracy in SQL-based questions and flawless tool selection. Its primary limitation lies in handling large SQL query results, which will be improved in the next steps. Another finding of

our work is that smaller models such as `llama3.2-3B` are unable to tackle such tasks effectively, highlighting the necessity of larger models for such complex applications.

Future work will focus on enhancing robustness and overall effectiveness of the system. The primary goal is to optimize how the agent handles user's requests that lead to large query outputs from the two following perspectives:

Addressing context flood issue: The primary improvement would involve customizing the SQL tool to limit the amount of data returned, and to adjust the prompts accordingly. For instance, the tool could limit the size of query results and trigger predefined fallback actions to handle the overflow when the output exceeds this threshold. Combined with this, prompts should instruct the model to produce SQL queries that aggregate results, ensuring the output is concise and limited to a single line. This measure ensures that the model receives only the most relevant information, minimizing the risk of flooding the context window, thus making the overall system more resilient, even in situations where large volumes of data might otherwise hinder its ability to complete tasks effectively.

Predictive tools integration: One important next step of this work aims to incorporate tools that leverage predictive models to predict potential blackouts or assess resilience over the coming weeks, months, or years. These predictions will either be available as historical data in the database or made available to the agent as another tool via function calling or code generation. Either way, it would improve the system's ability to answer complex queries that require insights into future scenarios.

Acknowledgements

This work was conducted within the framework of a collaborative initiative supported by the Federal Department of Defense, Civil Protection, and Sport (DDPS), under the Federal Office for Civil Protection (FOCP). The project involved contributions from Dr. Roland Bollin (Dr. sc. nat.), as well as collaboration with Cédric Vuilleumier (ASTRA) and Francesco Termine (HE-Arc), along with additional acknowledgment to Viteos and the STEP Neuchâtel for their valuable contributions.

References

[1] "OpenGIS4ET – Open Geographic Information System for Energy Transition." Swiss Federal Office of Energy (SFOE), 2021.

[2] "Disasters and Emergencies in Switzerland 2020, National risk analysis methodology," Federal Office for Civil Protection (FOCP), Bern, Dec. 2020.

[3] Y. Wang, A. O. Rousis, and G. Strbac, "On microgrids and resilience: A comprehensive review on modeling and operational strategies," *Renew. Sustain. Energy Rev.*, vol. 134, p. 110313, Dec. 2020, doi: 10.1016/j.rser.2020.110313.

[4] S. Jin and S. Abhyankar, "ChatGrid: Power Grid Visualization Empowered by a Large Language Model," in *2024 IEEE Workshop on Energy Data Visualization (EnergyVis)*, Oct. 2024, pp. 12–17. doi: 10.1109/EnergyVis63885.2024.00007.

[5] 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.

[6] J. Liu, *LlamaIndex*. (Nov. 2022). Python. Accessed: Jan. 20, 2025. [Online]. Available: https://github.com/jerryjliu/llama_index

[7] S. Yao *et al.*, "ReAct: Synergizing Reasoning and Acting in Language Models," presented at the The 11th International Conference on Learning Representations (ICLR), Sep. 2022. Accessed: Jan. 20, 2025. [Online]. Available: https://arxiv.org/abs/2210.03629

[8] *Chroma*. (Jan. 20, 2025). Rust. Chroma. Accessed: Jan. 20, 2025. [Online]. Available: https://github.com/chroma-core/chroma

[9] A. Dubey *et al.*, "The Llama 3 Herd of Models," Jul. 31, 2024, *arXiv*: arXiv:2407.21783. doi: 10.48550/arXiv.2407.21783.

[10] A. Q. Jiang *et al.*, "Mixtral of Experts," Jan. 08, 2024, *arXiv*: arXiv:2401.04088. doi: 10.48550/arXiv.2401.04088.

[11] *Ollama*. (Jan. 20, 2025). Go. Ollama. Accessed: Jan. 20, 2025. [Online]. Available: https://github.com/ollama/ollama

[12] H. Zhang *et al.*, "A Careful Examination of Large Language Model Performance on Grade School Arithmetic," presented at the The 38th Conference on Neural Information Processing Systems, Vancouver, Canada, Nov. 2024.