Bachelor Thesis

Causal Model Extraction - LLMs as Explainability Support in Cyber-Physical Systems

Maurer, Paul

Date of Birth: 03.09.2001
Student ID: 12124850

Subject Area: Information Systems and Operations Management

Studienkennzahl: 033 561

Supervisors:
Schreiberhuber, Katrin, Dipl.-Ing. B.Sc.
Sabou, Reka Marta, Univ.Prof., Ph.D.

Date of Submission: 16.06.2025

Department of Information Systems & Operations Management, Vienna University of Economics and Business, Welthandelsplatz 1, 1020 Vienna, Austria

Abstract

As technology advances at an ever-increasing pace, Cyber-Physical Systems (CPSs) are also evolving rapidly. With growing complexity, understanding their decision-making processes has become increasingly difficult. This thesis explores the potential of Large Language Models (LLMs) in assisting domain specific experts with generating initial causal models for CPSs.

The core of this thesis is a hybrid workflow that combines the reasoning capabilities and general knowledge of OpenAI's GPT-4 with expert input to create causal models for CPSs. A Smart Charging Garage serves as the focal real-world example to evaluate the workflow by comparing LLM-generated results to expert-created models.

While the results show promising outcomes, particularly in the number of reasonable answers generated, there are still gaps in precision and recall when compared to expert-only-developed causal models. These findings highlight the ongoing need for expert guidance in causal model creation. However, they also demonstrate that LLMs can significantly reduce the time and knowledge required for creating initial causal models in CPS environments.

Future studies should focus on refining LLM prompting strategies and exploring the use of other LLMs for broader applications in CPS. Other evaluation metrics and a larger sample size would also benefit the goal of generating knowledge in this topic.

Contents

1 Introduction

1.1 *Background*

In 2007, the European Union (EU) introduced a major plan to improve environmental sustainability. The initiative aimed to foster more sustainable practices and to accelerate the development of sustainable technologies [3].
With ecological sustainability being one of the main goals, the EU proposed the "Green Deal" in 2019, which can be seen as an expansion of the EU's sustainability strategy from 2007. It states that by 2050 the EU is expected to be carbon neutral. It also includes the goals of the 2030 Agenda, meaning a "55% reduction of greenhouse gas emissions compared to 1990 levels" [8].

A key aspect of this progress is its impact on the energy sector, where improving efficiency and reducing environmental harm have become major goals. To achieve this, new technologies such as Cyber-Physical Systems (CPS) are playing an important role.

"A system with a tight coupling of cyber and physical objects is called cyber–physical system (CPS)" [10]. Baheti and Gill emphasize that CPS are becoming more important, as they are able to combine computational and physical capabilities, which leads to more possibilities of interaction and functionality. They argue that CPS are expected to play a significant role in future technological developments across diverse fields [2].
To effectively carry out the ecological transition required by the EU, it is assumed by Kojonsaari and Palm that this "development will require the gradual evolution of distribution networks from passive to active and the development of so-called smart grids"[14]. Understanding how these systems behave and make decisions becomes more important, especially when aiming for efficient and transparent energy management. Smart grids are becoming an increasingly important element in the effort to revolutionize the energy sector, steering it toward greater sustainability. smart grids are "of great importance as they can be integrated with renewable energy resources and contribute towards alleviating environmental pollution" [26].
As digitalization advances and sustainability efforts grow, smart grids are gaining significance. With their increasing popularity, various use cases and challenges emerge. This reveals new research opportunities, previously unknown issues and potential benefits, not only, but especially in the energy sector.

1.2 *Problem*

To better understand the decisions that have been made by the CPS, the term Explainable Cyber-Physical Systems (ExpCPSs) has been introduced by Schreiberhuber et al. [28]. They argue that ExpCPS provide clear explanations of system decisions and behaviors, helping users understand and trust complex systems by revealing why certain states or events occur [28].
This transparency is crucial for effectively managing and controlling CPSs.
Despite the importance of ExpCPS in today's world across multiple areas, existing methods that help with explainability are developed for specific application areas only. This narrow focus limits their usefulness and applicability because it requires domain-specific experts to understand the underlying mechanisms of the CPS.
This makes it difficult to use and interpret these systems across different fields without

specialized knowledge.

To address this challenge, causal models can play a crucial role in improving the explainability of CPSs [23]. Pearl explains that by representing the cause-and-effect relationships within a system, causal models provide a structured way to understand how different components interact and influence each other. This approach helps uncover the underlying mechanisms of CPSs, making it easier for both experts and non-experts to interpret system behavior and identify potential risks or failures. This makes causal models an important part of building ExpCPSs, as they offer a way to explain system behavior in a clear and structured manner.

To tackle the problem of causal model creation, Artificial Intelligence in the form of Large Language Models (LLMs) may be useful, as this technology could provide initial causal models to support interpretability across various CPS domains [12]. Compared to existing methods that focus on explaining how decisions are made in CPS, Zhang et al. explain that LLMs offer a model-agnostic approach that can be applied to a wide range of CPS without requiring domain-specific knowledge. A model-agnostic approach means that the method does not rely on the internal structure or specific algorithms used in the system being explained. Instead, it analyzes observable inputs and outputs to generate explanations.
This flexibility allows LLMs to provide initial insights even for complex models without the help of domain experts [31].

This thesis explores how LLMs can support the creation, usage, interpretation, and optimization of initial causal models for CPSs through a tightly connected workflow that includes iterative expert feedback. The insights gained from this research will inform future work on causal models by examining how effectively LLMs can contribute to their design, interpretation, and improvement.

1.3 *Relevance*

Currently, research on hybrid workflows that combine the capabilities of LLMs with expert knowledge remains limited. In particular, there is a lack of studies exploring how effectively LLMs can suggest potential causal relations within CPSs and how accurate these causal model suggestions are.
There is a need for a quality comparison between causal models developed by experts with specialized knowledge and those generated by LLMs to evaluate the accuracy, reliability, and applicability of AI-generated models.
This would fill a knowledge gap in the subject of Cyber-physical systems, reducing the need for experts and therefore lowering development and maintenance costs while also improving the ease of access.

„The inherent complexity of an SG makes it increasingly difficult for engineers and operators to understand the system behaviour and identify root causes of anomalies“ [28].
As the complexity of CPSs will only increase with the ongoing digitalization, the need for experts that are able to create, maintain and interpret the causal models will rise as well.
The original way of creating causal models is resource-intensive and requires experts with

special knowledge to do the work.
By exploring a hybrid approach that combines the general knowledge capabilities of LLMs with expert feedback, this research aims to streamline the causal model creation process, making it more accessible and less demanding in a time and money perspective.

1.4 *Research questions and objectives of the thesis*

The thesis aims to fill the research gap by:

- Developing a hybrid-workflow that combines LLMs and feedback by expert users
- Implementing a prototype to test and evaluate the performance and accuracy of the generated causal models on a real life use case
- Investigating the quality of causal models generated by LLMs by comparing them to models created by experts.

By addressing these aspects, the research will contribute to the development of more efficient and scalable methods for causal model extraction in Cyber-Physical Systems.

The following research questions are to be answered:

- **Research Question 1**: To what extent can Large Language Models (LLMs) assist experts in providing the initial causal model within a Cyber-Physical System (CPS)?
- **Research Question 1.1**: How do the causal models generated by LLMs compare to those created by human experts in terms of quality and applicability in real-world CPS scenarios?
- **Research Question 1.2**: What are potential challenges when relying on LLMs for causal model creation?

The objectives of this thesis are to address the insufficient research on how effectively Large Language Models (LLMs) can suggest potential causal relations within Cyber-Physical Systems (CPSs).
This involves a comparative analysis of causal models developed by experts versus hybrid models created through the collaboration of LLMs and experts.
Ultimately, this work contributes to optimizing the energy sector by enhancing sustainability and energy management practices through more efficient and accesible causal model generation.

2 Literature Review

The following sections will define the core concepts required to understand the research presented in this thesis. As this research is a highly specialized topic, it is crucial to precisely define key terms in the beginning.

2.1 *Cyber-Physical-Systems*

2.1.1 *Introduction to Cyber-Physical-Systems*

Computers and software are most commonly used for tasks such as browsing the internet, writing documents, sending emails or managing personal finances. However, Lee and Seshia explain that a majority of computers operate behind the scenes in devices that are often not even visible. Examples would be the controlling device of a car, a mini computer inside a microwave or traffic management systems. These systems are called Embedded systems, and they control a wide range of functions in our daily lives. They also support systems that monitor environmental conditions, analyze data in scientific research, and facilitate interactive features in smart devices like wearable fitness trackers and smart speakers. The term for the software that drives these systems is known as embedded software.
Embedded Systems date back to the 1970s when "they were seen simply as small computers" [16]. Embedded systems have drawn significant attention not only from the IT sector for their vital role in real-time data processing, enabling instant analysis and reaction in dynamic settings, but also from various other industries. This evolution marked a shift away from the mostly sequential nature of traditional software processes toward a more flexible approach that enables fast and detailed processing of real-time events. In summary, embedded systems have played a crucial role in advancing from traditional computing methods to more dynamic and real-time computation, enabling better management and digital representation of real-world processes [16].

Baheti and Gill defined the term Cyber-Physical Systems as "a new generation of systems with integrated computational and physical capabilities that can interact with humans through many new modalities" [2]. In other words: Cyber-Physical Systems (CPSs) are complex systems that combine real-life physical processes with computational algorithms.
However, due to the wide range of applications and interdisciplinary nature of CPSs, various definitions have emerged depending on the perspective and field of study. They highlighted different industries to showcase the numerous application methods that CPSs hold.
In the medical sector for example, a CPS can be beneficial by taking over tasks such as image-guided surgery or fluid flow control. Another definition of what a CPS is, was introduced by Lee and Seshia. They claim a CPS is "an integration of computation with physical processes whose behavior is defined by both cyber and physical parts of the system" [16]. The focus shifts increasingly toward the seamless integration of physical systems and computational processes, where each influences the other through constant feedback mechanisms. Rather than examining these components individually, the real challenge lies in understanding their mutual dependencies and interactions. This approach emphasizes the need to analyze how computational models and physical actions work together dynamically, creating a complex system that cannot be fully grasped by

studying the elements in isolation.

As there is no single definition that fits all cases, it is useful to merge the different perspectives to create a more comprehensive view of Cyber-Physical Systems.
At their core, CPSs are embedded systems that link physical processes with computational algorithms, enabling real-time monitoring, control, and feedback. By combining sensors, actuators, and embedded computing, CPSs facilitate continuous interaction between the physical and digital domains, where changes in one affect the other. This dynamic interplay is essential for optimizing operations in non-digital areas , making CPSs vital for enhancing precision and efficiency in complex environments.

2.1.2 *Core Components*

As the name suggests, Cyber-Physical Systems consist mainly of a cyber- and a physical part. Liu et al. ordered the structured a CPS into three main parts. The user layer, the information system layer and in third, the physical system layer"[17].
The physical system layer includes physical elements, such as actuators, sensors and physical machines. These components are crucial for every CPS, as they measure and process physical processes in real time.
Sensors are the physical element that is capturing the event. They gather real time data from their surrounding environment.
Actuators are adapting their state in real time with information provided by the sensors, making a direct interaction between the cyber- and the physical world possible. "The sensors (cyber objects) can be used to monitor the physical environments, and the actuators /controllers can be used to change the physical parameters" [10]. In smart grids, both types of physical components play a crucial role.

Looking at a Wind Power System for example, a real life instance of a CPS, you can identify multiple sensors and actuators. A visual representation can be seen in Figure 1.

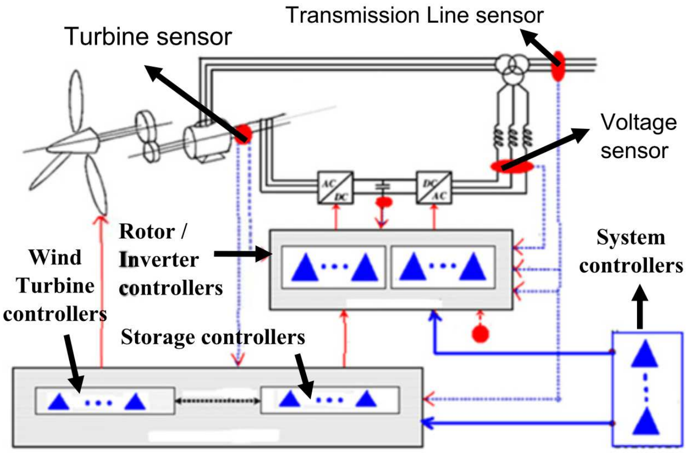

Figure 1: Wind Turbine CPS, adapted from [10]

Each of those elements has its own distinctive role in the smart grid, helping to digitally represent the events happening in the physical world. One example would be the wind

speed sensor located at the base of the revolving shaft of the turbine rotor.
When the rotor turns, the sensor captures this event and sends this information to the controlling unit of the CPS. This data can now be converted into the knowledge, that the wind is currently generating a certain amount of energy. The corresponding actuator would be the Wind Blade Controller. This unit is responsible for stopping the rotor, if a hardware malfunction is likely.
Looking at the digital counterpart, the information system layer, Liu et al. argue that it is "mainly responsible for data transmission and processing of data collected from the physical system" [17].
Hu et al. modeled the cyber part of a CPS in three sub-groups. The Next Generation Network, as described by Hu et al., is the cyber component responsible for ensuring data protection and "safety of data transmission" [10]. Here, existing algorithms are also optimized to improve the speed of data transmission.

Liu et al. explain the concepts the following way:
The data center is an element that validates the data received by the sensors. If the validation process has a negative result, the sensor will be instructed by the data center to retry the data collection process. If however the data validation is positive, the information will be stored in the data center.
According to the predefined rules that have been created by the user, the control center now sends queries to the data center. Depending on the query results, the data center structures the information based on controlling strategies. Based on the domain the CPS operates in, the actuator's state can now be adapted. The correctness of this sequence is closely monitored by a real-time controlling mechanism to ensure no data irregularities or incorrect actuator states. This sequence is in an ongoing loop, closely monitoring the events happening in real-life in order to quickly adapt to different situations [17].

To clarify this sequence, a simplified smart irrigation system provides a useful example:
The user input could be to water the plants if the soil is dry. After authentication, the data center checks the condition of the soil. If it is dry, the connected actuators are adapted to water the dry areas. If the soil is not dry, the process will be repeated. Throughout this sequence, the system is constantly being monitored on issues concerning actuators, sensors or collected data.

Another layer, the so-called User layer "mainly completes the work such as data query, strategy and safety protection under human-computer interaction". [17] It acts as the bridge between users and the system itself. It includes web servers, individual devices, and external tools that allow users to interact with the CPS.
They can send instructions to the control center, request data, and even update control strategies to improve system performance. In summary, this layer is responsible for smooth communication and helps to keep the system secure.

2.1.3 *Challenges in Cyber-Physical-Systems*

"The main obstacle of developing CPS is the lack of a unified theoretical framework of network and physical resources. Significant differences, both technically and culturally, exist between computer science theory and control theory, which almost extend to all the

areas of computer and physical systems" [17].
In other words, building CPS is challenging because there's no unified theory to connect cyber and physical components. Additionally, computer science and control systems take different approaches depending on their distinct focuses, such as data processing versus real-time control. Another important part of the design process of a CPS is choosing the level of abstraction. It needs to be high enough so that changes to different systems can be made easily without making the process too complex, but still detailed enough so that the underlying framework is clearly represented.

Liu et al. structure the main difficulties when creating CPS into three categories.
The first big challenge is Pattern Abstraction, where they argue that with currently used programming languages, handling timing, multitasking, and hardware connections is not reliable or precise enough for complex, synchronized systems. The next issue they see is Scale and Efficiency, where the main critique is focused on handling large networks with a high number of sensors with limited energy resources [17]. These systems need efficient data processing methods to minimize energy use, as power and resource constraints limit the sensors capabilities. The third and final challenge they defined is the robustness of a CPS, where the main focus lies on the vulnerability of a CPS against not only to security attacks but also to issues affecting its reliability and continuous operation.

2.1.4 *Smart Grids*

Based on the idea of Pagani and Aiello, a power grid refers to a system that transmits and distributes electricity, serving as a critical infrastructure for modern society [21]. The traditional power grid has been planned as a rather strict and non-dynamic environment. Over recent years, especially with factors like efficiency, ecological awareness and technical advancements, a switch to a more flexible approach is needed. Thus, the term Smart Grid (SG) has been introduced.
As argued by Jha et al., smart grids are CPS-based solutions that combine physical elements like sensors, actuators, data collection systems, network communication and control mechanisms to dynamically manage energy flow and adapt to different situations along the entire grid. A characteristic of a SG is its decentralized structure, where not only electricity, but also data flows in multiple directions at the same time [11]. As a SG is a subclass of a CPS, the underlying framework remains the same, meaning it consists of a physical and a cyber layer that are interconnected.
Hardware elements like energy sources, sensors, cables for energy transmission and all of the required material for the energy grid can be classified as the physical component.
Algorithms, embedded systems and data analytics for real time decision making and analysis can be categorized as the Cyber component.

2.2 *LLMs*

2.2.1 *Introduction to LLMs*

Language is essential for human communication, self-expression, and interaction with machines. The growing need for generalized models arises from the increasing demand for machines to tackle complex language-related tasks, such as translation, summarization, information retrieval, and conversational interfaces.

In the last couple of years LLMs "have emerged as cutting-edge artificial intelligence systems that can process and generate text with coherent communication, and generalize to multiple tasks"[19].

Vidgof et al. argue, that large language models have emerged as a groundbreaking development in the field of artificial intelligence, captivating the attention of researchers, industry professionals, and the general public alike over the past few years. [30]. Nicholas and Bhatia state, that these sophisticated models, which are a subclass of Artificial Intelligence (AI), have demonstrated remarkable advancements in natural language processing, generating human-like text, and tackling a wide array of tasks that were previously considered challenging for machines [20].

2.2.2 *Architecture of LLMs*

Vidgof et al. explain, that at the core of large language models lies the concept of deep learning, where models are trained on vast amounts of textual data to identify complex patterns and relationships. This training process often involves unsupervised pre-training methods, enabling the system to predict text sequences based on learned language structures and improve its performance through fine-tuning techniques [30].
These models are typically deep neural networks with billions or even trillions of parameters, making them capable of capturing intricate linguistic nuances and generating highly coherent and contextually relevant text.
Argued by Vidgof et al. LLMs are based on the transformer architecture, which has become the standard for natural language processing tasks. Transformers utilize an attention mechanism that allows the model to weigh the importance of different words in a sequence when making predictions. As described by Vaswani et al., the transformer model's success lies in its ability to replace recurrent layers with self-attention mechanisms, improving parallelization and enabling efficient training on large datasets. Patil and Gudivada state that "the transformer architecture reduced sequential computation and enabled parallelization, requiring less training time and achieving new state-of-the-art results" [22]. In simpler terms, the transformer looks at all words in a sentence at once and figures out which ones matter most, making it faster and better at understanding context.

The architecture typically consists of an encoder and a decoder. The encoder processes input text, while the decoder generates output based on the encoded information. One important feature of transformers is the multi-head self-attention mechanism, which allows the model to focus on different parts of the input text simultaneously. As noted in the text, the "multihead model is therefore able to jointly attend to information from different representations at different positions over the projected versions of queries, keys, and values" [22]. Put simply, this means the model can look at different parts of a sentence in multiple ways at the same time, which helps it better understand meaning and context.

Patil and Gudivada further explain that LLMs can follow different architectural patterns: encoder-only, decoder-only, and encoder-decoder structures. Encoder-only models like BERT are suitable for tasks requiring a deep understanding of the text, for example a classification if an email is spam or not. Decoder-only models, such as GPT, are autoregressive, meaning they generate the next word in a sequence based on previous words, making them well-suited for language generation tasks. Encoder-decoder models, use a bidirectional attention mechanism in the encoder and unidirectional attention in the decoder, making them suitable for tasks like translation and summarization.

These models are pretrained on large, unlabeled datasets to learn general language patterns. Pretraining involves objectives like masked language modeling (MLM), where random words are hidden, and the model predicts them based on the surrounding context. After pretraining, transfer learning allows these models to be fine-tuned for specific tasks, for example Prompt Tuning as presented by Patil and Gudivada.

The use of transformers in LLMs has significantly improved their ability to capture long-range dependencies in text and has made parallelization possible, reducing training time and computational costs [22].

2.2.3 *Reasoning Capabilities of Large Language Models*

Patil and Gudivada state that Large Language Models (LLMs) have shown promising reasoning capabilities, especially in tasks that require symbolic, commonsense, and arithmetic reasoning. Techniques like Chain-of-Thought (CoT) prompting allow models to break down complex problems into intermediate reasoning steps, improving performance on tasks that involve multi-step calculations, such as math word problems and quantitative reasoning tasks [22].
They discuss however, even with these advancements, LLMs can struggle with highly precise tasks that demand rigorous step-by-step problem solving, especially in cases where intermediate steps are missing or not easily derived from the pretraining data.
In areas where explicit reasoning paths can be provided, such as solving technical problems in mathematics or generating coherent logical chains in specific domains like science or engineering, LLMs have demonstrated strong capabilities.
Furthermore, challenges persist in arithmetic reasoning, where small errors in intermediate calculations can lead to incorrect results.
LLMs also encounter difficulties in tasks where model hallucination can lead to incorrect facts being presented as plausible outputs, especially in open-ended or creative reasoning scenarios. Patil and Gudivada argue that a reason for hallucination can be classified as the situation "when the provided contextual information conflicts with the parametric knowledge acquired during pretraining." [22]
Their performance generally improves with the addition of more examples or structured prompts, though computational costs increase as models are fine-tuned to better handle such tasks. Overall, while LLMs exhibit solid reasoning in structured environments, their limitations become apparent in more abstract or imprecise domains.

2.3 *Causal Models*

In this section, causal models are defined and their role in representing and analyzing cause-effect relationships between variables will be explained.

2.3.1 *Theoretical foundation of Causal Models*

Based on the ideas in Pearl's work, Causal models are structured frameworks used to represent and analyze the cause-and-effect relationships between different variables. These

models help in understanding how changes in one variable can lead to changes in another, allowing researchers to make informed inferences about causal mechanisms [25]. In simpler terms, they show how changes in one variable can directly lead to changes in another, helping to reveal the structure of the relationships within a system.
Imbens and Rubin explain that it's important to tell the difference between correlation and causation when trying to understand real-world problems. They say that causal models give a stronger base for making conclusions, especially in areas like economics, social sciences, and health research.
This means, that instead of focusing on just the relationships between variables, it allows for a more underlying insight. Researchers are able to find out why certain effects happen by using causal models to identify and analyze the underlying causes, rather than simply observing correlations.

A core foundation of Causal Models are the variables that are being used. They represent the focal system in the model, acting as a connector between the real life scenario and the abstracted model. Each variable can influence other variables in a direct or indirect way. Understanding these connections is critical for accurately modeling causation. A common method of representing these causal effects between variables is the Directed Acyclic Graph (DAG).
Explained by Feeney et al., DAGs "are a type of graph that illustrates an assumed causal structure between variables of interest" [7]. In these graphs, variables are represented as Nodes which are, depending on the causal direction, connected with unidirectional arrows.
These graphs allow for an easy to understand visualization of a proposed causation between a Cause and and a following Effect.
Directed Acyclic Graphs (DAGs) are commonly used to visualize causal models because they clearly show how different factors connect and influence each other. However, there are many other ways to represent causal relationships.

2.3.2 *Existing methods for causal model creation*

Explained by Kline, one of the most established approaches to creating causal models is called Structural Equation Modeling (SEM). In SEM, variables that represent real-world scenarios can be treated as hypothetical constructs, which helps in analyzing the relationships between observed and unobserved variables [13].
These hypotheses are then tested against actual data to assess how well the model fits. This allows researchers to explore possible causal relationships by comparing theoretical predictions to observed results.

Another important method for creating causal models is the Bayesian Network (BN). A BN is a graphical model used to represent relationships between variables using probabilities [15]. Pearl explains that in these networks, each variable is shown as a node, and causal relationships are illustrated with directed edges. Parent nodes directly influence child nodes, showing clear cause-and-effect relationships [24]. BNs are particularly useful because they combine prior knowledge and observed data, allowing researchers to understand complex interactions in uncertain environments [5].
A visual example of a BN can be seen in Figure 2. The illustration clearly demonstrates variables as nodes, with causal relationships depicted by arrows showing the direction

of influence. For instance, the variable *Season* influences both *Flu* and *Hayfever*, which subsequently affect symptoms such as *Muscle Pain* and *Congestion* [15].

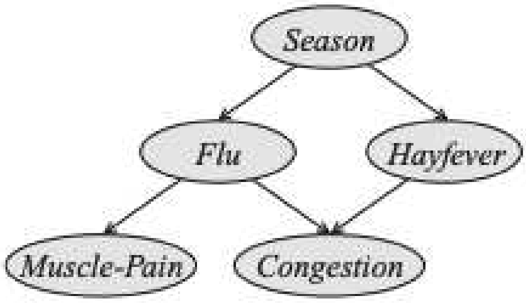

Figure 2: Example of a Bayesian Network (adapted from Koller & Friedman)

As shown, Bayesian Networks are a helpful and clear way to show how different variables affect each other. Traditional ways of figuring out causal relationships are costly because they depend heavily on expert knowledge [24]. This reliance on experts makes it important to look at alternative methods that might make causal inference more accessible without being so demanding in a cost and time perspective.

3 Methodology

This chapter explains how LLMs will be tested for creating causal models in a CPSs.
The study follows the *Design Science Research (DSR)* method, which helps find solutions to real-world problems by developing and testing new approaches [6]. Stated by Hevner, DSR can be be split into three interdependent cycles. The first cycle is called the Relevance Cycle, which has the purpose of connecting the research that is done to real life problems and actual needs [9]. The idea is to gain value in real life that is based on the research that was done.
The second cycle, the Rigor Cycle, is there to incorporate existing scientific findings into the research process. This way, the research will not be started from scratch, but built on existing knowledge.
The last cycle is the Design Cycle which brings together the other cycles. In this step, the concrete way of how the research will be performed is defined. The question of how the artifact or the result of the research will be obtained is clearly answered here.
To sum it up, synchronizing these three cycles, DSR remains both practically useful (Relevance) and scientifically sound (Rigor). Additionally, solutions will constantly be challenged and improved (Design) [9].

The main goal of this paper is to evaluate how well a Large Language Model can generate initial causal models for Cyber-Physical Systems by comparing its output against a ground truth defined by human experts. The focus lies on understanding the quality and reliability of causal models created by the LLM on its own, to evaluate its potential as a support tool in the early stages of causal model creation.
To enhance the realism of the evaluation, a Smart Charging Garage serves as the use case.

In this study, OpenAI's GPT-4 will be used as the LLM of choice to extract causal knowledge from ontology-based data.
GPT-4 has demonstrated advanced language understanding and strong reasoning capabilities. As noted, on a "suite of traditional NLP benchmarks, GPT-4 outperforms both previous large language models and most state-of-the-art systems" [1], making it a valuable tool for the purpose of the study. The data validation and feedback to or from the experts is improved by the ability to use natural language, as this enables intuitive refinements and clearer corrections.
Additionally, GPT-4 has achieved high performance in professional and academic benchmarks, including "passing a simulated bar exam with a score around the top 10% of test takers" [1].
These qualities make GPT-4 well-suited for processing structured ontology data and extracting meaningful causal relationships in Cyber-Physical Systems.

As this paper is part of a larger research project, other studies will also examine how different LLMs perform in extracting causal knowledge. These future studies will follow similar methods to compare the models and provide a better overall understanding of their strengths and weaknesses.

Causal models help us understand how different system states are connected [29], but creating them manually can be difficult due to hidden dependencies and complex relationships [4]. This methodology aims to test whether LLMs can assist in this process and how well they perform compared to human experts.

3.1 *Research Questions*

To guide the evaluation, this methodology focuses on three main aspects that build on the original research questions.
First, it explores whether a LLM can independently generate meaningful initial causal models from structured data in a Cyber-Physical System, by relying on it's general knowledge capabilites. The outputs created by the LLM are then compared to an expert-defined ground truth to assess how well the model performs in identifying relevant system elements and causal relations. This comparison focuses on both completeness and correctness, including an analysis across the causal, temporal, and topological dimension.
Finally, the study also aims to identify common challenges that occur when using LLMs for this task, such as typical errors, misunderstandings of system structure, or unclear reasoning paths, to better understand the model's limitations and where expert feedback is most needed.

3.2 *Research Approach*

Instead of relying on unstructured text, this study uses a table-based approach.
The tabular data used for the analysis was provided by an industry partner involved in smart grid operations, specifically related to a smart charging garage.
The table provides a structured map of the focal SG, the Smart Charging Garage, listing key components such as sensors, actuators, or system states in a concise format.

3.3 *Workflow for Causal Model Creation*

The methodology follows a step-by-step process to generate and refine causal models using LLMs. Below are the five key steps:

Step 1: Data Preparation
The first step is organizing the data into a structured format.
Reynolds and McDonell show that clarity and consistency in prompts can better control model outputs. Presenting data in structured, table-like formats is one way to reduce ambiguity and achieve that clarity [27].
The structured data includes:

- Sensors and their assigned states.
- Actuators and their relationships with system components.

Step 2: Data Validation
Before generating causal models, the LLM is asked to:

- Summarize its understanding of the uploaded data.
- Identify key system components and their roles.

- Suggest possible adjustments to the LLM's interpretation based on expert input.

Step 3: State Type Creation
Building on the validated data, this step focuses on defining *State Types* that link observable properties to the correct platform. These states serve as a foundation for the causal model by reflecting realistic conditions in the smart grid.
The key principles of creating these State Types are:

- **Clarity**: Each state type should distinctly relate an observable property to a valid platform, ensuring logical consistency.
- **Relevance**: The state type must add measurable value to the causal analysis, either by triggering an action or providing necessary context for decision-making.
- **Consistency**: Align each state type with actual system behavior, maintaining topological and operational accuracy.

In simple terms, State Types represent specific system conditions that help describe what is happening, where it happens, and why it matters.
After creating these State Types, they provide a strong foundation for causal inference, making each recognized system state both well-defined and relevant to real-world conditions.

Step 4: Causal Model Creation
The LLM is prompted to generate causal relationships across three key areas with the help of its general knowledge capabilities:

- **Temporal**: Does Event A happen *before* or *at the same time* as Event B?
- **Topological**: What is the relationship between these states based on system structure?
- **Causal**: Does State A *cause* State B?

Step 5: Possible Expert Adjustements
In this step, experts would typically refine or adjust the created State Types based on their domain knowledge; however, in this paper, this remains a theoretical part of the workflow.

- Experts review LLM-generated causal links to check for correctness.
- Any incorrect links are flagged and refined in an iterative feedback loop.
- The LLM is then re-prompted to improve its output based on expert feedback.

3.4 *Evaluation*

To evaluate the qualitiy of the results, precision and recall metrics will be used to assess the LLM's performance.
Recall will measure how well the LLM's State Types and relations match those defined by experts. Precision will evaluate how reasonable the LLM's outputs are, both overall and across causal, temporal, and topological dimensions. This approach allows for a structured and measurable comparison between the LLM's results and the expert-defined model.

3.5 *Evaluation Criteria*

To evaluate how well the LLM-generated causal models align with expert-created models, this chapter establishes key evaluation criteria that apply to both the identification of state types and the generation of causal relations. The evaluation will focus on two aspects: the completeness of the LLM's output and the accuracy of the generated results.

The completeness of the results will be assessed by comparing the number of state types and causal relations generated by the LLM to those defined by the expert. This will indicate whether the LLM under-produces, over-produces, or generates a comparable number of results.

The accuracy of the results will be examined by mapping the LLM-generated state types and relations to their expert-defined counterparts. This will determine how well the LLM captures key system interactions and whether its outputs align with expert knowledge.
To quantify these aspects, two evaluation metrics will be used: Recall and Precision

Recall measures how well the LLM-generated results cover the expert-defined ones and is calculated as follows:

$$\text{Recall} = \frac{\text{LLM-generated results found in expert model}}{\text{Total results in expert model}}$$

Precision assesses the reasonableness of the LLM-generated results by evaluating their validity and relevance. Each result was checked one by one by the author, using help from the LLM and other information sources to judge whether it made sense in the given CPS scenario. For every result that was classified as reasonable, a short written justification was added to explain the context and reasoning behind the classification and can be found in the data files.
Only results that do not appear in the expert-defined ground truth are considered for reasonableness. These unmatched results are evaluated individually to determine whether they still make sense in the CPS context.
Precision is calculated as follows:

$$\text{Precision} = \frac{\text{Reasonable LLM-generated results } (\textit{Matched results excluded})}{\text{Total LLM-generated results}}$$

In addition to the overall precision metric, dimension-specific precision values are introduced to evaluate the LLM's performance across causal, temporal, and topological aspects. Each dimension's precision is calculated as follows:

$$\text{Precision}_{\text{Dimension}} = \frac{\text{Reasonable LLM-generated results for Dimension}}{\text{Total LLM-generated results for Dimension}}$$

This refined metric allows for a more detailed assessment of the LLM's strengths and weaknesses in accurately identifying relations within each evaluation dimension.

To provide further structure, LLM generated results will be categorized as follows:

- Correct:
  Results that fully align with expert definitions.

- Reasonable:
  Results that are not part of the expert model but could be valid under certain conditions. These results make logical sense in the CPS context, even if they were not explicitly defined by experts.

- Incorrect:
  Results that do not logically fit within the system.

To summarize it, Recall was chosen to assess the completeness of the LLM's results, while precision highlights the model's ability to generate valid and reasonable outputs.
By applying these evaluation criteria, this analysis will quantify the quality of the LLM-generated causal models in terms of their reliability and practical usefulness.

4 Use Case: Smart Charging Garage as a Cyber-Physical System

This chapter introduces the Smart Charging Garage (SCG) as a representative Cyber-Physical System (CPS) and explains its role in evaluating the proposed hybrid workflow. The SCG serves as a real-world example where causal relationships between different system components can be explored and tested with the help of a Large Language Model (LLM).

The goal of the SCG is to manage the flow of energy in this smart grid. It aims to lower energy costs, avoid overloads, and support sustainability by using smart grid technology, where automatic decisions help manage energy flow and let the system react better to changing conditions.

4.1 *System Overview*

The focal smart grid in this thesis is a garage that also acts as an energy management system. Cars parked in the garage can be charged using electric vehicle (EV) chargers. The system continuously adjusts energy distribution based on demand, availability, and storage capacity.

On a functional level, the SCG can be divided into four main parts:

- **Energy Consumers** – Electric Vehicle (EV) chargers
- **Energy Producers** – Photovoltaic (PV) systems
- **Energy Storage** – Battery units
- **System Framework** – The building infrastructure that hosts all components

The system operates in real time and must handle constantly changing inputs, such as the number of EVs being charged or the amount of solar energy produced. A key concept in this system is the idea of an *Envelope Violation*, which occurs when certain operational limits are exceeded. For example, if the combined energy draw from all EV chargers becomes too high, it may exceed the system's safe power threshold, triggering an envelope violation.

This thesis does not focus on exact values that cause an envelope violation but stays on an abstract level to better illustrate general system behavior and causal relationships.

4.2 *Relevance of Approach*

Understanding the causes behind different system states is important in a changing environment like the Smart Charging Garage. Problems such as overloads or battery issues need to be traced back to their sources to improve system control. This thesis introduces a workflow where GPT-4 creates an initial causal model that is then compared to one made by experts. The aim is to test whether this method can support explainability and help with fault detection and energy management in similar systems. Another benefit is that when the system setup changes, the model can be adjusted more easily without starting from scratch.

The Smart Charging Garage is used as the evaluation case, helping to develop and test the proposed workflow in a realistic setting.

4.3 *Ground Truth Benchmark*

To test how well GPT-4 can extract causal knowledge, a ground truth model of the SCG has been created manually by experts. This expert-defined model outlines known causal relationships between system states. The LLM-generated outputs will later be compared against this ground truth created by domain experts to evaluate performance in terms of completeness, correctness and overall usefulness.

4.4 *LLM Prompt Configuration Framework*

In the LLM prompt, the following predefined options for relation types were included to guide the model and improve output quality:

- **Causal Relation:** "Causes" or "Enables"
- **Temporal Relation:** "Before" or "Overlaps"
- **Topological Relation:** "parentPlatform", "siblingPlatform", or "samePlatform"

These options help ensure that the generated causal models are not only logical but also structurally consistent with the Smart Charging Garage's design. They were selected based on the ground truth defined by experts, in order to narrow down the results generated by the LLM and make them more directly comparable to the expert model.

In addition, one expert defined relation was included in the prompt to guide the LLM's responses toward a desired outcome.
This relation was the Envelope Violation, which served as an example of how other results across different parts of the system should be structured and can be seen in Table 1.

StateType	**ObservableProperty**	**Platform**	**Description**
DemandEnvelopeViolation	EnvelopeViolation	Garage	Violation sensor value is above 0 or the active power of the garage is higher than the defined envelope value.

Table 1: Envelope Violation relation used to guide the LLM output

Another adaption has been made with the expert defined variable *isTriggerState*. Even though it was mentioned in the expert model, this condition was not used during the process, as it was not essential for generating or evaluating the causal relations.

5 Hybrid-Workflow Implementation

This section will present the overall workflow that combines Large Language Models (LLMs) and expert input to create a causal model for a CPS. As previously mentioned, the focal SG will serve as the benchmark. The ultimate goal is that with the help of this workflow, the existing causal relations can be found with the help of LLMs.
The mapping will happen on a rather abstract level to allow for later use in other domains, whereas the development chapter serves as the concrete instance for this paper.
The prompts were improved through trial and error until they gave results that were suitable for structured causal model evaluation in the CPS context.

5.1 *Mapping of hybrid-workflow*

5.1.1 *Step 1 - Data Input*

The starting point of the workflow can be seen as an introduction of the provided data to the LLM. A general overview of the focal SG should be provided, and the input data must be handed over to the LLM. In this step, tabular data from the SG should be inserted to establish a clear overview of all relevant properties, sensors, actuators, or other descriptions. If more data is available, it should also be included in this first prompt, enhancing the LLM's knowledge about the focal SG.

Various testing has shown that copying the tabular data into ChatGPT, along with the next prompt, creates a suitable starting point:

Prompt 1:
I have tabular data, and I would like it transformed into a descriptive natural language format where each row is expressed as a sentence. The first row contains headers, and subsequent rows contain values. The transformation should follow this pattern:
Example Input:

```
Header1    Header2
ItemA     TypeX
```

Example Result:

• ItemA is a type of TypeX.

If a cell is blank or missing, describe it appropriately. For example: *"ItemX does not have a type."*
Use your general knowledge and reasoning capabilities to decide the relation between the columns. It does not necessarily have to be "is a type of," but could also reflect a totally different relation depending on the context of the data.

Now transform the following table into the same format:

. . .
Insert your table data here [ ]
Insert a brief summary of the project, the focal smart grid, and related topics.
. . .

If there are issues, let me know.

To make the workflow better suited for the specialized needs of the SG, the first Feedback loop is implemented in the first prompt. The goal in this step is the verification, that the input data has been accurately processed and understood by the LLM. This involves asking the model to summarize the relationships within the data in clear, natural language sentences and flag any ambiguities or inconsistencies. The feedback from this step allows the experts to review the LLMs understanding of the SG. In this step, first corrections can be made to improve the foundation for the following tasks.

5.1.2 *Step 2 - State Type Creation*

In this step, the goal is the creation of possible System States that could occur in the focal SG, which will later serve as the foundation for the causal model creation.
The LLM will be prompted to generate plausible System State Types, based solely on the provided data from the first prompt, being Platforms, Sensors and observable Properties. Since smart grids vary in complexity, a one-size-fits-all prompt is not effective. Instead, a structured guidance approach is used to direct the LLM towards generating relevant and meaningful StateTypes.

Some key considerations for the creation of State types can be made:

- **Each observable property should be mapped to a possible system state, that can reflect real world conditions**
  This ensures that system behaviors are consistently represented and integrated into the causal model.

- **The state t must match the platform, that the sensor is hosted on**
  This maintains logical as well as topological consistency and prevents misalignment between system states and their components.

- **Each StateType should provide useful insights about the system.**
  All states should contribute to system understanding and decision-making.

Prompt 2:
Using the structured ontology data from the first prompt, generate additional StateTypes that describe system conditions based on observable properties and platform types. Ensure that each StateType reflects an actual system state that can be used for causal inference.
Follow this structure:

. . .
Insert a sample state t, including State type Name, observable property, platform and a short description

. . .

Now, generate additional relevant StateTypes for the Ontology. Ensure accuracy in platform association and describe each state in detail. The output format must be a table.

5.1.3 *Step 3 - Causal Model Creation*

Now that StateTypes have been defined, the next step in the workflow is to establish causal dependencies between them. These relationships help us understand how system states influence each other within the focal SG.
Once these dependencies are clearly defined, they create a structured representation of causality, explaining system behavior in a logical and interpretable way.

To maintain structural integrity and consistency throughout the process, a strict format must be followed. This format is designed to be compatible with various SGs, ensuring adaptability across different applications. It functions as a *desired output sheet*, where each column defines a specific condition or rule that the LLM must populate. Following this structure helps preserve key relationships, including time, topology, and causation rules, making the causal model adaptable to different SGs.

Dependencies are organized based on the following key considerations:

- **Cause (StateType_cause)**
  The initial system state that triggers the dependency.
- **Causal Relation**
  Defines whether the cause *enables* or *causes* the effect.
- **Temporal Relation**
  Specifies the timing of the effect in relation to the cause (e.g., *before*, *overlaps*).
- **Topological Relation**
  Defines the spatial or hierarchical relationship between states (e.g., *samePlatform*, *parentPlatform*).
- **Effect (StateType_effect)**
  The resulting state that follows from the cause.

To generate the desired results, Prompt 3 should be used.

Prompt 3:
Using the structured StateTypes from the smart grid, generate causal dependencies that describe how system states influence one another.
Each causal dependency should follow this format:
. . .
Insert a sample Causal Relation, including Causing state type, Effect state type and relations in the following dimensions:

- ***Causal Dimension***
- ***Temporal Dimension***
- ***Topological Dimension***

. . .

The options for each relation are as follows:

. . .
Insert available options for each dimension
Example: Temporal Relation can be "Before" or "Overlaps"
. . .

Now, generate additional relevant causal dependencies based on the existing StateTypes from the smart grid ontology. Both the Causing State and the Effect State must be strictly selected from the previously generated StateTypes. No new StateTypes should be created in this step. Only causal dependencies between existing StateTypes should be established.
Ensure accuracy in temporal and topological associations. If any concept is unclear, highlight uncertainties and request expert clarification to ensure correct understanding.
Also, after presenting results, you should prompt the following:

"Would you like to:
1. Confirm that these relationships are correct?
2. Modify any incorrect relationships?
3. Add missing causal dependencies?"

5.1.4 *Expert Validation*

With the initial causal dependencies generated, the following step is expert validation. This process ensures that LLM-generated relationships are accurate, logically sound, and aligned with real-world smart grid behavior.

Since causal model creation is highly complex, it is unlikely that the LLM-generated results will be perfect on the first attempt. To iteratively refine the model, this step provides a possibility, where experts identify errors and are able to suggest corrections. Through multiple feedback cycles, the causal model is improved until it accurately represents the system's behavior and interactions.

This step does not require an additional prompt, as the prompt is formed in a way that enables this feedback cycle.

The expert is able to refine the generated Causal Model by formulating improvements in natural language.

This process can be repeated until the desired output is achieved.

5.2 *Implementation of Hybrid Workflow on SCG Use Case*

This section presents a step-by-step walkthrough of the causal model extraction process. The focal smart grid will be used as an instance to showcase the LLMs capabilities in creating causal models.

To prevent a possible data spill and guarantee neutrality, the prompts have been posted in a newly created environment with no prior data logs. The Walk-Through has been done multiple times, but to ensure a clear and traceable evaluation, three instances were selected to assess overall performance.

5.2.1 *Prompt 1 - Data Input*

The first prompt has been enriched with ontology data from the focal SG as mentioned in the first Prompt.

A small sample of the included tabular data is presented in Table 3 and Table 4.

Platform	**Platform Type**	**Hosted By Platform**
EVChargerA	EVCharger	AllEVChargers1
EVChargerB	EVCharger	AllEVChargers1
Battery1	Battery	BatteryOverview

Table 2: Small sample of focal Platform, Platform Type, and Hosting Structure

Sensor	**Hosted By Platform**	**Observed Property**
AP_Garage1_Sensor	Garage1	ActivePower
EnvelopeViolation_Garage1_Sensor	Garage1	EnvelopeViolation
AP_EVChargerA_Sensor	EVChargerA	ActivePower

Table 3: Small sample of focal Sensor, Hosting Platform, and Observed Property

The LLM provided a natural language description for each platform, each sensor and the related observable properties.
For example, the *EVCharger1*-Entitity has been identified as *"a type of EVCharger"* [18]. The topological position has been assigned as well, as *EVCharger1 "is hosted by AllEVChargers1"* [18]. Additionally, the related sensor and the connected observable property have been identified the following way:
"AP_EVCharger1_Sensor is hosted by EVCharger1 and observes ActivePower" [18].

The contextual framework has been correctly identified, as the LLM was able to locate this dataset in the context of Energy Management Systems, even going as far as mentioning smart grids in once instance.
"The dataset represents a structured view of an energy system, likely within a smart grid or energy management framework" [18].

The data input did not raise errors and the LLM did not prompt for further clarification of input data.

5.2.2 *Prompt 2 - State Type Creation*

In the next step, the LLM is prompted to create StateTypes based on the provided ontology data from the first Prompt.
After execution, the LLM generated a wide range of StateTypes, all related to the provided Sensors, Platforms and observable Properties, while ensuring a certain relevance towards the causal model generation.
The StateTypes have been created to fit table format, making them better suited for comparison with the expert-created StateTypes.
Across the walkthroughs, the LLM generated between 8 and 10 different StateTypes.

One StateType was always generated across all iterations. The name of this state differs in some cases, but it is contextually always similar to *ActivePowerExcess_State* or *ActivePower_AboveThreshold_State* and represents a distinct StateType where "active power consumption exceeds a predefined threshold in any EV charger" [18].
The related property has always been identified as "ActivePower" and the platform as "EVCharger".
Table 5 represents one instance of a LLM generated result in tabular data format.

StateType	**Observable Property**	**Platform**	**Description**
Overload_State	ActivePower	EVCharger	Signals that the total active power demand from all EV chargers has exceeded the safe operational threshold.

Table 4: Representation of LLM output instance

In summary, the generated StateTypes align with the provided entities from the first step, ensuring that each one represents a clear and structured system condition. Their definitions follow a logical format that reflects the general behavior of the system. The LLM's general knowledge contributes additional descriptive details, making each state easier to interpret and relate to real-world scenarios. This structured presentation also allows for easier review and potential refinement by domain experts. Chapter 6 will take a closer look at how these generated results compare to expert-created StateTypes.

5.2.3 *Prompt 3 - Causal Model Creation*

In order to suit the focal SG, the third prompt has been enhanced with concrete options for each possible relation. In the SCG, the options are the following across all dimensions:

- Causal Dimension:
  Causes or *Enables*
- Temporal Dimension:
  Before or *Overlaps*
- Topological Dimension:
  parentPlatform or *samePlatform*

This individualization improves correctness of results, as the LLM does not generate values for these options randomly, but rather chooses the best fitting one from given options.

When executing the third prompt multiple times, between 7 and 10 instances of causal dependencies were created, all in the dimensions of causal, temporal and topological relation, as well as the predefined options for each dimension.

One example achievement is, that across all three chat-instances, the Battery Overcharge State appears as a recurring causal factor.
In Chat 1, overcharging leads to an Operational Envelope Violation, indicating that the battery's excess charge impacts the overall system. Chat 2 suggests that overcharging results in Battery Low Efficiency, emphasizing the degradation of the battery performance. Meanwhile, Chat 3 connects overcharging to Battery Balancing. This could mean that the system tries to redistribute excess energy across multiple batteries.
Despite these differences, the underlying causal mechanism remains the same: Battery Overcharge disrupts normal system operations, either through safety risks, efficiency loss, or the need for balancing measures.

Chat	**Cause (StateType_cause)**	**Causal Relation**	**Temporal Relation**	**Topological Relation**	**Effect (StateType_effect)**
C1	BatteryOvercharge_State	Causes	Overlaps	samePlatform	OperationalEnvelopeViolation_State
C2	BatteryOverCharge_State	Causes	Overlaps	samePlatform	BatteryLowEfficiency_State
C3	BatteryOvercharge_State	Causes	Before	samePlatform	BatteryBalancing_State

Table 5: Causal Relation of Battery Overcharge Across Different Chats

The LLM takes the given statetypes and turns them into causal relations by figuring out how system conditions affect each other. It usually picks cause states that show problems like high power demand, inefficiencies, or system overloads, while the effect states represent the system's reaction, like balancing power, reducing efficiency, or triggering a violation. This means the LLM builds cause-and-effect links based on how energy flows through the system and when certain limits are crossed.
For temporal relations, the LLM mostly uses "Overlaps" when two states happen at the same time, like an overcharging battery affecting efficiency while it's still charging. It applies "Before" when one state clearly leads to another, like a power deficiency happening first and then causing a demand violation later.

In topological relations, the model organizes states based on their system connections. It assigns "samePlatform" when both states belong to the same component, like a battery's charge level affecting its efficiency.

Overall, the LLM takes the statetypes and builds structured causal models by linking them through cause-and-effect, timing, and system structure, creating logical relationships that explain how different system states interact.

6 Performance evaluation

In this chapter, the LLM-generated causal models will be compared to the expert-created ones to see how well the LLM performs. The focus will be on how accurately the LLM identified cause-and-effect relationships, as well as how it handled timing and system structure. By looking at the similarities and differences, this evaluation will show where the LLM performs well and where refinement in results is still necessary.
In detail, three rounds of Workflow walkthroughs were conducted, producing three separate instances of results. As expected, due to the nature of LLMs, the outputs vary slightly across iterations.
These three result sets now serve as the benchmark for evaluation. They will be compared against the expert model to assess performance.

6.1 *Analysis of results and Comparison to traditional models*

6.1.1 *Prompt 1 - Data Input*

The data input process was completed successfully, with all platforms, sensors, and observable properties correctly recognized and categorized. No data was lost, misclassified, or incorrectly assigned, ensuring that the system had a solid and accurate foundation. Because everything was set up correctly, the next step, generating the State Types, could proceed without any issues. In this case, no further expert-refinements or adjustments were necessary, making the input phase smooth and fully reliable.

6.1.2 *Prompt 2 - State Type Creation*

The next prompt is responsible for generating state types that serve as the foundation of the causal model creation. The expert-defined model includes 11 state types, while the LLM-generated results range between 8 and 10 state types that were suggested across three walkthroughs. While this indicates that the LLM consistently produced fewer results, it still managed to capture a portion of the expert-defined state types. Specifically, 5 of the expert-defined state types were successfully identified by the LLM in each instance, resulting in an average recall rate of 63%, 50% and 50% across all walkthroughs. This demonstrates that the LLM captured on average 54% of the expert-defined state types, suggesting potential gaps in identifying core system states.
Table 6 summarizes the number of State Types identified.

In addition to evaluating recall, the results were also assessed for their reasonableness. The LLM-generated state types that were not part of the expert model were analyzed to determine whether they could still be considered logical and meaningful. Across the three

	Expert Model	**LLM (1st)**	**LLM (2nd)**	**LLM (3rd)**
Total State Types	11	8	10	10
Correct State Types	-	5	5	5
Recall (%)	-	63%	50%	50%
Reasonable State Types (GT excl.)	-	3	5	5
Reasonable State Types (GT incl.)	-	8	10	10
Precision (GT excluded) (%)	-	38%	50%	50%
Precision (GT included) (%)	-	100%	100%	100%

Table 6: Comparison of LLM-generated and expert StateTypes with precision and recall metrics

walkthroughs, the number of reasonable state types, including both correct and additional ones, was 8, 10, and 10. When excluding the correct matches, the number of additional reasonable state types was 3, 5, and 5. This variation shows that the LLM was able to generate extra state types that may not be in the expert model but still appear valid. However, these state types were only classified as reasonable after a detailed individual assessment.
The Precision score, calculated by looking at the number of reasonable state types in relation to all generated ones, yielded values of 38%, 50%, and 50% across the three walkthroughs. When including all reasonable results, precision rises to 100% in each case. These values show that while the LLM's overall output was limited in quantity, it still produced only meaningful state types, whether matching the expert model or not.

The LLM's ability to generate additional state types beyond the expert-defined ones underlines its exploratory potential. While all of these suggestions were considered reasonable and represented plausible system behavior, they still required individual review. This exploratory nature has both benefits and risks: the additional state types could reveal overlooked aspects of the system and increase model completeness, but expert validation remains necessary. Overall, these results show that LLMs can provide creative and helpful suggestions that may support a deeper understanding of complex CPS behavior.

6.1.3 *Prompt 3 - Causal Model Creation*

This phase of the evaluation focused on analyzing the relations generated by the LLM in comparison to the expert-defined model. This evaluation required an additional matching step to align LLM-generated StateTypes with their corresponding expert-defined counterparts. This was necessary to ensure that relations between matching StateTypes could be fairly compared.

Table 7 presents an overview of the number of relations generated by the expert and the LLM models:

	Expert Model	**LLM (1st)**	**LLM (2nd)**	**LLM (3rd)**
Total Relations	10	7	10	9
Relations Found from Expert Model	-	1	1	0
Recall (%)	-	14%	10%	0%

Table 7: Comparison of the number of relations identified by the expert and LLM models

The expert model contained 10 relations, while the LLM model generated between 7 and 10 relations across its three instances.
Although the total number of relations was similar, the low recall values (14%, 10%, and 0%) indicate that only a small fraction of the expert-defined relations were correctly identified by the LLM, meaning the correct Causal Statetype, as well as the correct Effect Statetype has been identified. A relation is only counted as a match if both the cause and effect State Types are present among the State Types defined in Step 2. This suggests that the LLM struggled to reconstruct the expert's structured understanding of causal interactions.

In analyzing the identified relations, one key observation is that only one causal relation was correctly identified across the three walkthroughs. In both cases (Chat 1 and Chat 2), the LLM correctly established a causal connection involving high charging load and overload conditions. However, the details of these connections reveal some nuances worth exploring.

In Chat 1, the identified relation connects the state "EVChargingHighLoad_State" to "Overload_State". This causal link was correctly identified, aligning with the expert model. The expert-defined relation originally describes that excessive EV charging activity may result in system overload due to high power demand. The LLM's result correctly reflects this dependency, suggesting that the model was able to infer the connection based on observable behavior patterns. Despite the correct causal identification, the LLM mismatched some structural details, notably misclassifying the topological relationship. While the expert model identified this as a parent-platform dependency, the LLM classified it as a same-platform relation. This suggests that although the LLM understood the causal dependency, it oversimplified the platform structure, potentially reflecting its limited understanding of the system's hierarchy.

In Chat 2, a slightly different but still correct relation was identified, connecting "EVChargerOverload_State" to "EVChargingCongestion_State." This causal relationship is similarly accurate, as excessive charger load can lead to congestion in EV systems. The LLM correctly matched the causal and temporal dimensions of this relation. However, like in Chat 1, the topological relation was incorrect, classified again as a same-platform relation instead of the parent-platform structure defined in the expert model.

Both cases reveal that while the LLM showed some success in recognizing meaningful causal links, there was a recurring issue with topological misclassification. This pattern suggests that the LLM may rely more heavily on the semantic similarity of state types rather than correctly interpreting their structural roles in the system. Consequently, while the causal relations generated by the LLM provide valuable insights, they often require further refinement, especially when it comes to understanding hierarchical dependencies.

Beyond relation matching, an additional step was taken to evaluate the reasonableness of the LLM-generated relations. Even if a relation was not present in the expert model, it was examined to assess whether it was logically sound or plausible. Table 8 shows the number of reasonable relations and the corresponding precision values.

	Expert Model	**LLM (1st)**	**LLM (2nd)**	**LLM (3rd)**
Reasonable Relations	10	4	6	4
Precision (%)	-	57%	60%	44%

Table 8: Evaluation of reasonableness and precision in the generated relations

The LLM models showed varying precision values (57%, 60%, and 44%) across iterations. This indicates that on average 54% of the LLM-generated relations made logical sense, even if they did not directly match the expert-defined model. Some of the newly generated relations introduced by the LLM, while reasonable, reflected alternate system interpretations.

The evaluation revealed differing patterns in the distribution of causal, temporal, and topological relations.The LLM demonstrated strong performance in identifying causal and topological relations, with precision values averaging 85% , and 88% respectively. However, the LLM showed improved accuracy in detecting temporal relations, achieving the highest precision average of 89%. This suggests that while the LLM effectively identifies straightforward cause-effect relationships and topological dependencies, it performs even better in identifying temporal dependencies that align with the GT.
Despite these positive outcomes, the results also indicate some inconsistency across the three walkthroughs. While causal and topological precision remained relatively stable, temporal precision fluctuated significantly in each iteration, ranging from 80% to 100%. Even though the highest average Precision has been achieved in this dimension, the variability highlights the LLM's inconsistent ability to recognize temporal structures, reinforcing the need for expert oversight when assessing timing-based dependencies.
In addition to evaluating overall precision, a more detailed assessment was conducted to measure precision across the three evaluated dimensions: causal, temporal, and topological relations. This dimension-specific evaluation provides further insights into the LLM's strengths and weaknesses in generating meaningful relations.

Table 9 summarizes the precision values for each dimension across three walkthroughs:
The results show that causal precision values ranged from 80% to 89%, with an average of 85%. Temporal precision varied between 80% and 100%, achieving the highest consistency with an average of 89%. Lastly, topological precision exhibited values between

	LLM (1st)	**LLM (2nd)**	**LLM (3rd)**	**Average**
Precision Causal	86%	80%	89%	85%
Precision Temporal	86%	80%	100%	89%
Precision Topological	86%	90%	89%	88%

Table 9: Precision values across causal, temporal, and topological dimensions

86% and 90%, resulting in an average of 88%.

These results indicate that while the LLM performs consistently across all three dimensions, its average accuracy in identifying temporal relations was slightly higher than in the other dimensions. The observed stability in causal and topological precision suggests that the LLM demonstrated a reasonably strong understanding of cause-effect dependencies and spatial relationships within the system, whereas it is limited within the Temporal category.

However, the variation in precision scores further emphasizes the LLM's inconsistent performance and highlights the importance of expert validation to ensure model reliability. The recurring misclassification of topological relations highlights a significant limitation in the LLM's understanding of system structure. In multiple instances, the LLM incorrectly assigned same-platform dependencies where the expert model defined parent-platform links. This repeated error suggests that the LLM relies more heavily on surface-level patterns than on accurately recognizing hierarchical dependencies. Such misclassifications may result in an oversimplified model structure, potentially obscuring critical control hierarchies. Addressing this limitation is essential for improving the LLM's reliability in future implementations.

In conclusion, the LLM-generated relations provide a valuable exploratory starting point but remain incomplete in reconstructing the expert-defined structure. While some newly introduced relations appear reasonable and could provide insights into alternate system behaviors, expert review and refinement are essential to ensure that the resulting model truly reflects real-world behavior. The LLM's ability to capture known relations remains limited, highlighting the need for improved guidance or iterative adjustments in future applications.

6.2 *Summary of Findings*

The evaluation of the LLM's performance reveals mixed results, highlighting both its strengths and weaknesses in generating causal models.
On the positive side, the LLM successfully identified some expert-defined state types and managed to create a number of reasonable new state types. This shows that there is the potential to identify key system elements and even introduce ideas that may have been overlooked by experts or could otherwise be useful.
However, the ability to fully capture the expert-defined model was limited, with recall rates showing that less than half of the expected state types were identified. This indicates that the LLM struggles to provide a complete and reliable overview of system behaviors.

Additionally, the quality of these results was inconsistent, with precision scores fluctuating significantly across different walkthroughs. While some results were reasonable and helpful, others lacked relevance or correctness, showing that the LLM's performance is not stable.

The LLM was even less reliable when it came to generating causal relations. Only a small fraction of expert-defined relations were correctly identified, revealing significant gaps in the LLM's understanding of the system's structure.
Despite this, the LLM occasionally produced reasonable new relations that were not part of the expert model, demonstrating some creative potential. This exploratory behavior suggests that the LLM can provide new insights, but these ideas require expert evaluation to ensure they make sense.
A recurring issue throughout the evaluation was the LLM's difficulty in understanding the system's structural dependencies. It frequently misclassified key relationships, especially when distinguishing between parent-platform and same-platform dependencies. This weakness reduces the reliability of the LLM's output and limits its usefulness without careful expert review.

With expert interaction, these issues are avoidable. Expert guidance can help refine the LLM's outputs, correct misclassifications, and ensure that generated state types and relations align more closely with real-world system behaviors.

In conclusion, the LLM showed potential in suggesting new ideas and highlighting possible system dynamics, but its overall performance was unreliable. Its inconsistent results and tendency to overlook key system components make it challenging to rely on without expert oversight.
Future improvements may enhance the LLM's ability to deliver more stable and comprehensive results, but for now, expert intervention remains crucial to ensure accuracy and reliability.

7 Discussion

7.1 *Interpretation of Key Findings*

The key findings of this research have been addressed in detail in the previous chapter. Overall, the LLM demonstrated some capability in identifying expert-defined state types and generating reasonable new system insights. However, it struggled with consistency, accuracy, and understanding system structure. While its exploratory nature provided some value, the LLM's incomplete results highlight the need for expert involvement to ensure reliability.

7.2 *Practical Implications*

The findings of this study offer important insights for both practitioners and researchers. Practitioners working with Cyber-Physical Systems (CPS) may find value in using LLMs as a tool to accelerate the early stages of causal model creation. The LLM can serve as a brainstorming tool, generating initial ideas that experts can refine and validate. This may reduce the time and knowledge required to build initial causal models, especially in complex CPS environments where system dynamics are challenging to analyze. However, given the LLM's inconsistent precision and recall rates, expert supervision is essential. By guiding the model, practitioners can improve its reliability, correct misclassifications, and ensure that generated models align with real-world behavior.

For researchers, these findings reveal valuable insights for improving hybrid workflows that combine LLM outputs with expert intervention. The LLM's creative yet unstable results highlight the need for improved integration techniques. Future research can focus on developing refined prompting strategies that guide the LLM toward producing more accurate and structured causal models. Additionally, designing clear frameworks for expert feedback will be crucial in enhancing the hybrid workflow's effectiveness. By improving these processes, researchers can create more reliable methods for integrating LLMs into CPS model development.

7.3 *Limitations*

While this research offers useful insights, several limitations should be acknowledged. First, the LLM's performance was evaluated using a limited number of walkthroughs, which may not fully represent its potential capabilities or weaknesses.
Additionally, the evaluation criteria being focused on precision and recall, may overlook other important aspects of causal model quality, such as model interpretability or complexity.
Lastly, the expert intervention process itself was not deeply examined, meaning further investigation into optimal expert-LLM collaboration strategies is necessary. Even though in theory experts could have improved the model, the evaluation was performed upon a model that was not ajdusted by experts, but relied solely on the LLM outputs.

7.4 *Recommendations for Future Work*

To address the identified limitations and improve the LLM's performance in causal model creation for CPSs, several future research directions are recommended:

- Refining LLM prompts to improve accuracy and ensure outputs better align with expert expectations
- Developing structured frameworks that enhance expert-LLM collaboration, enabling them to efficiently guide and refine the LLM's outputs
- Increasing the amount of walkthroughs to increase the sample size
- Performing a similar performance assessment of causal model extraction with other LLMs

8 Conclusion

This thesis set out to explore the potential of LLMs in supporting causal model creation within CPSs. Using a Design Science Research approach, the study combined theoretical insights with practical experimentation to assess whether LLMs can effectively assist experts in building initial causal models. The findings provide important insights into both the capabilities and limitations of LLMs, along with practical recommendations for improving their performance.

CPS environments are increasingly complex, creating a growing demand for improved explainability solutions. Existing methods for causal model creation often require extensive expert knowledge, making them time-consuming and costly.

LLMs were presented as a potential solution to provide initial model insights, reducing the reliance on expert knowledge as well as improving accessibility. While LLMs possess promising reasoning capabilities, their reliability in the structured context of CPS remained uncertain.

To evaluate the LLM's potential, a hybrid workflow was designed that integrates LLM-generated suggestions with expert validation. By structuring the process into distinct phases, the data input, state type creation, and causal model generation, the workflow aimed to balance the exploratory potential of LLMs with the necessary expertise to ensure accuracy. A Smart Charging Garage served the use case, providing a practical environment to assess the LLM's performance.

The study addressed the following research questions:

- **Research Question 1**:
  To what extent can Large Language Models (LLMs) assist experts in providing the initial causal model within a Cyber-Physical System (CPS)?

  The evaluation revealed that while the LLM demonstrated some capability in identifying expert-defined state types and generating new insights, its overall performance was inconsistent. Recall rates showed that less than half of the expert-defined state types were successfully identified. Allthough the LLM occasionally produced creative insights, its unstable output quality limits its independent application. Nevertheless, the LLM's ability to produce reasonable new insights that could be classified as useful, emphasizes its potential as a valuable support tool in the early phases of model development.

- **Research Question 1.1**:
  How do the causal models generated by LLMs compare to those created by human experts in terms of quality and applicability in real-world CPS scenarios?

  The LLM was not able to fully reconstruct the expert-defined model, but it demonstrated the ability to generate reasonable and relevant state types and relations that reflected plausible system behavior. Although many expert-defined elements were missed, the appearance of new, meaningful suggestions points to the LLM's potential to contribute creative ideas. These ideas, while not always accurate, could support and enrich expert-driven modeling when reviewed and refined through expert input.

- **Research Question 1.2**:
  What are potential challenges when relying on LLMs for causal model creation?

  The study identified significant challenges in precision and consistency. The LLM often misclassified state types and incorrectly linked causal dependencies. Despite this, the model occasionally produced new insights that experts could build upon. The findings underscore the importance of expert oversight to filter and improve the LLM's outputs.

The LLM's exploratory behavior introduces both risks and opportunities. While its creative outputs may reveal overlooked system behaviors, expert guidance remains crucial to correct errors and refine misclassified relations. For practitioners, this means that LLMs can be useful in accelerating early causal model development but should not yet be relied upon without expert supervision.
For researchers, the study highlights the need to improve hybrid workflows, refine LLM prompting strategies, and develop clearer frameworks for integrating expert feedback.

This thesis concludes that while LLMs show promise in supporting causal model generation, their current limitations prevent them from fully replacing expert-driven methods. However, with refined prompting techniques, improved expert interaction processes, and enhanced model understanding of CPS structures, LLMs may become a valuable tool in the future. By continuing to explore these improvements, future research can further bridge the gap between LLM capabilities and expert-driven causal modeling, ultimately enhancing the development of scalable and efficient solutions for CPS explainability.

9 Access to Data Files

If you wish to access the complete evaluation data, including detailed results and additional insights, please contact the author via email.

Author: Maurer, Paul
Email: h12124850@s.wu.ac.at
Orcid-ID: https://orcid.org/0009-0003-7008-027X

References

[1] Josh Achiam et al. “Gpt-4 technical report”. In: *arXiv preprint arXiv:2303.08774* (2023).

[2] Radhakisan Baheti and Helen Gill. “Cyber-physical systems”. In: *The impact of control technology* 12.1 (2011), pp. 161–166.

[3] Ignacio Calleja and Luis Delgado. “European environmental technologies action plan (ETAP)”. In: *Journal of Cleaner Production* 16.1, Supplement 1 (2008). Diffusion of cleaner technologies: Modeling, case studies and policy, S181–S183. ISSN: 0959-6526. DOI: `https://doi.org/10.1016/j.jclepro.2007.10.005`. URL: `https://www.sciencedirect.com/science/article/pii/S095965260700220X`.

[4] Haoyue Dai, Peter Spirtes, and Kun Zhang. “Independence testing-based approach to causal discovery under measurement error and linear non-gaussian models”. In: *Advances in Neural Information Processing Systems* 35 (2022), pp. 27524–27536.

[5] Adnan Darwiche. *Modeling and Reasoning with Bayesian Networks*. Cambridge, UK: Cambridge University Press, 2009.

[6] Aline Dresch et al. *Design science research*. Springer, 2015.

[7] Timothy Feeney, Fernando Pires Hartwig, and Neil M Davies. “How to use directed acyclic graphs: guide for clinical researchers”. In: *bmj* 388 (2025).

[8] Constanze Fetting. “The European green deal”. In: *ESDN Report, December* 2.9 (2020).

[9] Alan R Hevner. “A three cycle view of design science research”. In: *Scandinavian journal of information systems* 19.2 (2007), p. 4.

[10] Fei Hu et al. “Robust cyber–physical systems: Concept, models, and implementation”. In: *Future generation computer systems* 56 (2016), pp. 449–475.

[11] Amitkumar Vidyakant Jha et al. “Smart grid cyber-physical systems: Communication technologies, standards and challenges”. In: *Wireless Networks* 27.4 (2021), pp. 2595–2613.

[12] Emre Kıcıman et al. “Causal reasoning and large language models: Opening a new frontier for causality”. In: *arXiv preprint arXiv:2305.00050* (2023).

[13] Rex B. Kline. *Principles and Practice of Structural Equation Modeling*. Fourth. Methodology in the Social Sciences. New York, NY: The Guilford Press, 2016. ISBN: 978-1-4625-2334-4.

[14] A-R Kojonsaari and J Palm. “The development of social science research on smart grids: a semi-structured literature review”. In: *Energy, Sustainability and Society* 13.1 (2023), p. 1.

[15] Daphne Koller and Nir Friedman. *Probabilistic Graphical Models: Principles and Techniques*. Cambridge, MA, USA: MIT Press, 2009.

[16] Edward Ashford Lee and Sanjit Arunkumar Seshia. *Introduction to embedded systems: A cyber-physical systems approach*. MIT press, 2017.

[17] Yang Liu et al. “Review on cyber-physical systems”. In: *IEEE/CAA Journal of Automatica Sinica* 4.1 (2017), pp. 27–40.

[18] Paul Maurer. *Markdown Collection of LLM Results*. 2025.

[19] Humza Naveed et al. “A comprehensive overview of large language models”. In: *arXiv preprint arXiv:2307.06435* (2023).

[20] Gabriel Nicholas and Aliya Bhatia. “Lost in translation: large language models in non-English content analysis”. In: *arXiv preprint arXiv:2306.07377* (2023).

[21] Giuliano Andrea Pagani and Marco Aiello. “The power grid as a complex network: a survey”. In: *Physica A: Statistical Mechanics and its Applications* 392.11 (2013), pp. 2688–2700.

[22] Rajvardhan Patil and Venkat Gudivada. “A review of current trends, techniques, and challenges in large language models (llms)”. In: *Applied Sciences* 14.5 (2024), p. 2074.

[23] Judea Pearl. “Causal inference in statistics: An overview”. In: *Statistics Surveys* 3 (2009), pp. 96–146. ISSN: 1935-7516. DOI: 10.1214/09-SS057. URL: http://dx.doi.org/10.1214/09-SS057.

[24] Judea Pearl. *Causality: Models, Reasoning, and Inference.* 2nd. Cambridge, UK: Cambridge University Press, 2009.

[25] Judea Pearl. “The seven tools of causal inference, with reflections on machine learning”. In: *Communications of the ACM* 62.3 (2019), pp. 54–60.

[26] Kasaraneni Purna Prakash et al. “Comprehensive Bibliometric Analysis on Smart Grids: Key Concepts and Research Trends”. In: *Electricity* 5.1 (2024), pp. 75–92. ISSN: 2673-4826. DOI: 10.3390/electricity5010005. URL: https://www.mdpi.com/2673-4826/5/1/5.

[27] Laria Reynolds and Kyle McDonell. “Prompt programming for large language models: Beyond the few-shot paradigm”. In: *Extended abstracts of the 2021 CHI conference on human factors in computing systems.* 2021, pp. 1–7.

[28] Katrin Schreiberhuber et al. “Towards a State Explanation Framework in Cyber-Physical Systems”. In: *ACM SIGENERGY Energy Informatics Review* 4.4 (2025), pp. 142–154.

[29] Galit Shmueli. “To explain or to predict?” In: (2010).

[30] Maxim Vidgof, Stefan Bachhofner, and Jan Mendling. “Large language models for business process management: Opportunities and challenges”. In: *International Conference on Business Process Management.* Springer. 2023, pp. 107–123.

[31] Xijia Zhang et al. “Explaining agent behavior with large language models”. In: *arXiv preprint arXiv:2309.10346* (2023).