frontiers | Frontiers in Artificial Intelligence

TYPE Original Research
PUBLISHED 12 January 2026
DOI 10.3389/frai.2025.1745372

Check for updates

OPEN ACCESS

EDITED BY
Gaolei Li,
Shanghai Jiao Tong University, China

REVIEWED BY
Yukan Hou,
Northwestern Polytechnical University, China
Zhiqiu Xia,
Qilu University of Technology, China
Yuan Cao,
Ocean University of China, China

*CORRESPONDENCE
Xiting Peng
✉ xt.peng@sut.edu.cn

RECEIVED 13 November 2025
REVISED 02 December 2025
ACCEPTED 18 December 2025
PUBLISHED 12 January 2026

CITATION
Yang W, Liang J, Zhang X and Peng X (2026) Smart manufacturing-driven probabilistic process planning for components via AP-BiLSTM-ATT.
Front. Artif. Intell. 8:1745372.
doi: 10.3389/frai.2025.1745372

Smart manufacturing-driven probabilistic process planning for components via AP-BiLSTM-ATT

Wei Yang[1], Jinyan Liang[2], Xiaoyu Zhang[3] and Xiting Peng[1]*

[1]School of Information Science and Engineering, Shenyang University of Technology, Shenyang, China, [2]School of Computer Science and Engineering, College of Arts & Information Engineering, Dalian Polytechnic University, Dalian, China, [3]School of Artificial Intelligence, Shenyang University of Technology, Shenyang, China

In the context of smart manufacturing, improving the quality and efficiency of process planning, especially in the processing of complex parts, has become a key factor influencing the level of intelligence in manufacturing systems. However, most current process planning methods still heavily rely on manual expertise, leading to problems such as difficulty in knowledge reuse, low planning efficiency, and slow response times, which are inadequate to meet the diverse and changing needs of engineering applications. To address these issues, this paper proposes an algorithm for Assembly Process Reasoning and Decision-making based on Bidirectional Long Short-Term Memory with Attention (AP-BiLSTM-ATT), which aims to deeply explore the hidden relationships between the multi-dimensional features of parts and process plans, thereby achieving probabilistic modeling of process decisions. Specifically, the attributes, geometric features, and historical process plans of parts are first labeled and vectorized, transforming traditional process knowledge into structured data representations suitable for deep learning models. A BiLSTM network model, integrated with a multi-head attention mechanism, is then constructed to capture contextual dependencies and semantic weight distributions between features, enhancing the model's ability to express complex process relationships. During training, the model learns the mapping distribution between features and processes from a large-scale historical process dataset, enabling intelligent reasoning and recommendation of process plans for new parts. The results show that this method outperforms traditional methods in terms of accuracy, response speed, and generalization ability in process planning, providing effective support for enhancing the intelligence of complex part process planning and laying a foundation for the structured expression and intelligent application of manufacturing process knowledge.

KEYWORDS

AP-BiLSTM-ATT, intelligent reasoning, knowledge representation, process planning, smart manufacturing

1 Introduction

With the continuous advancement of intelligent manufacturing technology and the deep integration of advanced manufacturing and information technologies, manufacturing enterprises are achieving key breakthroughs in improving product quality, increasing production efficiency, and reducing production costs. Intelligent manufacturing has widely penetrated all stages of the product lifecycle, including product design, production manufacturing, and service maintenance. Research shows that as emerging manufacturing industries accelerate the deployment of intelligent technologies, the production efficiency of their manufacturing systems has increased by 17%–20%. However, as a critical link between product design and production execution, manufacturing process planning still relies primarily on human-machine interaction, with its core decision-making process heavily dependent on engineers' professional knowledge and experience.

In this context, the role of process planning in ensuring product quality, improving machining efficiency, and optimizing production costs becomes particularly significant. Especially in the manufacturing of complex structural parts, efficiently reasoning out the most appropriate process plan from the multi-dimensional features of parts has always been a core challenge of intelligent process planning. In contrast, traditional process planning often requires engineers to manually devise plans based on their experience, which is not only time-consuming and inefficient but also prone to subjective influences, making it difficult to meet the modern manufacturing industry's demands for rapid response and precise decision-making. The conceptual flow of this planning task is depicted in Figure 1.

With the gradual popularization of modern Computer-Aided Process Planning (CAPP) systems, enterprises have accumulated a wealth of historical process data. How to effectively extract valuable knowledge from this large amount of historical data and utilize it through intelligent methods has become a core issue in process reasoning and decision support. For example, Agrawal et al. (2009) proposed a multi-agent distributed CAPP system composed of a global management agent, design agent, and optimization agent. Using a backward-chaining reasoning mechanism, the system realized intelligent decision-making for complex processes. Qian et al. (2023) built a process-oriented knowledge ontology library for assembly sequence planning and applied a Mixed-Integer Linear Programming (MILP) model to optimize assembly actions and part sequences with the objective of minimizing assembly time. Combined with a human-computer collaboration visualization tool, they achieved rapid automatic generation of assembly plans. Mou and Gao (2020) proposed a fuzzy comprehensive evaluation method based on historical machining data for assessing process plan reliability, improving robustness in multi-objective and uncertain scenarios. Wang et al. (2015) modeled the process planning problem as a directed graph and used a two-stage ant colony algorithm for parallel optimal path search, significantly reducing production costs and improving algorithm efficiency. In addition, Rojek (2010) evaluated the performance of Multilayer Perceptrons (MLP) and Radial Basis Function (RBF) networks in intelligent CAPP systems, demonstrating that they outperform traditional rule-based methods in tool selection and operation sequencing. Deb et al. (2006) proposed a neural network-based method for selecting machining operations, automatically determining process parameters and tool configurations for rotationally symmetric parts.

However, traditional methods still face several critical challenges: high costs for knowledge acquisition and maintenance, with rules, ontologies, and templates requiring frequent updates by experts; insufficient structuring of historical data, making it difficult to fully leverage heterogeneous and multi-source process instances; and limited real-time responsiveness to dynamic environments, hindering rapid adjustments to production changes.

To overcome these bottlenecks, researchers have recently introduced machine learning, deep learning, and reinforcement learning technologies into CAPP systems. Zhang et al. (2022) developed an intelligent decision-making system that maps assembly units and process features into a multidimensional vector space, optimizing assembly sequence planning via supervised learning models, thereby significantly enhancing the system's generalization and automation capabilities. Jiang et al. (2024) proposed a fine-grained assembly sequence planning method based on knowledge graphs and deep reinforcement learning, where assembly operations are modeled as continuous and discrete processes, constructing a dynamic graph and applying an improved Deep Q-Network (DQN) to enable real-time decision-making under complex constraints with hierarchical Seq2Seq neural reasoning. Zhu et al. (2024) designed a two-stage Seq2Seq neural network that captures both assembly sequences and contact point selections through hierarchical reasoning, providing highly flexible process planning for robotic assembly. Mortlock et al. (2021) integrated Graph Neural Networks (GNNs) within a cognitive digital twin framework to couple real-time shop floor data with process models, supporting dynamic re-planning and predictive maintenance.

Despite the significant progress made by current intelligent process reasoning methods in improving decision-making efficiency and accuracy, they still face several major challenges. First, deep learning methods heavily rely on large amounts of accurately labeled historical data, and acquiring high-quality labeled data is both time-consuming and costly for manufacturing enterprises. Meanwhile, although knowledge graphs effectively integrate process knowledge, their construction and maintenance are complex and labor-intensive, becoming increasingly difficult as the system evolves.

To address these challenges, this paper proposes the following contributions:

A process reasoning model (AP-BiLSTM-ATT) based on part attributes, geometric features, and process plan labeling and vectorization is proposed. The model outputs a probability distribution

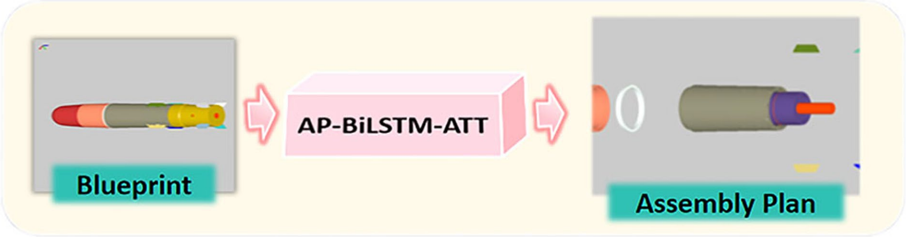


FIGURE 1
Simulation results for the network.

over candidate process plans via a final softmax layer, enabling uncertainty quantification and ranking of multiple feasible plans by their likelihood. This model effectively captures the multidimensional features of parts, reduces reliance on large-scale labeled datasets, and provides efficient process recommendations.

An attention mechanism (ATT) is introduced into the process reasoning framework, enabling the model to dynamically focus on key information in part features. This enhances the precision of process plan reasoning, reduces the dependency on complex ontology and knowledge graph construction, and improves model interpretability.

Experimental results show that the proposed method can quickly and accurately recommend optimal process plans for new parts without relying on complex graphs or massive labeled data. The results demonstrate that the method significantly outperforms traditional approaches in terms of accuracy, recommendation speed, and interpretability, effectively improving process planning efficiency.

2 Related work

Assembly Process Planning (APP), as a key stage within the smart manufacturing workflow, aims to generate efficient and rational assembly plans while satisfying assembly constraints and resource limitations. Among its components, Assembly Sequence Planning (ASP) constitutes the core of assembly, directly impacting the efficiency, quality, and production cost of product assembly. Consequently, how to efficiently and accurately achieve the automatic generation of assembly sequences has become a major focus of research both domestically and internationally.

2.1 Traditional rule-based and heuristic search methods

Early research primarily relied on traditional methods based on geometric features and assembly constraint rules. Torres et al. (2003) proposed an assembly relation model and solved the assembly sequence planning problem by leveraging the reverse logic of disassembly and assembly. Dini et al. (1999) developed a mathematical representation model of the assembly process based on the assembly interference matrix and contact matrix, achieving the quantitative evaluation and selection of assembly sequences. Although these methods offer good intuitiveness and interpretability, their modeling efficiency and degree of automation remain limited when applied to industrial scenarios characterized by increasingly complex assembly structures and a large number of components.

To overcome these bottlenecks, a significant body of research in recent years has introduced heuristic and intelligent optimization algorithms to enhance the efficiency of assembly sequence planning. For instance, Chen and Liu (2001) proposed an adaptive genetic algorithm (GA) to address the poor adaptability of traditional GA operators. Abdullah et al. (2019) constructed multi-objective assembly optimization models using the Artificial Bee Colony algorithm and the Moth-Flame Optimization algorithm, respectively. Beyond these, variants of Particle Swarm Optimization (PSO) have been explored. Zhang (2023) developed an Improved PSO (IPSO) that redefines particle update rules and incorporates GA-style mutation to accelerate convergence and escape local optima. Wu et al. (2019) applied a PSO-based method leveraging assembly direction, interference, and sequence-relation matrices to obtain optimal sequences under fixture constraints. To generate diverse Pareto-optimal assembly plans, Wan et al. (2024) introduced a Multiple Optimal Solutions GA (MOSGA), balancing assembly time and resource consumption for large modular assemblies MDPI. Hybrid swarm–behavior algorithms have also been developed. Wu et al. (2019) proposed SOS-ACO, coupling Symbiotic Organisms Search with Ant Colony Optimization to adaptively tune pheromone parameters, achieving near-optimal sequences in fewer iterations and Zhang et al. (2025) presented an SOS-PSO hybrid that integrates immune-inspired selection with PSO, demonstrating superior robustness and convergence in constrained multi-agent assembly scenarios.

Although these methods have demonstrated promising potential in improving assembly efficiency and reducing resource consumption, they fundamentally remain heuristic search frameworks—sensitive to initial parameter settings and prone to local optima in large-scale combinatorial spaces, with limited capabilities for deep modeling of assembly knowledge.

2.2 Machine learning and deep learning methods

In order to further enhance the intelligence level of assembly process planning, some studies have begun exploring the application of machine learning methods. Research has been conducted to develop an assembly prediction system based on artificial neural networks. This system constructs an assembly evaluation function and employs supervised learning to predict and optimize assembly steps. Furthermore, a hybrid assembly sequence optimization model has been proposed, which integrates multiple neural network structures with K-means clustering. Although these approaches perform well in specific experimental scenarios, their relatively shallow network structures and limited capability to model temporal features restrict their ability to fully capture the contextual dependencies and long-term constraint information inherent in the assembly process. Guo et al. (2024) proposed a DRL method with multiple starting-node exploration was introduced to address dynamic changes in machining resources. By augmenting the state-space exploration with varied initial conditions, it achieved superior resource utilization and planning robustness compared to standard RL baselines. Li et al. (2024) has been modeled as a Markov Decision Process and solved via a heterogeneous Graph Neural Network combined with Proximal Policy Optimization. This end-to-end approach captured operation–machine relationships and outperformed MILP-based methods in both solution quality and computation time on large-scale instances. In assembly sequence planning, Neves and Neto (2022) applied DRL with parametric action spaces and dual reward signals—reflecting user ergonomic preferences and cycle-time minimization—comparing A2C, DQN, and Rainbow; Rainbow achieved near-optimal performance after 10,000 episodes, surpassing tabular Q-Learning in complex deterministic and stochastic scenarios. For additive manufacturing, Mozaffar et al. (2020) developed a DRL-based toolpath planning platform that learns deposition strategies under dense reward structures, demonstrating high fidelity to expert-designed toolpaths and adaptability to arbitrary geometries. Wang et al. (2023) proposed a dual-attention DRL model was proposed for flexible job shop scheduling—a close relative of process planning—where interconnected operation-message and machine-message attention blocks guide priority decisions. This framework achieved solution quality comparable to exact methods on benchmark tasks, highlighting the promise of attention architectures in capturing complex process-machine interactions.

It is worth noting that deep learning-based methods for assembly sequence modeling are still in an exploratory stage, and related research remains relatively scarce. In tackling complex assembly tasks characterized by sequentiality and structural dependency, sequence modeling capability becomes critical. The selection of BiLSTM with attention is theoretically grounded in the sequential nature and long-range dependencies inherent in process planning tasks. Process plans typically involve ordered sequences of operations where previous decisions influence subsequent steps, and critical dependencies may span across multiple operations. BiLSTM effectively captures bidirectional contextual dependencies in these process sequences, while the attention mechanism dynamically weights important features and operations, addressing both local patterns and long-range dependencies that are characteristic of complex manufacturing processes. Bidirectional Long Short-Term Memory (BiLSTM) networks, known for their ability to capture both historical and future information simultaneously, have been widely used in fields such as semantic recognition and event prediction. When combined with the attention mechanism, the network can dynamically allocate focus weights, emphasizing key steps in the assembly process, thereby enhancing the model's ability to understand complex assembly logic.

3 Materials and methods

In this paper, we first extract critical assembly information from the three-dimensional (3D) CAD models of products, including the mating surfaces of each part, geometric attributes, and inter-surface constraints within the assembly. Based on the extracted information, we construct an assembly feature representation model oriented toward process planning, thereby establishing a training sample mapping between assembly features and typical process operations. This mapping captures implicit rules linking part assembly types, mating attributes, and corresponding assembly operations—such as insertion, press-fitting, welding, screwing, and positioning/clamping—which serve as supervised labels for the subsequent deep learning model training. The overall architecture of the proposed AP-BiLSTM-ATT framework is illustrated in Figure 2.

3.1 Data preprocessing

The core objective of the AP-BiLSTM-ATT algorithm is to maximize the prediction probability of part process plans based on the key attributes and features of the parts during model training. To achieve this goal, it is essential to first prepare a training dataset that includes the mapping relationships between parts and their corresponding process plans. During the training phase, the system learns the nonlinear mapping between part features and process plans, thereby enabling efficient and accurate process recommendations.

The data preprocessing involves a detailed "Labeling and Vectorization" procedure to transform part attributes and process knowledge into a structured format. First, process plan labeling is performed: each unique sequence of machining operations (e.g., "Drilling → Rough Turning → Finish Turning") is assigned a unique categorical label, which serves as the target for the model's multi-class prediction task. Second, part feature vectorization is conducted: categorical features (e.g., material types) are encoded using one-hot encoding; numerical features (e.g., diameter, length) are normalized to a [0, 1] range; and machining operations are mapped to dense embedding vectors through a trainable embedding layer, enabling the model to learn semantic relationships between operations.

In the specific data preprocessing phase, the training set consists of A part samples and their corresponding B process plans, forming a large number of part–process plan pairs as training examples. Each training sample not only contains the basic coding information of the part (such as part ID and type) but also includes the code of the associated process plan and multi-dimensional feature information of

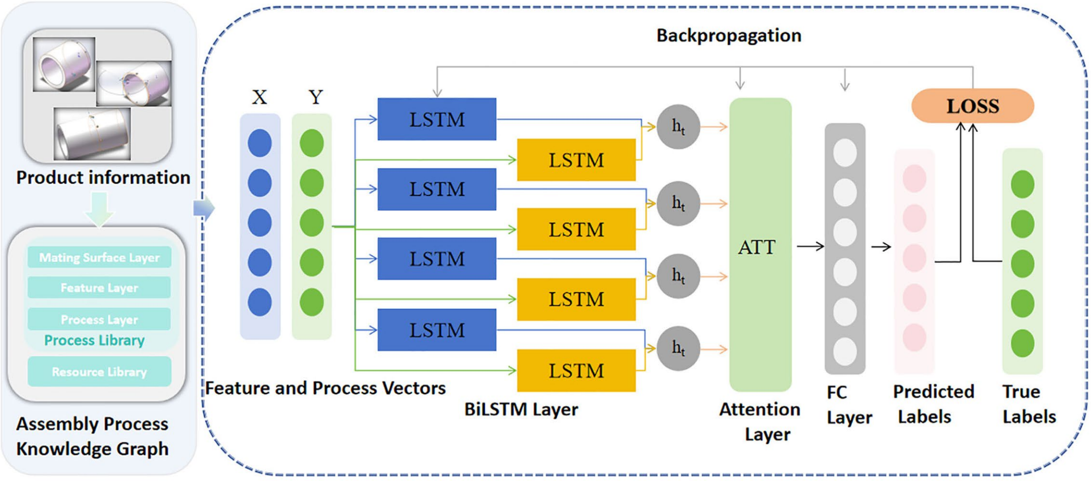


FIGURE 2
The proposed algorithm: AP-BiLSTM-ATT.

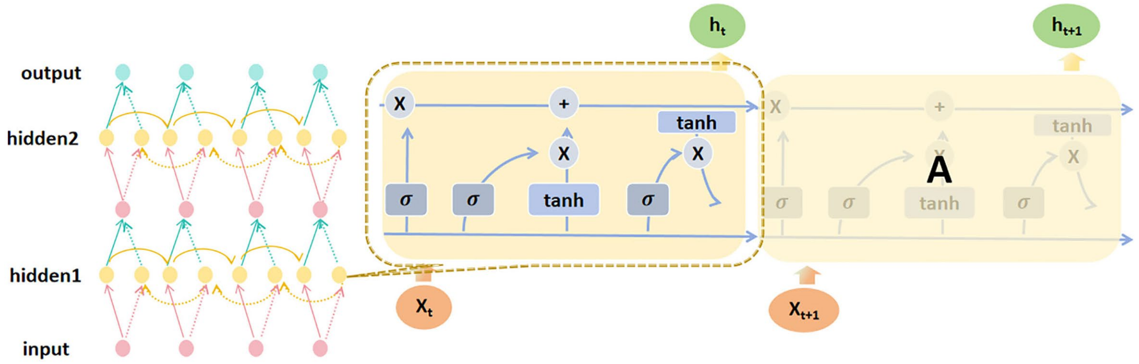


FIGURE 3
BiLSTM architecture diagram.

the part (such as dimensional parameters, surface requirements, material types, and structural complexity). These inputs are transformed into feature vectors that feed into the model to establish a deep mapping relationship between part features and process operations.

To enhance data quality and model performance, additional preprocessing steps are employed. Encoding and vectorization ensure all features are in neural-network-readable format, while data augmentation through slight feature value perturbation expands the effective sample size and improves model robustness to input variations. Collectively, these steps ensure the input data maintains its integrity while being optimally prepared for the AP-BiLSTM-ATT model, establishing a solid foundation for reliable process planning recommendations.

3.2 AP-BiLSTM-ATT

Specifically, this study incorporates Bidirectional Long Short-Term Memory (BiLSTM) networks combined with Attention Mechanism to address the problem of process plan prediction. BiLSTM is a deep learning model capable of capturing both forward and backward dependencies within sequential data, making it particularly suitable for handling complex sequential data and long-range dependencies. The detailed structure of the BiLSTM component is depicted in Figure 3. The architectural selection of BiLSTM with attention is strategically aligned with the fundamental characteristics of process planning tasks. In manufacturing environments, process sequences exhibit strong temporal dependencies where early-stage decisions (e.g., material selection and rough machining parameters) fundamentally constrain subsequent operations (e.g., finishing and quality control). The BiLSTM component excels at modeling these bidirectional process flows, while the attention mechanism, whose architecture is detailed in Figure 4, addresses the critical challenge of long-range dependencies in manufacturing processes. This synergistic combination allows the model to not only understand local sequential patterns but also recognize global process constraints that span multiple manufacturing stages. By leveraging its bidirectional structure, BiLSTM performs computations in both the forward and backward directions of a time sequence, enabling a more comprehensive understanding of the contextual information in the input data. This significantly enhances the model's ability to extract and model features. At each time step t, the forward LSTM sequentially computes the current hidden state $h_t^{(f)}$ and cell state $c_t^{(f)}$ using the following steps.

First, the Input Gate determines the influence of the current input x_t on the cell state. The output i_t of the input gate is calculated as shown in Equation 1:

$$i_t = \sigma\left(W_i x_t + U_i h_{t-1}^{(f)} + b_i\right) \tag{1}$$

where W_i is the input weight matrix, U_i is the hidden state weight matrix, b_i is the bias term, and σ is the sigmoid activation function. $h_{t-1}^{(f)}$ is the hidden state from the previous time step.

Next, the Forget Gate controls the proportion of information from the previous time step's cell state $c_{t-1}^{(f)}$ to be retained in the current cell state $c_t^{(f)}$. The output f_t of the forget gate is computed as shown in Equation 2:

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1}^{(f)} + b_f\right) \tag{2}$$

where W_f and U_f are the forget gate weight matrices, and b_f is the bias term.

Then, the Output Gate determines the current hidden state $h_t^{(f)}$. The output o_t of the output gate is calculated as shown in Equation 3:

$$o_t = \sigma\left(W_o x_t + U_o h_{t-1}^{(f)} + b_o\right) \tag{3}$$

where W_o and U_o are the output gate weight matrices, and b_o is the bias term.

Next, the Cell State is computed by combining the previous time step's cell state and the contributions from the input and forget gates. The update equation for the cell state is as shown in Equation 4:

$$c_t^{(f)} = f_t \cdot c_{t-1}^{(f)} + i_t \cdot \tanh\left(W_c x_t + U_c h_{t-1}^{(f)} + b_c\right) \tag{4}$$

where W_c and U_c are the memory update weight matrices, and b_c is the bias term. The $\tan h$ is the hyperbolic tangent activation function. This equation combines the influence of the forget gate, which retains

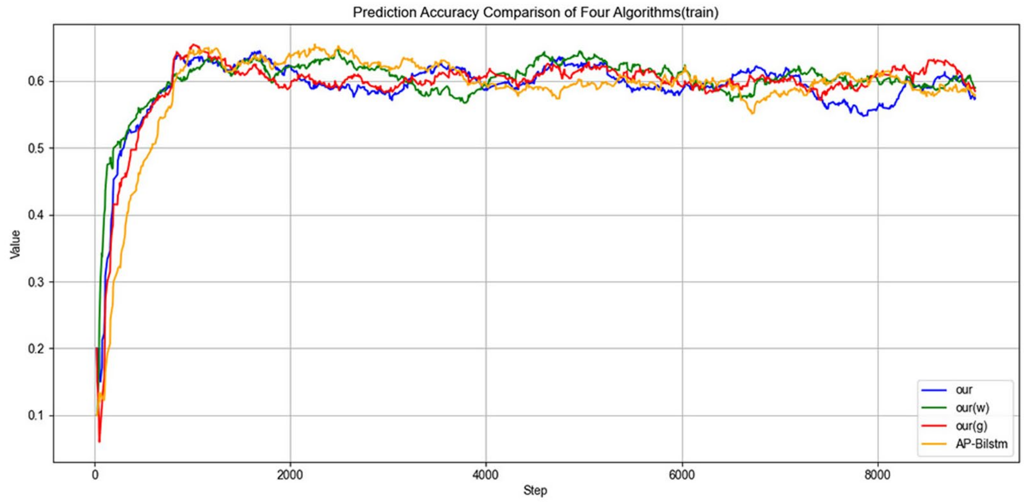


FIGURE 4
Attention mechanism architecture diagram.

the previous memory information, and the input gate, which introduces new information from the current time step to form the updated cell state $c_t^{(f)}$.

Finally, the hidden state is computed as the non-linear combination of the output gate and the current cell state, as shown in Equation 5:

$$h_t^{(f)} = o_t \cdot \tanh\left(c_t^{(f)}\right) \tag{5}$$

Through these steps, the forward LSTM effectively captures the long-term dependencies in the input sequence. At each time step, the forward LSTM updates its hidden state $h_t^{(f)}$ and cell state $c_t^{(f)}$, which are then passed on to the next time step.

Similarly, the backward LSTM processes the input features starting from the end of the sequence. It computes the hidden state and cell state in the reverse order. The forward and backward LSTM hidden states are then concatenated to form the final output of the bidirectional LSTM, which fully utilizes both past and future information from the input sequence.

After obtaining the concatenated bidirectional hidden representations from the BiLSTM network, an attention mechanism is introduced to further enhance the model's ability to capture critical information. While BiLSTM can effectively model long-range dependencies and contextual relationships in the input sequence, its output treats all time steps equally. This uniform treatment may dilute the influence of key time-step features on the final representation. The attention mechanism addresses this limitation by allowing the model to learn the relative importance of each time step in the sequence, enabling it to focus on features most relevant to process plan prediction.

Formally, let the hidden state sequence output by BiLSTM be $\mathbf{H} = \{\mathbf{h}_1, \mathbf{h}_2, \ldots, \mathbf{h}_T\}$, where $\mathbf{h}_t \in \mathbb{R}^d$ denotes the hidden state at time step t. The attention mechanism first computes a relevance score e_t for each hidden state, as shown in Equation 6:

$$e_t = \mathbf{v}^\top \tanh\left(\mathbf{W}_a \mathbf{h}_t + \mathbf{b}_a\right) \tag{6}$$

where $\mathbf{W}_a \in \mathbb{R}^{d_a \times d}$ is a learnable weight matrix, $\mathbf{v} \in \mathbb{R}^{d_a}$ is a weight vector, and $\mathbf{b}_a \in \mathbb{R}^{d_a}$ is a bias term. These scores are then normalized by a softmax function to obtain the attention weights α_t, as shown in Equation 7:

$$\alpha_t = \frac{\exp(e_t)}{\sum_{i=1}^{T} \exp(e_i)} \tag{7}$$

The context vector **c**, which serves as a weighted representation of the sequence, is computed as shown in Equation 8:

$$\mathbf{c} = \sum_{t=1}^{T} \alpha_t \mathbf{h}_t \tag{8}$$

This context vector integrates information from all time steps, with higher weights assigned to more informative steps, thereby enhancing the model's representation of critical process-related features.

The multi-head attention mechanism extends the standard attention by employing multiple attention heads (h = 8) in parallel. Each head learns distinct feature representations from different subspaces, enabling the model to capture diverse aspects of process planning semantics. Formally, for head i, the attention output is computed as shown in Equation 9:

$$head_i = Attention\left(\mathbf{H}W_i^Q, \mathbf{H}W_i^K, \mathbf{H}W_i^V\right) \tag{9}$$

where $W_i^Q \in R^{d \times d_k}, W_i^K \in R^{d \times d_k}$, and $W_i^V \in R^{d \times d_v}$ are learnable projection matrices for queries, keys, and values respectively, with $d_k = d_v = d / h = 12$. The outputs of all heads are concatenated and then linearly transformed, as shown in Equation 10:

$$MultiHead(\mathbf{H}) = Concat\left(\mathrm{head}_1,,..,,\mathrm{head}_h\right)W^O \tag{10}$$

where $W^{O} \in R^{\mathrm{h} \cdot d_v \times d}$ is the output projection matrix. This architectural design allows the model to jointly attend to information from different representation subspaces, effectively capturing various feature interactions in process planning, such as the relationships between geometric features, material properties, and process parameters.

A key innovation of our framework is its probabilistic output, which transforms the model from a deterministic classifier into a decision-support tool. Subsequently, the context vector **c** is passed through a fully connected layer followed by a softmax classifier to generate the predicted probability distribution over the candidate process plan labels, as shown in Equation 11:

$$\hat{\mathbf{y}} = Softmax\left(\mathbf{W}_s \mathbf{c} + \mathbf{b}_s\right) \tag{11}$$

Here, $\mathbf{W}_s \in \mathbb{R}^{K \times d}$ and $\mathbf{b}_s \in \mathbb{R}^{K}$ are the weights and bias of the classification layer, and K is the number of process plan categories. $\hat{\mathbf{y}}$ represents a probability distribution where each element denotes the likelihood of a corresponding process plan being the optimal choice. This output allows for: (1) Quantifying the uncertainty of the top recommendation, and (2) Ranking and presenting multiple feasible alternative plans to the process planner, thereby enabling more informed and flexible decision-making in a smart manufacturing context.

The entire model is trained with a cross-entropy loss function. This function quantifies the discrepancy between the predicted probabilities $\hat{\mathbf{y}}$ and the ground truth labels **y**, as presented in Equation 12:

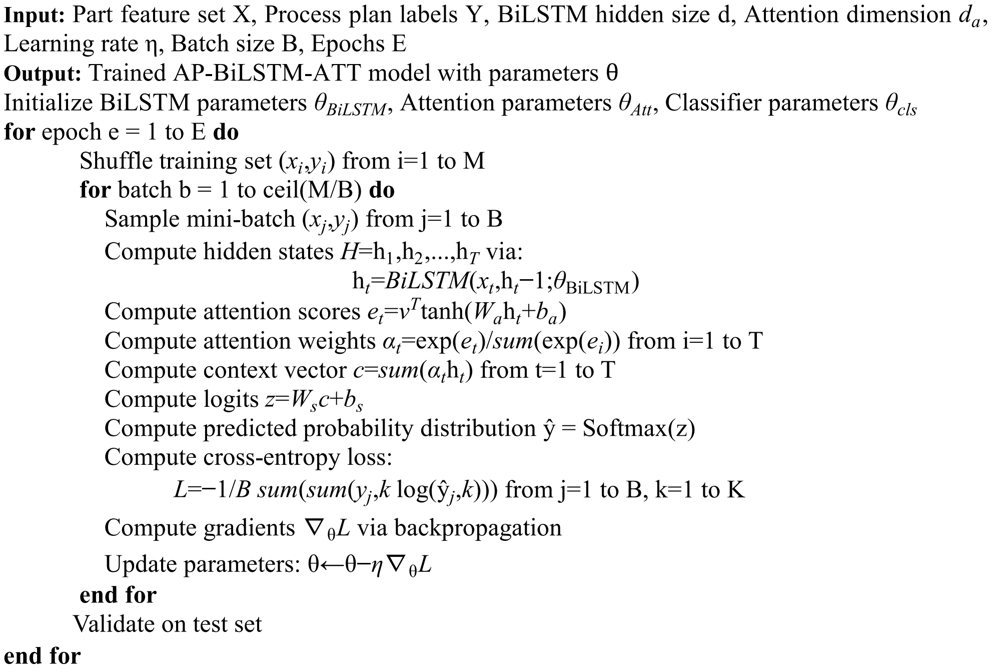
$$\mathcal{L} = -\sum_{i=1}^{K} y_i \log\left(\hat{y}_i\right) \tag{12}$$

where $\mathbf{y} \in \{0,1\}^{K}$ is a one-hot encoded vector representing the true label, and $\hat{y}_i$ denotes the predicted probability for class i. This loss is minimized during training using backpropagation to update all model parameters. The complete training procedure outlined above is summarized in Algorithm 1.

4 Experiments

4.1 Dataset

The custom dataset employed in this study consists of 1,000 instances of machining process data for various parts. This dataset was synthetically constructed to support research in data-driven process planning, comprising typical precision components such as shafts, plates, and housings to ensure diversity in part geometry and function. The historical process plans were generated based on domain expertise and standard manufacturing guidelines, with each plan representing a feasible sequence of operations (e.g., "Drilling → Rough Turning → Finish Turning"). These sequences were subsequently validated through simulation to ensure logical consistency and adherence to machining principles. Each instance includes a part description—encompassing material type (e.g., copper, 304 stainless steel, 45 steel, aluminum), geometric attributes (end face type, diameter, length), and technical specifications (flatness tolerance, surface roughness, hardness)—and its corresponding machining process plan, which records the sequential operations from rough to finish machining. This dataset provides a foundation for developing and validating intelligent process recommendation systems. Each instance is composed of two key components: part description and machining process plan. The part description



```
Input: Part feature set X, Process plan labels Y, BiLSTM hidden size d, Attention dimension $d_a$,
Learning rate η, Batch size B, Epochs E
Output: Trained AP-BiLSTM-ATT model with parameters θ
Initialize BiLSTM parameters $\theta_{BiLSTM}$, Attention parameters $\theta_{Att}$, Classifier parameters $\theta_{cls}$
for epoch e = 1 to E do
      Shuffle training set $(x_i,y_i)$ from i=1 to M
      for batch b = 1 to ceil(M/B) do
        Sample mini-batch $(x_j,y_j)$ from j=1 to B
        Compute hidden states $H$=$\mathrm{h}_1,\mathrm{h}_2,...,\mathrm{h}_T$ via:
                              $\mathrm{h}_t = BiLSTM(x_t,\mathrm{h}_t - 1;\theta_{\mathrm{BiLSTM}})$
        Compute attention scores $e_t = v^T \tanh(W_a \mathrm{h}_t + b_a)$
        Compute attention weights $\alpha_t = \exp(e_t)/sum(\exp(e_i))$ from i=1 to T
        Compute context vector $c = sum(\alpha_t \mathrm{h}_t)$ from t=1 to T
        Compute logits $z = W_s c + b_s$
        Compute predicted probability distribution ŷ = Softmax(z)
        Compute cross-entropy loss:
             $L = -1/B\ sum(sum(y_j,k \log(\hat{y}_j,k)))$ from j=1 to B, k=1 to K
        Compute gradients $\nabla_{\theta} L$ via backpropagation
        Update parameters: $\theta \leftarrow \theta - \eta \nabla_{\theta} L$
      end for
     Validate on test set
end for
```

ALGORITHM 1

Framework of AP-BiLSTM-ATT for process planning.

includes information such as material type (e.g., copper, 304 stainless steel, 45 steel, aluminum), end face type, flatness tolerance, part diameter, part length, hole requirements, coating requirements, surface roughness, and hardness grade. The machining process plan records the sequence of operations performed on each part, encompassing various stages from rough to finish machining, such as drilling, rough turning, finish turning, boring, coating, tapping, cutting, and surface treatment. This dataset reflects the diversity of materials and machining processes, providing valuable support for machine learning-based machining process recommendation systems. It aims to facilitate automated prediction and optimization, improving production efficiency while reducing manual intervention.

4.2 Experimental setup

The experiments in this study were conducted on an Intel(R) Core(TM) i7-8550 U CPU and an NVIDIA GeForce RTX 3080 Ti GPU with 8GB of RAM. The experimental model is implemented using the TensorFlow framework and employs a Bidirectional Long Short-Term Memory (Bi-LSTM) network coupled with an Attention Mechanism. Specifically, the Bi-LSTM consists of forward and backward LSTM units, each containing 100 hidden neurons and utilizing the ReLU activation function. Dropout regularization is applied, with a dropout keep probability of 0.7 for both the embedding and RNN layers. The Attention Mechanism weights the outputs of the LSTM layers to enhance the model's focus on critical information. Other hyperparameters, including embedding layer dimensions and L2 regularization, are detailed in Table 1. During training, the batch size is set to 160, the learning rate is initialized to 0.001, and the Adam optimizer with a learning rate decay strategy is employed. The dataset is divided into a 90 training set and a 10 validation set, with cross-validation used to ensure the robustness of the experiment.

TABLE 1 The description of experimental parameters.

Parameter	Configuration
System	Windows
GPU	GeForce RTX 3080 Ti
Coding environment	Python3.7
Tensorflow	v2.3.0
Numpy	v1.19.5
Rnn-dropout-keep-prob	0.7
Dropout-keep-prob	0.5
Emb-dropout-keep-prob	0.7
Embedding-dim	100
L2-reg-lambda	1e-5
Batch-size	10
Num-epochs	100
Evaluate-every	100
Learning rate	0.001
Decay-rate	0.9
Hidden-size	100
Dev-sample-percentage	0.1

To thoroughly evaluate the performance of the proposed AP-BiLSTM-ATT model (denoted as Ours), we compare it against several carefully designed baseline and ablation models to isolate the contribution of key components.

AP-BiLSTM: This is an ablation model that removes the attention mechanism from our full model. It retains the same BiLSTM layers but uses the last hidden state for classification instead of the attention-weighted context vector. The comparison between Ours and AP-BiLSTM is designed to directly quantify the performance gain attributable to the attention mechanism.

Ours(g) & Ours(w): These two model variants were designed to evaluate the impact of different word embedding initialization strategies on performance. Specifically, "Ours(g)" initializes the embedding layer using pre-trained GloVe vectors, while "Ours(w)" initializes it with vectors generated by the Word2Vec method. Our final model (Ours) employs an end-to-end trained embedding layer with random initialization. Their inclusion aims to demonstrate that our final choice of an end-to-end training strategy outperforms approaches reliant on external pre-trained models, thereby highlighting the simplicity and effectiveness of our final architecture.

4.3 Experimental evaluation metrics

This work uses five key metrics to evaluate the performance of the process plan prediction model: HR@n, MRR@n, Process Plan Prediction Accuracy (Seq-Acc), Computation Time (CT), and Loss. HR@n (Hit Rate at Top n) measures whether the predicted process plan is ranked within the top n positions in the candidate list. If the predicted result is within the top n, the value is 1; otherwise, it is 0. MRR@n (Mean Reciprocal Rank at Top n) calculates the average reciprocal rank of the first correct prediction in the candidate list, where the value at the k-th position is 1/k, and if the prediction is not in the candidate list, the value is 0. Process Plan Prediction Accuracy (Seq-Acc) evaluates whether the model successfully predicts the correctness of the entire process plan, indicating the model's ability to match the process plan. Computation Time (CT) measures the time required for the model to make the process plan prediction, reflecting the model's efficiency, which is especially significant in practical applications. Loss represents the difference between the model's output and the true labels; a lower loss indicates better model performance. By utilizing these metrics, the accuracy, efficiency, and optimization of the model can be comprehensively assessed.

4.4 Experimental result analysis

The experiment focuses on analyzing the model's performance on both the training and validation sets, with particular attention to the changes in accuracy and loss.

(1) Accuracy: First, Figure 5 illustrates the variation in accuracy on the training set. As the training progresses, the model's accuracy on the training set gradually increases, indicating that the model successfully learns the features of the data and optimizing its internal parameters to enhance prediction capability. Although there may be some fluctuations in the early stages, the overall accuracy stabilizes and steadily rises,

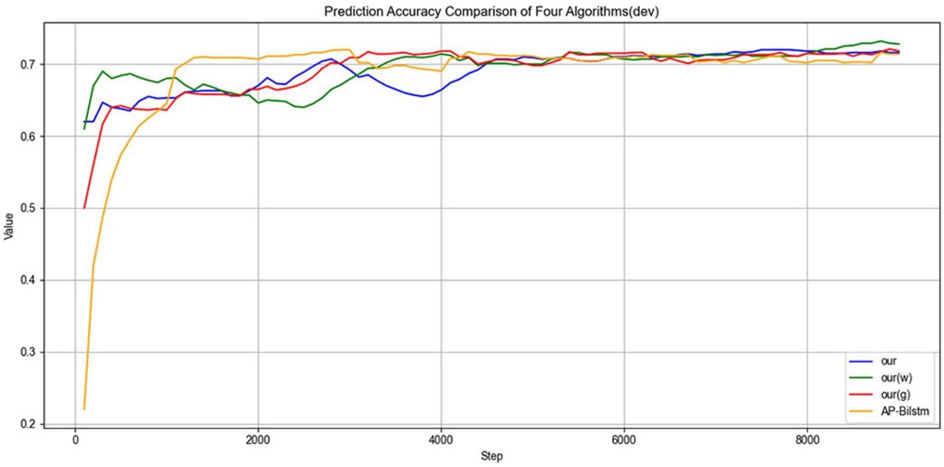


FIGURE 5
Training accuracy over epochs.

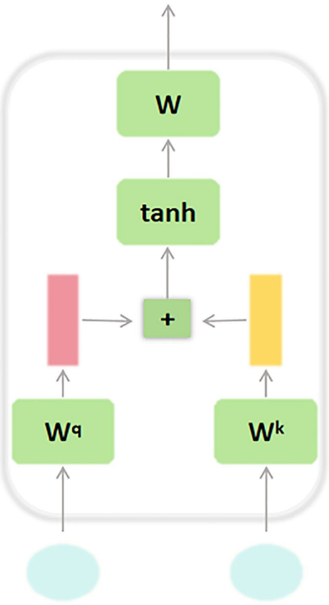


FIGURE 6
Validation accuracy over epochs.

reflecting the continuous improvement of the training process.

Figure 6 illustrates the accuracy trends on the validation set. It can be observed that the proposed method achieves a higher initial accuracy compared to other algorithms during the validation process, indicating that the model possesses strong feature extraction capabilities at an early stage. As training progresses, the proposed method consistently maintains its advantage in accuracy and ultimately converges to a higher accuracy level than the competing algorithms. These results demonstrate that the proposed model not only exhibits strong learning ability in the early training phase but also shows greater stability and convergence performance throughout the training process, thereby confirming its superiority in terms of generalization capability and predictive effectiveness.

(2) Loss: Figure 7 illustrates the variation in Loss on the training set. As training progresses, the Loss gradually decreases, indicating that the model is reducing prediction errors during the optimization process. Although some fluctuations may occur in the early stages, the Loss stabilizes and converges to a lower level as training continues, suggesting that the model progressively improves its fit to the training data.

Figure 8 illustrates the variation in Loss on the validation set. It is evident that the proposed method consistently achieves lower Loss values throughout the validation process compared to other algorithms. This indicates that the model is more effective in minimizing prediction errors on unseen data, demonstrating superior generalization capability. Moreover, the lower validation Loss suggests that the model maintains a good fit without overfitting, thereby exhibiting enhanced robustness and stability.

(3) HR@k and MRR@k: Based on the evaluation metrics presented in Table 2, we can observe the model's performance in recommendation accuracy at various positions.

HR@1 and MRR@1: At rank 1, the proposed method (ours) achieves both HR and MRR values of 0.70, significantly outperforming the other methods. This indicates that 70% of the test samples correctly identify the target item at the top of the recommendation list, with an average reciprocal rank of 1.0. These results demonstrate the superior predictive accuracy of the proposed model at the top recommendation position and its effectiveness in identifying the most relevant content for users.

HR@3 and MRR@3: At rank 3, the proposed method achieves an HR of 1.0000 and an MRR of 0.8433, indicating a 100% hit rate within the top three recommendations and a notably higher average reciprocal rank compared to baseline methods. This further confirms the model's strong recall ability and effective ranking performance within the top three positions.

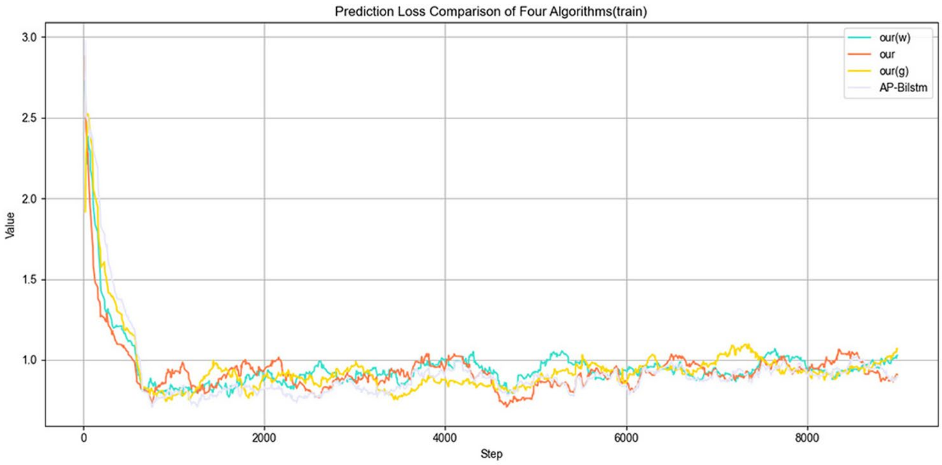


FIGURE 7
Training loss over epochs.

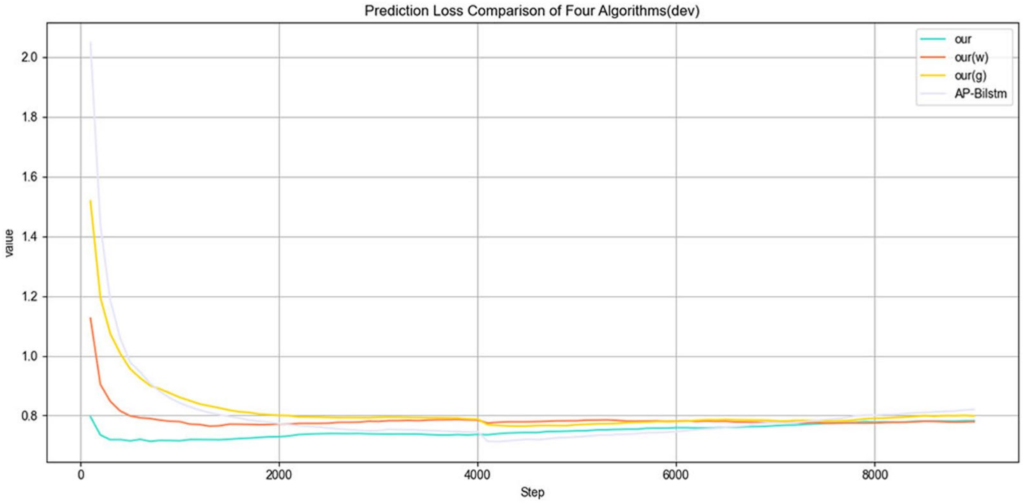


FIGURE 8
Validation loss over epochs.

TABLE 2 HR@k and MRR@k evaluation metrics.

Method	HR			MRR		
	@1	@3	@5	@1	@3	@5
AP-BiLSTM	0.5973	0.9360	0.9688	0.5973	0.7535	0.7613
Ours(g)	0.5978	0.9311	0.9674	0.5978	0.7509	0.7594
Ours(w)	0.5953	0.9292	0.9655	0.5953	0.7491	0.7575
ours	0.7000	1.0000	1.0000	0.7000	0.8433	0.8433

HR@5 and MRR@5: At rank 5, the proposed method continues to maintain an HR of 1.0000 and an MRR of 0.8433, demonstrating high stability. Although other methods also achieve relatively high HR values at this rank, their MRR scores remain lower than the proposed method. This indicates that our model not only achieves perfect recall within the top five recommendations but also ensures superior ranking quality, reflecting robust and accurate recommendation performance.

TABLE 3 Prediction speed and feature importance (FI) scores in process planning.

Method	AP-BiLSTM	Ours(g)	Ours(w)	Ours
Time	198	196	206	191
F1 score	0.5791	0.5768	0.5720	0.5830

Time Efficiency: Regarding time consumption, the proposed method (ours) demonstrates relatively good efficiency, requiring 191 units of time as shown in Table 3, which is slightly lower than the compared methods. This suggests that the model can achieve a relatively faster training or inference speed while maintaining comparable performance, indicating its practical potential.

F1 Score Performance: In terms of F1 score, the proposed method achieves a respectable value of 0.5830 (see Table 3), marginally higher than the other methods. This result indicates that the model attains a reasonable balance between precision and recall, showing stable predictive capability. Taken together with the time metric, the model demonstrates some improvement in both efficiency and effectiveness, reflecting promising potential for further optimization.

5 Discussion

This paper proposes an Assembly Process Reasoning and Decision-making algorithm based on Bidirectional Long Short-Term Memory (BiLSTM) and attention mechanisms (AP-BiLSTM-ATT), aiming to alleviate, to some extent, common challenges in traditional process planning such as difficulties in knowledge reuse, low efficiency, and slow response times. The approach transforms part attributes, geometric features, and historical process plans into structured data representations, and integrates a BiLSTM network with a multi-head attention mechanism to explore the potential relationships between part features and process plans, thereby better capturing contextual dependencies and semantic weight information. During training, the model learns mappings between features and processes from a large-scale historical process dataset, enabling a basic reasoning and recommendation for new part process plans. We analyzed the attention distribution during model prediction and found it effectively focuses on key features consistent with domain knowledge. For instance, the model assigns higher weights to "surface_roughness" when recommending finishing operations, while paying more attention to "material_hardness" for rough machining decisions. Experimental results indicate that, compared to some traditional approaches, the proposed method demonstrates measurable improvements in terms of accuracy, response speed, and generalization ability, suggesting its potential in complex part process planning tasks. However, this study has limitations that warrant attention. The primary constraint lies in the use of a custom-generated dataset, which may affect generalizability to diverse real-world scenarios. Furthermore, practical deployment challenges such as integration with existing PLM/CAM systems and meeting real-time requirements need addressing. Future work will focus on validation with larger industrial datasets and developing prototype systems for practical implementation.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: xt.peng@sut.edu.cn.

Author contributions

WY: Data curation, Methodology, Writing – review & editing. JL: Data curation, Writing – original draft, Writing – review & editing. XZ: Funding acquisition, Investigation, Project administration, Supervision, Writing – review & editing. XP: Formal analysis, Investigation, Resources, Supervision, Writing – review & editing.

Funding

The author(s) declared that financial support was received for this work and/or its publication. This study is supported in part by the Key Technologies Research and Development Program (grant no. 2024YFF0617200), Liaoning Science and Technology Major Project (grant no. 2024JH1/11700043), the Natural Science Foundation of Liaoning Province (grant no. 2024-bs-102), LLL24KF-01-01), the Basic Scientific Research Project of the Education Department of Liaoning Province (grant no. LJ222410142043).

Acknowledgments

The authors are thankful to the reviewers and editors for their valuable comments and suggestions.

Conflict of interest

The author(s) declared that this work was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

The handling editor GL declared a past co-authorship with the author XP.

Generative AI statement

The author(s) declared that Generative AI was not used in the creation of this manuscript.

Any alternative text (alt text) provided alongside figures in this article has been generated by Frontiers with the support of artificial intelligence and reasonable efforts have been made to ensure accuracy, including review by the authors wherever possible. If you identify any issues, please contact us.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

References

Abdullah, A., Ab Rashid, M. F. F., Ponnambalam, S. G., and Ghazalli, Z. (2019). Energy efficient modeling and optimization for assembly sequence planning using moth flame optimization. *Assembly Autom.* 39, 356–368. doi: 10.1108/AA-06-2018-091

Agrawal, R., Shukla, S. K., Kumar, S., and Tiwari, M. K. (2009). Multi-agent system for distributed computer-aided process planning problem in e-manufacturing environment. *Int. J. Adv. Manuf. Technol.* 44, 579–594. doi: 10.1007/s00170-008-1844-3

Chen, S. F., and Liu, Y. J. (2001). An adaptive genetic assembly-sequence planner. *Int. J. Comput. Integr. Manuf.* 14, 489–500. doi: 10.1080/09511920110034987

Deb, S., Ghosh, K., and Paul, S. (2006). A neural network based methodology for machining operations selection in computer-aided process planning for rotationally symmetrical parts. *J. Intell. Manuf.* 17, 557–569. doi: 10.1007/s10845-006-0026-0

Dini, G., Failli, F., Lazzerini, B., and Marcelloni, F. (1999). Generation of optimized assembly sequences using genetic algorithms. *CIRP Ann.* 48, 17–20. doi: 10.1016/s0007-8506(07)63122-9

Guo, K., Liu, R., Duan, G., and Liu, J. (2024). A deep reinforcement learning method with multiple starting nodes for dynamic process planning decision making. *Comput. Ind. Eng.* 194:110359. doi: 10.1016/j.cie.2024.110359

Jiang, M., Guo, Y., Huang, S., Pu, J., Zhang, L., and Wang, S. (2024). A novel fine-grained assembly sequence planning method based on knowledge graph and deep reinforcement learning. *J. Manuf. Syst.* 76, 371–384. doi: 10.1016/j.jmsy.2024.08.001

Li, H., Zhang, H., He, Z., Jia, Y., Jiang, B., Huang, X., et al. (2024). Solving integrated process planning and scheduling problem via graph neural network based deep reinforcement learning. arxiv preprint arxiv:2409.00968. doi: 10.48550/arXiv.2409.00968

Mortlock, T., Muthirayan, D., Yu, S. Y., Khargonekar, P. P., and Al Faruque, M. A. (2021). Graph learning for cognitive digital twins in manufacturing systems. *IEEE Trans. Emerg. Top. Comput.* 10, 34–45. doi: 10.1109/TETC.2021.3132251

Mou, W., and Gao, X. (2020). A reliable process planning approach based on fuzzy comprehensive evaluation method incorporating historical machining data. *Proc. Inst. Mech. Eng. Part B J. Eng. Manuf.* 234, 900–909. doi: 10.1177/0954405419889500

Mozaffar, M., Ebrahimi, A., and Cao, J. (2020). Toolpath design for additive manufacturing using deep reinforcement learning. arxiv preprint arxiv:2009.14365. doi: 10.48550/arXiv.2009.14365

Neves, M., and Neto, P. (2022). Deep reinforcement learning applied to an assembly sequence planning problem with user preferences. *Int. J. Adv. Manuf. Technol.* 122, 4235–4245. doi: 10.1007/s00170-022-09877-8

Qian, J., Zhang, Z., Shi, L., and Song, D. (2023). An assembly timing planning method based on knowledge and mixed integer linear programming. *J. Intell. Manuf.* 34, 429–453. doi: 10.1007/s10845-021-01819-7

Rojek, I. (2010). Neural networks as performance improvement models in intelligent CAPP systems. *Control. Cybern.* 39, 54–68. doi: 10.24426/control.and.cybernetics.v39i1.2495

Torres, F., Puente, S. T., and Aracil, R. (2003). Disassembly planning based on precedence relations among assemblies. *Int. J. Adv. Manuf. Technol.* 21, 317–327. doi: 10.1007/s001700300037

Wan, X., Liu, K., Qiu, W., and Kang, Z. (2024). An assembly sequence planning method based on multiple optimal solutions genetic algorithm. *Mathematics* 12:574. doi: 10.3390/math12040574

Wang, R., Wang, G., Sun, J., Deng, F., and Chen, J. (2023). Flexible job shop scheduling via dual attention network-based reinforcement learning. *IEEE Trans. Neural Netw. Learn. Syst.* 35, 3091–3102. doi: 10.1109/TNNLS.2023.3306421

Wang, J., Wu, X., and Fan, X. (2015). A two-stage ant colony optimization approach based on a directed graph for process planning. *Int. J. Adv. Manuf. Technol.* 80, 839–850. doi: 10.1007/s00170-015-7065-7

Wu, Y. J., Cao, Y., and Wang, Q. F. (2019). RETRACTED ARTICLE: assembly sequence planning method based on particle swarm algorithm. *Clust. Comput.* 22, 835–846. doi: 10.1007/s10586-017-1331-4

Zhang, W. (2023). Assembly sequence intelligent planning based on improved particle swarm optimization algorithm. *Manuf. Technol.* 23, 557–563. doi: 10.21062/mft.2023.056

Zhang, J., Chen, C., Su, S., Hu, W., and Zhu, A. (2025). Assembly sequence planning based on hybrid SOS-PSO algorithm. *Int. J. Adv. Manuf. Technol.* 136, 5487–5504. doi: 10.1007/s00170-025-15172-z

Zhang, S. W., Wang, Z., Cheng, D. J., and Fang, X. F. (2022). An intelligent decision-making system for assembly process planning based on machine learning considering the variety of assembly unit and assembly process. *Int. J. Adv. Manuf. Technol.* 121, 805–825. doi: 10.1007/s00170-022-09350-6

Zhu, X., Jha, D. K., Romeres, D., Sun, L., Tomizuka, M., and Cherian, A. (2024). "Multi-level reasoning for robotic assembly: from sequence inference to contact selection" in 2024 IEEE international conference on robotics and automation (ICRA) (IEEE), New York, NY, USA: IEEE 816–823.