



# OPEN Intelligent data-driven system for mold manufacturing using reinforcement learning and knowledge graph personalized optimization for customized production

Chengcai He, Jiaxing Deng, Jingchun Wu, Beicheng Qin✉, Jinxiang Chen, Yan Li & Qiangsheng Huang

Traditional manufacturing models are heavily dependent on standardized processes, which makes it challenging to accommodate customized production needs. To address this limitation, this study presents an optimized mold digitalization system grounded in knowledge engineering. The proposed system integrates knowledge graphs with intelligent algorithms to support the development of a smart quality control framework tailored to personalized manufacturing. This study places particular emphasis on the knowledge engineering module within the digitalization system. It focuses on converting domain-specific expertise into computable models that can be applied in real-world manufacturing scenarios. Additionally, reinforcement learning and graph neural networks are used to efficiently extract and utilize manufacturing knowledge. The experimental results reveal two key findings: (1) Within the enhanced learning knowledge graph framework, the algorithm—optimized using a graph convolutional network—achieves consistently higher qualification rates across test samples. When actual qualification rates exceed 88.1%, the model's regression fit also surpasses 88.1%, indicating strong alignment between predicted and actual performance. (2) Compared with other algorithmic models, the proposed approach achieves a predictive accuracy of over 94.7%. Overall, the proposed system significantly improves the level of customization in mold manufacturing while enhancing production efficiency and maintaining quality standards. The outcomes offer a new direction for the digital transformation of the manufacturing sector. Moreover, the approach holds practical value for enabling intelligent, flexible production processes, helping manufacturers better meet the growing demand for personalized products.

**Keywords** Mold intelligent data-driven system, Personalization, Reinforcement learning, Knowledge engineering, Knowledge graph

As technology advances and society evolves, consumer demand for personalized and customized products continues to grow<sup>1,2</sup>. These products not only align more closely with individual preferences but also increase product value and market competitiveness<sup>3</sup>. However, traditional supply chains and production models are increasingly unable to meet these evolving needs. They are often characterized by long lead times, high operational costs, limited flexibility, and slow adoption of new technologies—factors that limit their efficiency and adaptability in today's competitive market environment<sup>4</sup>. To address the rising demand for personalization and respond to rapid market shifts, the mold design and manufacturing industry must adopt more flexible, digitalized, and intelligent production models<sup>5,6</sup>. These new models promise to enhance production efficiency, improve product quality, and increase an enterprise's responsiveness to market changes—ultimately enabling more effective delivery of customized solutions<sup>7</sup>.

Shenzhen Ruipengfei Mold Co., Ltd, Shenzhen City 518000, China. ✉email: qbc1717@163.com

While personalized recommendation systems have seen widespread use in domains such as e-commerce, social media, and content distribution—with notable successes—there remains a significant gap in their application to mold digitalization systems. Existing techniques, such as collaborative filtering and content-based recommendation, often struggle with the complexity of industrial datasets and the demands of real-time performance. Consequently, they fall short in meeting the specific requirements of personalized customization in the mold manufacturing context<sup>8</sup>. Additionally, there is limited research on how emerging technologies—such as reinforcement learning and knowledge graphs—can be effectively integrated into mold digitalization systems to improve recommendation outcomes<sup>9,10</sup>.

In response to this gap, the present study proposes a novel approach that integrates a lightweight Graph Convolutional Network (LightGCN) with a Q-learning reinforcement learning framework. This combination supports the construction of a dynamic knowledge graph capable of efficiently extracting and continuously updating manufacturing knowledge. Key elements such as process parameter relationships and defect pattern inference are included in this framework. At the theoretical level, the model incorporates a multi-layer graph convolutional embedding mechanism. This structure captures high-order collaborative signals from user–mold interactions, resulting in notable improvements in both recommendation coverage and quality prediction accuracy. In practical terms, the method is validated using a dataset of automotive components. The results showed that the proposed model achieved an average qualification rate prediction error of less than 3.5% across six test sample categories. This represents a performance improvement of more than 15% compared to traditional regression models. Overall, this approach provides a scalable and effective technological pathway for building a closed-loop optimization system that spans design, production, and quality inspection in the mold manufacturing industry.

## Literature review

Personalized recommendation systems have been widely applied in e-commerce, social media, content distribution, and other fields. With the rapid development of big data and artificial intelligence (AI) technologies, research on personalized recommendation systems has made significant advances<sup>11</sup>. Collaborative filtering is one of the most classic and commonly used methods in personalized recommendation systems, which discovers similarities among users based on user behavior data to provide personalized recommendations. Nguyen et al. (2023) introduced latent semantic models and matrix factorization techniques to extract features from the hidden semantic information by decomposing the user-item interaction matrix, thereby enhancing the accuracy and computational efficiency of recommendation systems<sup>12</sup>. Content-based recommendation methods analyze not only users' historical behaviors but also the intrinsic features of content such as text, images, or audio characteristics to recommend content similar to users' demonstrated interests. de Campos et al. (2023) utilized content-based recommendation methods with advanced feature extraction techniques, such as natural language processing or image recognition, to precisely match user interests and needs, thereby achieving more accurate personalized recommendation services<sup>13</sup>. Moreover, Dudekula et al. (2023) suggested that integrating deep learning technologies, such as convolutional neural networks and recurrent neural networks, effectively learned complex feature representations of users and items, thereby improving the accuracy and personalization of recommendation systems<sup>14</sup>. Deep learning techniques automate the learning and extraction of high-order features from data, which are crucial for the effectiveness of personalized recommendation systems. Marinó et al. (2023) demonstrated that through deep neural networks, it was possible to more accurately map the feature spaces of users and items, significantly enhancing the effectiveness of recommendation systems<sup>15</sup>. Graph neural networks have gained attention in personalized recommendation in recent years. Li et al. (2023) constructed complex relational graph structures between users and items, leveraging graph neural networks for information propagation and feature extraction, providing a more precise and comprehensive solution for personalized recommendations<sup>16</sup>. Reinforcement learning techniques exhibit unique advantages in dynamic recommendation systems due to their ability to optimize decision strategies in dynamic environments. Sivamayil et al. (2023) proposed that using deep reinforcement learning algorithms, recommendation systems could model the recommendation process as a continuous sequence decision problem and adjust recommendation strategies based on real-time user feedback, significantly enhancing user satisfaction<sup>17</sup>. Knowledge graphs have gradually attracted attention in personalized recommendation systems. Zhao et al. (2023) discovered that by constructing knowledge graphs, it was possible to effectively capture multidimensional user interest information and correlate this information with item features, thereby improving the accuracy of recommendation systems<sup>18</sup>. Bhaskaran and Marappan (2023) improved traditional reinforcement learning algorithms, such as Q-learning and deep Q-networks, to enhance their performance in complex environments, such as mold quality prediction, further confirming the potential of reinforcement learning in optimizing recommendation systems<sup>19</sup>. Barata et al. (2023) employed enhanced digital engineering to deploy a hybrid customization and personalization strategy in traditional industries. By implementing key Industry 4.0 technologies such as cloud computing, mobile devices, augmented reality, and 3D printing, they transformed traditional home ceramic enterprises, achieving sustained customer interaction and data-driven design and marketing activities. Personalization is not the endpoint; exploring the synergistic advantages of mass customization and personalization through enhanced strategies offers significant benefits<sup>20</sup>. Huang et al. (2023) integrated deep learning and additive manufacturing technologies and successfully implemented a personalized conversion design for sofa models using the You Only Look Once version 4 (YOLOv4) algorithm<sup>21</sup>. Ruediger- Flore et al. (2023) proposed a deep learning-based industrial parts classification method. They discovered that, with deep learning and reinforcement learning techniques, complex features of parts could be automatically learned and extracted, enabling more accurate identification and classification of different types of parts. The method also continuously optimizes customized designs based on user needs and changing production environments, improving production efficiency and product quality<sup>22</sup>. Li et al. (2023) pointed out that the application of reinforcement learning and deep learning

in the manufacturing phase of the engineering lifecycle covered order selection, process planning, scheduling, process control, and quality control. These technologies, by providing adaptive and flexible solutions, are gradually becoming key factors driving smart manufacturing systems toward more cognitive and personalized directions. This application has promoted the development of industrial personalization, making manufacturing processes more precise and efficient while meeting the personalized needs of different customers<sup>23</sup>. Chen et al. (2024) proposed a personalized recommendation method based on reinforcement learning and multi-objective evolutionary algorithms, which significantly improved the effectiveness and stability of personalized recommendations by dynamically selecting the optimal task splitting scheme<sup>24</sup>.

As intelligent manufacturing technologies continue to evolve, recommendation systems are increasingly being applied beyond consumer-facing domains and into industrial manufacturing. However, these systems must be significantly adapted to suit the unique requirements of industrial contexts. Unlike traditional e-commerce applications—which focus on straightforward “user-product” interactions—recommendation systems in mold manufacturing must process high-dimensional, interrelated industrial data while also supporting real-time decision-making<sup>25</sup>. For instance, Sun et al. (2022) developed a recommendation system for industrial equipment maintenance using graph neural networks. Their approach constructed a relational network between fault patterns and maintenance strategies, enabling the dynamic generation of complex maintenance plans. This demonstrated the feasibility of using graph-based modeling in industrial applications<sup>26</sup>. In a similar vein, Fu et al. (2022) combined reinforcement learning with multi-objective optimization to recommend process parameters in real time for personalized production scheduling. Their system used a reward mechanism to dynamically balance production efficiency and quality constraints<sup>27</sup>. These studies underscore two essential capabilities for applying recommendation systems in manufacturing: the ability to model complex relationships and the ability to optimize strategies dynamically—capabilities that are largely absent in traditional mold production methods.

Despite substantial progress in personalized recommendation systems across domains such as e-commerce, social media, and digital content delivery, these systems face significant limitations when applied to industrial manufacturing. First, industrial settings—such as mold manufacturing—involve complex, heterogeneous data from multiple sources, including material properties, process parameters, and equipment conditions. Traditional collaborative filtering methods, which often assume feature independence, struggle to capture the high-dimensional and nonlinear relationships within this data. Second, manufacturing environments are dynamic. Factors like equipment failures and fluctuating order priorities require real-time decision-making and adaptive system behavior. However, most content-based recommendation methods are static and lack the mechanisms for online learning or feedback-driven optimization. Third, industrial applications demand a high degree of domain knowledge integration. Effective recommendations must incorporate process rules and quality control standards through structured knowledge representations. Systems that rely solely on user behavior data fail to meet the reliability and interpretability requirements of engineering contexts. To overcome these challenges, this study proposes a novel framework that integrates knowledge graphs with reinforcement learning. Domain knowledge—such as process constraints and defect patterns—is encoded into computable graph structures. A Q-learning algorithm is then used to dynamically refine recommendation strategies based on real-time feedback. This integrated approach enables the delivery of accurate, interpretable, and adaptive recommendations, making it particularly well suited for the complex and rapidly evolving needs of modern industrial manufacturing.

## Theoretical research and reinforcement learning in mold data systems

### *Mold data systems*

To support intelligent and data-driven mold manufacturing, this section presents a lifecycle-oriented mold data system. The system consolidates data from multiple stages—including design, production, and maintenance—and provides structured inputs to facilitate knowledge engineering and model development.

A mold data system is a system that utilizes information technology to integrate, manage, and analyze various data throughout the mold design, manufacturing, usage, and maintenance processes. These systems typically integrate functionalities such as Computer-Aided Design/Computer-Aided Manufacturing (CAD/CAM) technologies, Enterprise Resource Planning (ERP) system connections, remote monitoring, and maintenance, aiming to help enterprises enhance mold production efficiency, reduce costs, improve quality, and ensure stable operation and longevity of molds. By digitizing management and providing intelligent decision support in mold production processes, mold data systems offer essential tools for enterprises to maintain competitive advantages in fiercely competitive markets<sup>28</sup>. The characteristics and advantages of mold data systems are depicted in Fig. 1<sup>29</sup>.

### Based on mold data systems: knowledge engineering

Based on the various characteristics of the mold data system described above, this section introduces knowledge engineering techniques to transform domain expert knowledge and historical production data into computable rules and constraints. This process lays the semantic foundation for constructing the knowledge graph.

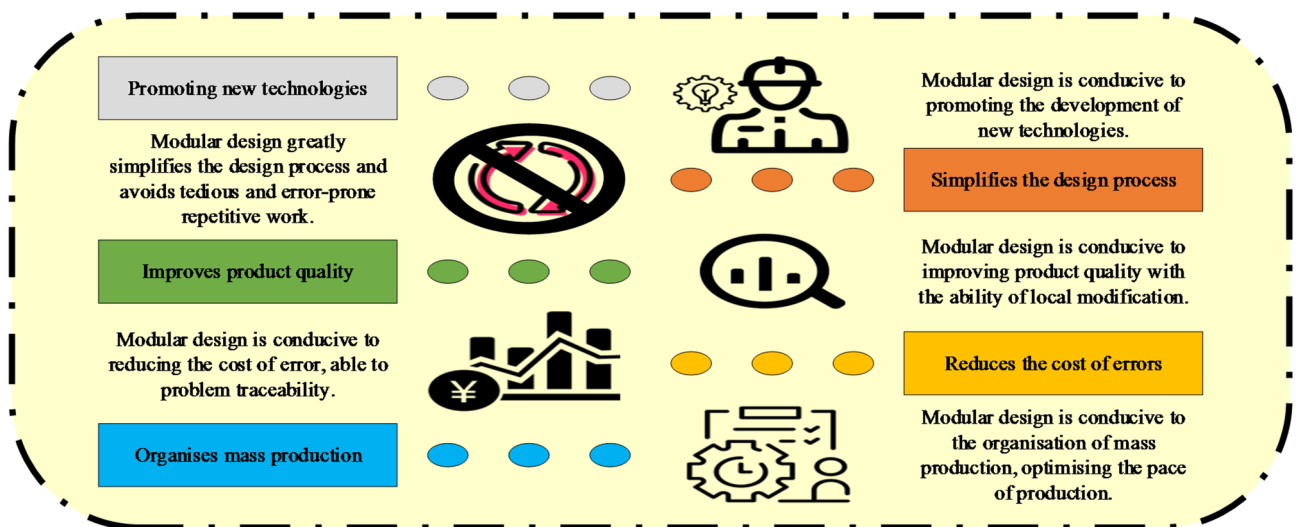
Knowledge engineering is an interdisciplinary field that combines theories and methods from computer science, AI, cognitive science, and knowledge management. Its primary goal is to develop and utilize computer systems to acquire, express, store, use, and manage knowledge. The core objective is to transform human knowledge and expertise into forms that computers can understand and process, facilitating the resolution of complex problems, decision-making, and automation<sup>30,31</sup>. In mold data systems, the key technologies and application characteristics of knowledge engineering are illustrated in Fig. 2<sup>32</sup>.

### The application of knowledge graphs in mold data systems

To better capture the complex relationships within the data, this section uses the output from the knowledge engineering module to build a multidimensional knowledge graph focused on the core elements of process,

Features	Interpretations	
Full life cycle management		It can achieve the whole process management and monitoring of moulds from design, manufacture, use to maintenance, ensuring the continuity and integrity of data.
Digitalisation and intelligence		Using information technology, various data related to moulds are digitally processed and intelligently analyzed to improve decision-making efficiency and precision.
CAD/CAM integration		Integrate CAD design and CAM manufacturing into the system to achieve a seamless connection between design and manufacturing, and accelerate the mould development and production cycle.
Traceability and quality management		Achieve traceability management of the mould production process, record and analyse the data of each link to ensure the quality and service life of the mould.

(a) Features of the mould datatable system



(b) Advantages of the mould datatable system

Fig. 1. Characteristics and advantages of mold data systems.

parameter, and quality. This semantic network explicitly represents the relationships between entities in mold design, and serves as a foundation for intelligent algorithms to perform effective feature extraction and reasoning.

The application of knowledge graphs in mold data systems involves constructing structured knowledge representations based on semantic networks, effectively capturing and managing various entities and their complex relationships in mold design<sup>33</sup>. The basic framework and technological structure of knowledge graphs are illustrated in Fig. 3<sup>34</sup>.

Knowledge graphs represent structured data as nodes and edges to facilitate the representation, storage, and inference of knowledge. Let  $X$  and  $Y$  be entities represented as vectors,  $X \cdot Y$  denotes the dot product of vectors  $X$  and  $Y$ ,  $\|X\|$  and  $\|Y\|$  represents the norms of vectors  $X$  and  $Y$ ,  $|X \cap Y|$  indicates the size of the intersection, and  $|X \cup Y|$  denotes the size of the union. The similarity calculation is shown in Eqs. (1) and (2)<sup>35</sup>:

$$Similarity(X, Y) = \frac{X \cdot Y}{\|X\| \|Y\|} \tag{1}$$

$$JS(X, Y) = \frac{|X \cap Y|}{|X \cup Y|} \tag{2}$$

Equations (1) and (2) illustrate that cosine similarity closer to 1 indicates higher similarity, whereas closer to -1 indicates dissimilarity. Jaccard similarity measures the similarity between two sets and is commonly used to

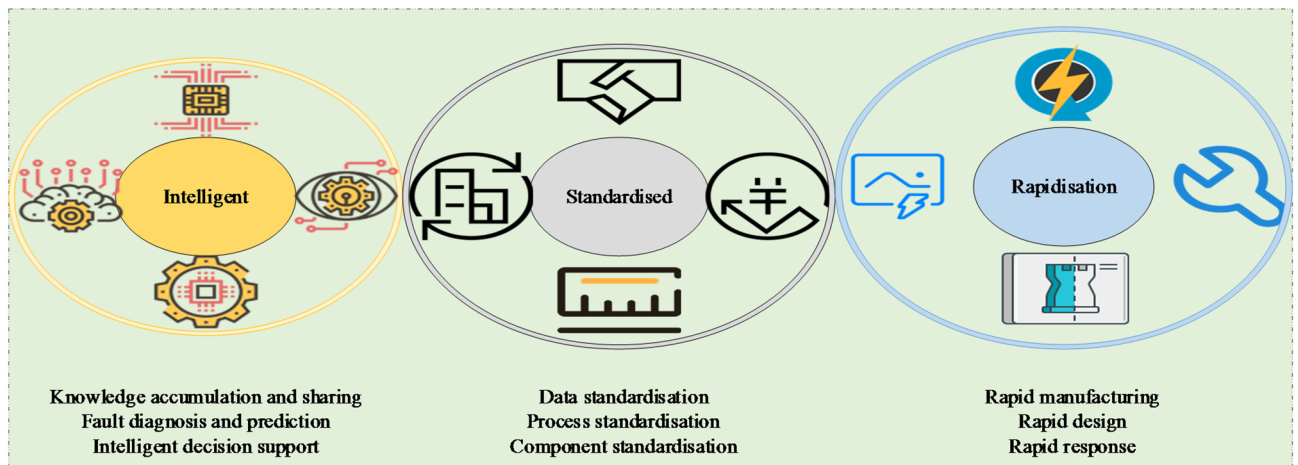
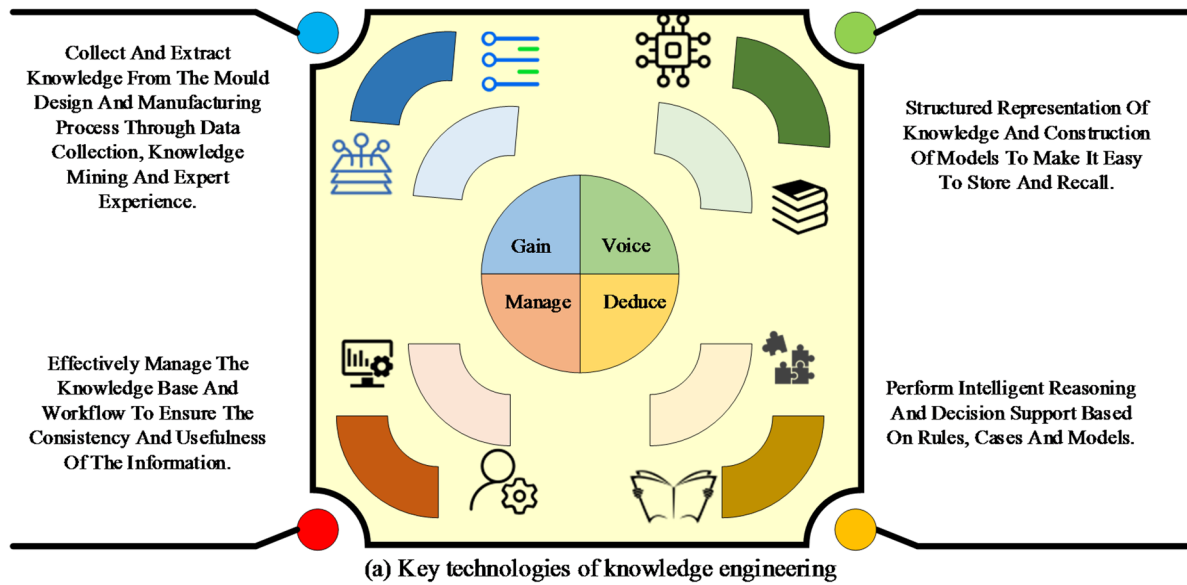


Fig. 2. Key technologies and application characteristics of knowledge engineering.

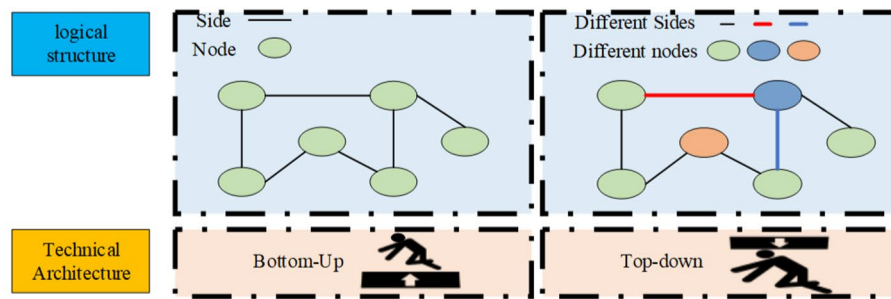
compare sets of neighboring nodes or attribute sets of entities, with values ranging between 0 and 1; values closer to 1 indicate greater similarity.

### Enhancing learning algorithms to optimize knowledge graphs

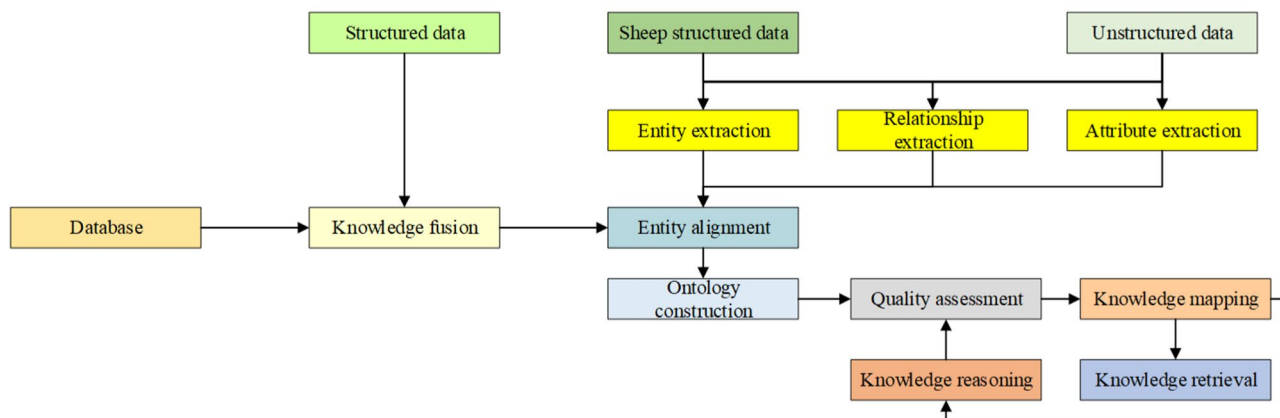
Traditional knowledge graphs are static and thus have difficulty adapting to the rapidly changing conditions of production environments. To address this limitation, this section proposes a dynamic optimization framework that incorporates reinforcement learning. Specifically, the framework employs the Q-learning algorithm to evaluate inference results from the knowledge graph in real time. Based on this evaluation, it adjusts the weights within the graph structure, allowing for the adaptive and continuous updating of manufacturing knowledge.

Reinforcement learning is a machine learning method that enables intelligent systems to interact with their environment and learn through trial and error to maximize cumulative rewards. Unlike traditional supervised and unsupervised learning methods, reinforcement learning does not require labeled data but rather improves decision-making capabilities through iterative feedback<sup>36</sup>.

The process of optimizing knowledge graphs using reinforcement learning begins with the interaction between the intelligent mold system and the knowledge graph. Based on the current state, actions are chosen—typically selecting new triples involving specific head entities from the knowledge graph. The system employs a retrieval function or random selection function to obtain a series of triples related to the selected head entity. Subsequently, a transformation model computes attribute vectors for each retrieved triple. These attribute vectors capture semantic associations between entities and relations, forming the basis for subsequent evaluations. The system assesses the relevance between new triples and known triples by calculating the similarity between



(a) Basic framework of knowledge graph



(b) Technical structure of the knowledge graph

**Fig. 3.** Basic framework and technological structure of knowledge graphs.

attribute vectors. During evaluation, if the similarity between a new triple and known triples is sufficiently high, the system adds it to the candidate set. Concurrently, based on evaluation results, the system may provide positive rewards to reinforce strategies for future action selection. As the system continues to interact with and learn from the knowledge graph, employing Q-learning update policies from reinforcement learning optimizes the accuracy and efficiency of action selection. This iterative process enhances the system's understanding of complex knowledge graph structures, thereby improving and expanding its application effectiveness across various intelligent domains. Ultimately, through continuous evaluation and optimization, the intelligent system effectively enhances the quality and utility of the knowledge graph, providing users with more precise and efficient knowledge acquisition and application experiences.

Q-learning is a classic reinforcement learning algorithm used to learn optimal policies for decision-making in unknown environments. Its core involves updating a state-action value function known as Q-value to learn the optimal strategy. The basic process and algorithmic structure of Q-learning are depicted in Fig. 4<sup>37</sup>.

According to the Bellman equation, the calculation of the Q-value update rule is shown in Eq. (3)<sup>38</sup>:

$$Q(n, m) \leftarrow Q(n, m) + \beta [\delta + \mu \max_{m'} Q(n', m') - Q(n, m)] \quad (3)$$

In Eq. (3),  $Q(n, m)$  represents the Q-value when action  $m$  is taken in state  $n$ ;  $\beta$  is the learning rate, controlling the magnitude of each update.  $\delta$  is the immediate reward obtained after taking action  $m$ .  $\mu$  is the discount factor, measuring the importance of future rewards.  $n'$  is the next state transitioned to after taking action  $m$ .  $\max_{m'} Q(n', m')$  denotes the maximum Q-value among all possible actions in state  $n'$ .

### Data-driven system for mold design based on reinforcement learning knowledge graph recommendation algorithm

This section builds on the dynamic knowledge graph's enhanced learning capabilities by introducing the LightGCN-Q recommendation algorithm. LightGCN-Q integrates the high-order feature propagation of graph convolutional network (GCN) with the real-time decision-making power of reinforcement learning. Together, these components form a closed-loop optimization system designed specifically for personalized mold production.

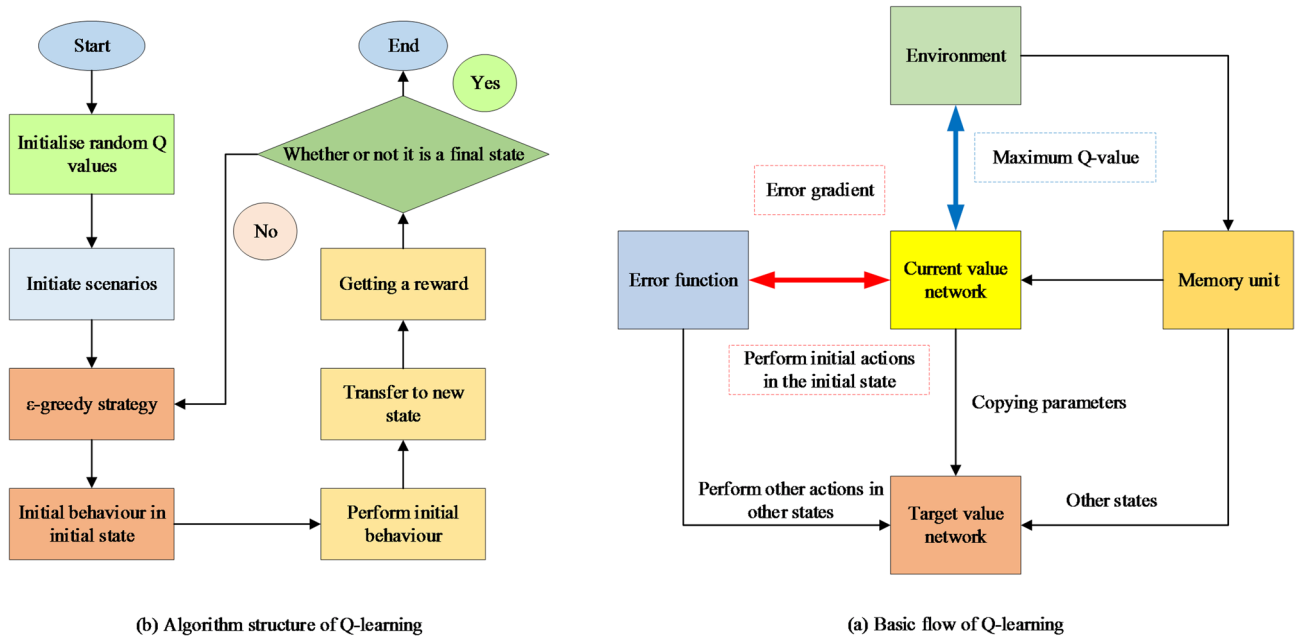


Fig. 4. Basic process and algorithm structure of Q-learning.

In mold design, geometric features represent key position constraints and design parameters. Through knowledge engineering, feature-driven parameters can be altered to achieve parametric modeling of parts, including feature selection, parametric modeling, and design knowledge storage centered around features. Based on the reinforcement learning knowledge graph recommendation algorithm, this study initially utilizes the open application programming interface of the Nutanix nx platform to implement part modeling, ensuring the technical feasibility of feature-driven knowledge modeling methods. Subsequently, design rules are translated into programming languages to accurately express them. Starting from key parameters of dedicated and associated modules, with parameter changes and transfers, the modeling process is driven to generate dedicated modules that meet requirements.

To further enhance the performance of the knowledge graph recommendation algorithm, this study integrates LightGCN into the core of the reinforcement learning system, optimizing the representation of user and module features and recommendation decisions. LightGCN is a lightweight GCN model specifically designed for recommendation systems to efficiently capture collaborative signals between users and items. Compared to traditional graph neural networks, LightGCN simplifies the model structure by removing unnecessary operations and retaining only the core neighbor information aggregation operation, significantly improving computational efficiency and representational ability. Its multi-layer nested graph convolution design can capture higher-order collaborative relationships between users and items in the interaction graph, enabling the recommendation system to achieve a balance between accuracy and efficiency.

In the mold digitization system, the interaction behavior between users and modules forms a sparse user-module interaction matrix. Based on this matrix, this study optimizes the construction of the user-module interaction graph using LightGCN, where the graph's nodes include two categories: users and modules. The edges of the graph represent user operations on the modules, and the edge weights can be dynamically adjusted based on interaction frequency or importance. LightGCN aggregates neighborhood information of users and modules in the interaction graph through multiple layers of graph convolution, capturing higher-order collaborative signals. The specific calculation is as shown in Eqs. (4) and (5):

$$e_u^{(l+1)} = \sum_{i \in \mathcal{N}_u} \frac{1}{|\mathcal{N}_u|} e_i^{(l)} \tag{4}$$

$$e_i^{(l+1)} = \sum_{u \in \mathcal{N}_i} \frac{1}{|\mathcal{N}_i|} e_u^{(l)} \tag{5}$$

$e_u^{(l+1)}$  represents the embedding of user  $u$  at layer  $l$ .  $\mathcal{N}_u$  represents the set of neighboring modules of user  $u$ .  $e_i^{(l)}$  represents the embedding of module  $i$ .

By stacking multiple layers of convolution, the representations of users and modules not only include information from direct neighbors but also capture second-order and higher-order collaborative signals.

In the reinforcement learning framework, the output of LightGCN is used as part of the state representation to improve the accuracy and generalization ability of policy decisions. Let the high-order embeddings of users

and modules output by LightGCN,  $e_u$  and  $e_m$ , be integrated into the state vector, and the state  $S_t$  is calculated as shown in Eq. (6):

$$S_t = [e_u, e_m, E] \tag{6}$$

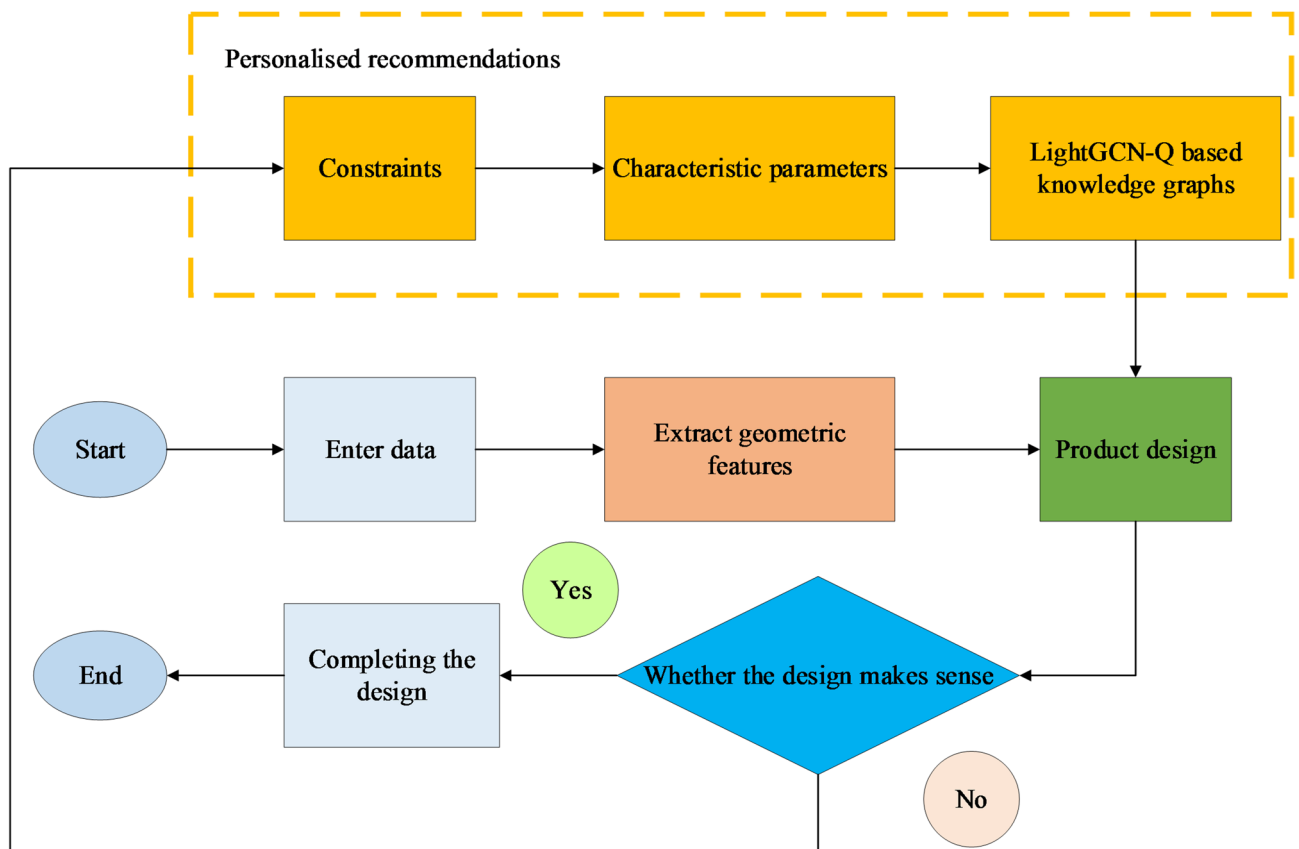
$e_u$  and  $e_m$  are the final embedding representations of the user and module, which more precisely describe the latent relationship between user demands and module features.  $E$  represents other modeling parameters.

In the reinforcement learning algorithm, the core task of the policy network is to generate an action based on the current state  $S_t$ . This action may include recommending a module or adjusting specific parameters to optimize the performance of the mold digitization system. Traditional policy networks might rely solely on simple historical behavior or static rules. However, after integrating LightGCN, by aggregating collaborative signals from user and item nodes, the policy network can obtain richer and more accurate contextual information. This enhanced signal allows the policy network to more accurately predict user preferences and select better actions. When the action chosen by the policy network is accepted by the user, the system provides positive rewards, encouraging the network to learn similar behaviors. Conversely, if the recommended action is not accepted or does not achieve the expected result, the system gives negative rewards to guide the policy network in optimizing the decision-making process. Through this reward-based feedback mechanism and combined with update algorithms such as Q-learning, the policy network can continuously adjust its decision-making strategy, gradually improving the accuracy and efficiency of action selection. Furthermore, the collaborative signals from LightGCN play an additional role in the reward mechanism, helping the system better understand complex user behavior patterns. This enhances its ability to adapt to user demands, thereby achieving comprehensive optimization of the mold digitization system. The process of the mold digitization system under the LightGCN-optimized reinforcement learning knowledge graph recommendation algorithm is shown in Fig. 5.

The calculation of model prediction *Accuracy*, F1 score, *Precision*, and *Recall* under the knowledge graph recommendation algorithm based on LightGCN-Q is as shown in Eqs. (7)–(10):

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{7}$$

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{8}$$



**Fig. 5.** Process of mold digitization system under the knowledge graph recommendation algorithm based on LightGCN-Q.

$$\textit{Precision} = \frac{TP}{TP + FP} \quad (9)$$

$$\textit{Recall} = \frac{TP}{TP + FN} \quad (10)$$

*TP* is True Positive; *TN* is True Negative; *FP* is False Positive; *FN* is False Negative.

The calculation of model Area Under the Receiver Operating Characteristic Curve (AUC-ROC), mean absolute error (MAE), and root mean square error (RMSE) under the knowledge graph recommendation algorithm based on LightGCN-Q is as shown in Eqs. (11)–(13):

$$\textit{AUC} = \int_0^1 \textit{TPR}(\textit{FPR}) d\textit{FPR} \quad (11)$$

$$\textit{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i| \quad (12)$$

$$\textit{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (13)$$

*TPR* is True Positive Rate; *FPR* is False Positive Rate;  $y_i$  is the actual value;  $\hat{y}_i$  is the predicted value;  $N$  is the total number of data points.

### Personalized product quality assessment algorithm

To validate the proposed model's effectiveness in industrial personalized customization, this study uses multiple linear regression, stepwise regression, and Partial Least Squares (PLS) regression. These methods analyze the relationship between personalized part data and quality parameters. Multiple linear regression models the linear relationship between several independent variables—such as production parameters—and a dependent variable, like quality indicators. Regression coefficients are estimated using the least squares method, which minimizes the sum of squared errors to find the best-fit regression equation. PLS regression serves as a supplementary tool. Its main purpose is to verify the latent correlations between the LightGCN-Q model's predictions and actual production parameters, thereby improving the recommendation system's interpretability. Specifically, PLS extracts latent variables that represent the relationship between process parameters and the predicted qualification rates from LightGCN-Q. This method effectively quantifies complex, high-dimensional nonlinear interactions—for example, the combined influence of injection molding temperature and cooling time on qualification rates.

Let  $Y$  denote the quality parameter to be predicted,  $X$  represent the production parameters of personalized parts,  $\beta_0, \beta_1, \beta_2, \dots, \beta_i$  denote regression coefficients indicating the impact of each independent variable on the dependent variable, and  $\epsilon$  represents the error term. The calculation of the multiple linear regression model is shown in Eq. (14)<sup>39</sup>:

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \dots + \beta_i X_i + \epsilon \quad (14)$$

Stepwise regression is a variable selection method aimed at finding the optimal regression model by iteratively introducing or removing variables based on their statistical significance. The basic idea is to initially establish an initial model and then incrementally add variables that have the greatest impact on the dependent variable according to statistical significance tests, while simultaneously removing variables with smaller effects, until no more variables can be introduced or removed.

Let  $M_K$  denote the model at step  $K$ ,  $X_i$  denote the independent variable to be added, and  $F(X_i|M_K)$  denote the  $F$ -statistic or other criteria for variable  $X_i$  given the model  $M_K$ . The calculation for introducing variables is shown in Eq. (15)<sup>40</sup>:

$$M_{K+1} = M_K + \textit{Arg}[\textit{max} X_i F(X_i|M_K)] \quad (15)$$

The PLS Regression model combines the advantages of Principal Component Analysis and regression analysis, particularly suitable when there is multicollinearity among the independent variables. By reducing the dimensions of the independent variables, the original variables are transformed into a set of orthogonal principal components, thereby eliminating the multicollinearity issue. Subsequently, regression models are built using these principal components to predict the dependent variable.

### Experimental data design

This study establishes a mold intelligent digitization system based on the knowledge graph recommendation algorithm optimized by LightGCN and reinforcement learning. It uses two datasets: the 50 Types of Car Parts Dataset and the Car Parts Classification Dataset. The datasets selected in this study cover a wide range of car part types, including engines, brake pads, drive shafts, etc., and effectively reflect the characteristic requirements and complexity of different components in mold manufacturing. Both datasets include high-resolution images

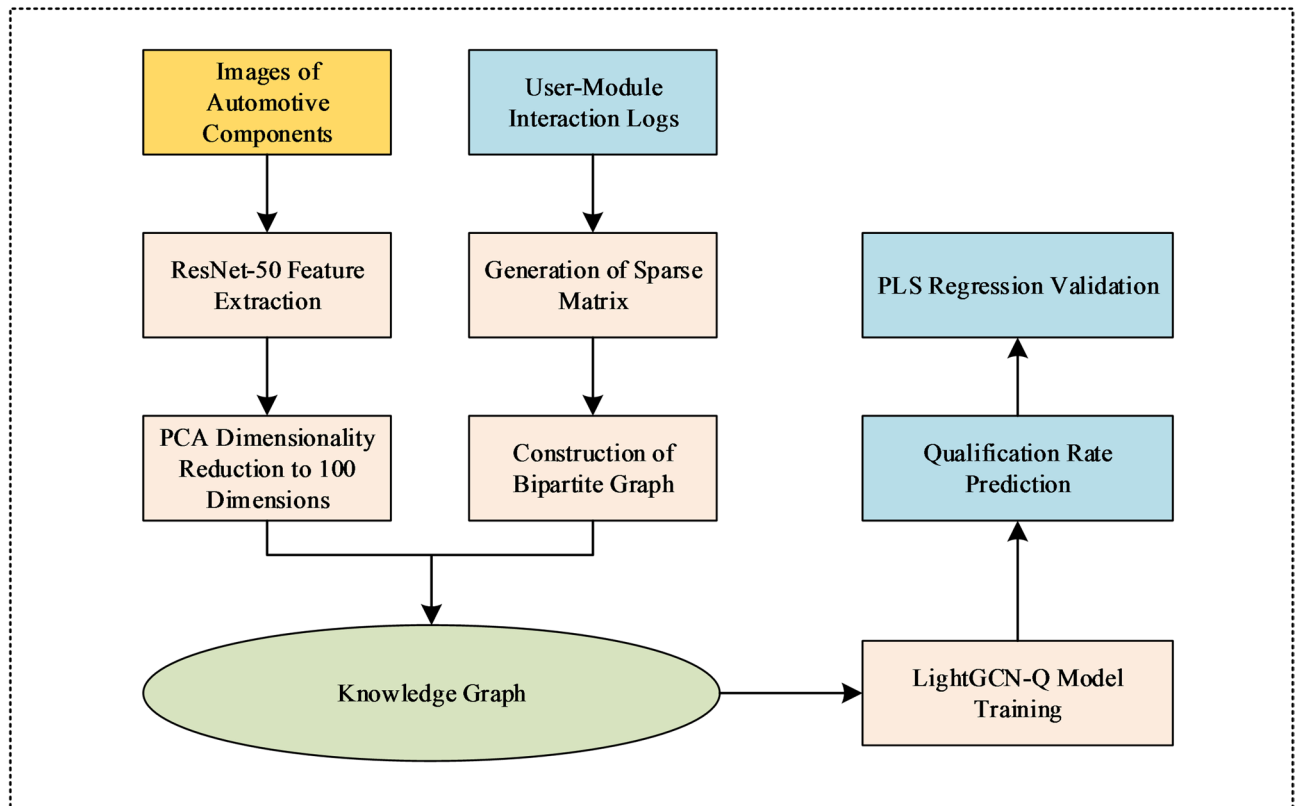
( $224 \times 224 \times 3$ ), providing clear feature information that facilitates model training and feature extraction. Additionally, the Car Parts Classification Dataset contains 14 categories of car parts, with a total of 16,600 high-quality real images, approximately 50 images per category, ensuring the authenticity of the data and its relevance to actual mold manufacturing needs. In the experiment, the datasets are split into a training set and a test set in an 80:20 ratio. This split is based on a common rule of thumb in machine learning practice, designed to ensure that the model receives sufficient data for training while retaining a portion of unseen data for evaluation to test model performance. To enhance the rigor of the experiment, five-fold cross-validation is also employed to assess the model's generalization ability. In cross-validation, the training set is further divided into five folds, with one fold used as the validation set and the remaining four folds used as the training set. Training and validation are alternated, and the final average value is taken as the performance evaluation metric of the model. This experimental design ensures the rationality of data distribution and strengthens the model's applicability and generalization ability in a broader range of mold manufacturing fields.

Additionally, the specific data preprocessing steps employed in this study are illustrated in Fig. 6.

As illustrated in Fig. 6, the data preprocessing procedure comprises the following steps: For the automotive component image dataset, each  $224 \times 224 \times 3$  image is passed through the ResNet-50 network to extract 1,024-dimensional deep features. These features are then reduced to 100 dimensions using Principal Component Analysis (PCA) to serve as the initial embedding representations of modules. The extracted image features are linked with process parameters using annotated files to establish a “module–parameter” entity relationship. For the user–module interaction dataset, an interaction matrix is constructed based on historical operation logs from engineers. Non-zero elements in the matrix indicate design preference intensities on a scale of 1 to 5. This matrix is then transformed into a user–module bipartite graph, where edge weights are determined by interaction frequency. This bipartite graph serves as the topological input structure for the LightGCN model. In summary, the preprocessing phase encodes both datasets into a unified graph structure to support the embedding propagation of LightGCN and the decision-making process of the reinforcement learning framework.

This study applies a standardized approach to parameter tuning across different models, while introducing customized adjustments based on the characteristics of each algorithm. The specific hyperparameter configurations are summarized in Table 1.

The settings for the number of graph convolution layers were determined through preliminary experiments. By evaluating performance across different layer configurations (1 to 5 layers), the model performed best with three layers. At this setting, both the hit rate and normalized discounted cumulative gain (NDCG) showed notable improvements, indicating that three layers effectively captured high-order interactions between users and modules. When fewer than three layers were used, limited information aggregation led to reduced recommendation accuracy. Conversely, using more than three layers resulted in over-smoothing, which diminished the discriminative power of the node features. Based on these findings, the final number of graph



**Fig. 6.** Data preprocessing workflow.

Parameter category	LightGCN-Q parameters	Reinforcement learning parameters
Embedding dimension	100	
Batch size	5000	
Distance metric	L1 norm (distance = 1)	
Learning rate	0.003	
Regularization	0.02	
Optimizer	Adam	
Early stopping	Stop after 30 non-improving epochs	
Neighborhood sample size	10	-
Number of GCN layers	3	-
Node weight decay	0.5	-
Reward discount factor	-	$\gamma = 0.9$
Exploration rate	-	Decreasing from 0.2 to 0.01

**Table 1.** Hyperparameter settings for the models.

convolution layers was set to three. L2 regularization is also applied to enhance the model's generalization capability. In the collaborative filtering recommendation algorithm, the number of latent factors is set to 50, and cosine similarity is utilized for similarity computation. For the content-based recommendation algorithm, the feature dimension is set to 150, with Jaccard similarity used for similarity calculations. In the Bayesian recommendation algorithm, the Beta distribution ( $\alpha = 1$ ,  $\beta = 1$ ) is adopted, with Gibbs sampling employed as the sampling method. Through the combination of standardized and differentiated parameter settings, this study ensures fairness in the comparative experiments while fully leveraging the performance characteristics of each model. This approach provides a valuable reference for selecting recommendation algorithms within mold digitization systems.

### Evaluation and analysis of intelligent data-driven mold system based on reinforcement learning knowledge graph recommendation algorithm Analysis of personalized custom part quality assessment results under the algorithm introduced in this study

This study selected six types of test samples (Samples 1–6) from an automotive component dataset, focusing on parts commonly associated with high complexity in mold manufacturing. Sample selection is guided by a composite score (1–10) reflecting structural complexity, variety of process parameters, and variability in historical qualification rates—where higher scores indicate greater production difficulty. The selected components include engine cylinder blocks, brake pad molds, drive shafts, turbine blades, interior door panels, and transmission gears. These samples span a spectrum of manufacturing scenarios, ranging from basic injection-molded parts to intricate precision components. Each category contains 100 to 150 data entries, ensuring that the evaluation results are representative of industry practice.

Figure 7 shows the quality evaluation results for personalized custom parts in the mold intelligent digitization system based on the LightGCN-Q knowledge graph recommendation algorithm.

As shown in Fig. 7, in Sample 1, the actual pass rate is 96.8%, while the predicted value based on the LightGCN-Q knowledge graph algorithm is 93.6%, with a 3.2% difference from the actual value. This discrepancy may arise from biases in the dataset or the model's insufficient feature extraction ability for a small number of samples. In contrast, the multiple linear regression model's fitted pass rate is 80.2%, significantly lower than the actual pass rate, indicating that the model fails to capture complex relationships accurately. On the other hand, the predicted value of the PLS regression model is close to the actual pass rate at 93.6%, indicating that PLS regression fits the data well within a smaller error range. In Sample 2, the actual pass rate is 98.5%, and the pass rates for the stepwise regression and PLS regression models are 95.1% and 95.6%, respectively, with predicted results that are relatively close to the actual value. Although stepwise regression performs slightly lower than PLS regression, the difference between the two models is minimal, suggesting that both models can adapt well to the complex dataset. However, the multiple linear regression model's predicted value is only 80.3%, significantly lower than the actual pass rate, reflecting its poor adaptability to high-dimensional data. In Sample 3, the actual pass rate is 95.9%, but the multiple linear regression model's predicted value is overly high at 101.3%, leading to a larger error. The PLS regression's predicted value is 94.5%, closest to the actual value, demonstrating its advantage in handling complex nonlinear relationships. Additionally, the stepwise regression model's pass rate is 93.7%, also within a close range to the actual pass rate. In Sample 4, the actual pass rate is 95.9%, and the multiple linear regression's predicted value is low at only 88.4%. The stepwise regression and PLS regression models predict pass rates of 93.7% and 93.2%, respectively, which are closer to the actual value. This indicates that both models are effective in controlling errors when handling small sample sizes. In Sample 5, the predicted value from PLS regression is 95.6%, slightly higher than the actual pass rate, but still quite close, showcasing the model's stability on more complex datasets. In contrast, the multiple linear regression model predicts 80.2%, showing a large error. In Sample 6, the actual pass rate is 98.8%, but the predicted values from multiple linear regression and stepwise regression are significantly lower, at 80.2% and 88.4%, respectively. The PLS regression model's predicted value is 93.2%, which, while still having a gap from the actual value, performs the best among

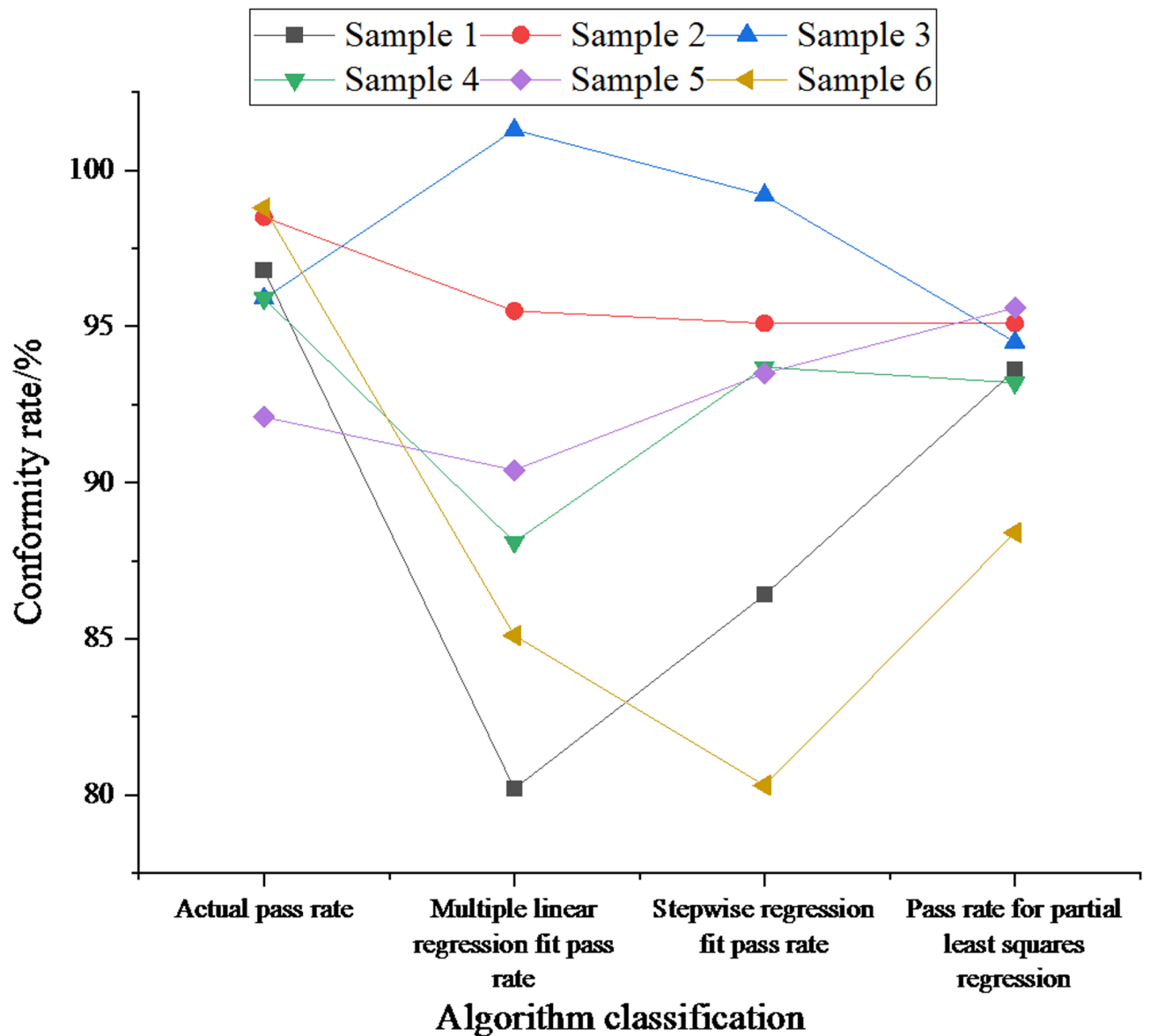


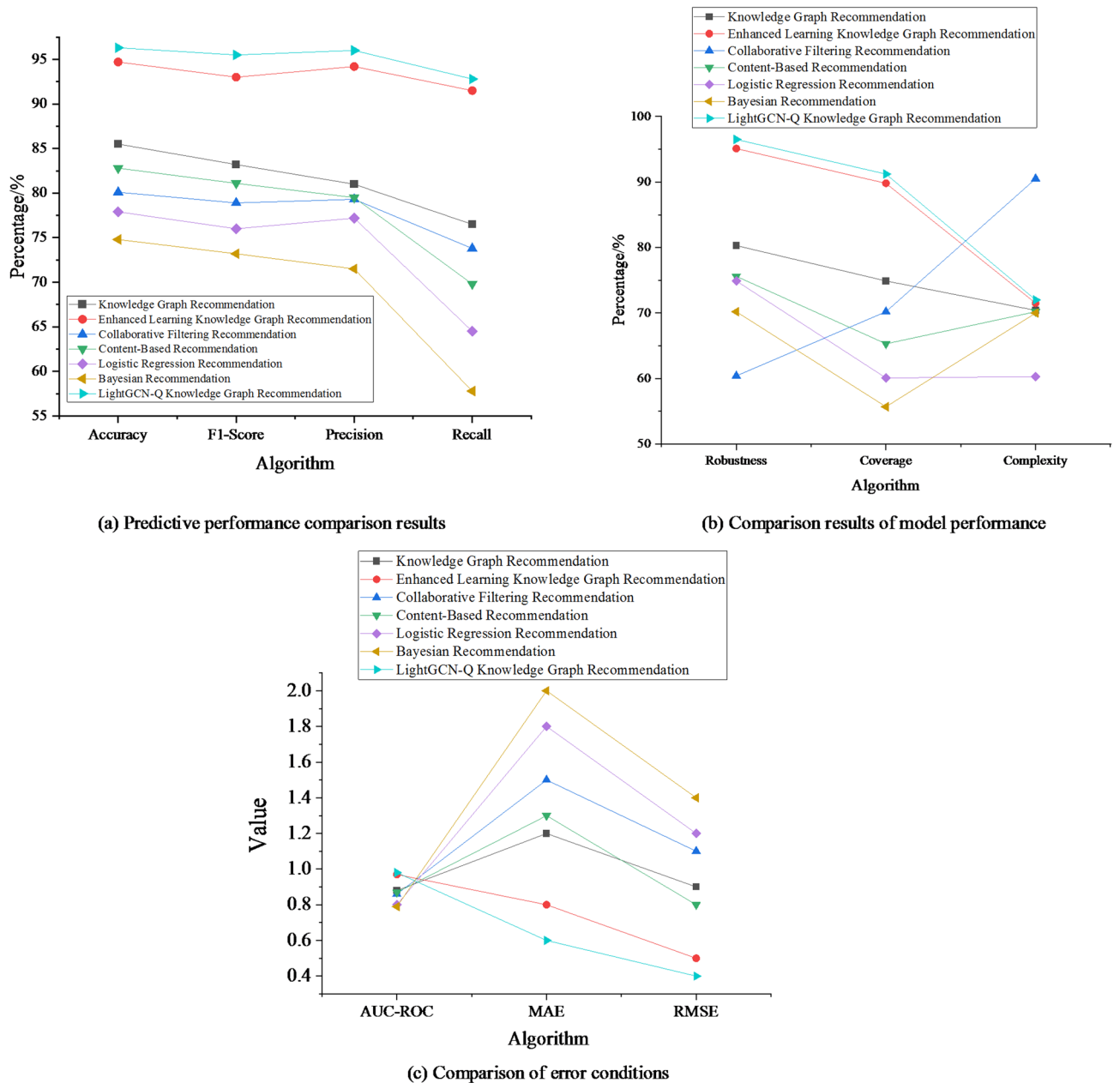
Fig. 7. Quality assessment results of personalized custom parts under the proposed algorithm.

all models. Overall, the LightGCN-Q knowledge graph algorithm performs well in predicting the actual pass rate for most samples, demonstrating high accuracy and stability.

#### Performance comparison of this study's algorithm with other algorithms

Figure 8 shows the comparative results of the LightGCN-Q knowledge graph recommendation algorithm and other algorithms across three categories of performance metrics: prediction accuracy, model performance, and error analysis.

As shown in Fig. 8(a), the LightGCN-Q knowledge graph recommendation algorithm used in this study performs the best, excelling in several performance metrics such as prediction accuracy, F1-Score, precision, and recall. In terms of accuracy, LightGCN-Q achieves 96.3%, higher than the collaborative filtering recommendation algorithm at 80.1% and the Bayesian recommendation algorithm at 74.8%. In terms of F1-Score, LightGCN-Q reaches 95.5%, significantly outperforming the logistic regression recommendation algorithm at 76% and the content-based recommendation algorithm at 81.1%. Regarding precision, LightGCN-Q achieves 96%, far surpassing the collaborative filtering recommendation algorithm at 79.3% and the logistic regression recommendation algorithm at 77.2%. In terms of recall, LightGCN-Q reaches 92.8%, considerably higher than the collaborative filtering recommendation algorithm at 73.8% and the content-based recommendation algorithm at 69.8%. These results indicate that the LightGCN-Q algorithm provides more accurate and stable performance in the mold intelligent digitization system, with higher performance advantages and significant improvement over traditional algorithms.



**Fig. 8.** Comparison of the LightGCN-Q Knowledge Graph Recommendation Algorithm with Other Algorithms.

As shown in Fig. 8(b), the robustness of the LightGCN-Q knowledge graph recommendation algorithm is 96.5%, much higher than the collaborative filtering recommendation algorithm at 60.4% and the logistic regression recommendation algorithm at 74.9%. Compared to the knowledge graph recommendation with enhanced learning, it shows an improvement of 1.4%. In terms of recommendation coverage, LightGCN-Q reaches 91.2%, clearly outperforming the content-based recommendation algorithm at 65.3% and the Bayesian recommendation algorithm at 55.7%. In terms of computational complexity, LightGCN-Q also shows a good balance, achieving 72%, slightly higher than other algorithms but still within an acceptable range. These results suggest that the LightGCN-Q algorithm exhibits high performance advantages in robustness, recommendation coverage, and computational complexity, with significant improvements over other algorithms.

As shown in Fig. 8(c), in terms of error performance across models, the LightGCN-Q knowledge graph recommendation algorithm demonstrates significant advantages. Its AUC-ROC value is 0.98, indicating excellent classification performance, with a notable improvement compared to other algorithms such as the collaborative filtering recommendation algorithm at 0.86 and the logistic regression recommendation algorithm at 0.8. Additionally, the MAE value is 0.6 and the RMSE value is 0.4, both significantly lower than the collaborative filtering recommendation algorithm's 1.5 and 1.1, the content-based recommendation algorithm's 1.3 and 0.8, and the Bayesian recommendation algorithm's 2 and 1.4. Compared to the knowledge

graph recommendation algorithm with enhanced learning before optimization, the LightGCN-Q knowledge graph recommendation algorithm outperforms in all error metrics. The AUC-ROC value of 0.98 is significantly higher than the knowledge graph recommendation algorithm with enhanced learning at 0.97, indicating more stable and accurate classification performance. Moreover, the MAE and RMSE values are 0.6 and 0.4, much lower than the enhanced learning-based knowledge graph recommendation algorithm's 0.8 and 0.5. These findings demonstrate that the LightGCN-Q algorithm more effectively reduces prediction errors and enhances the accuracy and reliability of recommendation results. In practical applications, compared to the knowledge graph recommendation algorithm with enhanced learning, LightGCN-Q shows stronger robustness and higher performance advantages.

The LightGCN-Q knowledge graph recommendation algorithm effectively supports intelligent mold manufacturing by integrating lightweight GCNs with reinforcement learning. This combination enables both efficient knowledge extraction and real-time system updates. The algorithm demonstrates notable advantages in prediction accuracy, robustness, recommendation coverage, computational efficiency, and fault tolerance. These strengths are largely due to its ability to automatically extract complex features and model high-dimensional, nonlinear relationships within manufacturing data. As a result, LightGCN-Q is highly effective in addressing the personalized customization needs of mold production, offering clear performance improvements over traditional recommendation approaches.

## Discussion

This study introduces the LightGCN-Q-based knowledge graph recommendation algorithm into the mold digitization system. By leveraging reinforcement learning and knowledge graph technologies, it significantly enhances the quality assessment of personalized custom parts. Compared to traditional collaborative filtering or content-based recommendation methods, the LightGCN-Q algorithm demonstrates significant advantages in handling complex mold data and real-time recommendation requirements. Traditional recommendation methods, such as collaborative filtering or content-based approaches, typically rely on user historical behavior and item features for recommendations. However, these methods face limitations when dealing with large, high-dimensional, or sparse data, especially when complex component information is involved in the mold digitization system, making it difficult to accurately extract complex features. In contrast, the LightGCN-Q-based knowledge graph recommendation algorithm can automatically extract features and relationships associated with components through deep learning and graph structure mining, thereby significantly improving prediction accuracy, robustness, and recommendation coverage. The results show that the algorithm outperforms traditional methods in both prediction qualification rates and recommendation coverage across multiple test samples. When handling complex mold data, this algorithm captures complex relationships between data more effectively, enhancing the stability and accuracy of predictions. Although the computational complexity is slightly higher, it provides better performance and results in practical applications. Furthermore, the model in this study demonstrates high robustness, maintaining high accuracy even when data quality or quantity fluctuates. When sample sizes increase or data quality varies, the LightGCN-Q algorithm continuously improves prediction capabilities through optimized feature extraction and reinforcement learning mechanisms. This robustness enables the model to handle real-time demands and personalized customization needs more effectively in a more complex and dynamic mold design environment, thereby enhancing its practical application value. In summary, the predictive outputs generated by the LightGCN-Q and PLS methods play a crucial role in enabling personalized mold manufacturing. The LightGCN-Q algorithm accurately forecasts part qualification rates, allowing for timely adjustments to process parameters during production. When a low qualification rate is predicted, the system can proactively implement corrective measures to prevent potential quality issues. The PLS method complements this by uncovering latent relationships between process parameters and qualification outcomes. This provides engineers with an interpretable analytical tool to better understand and optimize the manufacturing process. Together, these methods improve production efficiency, reduce scrap rates, and deliver tangible economic benefits. By enabling real-time feedback and adaptive decision-making, both LightGCN-Q and PLS have demonstrated strong practical value in industrial settings.

## Conclusion

This study analyzes the knowledge engineering module within the mold digitization system and designs dedicated models for knowledge representation. Additionally, it applies reinforcement learning algorithms to extract key information and features from the knowledge graph of the mold digitization system, and optimizes the reinforcement learning-based knowledge graph algorithm using the LightGCN algorithm. Finally, the study uses multiple linear regression models, stepwise regression models, and PLS regression models to evaluate the quality of personalized custom parts, comparing the performance of these models across three categories: prediction performance, model performance, and model error. The results show that the algorithm based on the LightGCN-Q knowledge graph performs excellently in terms of prediction ability, closely aligning with the actual qualification rate. For test sample 1, the actual qualification rate is 96.8%, while the PLS regression model predicts a qualification rate of 93.6%. For test sample 2, with an actual qualification rate of 98.5%, the predictions from the stepwise regression model and PLS regression model are close to the actual value. For test sample 5, the actual qualification rate is 92.1%, and the PLS regression model predicts a qualification rate of 95.6%. The algorithm based on reinforcement learning in the knowledge graph performs well in real-world applications, closely matching the actual qualification rate and demonstrating its strong predictive ability in personalized custom part quality evaluation. The LightGCN-Q knowledge graph recommendation algorithm used in this study achieves prediction accuracy, F1 score, precision, and recall rates of 96.3%, 95.5%, 96%, and 92.8%, respectively. In terms of model performance, the LightGCN-Q knowledge graph recommendation

algorithm demonstrates robustness and recommendation coverage rates of 96.5% and 91.2%, both significantly outperforming other algorithms. Although its computational complexity is 72%, slightly higher than some algorithms, the algorithm's outstanding performance across major metrics demonstrates its clear advantage in this study. In terms of error performance, LightGCN-Q achieves an AUC-ROC value of 0.98, an MAE of 0.6, and an RMSE of 0.4, outperforming other algorithms in terms of both excellence and stability. The results indicate that the proposed LightGCN-Q algorithm can adapt to different manufacturing demands in complex mold systems, extracting key features and information, thereby improving the accuracy and stability of quality evaluation. Furthermore, despite the slightly higher computational complexity, the algorithm's high robustness and high recommendation coverage enable it to demonstrate strong adaptability in various manufacturing scenarios. Therefore, the methods and algorithms proposed in this study have broad potential for application in other types of molds or manufacturing processes, contributing to the improvement of personalized custom part quality evaluation levels.

The limitations of this study include the lack of full automation in mold design and optimization, requiring engineers to manually make detailed adjustments. In certain samples, the prediction model's performance still shows errors, which may be related to data preprocessing, feature selection, and the model's generalization ability. Future research can optimize the system by developing automated optimization tools and introducing personalized recommendation strategies based on user group relationships. This will not only improve the accuracy of quality detection and recommendation systems but also enhance the automation level of mold design and optimization. Additionally, further improving the model's robustness and reducing the impact of data noise on the results will help enhance the system's stability and accuracy.

### Data availability

All data generated or analysed during this study are included in this published article [and its supplementary information files].

Received: 15 August 2024; Accepted: 20 June 2025

Published online: 04 July 2025

### References

- Chandrashekeran, S. & Keele, S. Making markets from the data of everyday life. *Environ. Plann. A: Econ. Space*. **56** (1), 288–310 (2024).
- Turner, C. & Oyekan, J. Personalised production in the age of circular additive manufacturing. *Appl. Sci.* **13** (8), 4912 (2023).
- Behera, R. K., Bala, P. K., Tata, S. V. & Rana, N. P. Retail atmospheric effect on store performance and personalised shopper behaviour: a cognitive computing approach. *Int. J. Emerg. Markets*. **18** (8), 1948–1977 (2023).
- Zhong, C., Li, G. & Meng, Z. A hybrid teaching-learning slime mould algorithm for global optimization and reliability-based design optimization problems. *Neural Comput. Appl.* **34** (19), 16617–16642 (2022).
- He, X. et al. Simplifying and powering graph convolution network for recommendation. In Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, 639–648. (2020).
- Chien, C. F. & Peng, J. Y. Bayesian inference for multi-label classification for root cause analysis and probe card maintenance decision support and an empirical study. *J. Intel. Manufact.* **36**(3), 1943–1958 (2025).
- Xia, L., Liang, Y., Leng, J. & Zheng, P. Maintenance planning recommendation of complex industrial equipment based on knowledge graph and graph neural network. *Reliab. Eng. Syst. Saf.* **232**, 109068 (2023).
- Peng, S., Siet, S., Ilkhomjon, S., Kim, D. Y. & Park, D. S. Integration of deep reinforcement learning with collaborative filtering for movie recommendation systems. *Appl. Sci.* **14** (3), 1155 (2024).
- Villegas-Ch, W. & García-Ortiz, J. Enhancing learning personalization in educational environments through ontology-based knowledge representation. *Computers* **12** (10), 199 (2023).
- Ma, N., Zhao, F., Zhou, P. Q., He, J. J. & Du, L. Knowledge map-based online micro-learning: impacts on learning engagement, knowledge structure, and learning performance of in-service teachers. *Interact. Learn. Environ.* **31** (5), 2751–2766 (2023).
- Shi, H. & Shang, L. Research on big data personalised recommendation model based on deep reinforcement learning. *Int. J. Networking Virtual Organ.* **28** (2–4), 364–380 (2023).
- Nguyen, L. V., Vo, Q. T. & Nguyen, T. H. Adaptive KNN-Based extended collaborative filtering recommendation services. *Big Data Cogn. Comput.* **7** (2), 106 (2023).
- de Campos, L. M., Fernández-Luna, J. M. & Huete, J. F. Use of topical and Temporal profiles and their hybridisation for content-based recommendation. *User Model. User-Adapt. Interact.* **33** (4), 911–937 (2023).
- Dudekula, K. V. et al. Convolutional neural network-based personalized program recommendation system for smart television users. *Sustainability* **15** (3), 2206 (2023).
- Marinó, G. C., Petrini, A., Malchiodi, D. & Frasca, M. Deep neural networks compression: A comparative survey and choice recommendations. *Neurocomputing* **520** (1), 152–170 (2023).
- Li, X., Sun, L., Ling, M. & Peng, Y. A survey of graph neural network based recommendation in social networks. *Neurocomputing* **549** (1), 126441 (2023).
- Sivamayil, K. et al. A systematic study on reinforcement learning based applications. *Energies* **16** (3), 1512 (2023).
- Zhao, N., Long, Z., Wang, J., Zhao, Z. D. & AGRE A knowledge graph recommendation algorithm based on multiple paths embeddings RNN encoder. *Knowl. Based Syst.* **259** (1), 110078 (2023).
- Bhaskaran, S. & Marappan, R. Enhanced personalized recommendation system for machine learning public datasets: generalized modeling, simulation, significant results and analysis. *Int. J. Inform. Technol.* **15** (3), 1583–1595 (2023).
- Barata, J., Cardoso, J. C. S. & Cunha, P. R. Mass customization and mass personalization Meet at the crossroads of industry 4.0: A case of augmented digital engineering. *Syst. Eng.* **26** (6), 715–727 (2023).
- Huang, D. et al. P2P cloud manufacturing based on a customized business model: an exploratory study. *Sensors* **23** (6), 3129 (2023).
- Ruediger-Flore, P., Glatt, M., Hussong, M. & Aurich, J. C. CAD-based data augmentation and transfer learning empowers part classification in manufacturing. *Int. J. Adv. Manuf. Technol.* **125** (11), 5605–5618 (2023).
- Li, C., Zheng, P., Yin, Y., Wang, B. & Wang, L. Deep reinforcement learning in smart manufacturing: A review and prospects. *CIRP J. Manuf. Sci. Technol.* **40** (1), 75–101 (2023).
- Chen, X. et al. Reinforcement learning for distributed hybrid flowshop scheduling problem with variable task splitting towards mass personalized manufacturing. *J. Manuf. Syst.* **76** (1), 188–206 (2024).
- Ma, Y., Wang, X., Dang, K., Zhou, Y. & Yang, W. Intelligent recommendation system of the injection molding process parameters based on CAE simulation, process window, and machine learning. *Int. J. Adv. Manuf. Technol.* **128** (9), 4703–4716 (2023).

26. Sun, J. et al. Separated graph neural networks for recommendation systems. *IEEE Trans. Industr. Inf.* **19** (1), 382–393 (2022).
27. Fu, M. et al. A deep reinforcement learning recommender system with multiple policies for recommendations. *IEEE Trans. Industr. Inf.* **19** (2), 2049–2061 (2022).
28. Mgbemena, C. E. & Okeagu, F. N. Development of an IoT-based real-time remote monitoring device for the maintenance of injection moulding machines in plastic industries. *UNIZIK J. Eng. Appl. Sci.* **2** (1), 260–278 (2023).
29. Clark, S. N. et al. The burden of respiratory disease from formaldehyde, damp and mould in english housing. *Environments* **10** (8), 136 (2023).
30. Hu, X., Tian, Y., Nagato, K., Nakao, M. & Liu, A. Opportunities and challenges of ChatGPT for design knowledge management. *Procedia CIRP.* **119** (1), 21–28 (2023).
31. Zheng, C. et al. Knowledge-based engineering approach for defining robotic manufacturing system architectures. *Int. J. Prod. Res.* **61** (5), 1436–1454 (2023).
32. Asprino, L., Daga, E., Gangemi, A. & Mulholland, P. Knowledge graph construction with a façade: a unified method to access heterogeneous data sources on the web. *ACM Trans. Internet Technol.* **23** (1), 1–31 (2023).
33. Demeter, C., Fechner, D. & Dolnicar, S. Progress in field experimentation for environmentally sustainable tourism—A knowledge map and research agenda. *Tour. Manag.* **94** (1), 104633 (2023).
34. Liu, X. & Zhong, S. Personalised recommendation algorithm of network resources based on collaborative knowledge map. *Int. J. Auton. Adapt. Commun. Syst.* **16** (2), 159–174 (2023).
35. Tamašauskaitė, G. & Groth, P. Defining a knowledge graph development process through a systematic review. *ACM Trans. Softw. Eng. Methodol.* **32** (1), 1–40 (2023).
36. Turukmane, A. V. et al. Multispectral image analysis for monitoring by IoT based wireless communication using secure locations protocol and classification by deep learning techniques. *Optik* **271**, 170122 (2022).
37. Gao, X. et al. Divide and conquer Q-Learning (DCQL) algorithm based photovoltaic (PV) array reconfiguration scheme for alleviating the partial shading influence. *Sol. Energy.* **249** (1), 21–39 (2023).
38. Jia, Y. & Zhou, X. Y. q-Learning in continuous time. *J. Mach. Learn. Res.* **24** (161), 1–61 (2023).
39. Nejatdarabi, S. & Mohebbi, M. Predicting the rehydration process of mushroom powder by multiple linear regression (MLR) and artificial neural network (ANN) in different rehydration medium. *J. Food Meas. Charact.* **17** (2), 1962–1973 (2023).
40. Del Serrone, G. & Moretti, L. A Stepwise regression to identify relevant variables affecting the environmental impacts of clinker production. *J. Clean. Prod.* **398** (1), 136564 (2023).

### Author contributions

Chengcai He, Jiaying Deng, Jingchun Wu, Beicheng Qin, Jinxiang Chen, Yan Li, Qiangsheng Huang contributed to conception and design of the study. Chengcai He, Jiaying Deng organized the database. Jingchun Wu performed the statistical analysis. Beicheng Qin, Jinxiang Chen, Yan Li wrote the first draft of the manuscript. Chengcai He, Jiaying Deng, Jingchun Wu, Beicheng Qin, Jinxiang Chen, Yan Li, Qiangsheng Huang wrote sections of the manuscript. All authors contributed to manuscript revision, read, and approved the submitted version.

### Declarations

#### Competing interests

The authors declare no competing interests.

#### Conflict of interest

This study does not have competing interests as defined by nature research, or other interests that may be considered to influence the results reported and discussed herein.

### Additional information

**Supplementary Information** The online version contains supplementary material available at <https://doi.org/10.1038/s41598-025-08399-z>.

**Correspondence** and requests for materials should be addressed to B.Q.

**Reprints and permissions information** is available at [www.nature.com/reprints](http://www.nature.com/reprints).

**Publisher's note** Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

**Open Access** This article is licensed under a Creative Commons Attribution-NonCommercial-NoDerivatives 4.0 International License, which permits any non-commercial use, sharing, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if you modified the licensed material. You do not have permission under this licence to share adapted material derived from this article or parts of it. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by-nc-nd/4.0/>.

© The Author(s) 2025