The 9th European Congress on Computational Methods in Applied Sciences and Engineering
ECCOMAS Congress 2024
3 – 7 June 2024, Lisbon, Portugal

Research on Airborne System Simulation Methodology Based on AI-Enhanced Surrogate Modelling Approach

Chuang Wang[1]*, Shenglai Wan[1],Chunxu Wang[1],Wen Zhao[1]

[1] AVICAS GENERIC TECHNOLOGY CO., LTD, Jiangsu, Yangzhou, 225000, China

Abstract: Model-based system engineering is a promising approach to achieve the design goal of green aviation by running virtual simulation to reduce energy consumption and emission. The virtual simulation of airborne systems encompasses the logic simulation of avionics and electromechanical controllers, as well as functional simulations of subsystems such as power system and hydraulics system [1]. Certain complex electromagnetic and fluid equations involved in this simulation possess characteristics of high computational costs and small convergence solving steps. Such inconsistency in the timescales of the entire simulation system results in a delay in the overall numerical simulation time. A common solution to address this issue is to employ surrogate models for model reduction [2].

The current model reduction methods used in airborne system integrated simulations in practical engineering tasks remain at the interpolation table stage, with low data complexity. However, most airborne system or module models are multidimensional, making it challenging to construct high-dimensional surrogate models. To tackle these challenges, this paper proposes a semi-automatic model reduction method based on large language models and existing mature fitting algorithms. A complete toolchain for model reduction has been developed, firstly, utilizing large language models to assist in selecting fitting algorithms and constructing fitted models. Then, training and deployment of models are conducted using an artificial intelligence model development platform, which is integrated into the GvSimLab platform to connect to other airborne system models for joint simulation.

Taking a particular IPMSM module(Internal Permanent Magnet Synchronous Motor) in airborne VFG system(Variable Frequency Generator system) as an example, this toolchain can elevate the model-in-loop solving time from 30s per thousand steps to nearly real-time at 0.8s with R^2 equals 0.9996.

Key Words: *Airborne System, Virtual Integration, Complex Model Computation, Surrogate model, Large Language Models*

1. Introduction

With the increasingly severe challenges posed by global climate change, the concept of green aviation has emerged. It aims to reduce carbon emissions in the aviation transportation industry through technological innovation and optimized operational strategies, thereby achieving sustainable development in aviation. Model-Based Systems Engineering (MBSE), as an advanced systems engineering approach, provides robust support for the development of green aviation. By constructing and simulating complex system models, MBSE optimizes the design process and enhances research and development efficiency, playing a crucial role in aircraft engine design, aircraft system development, and more[3].

In the context of green aviation, the application of MBSE not only improves the energy efficiency and performance of aviation products but also promotes the aviation industry's transition towards low-carbon and environmentally friendly practices. For example, researchers can use MBSE methods to systematically consider the requirements, architecture, and validation of aircraft engines during the design phase, enabling efficient development of aviation engine systems [4, 5]. Additionally, MBSE is employed in the design and development of safety-critical systems in civilian aircraft, enhancing aircraft system safety while contributing to the goals of green aviation[6, 7].

In summary, MBSE plays a critical role in driving the development of green aviation. By providing a systematic, model-driven approach, it helps the aviation industry achieve higher energy efficiency and lower environmental impact in the design and manufacturing processes, which is essential for the industry's long-term sustainable development.

1.1 Heterogeneous Model Integration

In the process of implementing Model-Based Systems Engineering development for airborne systems, various stages and types of models are involved, collectively referred to as heterogeneous models. A background investigation into methods for integrating different types of models has been conducted, focusing on FMI (4 papers) and Modelica (3 papers). The challenge with these two methods lies in the necessity of converting source models into FMI or Modelica, which limits their universality. This study proposes a heterogeneous model integration framework that leverages the characteristics of both FMI and the source models, establishing an integration platform to achieve the integration of multiple heterogeneous models, including FMI, Simulink, C code, and AMESim.

1.2 Surrogate Model for System Integration

When dealing with practical engineering applications, it is inevitable to use high-precision and long-time-scale models for simulation. These models typically involve large computational costs and are difficult to solve, especially for simulations with longer time scales, where a single simulation may take several hours or even days to complete, resulting in high time and computational costs. To improve this situation, researchers prioritize the use of surrogate models [8, 9, 10, 11, 12]. The computational results of surrogate models are very close to those of the original models; however, due to their lower computational complexity, they greatly reduce time costs. Surrogate models are usually built using data-driven, bottom-up methods. They adopt a black-box approach, calculating the model's response based on a limited number of data points and establishing behavioral rules between inputs and outputs to build the surrogate model [13, 14]. However, due to the complexity of surrogate model algorithms and their high learning curves, they have not yet been widely implemented in engineering applications. Currently, the common solution in field engineering is still the use of interpolation tables, which can maintain relatively high accuracy for low-dimensional models. However, for high-dimensional models, the simulation effects using interpolation tables are not accurate enough [15, 16, 17].

1.3 LLM Enhanced AutoML

The application of machine learning requires human intervention, involving aspects such as feature extraction, model selection, and parameter tuning. Automated Machine Learning (AutoML) is the process of automating end-to-end machine learning application to real-world problems [18]. A complete AutoML process typically includes several stages such as data preparation, feature engineering, model generation, and model evaluation [19]. Among these, neural architecture search (NAS) is a hot research topic in AutoML tasks [20], capable of automatically learning the optimal network structure.

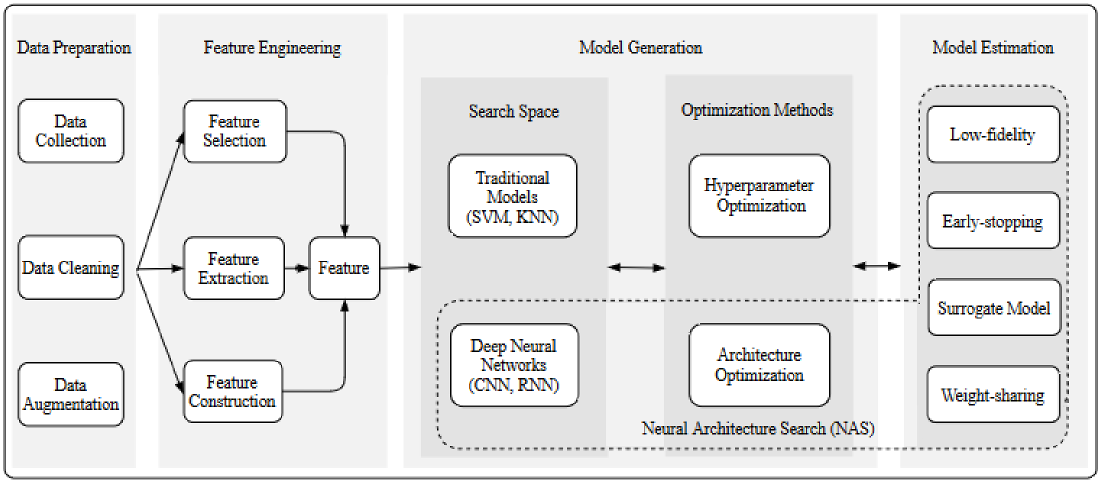

Figure 1 An overview of AutoML pipeline[4]

Large Language Models (LLM) [21] are a class of language models pretrained on large-scale text datasets. LLMs demonstrate remarkable capabilities across various natural language processing (NLP) tasks, with pretrained versions readily available to a wide user base [22, 23, 24].

According to research [25], the integration of these two fields can fundamentally push the boundaries of each other. Existing LLMs, such as GPT-4 [26], LLaMa [27], and PalM [28], exhibit a profound understanding of natural language and can generate coherent, context-aware responses. This advancement opens up new potential applications for challenging tasks involving diverse data domains, such as image and text processing, as well as the integration of domain-specific knowledge [29]. Utilizing LLMs to automate the model training process appears to be a promising choice, leveraging GPT as a bridge for various algorithmic models, avoiding the domain knowledge requirements of AutoML, and changing the user interaction with AutoML systems [30, 31].

Given the challenges faced by current onboard system integrations and considering the practical technical difficulties such as the high learning curve and limited generality of surrogate models, this research proposes a semi-automatic model reduction method that combines LLM-enhanced AutoML with surrogate model construction and training. Our work has developed a complete model reduction toolchain. Firstly, the toolchain leverages LLMs to assist in selecting fitting algorithms and constructing fitting models. Then, it utilizes an artificial intelligence model development platform for model training and deployment, integrated into the GvSimLab platform, connecting other onboard system models for joint simulation.

This study applies the proposed approach to the specific IPMSM module (Interior Permanent Magnet Synchronous Motor) within the onboard VFG system (Variable Frequency Generator system), yielding promising results. Compared to the interpolation tools currently used, the toolchain improves the accuracy of simulating non-sample data by 15%. Compared to Simulink models, the toolchain achieves near real-time performance, reducing the solution time from 30 seconds per thousand steps to approximately 0.8 seconds with an accuracy of R^2 0.996.

2.Platform Framework

2.1 Overall Framework

The complete toolchain for model reduction consists of three components: AI platform, LLM based Modeling pipeline and SimLab platform. The primary workflow, illustrated in

Figure 2, encompasses task decomposition, model selection, iterative optimization, model deployment, model invocation, interface parsing, and collaborative simulation. Initially, a large language model assists in selecting fitting algorithms and constructing fitting models. Subsequently, the AI model development platform is employed for training and deploying the models, which are then integrated into the SimLab platform to facilitate collaborative simulation with other onboard system models.

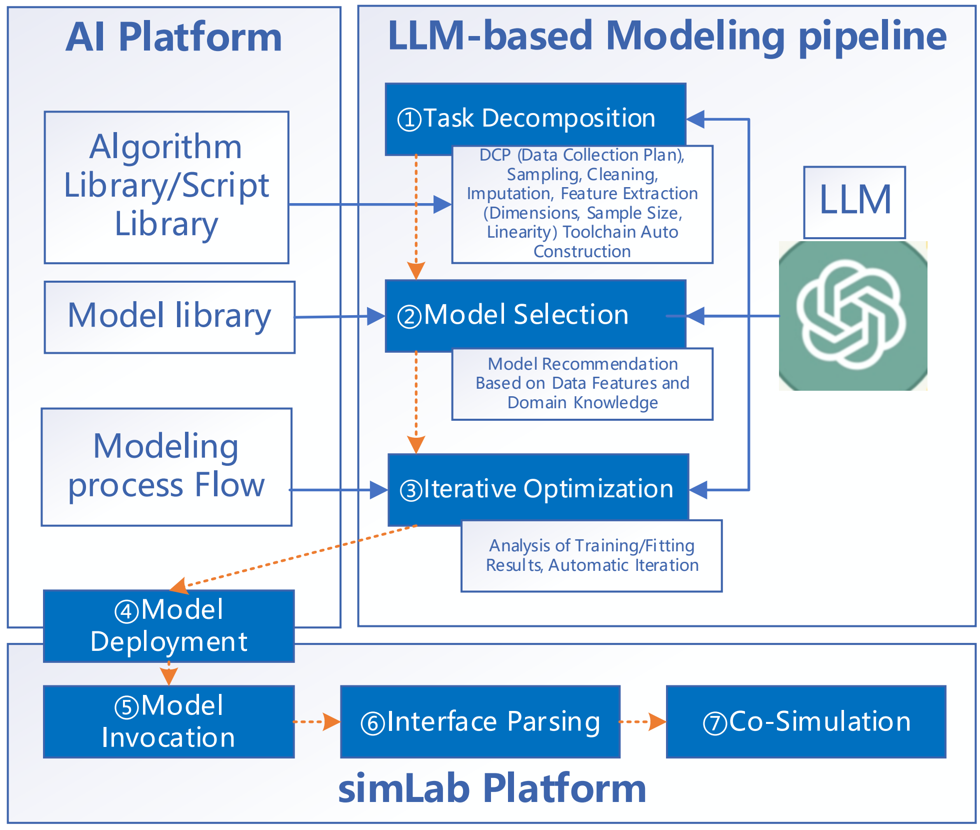

Figure 2 toolchain for model reduction

2.2 SimLab Platform

As illustrated in Figure 3, the Simlab platform is a model integration and simulation validation platform that supports multi-source heterogeneous models. Simlab platform supports a seamless integration of fully digital to semi-physical virtual simulation testing, offering an end-to-end, one-stop simulation service that encompasses project preparation, execution, and result evaluation. It facilitates simulation operations in both full-speed and real-time modes, and provides capabilities for real-time online monitoring, storage, and playback analysis of simulation data. In the construction of surrogate models, the model integration platform SimLab integrates these surrogate models with other models such as Simulink, SCADE, and FMU.

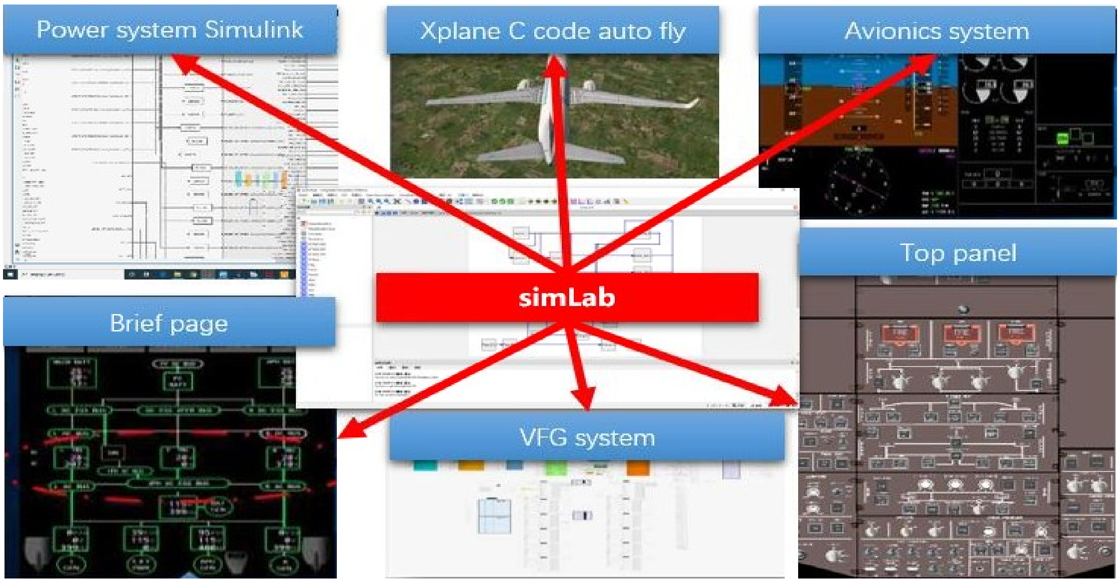

Figure 3 SimLab Platform

2.3 The AI Model Development and Training Platform

The AI model development and training platform provides a low-code "end-to-end" solution for the entire process from data access, data preprocessing, model training, model evaluation, model deployment, to model management. It integrates various algorithm components such as large language models, deep learning, and machine learning, allowing for rapid model building and deployment through a drag-and-drop modeling approach. This meets the engineering requirements of artificial intelligence, including the comprehensive utilization of big data resources and efficient collaboration.

The overall architecture of the platform, as shown in Figure 4, consists mainly of two parts: the big data basic services and the smart application model development platform. The big data basic services serve as the core underlying engine for machine algorithm environments, comprising four parts: single sign-on, data center interfaces, artificial intelligence computing frameworks, and programming language interfaces. The smart application model development platform provides functions such as data resource library, algorithm library, model library, platform basic services, model design, model deployment, API invocation, and SDK. The platform facilitates the correlation of data, models, and intelligent decision-making applications through a series of activities including preprocessing of raw data, feature extraction based on intelligent decision-making objectives, model training, model evaluation, and final model generation, as illustrated in Figure 5.

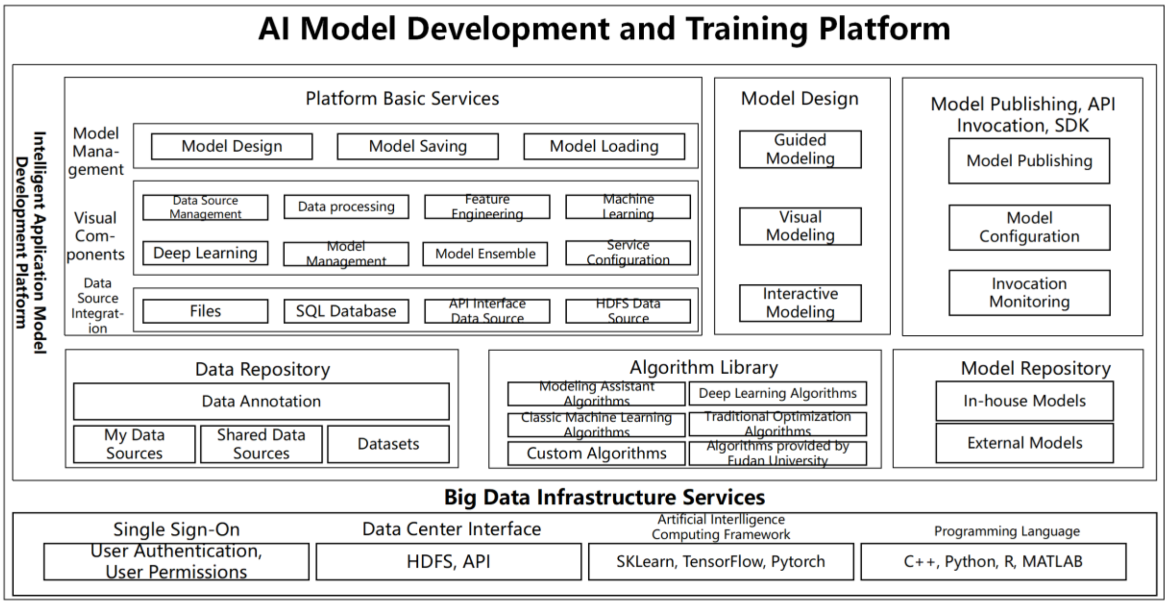

Figure 4 AI Model Development and Training Platform

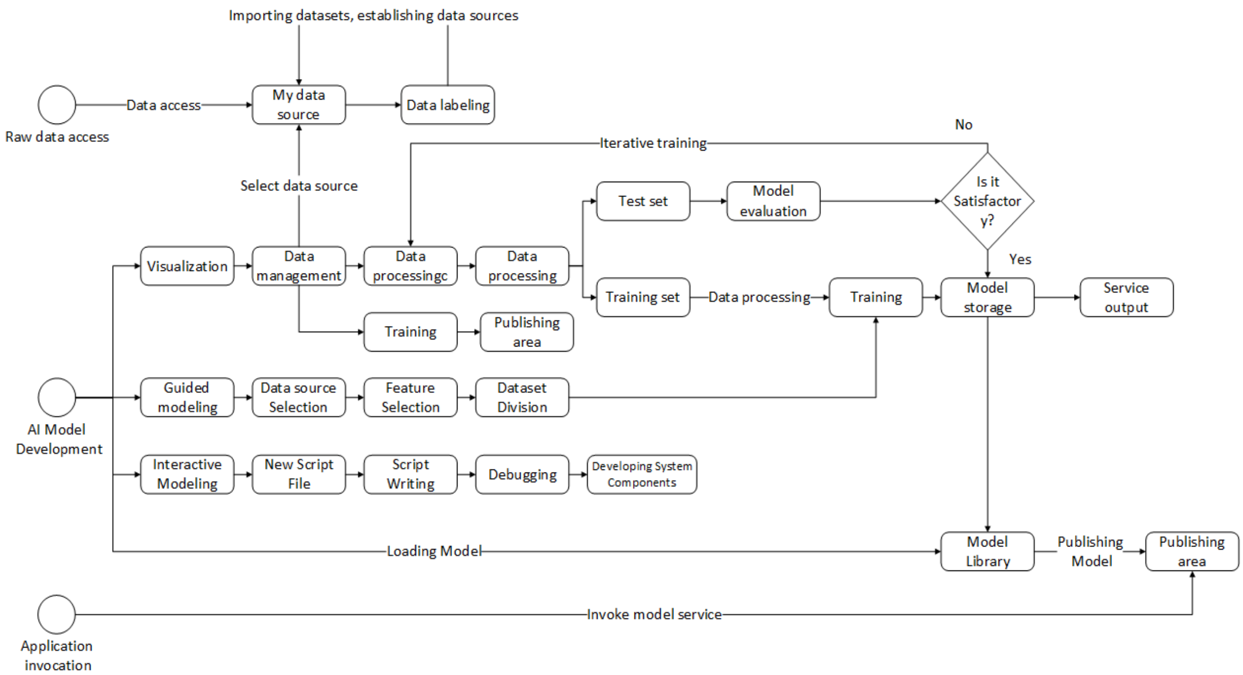

Figure 5 AI Model Development and Training Platform Business Processes

In addition, to address diverse application environments where no generic algorithm model fits all problems, the AI model development and training platform integrates a model library for managing historical data extraction rules and an algorithm pool for large-scale intelligent analysis. It offers subscription-based big data analysis and application services as per demand, enabling platform users to access resources from the model library and algorithm pool according to their business requirements for conducting business intelligence analysis. Currently, from the perspective of practical data applications, the algorithm library covers over 40 built-in algorithm components, including modeling assistant algorithms, deep learning algorithms, classic machine learning algorithms, and traditional optimization algorithms.

2.4 LLM Based AutoML Toolchain

The AutoML toolchain based on Large Language Models (LLMs) provides an automated solution for processes such as data access, data preprocessing, model training, and model evaluation. It integrates algorithm libraries and model repositories from AI model development and training platforms. Leveraging the inference capabilities of LLMs in areas such as data processing, model structure design, and hyperparameter tuning, it rapidly constructs and deploys high-quality surrogate models, further meeting the demands of AI engineering.

The overall workflow of this toolchain is illustrated in Figure 6, consisting primarily of large language models such as GPT-4 and the AutoML workflow.

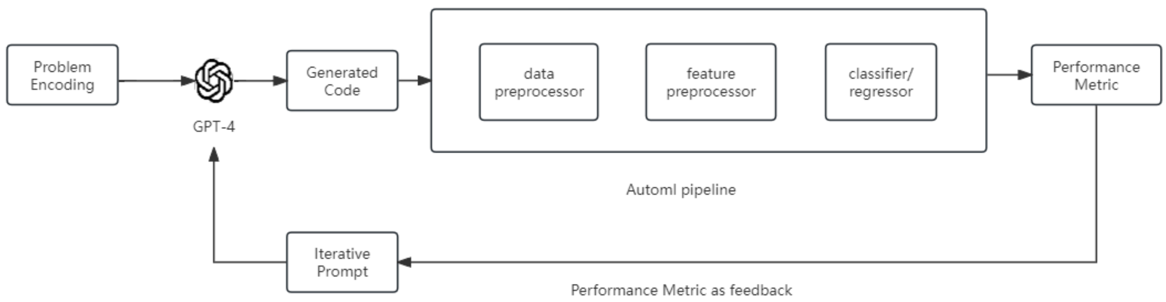

Figure 6 LLM based AutoML workflow

This AutoML toolchain aims to utilize GPT-4 as an optimizer to efficiently complete automated machine learning tasks. When handling each AutoML task, the first step is to convert the task description and data into a text format that language models can parse, such as the form of Context Awareness (Chain of Thought, CoT). These texts are input as prompts into pretrained large-scale language models, based on which the models generate AutoML configurations. The objective of these configurations is to construct an AutoML workflow that can optimize evaluation metrics on specific datasets to achieve optimal performance.

Subsequently, the generated configurations are used to train and evaluate the model. The performance evaluation metrics of the model are fed back to GPT-4, and based on this feedback and existing knowledge, the model generates an improved model configuration, thus entering the next iteration process.

Based on this process, the AutoML toolchain can rapidly generate surrogate models based on text descriptions and data, significantly enhancing the user-friendliness of the AutoML system. By introducing large language models (LLMs) as optimizers, the time and computational resources required for searching the optimal model structure in AutoML are significantly reduced.

Table 1 prompts in the AutoML toolchain

	Prompt
Problem Encoding	Problem Encoding Stage - You are a 'AutoML Expert' to help us throughout the entire AutoML process. This role would involve:1. Understanding the dataset,2. Preprocessing and feature engineering,3. Model selection and hyperparameter tuning,4. Evaluation and interpretation,5. Deployment and monitoring. By taking on this role, you can provide valuable guidance and expertise throughout the AutoML process, ensuring that you make informed decisions and obtain the best possible results from the models you build. Use '#' before every line except the python code. The goals would involve: 1. Load dataset from {{ *file* }} and preprocess the dataset and perform feature engineering. 2. Analyze the data if it's a classification problem or Regression problem. 3. Check if there is any non-numeric column, if there is any use pd.get_dummies(). 4. Split the preprocessed data into

	training and testing sets and use target column as {{ *target* }}. 5. Find the best model for our problem. 6. Evaluate the best model's performance on the testing set. 7. Get the best model python file for prediction on new data.
	Prompt
Iterative Prompt	Iterative Prompt Stage - Read {{ *model_code* }} file and update the code according to below info. By using this model, we achieved R-squared with {{ *R-squared* }}. Recommend a new model that outperforms prior architectures. Provide an explanation why the suggested model surpasses previous one. Use '#' before every line except the python code.

3.Case Study and Experiment Setup

3.1 Case Study

This paper selects a three-level synchronous generator model built using Simulink as a research example, as illustrated in Figure 7. The three-level synchronous generator typically consists of three main parts: a permanent magnet generator, an AC exciter, and a main generator. The permanent magnet generator adopts a rotating magnetic pole structure, while the AC exciter uses a rotating armature structure, and the main generator also uses a rotating magnetic pole structure.

The working process of the three-level synchronous generator can be summarized as follows: the winding of the permanent magnet generator generates three-phase AC power, which is rectified and used as the DC excitation power source for the AC exciter. The AC power generated by the three-phase armature of the AC exciter is converted into DC power under the action of the rotating rectifier installed on the rotor, and then serves as the excitation source for the excitation winding of the main generator. Finally, the armature of the main generator is located on the stator and can directly output three-phase AC power.

When building the model, the sampling time is set to 2×10^{-7} seconds. If the sampling time exceeds this value, the model will fail to converge. The reason for this phenomenon is that during multi-step and multi-iteration calculations, the sparse distribution of sample points leads to a low sampling frequency and a long sampling period, resulting in increased errors between the linear estimation and fitting process and the original real dynamic process, which may lead to problems of no solution or ill-conditioned solutions. To ensure the stable operation of the model, it is necessary to use a relatively short sampling time. However, excessively short sampling times significantly increase the simulation time, thereby increasing the time cost of platform integration simulation and reducing the efficiency of simulation. Therefore, this paper conducts comparative experiments between the original model running directly and the model running after optimization using surrogate models to highlight the significant advantages of surrogate models in terms of efficiency and cost-effectiveness.

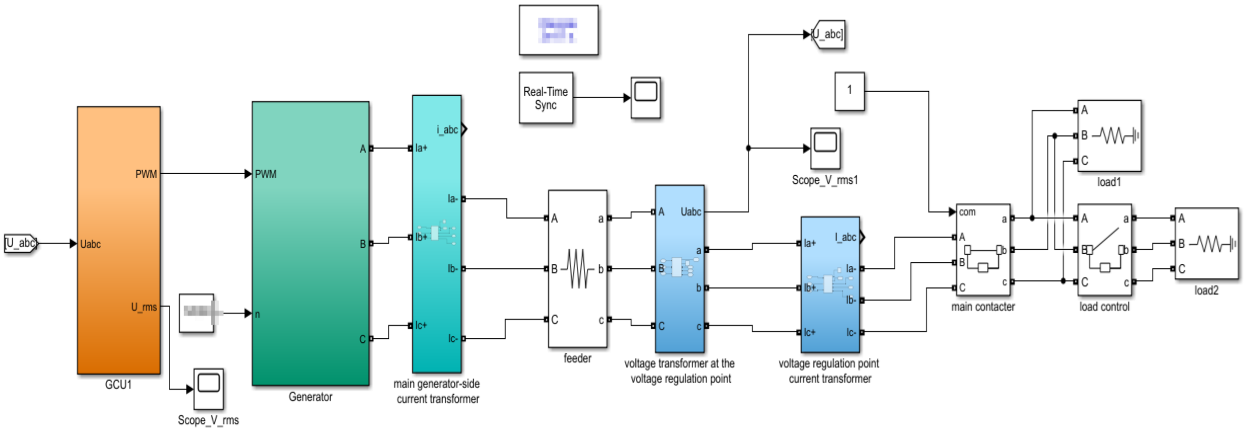

Figure 7 three-level synchronous generator mode

3.2 Objective

This study aims to deeply optimize motor models using "surrogate models" within the context of green aviation and model-driven development. Our goal is to construct a surrogate model using state-of-the-art machine learning algorithms to significantly improve simulation efficiency while ensuring model prediction accuracy (R^2 not less than 0.995). The objective is to achieve at least a 30-fold simulation acceleration, synchronizing model simulation time with real time, thereby greatly enhancing the practicality and responsiveness of simulation.

Within this interdisciplinary research framework, we will focus on the following aspects:

1. Green Aviation: Explore the potential of motor models in improving aircraft efficiency and reducing environmental impact. This includes optimizing motor performance to support more efficient electric propulsion systems, thus promoting the development of green aviation technology.

2. Model-Based Systems Engineering: Investigate how to accelerate the design and development process of motor control systems using surrogate models. We will assess the effectiveness of surrogate models in automating design processes, reducing development time and costs, and improving design quality.

In the process of building surrogate models, we will consider algorithm selection, optimization of feature engineering, and adjustment of model hyperparameters. Our goal is to develop a surrogate model that accurately reflects the dynamic characteristics of motor models while quickly responding to simulation demands. Additionally, we will explore different machine learning frameworks to determine the most suitable approach for our research needs.

Through this study, we anticipate not only providing an optimized motor model but also offering a new strategy and methodology for motor system design and model-driven development processes in the field of green aviation. The research outcomes will contribute to the optimization of aircraft electric propulsion systems, accelerate the application of model-driven development in the aviation sector, and contribute to achieving the goals of green aviation and sustainable development.

3.3 Experiment Setup

This study selects a specific Interior Permanent Magnet Synchronous Motor (IPMSM) module from the Variable Frequency Generator system (VFG) on aircraft as the subject of the case study to validate the effectiveness of the AutoML method in constructing surrogate models in real-world application scenarios. The validation process will be based on multiple dimensions, including the surrogate model's coefficient of determination (R-squared), inference speed, and required computational resources.

The coefficient of determination, R-squared, is a key statistical metric for measuring the accuracy of model predictions, reflecting the goodness of fit of the model to the data. R-squared values range between 0 and 1, with values closer to 1 indicating a stronger explanatory power of the model and better fit.

The experimental data for this study are derived from simulation of a three-level synchronous generator model. The simulation duration is set to 1 second, during which 200,000 data points are generated and recorded in the form of a time series. To evaluate the time cost of the simulation process, timing monitoring is performed during the simulation.

After obtaining the simulation output data, we use algorithms to train and learn from the data to extract key parameters in constructing the surrogate model. Subsequently, the obtained surrogate model is used to re-predict data within a 5-second time range. Since the prediction process of the surrogate model is very rapid, extending the prediction time can effectively reduce statistical errors caused by uncertainties such as compilation processes. The prediction time is averaged to estimate the time cost required for predicting 1 second. To improve the reliability of the results, the experiment is repeated 6 times, and the time cost and coefficient of determination R-squared obtained from each experiment are averaged to eliminate the influence of random errors.

In this experimental scenario, the motor model contains multiple dimensions of features. By gradually removing and adding features, multiple sets of experiments are designed to study the impact of different feature dimensions on the predictive performance of the surrogate model. This process not only helps reveal which features play a decisive role in the prediction results but also provides insights into feature selection for the model.

The hardware platform used in the experiment is the AMD Ryzen Threadripper PRO 5995WX, with 64 cores and a main frequency of 2.70GHz, as well as 64GB of RAM. When conducting machine learning training and inference, we ensure that the memory used does not exceed 4GB to simulate the performance of the model under resource-constrained environments.

4. Result and Analysis

To generate a surrogate model capable of fitting our dataset, we adopted an interactive approach. Specifically, we designed a detailed prompt that clearly specified the requirements for data analysis, data processing, and model generation. Then, we uploaded the prompt along with the data file to a large language model (LLM) to generate customized code. Using the code generated by the LLM, we conducted preliminary data analysis and surrogate model construction. In each iteration, we provided feedback through new prompts on the current model's fitting results and areas needing improvement. The LLM used this feedback to further optimize the code until satisfactory fitting results were achieved.

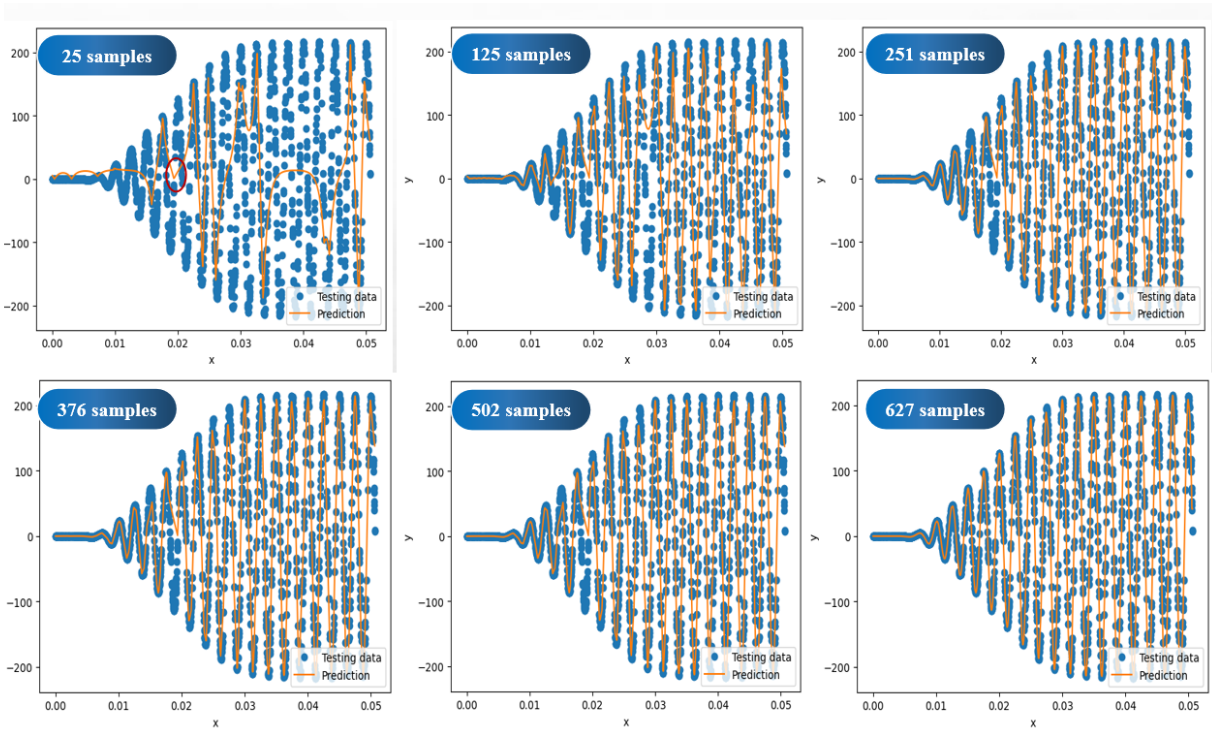

Figure 8 Multiple Sets of Surrogate Model Fitting Curves

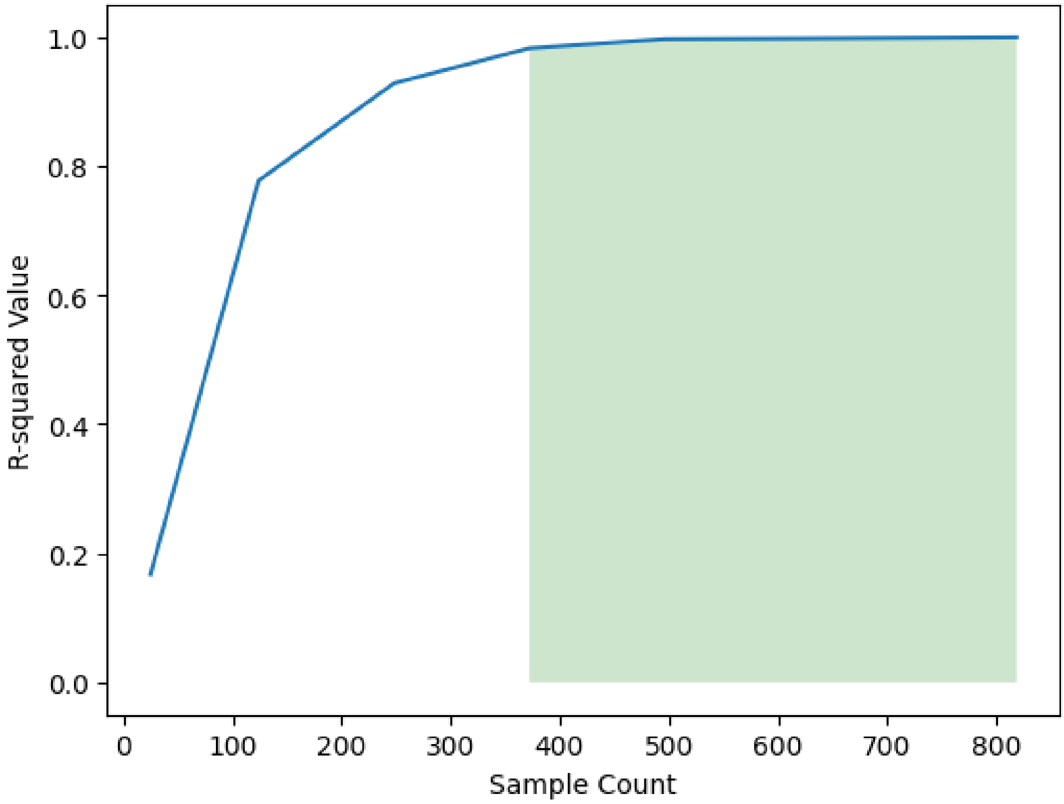

Figure 9 R^2 values with different sample count

In this study, we designed experiments to compare the performance of surrogate models with traditional Simulink simulations in terms of time cost and hardware resource consumption. The experimental results are as follows:

Model Fit Comparison: Figure 8 shows the model fitting curves for six different data volumes. It is evident from the figure that as the data volume increases, the predictive curve of the surrogate model becomes more consistent with the trend of the original dataset, proving that the surrogate model can maintain a high degree of fit with a certain data scale.

R^2 Value: In Figure 9, we not only show the trend of increasing R^2 values with increasing data volumes but also highlight a shaded area to mark a critical observation point. The shaded area indicates that when the sample size approaches 400 data points, the surrogate model's fit has at least 0.95 confidence. This observation further strengthens our confidence in the model's predictive ability, as it shows that even with a relatively small sample size, the surrogate model can provide highly reliable predictions. The figure clearly shows that as the data volume gradually increases, the R^2 value steadily rises, eventually stabilizing at a very high level of 0.9996. This excellent performance not only far exceeds our expectations but also, combined with the confidence indicated by the shaded area, further confirms that the surrogate model can maintain its superior accuracy under various data volume conditions. This finding is particularly important for practical applications because it means that even in situations where data acquisition is limited or costly, the surrogate model can still provide high-quality prediction results. Additionally, the high-confidence fit makes the surrogate model a powerful tool suitable for scenarios requiring fast and accurate predictions, such as green aviation and model-driven development.

Time Cost: We compared the time cost of completing a 1-second simulation task using the surrogate model and the traditional Simulink simulation. The surrogate model achieved a significantly better performance of 0.8571 seconds compared to the Simulink model's 535 seconds, achieving near-real-time simulation speed.

Hardware Resource Consumption: We compared the memory consumption of both methods when completing a 1-second simulation task. The surrogate model's memory consumption was about 240 MiB, whereas the Simulink model's memory consumption was as high as 3.3 GB, more than 14 times that of the surrogate model. This comparison highlights the significant advantage of the surrogate model in resource utilization efficiency.

These experiments not only visually demonstrate the advantages of surrogate models in simulation efficiency and resource consumption but also validate their potential in high-precision simulations. These results are significant for advancing the fields of green aviation and model-driven development.

5.Conclusion

This paper has effectively demonstrated the potential of leveraging Large Language Models (LLMs) and AutoML toolchains in the field of model-based system engineering, particularly within the context of green aviation. By integrating advanced machine learning algorithms and AI-optimized workflows, we have achieved a significant enhancement in simulation efficiency and accuracy through the development of high-fidelity surrogate models. Our case study on the IPMSM module within an airborne VFG system highlighted the transformative impact of these models, drastically reducing the model-in-loop solving time while maintaining a high degree of accuracy ($R^2 = 0.9996$).

The application of LLMs in automating the model reduction process has not only streamlined the design and development of airborne systems but also contributed to reducing the environmental footprint by enabling more energy-efficient simulations. The proposed semi-automatic model reduction method proved effective in tackling the challenges of high-dimensional model construction, showcasing a substantial improvement in computational efficiency and reduction of simulation time costs.

Furthermore, the practical implications of this research extend beyond green aviation, offering insights and methodologies that can accelerate the application of model-driven development across various sectors. The successful implementation of our toolchain represents a significant step forward in the automation of complex systems engineering, potentially revolutionizing the design and optimization of electromechanical systems.

In conclusion, the integration of AI technologies into the development of surrogate models offers a promising avenue for enhancing system performance and efficiency in aerospace applications. This research not only contributes to the field of green aviation but also sets a benchmark for future studies aiming to harness the power of artificial intelligence in engineering simulations.

REFERENCES

[1] W. Shenglai , W. Chuang , L. Yang , et al. Design and Implementation of Digital Integration Platform for Airborne System. Command Control and Simulation, https://link.cnki.net/urlid/32.1759.TJ.20231220.1402.008 . 2023

[2] N. Delaissé, P. Havaej, D. Fauconnier and J. Degroote. Accelerating the simulation of elastohydrodynamic lubrication in a line contact using a surrogate model. In 8th European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2022). 2022

[3] Magdalena, Peciak., Wojciech, Skarka. (2022). Assessment of the Potential of Electric Propulsion for General Aviation Using Model-Based System Engineering (MBSE) Methodology. Aerospace, 9(2):74-74. doi: 10.3390/aerospace9020074

[4] Pier, Davide, Ciampa., Björn, Nagel., Gianfranco, La, Rocca. (2020). A mbse approach to mdao systems for the development of complex products. 1-31. doi: 10.2514/6.2020-3150

[5] .Nabeel, Mahmood., Selcuk, Cimtalay., Dimitri, N., Mavris. (2022). Model Based Systems Engineering (MBSE) Applied to Fault Detection Analysis of Vehicle Subsystems. 1-11. doi: 10.1109/AERO53065.2022.9843533

[6] Maria, Cecilia, Perantoni, Fuchs., Yassine, Ghanjaoui., J., Abulawi., Jörn, Biedermann., Björn, Nagel. (2022). Enhancement of the virtual design platform for modeling a functional system architecture of complex cabin systems. CEAS Aeronautical Journal, 13(4):1101-1117. doi: 10.1007/s13272-022-00608-z

[7] Daewoon, Kim., Meric, Taneri., Ehiremen, N., Omoarebun., Michael, Balchanos., Dimitri, N., Mavris. (2023). MBSE-Enabled System Verification and Process Improvement of Transport Aircraft Certification. doi: 10.2514/6.2023-1897

[8] Vetrano, F., Worton, P., Smaoui, H., Szukala, A., Thukaram, S., Al-Logmani, A., ... & Mabboux, G. (2024). Recommendations on Increased Use of Modelling and Simulation for Certification/Qualification in Aerospace Industry. In AIAA SCITECH 2024 Forum (p. 1625).

[9] Novák, D., & Pukl, R. (2023, December). Advanced integrated assessment of concrete bridges considering nonlinearities, uncertainties and degradation. In AIP Conference Proceedings (Vol. 2950, No. 1). AIP Publishing.

[10] Ji, J. Y., Tan, Z., Zeng, S., See-To, E. W., & Wong, M. L. (2023). A Surrogate-Assisted Evolutionary Algorithm for Seeking Multiple Solutions of Expensive Multimodal Optimization Problems. IEEE Transactions on Emerging Topics in Computational Intelligence.

[11] Liu, H., Tian, Y., Gao, Y., Bai, J., & Zheng, J. (2015). System of systems oriented flight vehicle conceptual design: Perspectives and progresses. Chinese Journal of Aeronautics, 28(3), 617-635.

[12] Guo, F., Xiao, Q., Xiao, S., & Wang, Z. (2023). Analysis on quantifiable and controllable assembly technology for aeronautical thin-walled structures. Robotics and Computer-Integrated Manufacturing, 80, 102473.

[13] Himakar, Ganti., Prashant, Khare. (2020). Data-driven surrogate modeling of multiphase flows using machine learning techniques. Computers & Fluids, 211:104626-. doi: 10.1016/J.COMPFLUID.2020.104626

[14] Meng, Tang., Yimin, Liu., Louis, J., Durlofsky. (2020). A deep-learning-based surrogate model for data assimilation in dynamic subsurface flow problems. Journal of Computational Physics, 413:109456-. doi: 10.1016/J.JCP.2020.109456

[15] Xiao, Li., Fan, Yang., Dake, Wu., Zhenya, Zhou., Xuan, Zeng. (2015). MOS Table Models for Fast and Accurate Simulation of Analog and Mixed-Signal Circuits Using Efficient Oscillation-Diminishing

Interpolations. IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 34(9):1481-1494. doi:10.1109/TCAD.2015.2413392
[16] Baolin, Yang., Bruce, W., McGaughy. (2004). An essentially non-oscillatory (ENO) high-order accurate adaptive table model for device modeling. 864-867. doi: 10.1145/996566.996796
[17] Karp, F.H.S., Adamchuk, V., Dutilleul, P. et al. Comparative study of interpolation methods for low-density sampling. Precision Agric (2024). https://doi.org/10.1007/s11119-024-10141-0
[18] Yao, Q., Wang, M., Chen, Y., Dai, W., Li, Y. F., Tu, W. W., ... & Yu, Y. (2018). Taking human out of learning applications: A survey on automated machine learning. arXiv preprint arXiv:1810.13306.
[19] He, X., Zhao, K., & Chu, X. (2021). AutoML: A survey of the state-of-the-art. Knowledge-Based Systems, 212, 106622.
[20] Elsken, T., Metzen, J. H., & Hutter, F. (2019). Neural architecture search: A survey. The Journal of Machine Learning Research, 20(1), 1997-2017.
[21] Min, B., Ross, H., Sulem, E., Veyseh, A. P. B., Nguyen, T. H., Sainz, O., ... & Roth, D. (2023). Recent advances in natural language processing via large pre-trained language models: A survey. ACM Computing Surveys, 56(2), 1-40.
[22] Floridi, L., & Chiriatti, M. (2020). GPT-3: Its nature, scope, limits, and consequences. Minds and Machines, 30, 681-694.
[23] Ethayarajh, K. (2019). How contextual are contextualized word representations? Comparing the geometry of BERT, ELMo, and GPT-2 embeddings. arXiv preprint arXiv:1909.00512.
[24] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., ... & Scialom, T. (2023). Llama 2: Open foundation and fine-tuned chat models. arXiv preprint arXiv:2307.09288.
[25] Tornede, A., Deng, D., Eimer, T., Giovanelli, J., Mohan, A., Ruhkopf, T., ... & Lindauer, M. (2023). AutoML in the Age of Large Language Models: Current Challenges, Future Opportunities and Risks. arXiv preprint arXiv:2306.08107.
[26] Achiam, J., Adler, S., Agarwal, S., Ahmad, L., Akkaya, I., Aleman, F. L., ... & McGrew, B. (2023). Gpt-4 technical report. arXiv preprint arXiv:2303.08774.
[27] Touvron, H., Lavril, T., Izacard, G., Martinet, X., Lachaux, M. A., Lacroix, T., ... & Lample, G. (2023). Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971.
[28] Anil, R., Dai, A. M., Firat, O., Johnson, M., Lepikhin, D., Passos, A., ... & Wu, Y. (2023). Palm 2 technical report. arXiv preprint arXiv:2305.10403.
[29] Shen, Y., Song, K., Tan, X., Li, D., Lu, W., & Zhuang, Y. (2023). Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface. arXiv preprint arXiv:2303.17580.
[30] Zhang, S., Gong, C., Wu, L., Liu, X., & Zhou, M. (2023). AutoML-GPT: Automatic Machine Learning with GPT. arXiv preprint arXiv:2305.02499.
[31] Zheng, M., Su, X., You, S., Wang, F., Qian, C., Xu, C., & Albanie, S. (2023). Can GPT-4 Perform Neural Architecture Search?. arXiv preprint arXiv:2304.10970.