Large Language Model-assisted Clustering and Concept Identification of Engineering Design Data

Felix Lanfermann, Thiago Rios, Stefan Menzel

Honda Research Institute Europe, Offenbach, Germany

***Abstract*—Recent advances in Large Language Models (LLM) open up opportunities for users to interact with domain specific knowledge and execute (semi-)professional tasks in a dialog fashion. Without profound knowledge in data science and programming languages, basic statistics and further detailed analyses can be conducted intuitively through natural language prompts. Accessing common data science methods, LLMs can assist users in visualizing, interpreting, and analyzing large and complex data sets. In this paper, we study the potential of ChatGPT from the perspective of different user expertise levels to process and interpret existing data sets which originate from engineering optimization tasks with varying complexity—–a proportional-integral-controller configuration, an aerodynamic design optimization, and an energy management task—–and use it to cluster the data into meaningful groups. Furthermore, we provide the LLM with a recently developed concept identification metric and assess its capability to use the metric for an allocation of the samples into technically reasonable concepts. On one hand, the formulation of conceptual machine learning pipelines, as well as using well-established methods, such as typical clustering algorithms, are conducted seamlessly by ChatGPT for the given tasks. On the other hand, it is observed that automatically constructing a concept identification algorithm from its mathematical formulation using ChatGPT still requires supervision and support from a domain expert.**

***Index Terms*—Large language models, generative AI, data analysis, clustering, concept identification, engineering application.**

I. Introduction

Generative Artificial Intelligence (GenAI) and Large Language Models (LLM) create a paradigm shift in various business fields by supporting processes and product development, utilizing content generation, dialog systems, software implementation and creative tasks, among others. Being able to interpret and reason over several data modalities, combining different domain knowledge, the models offers a huge potential for method development and application. Another valuable aspect is the change in the way on how to communicate with computer systems and how to access domain knowledge. In industry, development teams typically focus on various subdomains, *e.g.*, creative design, structural optimization, or aerodynamic efficiency in automotive, where domain experts generate and evaluate components using specialized tool-sets and cooperate with colleagues to innovate a new product. LLMs such as GPT-4 [1] or Llama 2 [2] democratize this process. They provide access to domain-specific knowledge for non-experts through user-friendly natural language interfaces and foster interactions for solving problems together. However, due to the generative character of the models, one needs to consider potential hallucination effects [3] producing non-factual statements, which may a major obstacle when numerical accuracy is required, *e.g.*, in safety-critical systems. Another important aspect is consistency of the generated content, *i.e.*, achieving similar results based on similar inputs.

One major field which profits from the LLM progress and democratization of domain knowledge is data science [4], [5], where LLMs can support processes in the analysis steps such as data cleaning and data organization, or provide customized implementations of machine learning models. In this paper, we evaluate and discuss the potentials and limits of data science in an engineering context. There are two overarching questions that we address within this work for engineering application: (1) What is the impact of LLMs' recent advances for data science? (2) Can LLMs increase the accessibility of data science for non-experts? To answer both questions, we test whether an LLM can give pivotal support for a typical data science procedure. We assess if an LLM can process, visualize and interpret engineering data in a way that is useful to the user. We investigate whether basic statistics are calculated correctly and if a machine learning procedure can be carried out by the LLM directly. In a series of experiments, we imitate several levels of data science expertise. We test if the LLM can split the given data set into reasonable groups by applying (a) a typical clustering method and (b) a recently developed concept identification method [6]. For the second method, we provide the LLM with the scientific description of the algorithm and implementation details. Both methods identify groups of similar samples based on different preconditions. In engineering design tasks, such groups provide several benefits [7], such as illustrating correlations and dependencies of the features, as well as forming sets of alternative design options and representative prototypes for further investigation. Based on the results of the experiments, we show that an LLM can be utilized for data analysis and conduct typical procedures, such as statistical investigations and simple clustering methods. We further illustrate the current limitations for the application of more complex machine learning algorithms, such as deriving algorithms and code from mathematical descriptions, and for which tasks (human) expert supervision is still necessary.

Section II discusses related work. Section III provides an overview on the utilized methods and Section IV describes the conducted experiments. Section V discusses our general findings and Section VI concludes the work.

979-8-3503-5409-6/24/$31.00 ©2024 IEEE
DOI 10.1109/CAI59869.2024.00150

II. Related work

Recently, data analysis methods, which are available through LLMs, have been more and more explored on various tasks. In [4], the authors studied the capabilities of GPT-4 as a data analyst on benchmark data, concluding that GPT-4 can outperform a junior and achieve similar results as a senior data analyst. Also, the usability and interfacing improved, *e.g.*, in ChatGPT by first being available through the integration of Third Party plugins, such as Wolfram, then through code interpreter [8] and as of today through direct prompting in ChatGPT. There exist different ways for users to interact with data. Users can upload and explore data sets directly within the ChatGPT interface or by requesting software code, which afterwards can be locally executed, *e.g.*, in python environments. However, despite the rapid development and improved user interaction with data sets, the capabilities of ChatGPT need to be further evaluated to understand the applicability in specific expert domains. Due to the diversity of technical fields and required expert knowledge, these assessments typically include human observations through domain experts. In [9], the authors evaluate the effectiveness of LLMs (GPT-3.5, GPT-4) and prompt engineering techniques for data analysis automation in bioinformatics tasks. By coupling LLMs with R packages, experiments have been carried out to assess the capabilities to pre-process, visualize, and apply machine learning and statistics on data sets. Through human assessment, Jansen et al. conclude that while GPT-4 shows improved performance compared to GPT-3.5, their system cannot compete with a domain expert. In addition, complexifying the tasks reduced the success of executable code which could also not been compensated by prompt engineering, *i.e.* providing more user guidance in the prompts. Using ChatGPT in the automotive domain, Lingo [10] studied the potential of synthesizing data in telematics tasks. Advantages include the opportunity to generate data samples in the size requested by the user containing also extreme conditions and preserve data privacy. In [11], Lingo discussed different ways on how non-expert users can interact through ChatGPT with data, including pre-processing, basic analysis and post-processing steps and concluded that ChatGPT is capable to synthesize (telematics) data through prompts. However practical relevance of such data still needs to be verified by human experts due to the lack of benchmark data. Further advances on utilizing LLMs to analyze and procecute data in engineering tasks include text-to-3D models for design optimization [12], a vision language model for a wide spectrum of design tasks [13], as well as concurrent work on assisted surrogate modelling for optimization tasks [14].

In this paper, we explore typical measures to allocate data samples into similar sets as another typical data science technique. The processing steps we are assessing are twofold. On one hand, we consider clustering due to its frequent application in data science for sorting data into groups. Typical clustering algorithms, such as k-means [15], allocate samples to clusters based on the provided full feature vector. Recent work, such as [16], [17] has shown, that LLMs can improve text clustering performance, and we want to investigate the capabilities in the engineering data domain. On the other hand, we consider concept identification [6]—a recently proposed data allocation methodology that identifies groups of data that are similar with respect to a predefined set of description spaces. Here, the identified concepts represent groups of samples that are consistently non-overlapping in subspaces of the full feature vector. In particular for many engineering applications, these description spaces, such as the space of all defining parameters or all performance criteria, share a semantic context. Identified concepts that are unique, non-overlapping, and coherent in such a semantic context provide a large benefit for the engineer. Related clustering techniques, such as subspace clustering [18], which can identify clusters in lower-dimensional projections of high-dimensional sparse data, however, cannot predetermine the projections. Multi-view clustering [19] utilizes multiple concurrent "views", *i.e.*, representation of the same data, to exploit complementary information. This technique cannot assure consistency of the found clusters across the separate views. This inability to maintain consistency of data concepts for a fixed set of projections also applies to many other related approaches, such as Gaussian Mixture Models [20], density-based clustering methods DBSCAN [21], reduction techniques like UMAP [22], information maximization clustering [23], or hierarchical clustering algorithms. Therefore, in case consistency of groups of data across a-priori defined subspaces is required, concept identification is most suitable.

III. Methods

A. Conducting data science experiments using a large language model

When evaluating LLMs on their capabilities in application fields such as engineering data science, it is important to keep in mind that LLMs have learnt a large text corpus and can likely reproduce and apply different methods from literature. As such, it is already an achievement to produce results with similar or the same accuracy, since the advantage is in the level of domain knowledge required to access the methods. With the lack of defined benchmark tasks, we as humans and domain experts need to explore and understand the quality of results, identify non-factual statements or biases towards over-represented data.

However, recent advances of LLMs offer capabilities to support a user in the analysis of data. Starting with a raw data set, a typical analysis procedure involves cleaning the initial data, calculating basic statistics, as well as visualizing and interpreting the data (Fig. 1). This is usually followed by a processing step, *e.g.*, the application of a machine learning method to the data to solve a given task. Finally, a post-processing step or interpretation of the results concludes the procedure. The capabilities of ChatGPT we want to address and evaluate within the scope of this work include an initial visualization and interpretation of a raw data set, as well as the calculation of basic statistics. We further test if and how well the model can apply standard clustering methods and if

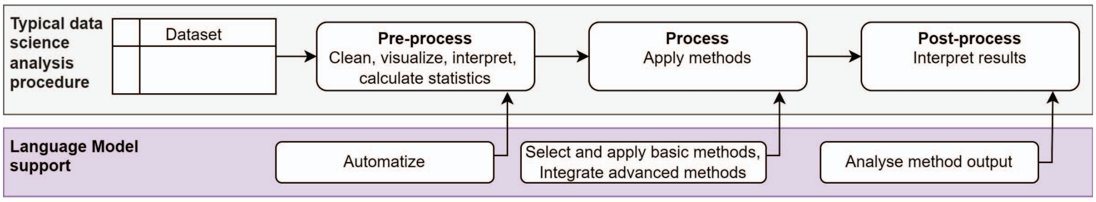

Fig. 1: Typical procedure of data analysis and the evaluated support by ChatGPT

a more advanced concept identification method can be carried out, based on the description of the method from its original publication. In a last step, we investigate if the method output can be analyzed and interpreted by the LLMs, and to which degree the interpretation is meaningful to the user.

The different tasks are conducted using ChatGPT 4 (denoted as ChatGPT in the remainder of the paper) from the perspective of three levels of data science expertise. The first considers a user without prior data science knowledge, the second an intermediate level of experience, and the third level accounts for expert knowledge.

B. Description of the investigated data sets

First, we consider a simple control systems design case. The abstract system to be controlled is a standard mechanical mass-spring-damper model which connects a mass $m = 1\,\mathrm{kg}$ via a parallel configuration of a spring $k = 100\,\mathrm{N/m}$ and damper $b = 20\,\mathrm{Ns/m}$ to an infinitely larger second mass, *i.e.*, a "wall" (Fig. 3). Gravitation is not considered. The control task implies applying the force $F_{\mathrm{c}}(t)$ to the mass m to reach a desired position $x_{\mathrm{d}} = 1\,\mathrm{m}$. The system can be modelled by the differential equation

$$m\ddot{x} = kx + b\dot{x} - F_{\mathrm{c}}(t), \tag{1}$$

where x, $\dot{x}$, and $\ddot{x}$ represent the lateral position, velocity, and acceleration, respectively, of the mass. The force applied to the mass is controlled by a proportional-integral controller (PI-controller)

$$F_{\mathrm{c}}(t) = Px_{\mathrm{e}}(t) + I\int_t x_{\mathrm{e}}(t), \quad \text{with} \quad x_{\mathrm{e}}(t) = x(t) - x_{\mathrm{d}} \tag{2}$$

describing the control error over time. The scalar parameters P and I refer to the proportional and integral gain of the controller that need to be specified in order to control the mass. The problem, defined by equations 1 and 2, is a well-understood problem in the control design domain, exhibiting known patterns for the control parameters and objectives.

The control case is modelled in SimulationX[1] and simulated for a time span of $5\,\mathrm{s}$. A data set is automatically created by simulating a large amount of parameter combinations in a multi-objective optimization. Each configuration is evaluated with respect to two performance criteria, the overshoot $x_{\max}$, *i.e.*, the mass' maximum position value over time, and the time value $t_{\pm 5\%}$, describing the point in time, after which the difference of the trajectory to the desired position is less than $5\,\%$, *i.e.*, $|x_{\mathrm{e}}(t)| \leq 0.05$. In total, the data set consists of roughly 1000 samples, each described by 4 features, *i.e.*, the control parameters P and I, as well as the performance indicators $x_{\max}$ and $t_{\pm 5\%}$ (Table I, Fig. 3). Although describing a very realistic configuration task, the data set can (by engineering standards) still be considered simple, due to the limited number of features. It nevertheless serves the purposes of the proposed evaluation task well, as it contains the typical control parameters and objectives that an engineer uses as decision basis for the given scenario.

We further consider two more complex engineering data sets from industry-procedural engineering tasks. Both sets are significantly more complex than the PI-controller case, both, in terms of overall size, and more importantly, also in terms of number and correlation of features. One set contains 2500 airfoil designs from an optimization study [6]. The samples are given as the combination of their defining parameters, descriptive geometric features, as well as corresponding performance metrics for various boundary conditions. The other set contains more than $20\,000$ energy management configuration options resulting from a many-objective optimization task [24]. In this set, each sample is represented by nine different decision variables and ten partially conflicting objectives. We consider these three data sets because they have been extensively studied and evaluated in the context of concept identification in previous work [6], [25], [26], which allows for a simplified evaluation and comparison of the results.

TABLE I: Data set overview.

#sample number	P	I	overshoot	t5%
0	127.0	641.0	0.0000	0.281
1	140.0	673.0	0.0000	0.256
...	...	...	...	...
905	162.0	746.0	0.0000	0.215
906	175.0	846.0	0.0244	0.185

IV. Experiments and Results

The experiments are conducted in a set of dialogs[2], having different levels of expert knowledge in mind (Fig. 2). The following section analyzes typical examples and highlights the relevant differences for the respective levels of expertise. The prompts and answers are quoted in full or in part for each task A-H and enumerated separately for each level of expertise 1, 2, 3, *i.e.*, the prompt A.3 refers to the expert level 3 dialog

[1] www.esi-group.com

[2] The dialogs have been repeated multiple times and similar prompts have been tested. The plots are adapted with respect to figure size, format, font size and style, but the content is not modified.

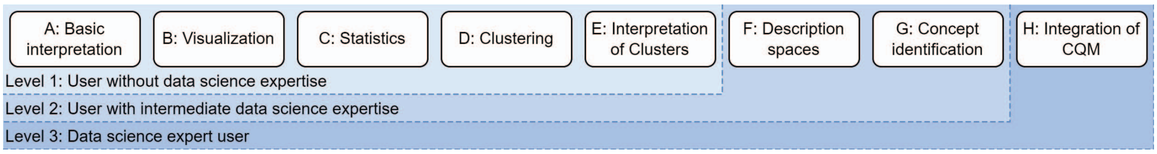

Fig. 2: Evaluated data analysis tasks and the dialog level for which they are considered.

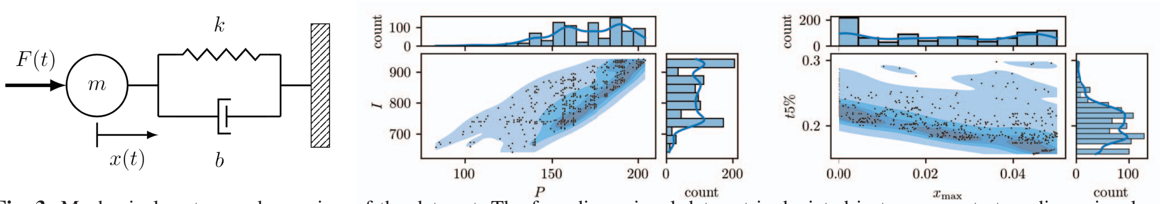

Fig. 3: Mechanical system and overview of the data set: The four-dimensional data set is depicted in two separate two-dimensional spaces, one containing the parameters (P and I), the other containing the performance criteria $x_{\max}$ and $t_{\pm 5\%}$.

regarding the basic interpretation A of the data. If the prompt is taken from a sequence of prompts for a task, it is followed by a subscript, *e.g.*, A.3_1.

A. Basic interpretation

For the basic interpretation of the data set, we provide ChatGPT with a text file (comma-separated values), representing the set of configuration-performance pairs that have been obtained from the optimization. The results for all three levels of expertise are similar. For example, for the expert level prompt

> ***Prompt A.3$_1$:*** < `controllers.csv` > Interpret this data set

we receive a general description of the included content:

> ***Answer A.3$_1$:*** $[\cdots]$ Given the column names, it seems this dataset is related to control systems, specifically PID (Proportional-Integral-Derivative) controllers, which are commonly used in industrial control systems for regulating temperature, speed, pressure, and other variables. $[\cdots]$

Without further description, ChatGPT correctly infers that the data set originates from the PID-controller design domain. The variables P and I are correctly identified as the proportional and integral controller parameters, respectively. Further, overshoot, the extent to which a signal or system exceeds its target value, is labeled correctly. The variable $t_{\pm 5\%}$ is correctly identified as a time-based metric that describes the performance of the system, however incorrectly attributed to the time the system takes to reach $5\,\%$ of the target value. Only for the expert level dialog, we explicitly correct this assumption to improve the context for the following interaction:

> ***Prompt A.3$_2$:*** $t_{\pm 5\%}$ refers to the time taken to reach and stay within 95% of the target value

B. Visualization of the data set

The request to produce a visualization of the data set is handled rather differently depending on the level of detail introduced in the prompt. The basic inquiry

> ***Prompt B.1:*** Visualize the data

leads to four two-dimensional scatter plots, each one featuring the sample number on the x-axis, as well as one of the respective four variables. The sample number itself does not provide any information about the data set, such as optimization progress. It can rather be considered to be a random index, hence the information gain from the plots is low. Neither can the distributions of the variables be easily obtained, nor do the plots provide information about the correlation between them. This can be achieved using a precise prompt, such as

> ***Prompt B.3:*** Create 3 plots. The first should contain the control parameters. The second and third should the overshoot and $t5\%$ as a histogram, respectively.

C. Statistics

For all dialog levels, when asked to calculate statistics of the data, *e.g.*, with the basic request to

> ***Prompt C.2:*** Calculate statistics of the data

ChatGPT provides a list presentation of the results obtained from `pandas.DataFrame.decribe()` and a brief statement about statistical measures:

> ***Answer C.2:*** $[\cdots]$ The mean and standard deviation are particularly useful for understanding the central tendency and variability of each parameter $[\cdots]$

The provided results are typical and correct, however the atypical list-form does not provide a benefit over a table-form output, which an experienced user would expect.

D. Clustering

Identifying groups within the data is a task that depends on the exact prompt. Simply asking to

> ***Prompt D.1$_1$:*** Group the data

or to

Prompt D.1$_2$: Group the data based on ranges of P and I values, as well as ranges of overshoot and $t_{\pm 5\%}$

leads to an attribution of all samples to the categories "Low", "Medium", "High", and "Very high" for each individual feature based on percentiles. However, it is not specified if those individual groups for each respective feature share commonalities or correspond to each other in any way. Prompting to *find* groups in the data, on the other hand, *e.g.*,

Prompt D.1$_3$: Can you find groups in the data?

triggers a response to use clustering algorithms:

Answer D.1: To find groups or clusters in the data, we can use clustering algorithms such as K-Means or Hierarchical Clustering. These methods will help to identify groups based on similarities in the data points. $[\cdots]$

When a specific number of desired clusters is provided, k-means is applied to the full four-dimensional data set. ChatGPT further identifies the necessity to standardize the data in order to avoid bias in the clustering algorithms, as well as proposes the elbow method for cases without a desired number of groups. However, when asked to

Prompt D.3: Cluster the data into three distinct groups

k-means clustering is used directly, without scaling the data first and without visualizing the result[3].

E. Interpretation of clusters

Simply inquiring

Prompt E.1: What do the groups mean?

leads to an interpretation that is understood as

Answer E.1: $[\cdots]$ examining the characteristics of each cluster. This usually involves looking at the central tendencies (like the mean or median) of the features within each cluster. $[\cdots]$

However, looking at the central tendencies of the clusters is problematic, since it does not capture the entire ranges of the distributions. The interpretation of the clusters is given as:

- Cluster 0: "Configurations in this cluster have lower 'P' and 'I' values, minimal overshoot, and slower response times (higher t5%)"
- Cluster 1: "This cluster represents configurations with moderately high 'P' and 'I' values, moderate overshoot, and faster response times (lower t5%)"
- Cluster 2: "This cluster signifies configurations with the highest 'P' and 'I' values, the highest overshoot, and the quickest response times (lowest t5%)"

When inspecting the clusters, the completeness of the given interpretation is doubtful. While the interpretations are not generally wrong, they clearly do not capture the complexity of the identified clusters, one example being the wide range of overshoot values for cluster 0, which is overly simplified in the interpretation ("minimal overshoot"). When explicitly asking to

Prompt E.3: Interpret the clusters in the 3 spaces I described above, *e.g.* P vs I, Overshoot, $t_{\pm 5\%}$,

i.e., when the spaces of interest from an expert view are emphasized, ChatGPT describes the found clusters with respect to their system response time and stabilization properties.

Answer E.3:
- Cluster 0 could be a strategy that leads to higher overshoot and longer stabilization times.
- Cluster 1 might represent a more conservative approach, leading to lower overshoots and a range of stabilization times.
- Cluster 2 appears to be a more aggressive strategy, characterized by higher P values and mixed responses in overshoot and stabilization time.

Also in this case, the interpretation is over-simplified, due to referencing only the central tendencies of the clusters.

F. Visualization of the clusters in the description spaces

When simply prompting to

Prompt F.1: Visualize the groups

the basic level visualization of the identified clusters is given as three two-dimensional figures, each one featuring the proportional gain P on the x-axis and the other three variables on the y-axis, respectively. There is no explanation for why, in particular, the second and third space are chosen for the representation. The prompt

Prompt F.2: Show three spaces. P and I in one, overshoot in another and $t_{\pm 5\%}$ in the third

provides three non-ideal visualization of the requested spaces, using the cluster indicator as the x-axis in a scatter plot. A better option is to combine a scatter plot with histograms (Fig. 4), which can be achieved when specifically asking for it:

Prompt F.3: Visualize the clusters in 3 plots. The first should contain the control parameters. The second and third should be the overshoot and $t5\%$ as a histogram, respectively.

G. Concept identification

Trying to identify concepts in the data, *i.e.*, groups of samples that are similar and consistent with respect to the given description spaces, the overlap between the groups needs to be avoided. When asking ChatGPT to

Prompt G.2: Group the data in a way that the groups do not overlap in the three spaces,

[3] We chose three groups based on experience with the corresponding data set in previous work [25].

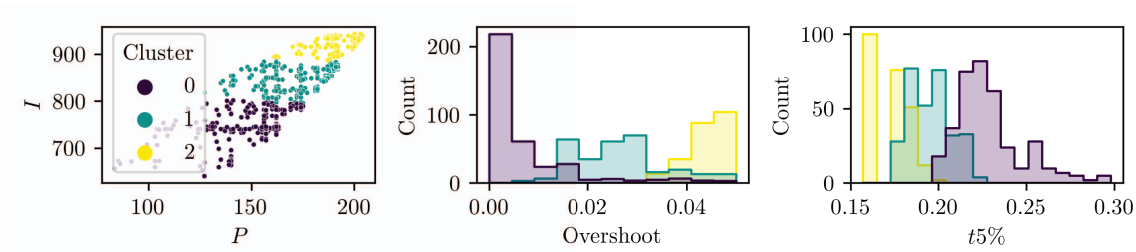

Fig. 4: Identified clusters for the expert dialog, split into the three description spaces.

it correctly states that k-means is not a suitable algorithm:

> ***Answer G.2:*** $[\cdots]$ Since K-means clustering primarily focuses on minimizing the within-cluster variance, it might not always result in non-overlapping clusters, especially in spaces where the clusters are not clearly separable. $[\cdots]$

In the expert setting, we first define the three spaces as description spaces. We then ask to

> ***Prompt G.3$_1$:*** Find three clusters that do not overlap in the three description spaces.

and receive a problematic result:

> ***Answer G.3$_1$:*** To find three clusters that do not overlap in the three description spaces (P vs I, Overshoot, $t_{\pm 5\%}$), we can apply clustering algorithms that consider all these factors simultaneously. The goal is to identify clusters where each cluster is distinct across all three description spaces. I'll use a clustering algorithm, such as k-means, on the combined features of P, I, Overshoot, and $t_{\pm 5\%}$ and then examine the resulting clusters to see if they are distinct in each of the description spaces. $[\cdots]$ From these visualizations, it appears that the clusters are somewhat distinct in each description space. $[\cdots]$

Aside from being a generally vague statement, it is also incorrect. The previous clustering procedure and visualization are repeated, leading to the same result as given in Fig. 4. Hence, in the second and third space, the clusters are clearly overlapping. We therefore correct this statement in the dialog and ask for alternative options, *i.e.*,

> ***Prompt G.3$_2$:*** No, the clusters are overlapping in the three spaces. What algorithm can I use to find non-overlapping clusters in the three spaces?

The recommended algorithms (including Hierarchical Clustering, Density-Based Clustering, Gaussian Mixture Models, Spectral Clustering) are, however, also not particularly suitable to solve the problem, as mentioned in Section II.

H. Integration of concept quality metric

In the expert level dialog, we further test whether we can integrate a recently developed metric that evaluates the quality of identified groups in a data set with regard to consistency across multiple spaces:

> ***Prompt H.3$_1$:*** Can you use the concept quality metric from the paper *Concept Identification for Complex Engineering Datasets*" by Lanfermann et. al?

The summary of the publication content is correct:

> ***Answer H.3$_1$:*** The concept quality metric proposed by Lanfermann et al. in their paper "Concept Identification for Complex Engineering Datasets" provides a method for assessing the distinctiveness of clusters in a multi-dimensional dataset. This approach focuses on evaluating how well-separated and internally cohesive the clusters are, which is particularly useful in engineering and scientific datasets where clear differentiation of concepts (clusters) is crucial. $[\cdots]$

However, the recommended steps to implement the approach (cluster the data, calculate the concept quality metric, iterate with different parameters, evaluate and select the best clustering) are not correct. This would involve optimizing connected regions for all concepts for all description spaces using the quality metric as an evaluation function.

Providing ChatGPT with the full text version of the corresponding section from the respective publication produces a better result. For this, the correct steps are identified (define the description spaces, define concepts and candidates, calculate the CQM, optimization, integration of user preferences). Nevertheless, when asked to produce python code for the evaluation of concept quality based on the publication, the implementation exhibits flaws:

Answer H.3$_2$: $[\cdots]$

```
1 for space in description_spaces:
2   in_ellipse = cdist(space,
3     [concept['center']]).flatten()
4     < concept['radius']
5   fraction = np.sum(in_ellipse
6     & concept['mask']) / np.sum(in_ellipse)
7   cqm_components.append(fraction)
```

In particular, the fraction component is calculated incorrectly in lines 5 and 6, as it divides by the number of samples belonging to each ellipse per space. Dividing by the number of samples per concept, *i.e.*, the samples that consistently belong to the corresponding ellipses in all description spaces, would be correct.

Given that the evaluation function is provided by the user code-ready, ChatGPT is able to create a concise optimization framework around it:

Prompt H.3*$_3$*: Can you write a code framework in python for the optimization loop? Use CMA-ES as the optimization strategy, assume that the concept quality metric is given as the output of an existing method (`evaluate_quality`).

In this way, a co-generation of the envisioned pipeline is possible, where ChatGPT provides the structural framework around user-defined functions.

I. Airfoil data set investigation

The method can be applied in more complex engineering tasks. Using a similar dialog as for the PI-control systems optimization task, we evaluate if it is possible to achieve comparable results as [6] for an airfoil design case study. Here, the shape of an airfoil is modified in a multi-objective optimization and evaluated with respect to its aerodynamic performance under multiple conditions. Four design parameters p_0, p_1, p_2, p_3 can be modified by the optimization framework to adapt an RAE2822 airfoil base profile. Additional features, given as the camber line displacement $c_{.03}$, $c_{.2}$, $c_{.4}$, $c_{.6}$, $c_{.8}$ at five points of interest are collected for each sample. The aerodynamic performance is evaluated for each sample using the computational fluid dynamics solver OpenFOAM[4], for three different angles of attack ($\alpha_1 = 0°$, $\alpha_2 = 1°$, $\alpha_3 = 3°$) to reflect characteristic operating conditions such as cruise flight, landing or take-off (see [6] for full details). Hence, each sample is associated with three lift and three drag coefficients. The full data set thus contains roughly 2500 samples with 15 features each. In this use case, five different description spaces are of interest to the engineer: the space of parameters, the space of geometric features, and the three spaces of performance for the respective boundary conditions. We use the same dialog structure as detailed in subsections A-H above to assess the resulting data set of the parameter-objective combinations and identify several aerodynamic concepts (Fig. 5). The k-means clustering algorithm sorts the data into three groups that are distinguishable in all description spaces. Particularly in the three performance spaces, we observe three conceptually different groups: Cluster 1 demonstrates a high performance (high lift and low drag) for the largest angle of attack and can thus be interpreted as well-suited for take-off. Cluster 2, on the other hand, achieves better performance for cruise-flight conditions (lowest drag for $\alpha = 0°$), while cluster 0 offers an intermediate performance between the two extremes.

J. Energy management data set investigation

Analogously, we investigate a data set of energy management configuration options obtained through a many-objective optimization [24]. Each sample represents a configuration defined by nine different parameters concerning the electrical demand and supply system of the research facility. Several scalable components, such as a photovoltaic (PV) system, a stationary battery and a combined heat and power plant are considered. For example, the optimization may influence the

[4]www.openfoam.org

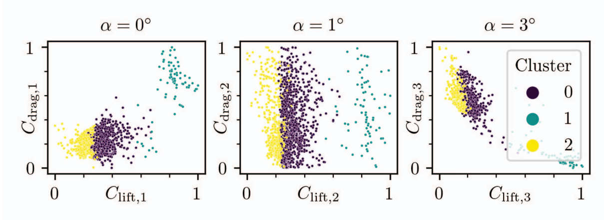

Fig. 5: Identified airfoil clusters. The clusters represent airfoil designs that are similar with respect to their parameters, geometric features, as well as the drag and lift coefficients (all normalized).

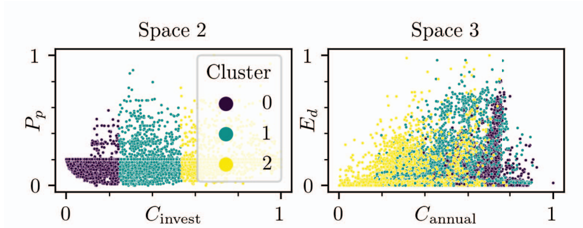

Fig. 6: Identified energy management configuration clusters. The clusters represent configuration options that are similar with respect to the full set of parameters and performance metrics (all normalized). The clusters strongly overlap in the description space of E_{d} and C_{annual}.

overall PV size and orientation, as well as battery capacity and operation settings. The system is evaluated with respect to ten partially conflicting performance metrics, including investment and operational cost, CO_2 emission, and factors that represent battery aging and electrical stress for the supply grid. Due to the large number of objectives, the data set contains more than $20\,000$ pareto-optimal configuration options. Since a manual assessment of this vast amount of variations is impractical, sorting the set automatically into semantically meaningful groups is highly desirable. To enable the systems to unravel reasonable trade-off options, we choose the same description spaces as [26]: Space 1: Peak power of the photovoltaic system P_{PV}, nominal battery capacity C_{B}. Space 2: Investment costs C_{invest}, maximum peak power P_{P}. Space 3: Annual operation costs C_{annual}, yearly discharged energy E_{d}. Space 4: Mean battery state of charge $\bar{b}$, yearly feed-in energy E_{f}. Using the described dialog structure as detailed in subsections A-H above, we are quickly able to identify clusters of configuration options which are similar with respect to the eight parameters and objectives. However, simply applying k-means clustering leads to an undesirable result. While the found clusters are distinguishable in space 2, this is not the case, for example, for space 3, where the three clusters strongly overlap (Fig. 6). No groups of configurations are found that provide a trade-off between annual operation cost and the discharged energy, while being distinguishable in the other description spaces. The basic clustering approach is thus not able to identify semantically reasonable concepts based on the given description spaces. But this can be achieved when a concept identification procedure is employed using the framework provided by ChatGPT as a response to prompt H.3$_3$.

V. Discussion

ChatGPT is, without a doubt, a practical tool for the analysis of data. Its profound capabilities to conduct typical statistical investigations and typical machine learning procedures can be of great help to engineers, data scientists, and researchers. As shown in the experiments, ChatGPT derives correct statistical metrics from a given data set and seamlessly provides code to apply basic clustering approaches. But as with any other tool, one also needs to know its limitations. However, a scientific evaluation of the capabilities and limitations of ChatGPT is challenging for several reasons. Slightly differently formulated prompts—in some cases even identical prompts—lead to different results and hinder the reproducibility of experiments. This effect alone is worth a full investigation, however out of scope of this work. A deeper analysis of a simple data science procedure would also be a worthwhile investigation, but within our work, we explicitly want to identify and address limitations for an expert setting. Another limitation is the fact that the internal python execution environment of ChatGPT is unstable, *e.g., opening (larger) data sets and creating figures frequently malfunctions.* Furhter, the experiments exhibit a tendency to visualize data as two-dimensional (scatter) plots, when not otherwise stated, which is not always the most sensible choice. Similarly, the recommendation to scale or normalize the data prior to the processing is inconsistent and often not realized in the provided python implementation. ChatGPT sometimes provides non-factual statements about the analysis and interpretation results. In several answers, the model argues that the identified groups are non-overlapping, when in fact, they actually are.

While the suggested python code for basic tasks, such as conducting a clustering process using a pre-implemented function from a standard library, can often be used without any modification (or even run by ChatGPT directly), direct method implementations seem to be more problematic. Often, the produced code did neither capture the envisioned method correctly, nor was is executable without error. Nevertheless, the capabilities of ChatGPT with regard to creating a structured framework and sketching an effective machine learning pipeline is astonishing.

VI. Conclusion

In this paper, we show that data science tasks like summarizing and visualizing data sets, as well as conducting well-established machine learning approaches, such as many clustering approaches, can be seamlessly conducted using ChatGPT, and the results are equivalent to an expert approach. For the investigated technical fields, ChatGPT integrates basic domain knowledge into the analysis and provides reasonable interpretations for identified data clusters. For drafting algorithms from mathematical formulations, expert support is, nevertheless, required to assure a correct implementation. In those cases, ChatGPT is able to provide support for the creation of a code framework. We thus believe that ChatGPT can further serve an educational purpose by illustrating how to create scientific code efficiently and introduce unknown functions and libraries even to experienced developers. In summary, we hence conclude that LLMs, and in particular ChatGPT are valuable tools for data science.

References

[1] OpenAI, "Gpt-4 technical report," *arXiv:2303.08774*, 2023.

[2] H. Touvron et al, "Llama 2: Open foundation and fine-tuned chat models," *arXiv:2307.09288*, 2023.

[3] N. McKenna, T. Li, L. Cheng, M. J. Hosseini, M. Johnson, and M. Steedman, "Sources of hallucination by large language models on inference tasks," 2023, arxiv:2305.14552.

[4] L. Cheng, X. Li, and L. Bing, "Is gpt-4 a good data analyst?" *arXiv:2305.15038*, 2023.

[5] X. Tu, J. Zou, W. J. Su, and L. Zhang, "What should data science education do with large language models?" *arXiv:2307.02792*, 2023.

[6] F. Lanfermann and S. Schmitt, "Concept identification for complex engineering datasets," *Advanced Engineering Informatics*, 2022.

[7] D. T. Pham and A. A. Afify, "Clustering techniques and their applications in engineering," *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 2007.

[8] E. Davis and S. Aaronson, "Testing gpt-4 with wolfram alpha and code interpreter plug-ins on math and science problems," *arXiv:2308.05713*, 2023.

[9] J. A. Jansen, A. Manukyan, N. A. Khoury, and A. Akalin, "Leveraging large language models for data analysis automation," *bioRxiv*, 2023.

[10] R. Lingo, "Exploring the potential of ai-generated synthetic datasets: A case study on telematics data with chatgpt," *arXiv:2306.13700*, 2023.

[11] ——, "The role of chatgpt in democratizing data science: An exploration of ai-facilitated data analysis in telematics," *arXiv:2308.02045*, 2023.

[12] T. Rios, S. Menzel, and B. Sendhoff, "Large language and text-to-3d models for engineering design optimization," in *IEEE Symposium Series on Computational Intelligence*. IEEE, December 2023.

[13] C. Picard, K. M. Edwards, A. C. Doris, B. Man, G. Giannone, M. F. Alam, and F. Ahmed, "From concept to manufacturing: Evaluating vision-language models for engineering design," *arXiv:2311.12668*, 2023.

[14] T. Rios, F. Lanfermann, and S. Menzel, "Large language model-assisted surrogate modelling for engineering optimization," *IEEE Conference on Artificial Intelligence*, 2024.

[15] J. MacQueen, "Some methods for classification and analysis of multivariate observations," *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability*, 1967.

[16] V. Viswanathan, K. Gashteovski, C. Lawrence, T. Wu, and G. Neubig, "Large language models enable few-shot clustering," 2023.

[17] Y. Zhang, Z. Wang, and J. Shang, "Clusterllm: Large language models as a guide for text clustering," 2023.

[18] L. Parsons, E. Haque, and H. Liu, "Subspace clustering for high dimensional data," *ACM SIGKDD Explorations Newsletter*, 2004.

[19] S. Bickel and T. Scheffer, "Multi-View Clustering," in *Fourth IEEE International Conference on Data Mining (ICDM'04)*. IEEE, 2004.

[20] C. E. Rasmussen, "The infinite Gaussian mixture model," *Advances in Neural Information Processing Systems*, 2000.

[21] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*. AAAI Press, 1996.

[22] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," *arXiv:1802.03426*, 2018.

[23] F. Ntelemis, Y. Jin, and S. A. Thomas, "Information maximization clustering via multi-view self-labelling," *Knowledge-Based Systems*, 2022.

[24] Q. Liu, F. Lanfermann, T. Rodemann, M. Olhofer, and Y. Jin, "Surrogate-assisted many-objective optimization of building energy management," *IEEE Computational Intelligence Magazine*, 2023.

[25] F. Lanfermann, "Concept identification for complex data sets," Ph.D. dissertation, Universität Bielefeld, 2023.

[26] F. Lanfermann, Q. Liu, Y. Jin, and S. Schmitt, "Identification of energy management configuration concepts from a set of pareto-optimal solutions," *Energy Conversion and Management: X*, 2024.