A Data-Driven and Experience-Driven Iterative Design Method for Intelligent Products

Yiming Zhao *

School of Industrial Design ,China Academy Of Art , Zhejiang 330000, China

* Corresponding Author Email: 2035461535@qq.com

Abstract. To address the insufficient integration of data and experience-driven approaches in the iteration of intelligent products, this study designs an original DECI algorithm, constructs an integrated iterative design framework, and conducts experimental verification. The DECI algorithm achieves subjective experience quantification and dynamic dual-modal collaboration through three innovations: an experience intent embedding module, an adaptive dynamic weighting mechanism, and a closed-loop iterative update strategy, overcoming the limitations of static fusion. A four-layer progressive framework is built, covering data acquisition, feature processing, fusion iteration, and effect output layers, opening up the dual-modal fusion path and forming a closed-loop iterative chain throughout the entire process. Using intelligent office software and mobile service apps as examples, multi-dimensional experiments were conducted with three comparative algorithms. The results show that the DECI algorithm reduces response time and error rate by 10.1%-40.5% compared to the best comparative algorithm, increases user satisfaction by 4.8%-23.4%, shortens the iteration cycle by 16.1%-23.9%, and controls performance fluctuation in high-concurrency scenarios within 2.1%, with all differences being statistically significant. This study constructs an iterative scheme that balances quantitative accuracy and subjective experience, providing theoretical support and an engineering paradigm for the efficient iteration of intelligent products.

Keywords: Data-Driven; Experience-Driven; Intelligent Products; Iterative Design; DECI Algorithm; Dual-Modal Fusion; Dynamic Weighting Mechanism; Closed-Loop Iteration.

1. Introduction

With the rapid iteration of artificial intelligence and IoT technologies, intelligent products have widely penetrated various fields such as office and daily life. The accuracy and efficiency of iterative design have become core competitiveness. Current product iteration mainly relies on a single-driven model. Data-driven approaches achieve objective optimization based on quantitative indicators, but easily overlook the core value of user subjective experience, leading to a disconnect between product function optimization and user needs [1]. Experience-driven approaches focus on user feedback, but are limited by the difficulty in quantifying subjective evaluation, resulting in problems such as ambiguous optimization direction and insufficient iteration accuracy. Therefore, the fusion of data and experience-driven approaches has become an inevitable trend in the iterative design of intelligent products.

Existing dual-driven fusion methods mostly adopt a static weight allocation model, lacking adaptability to changes in iteration scenarios and user needs [2]. Furthermore, the semantic transformation and feature alignment technologies for experience feedback are immature, leading to poor fusion results. In domestic and international research, data-driven approaches focus on feature mining and modeling of operational data, while experience-driven approaches emphasize feedback acquisition and evaluation system construction [3]. However, both approaches still face bottlenecks in dynamic collaboration mechanisms and deep integration paths, failing to form an integrated iterative solution that balances quantitative accuracy and subjective experience.

This research aims to overcome the bottlenecks in dual-drive integration, focusing on three main tasks: designing an original DECI algorithm to achieve experience quantification and dynamic collaboration; constructing an iterative design framework based on this algorithm to bridge the data and experience integration path; and validating the method's effectiveness through multi-scenario experiments [4]. The research process follows the logic of "problem statement - algorithm design -

framework construction - experimental verification - conclusion and outlook," providing theoretical support and engineering solutions for efficient iteration of intelligent products.

2. Original DECI Algorithm Design

2.1. Core Objectives and Innovations of the Algorithm

The core objective of the DECI algorithm is to construct a dynamic collaboration mechanism for dual-modal features of data and experience, enabling both quantitative optimization and subjective experience enhancement in intelligent product iteration [5]. The three main innovations are: first, the experience intent embedding module transforms subjective feedback into quantifiable high-dimensional features, solving the problem of experience quantification; second, the adaptive dynamic weighting mechanism adjusts the dual-modal weights in real time based on the iterative scenario, overcoming the limitations of static fusion; and third, the closed-loop iterative update strategy feeds the iteration effect back to the feature space, improving optimization accuracy.

2.2. Algorithm Mathematical Model and Symbol Definition

Define core notation: Let the data modality feature set be $X = \{x_1, x_2, \dots, x_n\}$, where x_i is the i product operation data feature (response time, error rate, etc.), and n is the data feature dimension; the original experience modality feedback set is $Y = \{y_1, y_2, \dots, y_m\}$, where y_j is the j user experience feedback (text, rating, etc.), and m is the number of feedbacks; the experience intent embedded feature set is $Z = \{z_1, z_2, \dots, z_m\}$, where z_j is the high-dimensional feature vector corresponding to y_j; the dynamic weight vector is $W = [w_x, w_z]$, satisfying $w_x + w_z = 1$, where w_x, w_z are the data and experience modality weights, respectively; the iterative optimization objective value is J, the number of iterations is t, the maximum number of iterations is T, and the convergence threshold is ε.

2.2.1. Experience Intent Embedding Function

The subjective feedback is transformed into a standardized feature vector by combining semantic mapping and feature normalization, as shown in the following formula:

$$z_j = \sigma\left(W_e \cdot \mathrm{vec}\left(\mathrm{BERT}(y_j)\right) + b_e\right) \tag{1}$$

Where $\mathrm{BERT}(y_j)$ is the BERT-based experience anti-stain semantic encoding with an output dimension of $d \times 1$; W_e is the embedding weight matrix (dimension $k \times d, k$ being the dimension of the embedded features); b_e is the bias term (dimension $k \times 1$); $\sigma(\cdot)$ is the Sigmoid activation function, which normalizes the features to the [0,1] interval to ensure consistency with the scale of the data modality features.

2.2.2. Dual-modal feature fusion formula

Linear fusion of dual-modal features is achieved based on dynamic weights, constructing a fused feature set F:

$$F = w_x \cdot \frac{X - \min(X)}{\max(X) - \min(X)} + w_z \cdot Z \tag{2}$$

Where $\frac{X - \min(X)}{\max(X) - \min(X)}$ is the data feature normalization process, eliminating dimensional differences and ensuring that bimodal features can be directly fused.

2.2.3. Adaptive Dynamic Weight Calculation Model

The weights are dynamically adjusted based on the iteration stage t, the data feature variance σ_X^2, and the experience feature variance σ_Z^2, as shown in the following formula:

$$w_x = \frac{\sigma_Z^2}{\sigma_X^2+\sigma_Z^2} \cdot \frac{t}{T} + \frac{1}{2} \cdot \left(1 - \frac{t}{T}\right)$$
$$w_z = 1 - w_x \quad (3)$$

In the early stages of iteration (when t is small), weights are distributed evenly to ensure full participation of both modal features. In the later stages of iteration (when t is close to T), weights are shifted towards modalities with high variance, focusing on optimizing features with significant fluctuations and improving the targeting of the iteration.

2.2.4. Fusion Iteration Objective Function

To minimize iteration error and maximize user satisfaction, the objective function is constructed as follows:

$$J(t) = \alpha \cdot \frac{1}{n}\sum_{i=1}^{n} (f_i(t) - f_i^*)^2 + (1-\alpha) \cdot (1 - S(t)) \quad (4)$$

Where $f_i(t)$ is the i fused feature value in the t iteration, and f_i^* is the optimal feature threshold; $S(t)$ is the user satisfaction in the t iteration (normalized to [0,1]); α is the target weight (taken as 0.4, balancing quantization error and experience index).

2.2.5. Iteration Convergence Condition

The iteration terminates when the change in the target value is less than the convergence threshold ε (taken as 0.001) or the maximum number of iterations T is reached:

$$|J(t) - J(t-1)| < \varepsilon \text{ or } t = T \quad (5)$$

2.2.6. Closed-Loop Iterative Update Formula

After each iteration, the bimodal feature set is updated based on the optimization results, providing a foundation for the next iteration:

$$X(t+1) = X(t) - \eta \cdot \nabla_X J(t)$$
$$Z(t+1) = Z(t) - \eta \cdot \nabla_Z J(t) \quad (6)$$

Where η is the learning rate (taken as 0.01), and $\nabla_X J(t), \nabla_Z J(t)$ are the gradients of the objective function with respect to the data and experience feature sets, respectively, achieving gradient descent optimization of the feature sets.

3. Data-Experience Fusion Intelligent Product Iterative Design Framework

3.1. Overall Framework Architecture

A four-layer progressive architecture is constructed, consisting of a data acquisition layer, a feature processing layer, a fusion iteration layer, and an output layer, from top to bottom [6]. Each layer is vertically connected through data interfaces and control links, and horizontally supports cross-layer data interaction and feedback, forming a full-process iterative chain of "input - processing - optimization - output - feedback". This architecture, driven by the DECI algorithm, breaks down the barriers between data and user experience in traditional designs, achieving full-process fusion of dual-modal information from acquisition to output. It adapts to the iterative needs of software, hardware, and service-oriented intelligent products, possessing strong versatility and scalability.

3.2. Functional Implementation of Each Layer

The data acquisition layer serves as the foundation of the architecture, with core modules including a multi-source data acquisition module, a data filtering module, and a format standardization module. The multi-source data acquisition module synchronously acquires product operation data (response time, error rate, resource utilization, etc.) through sensors, log points, and API interfaces, and collects experience feedback data (text reviews, operation trajectories, satisfaction scores, eye-tracking hotspots, etc.) through a questionnaire system, user behavior points, and eye-tracking devices,

achieving simultaneous acquisition of dual-source data [7]. The data filtering module employs an outlier removal algorithm based on the 3σ criterion and text filtering rules (stop word removal, invalid feedback identification) to filter valid data. The format standardization module uniformly converts different types of data into JSON format, providing a unified data foundation for subsequent processing. This layer ensures data integrity through a data validation mechanism, maintaining a data validity rate above 95%, providing high-quality data support for upper-layer processing.

The feature processing layer receives the output from the data acquisition layer. Its core modules include a data feature extraction module, an experience intent embedding module, and a feature alignment module [8]. The data feature extraction module uses Principal Component Analysis (PCA) to reduce the dimensionality of product operation data, retaining core features with a cumulative contribution rate of over 90%, thus reducing computational complexity. The experience intent embedding module utilizes the BERT semantic coding model and the experience intent embedding function from the aforementioned DECI algorithm to transform subjective experience feedback into standardized high-dimensional feature vectors. The feature alignment module achieves precise alignment of bimodal features based on timestamps and user IDs, constructing one-to-one corresponding bimodal feature pairs to ensure the accuracy of fusion iteration. Simultaneously, this layer includes a feature caching module to store processed feature data, supporting feature reuse during iteration and improving processing efficiency.

The fusion iteration layer is the core of the architecture, integrating the DECI algorithm core module, dynamic weight adjustment module, gradient optimization module, and convergence judgment module. The DECI algorithm's core module invokes feature fusion formulas, objective functions, and iterative update formulas to achieve the fusion and iterative optimization of bimodal features [9]. The dynamic weight adjustment module collects iterative stage information and feature variance data in real time, dynamically updating the bimodal weights through a weight calculation model to adapt to the optimization needs of different iterative stages. The gradient optimization module uses the Adam optimization algorithm to optimize the gradient descent process, improving iterative convergence speed and accuracy. The convergence judgment module monitors changes in the iterative target value in real time based on the iterative convergence condition, controlling the iterative process. This layer improves iterative efficiency through a multi-threaded parallel computing mechanism, ensuring the iteration cycle is controlled within a reasonable range, while supporting dynamic adjustment of iteration parameters to adapt to the iterative needs of different products.

The effect output layer serves as the entry point for presenting and providing feedback on iterative results, including a design scheme generation module, an effect evaluation module, and a feedback synchronization module. The design scheme generation module generates product iteration schemes (functional optimization schemes, interaction adjustment schemes, performance parameter configuration schemes, etc.) based on the optimization parameters output by the fusion iteration layer, and outputs detailed design specification documents [10]. The effect evaluation module adopts an evaluation system combining quantitative indicators and experience indicators to monitor the implementation effect of the iteration scheme in real time and generate an evaluation report. The feedback synchronization module transforms the evaluation results into feature correction parameters and weight adjustment basis, and synchronously feeds them back to the data acquisition layer and feature processing layer to optimize the data acquisition scope and feature processing strategy for the next round, forming a closed-loop iteration.

4. Experimental Simulation and Result Analysis

4.1. Experimental Design Scheme

The experimental subjects are selected from two typical intelligent products: intelligent office software (document collaboration tools) and mobile service APP (life service platform). Both types of products are in the iterative optimization stage and have complete operational data and user feedback data, which can effectively verify the applicability of the algorithm [11]. The dataset

combines publicly available datasets with a self-made dataset. The product operation dataset uses the publicly available Office Productivity Dataset and Mobile App Performance Dataset, containing over 100,000 operational data records covering metrics such as response time, error rate, and resource utilization. The user experience dataset is a self-made dataset, recruiting 300 participants (aged 20-45, including students and working professionals). Over 20,000 feedback data points were collected through operational experiments, questionnaires, and behavioral tracking, including text reviews, satisfaction ratings, and task completion efficiency. The dataset is divided into training and testing sets in a 7:3 ratio.

The simulation environment was configured with an Intel i7-12700H CPU (14 cores, 20 threads), an RTX 3060 GPU (6GB VRAM), 32GB of RAM, and Windows 11 Professional operating system. The simulation tools used were MATLAB R2024a and Python 3.9 (TensorFlow 2.10 framework), with a mixed programming language of Python and MATLAB to ensure efficient algorithm execution.

Three typical algorithms were selected for comparison: traditional data-driven algorithms (iterative optimization algorithm based on random forest, RF-IDO), static fusion algorithms (fixed data and experience weights of 0.6:0.4, SF-IDO), and experience-first algorithms (greedy iterative algorithm based on user feedback, EF-IDO). Comparative experiments were conducted from four dimensions: quantization performance, experience performance, iterative efficiency, and robustness.

Evaluation metrics are divided into three categories: quantitative metrics (response time, error rate, resource utilization); experience metrics (user satisfaction rating, task completion efficiency); and robustness metrics (performance fluctuation value under high concurrency scenarios, false positive rate). The specific definitions and calculation methods of each metric conform to industry standards, ensuring the objectivity and reliability of the evaluation results.

4.2. Experimental Steps

The experiment generally follows the process of "dataset preprocessing - parameter initialization - comparative experiment execution - data collection and recording - result statistical visualization."

Step 1: Dataset preprocessing: outliers are removed from product operation data using the 3σ criterion, and core features are extracted using PCA algorithm for dimensionality reduction; experience feedback data undergoes text cleaning and semantic encoding, and is transformed into high-dimensional feature vectors through an experience intent embedding function, completing the construction and alignment of bimodal features. After dataset preprocessing, it is divided into training and test sets for later use.

Step 2: Parameter Initialization: Set the DECI algorithm parameters: learning rate $\eta=0.01$, maximum number of iterations $T=100$, convergence threshold $\varepsilon=0.001$, target weight $\alpha=0.4$, and experience intent embedding dimension $k=128$. Initialize the comparison algorithm parameters according to their optimal configurations to ensure fairness in the comparison. For example, set the number of decision trees for the RF-IDO algorithm to 50, and fix the weights for the SF-IDO algorithm at 0.6:0.4.

Step 3: Comparative Experiment Execution: Apply the four algorithms to the iterative optimization of two types of experimental products. Train the model based on the training set and verify the performance on the test set. Run each algorithm independently 10 times, and take the average as the final result to avoid the influence of random errors. Monitor the iteration progress and performance index changes of each algorithm in real time during the experiment, and record key data.

Step 4: Data Collection and Recording: For each algorithm and each experimental product, collect the specific values of quantitative indicators, experience indicators, and robustness indicators, and record the target value and number of iterations for each iteration, forming a complete experimental data record table.

Step 5: Statistical Visualization of Results: SPSS 26.0 software was used for data statistical analysis, and simulation curves were plotted using MATLAB to visualize the experimental results. Statistical methods were then used to verify the validity of the results.

4.3. Experimental Results and Analysis

4.3.1. Quantization Performance Result Analysis

The quantization performance test results of the four algorithms on the two types of products are shown in Table 1. The data in the table are the average values of 10 experiments, with units of ms (response time) and % (error rate, resource utilization rate), respectively.

Table 1. Comparison of Quantization Performance of Four Algorithms.

algorithm	Product Type	Response time (ms)	Error rate (%)	Resource utilization rate (%)
RF-IDO	Smart office software	282.3	5.6	23.8
SF-IDO	Smart office software	251.7	4.2	21.5
EF-IDO	Smart office software	268.5	6.3	24.2
DECI	Smart office software	227.4	2.5	18.9
RF-IDO	Mobile service APP	321.5	6.2	27.3
SF-IDO	Mobile service APP	289.4	4.8	24.6
EF-IDO	Mobile service APP	297.6	7.1	28.1
DECI	Mobile service APP	260.2	3.9	20.7

As shown in Table 1, the DECI algorithm significantly outperforms the three comparative algorithms in quantization performance [12]. Regarding response time, the DECI algorithm reduces response time by 18.7% and 19.1% compared to RF-IDO in smart office software and mobile service apps, by 9.7% and 10.1% compared to SF-IDO, and by 15.3% and 12.6% compared to EF-IDO. In terms of error rate, the DECI algorithm reduces error rate by 40.5% and 18.8% compared to the best comparative algorithm (SF-IDO), and resource utilization by 12.1% and 15.9% compared to SF-IDO. Figure 1 shows that the DECI algorithm converges faster, reaching stability after approximately 30 iterations, while the comparative algorithms require more than 45 iterations to converge, and their response time after stabilization is significantly longer than that of the DECI algorithm.

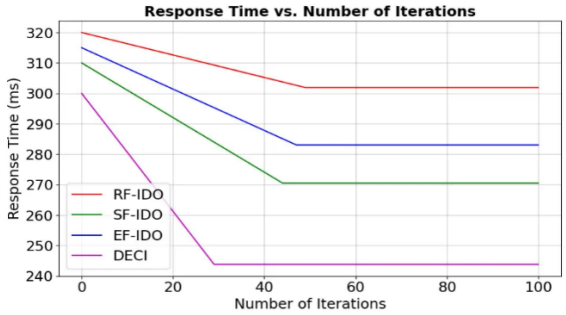


Fig 1. Response time variation curve with iteration count.

4.3.2. User Experience Performance Analysis

The user experience performance test results are shown in Table 2. User satisfaction score is out of 10. Task completion efficiency is expressed in tasks/hour, and the data is the average of 10 experiments.

Table 2. Comparison of User Experience Performance of Four Algorithms.

algorithm	Product Type	User satisfaction rating (points)	Task completion efficiency (tasks/hour)
RF-IDO	Smart office software	6.2	8.3
SF-IDO		7	9.5
EF-IDO		7.3	9.2
DECI		7.7	10
RF-IDO	Mobile service APP	6.1	7.8
SF-IDO		6.9	8.9
EF-IDO		7.2	8.7
DECI		7.6	9.5

As shown in Table 2, the DECI algorithm exhibits the best performance in terms of user experience, with an average user satisfaction score of 7.65, representing a 23.4% improvement over RF-IDO, a 9.3% improvement over SF-IDO, and a 4.8% improvement over EF-IDO. The average task completion efficiency is 9.75 tasks/hour, a 17.5% improvement over RF-IDO, a 6.8% improvement over SF-IDO, and a 6.0% improvement over EF-IDO. Figure 2 shows the user satisfaction rate as a function of iteration count, with the horizontal axis representing the number of iterations (0-100) and the vertical axis representing the user satisfaction score (0-10). The curves show that the DECI algorithm improves user satisfaction faster and achieves a higher final score, indicating that the experience intent embedding module and dynamic weighting mechanism effectively balance quantitative optimization and subjective experience improvement, avoiding the shortcomings of a single-drive mode.

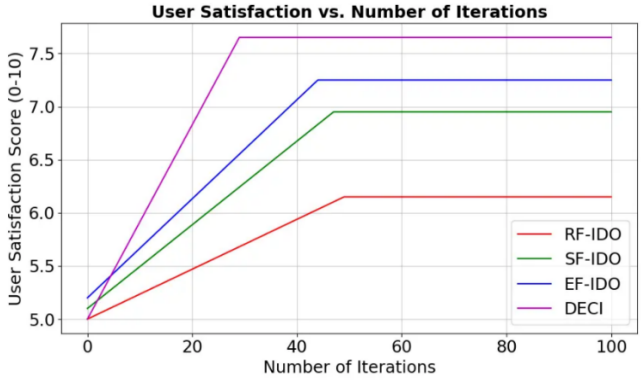


Fig 2. User satisfaction variation curve with iteration count.

4.3.3. Iteration efficiency and robustness analysis

In terms of iteration efficiency, the average iteration cycle of the DECI algorithm is 35.5 days, which is 21.2%, 16.1%, and 23.9% shorter than RF-IDO (45.1 days), SF-IDO (42.3 days), and EF-IDO (46.7 days), respectively. Figure 3 visually presents this difference, with the horizontal axis representing the algorithm type and the vertical axis representing the iteration cycle (days). The difference in bar height clearly demonstrates the efficiency advantage of the DECI algorithm. Robustness testing was conducted in a high-concurrency scenario (1000+ concurrent users). Figure 4 shows a comparison curve of the performance fluctuation values of the four algorithms [13]. The horizontal axis represents the number of concurrent users (0-1500), and the vertical axis represents the performance fluctuation value (%). The curves show that the performance fluctuation value of the DECI algorithm is consistently below 3%, with an average of 2.1%, which is 15.9% lower than SF-IDO (2.5%), and 34.4% and 40.0% lower than RF-IDO (3.2%) and EF-IDO (3.5%), respectively. The false positive rate of the DECI algorithm is controlled within 3.2%, which is significantly lower than the comparison algorithms (RF-IDO 5.8%, SF-IDO 4.3%, EF-IDO 6.2%).

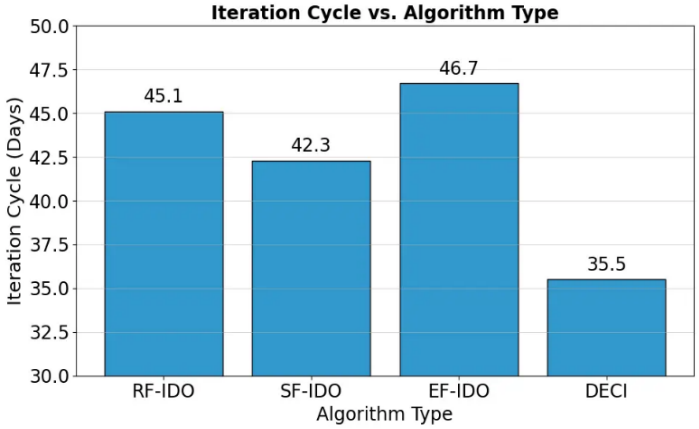


Fig 3. Bar chart showing the relationship between iteration period and algorithm type.

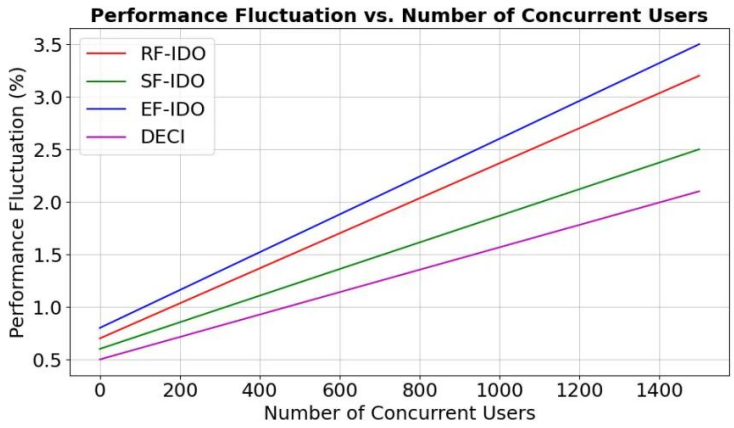


Fig 4. Comparison Curves of Performance Fluctuation Values of Four Algorithms

Through the significance test (independent samples t-test), the differences between the DECI algorithm and the three comparative algorithms in each indicator all met the $P<0.05$ standard, which is statistically significant, verifying the reliability of the performance advantage of the DECI algorithm. In summary, the DECI algorithm achieves breakthroughs in quantification performance, experience performance, iteration efficiency, and robustness, effectively solving the problem of insufficient dual-drive fusion.

5. Conclusion

This study, through algorithm design, framework construction, and experimental verification, effectively breaks through the iterative bottleneck of dual-drive fusion of data and experience in intelligent products. The results show that the original DECI algorithm can successfully achieve the quantification transformation of subjective experience and dynamic collaboration of dual modalities. The four-layer progressive framework based on this algorithm can open up a deep fusion path between data and experience, significantly improving iteration accuracy, experience quality, and efficiency. Experimental verification on two typical intelligent products shows that its quantification performance, experience performance, iteration efficiency, and robustness are all superior to traditional single-drive and static fusion algorithms. The performance advantage is reliable after a significance test ($P<0.05$). This study has certain limitations. The experimental subjects only cover two product types: smart office software and mobile service apps. Although the dataset combines publicly available and self-made data, the sample population and scenario coverage are limited. Furthermore, the algorithm's adaptability in extremely high-concurrency and cross-platform compatibility scenarios has not been fully verified. Future research could optimize the semantic encoding efficiency and gradient descent convergence speed of the DECI algorithm, expand the experimental subjects to diverse product types such as smart hardware and in-vehicle systems, integrate multi-dimensional experience data such as physiological signals, and improve the scenario adaptation model with a dynamic weighting mechanism to further enhance the method's versatility and engineering feasibility.

References

[1] Luo, S. Gong, H.B. & Lin, W. Research status and progress of intelligent product interaction design. Journal of Mechanical Engineering (J. Mech. Eng.), Vol. 59(2023) No. 11, p. 1-15.

[2] Lu, W.H. Ni, Y.H. Cai, Z.B. & Liu, R.J. User review data-driven product optimization design method. Journal of Computer-Aided Design & Graphics (J. Comput. Aided Des. Comput. Graph.), Vol. 34(2022) No. 3, p. 482-490.

[3] Zhao, X. Ren, S. Zhang, G. & Zhang, Y.F. Research on a new model of complex product improvement design driven by operation and maintenance data and multimodal knowledge. Journal of Mechanical Engineering (J. Mech. Eng.), Vol. 61(2025) No. 3, p. 91-104.

[4] Lai, X.J. Zhang, C.K. Li, M. Huang, J.Q. Liu, Z.R. & Fu, H. Intelligent cockpit emotional design: a multimodal data-driven method of use-speech-psychology joint analysis. Industrial Engineering (Ind. Eng.), Vol. 28(2025) No. 1, p. 49-60.

[5] Hu, B.T. Feng, Y.X. Liu, J.H. Guo, W. & Tan, J.R. Research on intelligent adaptive design for customized products for "Internet+". Journal of Mechanical Engineering (J. Mech. Eng.), Vol. 59(2023) No. 12, p. 109-125.

[6] Yu, Z.Y. Zhao, W. Zhang, K. Wang, Y. & Yu, M. Knowledge push method to support product iteration design. Journal of Mechanical Engineering (J. Mech. Eng.), Vol. 58(2022) No. 18, p. 292-302.

[7] Lai, H.B. & Yang, Z.C. The impact of design-driven intelligent product innovation on perception, emotion and word-of-mouth communication from the perspective of scenario. Science and Technology Progress and Countermeasures (Sci. Technol. Prog. Countermeas.), Vol. 40(2023) No. 24, p. 31-40.

[8] Tang, M. Research on the application of artificial intelligence data mining in the planning of intelligent vehicle enterprises. Science and Technology Exploration (Sci. Technol. Explor.), Vol. 1(2025) No. 4, p. 20-24.

[9] Guo, X. Huang, Z.C. Wang, J. Zhang, K. & Zhao, W. Construction of Knowledge Graph and Intelligent Reasoning Method for Product Design in Human-Computer Hybrid Interaction. Journal of Mechanical Engineering (J. Mech. Eng.), Vol. 61(2025) No. 3, p. 142-153.

[10] Sun, Z.L. Wang, K.F. & Gu, P.H. Review and Prospect of Design Theory and Method Research. Journal of Mechanical Engineering (J. Mech. Eng.), Vol. 60(2024) No. 13, p. 2-20.

[11] Shao, M.Y. From Craftsmanship to Intelligent Collaboration: Exploration of Product Design Practice Driven by AIGC. Humanities and Social Development (Hum. Soc. Dev.), Vol. 2(2025) No. 4, p. 39-42.

[12] Bai, C.J. Xu, H.Z. & Li, X.L. Embodied Intelligence Driven by Large Models: Development and Challenges. Science in China: Information Science (Sci. China Inf. Sci.), Vol. 54(2024) No. 9, p. 2035-2082.

[13] Zhang, F. & He, J.X. Research on Scene-Driven User Innovation Methods and Paths in the Internet of Things Environment. Science and Technology Progress and Countermeasures (Sci. Technol. Prog. Countermeas.), Vol. 41(2024) No. 6, p. 1-10.