Honda Research Institute EU
Honda Research Institute Europe GmbH
https://www.honda-ri.de/

Large Language Model-assisted Surrogate Modelling for Engineering Optimization

Thiago de Jesus de Araujo Rios, Felix Lanfermann, Stefan Menzel

2024

Preprint:

This is an accepted article published in IEEE Conference on Artificial Intelligence. The final authenticated version is available online at: https://doi.org/10.1109/CAI59869.2024.00151 Copyright 2024 IEEE

Large Language Model-assisted Surrogate Modelling for Engineering Optimization

Thiago Rios
Honda Research Institute Europe
Offenbach am Main, Germany
thiago.rios@honda-ri.de

Felix Lanfermann
Honda Research Institute Europe
Offenbach am Main, Germany
felix.lanfermann@honda-ri.de

Stefan Menzel
Honda Research Institute Europe
Offenbach am Main, Germany
stefan.menzel@honda-ri.de

***Abstract*—Engineers often utilize data-driven surrogate models in optimization to partially replace the costly computational simulations of physics-based models. However, different aspects affect the accuracy and prediction capability of the surrogate models, *e.g.*, the dimensionality of the data and nonlinearities in the mapping between input and output. Learning well-performing models by selecting appropriate techniques to fit the data benefits from the machine learning and data science expertise of the engineer which may vary depending on the application domain. Recently, large language models (LLMs) have shown promising capabilities to support humans through natural language-based interfaces in approaching technical problems, as well as to co-develop software and to democratize domain knowledge. In this paper, we utilize ChatGPT 4 to co-develop a framework to select and train surrogate models for engineering optimization tasks. More specifically, we interact with ChatGPT to outline a process and software to support the selection and application of regression techniques based on characteristics of the available data and target application. We evaluate the developed methodology on synthetic and realistic engineering optimization data and problems. In our experiments, we demonstrate that the models obtained through the methodology developed with ChatGPT achieve comparable performance in regression and optimization tasks than observed in existing works in the literature. Hence, despite some limitations, such as missing updates of available software libraries, LLMs can support less experienced engineers to solve surrogate-assisted optimization problems more efficiently by providing insights on the application data and software for deploying surrogate models.**

***Index Terms*—Large language models, surrogate modeling, design optimization, generative AI, democratization.**

I. Introduction

Generative artificial intelligence (GenAI) and large language models (LLMs) currently have a major impact on a variety of business fields, *e.g.*, dealing with dialog-based systems [1], software development and refactoring [2], content generation through text-to-image [3] and text-to-shape models [4], [5], and numerical optimization [6], [7]. Due to the memory effect, LLMs allow users to provide contextual information and personalize the generated content. The analysis of the performance of LLMs is central to understand their capabilities and identify limitations, and is currently researched especially in the field of natural language processing (NLP) [8]. However, there is also the need to evaluate the performance of LLMs in application fields where user expertise and experience play a major role for optimal results. In the present paper, we utilize an LLM, namely ChatGPT 4 (short: ChatGPT), to assist a user in a typical machine learning task in the engineering context: We explore the capability of ChatGPT to assist in the selection and training of feasible data-driven models for performing surrogate-assisted optimization.

During product development, *e.g.*, in the automotive industry, computational engineering design optimization is a key for realizing fast time-to-market cycles. To reduce the number of costly evaluations with physical prototypes, engineers utilize digital computer-aided engineering (CAE) models of the physical system to perform simulations and estimate the performance of designs, *e.g.*, aerodynamic analysis for drag minimization or structural analysis for vehicle safety. When combined with computational optimization, these CAE models are iteratively updated and simulated, which often requires higher computational effort than other components of the optimization process [9]. Therefore, different approaches have been proposed to balance computational resources and optimization performance for different problems, with surrogate-assisted optimization [10] as prominent representative. Here, data-driven regression models are often selected as surrogates for computer simulations in engineering optimization. In the recent years, a wide portfolio of surrogate models has been developed, which are typically mathematical functions with known formulations and capable to be coupled with various state-of-the-art optimizers, such as gradient-based, as well as stochastic search algorithms, *e.g.*, evolutionary optimization. However, the application of surrogate models is still challenging, as it requires machine learning and domain expertise for (1) preparing and pre-processing the data, (2) selecting a reasonable model adequate for the given problem at hand, (3) learning and validating the model on training and test data, and (4) preparing the model for downstream applications such as integration into the optimization process flow.

In the present paper, we propose to use ChatGPT in the role of a data science expert to assist human users to select and process surrogate models for engineering data in design optimization scenarios. Our objectives are to (1) evaluate the process for generating surrogate models through user

©2024 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

dialog interaction with ChatGPT and (2) validate the model performance on typical benchmark functions available from literature as well as in the context of engineering optimization tasks. Hence, we first generate a process and software to select a regression technique for a generic data set by interacting with ChatGPT and compare the suggested techniques to approaches available in the literature and to standard data analysis practices. Secondly, we validate our process by applying the generated software on two types of data sets: A set of analytical functions based on [11], and the OSU-Honda automobile hood data set (CarHoods10k) [12], [13], which comprises geometric and structural data of over 10k realistic car hood designs.

Our main contribution is to demonstrate that LLMs can generate a feasible engineering and data analysis process in a cooperative fashion with the human user. Our method unveils new ways for engineers to interact with machine learning methods to address data analysis tasks, which has the potential to accelerate development processes leading to novel product designs. Therefore, we strongly believe that reasoning about complex data and tasks in a discussion-style with LLMs offers a new way of democratizing knowledge in engineering and data science, particularly if the availability and quality of LLMs increases.

The remainder of this paper is structured as follows: In Section II, we present a survey of recent works on surrogate-assisted optimization and applications of LLMs. In Section III, we provide details on our interaction with ChatGPT to develop the method for selecting and fitting surrogate models for different applications. Then, in Section IV, we present the experimental settings for verifying the proposed method using the benchmark and engineering data sets. In Section V, we discuss the results obtained in the data regression and optimization tasks. Finally, in Section VI, we conclude the paper and highlight future reseacrh directions.

II. Related Work

In surrogate-assisted optimization, domain experts utilize data-driven models to estimate the performance of a system when the mathematical model of the system is unknown or the computation of the system, *e.g.*, CAE simulation of vehicles and their components, is prohibitively expensive [14]. As black-box models, surrogate approaches have been successfully applied together with evolutionary optimization methods for both, single and multi-objective problems, as well as for problems of static and dynamic nature [15]. Although different aspects of surrogate-assisted optimization are explored in literature, a canonical methodology for selecting and fitting surrogate modeling techniques is still missing, which is a challenging task particularly for novice users of optimization methods [16].

Chatterjee *et al.* [17] surveyed different surrogate modeling techniques applied to robust design optimization. The authors also included the application of surrogate modeling in realistic design problems, *e.g.*, crashworthiness of car structures, but focused on the performance of a set of surrogate models on the selected examples rather than a general approach to select a particular technique. Salem and Tomaso [18] proposed the penalized predictive score (PPS) metric for comparing the performance of surrogate models. The authors also utilized evolutionary algorithms to search and tune surrogate models based on the PPS for specific applications. As often observed in data-driven methods, the approach underperforms in cases with limited data availability, and only yields surrogate models from a predefined set of techniques.

Towards an automatic selection of surrogate models, Saini *et al.* [19] developed a proof-of-concept system for automatizing the selection of surrogate models based on the features of the data landscape and available information about the utilized data sampling method. The authors utilized a classifier to map the data features to a set of standard surrogate modeling techniques, which was trained on data available from benchmark data sets, and achieved promising results on known benchmark optimization problems. However, despite the advances in the automation of the decision process, the understanding of the decision process still requires advanced expertise in data science, which can reduce the trust of the user on the system.

Recently introduced LLMs, *e.g.*, ChatGPT[1], have the potential to support the users to create more explainable systems by democratizing the knowledge on data analysis and engineering topics [20]. Makatura *et al.* [21] provide an extensive analysis on the capabilities of ChatGPT in different 3D design tasks considering manufacturing constraints. The authors point out that the interaction with the LLM has potential to boost the creativity of designers and speed up the design process but, at the current state of the available tools, manual work is still required to verify and fine tune the generated designs. Picard *et al.* [22] performed a similar analysis on four classes of design tasks, which include engineering education. The authors show that ChatGPT answers questions about general engineering knowledge better than solving detailed design problems, where ChatGPT lacks precision and tends to take overly cautious decisions. Hence, as LLMs are well capable to provide assistance to outline processes in a dicussion-style, we utilize ChatGPT to outline a framework for selecting and testing regression functions for surrogate-assisted engineering design optimization tasks.

III. LLM-assisted Surrogate Model Selection

In this section, we present the workflow utilized to select, train, and apply surrogate models in optimization tasks, which we utilized in our experiments (Fig. 1). In a first step, we assume practical relevant conditions as detailed below for the optimization problem and interact with ChatGPT to develop a method to select a surrogate model for the optimization task. In a second step, we apply the developed method on two sets of data, *i.e.* benchmark data sets and a realistic application data set, and perform the optimizations to verify the performance of the developed methodology.

[1]https://openai.com/

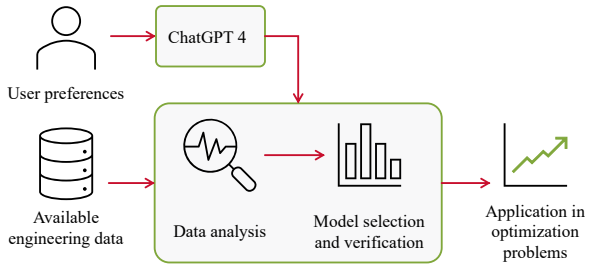

Fig. 1: Utilized workflow to select and fit surrogate models for optimization based on the user preferences and available training data.

A. ChatGPT-based Methodology

We start the development of the methodology by providing the context and objectives of our task to ChatGPT (Prompt 1). Here, we assume a typical engineering scenario where the user wants to optimize a system that is computationally expensive to simulate but has performance data available from previous simulations or physics experiments. Furthermore, as gradient information is often unavailable in complex simulation models, we constrain our cases to optimizations with evolutionary algorithms. The corresponding answer (Answer 1) outlines a 5-step process, which starts with the analysis of the data characteristics and ends with the integration of the model to the optimization task.

***Prompt** 1:* I need to train regression models as objective functions (or part of the objective functions) for different optimization tasks. My data sets have different sizes (samples and features) and the optimization method is an evolutionary algorithm. Can you please help me to create a process to select a technique to fit the data and to code the created process into a Python script?

***Answer** 1:* Certainly! To create a process for selecting a regression model as an objective function for different optimization tasks, and to implement it in Python, you can follow these steps:

- Step 1: Analyze your data (data size, feature types, data quality)
- Step 2: Select regression techniques (linear regression, ridge regression, decision trees, support vector regression, neural networks)
- Step 3: Define model selection criteria (performance, computational efficiency, interpretability)
- Step 4: Implement the process in Python
- Step 5: Integrate with optimization task
- Final Steps: Test the script with your datasets. Iterate to fine-tune model selection and preprocessing steps as needed.

This outline gives you a starting point. You'll need to adapt and expand it based on the specifics of your datasets and optimization tasks.

Since the obtained answer outlines a general process, we proceeded by specifically requesting the Python scripts for performing the tasks in each of the steps (Prompt 2). Also, we requested ChatGPT to convert the software generated based on Prompt 2 into a function, which is easier to integrate to multiple scripts, and to generate only numerical data, which avoids extensive comparisons based on visual analyses (Prompt 3).

***Prompt** 2:* In the data processing steps (steps 1 to 3), can you generate a script to evaluate the data? And how should I use the values to select the surrogate modeling technique?

***Prompt** 3:* Can you convert the script into a function which outputs the classification of the data set (large data set: boolean), type of features (continuous, categorical or mixed), and data quality metrics (missing values: boolean, outliers: boolean, skewness)?

The obtained answer (Answer 3) comprises a Python script that evaluates the data set with respect to size, type of features, outliers, skewness, and missing values. Most of the properties are directly obtained by importing the data into a data frame using the pandas library. ChatGPT proposed to utilize the interquartile range (IQR) method to detect outliers in the data set, which is one of the most used techniques for this purpose [23]. In the Python script, ChatGPT suggests values for classifying the data set with respect to size, presence of outliers, and skewness criteria, but also includes a disclaimer at the end of the answer, which warns the user that the script potentially requires further tuning to perform well on specific cases.

***Answer** 3:* Sure, I can modify the script into a function [...] The function will return whether the dataset is large, the type of features (continuous, categorical, or mixed), and data quality metrics including the presence of missing values, outliers, and skewness.

```
import pandas as pd
import numpy as np

def evaluate_data(data):
    # Define thresholds
    # Define what you consider as a
    #large dataset
    large_dataset_threshold = 100000
    # Typically used in boxplots
    outlier_threshold = 1.5
    # Define your skewness threshold
    skewness_threshold = 0.5

    # Data Size Evaluation
    [...]
```

```
    return {
        'is_large_dataset': \
            is_large_dataset,
        'feature_type_classification': \
            feature_type_classification,
        'has_missing_values':\
            has_missing_values,
        'has_outliers': has_outliers,
        'has_skewness': has_skewness}
```

This function provides a basic framework for dataset evaluation, and you may need to extend or modify it according to the specifics of your data and the requirements of your models.

To generate the algorithms for steps 2 (fit the data) and 3 (selection criteria), we asked ChatGPT how to fit the data and compute the performance indicators to select the methods (Prompt 4). However, the obtained script lacked artificial neural networks (ANNs) as an option of regression technique, compared to the suggestions in Answer 1-Step 3. Hence, in two additional prompts, we requested ChatGPT to generate a new script that includes ANNs and to consider other potential models, such as multivariate adaptive regression spline (MARS) [24] and automatic learning of algebraic models for optimization (ALAMO) [25]

***Prompt** 4:* Moving to Step 2, how can I fit the data using the techniques that you suggested? Also, on Step 3, how can I calculate the metrics to evaluate the criteria that you suggested?

In the following step, we asked ChatGPT to modify the software into a Python function to ease the integration with other applications. The obtained script (Answer 4) includes only the fitting techniques suggested by ChatGPT and the implementation in our hardware setup required only minor adaptations. For example, some of the functions from the Scikit-Learn library that are utilized to evaluate the models lack compatibility with the ANNs built using the tensorflow library. Furthermore, ChatGPT included a 5-fold cross validation procedure, which is often utilized to verify data-driven models against overfitting [26], but neglected any data normalization procedure before training the models. However, we did not include the normalization in our experiments to evaluate the method as generated by ChatGPT, and only added a custom wrapper for the Tensorflow functions to enable the script to evaluate all generated models using the same techniques.

***Answer** 4:* Creating a Python function that takes a dataset (as a CSV file) and evaluates various regression models [...] Here's how you can structure such a function: Reading and Preparing the Data: The function will read [...] Model Training and Evaluation: It will train different models [...] Model Complexity and Interpretability [...]

Python Function Implementation

```
import pandas as pd
[... import other libraries ...]

# Function to evaluate models
def evaluate_models(csv_file):
    # Load and prepare data
    data = pd.read_csv(csv_file)
    [...]
    # Define models
    models = {
        'LinearRegression': \
            LinearRegression(),
        'Ridge': Ridge(),
        'RandomForest': \
            RandomForestRegressor(),
        'SVR': SVR(),
        'ANN': Sequential([...])}

    # Model evaluation
    results = {}
    for name, model in models.items():
        [...]
        model.fit(X_train, y_train)
        # Cross-validation scores
        scores = cross_val_score([...])
        [...]
        # Calculate metrics
        ## mse, rmse, mae, mae, r2
        [...]
        # Number of parameters
        ## layers*neurons for ANN
        ## n_features* for linear
        [...]
        # Interpretability
        ## LR, Ridge --> 'HIGH'
        ## Else --> 'LOW'

        results[name] = {
            'MSE': mse,
            'RMSE': rmse,
            'MAE': mae,
            'R2': r2,
            'Num_Params': num_params,
            'Interpretability': \
            interpretability}
    return results
```

Notes: [...] This function serves as a comprehensive starting point for evaluating various regression models on a given dataset and can be further customized and expanded based on specific requirements and advanced modeling techniques.

In a last step, we requested ChatGPT to generate a script to select a technique for the surrogate modeling task based on the output of the previous functions (Prompt 5). The objective is to seamlessly provide the data set as a comma-separated value (CSV) file as input and obtain as output the technique for surrogate modeling.

***Prompt** 5:* Now, to select the model for my task: If I have the output of the functions evaluate_data and evaluate_models, how can I use these outputs and my

preferences to select a technique? Can you generate the Python script for that?

As a result (Answer 5), ChatGPT created a function that performs the data analyses, feeds the result into the model selection and evaluation, and yields the best performing model. Although in the script only the mean-squared error is utilized to select the fitting technique, ChatGPT also added an in-line comment that the user should change the metric according to his/her preference, *e.g.*, the 5-fold cross validation scores to select more robust techniques against overfitting. Hence, the algorithm generated by ChatGPT accounts for the data properties, performance of the techniques, and, indirectly, the preferences of the user.

***Answer** 5:* Certainly! To select the best model for your task based on the outputs from evaluate_data and evaluate_models, and considering your preferences [...]

Python Script for Model Selection:

```
def select_model(data_evaluation,\
                 model_evaluation,):
    # Filter models based on data size
    #and type
    suitable_models = []
    for model, properties in\
           model_evaluation.items():
        # Data set size
        ## Skip complex models for large
        ##datasets
        [...]
        # Features type
        ## Skip linear models for purely
        ##categorical data
        [...]
        # Missing values
        ## Skip sensitive models for poor
        ##quality data
        [...]
        # Add model to the list if it
        #passes the filters
        suitable_models.append(\
                    (model, properties))

    # Rank models based on metrics
    # Here, we consider MSE as the primary
    #metric; modify as per your preference
    suitable_models.sort(key=lambda x:\
                         x[1]['MSE'])

    # Return the best model based
    #on the above criteria
    return(suitable_models[0][0] \
           if suitable_models else None)
```

[...] This combined script should provide a comprehensive evaluation of various regression models, including ANNs, facilitating an informed decision for your modeling task.

IV. Validation Experiments

We validate the method developed with ChatGPT (Section III) by implementing and applying the generated software on the following sets of surrogate-assisted optimization problems, where we do not directly interact with ChatGPT.

In the first set of experiments, based on [11], we utilize analytical functions to generate data sets of different size and number of features, as well as to perform the surrogate-assisted optimizations. The objective is to verify how the performance of the method varies with respect to the availability of data, function characteristics, and dimensionality of the design space. For our first set of experiments, we implemented the n-dimensional versions of the following functions: Ackley, Griewank, Levy, Zakharov, Dixon-Price, Rosenbrock, Michalewicz, Perm d, β, and Stybtang [27]. For all the selected functions, we generated data sets that comprise $\{500, 1000, 2000, 5000\}$ samples within the domain $(-1, 1)^D$, where $D = \{2, 4, 6, 8, 10, 50, 100\}$ is the number of features.

In the second set of experiments, we replace the analytic functions by engineering data from CarHoods10k [28], which is a large data set containing realistic car hood designs along with a structural performance. Specifically, we utilize 50 subsets of the data to fit the surrogate models for predicting the stress response on the car hoods based on the design parameters that were utilized to generate the geometries.

For both, the analytical and engineering data sets, we afterwards utilize the covariance matrix adaptation evolutionary strategy (CMA-ES [29]) to optimize the designs. The utilized strategy is (μ, λ), where the population size λ and number of parent samples μ are adapted based on the dimensionality of the domain. We perform the optimizations for a maximum of 100 generations or until the changes in the performance of the fittest individual becomes less than 1%.

V. Results and Discussion

A. Data from Analytic Functions

In a first analysis, we verified the MSE obtained in the training and cross-validation (MSE_{CV}) steps of the algorithm developed with ChatGPT for all surrogates and functions (Fig. 2). We observed that linear and ridge regression methods are the techniques that yield the worst performance and, thus, were selected only 13 and 16 times (combined) out of 224 experiments based on the training and cross validation errors, respectively. However, since all utilized functions are nonlinear, we expected these models to perform well only in particular cases.

Differently, the ANN and random forest achieved the best performances overall. Each technique was selected as best model for a third of the tested cases and for both MSE and MSE_{CV} metrics. Interestingly, the developed method selected random forest models more often than ANNs based on the training MSE, while the opposite happened when the cross-validation error was utilized. Furthermore, the random forest technique was selected more frequently to fit the data from the Michalewitcz function than from the other functions, while the number of times the ANN was selected is more evenly distributed over the functions (Fig. 3).

In a second step, we applied the selected models as objective functions in surrogate-assisted optimization problems

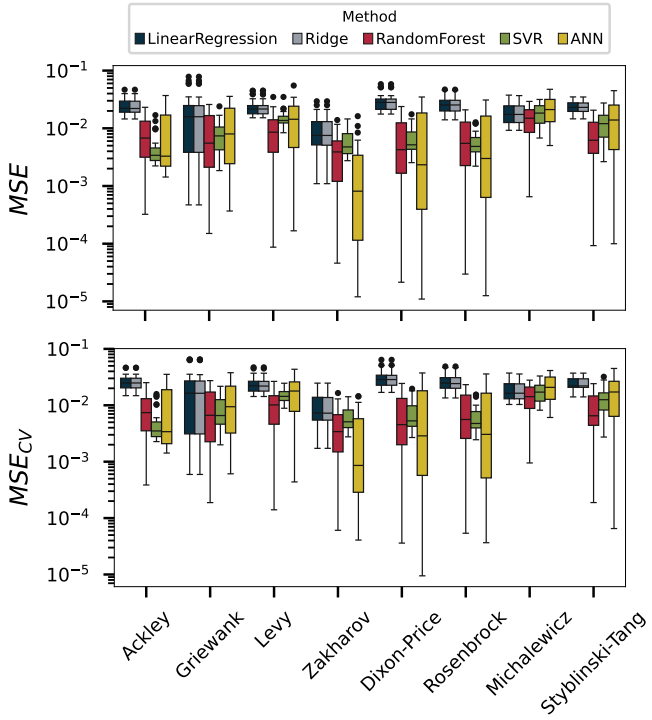

Fig. 2: Distribution of the mean squared error (MSE) measured on the test set (10% of the data) and mean over the cross validation folds (MSE_{CV}) over all dimensionalities and data set sizes.

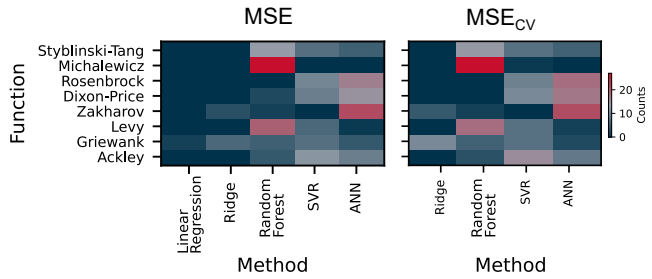

Fig. 3: Number of times each technique was selected based on the developed methodology to fit the data generated by each of the utilized functions.

according to the aforementioned settings (Section IV). To set a baseline performance, we also performed the optimizations using the corresponding analytical functions.

By verifying the variation of the normalized fitness value $f(x)_N$ over the generations (Fig. 4), we observed that the optimizations with the surrogate models behave similarly as in the experiment with the analytic functions, apart from the optimizations with the *Zakharov* and *Griewank* functions. In these two cases, the optimizations perform slower and yield an abrupt drop of the $f(x)$ value towards the 100$^{\text{th}}$ generation. The potential causes for such behavior are the multi-modality of the *Griewank* function landscape, which hinders the accuracy of the surrogate model, and the lack of data normalization, which leads to scaling issues and lack of variance to learn the *Zakharov* function.

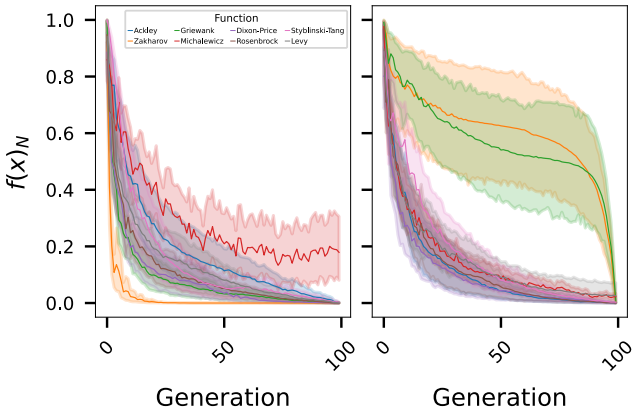

Fig. 4: Best fitness values obtained in each generation of the optimizations performed with the analytic functions and surrogate models.

We confirmed our hypotheses by computing the Euclidean distance between the fittest individuals X_A and X_S obtained in the optimization with the analytical functions and in the corresponding surrogate-assisted optimization case for each generation, respectively (Eq. 1). Since the distance scales differently with the dimensionality D of the problem, we normalized the distance by the length of the main diagonal of the hypercube formed by the training space. In this analysis (Fig. 5), we observed that the fittest individuals of the surrogate-assisted optimization drifted apart from the samples in the corresponding analytical cases. Hence, the surrogate models selected by the proposed method are the best performing among the available set of methods for the experiments, but they still have limited prediction (extrapolation) capability, which is a known issue of data-driven models and an expected limitation of the proposed framework.

$$\|X_A - X_S\|_N = \frac{\|X_A - X_S\|_2}{2\sqrt{D}} \tag{1}$$

B. Structural Optimization of Car Hoods

In the second set of experiments, we extend our experiments to a data set of realistic engineering data. For this purpose, we considered the metadata of 50 subsets of the CarHoods10k data set [28], which comprise the values of geometric parameters that represent each hood structure and corresponding performance metrics, *e.g.*, mass and maximum equivalent stress. The geometries in each subset are defined by a template structure, namely a *skin*, and differ both in terms of topology, *e.g.*, number of cuts and ribs, as well as dimensions of geometric features, such as radii of bends. The hoods were simulated for bending and twist cases (Fig. 6), since the performance in those cases are considered as practically relevant design criteria during the development of car hoods. For our experiments, we consider the geometric features as design space and the maximum equivalent stress as performance (output) value to fit the surrogate models.

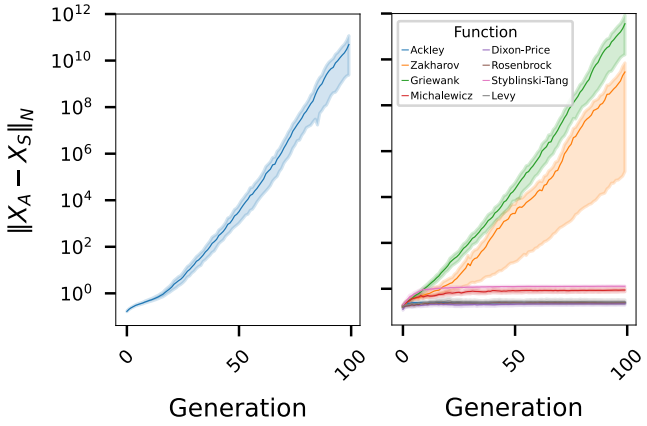

Fig. 5: Distance between the fittest individuals in the optimizations with the analytic functions (X_A) and surrogate models (X_S) for each generation, and the corresponding breakdown based on the utilized objective functions.

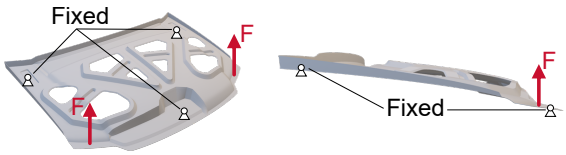

Fig. 6: Schematic of the boundary conditions utilized to simulate the hood structures.

In terms of data properties, despite the car hoods were simulated as static linear-elastic models, the differences in the shape topology lead to different locations and magnitudes of maximum stress. Furthermore, the space defined by the geometric features mixes continuous and discrete variables, *e.g.*, width and number of reinforcement ribs. Hence, the data set potentially contains outliers and multi-modal performance landscapes, which hinder the training of surrogate models.

In the first step of our analysis, we applied the proposed selection method and compared the performance of the regression techniques across the 50 sets of designs (Fig. 7). As expected, the linear regression model outperformed the other methods in only 4 out of 50 cases, while random forest was the best performing method in 22 out of 50 cases. ANN was selected as best model only in 5 cases, which is significantly less frequent than in the experiments with analytical functions. The potential cause for selecting ANNs less frequently is the low number of training samples (around 100), which hinders the training of neural networks.

In a second step, we applied the selected models in surrogate-assisted optimizations to minimize the equivalent stress for the selected load conditions. By evaluating the fitness values during the optimizations (Fig. 8), we observed a similar responses as in the cases with the analytical functions. For some of the skins, the fitness value converged smoothly over the generations, while for other skins the fitness abruptly changes at late stages of the optimization. By comparing to the MSE values of the utilized surrogate models, we also observed

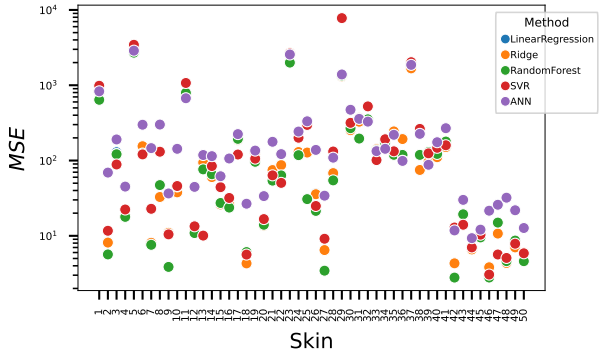

Fig. 7: MSE computed on the test set after training the surrogate models on different subsets (skins) of the CarHoods10k data set.

that there is no direct correlation between the performance of the models and the optimization convergence behavior.

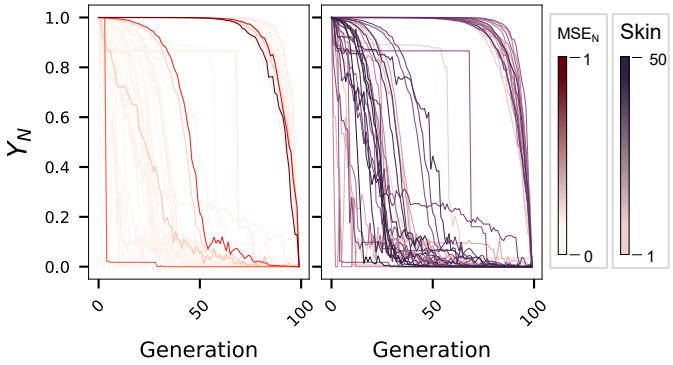

Fig. 8: Normalized fitness value of the fittest individual obtained during the optimizations colored with respect to the normalized MSE of the utilized surrogate model (left) and skin number (right).

In this case, as the optimizations are unconstrained, the optimizer potentially sampled significantly different regions of the input space compared to the training data, or even generated unfeasible values for the input parameters. We confirmed this hypothesis by computing the Euclidean distance between the representation of the fittest individuals obtained during the optimizations and the fittest individual in the corresponding data set (Fig. 9). In this analysis, we observed that some of the samples diverged from the fittest designs of the data set, which is an indication that the obtained optimal set of parameters is not feasible. Furthermore, by comparing the predicted performance y_{opt} with respect to the values y_D of the designs with lowest equivalent stress, we observed that the optimization cases that diverged from the best sample in the set yield unrealistic differences in stress (over 10E4 MPa). Therefore, in those cases, adding design constraints and tuning the selected surrogate models are potential measures to avoid these problems and obtain more realistic solutions.

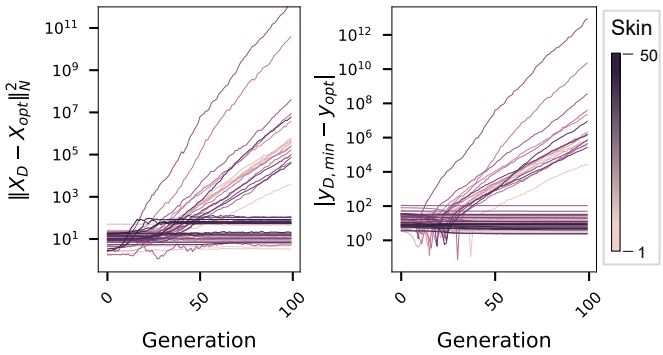

Fig. 9: Euclidean distance between the representation of the fittest individual in each generation and the fittest design in the data set (left) and absolute difference in performance compared to the fittest design (right).

VI. Conclusion and Outlook

In this paper, we propose a framework to select regression methods for surrogate-assisted optimization by interacting with ChatGPT. We generated the process by formulating prompts that focused on the process rather than specific data properties, such that the obtained framework can be extended to other cases than benchmark functions or specific optimization problems. Then, we applied the obtained framework on analytical benchmark optimization functions and an engineering data set of realistic car hood designs. We showed that the process created together with ChatGPT follow most common practices of data analysis and allowed us to train feasible surrogate models for design optimization problems.

We also noticed that the process generated by ChatGPT has technical and fundamental flaws, *e.g.*, neglecting the data normalization. However, the interaction with ChatGPT still accelerated the development of the framework, as LLM facilitates the access to data science knowledge, and the observed limitations can also be addressed by further human-machine interaction. Thus, the creation of the framework in a discussion-style is a considerable step towards improving the explainability of the decision process and democratization of data science knowledge, which can be potentially utilized for assisting novice users and accelerating different development processes in engineering.

References

[1] O. J. Achiam *et al.*, "GPT-4 Technical Report," *ArXiv preprint:2303.08774 [cs.CL]*, 2023.

[2] J. White, S. Hays, Q. Fu, J. Spencer-Smith, and D. C. Schmidt, "ChatGPT Prompt Patterns for Improving Code Quality, Refactoring, Requirements Elicitation, and Software Design," 2023. [Online]. Available: https://api.semanticscholar.org/CorpusID:257505363

[3] R. Rombach, A. Blattmann, D. Lorenz, P. Esser, and B. Ommer, "High-resolution image synthesis with latent diffusion models," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. Los Alamitos, CA, USA: IEEE Computer Society, jun 2022, pp. 10 674–10 685. [Online]. Available: https://doi.ieeecomputersociety.org/10.1109/CVPR52688.2022.01042

[4] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-E: A System for Generating 3D Point Clouds from Complex Prompts," *ArXiv preprint: 2212.08751 [cs.CV]*, 2022.

[5] H. Jun and A. Nichol, "Shap-E: Generating Conditional 3D Implicit Functions," *ArXiv preprint: 2305.02463 [cs.CV]*, 2023.

[6] C. Yang *et al.*, "Large Language Models as Optimizers," *ArXiv preprint: 2309.03409 [cs.LG]*, 2023.

[7] S. Liu, C. Chen, X. Qu, K. Tang, and Y.-S. Ong, "Large Language Models as Evolutionary Optimizers," *ArXiv preprint: 2310.19046 [cs.NE]*, 2023.

[8] L. Zheng *et al.*, "Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena," *ArXiv preprint: 2306.05685 [cs.CL]*, 2023.

[9] N. Minh, Y. S. Ong, S. Menzel, Y. Jin, and B. Sendhoff, "Evolution by Adapting Surrogates," *Evolutionary Computation Journal*, vol. 21, no. 2, pp. 313–340, 2013.

[10] Y. Jin, "A comprehensive survey of fitness approximation in evolutionary computation," *Soft Computing*, vol. 9, no. 1, pp. 3–12, 2005.

[11] B. Williams and S. Cremaschi, "Selection of Surrogate Modeling Techniques for Surface Approximation and Surrogate-based Optimization," *Chemical Engineering Research and Design*, vol. 170, pp. 76–89, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0263876221001465

[12] P. Wollstadt, M. Bujny, S. Ramnath, J. Shah, D. Detwiler, and S. Menzel, "Carhoods10k: An industry-grade data set for representation learning and design optimization in engineering applications," *IEEE Transactions on Evolutionary Computation Special Issue on Benchmarking Sampling-Based Optimization Heuristics: Methodology and Software (BENCH)*, January 2022.

[13] S. Ramnath, J. Shah, P. Wollstadt, M. Bujny, S. Menzel, and D. Detwiler, "Osu-honda automobile hood dataset (carhoods10k)," *Dryad.org*, February 2022.

[14] Z.-H. Han and K.-S. Zhang, "Surrogate-Based Optimization," in *Real-World Applications of Genetic Algorithms*, O. Roeva, Ed. Rijeka: IntechOpen, 2012, ch. 17. [Online]. Available: https://doi.org/10.5772/36125

[15] Y. Jin, "Surrogate-assisted evolutionary computation: Recent advances and future challenges," *Swarm and Evolutionary Computation*, vol. 1, no. 2, pp. 61–70, 2011. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2210650211000198

[16] J. Stork *et al.*, *Open Issues in Surrogate-Assisted Optimization*. Cham: Springer International Publishing, 2020, pp. 225–244.

[17] T. Chatterjee, S. Chakraborty, and R. Chowdhury, "A critical review of surrogate assisted robust design optimization," *Archives of Computational Methods in Engineering*, vol. 26, no. 1, pp. 245–274, Jan 2019. [Online]. Available: https://doi.org/10.1007/s11831-017-9240-5

[18] M. Ben Salem and L. Tomaso, "Automatic selection for general surrogate models," *Structural and Multidisciplinary Optimization*, vol. 58, no. 2, pp. 719–734, Aug 2018. [Online]. Available: https://doi.org/10.1007/s00158-018-1925-3

[19] B. S. Saini, M. Lopez-Ibanez, and K. Miettinen, "Automatic Surrogate Modelling Technique Selection Based on Features of Optimization Problems," in *Proceedings of the Genetic and Evolutionary Computation Conference Companion*, ser. GECCO '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 1765–1772. [Online]. Available: https://doi.org/10.1145/3319619.3326890

[20] R. Lingo, "The Role of ChatGPT in Democratizing Data Science: An Exploration of AI-facilitated Data Analysis in Telematics," *ArXiv preprint: 2308.02045 [cs.CY]*, 2023.

[21] L. Makatura *et al.*, "How Can Large Language Models Help Humans in Design and Manufacturing?" *ArXiv preprint: 2307.14377 [cs.CL]*, 2023.

[22] C. Picard *et al.*, "From Concept to Manufacturing: Evaluating Vision-Language Models for Engineering Design," *ArXiv preprint: 2311.12668 [cs.AI]*, 2023.

[23] J. Yang, S. Rahardja, and P. Fränti, "Outlier detection: How to threshold outlier scores?" in *Proceedings of the International Conference on Artificial Intelligence, Information Processing and Cloud Computing*, ser. AIIPCC '19. New York, NY, USA: Association for Computing Machinery, 2019. [Online]. Available: https://doi.org/10.1145/3371425.3371427

[24] J. H. Friedman and C. B. Roosen, "An Introduction to Multivariate Adaptive Regression Splines," *Statistical Methods in Medical Research*, vol. 4, no. 3, pp. 197–217, 1995, pMID: 8548103. [Online]. Available: https://doi.org/10.1177/096228029500400303

[25] A. Cozad, N. V. Sahinidis, and D. C. Miller, "Learning surrogate models for simulation-based optimization," *AIChE Journal*, vol. 60, no. 6, pp. 2211–2227, 2014. [Online]. Available: https://aiche.onlinelibrary.wiley.com/doi/abs/10.1002/aic.14418

[26] B. Ghojogh and M. Crowley, "The theory behind overfitting, cross validation, regularization, bagging, and boosting: Tutorial," *ArXiv preprint:1905.12787 [stat.ML]*, 2019. [Online]. Available: https://api.semanticscholar.org/CorpusID:170078761

[27] S. Surjanovic and D. Bingham, "Virtual library of simulation experiments: Test functions and datasets," Retrieved December 21, 2023, from http://www.sfu.ca/ ssurjano.

[28] P. Wollstadt, M. Bujny, S. Ramnath, J. J. Shah, D. Detwiler, and S. Menzel, "CarHoods10k: An Industry-Grade Data Set for Representation Learning and Design Optimization in Engineering Applications," *IEEE Transactions on Evolutionary Computation*, vol. 26, no. 6, pp. 1221–1235, 2022.

[29] N. Hansen, "The CMA evolution strategy: A tutorial," *ArXiv preprint: 1604.00772 [cs.LG]*, vol. 1604.00772, 2016. [Online]. Available: http://arxiv.org/abs/1604.00772