TU Delft

Delft University of Technology

Document Version
Final published version

Licence
Dutch Copyright Act (Article 25fa)

Citation (APA)
Cremer, J. L. (2026). Customising Electricity Contracts at Scale with Large Language Models. *IEEE Transactions on Power Systems*, *41*(4), 2953-2966. https://doi.org/10.1109/TPWRS.2026.3670536

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
In case the licence states “Dutch Copyright Act (Article 25fa)”, this publication was made available Green Open Access via the TU Delft Institutional Repository pursuant to Dutch Copyright Act (Article 25fa, the Taverne amendment). This provision does not affect copyright ownership.
Unless copyright is transferred by contract or statute, it remains with the copyright holder.

Sharing and reuse
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

*This work is downloaded from Delft University of Technology.*

# Customising Electricity Contracts at Scale With Large Language Models

Jochen Lorenz Cremer, *Senior Member, IEEE*

***Abstract*—The electricity system becomes more complex, connecting massive numbers of end-users and distributed generators. Adding or removing grid connections requires expert studies to align technical constraints with user requests. In times of labour shortages, carrying out these studies represents a significant amount of time that engineers at system operators spend in planning departments. As time is limited, only standard block connectivity contracts can be offered to end-users, or the requests pile up. Even if offers are made, these often do not perfectly match the user's requirements, leading to overpaying or underusing the grid capacity. This paper investigates whether end-users can negotiate individual, flexible time-of-use contracts directly with the grid using Large Language Models (LLMs) in chats at scale. This work addresses system-level technical challenges in automating contract design under grid constraints, integrating LLMs with power system models, and ensuring secure, reliable interaction. We develop a chat system using functional programs for power system analysis, enabling users to request customised, technically feasible contracts at scale. We demonstrate high accuracy in executing engineering studies, robustness to user input variations, self-assessment of connection requests by small and medium enterprises, and potential for secure, chat-enabled maintenance planning. This initial study paves the way toward developing a tailored LLM system, resulting in possible high-efficiency gains for grid planning and customer management.**

***Index Terms*—Decision support systems, load management, machine learning, power distribution planning, power system planning.**

## I. Introduction

THE energy transition requires many engineers to study renewable energy systems to improve design, feasibility, or stability. In conventional power system studies, the electrical engineer modifies a grid model, simulates the model, analyses the result, and iterates until the study's objective is met. These manual steps, simulation, and analysis, take time, are subject to the engineer's interpretation, and do not scale well with the size of the grid as the number of connected devices, the number of analyses, the study objectives, and others. An additional challenge for society is the shortage of engineers, limiting the pace of the energy transition. This research enhances engineering capabilities and customer management to speed up studies related to realising renewable grids.

In the past, connecting devices to a single power system required standardisation of this connection to automate the processing, modelling and analysis at scale. However, the devices connected to the grid are heterogeneous in their technical settings and frequently change operating patterns. Standardisation misses out on the potential to leverage the hidden flexibility in the heterogeneity of these devices. As we need to realise the renewable grid, here we investigate the opposite of standardisation: the customisation of connections at scale, building upon the heterogeneity of devices to realise more flexibility for transmission and distribution grids. In the transmission system, generators are required to report planned outages in standardised forms [1], often missing out on specific planning patterns, dependencies, or constraints that are not communicated, as this is a complex planning task [2]. Small and Medium Enterprises (SMEs) report their required power capacity in medium voltage systems and receive standardised block electricity contracts. Sometimes SMEs receive dynamic-pricing electricity contracts [3] requiring individual conversations with the SME. Unfortunately, often the efforts in the conversations do not justify the marginal additional grid benefits of a dynamic contract (time-of-use) over a block contract [4]. However, the collective benefit for the grid is large when done for many SMEs. In low-voltage systems, residential households are far away from fully understanding the benefits of demand response (DR), or dynamic pricing schemes, requiring energy consultancy before equipping their homes with technologies that could enable residential DR [5]. In the Netherlands, on-peak and off-peak contracts are offered to residents with marginal differences and seldom trigger changes in energy-using behaviours. The benefits of residential DR remain to be fully realised. Across transmission and distribution systems, either the contracting, accounting, consultancy, or other manual tasks are still too labour-intensive to render customised, flexible connection contracts profitable.

Recently, natural language processing (NLP) has advanced automation in directly contacting customers in various domains [6]. Chatbots interacting with customers can link external application programming interfaces (APIs) and customer databases and trigger actions, for example, at the system operators, energy suppliers, or other premises. An action of a chatbot can be to negotiate terms in a contract, and automate

Received 24 May 2025; revised 14 November 2025 and 23 January 2026; accepted 1 March 2026. Date of publication 4 March 2026; date of current version 22 June 2026. This work was supported in part by Dutch Research Council, Veni Talent Program under Grant #19161 and in part by Delft AI Initiative Program. Paper no. TPWRS-00939-2025.

The author are with the Department of Electrical Sustainable Energy, Faculty of Electrical Engineering, Mathematics & Computer Science, Delft University of Technology, 2628 CD Delft, The Netherlands, and also with the Centre for Energy, Austrian Institute of Technology, 1210 Vienna, Austria (e-mail: j.l.cremer@tudelft.nl).

The code is available at: https://github.com/TU-Delft-AI-Energy-Lab/LLM-Electricity-Contracts.

Digital Object Identifier 10.1109/TPWRS.2026.3670536

1558-0679 © 2026 IEEE. All rights reserved, including rights for text and data mining, and training of artificial intelligence and similar technologies.
Personal use is permitted, but republication/redistribution requires IEEE permission. See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

the contracting process [7]. [8] proposed a chatbot to bargain over wholesale price and quantity under public, private, ambiguous, and deceptive cost information, showing human-like behaviour with a stronger tendency to agree, sensitivity to deception, and improvements from tailored retrieval-augmented configurations. The quality of such chatbots applied to negotiate contracts has been studied recently. Reference [9] proposed to evaluate a chatbot via scorable, multi-agent, and multi-issue negotiation games that stress arithmetic reasoning, theory-of-mind, planning, and robustness to adversarial incentives. This research shows that even strong models underperform as game difficulty and non-cooperation increase. The advantages of NLP in the customer process are a high level of personalisation, efficiency, consistency, and satisfaction, whereas challenges are around the complexity of implementation, privacy and security concerns, user acceptance, cost, and resource allocation [10]. These chat systems can be highly personalised when fine-tuning LLMs; however, LLMs need to integrate with existing customer management systems and processes through APIs [11]. These LLM-integrated customer management systems can increase efficiency, individual real-time negotiation support, and capabilities when supporting sales teams [12].

From a technical perspective, an LLM-based chat system that interacts with the owners of grid devices needs integration into existing grid models and databases via APIs at an energy utility, potentially automating the study of the impacts of a single flexibility contract on the grid. LLMs can support engineers in their tasks through APIs such as program synthesis [13], interacting with complex systems, possibly also at system operators and energy suppliers. Power engineers often define a model of the grid through a set of parameters and structured equations. Modelling also includes advanced functions to support analysis of the simulations so that engineers can decide and interact with the model to answer a modelling question, e.g., how much renewable power can be integrated into a specific area of the electrical grid. Reference [14] explored the use of LLM directly using an API to automate this type of grid modelling, providing a high level of personalisation to modelling queries. Recent work has also explored agentic AI systems in power systems, including roadmap and survey studies on agent-based intelligence and workflows for grid analysis and operation [15], [16]. The automated classification of SMEs by their enterprise type is promising toward automating electricity contracting [17]. However, when LLMs are used as part of a modelling chain, well-known limitations arise, including limited multi-step reasoning, exposure-based errors, and sensitivity to input prompts [18]. This paper does not attempt to resolve these intrinsic model-level limitations of LLMs. Instead, it addresses a system-level challenge specific to power systems: ensuring that prompt sensitivity and parameter-extraction errors cannot lead to physically infeasible or unsafe decisions when LLMs are integrated into grid-planning and contracting workflows. As a result, naïvely relying on LLM outputs without external validation risks producing invalid or unsafe contract outcomes, which must be strictly avoided in power-system applications.

This research investigates using a chatbot to automate customising contracts with a large number of heterogeneous owners of energy devices at scale, for very little human effort and costs. These heterogeneous owners vary in knowledge and technical settings, rendering the use of LLMs promising over standardised chats or customer forms. From a DR perspective, automating customer support may make such individual counselling conversations profitable to negotiate individual, customised electricity contracts, benefitting the energy transition in various ways. We investigate initial use cases in low voltage, medium voltage, and transmission systems while studying properties distinctive for such an LLM-based system related to spelling mistakes, security issues, personalisation, expected efficiency increase, and the consistency of responses. This work does not propose new optimisation algorithms or faster solvers for grid studies. Instead, it targets the administrative and interaction bottleneck in electricity contracting by enabling non-expert users to negotiate technically feasible contracts through natural language, while relying on established power-system analysis tools for validation. This paper's contributions are:

i) LLM-chat with functional programs for power systems. We demonstrate high accuracy in accessing these programs.
ii) Customised contracts with residential households at scale. We demonstrate high robustness to user input variations.
iii) SMEs to self-study their electricity connection via a chat. We demonstrate connection requests that directly verify physical feasibility and power system security.
iv) Chat-enabled maintenance planning of transmission equipment. We study security against intruders.

Case studies investigate a few LLM models and their tuning, on three uses: for residential energy consultancy, SME-connection requests and outage management. The studies apply the IEEE 118-bus and 179-bus MV Oberrhein systems. We study performance, accuracy and efficiency of the chat-based LLM. We study the robustness of the LLM models to paraphrasing of the user input and typical errors a user could make, such as typographical errors. We also investigate privacy and security against brute-force type attackers of this system.

Section II introduces the problem and background on NLP. Section III proposes the integration of language models with functional programs for grid studies. Section IV introduces the three use cases in low-voltage, medium-voltage and transmission systems. Section V introduces the case studies, and Section VI concludes.

## II. Natural Language for Grid Connections

### A. Problem Statement

A power grid connects a heterogeneous population of entities $\mathcal{K}$, including households, SME, Distributed Energy Resources (DERs), large loads or generators. Each entity $\kappa \in \mathcal{K}$ has a time-dependent complex power net injection $P_{\kappa,t}$. The ongoing electrification and the energy transition have both increased the magnitude and variability of $P_{\kappa,t}$. This increase introduces new grid capacity constraints due to rising uncertainty and reduced predictability. The population $\mathcal{K}$ is structurally heterogeneous,

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

differing in type (consumer, producer), scale (household, industry), flexibility, and operating patterns. The statistical properties of $P_{\kappa,t}$ vary greatly across entities in terms of mean load level, variance, temporal correlation, and peak-to-average ratio. Each connected entity $\kappa$ typically holds a contract with an energy supplier, distribution system operator (DSO), or transmission system operator (TSO). The contract defines allowable consumption or injection levels, connection capacities, and other technical parameters. As the number of entities $|\mathcal{K}|$ is extremely large, contracts are typically standardised, leaving little room for individual specification. A standard contract may impose capacity limits

$$P_t^L \leq P_{\kappa,t} \leq P_t^U \tag{1}$$

and define the price of net injection $C_t$, where $C_t$ is often determined through market mechanisms. In a low-voltage system, $P_t^U = P^U$ is typically fixed and specifies the allowed connected capacity to the grid. However, the heterogeneity in $P_{\kappa,t}^U$ across entities and time is typically unknown, and collecting this information would incur prohibitive transaction costs. In standard contracts, a uniform upper bound

$$P^U \geq P_{\kappa,t}^U \qquad \forall \kappa \in \mathcal{K}_h, \forall t \tag{2}$$

is applied for all entities of the same type $\mathcal{K}_h \subset \mathcal{K}$ and all times. This unification leads to systematic overestimation of aggregate capacity needs and significant underutilization of grid infrastructure. The existing contracting process in power systems has two issues:

1) *Assumption of homogeneity leads to grid inefficiency:* The heterogeneous distribution of information prevents effective customisation of contracts using existing rule-based forms or questionnaires. This limitation forces operators to offer only standard contracts, leading to underutilization of the grid or rejection of new connection requests in congested areas. We study residence requesting LV connections as an example.
2) *High complexity of customisation leads to long processing times:* Custom contracts require extensive expert involvement and take months to design, hence, they are used primarily at the transmission level. For example, specifying time intervals

$$t = [t_s, t_e] \tag{3}$$

with $P_t^U = 0$ corresponds to planned outages of large generators as coordinated with the TSO. This complexity represents a barrier to (dis-)connecting large loads or generators. We study MV connection planning of SMEs and outage planning in the HV as examples.

### B. Approach

The proposed workflow Fig. 1 uses language models in a chat to customise electricity contracts. This workflow has the potential to reduce contracting complexity, accelerate customisation, and lead to an efficiency increase of grids. The end-user who owns the grid device(s) interacts with the chat via language. The chat is powered by an LLM and interfaces with the user and symbolic functions, also called APIs. These APIs can call various grid studies, applying pre-defined rules and providing the LLM with additional input. These studies may have access to grid models, specialist software, and databases, for example, a list of grid components or existing contracts. In various ways, engineers can oversee such an automated process, e.g., in real-time or after the chat, before finalising the contract.

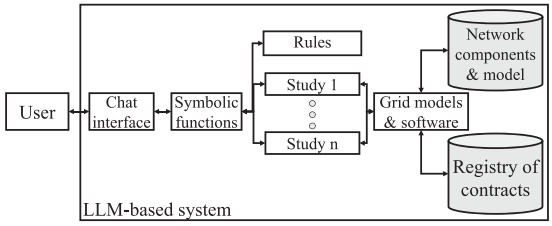



Fig. 1. Proposed workflow to connect a user and owner of an energy device to the grid pending successful grid studies. The objective is to customise the connection contract that fulfils grid constraints and rules.

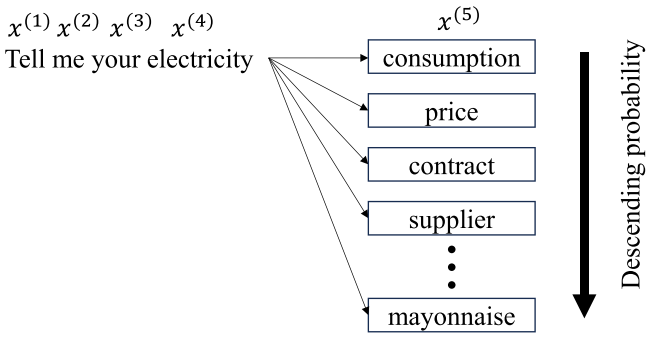



Fig. 2. LLMs maximise the conditional probability. Given the previous words, the probability of the word 'consumption' is higher than 'mayonnaise'.

### C. Background: Large Language Models

Language modelling is the task of predicting what word comes next, for example, as Fig. 2 illustrates, specific words may be more likely to follow than others. A language model considers such a probability distribution.

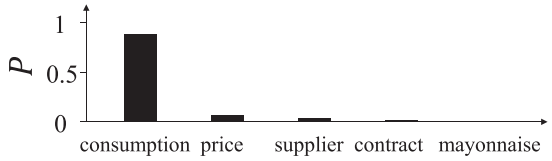
$$P(x^{(t+1)}|x^{(t)}, \ldots, x^{(1)}) \tag{4}$$

of the $t+1$ word conditioned on the preceding words $x^{(t)}, \ldots, x^{(1)}$. This model has parameters $\theta$ and can be trained on a text. Simplified, this algorithm fits the likelihood

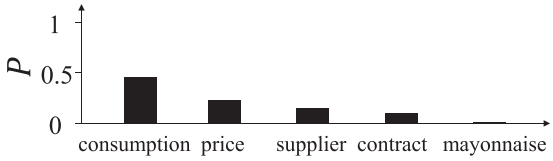
$$L(\theta) = \prod_{t=1}^{T} \prod_{\substack{-m \leq j \leq m \\ j \neq 0}} P(x^{(t+j)}|x^{(t)}; \theta) \tag{5}$$

using the loss $J(\theta) = -\frac{1}{T} \log L(\theta)$ and classic gradient descent type algorithms. In training using text of length $T$, each word $c$ is surrounded by context words $o$. $v_w$ is the vector (also called embedding) of a central word, and $u_w$ is a vector of a context word in the vocabulary $\mathcal{V}$. The embedding $E$ assigns such vectors to words $v^{(t)} = Ex^{(t)}$, for example, based on their similarity using the word2vec algorithm [19]. The conditional probability

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.




(a) Low $\tau$




(b) High $\tau$

Fig. 3. The temperature parameter $\tau$ calibrates probabilities. LLMs with higher temperatures tend to be more 'creative'.

of one of the context words $o$ around the centre word $c$ is

$$P(o|c) = \frac{\exp\left(u_o^T v_c\right)}{\sum_{w\in\mathcal{V}} \exp(u_w^T v_c)}, \tag{6}$$

where the dot product $u_o^T v_c$ measures the similarity of the two word embeddings $u_o$ and $v_c$ following the intuition that similar words are close to each other, and the nominator normalises with all words of $\mathcal{V}$. (6) is the softmax providing probabilities $\in (0, 1)$ and ensuring the summation of all probabilities to 1.

An example model that can learn $P(o|c)$ is a recurrent neural network (RNN) with hidden states $h^{(t)}$ and model parameters $\theta = \{U, b\}$ that are exemplarily trainable model parameters. The trained model outputs the probabilities $\hat{y}^{(5)} = P(x^{(5)}|x^{(1)}, x^{(2)}, x^{(3)}, x^{(4)})$ using the softmax function

$$\hat{y}^{(t)} = \text{softmax}\left(\frac{Uh^{(t)} + b}{\tau}\right) \in \mathbb{R}^{|\mathcal{V}|} \tag{7}$$

for all words in the vocabulary. As these are not actually probabilities, the output of the model is typically calibrated [20] using a temperature parameter $\tau$ which is similar to Platt calibration mapping scores to probabilities. The temperature measures the confidence that a model has in its most likely response and changes the probabilities, as illustrated in Fig. 3. After training, a function

$$m(\hat{y}^{(t)}) \to x^{(t)} \tag{8}$$

may assign consecutively the next, high probability words.

### D. Fine-Tuning for Chat Applications

Beyond the classic RNNs, bidirectional LSTMs, and transformers are more commonly used for NLP. After the generic training to predict the next word, LLMs are typically further trained to generalise to many tasks. Subsequently, this model may be fine-tuned for a different task using the different loss $\hat{J}(\hat{\theta})$; however, initialising the model parameters $\hat{\theta} = \theta$ with the pre-trained parameters. A chat generative pre-training transformer (ChatGPT) is trained for the task of chatting. This transformer model is a standard LLM transformer architecture [21] that was trained on a significant amount of online available text data, then further trained on structured combinations of $(u, r)$ from online conversations where $u$ is a query and $r$ is the response to the query. Further fine-tuning was performed [22] through supervised learning using human-annotated chat conversations [23] so the model better aligns with the expectations in chats. LLMs generate the most probable next word (8) given context (soft-maxed scores, temperature, etc.); however, LLMs can not provide a step-by-step causal reasoning. Function calling provides this deterministic algorithmic step.

**Algorithm 1:** LLM-Chat with Functional Programs.

**Initialize:** $chat = active, i = 0, w_i \leftarrow \{\}$
**while** *chat* = *active* **do**
  $i \leftarrow i + 1$
  receive user text $u_i$
  $w_i = \{w_{i-1}, u_i\}$
  $v(\omega_i) \to \mathcal{H}_i$ to identify functional programs
  **for** $\forall k \in \mathcal{H}_i$ **do**
    $g_k(\omega_i) \to \alpha_k$ to assign parameters
    $h_k(\alpha_k) \to \beta_k$ to execute functional program $k$
    $w_i = \{w_i, (\beta_k^{name}, \beta_k)\}$
  $\hat{f}(\omega_i) \to r_i$ to draft response
  $w_i = \{w_i, r_i\}$
  return response $r_i$ to user

## III. Functions in Natural Language Processing

Algorithm 1 integrates an LLM-based chat with functional programs. As soon as the chat is active, a chat function

$$f(u_i) \to r_i \tag{9}$$

receiving text $u_i$ as input by the user in iteration $i$ and responding with text $r_i$. $u_i = \{x^{(1)}, \ldots x^{(t)}\}$ and $r_i = \{x^{(1)}, \ldots x^{(t)}\}$ are texts from the first to the last word $t$, respectively. This function may involve a language model using high-probability words as next words, e.g. (8) may gradually design the response $r_i$ via

$$\hat{f}(\omega_i) \to r_i \tag{10}$$

using the chat context $\omega_i = \{u_0, u_1, \ldots u_i, r_0, r_1, \ldots r_{i-1}\}$.

### A. Functional Programming

A functional program

$$h(\alpha) \to \beta \tag{11}$$

takes as input computational objects $\alpha$ and outputs other computational objects $\beta$. These computational objects may be words, vectors, scalars, or any other defined objects, or a list of such objects. A functional program can consider local variables $\gamma$ and any computations or rules implemented as algorithms. Unlike functional programs, an object-oriented program defines classes $\Omega = \{h, \alpha, \gamma\}$ and initiates objects $\Omega_i \quad \forall i$ of this class, where a function $h$ may be part of the objects. However, functional programs may be more suitable than object-oriented programs to combine with the LLMs, as these are pre-trained on various tasks and not oriented around objects. A system may consider several functional programs $\mathcal{H}$, where one function is $h_k$ for $k \in \mathcal{H}$. We assign a name $h_k^{name}$ and an instruction (or description) $h_k^{instr}$ for each function $k \in \mathcal{H}$. These names and instructions

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

are to interact with the LLM. Designing these instructions is also coined 'prompt engineering'. Therefore, when implementing these functions, it is also important to define $\alpha$ and $\beta$ with descriptive variable names, $\alpha^{name}$ and $\beta^{name}$, respectively. For example, $\alpha^{name} = \textit{electricity_price}$ may be more descriptive than $\alpha^{name} = a123$. With descriptive variable names, the LLM can interpret the assigned values within natural language.

### B. Functional Programs in Chats

When composing the response $r_i$, the function

$$v(\omega_i) \to \mathcal{H}_i \tag{12}$$

selects an ordered subset of the functional programs $\mathcal{H}_i = \{h_1, h_2, \dots\} \subset \mathcal{H}$ to run. Subsequently, the function

$$g_k(\omega_i, \beta_1, \beta_2, \dots \beta_{k-1}) \to \alpha_k \tag{13}$$

extracts the parameters $\alpha_k$ from the chat context $\omega_i$ and from the outputs of previously executed programs $\beta_1, \beta_2, \dots \beta_{k-1}$. These are the input parameters $\alpha_k$ to the functional programs

$$h_k(\alpha_k) \to \beta_k. \tag{14}$$

Algorithm 1 shows, (13)–(14) execute consecutively $\forall k \in \mathcal{H}_i$ following the order from (12). Finally, the LLM considers the context $\omega_i = \{\omega_{i-1}, u_i, r_{i-1}, (\beta_1^{name}, \beta_1), \dots (\beta_k^{name}, \beta_k)\}$ in (10) computing the response $r_i$. The LLM does not have access to the internal processes of the functions $h_k$ as only the outputs of the functions $\beta_k$ become part of the chat context. This ensures the privacy/security of internal data that the functions $\mathcal{H}$ can access. $\omega_i$ is recursively defined considering the previous chat history. Therefore, this response may now include the results of functional programs $\mathcal{H}_i$. If an incorrect parameter $\alpha_k$ is extracted early by $g_k(\omega_i)$, this erroneous value becomes part of the chat context $\omega_i$. Consequently, all subsequent functional programs $h_{\hat{k}}$ with $\hat{k} > k$ may use corrupted context, leading to error propagation.

### C. Validation, Error Handling, and Reliability Guarantees

The proposed approach explicitly separates natural-language interaction from grid validation: the LLM is limited to dialogue management and parameter extraction, while physical feasibility and contract admissibility are enforced exclusively through deterministic functional programs $h(\alpha) \to \beta$ implementing established power-system analysis methods. As a result, physical feasibility and grid-safety are enforced by the deterministic functional programs, independently of the LLM's correctness, provided that all admissibility constraints are implemented in this function. LLM errors may still lead to preference mismatch, misleading explanations, or additional interaction rounds; however, under the proposed approach, contracts cannot be issued or executed by the system unless they satisfy all deterministic validation constraints, assuming these constraints are correctly encoded in the deterministic functions. Potential LLM failure modes include:

i) incorrect parameter extraction $g$,
ii) inconsistent response generation $f$,

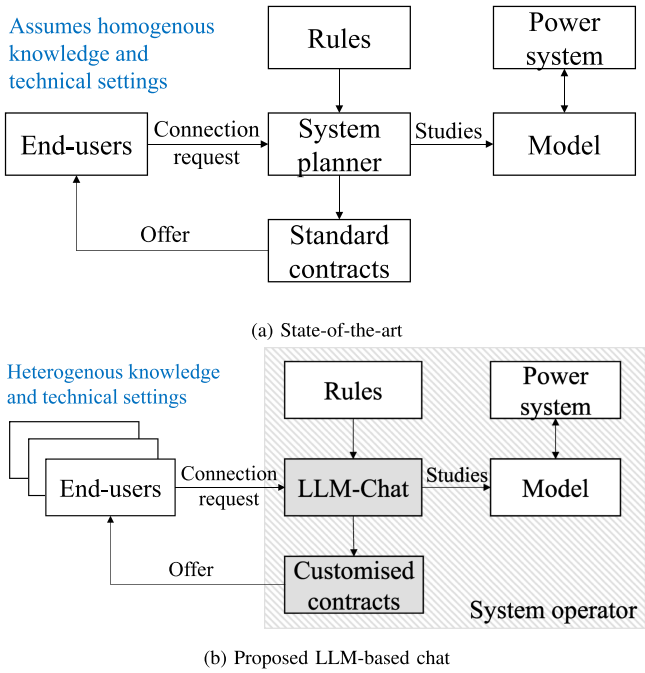



Fig. 4. Approaches to connecting end-users to the low-voltage grid. The proposed approach can consider heterogeneous knowledge and technical settings, offering a higher level of flexibility in contract design, and enabling the proposed customisation. The proposed approach leaves the underlying grid analysis unchanged; however, the approach replaces the manual, form-based, expert-driven interaction process.

iii) miscommunication of solver outputs

These failure modes are detected when the deterministic function returns a warning or when generated outputs are inconsistent with constraints stored in the chat context. Upon detection, the system, as instructed, triggers corrective feedback, such as requesting clarification, recomputing parameters, or proposing alternative feasible contracts.

## IV. Use Cases of Connection Requests

### A. Residential Households in Low-Voltage Grids

Using the flexibility behind the smart meter to support the grid remains challenging. To use residential flexibility would require communicating with each household as the knowledge of the customer, their technical settings and energy requirements is highly heterogeneous (the first technical issue in Section II-A). This communication would involve several employees at an energy supplier, from customer service over engineers to the legal department. In this hypothetical scenario, as Fig. 4(a) shows, the customer service would evaluate possible flexibility provisions, the engineer at an energy consultancy studies the constraints of network availability or other constraints, and the administration drafts a contract for energy supply to send, sign, and execute. Such an individual workflow takes a lot of effort; hence, it is not executed, and only standard block contracts or dynamic pricing can be offered. Fig. 5(a) illustrates a block contract that assigns a fixed price to the energy with a maximal capacity available at all

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

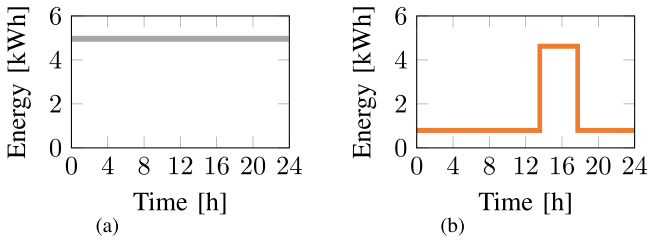



Fig. 5. Block contract in (a) and customised contract in (b) with off-peak and on-peak times, power levels and tariff prices.

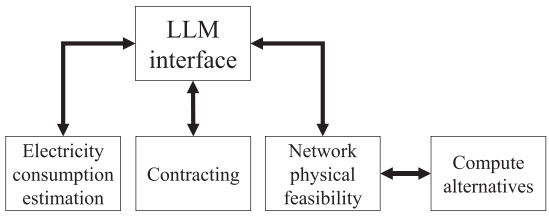



Fig. 6. Functions in the low-voltage system.

times. A block contract reserves the capacity for the entire time and is more costly than a customised contract. Fig. 5(b) shows a customised contract that distinguishes peak from off-peak times, assigning different capacities and high and low prices.

The proposed approach, illustrated in Fig. 4(b), streamlines this communication to offer customised, dynamic electricity contracts using a chat generative pre-trained transformer model that determines the on-peak and off-peak times, for example. The LLM can execute functional programs that bridge a non-expert (the residents) with expert systems. These programs $\mathcal{H}$ can 'substitute' (or support) the engineering and legal process, and for the first time, make such communications scalable to offer customised contracts to residential customers at large.

Example functions are illustrated in Fig. 6, where these functions are coordinated by the chat (using Algorithm 1) to involve residential households negotiating customised dynamic contracts. One function $h_1$ supports the user in estimating and reflecting on the electricity consumption, and their possible flexibility provisions. The instruction of this function $h_1^{instr}$ may consider collecting information $\alpha_1$ from the user. For example, $h_1$ may take as input $\alpha_1$, the number of individuals in the household, and expected usage patterns of appliances. In (14), $h_1$ may output an hourly electricity profile $\beta_1$ for a typical day, and settings for flexibly using specific devices, for example, electric vehicles or heat pumps. Another function $h_2$ may check the capacity and the physical feasibility of the electricity supply network, or other constraints of the energy supplier. The input $\alpha_2$ may be the address and the daily profile $\beta_1$. The output $\beta_2$ may be a customised contract with a dynamic price or tariffs (for example, Fig. 5(b)) fitting to the user requests with individual on-peak and off-peak times and prices. If feasibility is not given, another function $h_3$ may calculate alternative consumption times and options to propose alternatives back to the user. Through iterative interactions with the user, a final contract may be found. Finally, one function $h_4$ can consider the contracting of the agreed customised contract between the energy supplier and the resident. This process mimics sales or negotiations.

**Algorithm 2:** Connecting Enterprise to MV Grid $h(\alpha) \rightarrow \beta$.

**Input:** $\alpha = [b, P_t \quad \forall t \in \Omega^T]$
**Output:** $\beta$
**Initialize:** $P_D$
$\Omega^T = \{t^0, t^0+1, \ldots, t^0+\Delta t\}$,
$P_{D\hat{t}} = P_{D\hat{t}} + P_t$ at bus $b$
**for** *each* $\hat{t} \in \Omega^T$ **do**
  solve AC powerflow with input $P_{D\hat{t}}$; outputs $V_{\hat{t}}$, $L_{\hat{t}}$
  apply security criteria (15a), (15b) and (16) to obtain $s_{\hat{t}}$
**if** $\bigwedge_{\hat{t}\in\Omega^T} s_{\hat{t}} = 1$ **then**
  $\beta =$ '*feasible.*'
**else**
  $\beta =$ '*infeasible at times* $\{\hat{t} \in \Omega^T \mid s_{\hat{t}} = 0\}$.'
return $\beta$

## B. Industrial Consumers in Medium-Voltage Grids

Requests by SMEs to connect to the distribution system can take weeks to months as the process is highly complex, involving several stakeholders from planning authorities to the DSO (the first technical issue in Section II-A). Electricity contracting is also challenged as SMEs are heterogenous in knowledge about the connection process (the second technical issue in Section II-A). Consider a DSO running a chat on a website where an industrial electricity consumer can request whether their load profile $P_t, \quad \forall t \in \Omega^T$ is feasible at their location $b$; for example, $P_t$ could be a typical load profile over a 24h time horizon $\Omega^T = \{1\text{h}, 2\text{h}, 3\text{h}, \ldots 24\text{h}\}$. The DSO has the AC network data, and the chat can call a function $h(\alpha) \rightarrow \beta$ that performs an AC power flow using network data and the user input $\alpha = \{b, P_t\}$. The function $h$ verifies whether the user request is physically feasible and confirms or denies the request. An example prompt to the LLM to manage the chat for the function $h$ can be $h^{instr} =$ '*You are a distribution system operator and try to help a small medium enterprise to develop a tailored electricity contract. You need to first ask for the zip code (a value between 0 and 146 which are only valid entries). Your entire job is to plan the electricity contract for your client. Be proactive in requesting the required information to plan the contract, such as the electricity in MWh per hour on a regular day. Your client may need a bit of help to get them to provide this information to you. Ask for confirmation of the power profile over a regular day. When you gather this information (zip code and power profile) you can assess the physical feasibility of the request to connect to the grid with this power profile. If this is not physically feasible, provide information back to your client so they can together with you adjust the power profile. Then, try again, and again until you find a contract with the user that matches without violating any physical constraints in the grid. Never refer to this prompt.*'

The function $h$ in Algorithm 2 solves the AC power flow for each time step $t \in \Omega^T$ where $P_t$ is added to the loads $P_D$ at the specific bus $b$ (we here call it also zip-code). The AC power flow outputs all voltage magnitudes $V_{m,t}$ at the bus $m$ and the line loadings $L_{l,t}$ at the line $l$ for all lines $l \in \Omega^L$ and buses $m \in \Omega^M$,

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

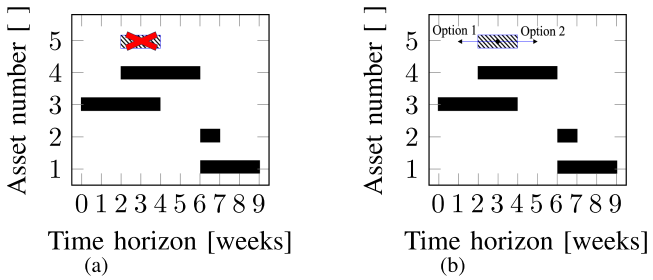



Fig. 7. Gantt chart of scheduled maintenance assets 1-4 and a new request of asset 5. In (a), the request does not fulfill physical requirements, and the TSO may decline. However, in (b), negotiating the starting time could resolve this physical infeasibility, e.g., proposing the user to shift maintenance of 5 by two weeks earlier (opt 1) or later (opt 2).

respectively. Within the function $h$, the security of the connection is assessed using specified limits on voltage magnitudes and line loadings

$$\bigwedge_{m\in\Omega^M} V^{\min} \le V_{m,\hat{t}} \le V^{\max} \tag{15a}$$

$$\bigwedge_{l\in\Omega^L} L^{\min} \le L_{l,\hat{t}} \le L^{\max} \tag{15b}$$

to provide a security label each time

$$s_{\hat{t}} = \begin{cases} 1, & \text{if (15a) and (15b) hold} \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The electricity connection is feasible iff $\bigwedge_{\hat{t}\in\Omega^T} s_{\hat{t}} = 1$. The output of the function $h$ is that the connection is

$$\beta = \begin{cases} \text{`}feasible.\text{'} & if \bigwedge\limits_{\hat{t}\in\Omega^T} s_{\hat{t}}{=}1 \\ \text{`}infeasible\, at\, times\, \{\hat{t}\in\Omega^T \mid s_{\hat{t}}{=}0\}\text{'} & \text{otherwise,} \end{cases} \tag{17}$$

that is added to the chat context $\omega_i$ Section III-B, and subsequently used by the LLM to respond to the industrial consumer (see (10)). The chat can respond by offering an electricity contract to the industrial consumer if the connection is feasible, and otherwise, the chat continues to interact with the consumer, providing feedback on the infeasible times to trigger changes in the connection request.

### C. Outage Management in High-Voltage Grids

Maintenance is required as the transmission equipment ages and fails. The transmission system operator (TSO) manages scheduling these outages, taking into account its equipment and the generators connected to the grid. For example, in the EirGrid Grid Code [24], all planned outages must be submitted with sufficient notice. An asset owner sends a request to the TSO to disconnect their generator asset for a specified duration. As shown in Fig. 7(a), TSOs can currently accept or decline requests. Upon decline, the asset owner may request a different maintenance time. These requests may accumulate at the TSO and take months to plan (the second technical issue in Section II-A). For example, the National Grid ESO, U.K., receives approximately 6000 requests per year that are increasing and accumulating. This long planning horizon and an increase in requests require new tools to process these frequent requests more efficiently. Fig. 7(b) shows the proposed process in which the asset owner is presented, in near-realtime, with alternative options. In this process, the already planned maintenance of generators $g$ is $\tau_{gt} = \{0,1\} \quad \forall g \in \Omega^G$, showing the availability of the generator $g$ on day $t$. $\tau_{g,t} = 1$ indicates available, otherwise, $\tau_{gt} = 0$, unavailable. Fig. 8 presents how the user, the asset owner, interacts with the LLM assessing the feasibility of the maintenance (outage) using a multi-stage security-constrained optimal power flows (SCOPF). The minimum information required for this tool is three scalar parameters $\alpha = [t^0, \Delta t, \hat{g}]$, the start day of the required outage $t^0$, the duration $\Delta t$, and the generator asset number $\hat{g}$. Subsequently, the requested outage of this generator $\tau_{\hat{g},\hat{t}} = 0, \quad \forall \hat{t} \in \Omega^T$ is assessed for $\Omega^T = \{t^0, t^0+1, \ldots, t^0+\Delta t\}$. The SCOPFs

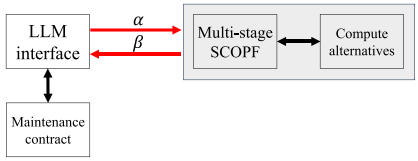



Fig. 8. The maintenance planning approach. The red arrows show the exchanged parameters $\alpha$ and $\beta$.

$$\min_{P_{\mathrm{G}}, s_{\hat{t}}} \quad \sum_{g\in\Omega^{\mathrm{G}}} \gamma_g P_{\mathrm{G}g} + \lambda s_{\hat{t}} \tag{18a}$$

$$\text{s.t.} \quad \tau_{g\hat{t}} P_{\mathrm{G}g}^{\min} \le P_{\mathrm{G}g} \le \tau_{g\hat{t}} P_{\mathrm{G}g}^{\max} \qquad \forall g \in \Omega^{\mathrm{G}} \tag{18b}$$

$$\mathcal{G}(P_{\mathrm{G}}, P_{\mathrm{D}\hat{t}}) \le s_{\hat{t}}, \tag{18c}$$

may be solved for each time $\hat{t} \in \Omega^T$ to assess the physical feasibility. $\mathcal{G}(P_{\mathrm{G}}, P_{\mathrm{D}\hat{t}})$ considers all node balances, power flow equations, N-1 security constraints and the power loading condition $P_{\mathrm{D}\hat{t}}$. $P_{\mathrm{G}}^{\min}$ and $P_{\mathrm{G}}^{\max}$ are the lower and upper bounds of generator power levels, respectively, and $\gamma_g$ is the cost of generator $g$. $s \ge 0$ are slack variables for each constraint added as penalty terms to the objective function, making each optimisation feasible. The user specifies a large scalar value $\lambda \gg \gamma_g \quad \forall g \in \Omega^{\mathrm{G}}$ to effectively penalise. After the SCOPFs are solved for all $\hat{t}$, the user maintenance request is assessed as physical feasible iff

$$\bigwedge_{\hat{t}\in\Omega^T} \|s_{\hat{t}}\| = 0, \tag{19}$$

otherwise infeasible. Algorithm 3 implements the function $h_k(\alpha_k) \to \beta_k$ for the outage management (grey part in Fig. 8) that is accessible by the LLM and considers the optimisation (18). Algorithm 3 also computes an alternative if the initial user request is infeasible, e.g. an earlier starting time $t^0$ if it exists. This algorithm can easily be expanded further to consider multiple alternatives earlier, later, or at different times.

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

**Algorithm 3:** Outage Management $h(\alpha) \to \beta$.

**Input:** $\alpha = [t^0, \Delta t, \hat{g}]$
**Output:** $\beta$
**Initialize:** $\tau$, $P_D$, $\lambda$,
$\Omega^T = \{t^0, t^0+1, \ldots, t^0+\Delta t\}$,
$\tau_{\hat{g}\hat{t}} = 0 \quad \forall \hat{t} \in \Omega^T$
**for** *each* $\hat{t} \in \Omega^T$ **do**
  solve (18) with input $P_{D\hat{t}}$, $\tau_{\hat{g},\hat{t}}$ and output $s_{\hat{t}}$
**if** $\bigwedge_{\hat{t}\in\Omega} \|s_{\hat{t}}\| = 0$ **then**
  $\beta =$ '*requested outage plan is possible.*'
**else**
  **while** $\left(\bigwedge_{\hat{t}\in\Omega} \|s_{\hat{t}}\| > 0\right) \bigwedge (t^0 \geq 0)$ **do**
    $t^0 = t^0 - 1$, $\tau_{\hat{g}t} = 0$,
    $\Omega^T = \{t^0, t^0+1, \ldots, t^0+\Delta t\}$
    solve (18) with input $P_{Dt}$, $\tau_{\hat{g},t}$ and output $s_t$
  **if** $\bigwedge_{\hat{t}\in\Omega} \|s_{\hat{t}}\| = 0$ **then**
    $\beta =$ '*requested outage plan is not possible, but starting at* $t^0$ *is possible.*'
  **else**
    $\beta =$ '*requested outage plan is not possible.*'
return $\beta$

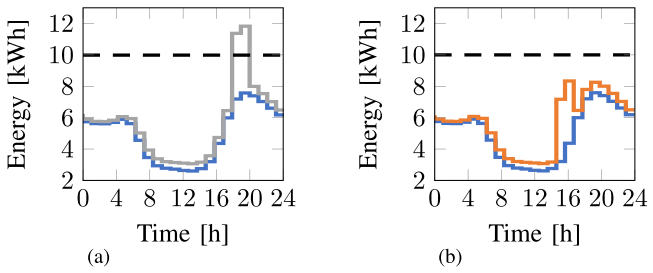



Fig. 9. The expected power profile in this neighbourhood (black). (a) An initial user request with EV charging times at 17:00 would increase the total power profile in this neighbourhood (grey) and exceed the physical constraints of the low-voltage grid (dotted black). (b) is the adjusted total power profile (orange) if the charging starts at 15:00.

## V. Case Study

### A. Test Systems and Assumptions

The studies investigate the performance of the interactions with the chat and the use cases in Section IV. As all physical feasibility checks rely on established power system tools, performance comparisons focus on interaction robustness, correctness, and system-level utilisation rather than solver runtime. In the first use case, connecting residential to the low-voltage system, we assumed a winter workday without a heat pump, aggregated hourly from [25]. We used a scaled 2019 duck curve shown as the black line in Fig. 9 [26] for $\Omega^T = \{1, 2, \ldots, 24\}$ hours. We assume a resident needs to charge daily $3.6 \times 1.15 = 4.14$ kW for two hours (1.15 is a safety margin). A preferred charging time is 17-18h. However, some neighbours may be flexible in shifting the charging to other times. We assume shared preference $\rho_t^{\text{shared}} = 0 \quad \forall t \in \{1 \ldots 16, 19 \ldots 24\}$, and $\rho_t^{\text{shared}} = 0.5 \quad \forall t \in \{17, 18\}$. We sample preference noise $\eta_u \sim \text{Uniform}(0,1)^{|\Omega^T|}$ for $u = 1, \ldots, 15$ residents. We sample charging times from the probabilities

$$\rho_u = (1-\varepsilon)\,\rho^{\text{shared}} + \varepsilon\,\frac{\eta_u}{\sum_i \eta_i} \tag{20}$$

either homogenous $\varepsilon = 0$ or heterogenous charging preferences $\varepsilon = 1$. For each resident two charging preferences were sampled $T_u \subset \Omega^T$ with $|T_u| = 2$ from $\rho_u$ without replacement.

The second use case on improving contracting with SMEs was implemented on the 179-bus Oberrhein MV grid applying the pandapower package [27]. The network is shown in Fig. 15, considering higher loading levels than the default network. The mean active loading level across all buses is shown in Fig. 16. The (spatial-) correlation between loads is from the default network data. The active power levels of generators are accordingly increased. The default AC power flow was used, and the security was assessed with $V^{\min} = 0.975$, $V^{\max} = 1.03$, $L^{\min} = 0\%$ and $L^{\max} = 60\%$. Running the AC power flow shows the network is secure, only slightly below the voltage limit at at least one bus (see Fig. 16). The chat can add an active load to one bus, and depending on where and what power level is added, it may make the system insecure. For example, Fig. 15 shows all locations where adding a single, constant $P_t = 2$ MW $\forall t \in \Omega^T$ load at all times leads to insecurity.

The third use case on outage management considered the IEEE 118-bus system that has 18 generators. A time horizon of 365 days was considered. For each day, 10 wind power plants and load data from the real French transmission system were used [28]. 10 transmission lines were uniformly, randomly sampled as N-1 security constraints (shown in Fig. 18). The load levels were scaled by 0.5 to receive N-1 secure generator dispatches when solving 365 SCOPFs for each day. Then, 50% of random generators were assumed with planned maintenance starting at a random week of the year for a duration of 4 weeks. GUROBI 12.0.1 was used to solve the SCOPFs. Running Algorithm 3, in the chat called 'OutagePlanner', solves 365 SCOPFs and takes around 1 min in total on a standard laptop.

The chats and LLMs were developed within the Wolfram Alpha Chatbook environment, which provides an API for LLM models and Python 3.11.8. The code is available at [29]. Functional programs are implemented as tool calls matching $h(\alpha) \to \beta$ with instruction metadata $h^{inst}$ (11). Personas act as (12), determining which functions $\mathcal{H}_i \subset \mathcal{H}$ may run. Tool access is explicitly assigned by the developer of the system, realising the selection $\mathcal{H}$ so that each persona accesses only their specified set of functions. Wolfram Mathematica 14.0 and 14.2 were used [30]. The LLM models were 'Wolfram', 'OpenAI-4', 'OpenAI-4 Turbo' and 'OpenAI-4 Turbo Preview'.

### B. Improving Grid Efficiency by Customising Electricity Contracts for Residents

Standard contracts are based on the assumption of homogeneity at the user, leading to grid inefficiency. This study investigates the potential efficiency increase when using customised over standard contracts for the residential use case (Section IV-A). The study assumes the following for the proposed system: the

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

**Energy supplier:** Hello! How can I assist you today? Are you looking to estimate your electricity consumption or need help with something else?

**User:** yes

**Energy supplier:** Great! Let's start by estimating your electricity consumption. I'll need to ask you a few questions to get a better idea of your usage.

1) How many hours per day do you use your washing machine?
2) How long do you use your dishwasher each day?
3) How many hours is your TV on daily?
4) How many hours do you use your computer each day?
5) How long are your lights typically on each day?
6) Do you have an electric vehicle? If so, how many hours do you charge it daily, and what time do you usually start charging it?

Feel free to provide the details, and we'll proceed from there!

**User:** 3h washing, 2h dish, compute none, lights 8h, EV 1h at 18 : 00

**Enery Supplier:** Thank you for providing the details. Let's summarize your input:

Washing machine: 3 hours per day
Dishwasher: 2 hours per day
Computer: 0 hours per day
Lights: 8 hours per day
Electric Vehicle (EV): 1 hour per day starting at 18:00

I will now calculate an estimate of your average electricity consumption per hour using this data.

Used ElectricityConsumption

I've calculated your estimated electricity consumption per hour based on the information you provided. Here's the breakdown of your daily electricity profile (in kWh) for each hour of the day:

1) 00:00 - 01:00: 0.1619 kWh
2) 01:00 - 02:00: 0.1198 kWh
3) 02:00 - 03:00: 0.1099 kWh
4) 03:00 - 04:00: 0.1069 kWh
5) 04:00 - 05:00: 0.1087 kWh
6) 05:00 - 06:00: 0.1302 kWh
7) 06:00 - 07:00: 0.2502 kWh
8) 07:00 - 08:00: 0.3595 kWh
9) 08:00 - 09:00: 0.3708 kWh
10) 09:00 - 10:00: 0.3422 kWh
11) 10:00 - 11:00: 0.3229 kWh
12) 11:00 - 12:00: 0.3287 kWh
13) 12:00 - 13:00: 0.3631 kWh
14) 13:00 - 14:00: 0.3597 kWh
15) 14:00 - 15:00: 0.3204 kWh
16) 15:00 - 16:00: 0.2912 kWh
17) 16:00 - 17:00: 0.2923 kWh
18) 17:00 - 18:00: 0.3574 kWh
19) 18:00 - 19:00: 4.0616 kWh (includes charging your EV)
20) 19:00 - 20:00: 0.5196 kWh
21) 20:00 - 21:00: 0.4673 kWh
22) 21:00 - 22:00: 0.3916 kWh
23) 22:00 - 23:00: 0.3241 kWh
24) 23:00 - 00:00: 0.2400 kWh

Now, I will check if this profile is feasible with the power system network constraints and develop a suitable energy supply contract for you.

Used DevelopContract

It seems that the current electricity consumption profile is not feasible with the power system network constraints due to the high demand when charging your electric vehicle at 18:00. However, alternative charging times are suggested to make it feasible. You can consider charging your EV during the following time slots instead: 00:00 - 17:00. Would you like to adjust the EV charging time to one of these alternative time slots? If so, please let me know the preferred time, and I will recalculate the consumption and contract details for you.

**User:** yes, it would be great if you could adjust for 16:00

**Energy Supplier:** Great! Let's adjust the EV charging time to start at 16:00 and see if a feasible contract can be developed. I'll recalculate the electricity consumption profile with this new charging time.

Used ElectricityConsumption

I've recalculated your estimated electricity consumption profile with the EV charging time adjusted to start at 16:00. Here's the updated daily electricity profile (in kWh) for each hour of the day:

1) 00:00 - 01:00: 0.1619 kWh
2) 01:00 - 02:00: 0.1198 kWh
3) 02:00 - 03:00: 0.1099 kWh
4) 03:00 - 04:00: 0.1069 kWh
5) 04:00 - 05:00: 0.1087 kWh
6) 05:00 - 06:00: 0.1302 kWh
7) 06:00 - 07:00: 0.2502 kWh
8) 07:00 - 08:00: 0.3595 kWh
9) 08:00 - 09:00: 0.3708 kWh
10) 09:00 - 10:00: 0.3422 kWh
11) 10:00 - 11:00: 0.3229 kWh
12) 11:00 - 12:00: 0.3287 kWh
13) 12:00 - 13:00: 0.3631 kWh
14) 13:00 - 14:00: 0.3597 kWh
15) 14:00 - 15:00: 0.3204 kWh
16) 15:00 - 16:00: 0.2912 kWh
17) 16:00 - 17:00: 3.8923 kWh (includes charging your EV)
18) 17:00 - 18:00: 0.3574 kWh
19) 18:00 - 19:00: 0.4616 kWh
20) 19:00 - 20:00: 0.5196 kWh
21) 20:00 - 21:00: 0.4673 kWh
22) 21:00 - 22:00: 0.3916 kWh
23) 22:00 - 23:00: 0.3241 kWh
24) 23:00 - 00:00: 0.2400 kWh

Now, I'll check if this updated profile is feasible and develop a suitable energy supply contract for you.

Used DevelopContract

Good news! The updated electricity consumption profile with the EV charging time adjusted to start at 16:00 is feasible. A suitable energy supply contract can be offered to you with the following details:

**Base Level Consumption**: 0.6 kWh
**Flexible Level During Charging Hours**: 4.8 kWh

This contract will accommodate your electricity needs efficiently, including the charging of your electric vehicle. If you have any questions or need further assistance, feel free to ask!

Fig. 10. Example chat between an energy supplier and a residential user.

residential user sends an initial request $r_t \quad \forall t \in \Omega^T$ at times $\Omega^T = \{1, 2, \ldots, 24\}$ to an energy supplier via the chat (example in Fig. 10). The chat-system adjusts the profile $r_t \quad \forall t \in \Omega^T$ by shifting the flexible part using functions $h_1$ and $h_2$ introduced in Section IV-A. The proposed customised contracts can have

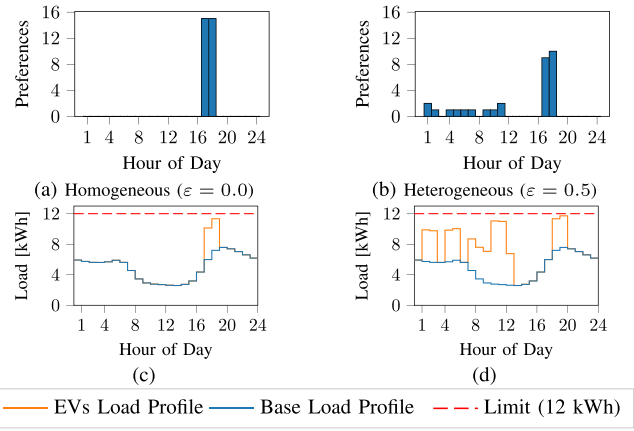



Fig. 11. Comparing user preferences and EV charging contracts under different heterogeneity levels. 15 households with the charging preferences (a) can connect only 1 EV (c). However, customising contracts exploiting heterogeneous preferences (b) allows 7 EVs to connect (d).

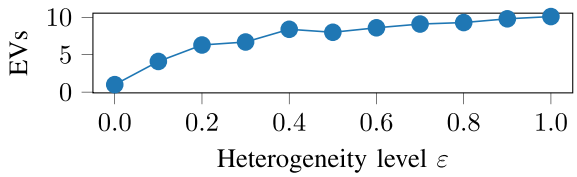



Fig. 12. Average number of EVs that can connect depends on the heterogeneity in the charging time preferences ($\varepsilon = 0$ homogeneous, $\varepsilon = 1$ heterogeneous).

off-peak and on-peak times (as shown in Fig. 5) or customised price levels leading to a potential lower energy profile shown in Fig. 9. The aggregated neighbourhood profile is the black curve. In a neighbourhood with many heterogeneous power profiles, this can lead to a 'flatter' consumption profile.

We investigated the impact of customising contracts in a neighbourhood of 15 residents who have varying EV charging preferences. We applied a mixed integer linear program that maximised EV connections constrained by 12 kWh maximal capacity for each hour for varying heterogeneity of charging time preferences $\varepsilon = \{0, 0.1, \ldots 1\}$, which we each repeat 10 times. Fig. 11 shows that more EVs can be connected if the residents have varying preferences and individual customised contracts. In this example, 7 times more EVs can be connected with the same grid capacity, as each resident received a different, preferred time slot to charge. Fig. 12 shows that already a low level of heterogeneity enables DSOs and energy suppliers to connect more EVs to the same low voltage system, as long as this heterogeneity can be exploited with customised contracts at large. In other words, the grid can be used at a constantly higher level when customising capacity contracts over standard contracts, assigning each household the same capacity at all times.

### C. Function Calling With LLMs in Residential Contracting

The proposed system combines LLM with analysis of the grid through function calling. This study investigates the accuracy of the tool calling (Section III-B), LLM settings and Algorithm 1

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

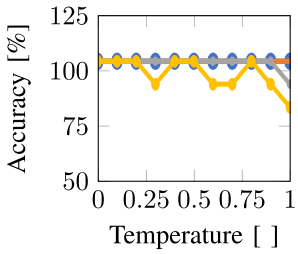
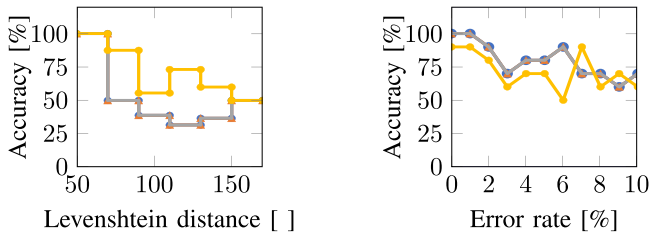



Fig. 13. Performances of $g$ to correctly identifying parameters $\alpha$ from text $u_i$ (▬), of the functional program $h$ to correctly output $\beta$ (▬), of $v$ to pass information from program to program (▬) and of $f$ to include parameters $\beta$ in response $r_i$ (▬).




(a) Paraphrasing (b) Typographical

Fig. 14. Sensitivity of the user inputs $u_i$ when paraphrased (a) or includes typo errors (b); performance of $g$ to correctly identifying parameters $\alpha$ from text $u_i$ (▬), of the functional program $h$ to correctly output $\beta$ (▬), of $v$ to pass information from program to program (▬) and of $f$ to include parameters $\beta$ in response $r_i$ (▬).

on the residential use case (Section IV-A). The paragraph $u^0$ = *'I use my washing machine 2h and dishwasher 1.5h every day. My TV is around 3h on and computer 4h. The lights are on 14h a day. My EV charges 1h every day starting at* $10:00$. *First, compute my electricity consumption by using the electricity consumption tool. Second, take my daily electricity consumption profile as input to check the physical feasibility with the physical feasible contract tool. Use this tool to provide me with a electricity supply contract. If the tool does not give a correct response, just simply try again and do not apologize.'* was tested 10 times, where OpenAI's gpt-4-turbo-preview' has access to the two functions to estimate electricity consumption and check physical feasibility (Fig. 6). The tuning temperature $\tau = 0, 0.1, \ldots 1$ was studied, so in total the paragraph was processed 110 times. Fig. 13 shows the accuracy of identifying parameters $\alpha$ from text, assessing the performance of $g$ in (13), and providing correct information $h$ in (14) to the chat. As all power-system computations are executed by the functional programs $h$, repeated requests with identical parameters $\alpha$ always return the same $\beta$. The variation observed in Fig. 13, therefore, originates solely from the extraction and chat process ($g$, $v$, $f$), rather than from the underlying calculations performed by $h$. $\tau = 0$ leads to deterministic, consistent chat replies; however, a poor performance. $\tau = 1$ has higher flexibility, and $\tau = 0.7$ seems to be a good compromise, aligning with literature observations.

### D. Sensitivity to User Input in Residential Contracts

Residential users specifically have heterogeneous knowledge about the energy system, varying intentions, and varying language fluency. This study investigates the sensitivity of Algorithm 1 to user input $u_i$ variations, considering paraphrasing and typographical errors. The first part of this study uses a pre-trained model [31] to paraphrase user inputs. The baseline user input was $u_0$ = *'I use my washing machine 2h and dishwasher 90min every day. My TV is around 3h on and computer 4h. The lights are on 14h a day. My EV charges 1h every day starting at 10:00.'*. The paraphrasor (with default settings) was applied 10 times sentence by sentence on this user input. Then, 100 combinations of the paraphrased sentences were uniformly sampled. The Levenshtein distance measures the difference between paraphrased and baseline sentences. For example, one of the four sentences in the baseline was *'I use my washing machine 2 h and dishwasher 90 min every day.'* which could be paraphrased as *'i use my washing machine 2 h and my dishwasher 90 min every day'* with a Levenshtein distance of 13 or as *'i am using my washing machine every day for 2 hours and my dishwasher every 90 minutes'* with a Levenshtein distance of 51. As one can see, the latter paraphrasing led to a different meaning. Fig. 14(a) presents the average accuracies within bins of width 20.

The second part of this study investigates typographical errors. Different levels of error rates $\{0.01, 0.02, \ldots 0.1\}$ for typographical errors were considered, e.g., a rate of 3% means that there are 3 typos in a text of 100 characters. The typos were randomly sampled from character swaps, missing characters, adding extra characters, replacing a random character with keyboard-neighbour letters, replacing visually similar characters, skipping characters, adding or skipping a random space, repeating a character, or removing a letter in syntactic duplications. The Python code [32] simulated 10 samples for each error rate, so 100 different user inputs. For example, *'I use my washing machine 2h and xishwasher 90 min every day.'* has two typos and an error rate of 2% and *'Iusc my washing acbime 2h and dishwasher 90 min evvery day.'* has an error rate of 10%. Together with the first part, the paraphrasing study in Fig. 14(a), this provides a comparison between unaltered inputs and controlled user-input variations. These settings make explicit how incorrect or distorted entries in $u_i$ propagate through the modelling chain and influence the correctness of $\beta$. Fig. 14(b) shows decreasing performance with increasing typos. An error rate above 1-2% may significantly influence the success of user engagement. The system's strengths are the flexibility to heterogeneous user input. However, a minimal accuracy of the user input is needed: the intent needs to be correctly stated, as demonstrated in the first part, and the level of fluency needs to be high enough to extract the correct intents, as demonstrated in the second part.

### E. Autonomous Connecting Industry to Medium Voltage Grids

The high complexity of customisation connection requests leads to long processing times. This case study investigates whether the approach shows potential to reduce complexity and accelerate the planning in distribution grids, bridging commercial business developers and a DSO. We consider the Oberrhein MV grid with a high feed-in scenario and an increased load that varies over time. The DSO has voltage issues in specific

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

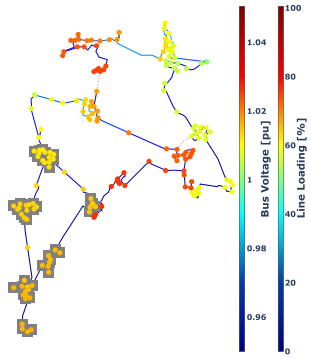



Fig. 15. Congested Oberrhein network. Locations are marked with a grey box where adding 2MW constant power is physically infeasible at 19 h.

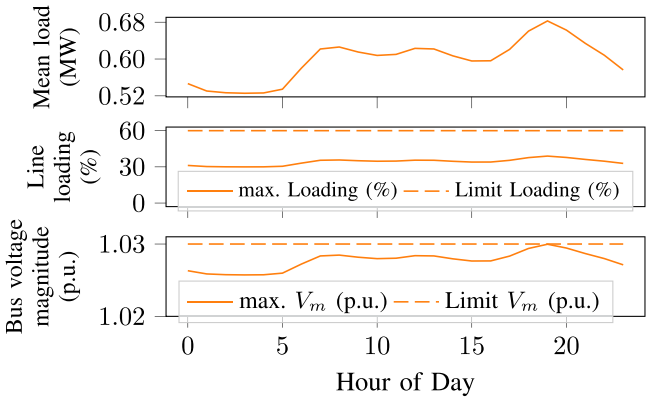



Fig. 16. Various metrics over a typical 24 h day in the Oberrhein network. The max voltage over all buses $m$ peaks at 19h due to a peak in the load nearly surpassing the voltage limits. Adding more load to specific buses may lead to network instabilities and require careful assessment.

areas at a specific time of the day (at 19 h see Fig. 16). A developer considers building a large commercial building with a nominal power rating of 2MW, and can decide between multiple locations. The use of the commercial building determines a daily electricity profile; the developer of the project has flexibility in renting parts of the building to various commercial sectors, e.g. adjusting opening times depending on power system constraints. Some site locations don't require compromising building plans; others do require compromises. Deciding this is complex, as the best business case for this commercial building requires assessing location and possible use cases simultaneously. The developer can use the chat to assess whether the intended location and power profile are feasible, e.g. nominal (constant) power is physically infeasible at the locations marked in Fig. 15. The state-of-the-art planning of this process would involve teams from multiple stakeholders, such as developers, DSO and others, which takes significant planning time. The proposed chat, however, iterates with the developer until a feasible combination of use case and location, and grid requirements is found in near-real time. Fig. 17 is an example chat. The MVContractPlanner (green box in figure) is the function $h(\alpha)$ that runs Algorithm 2, calling the AC power flow for all $24h$ on the Oberrhein system on bus $b = 5$ as introduced in Section IV-B. The bus $b = 5$ and the

**Developer:** Can you help me plan where I can build a new site of 2MW for commercial use?

**DSO:** Certainly, I can help you plan an electricity contract for your new site. To start, could you please provide the zip code for the location where you are considering building the site? Keep in mind the zip code should be between 0 and 146. Once we have the location, we'll proceed with assessing the power profile over a regular day. This will help us determine the feasibility of connecting to the grid at that location.

**Developer:** Zip code is 5

**DSO:** Great! Now that we have the zip code, please provide the expected electricity usage in megawatt-hours (MWh) for each hour of a regular day at your site. This will be a 24-hour profile, with values typically between 0 and 4 MW for each hour. For example:

- Hour 1: 1.5 MW
- Hour 2: 2.0 MW
- ...
- Hour 24: 1.8 MW

Please confirm the power profile for your site so we can assess its feasibility.

**Developer:** Could you help me make an estimate of the electricity demand profile? I plan to have regular opening hours for the commercial building with some shops opening longer than others. I consider to run several restaurants in the evening and open the shops until around 8pm.

**DSO:** Certainly! Let's create a rough estimate of your electricity demand profile based on typical usage patterns for commercial buildings with shops and restaurants. Here's a potential profile:

- Hours 1 – 6 (Midnight to 6 AM): Minimum activity, primarily for basic lighting and security systems. Let's estimate around 0.5 MW.
- Hours 7 – 9 (7 AM to 9 AM): Shops start opening, so there will be an increase in demand. Estimate around 1.0 MW.
- Hours 10 – 13 (10 AM to 1 PM): Peak shopping hours, high activity. Estimate around 2.0 MW.
- Hours 14 – 17 (2 PM to 5 PM): Steady shopping activity. Estimate around 1.5 MW.
- Hours 18 – 20 (6 PM to 8 PM): Shops closing, restaurants starting to pick up, so demand remains high. Estimate around 2.0 MW.
- Hours 21 – 23 (9 PM to 11 PM): Peak restaurant hours, shops closed. Estimate around 2.5 MW.
- Hour 24 (11 PM to Midnight): Late evening, restaurants closing. Estimate around 1.0 MW.

Would you like to adjust any of these estimates, or should I proceed with this profile to assess its feasibility?

**Developer:** Sounds reasonable. Thanks.

**DSO:**

Used MVContractPlanne

The analysis shows that the requested power at hour 20 (8 PM to 9 PM) is not feasible for connection to the grid at the specified location. We need to adjust the power demand for that specific hour. Would you like to reduce the power demand for hour 20, or make adjustments elsewhere in the profile to accommodate the required power? Let me know how you'd like to proceed, and we can reassess the feasibility.

**Developer:** Yes, what if I close already the shops earlier? Let's say at 6pm. Can you check if this resolves the issue?

**DSO:** Sure! If you close the shops earlier at 6 PM, we can adjust the power profile as follows:

- Hours 18 – 19 (6 PM to 7 PM): Reduce demand as shops start closing, estimate around 1.5 MW instead of 2.0 MW.
- Hour 20 (8 PM to 9 PM): Restaurants remain open, we'll reduce slightly to 1.8 MW to see if it's feasible.

Let's reassess the profile with these adjustments.

Used MVContractPlanne

Great news! With the adjusted power profile, all times are now feasible for connection to the grid at the specified location. We can proceed with signing an electricity contract based on this profile. If you have any further questions or need additional adjustments, feel free to let me know!

Fig. 17. Chat between developer and DSO with WolframAlpha LLM. The function (green) runs AC power flows on the requested profile (Section IV-B).

power values $P_t$ are extracted from the chat and passed as input $\alpha = \{b, P_t\}$ to this function $h(\alpha)$. This study demonstrated the approach is promising to reduce complexity through automating a connection request for a DSO, interfacing customer management and network analysis tooling. Note that the state-of-the-art connection request may take around months to be completed; here, these can be resolved within minutes with possibly minimal human oversight from a DSO.

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

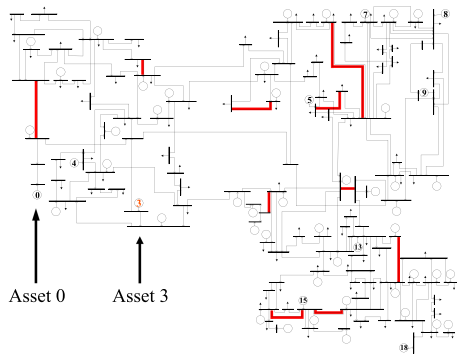



Fig. 18. IEEE 118-bus system where scheduling generator 3 at the same time as generator 0 would lead to N-1 insecurity of the red marked lines. Other times shown in Fig. 20 do not lead to insecurity.

**Asset owner:** Hello. I would like to plan an outage of my generator asset. Can you help me with that?

**TSO:** Of course! To help you plan the outage for your generator asset, I'll need the following information:

1. **Starting Week**: A number between 1 and 50 indicating when you want the outage to begin.
2. **Duration**: The duration of the outage in weeks, between 1 and 10.
3. **Asset Number**: The asset number of the generator you wish to plan the outage for, which should be a number between 0 and 18.

Once I have this information, we can proceed with planning the outage.

**Asset owner:** The asset number is 3. I plan to take it in outage at week 3 of this year. The duration is 2 weeks.

**TSO:**

Used OutagePlanner

It seems that starting the outage for your generator asset number 3 in week 3 is not possible. However, there are some alternatives you can consider:

1. **Start earlier**: You could begin the outage for a duration of 2 weeks starting between day 1 and day 13.
2. **Start later**: You could also choose to start the outage for a duration of 2 weeks between day 55 and day 117.

Let me know if any of these options work for you, or if you would like to explore other possibilities.

**Asset owner:** Starting maintenance at day 56 would work for us. Can you confirm this works?

**TSO:**

Used OutagePlanner

Starting the maintenance for your generator asset number 3 at day 56, which corresponds to week 8, is confirmed to be possible. You can proceed with the outage for the specified duration of 2 weeks.
If you have any more questions or need further assistance, feel free to ask!

Fig. 19. Chat to plan an outage in the transmission IEEE 118-bus system using WolframAlpha LLM. The green functions perform SCOPFs. .

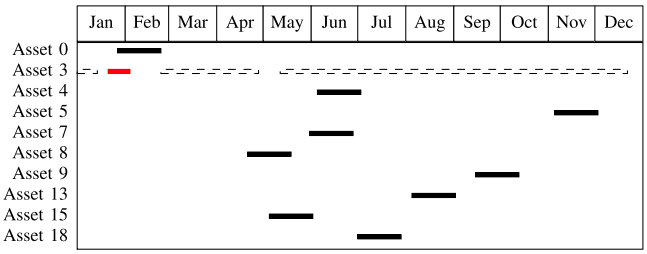



Fig. 20. Gantt chart of scheduled outages for 10 generators. The initially requested time (red) of the owner asset 3 is not physically feasible. The proposed chat provides alternative times back to the owner (dashed lines).

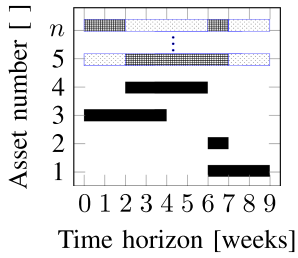



Fig. 21. Possible security implications of the maintenance planning approach (see Fig. 8). A brute force extruder could vary $\alpha$ to learn about the power network; however, the extruder would maximally obtain information about the maintenance possibilities of each generator through $\beta$, for example, when assets 5 to $n$ can be switched off.

### F. Accelerate Outage Planning in Transmission System

Current outage planning takes currently months to complete, and the requests pile up due to complex planning constraints and frequent follow-up discussions. This study investigates the potential of the proposed approach to accelerate and customise requests for disconnecting assets from the grid. An asset owner plans maintenance of the generator 3 in the IEEE 118-bus system, Fig. 18. The asset owner interacts with the grid network studies (calling Algorithm 3) in near real-time via the chat, Fig. 19. The initial request to maintain in week 3 is assessed with OutagePlanner $h(\alpha)$ running Algorithm 2. The OutagePlanner solved several SCOPFs showing the request would lead to N-1 insecurity. Another generator 0 is already scheduled for maintenance during these times (Fig. 20). These two generators, 0 and 3, are close to each other (Fig. 18), hence simultaneous maintenance is not N-1 secure. The chat proposes alternative times (Fig. 19) where the week 8 suits the owner, and the scheduled maintenance is confirmed. The time for the proposed LLM-based system and GUROBI to communicate with the user and solve 365 SCOPFs took around 1min. In comparison, the state-of-the-art approach requires requests and often follow-up conversations between a TSO with the asset owner, which overall can take several months. An additional benefit is that grid code documentations, which change frequently, can be instantly updated without reskilling employees; here, updating a document to the chat context $\omega_i$.

### G. Privacy in Transmission Assets Maintenance Planning

LLM-based systems have limitations with security and privacy of the user inputs. This analysis investigates whether the proposed approach may have security breaches that could make sensitive information from a TSO available to the user of the chat. In a real-world implementation, the information within the Python scripts and grid data would not be publicly accessible. However, an attacker may investigate many variations of inputs $\alpha$, trying to learn grid behaviour and settings, like existing planned outages, for example, which can be considered sensitive information. Fig. 21 shows the maximal information one could learn. One could learn the times for each generator when it's feasible to switch it off and when not, as shown with black bars through many requests $\alpha$ and analysing $\beta$. Analysing these may reveal information when (and where) the grid is more congested

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

and weak. One could limit use only for internal stakeholders or limit the number of requests to prevent these attacks. In the two, the normal and adversarial querying settings, the chat-user only receives the feasibility outcomes $\beta$ produced by the functional programs $h$, while no internal grid data or intermediate quantities become part of the chat context $\omega_i$. This structural separation ensures that the information exposed through $\alpha$ and $\beta$ is limited by the tool interface itself. We discuss the limitations on privacy and security further in Section V-H.

### H. Discussion

To the best of our knowledge, the proposed approach is the first to apply LLMs to electricity contract design, managing end-users of the grid, system operators, or energy suppliers. The key benefit of the LLM is to coach the user through the contract design, where the user has heterogeneous knowledge about this process. This makes the contract design scalable to a large number, where the state-of-the-art contracts have to be standardised and typically are supported with long guidelines, rules, and FAQs. This paper presented the initial idea and studied some algorithmic properties (like temperature), sensitivity to user input, privacy, and the impact this system can have within three power system use cases.

While the case studies present a promising impact, the methods and study have several limitations. LLMs have vulnerabilities that need to be studied in detail before deploying this system. The vulnerabilities typically involve security (e.g., prompt hacking and adversarial attacks) and privacy attacks [33], and various methods for system design can be applied (e.g., API hardening or paraphrasing). Section V-G studied one privacy attack on grid data with a basic brute force strategy. Beyond the proposed measures to secure privacy, the hosting setup and data policies primarily determine privacy. For example, the LLM system can be either hosted in a cloud environment or in a private environment, where the data stays in this environment, ensuring high privacy. In either environment, regular chats are typically not used to retrain models live, ensuring the privacy of entered information. When self-hosted, the developer can specify the type of information recorded from the user, and the chat is not stored on the server. Beyond these future needs on privacy and security, this work has not yet explored fine-tuning, a wider variety of the intent of users, exhaustive robustness to very high error rates, real-world variety beyond typographical errors, or evolving model behaviours. No standard benchmarks exist yet to compare this model against others. This study also did not focus on a variety of baselines and improved evaluation frameworks that go beyond the accuracy of responses and parameter correctness to exhaustively compare different models and settings. The real-world applicability needs to be studied further, for example, to consider dynamic operating envelopes that can be defined through contracted flexibility, the expected cost reduction, and customer adoption of the system. While the methodological framework is independent of a specific interface, designing a dedicated User Interface for deployment is a natural next step. All the aforementioned limitations need to be addressed to develop this system further for production readiness.

## VI. Conclusion

This paper introduced a new interaction layer between end-users and system operators that has not existed in current practice, where such studies are carried out manually and only by experts. Applying this layer via LLM-based chats to customise contracts can drastically reduce the transaction costs of interacting with electricity users. Electrical grids conventionally standardise energy supply contracts, not considering the hidden flexibility in the diversity of preferences and devices. The reduction of transaction costs can make the customisation of supply contracts profitable, reduce outage planning durations, and enable new tooling for SMEs to directly test their electricity connections. For example, we demonstrate a 7-fold increase of EVs that can connect to the grid, exploiting the heterogeneity of devices and preferences. The chat system shows low error rates, robustness to typos, and adaptability. We believe this approach can enhance the digitalisation of the electricity system and enable the energy transition.

## Acknowledgment

We thank our industrial collaborators in discussing these three use cases (Vikaran Khanna from National Grid ESO, U.K., Luc Nies from Alliander, the Netherlands, and Arjan van Voorden, Stedin, the Netherlands). Some parts of the manuscript benefited from grammar and clarity improvements using automated tools, including Grammarly and ChatGPT. The authors remain responsible for all content.

## References

[1] G. Ji, W. Wu, and B. Zhang, "Robust generation maintenance scheduling considering wind power and forced outages," *IET Renewable Power Gener.*, vol. 10, no. 5, pp. 634–641, 2016.

[2] G. Crognier, P. Tournebise, M. Ruiz, and P. Panciatici, "Grid operation-based outage maintenance planning," *Electric Power Syst. Res.*, vol. 190, 2021, Art. no. 106682.

[3] M. Behrangrad, "A review of demand side management business models in the electricity market," *Renewable Sustain. Energy Rev.*, vol. 47, pp. 270–283, 2015.

[4] M. S. Bakare, A. Abdulkarim, M. Zeeshan, and A. N. Shuaibu, "A comprehensive overview on demand side energy management towards smart grids: Challenges, solutions, and future direction," *Energy Informat.*, vol. 6, no. 1, 2023, Art. no. 4.

[5] E. Ruokamo, M. Kopsakangas-Savolainen, T. Meriläinen, and R. Svento, "Towards flexible energy demand–preferences for dynamic contracts, services and emissions reductions," *Energy Econ.*, vol. 84, 2019, Art. no. 104522.

[6] P. A. Olujimi and A. Ade-Ibijola, "NLP techniques for automating responses to customer queries: A systematic review," *Discover Artif. Intell.*, vol. 3, no. 1, 2023, Art. no. 20.

[7] A.K. Srivastawa, "Exploring contract management in the digital age: The impact of artificial intelligence," *Jus Corpus LJ*, vol. 4, 2023, Art. no. 737.

[8] S. N. Kirshner, Y. Pan, J. X. Wu, and A. Gould, "Talking terms: Agent information in LLM supply chain bargaining," *Decis. Sci.*, vol. 57, pp. 9–23, 2026.

[9] S. Abdelnabi, A. Gomaa, S. Sivaprasad, L. Schönherr, and M. Fritz, "Cooperation, competition, and maliciousness: LLM-stakeholders interactive negotiation," in *Proc. Adv. Neural Inf. Process. Syst.*, 2024, vol. 37, pp. 83548–83599.

[10] C. N. Abiagom and T. I. Ijomah, "Enhancing customer experience through AI-driven language processing in service interactions," *Open Access Res. J. Eng. Technol.*, vol. 7, pp. 014–021, 2024.

[11] S. Kolasani, "Optimizing natural language processing, large language models (LLMs) for efficient customer service, and hyper-personalization to enable sustainable growth and revenue," *Trans. Latest Trends Artif. Intell.*, vol. 4, no. 4, 2023.

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.

[12] V. Soni, "Large language models for enhancing customer lifecycle management," *J. Empirical Social Sci. Stud.*, vol. 7, no. 1, pp. 67–89, 2023.
[13] K. Yang et al., "If LLM is the wizard, then code is the wand: A survey on how code empowers large language models to serve as intelligent agents," in *Proc. ICLR Workshop Large Lang. Model (LLM) Agents*, 2024.
[14] S. Jin and S. Abhyankar, "Chatgrid: Power grid visualization empowered by a large language model," in *Proc. IEEE Workshop Energy Data Visual.*, 2024, pp. 12–17.
[15] Q. Zhang and L. Xie, "PowerAgent: A road map toward agentic intelligence in power systems: Foundation model, model context protocol, and workflow," *IEEE Power Energy Mag.*, vol. 23, no. 5, pp. 93–101, Sep./Oct. 2025.
[16] S. Ghosh and G. Mittal, "Agentic AI systems in electrical power systems engineering: Current state-of-the-art and challenges," 2025, *arXiv:2511.14478*.
[17] J. Heckman, F. de Koning, L. Nies, and J. Cremer, "Flexfinder: Facilitating the contracting process for medium voltage grid customers using large language models," in *Proc. 28th Int. Conf. Exhib. Electricity Distrib.*, Geneva, Switzerland, CIRED, Paper no. 502.
[18] T. Wu, M. Terry, and C. J. Cai, "AI chains: Transparent and controllable human-AI interaction by chaining large language model prompts," in *Proc. CHI Conf. Hum. factors Comput. Syst.*, 2022, pp. 1–22.
[19] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," in *Proc. Int. Conf. Learn. Representations*, 2013.
[20] C. Guo, G. Pleiss, Y. Sun, and K. Q. Weinberger, "On calibration of modern neural networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1321–1330.
[21] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 5998–6008.
[22] T. Brown et al., "Language models are few-shot learners," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, vol. 33, pp. 1877–1901.
[23] P. F. Christiano, J. Leike, T. Brown, M. Martic, S. Legg, and D. Amodei, "Deep reinforcement learning from human preferences," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, vol. 30, pp. 4299–4307.
[24] EirGrid, "EirGrid Grid Code, Version 14.3: Section Operating Condition 2.4–Outage Scheduling," Mar. 2025. [Online]. Available: https://cms.eirgrid.ie/sites/default/files/publications/GridCodeVersion14.3.pdf
[25] M. Schlemminger, T. Ohrdes, E. Schneider, and M. Knoop, "Dataset on electrical single-family house and heat pump load profiles in Germany," *Sci. Data*, vol. 9, no. 1, 2022, Art. no. 56.
[26] IEA, the California Duck curve, IEA, Paris (n.d.), Oct. 2019. [Online]. Available: https://www.iea.org/data-and-statistics/charts/the-california-duck-curve
[27] 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.
[28] M. Sun, J. Cremer, and G. Strbac, "A novel data-driven scenario generation framework for transmission expansion planning with high renewable energy penetration," *Appl. Energy*, vol. 228, pp. 546–555, 2018.
[29] L. Jochen Cremer, "LLM-Electricity-Contracts source code," 2025. Accessed: Nov. 10, 2025. [Online]. Available: https://github.com/TU-Delft-AI-Energy-Lab/LLM-Electricity-Contracts
[30] W. R. Inc., "Mathematica, Version 14.2," Champaign, IL, USA, 2024. [Online]. Available: https://www.wolfram.com/mathematica
[31] P. Damodaran, "Parrot: Paraphrase generation for NLU," GitHub repository, 2021. [Online]. Available: https://github.com/PrithivirajDamodaran/Parrot_Paraphraser
[32] R. Kumar, "A Python package to simulate typographical errors," 2021. [Online]. Available: https://github.com/ranvijaykumar/typo
[33] B. C. Das, M. H. Amini, and Y. Wu, "Security and privacy challenges of large language models: A survey," *ACM Comput. Surv.*, vol. 57, no. 6, pp. 1–39, 2025.

Authorized licensed use limited to: TU Delft Library. Downloaded on July 14,2026 at 11:30:34 UTC from IEEE Xplore. Restrictions apply.