



# GridGPT: AI Virtual Assistants for Smart Grid Applications

**sddec24–20:** Tin Ngo, Eddy Andrade, Jackson Phillips,  
Emma Heithoff & Nick Doty

**Faculty Advisor/ client:** Dr. Ravikumar Gelli

# Project Overview

- Simplifies power grid management through AI-based solutions.
- Streamlines the complex and ever evolving power system data.
- Translates DSS scripts into natural language for ease of understanding by power plant employees.



# Distribution System Simulation (.dss) files

## **GridGPT's more efficient usage of .dss files:**

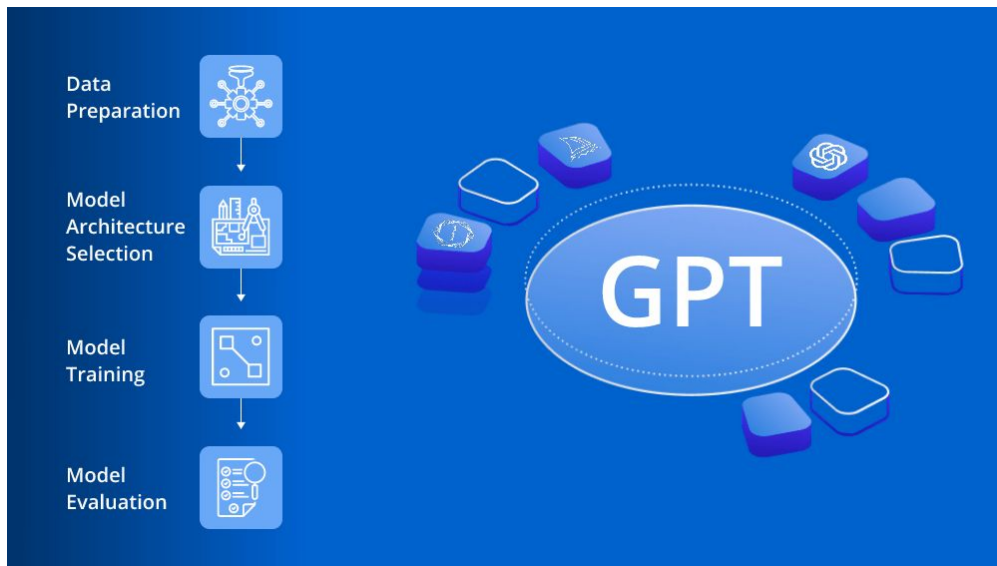
First, the grid operator uses the chat to input natural language instructions.

Second, the model will create .dss files (with Python) to run the simulation for the specified power event.

Lastly, the produced dss files for simulation results will be shown to the user in natural language and visual grid updates will be displayed in interface.

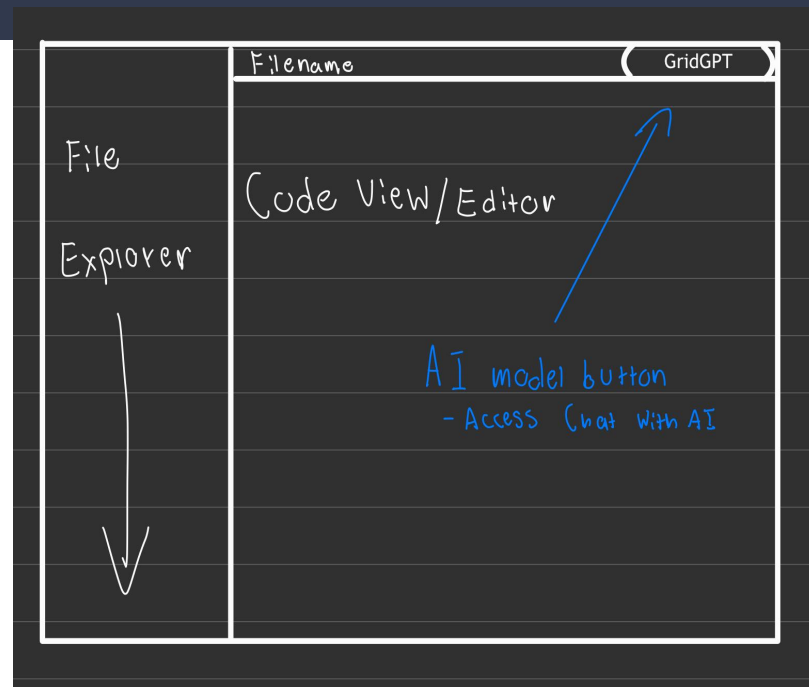
# GPT Model Training

- Our GPT model will be trained on a dataset of DSS python scripts so it understands grid models and can solve problems for users.
- Once we generate the dataset containing the DSS scripts we load it using the training library from [huggingface.co](https://huggingface.co) and call the trainer to train the given model on our dataset.



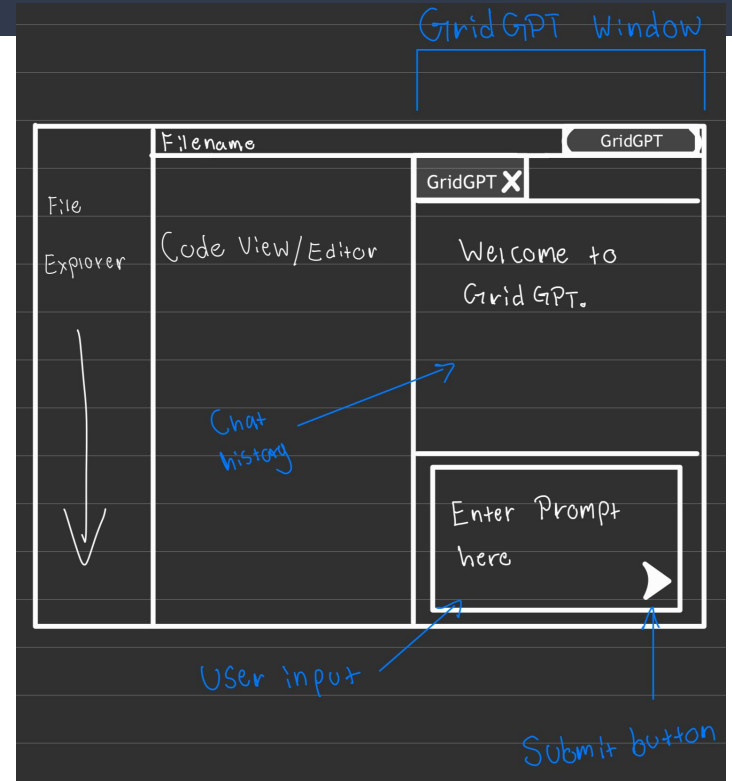
# User Interface Design

- The User Interface will be following similar AI implementation UIs such as:
  - Google Colab
  - GitHub's Copilot
- Our UI will be made using both React and the Next.js framework.



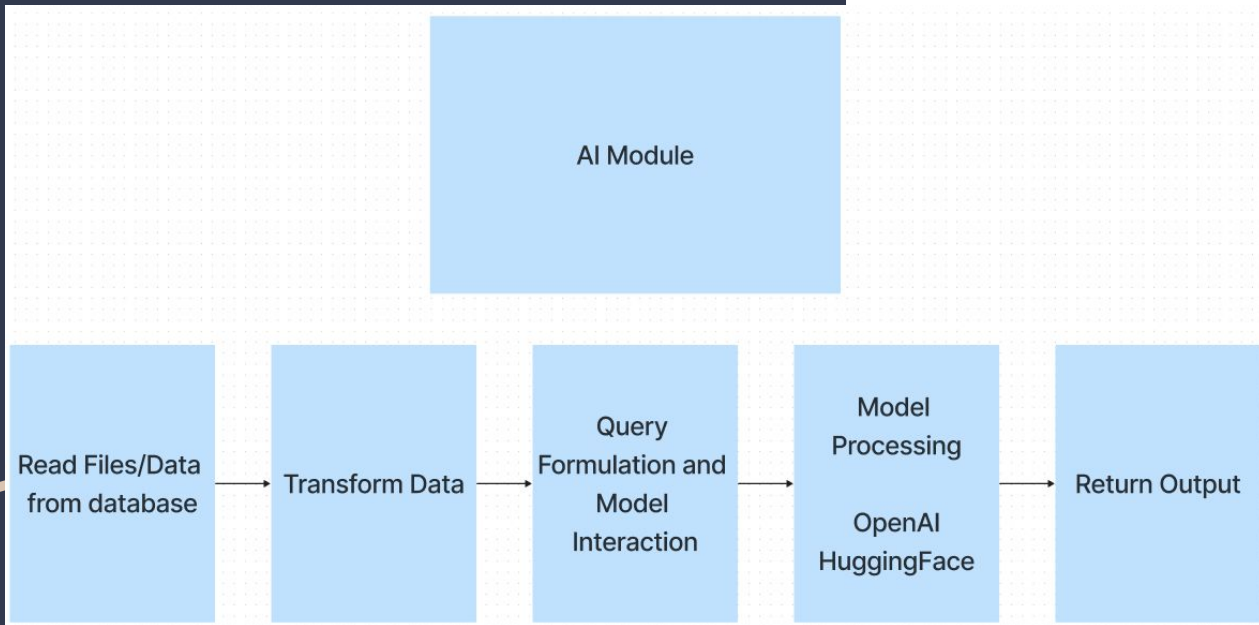
# UI Design Continued

- GridGPT access button located on the top left.
- Once clicked, the GridGPT window will pop up on the right side.
- User input is taken from the UI, then sent to the Backend to begin running the model.
- Output then comes from the Backend and is added into the “Chat History” window.



# Functionality – AI Module

- Read data given by Backend
- Transform Data into readable input for AI Model
- Query Formulation and Instructions
- Model Processing
- Return Output



# Technology Consideration

	OpenAI - GPT 3.5 - 175B Parameters	Hugging Face - Bloom AI model - 176B Parameters
Token Limit	16,385 Tokens / 12,289 words	2,048 Tokens / 1,536 words
Cost	Input: \$.50 cents / 1M Tokens Output: \$1.50 / 1M Tokens	\$32 / Hour on 8 Nvidia A100 with 40 GB RAM each (AWS)
Ease to Use	No set up required	Self-Manage Infrastructure
Scalability, Reliability, and Maintainability	Frequent updates, enhancements, and is well supported.	Self-Manage Infrastructure
Accuracy	19th place out of 78 different models place in Chatbot Arena Leaderboard	Not Ranked
Data Privacy	Depends on OpenAI	Control your Data



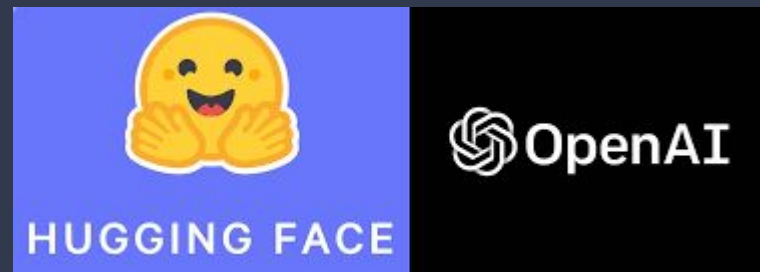
# Chatbot Arena Leaderboard

benchmarks. Empty cells mean not available.

 Model	 Arena Elo	 MT-bench
Claude-3-Sonnet	1200	
GPT-4-0314	1185	8.96
Claude-3-Haiku	1177	
GPT-4-0613	1160	9.18
Mistral-Large-2402	1157	
Qwen1.5-72B-Chat	1149	8.61
Claude-1	1146	7.9
Command-R	1146	
Mistral-Medium	1146	8.61
Claude-2.0	1127	8.06
Gemini-Pro-(Dev-API)	1127	
Mistral-Next	1123	
Starling-LM-7B-beta	1118	8.12
Claude-2.1	1116	8.18
GPT-3.5-Turbo-0613	1115	8.39
Mistral-8x7b-Instruct-v0.1	1114	8.3

# Area of Concern and Development

- Our Client wants to keep data private. OpenAI claims that the data is secure.
  - Hugging Face will keep the data local.
- Training a model with our given computing resources.
  - We will have access to High-Performance Computing (HPCs) devices to train our models.
- Will our Hugging Face be as good as OpenAI?



# Conclusion

Our solution will simplify power management through the recent developments in AI and improve efficiency for grid operators analyzing grid data.

Any Questions?

