Journal of Zhejiang University-SCIENCE A 2026 27(3):200-214
www.jzus.zju.edu.cn; www.springer.com/journal/11582
E-mail: jzus_a@zju.edu.cn

Research Article
https://doi.org/10.1631/jzus.A2500016

Intelligent segment typesetting in shield tunneling based on artificial neural networks and transfer learning

Rui LIU[1], Quanyong ZENG[1], Haibo XIE[2,3], Guoli ZHU[1]✉

[1]*School of Mechanical Science and Engineering, Huazhong University of Science and Technology, Wuhan 430074, China*
[2]*Institute of Advanced Machines, Zhejiang University, Hangzhou 310014, China*
[3]*State Key Laboratory of Fluid Power and Mechatronic Systems, Zhejiang University, Hangzhou 310058, China*

Abstract: Shield tunneling is the method most commonly used for underground projects. Segment typesetting, which involves determining the optimal assembly point for each segment ring and sequentially assembling them into a complete tunnel, is a critical step in shield tunneling. Currently, this typesetting process relies heavily on the operator's experience at construction sites, which does not guarantee quality. Furthermore, research focuses mainly on the commonly used 16-point segment typesetting, largely ignoring other segment types. In addition, the reliability of these studies in site applications remains unsatisfactory. To address these issues, we propose an intelligent method for segment typesetting using an artificial neural network (ANN) and transfer learning. Due to insufficient historical data for ANN training, a dataset creation method was devised based on the Monte Carlo method and manual annotation. An ANN model was then developed to typeset 16-point segments, with its hyperparameters optimized through Bayesian optimization. Subsequently, the trained model was adapted to other segment types via transfer learning, using 10-point segments as a case study. Based on the test set established in this study, our proposed method showed superior performance compared with several commonly used machine learning methods and a representative and well-validated segment typesetting method. It was also validated using real data collected from construction sites, achieving an accuracy of 93.75% for 16-point segments and 91.43% for 10-point segments, both of which significantly surpass the results from manual typesetting on-site. The proposed method achieves accurate, rapid, and intelligent segment typesetting, which is adaptable to various segment types.

Key words: Shield tunneling; Segment typesetting; Assembly points; Artificial neural network (ANN); Bayesian optimization; Transfer learning

1 Introduction

Shield tunneling has emerged as the preferred approach for underground construction in soft soil, due to its rapid, safe, and economical nature, and its minimal impact on the surface (Su et al., 2020; Zhang et al., 2022). During shield tunneling, the shield machine advances forward while simultaneously installing precast concrete segments under the protection of a tail shield (Ye et al., 2024). The most commonly used segment is the universal segment, which is a circular ring with a certain amount of wedge shape (Zhu et al., 2020). Only one type of universal segment ring is required to assemble the entire tunnel. Segment typesetting involves determining the optimal assembly point for each segment ring and sequentially assembling them to form the complete tunnel, which is crucial for tunnel quality (Zhou et al., 2018). Assembly segments with different assembly points can offer various advance amounts in both horizontal and vertical directions, thereby enabling the fitting of curved sections of the design tunnel axis (DTA) and adjusting the relative pose of the shield machine.

At construction sites, segment typesetting still relies on the shield machine operator's discretion, which is based on personal experience. This approach is not only time-intensive, detracting from construction efficiency, but also imposes a considerable workload on the operators (Zhang et al., 2024). Moreover,

✉ Guoli ZHU, glzhu@mail.hust.edu.cn
Rui LIU, https://orcid.org/0000-0002-2691-579X
Guoli ZHU, https://orcid.org/0000-0002-0749-3835

Received Jan. 17, 2025; Revision accepted Sept. 1, 2025; Crosschecked Dec. 28, 2025; Online first Jan. 22, 2026

the high subjectivity and potential for personal error render this method incapable of guaranteeing the quality of segment typesetting. If an unreasonable segment typesetting plan is chosen, serious engineering accidents, such as segment breakage, failure to fit the DTA, and loss of control of the shield machine, may occur. Therefore, accurate, rapid, and intelligent segment typesetting in shield tunneling is an urgent issue that must be addressed.

In response to this requirement, scholars have conducted research on the preliminary typesetting of segments before tunneling. Some studies (Zhou et al., 2018; Zhu et al., 2020; Luo et al., 2022; Liu et al., 2024) have derived pose transfer algorithms for adjacent segment rings, which makes it possible to calculate the pose of the current segment ring at different assembly points. Subsequently, the deviations between the ring and the DTA can be computed, and the assembly point with the least deviation can be selected. This method for selecting assembly points aims to minimize the axis fitting deviation. Preliminary typesetting occurs before tunnel construction and is typically used to evaluate the segments' ability to fit the tunnel. However, field implementation of this method remains challenging, as practical construction involves segment manufacturing tolerances, assembly errors, and segment flotation phenomena. These factors result in significant error propagation when using pose transfer algorithms to calculate segment poses. As a result, it becomes difficult to accurately determine the axis fitting deviations between the segments and the DTA. Moreover, in shield tunneling, segment typesetting significantly affects construction condition factors, such as thrust cylinder stroke differences and shield tail gaps. If these parameters exceed acceptable thresholds, they may lead to serious engineering incidents, including segment breakage and loss of shield machine control. Therefore, these factors must be carefully considered during construction.

Other studies focus on segment typesetting in shield tunneling with the goal of adjusting construction condition factors, which is essentially a multi-objective optimization problem. Zhang et al. (2021) analyzed the impact of selecting different 16-point segment assembly points on shield tail gaps, thrust cylinder stroke differences, and lining trends. They established an objective function by weighted summation of these factors. However, the weights are set by operators in real time based on their experience, which leads to a significant decrease in the automation and feasibility of the method. Although Xie et al. (2023) provided a dynamic relationship between the weights and construction condition factors, it lacked theoretical justification. Furthermore, the axis fitting deviation included in their objective function cannot be accurately determined in practical engineering, resulting in limited applicability on construction sites. Liu et al. (2022) proposed a method with on-site validation. They divided the construction condition factors into 81 intervals and used a genetic algorithm to determine weights for each interval. However, this approach piecewise linearizes the issue of continuous variation in weights with construction condition factors, which could inevitably lead to errors. As a result, the accuracy of typesetting still needs improvement. Note that existing research focuses mainly on 16-point segments, as they are the most commonly used. However, there is a gap in research on other segment types with different numbers of assembly points, such as 10, 12, 14, 19, and 22. Therefore, methods for segment typesetting applicable to various segment types need to be studied, and the accuracy of typesetting must be improved and validated through field applications.

The core of intelligent segment typesetting in shield tunneling is determining the segment assembly points. This is a multi-classification problem, traditionally relying on operator experience. Machine learning, renowned for its powerful mapping capabilities, has been widely used to tackle complex issues that are challenging for traditional approaches (Vinod et al., 2024; Cui et al., 2025). In the realm of machine learning, common multi-classification methods include decision tree (DT) (Guan et al., 2017), *k*-nearest neighbors (KNN) (Hu et al., 2017), support vector machine (SVM) (An et al., 2007; Zhou et al., 2017), and artificial neural network (ANN) (Liu et al., 2025). The ANN stands out due to its robust non-linear modeling capabilities, superior feature extraction abilities, and strong adaptability, which has contributed to its widespread applications across various fields (Liu et al., 2021; Li et al., 2023). As an important part of machine learning, transfer learning is highly valued for its ability to transfer knowledge from one problem domain to another, particularly when labeled data are scarce (Na, 2024; Paldino et al., 2024). Its application is especially useful when segment typesetting focuses on 16-point segments and neglects other segment types.

In this article, we introduce a novel method for intelligent segment typesetting in shield tunneling based on ANN and transfer learning. To address the issues of excessive reliance on human experience and the unsatisfactory field application of existing methods, we developed an ANN-based intelligent typesetting model for 16-point segments. Due to the lack of historical data, we propose a method for generating example datasets based on the Monte Carlo method combined with manual annotation. Transfer learning is then used to adapt the model to various segment types. The proposed method can perform segment typesetting accurately, quickly, and intelligently, and is applicable to various segment types.

The contributions of this study can be summarized as follows:

(1) The impacts of various constraints on segment typesetting are analyzed, including thrust cylinder stroke differences, shield tail gaps, and the previous segment assembly point. An intelligent 16-point segment typesetting model is then developed using ANN, with Bayesian optimization used to fine-tune the hyperparameters.

(2) To address the scarcity of historical field data, the Monte Carlo method simulates the pose of the segment and shield machine. Coordinate transformation and spatial analytical geometry are used to calculate the constraint parameters. These data are then manually annotated to create example datasets.

(3) To address the gap in research on non-16-point segment typesetting, transfer learning is used to adapt the 16-point segment typesetting model to other segment types, with the 10-point segment taken as a case study. This approach fills the gap in research on non-16-point segment typesetting.

(4) The performance of the developed models is validated using both simulated test set data and real-world data from construction sites. The proposed models achieve satisfactory accuracy for both 16-point and 10-point segments, demonstrating their superiority in guiding field construction.

The remainder of this article is structured as follows: Section 2 provides an overview of the basic information about segment typesetting. Section 3 details the generation of datasets for 16-point segment typesetting, the development of the ANN model, and the establishment of a 10-point segment typesetting model using transfer learning. Section 4 demonstrates the validation of this method through simulation data and site application. Sections 5 and 6 present the discussion and conclusions, respectively.

2 Basic information of segment typesetting

2.1 Segment assembly points

Segment typesetting involves assembling each segment ring at a predetermined assembly point to form a complete tunnel. Fig. 1 shows an actual tunnel after the typesetting process is completed. A universal segment ring is generally composed of one capping block (K-block), two adjacent blocks (B-blocks), and several standard blocks (A-blocks). The rings are connected to each other by longitudinal bolts.

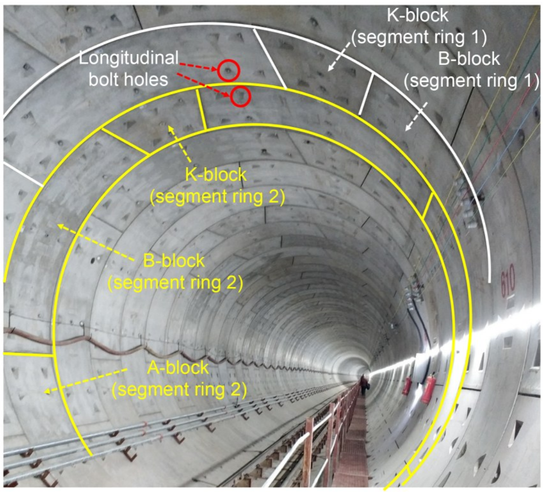

Fig. 1 An actual formed tunnel assembled by segments

Fig. 2 shows the structures of a 16-point and a 10-point universal segment ring, each composed of three A-blocks, two B-blocks, and one K-block. From the front view, the universal segment ring is a circular ring, while from its side view, it is an isosceles trapezoid with a wedge shape. The longitudinal bolt holes are evenly distributed on the side surface of the segment ring. The relative rotation angle between two adjacent rings must be an integer multiple of the unit rotation angle. For a 16-point segment ring, this angle is 22.5°, and for a 10-point segment ring, it is 36°. The K-block is typically the last component to be assembled within a segment ring, representing the most critical phase of assembly. The assembly point of a segment ring is the spatial installation position of the K-block. As illustrated in Fig. 3, for the 16-point segment, when the assembly point is 16, the K-block is located at the top of the ring. At point 4, it is positioned directly to the right, corresponding to a clockwise rotation of 90° (4×22.5°).

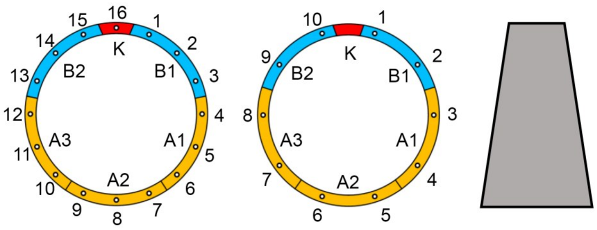

Fig. 2 Structure of the segment ring

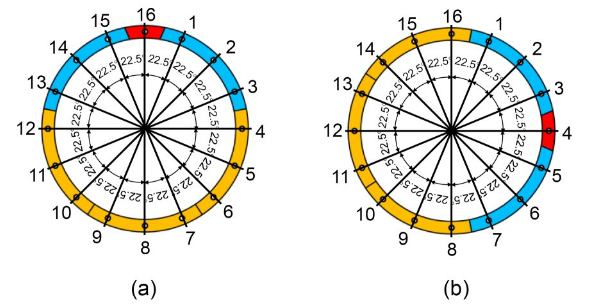

Fig. 3 Assembly point diagram of a 16-point segment: (a) assembly point 16; (b) assembly point 4

2.2 Segment typesetting process

As depicted in Fig. 4, a shield machine typically consists of a front shield and a tail shield, connected by articulation cylinders. The thrust cylinder is mounted on the front shield. During operation, the cylinder extends and presses against the sidewall of the previous segment ring. The reaction force from the segment ring pushes the shield machine forward. At the end of each working stroke, the shield machine stops boring and retracts the thrust cylinder, and a new segment ring with an optimal assembly point is assembled. This cycle repeats until the tunnel is fully constructed.

The connection of the center points of the segment rings forms the formed tunnel axis (FTA). The FTA needs to align as closely as possible with the DTA. To guarantee the quality of the formed tunnel, the method used in engineering is to measure the pose of the shield machine in real time through the guidance system and to ensure that the shield machine advances along the DTA. The goal of assembling segments is to follow the advance of the shield machine, adjusting the relative pose with the shield machine to ensure that the thrust cylinder stroke differences and the shield tail gaps are within an allowable range (Liu et al., 2022).

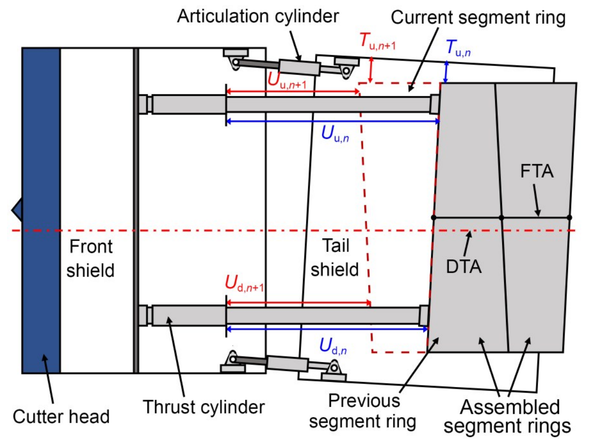

Fig. 4 Schematic diagram of the shield machine and segment ring. $U_{u,n}$ and $U_{d,n}$ are the strokes of the upper and lower thrust cylinders before the assembly of the current segment ring, respectively; $U_{u,n+1}$ and $U_{d,n+1}$ are the strokes of the upper and lower thrust cylinders after the assembly of the current segment ring, respectively; $T_{u,n}$ and $T_{u,n+1}$ are the shield tail gaps on the upper side before and after the assembly, respectively

2.3 Constraints for segment typesetting

2.3.1 Thrust cylinder stroke differences

The thrust cylinders are divided into four groups: upper, lower, left, and right. The differences in stroke lengths of thrust cylinders in the vertical direction (upper minus lower) and the horizontal direction (left minus right) serve as critical metrics for assessing the operating conditions of the shield tunneling. A large thrust cylinder stroke difference can result in excessive radial forces acting on the segments, leading to their breakage or loss of control of the shield machine.

Assembling segments at different assembly points will influence the thrust cylinder stroke difference to variable degrees. For instance, in Fig. 4, by selecting an assembly point closer to the bottom, the vertical thrust cylinder stroke difference is adjusted from $U_{u,n}-U_{d,n}$ to $U_{u,n+1}-U_{d,n+1}$.

2.3.2 Shield tail gaps

Segments are assembled into a ring within the tail shield and are then progressively pushed out as the shield machine advances. The space between the inner surface of the tail shield and the outer surface of the segments is called the shield tail gap. If this gap is too narrow, it may cause compression of the segments by the tail shield, leading to segment breakage and damaging the shield tail seal system (Shen et al., 2022). Therefore, maintaining a sufficient uniform gap at the shield tail is considered a better condition.

The shield tail gaps in the upper, lower, left, and right directions are monitored. The selection of different assembly points has a significant impact on the gap size. For example, as shown in Fig. 4, selecting a lower positioned assembly point increases the upper gap from $T_{u,n}$ to $T_{u,n+1}$.

2.3.3 Principle of staggered-joint assembly

To ensure the structural stability and load-bearing capabilities of the tunnel lining, the stitches formed between adjacent blocks within each segment ring should be staggered from ring to ring. This concept is known as the principle of staggered-joint assembly (Lin et al., 2024), as illustrated in Fig. 5a. Conversely, Fig. 5b represents the result of continuous-joint assembly, a scenario that should be largely avoided in engineering.

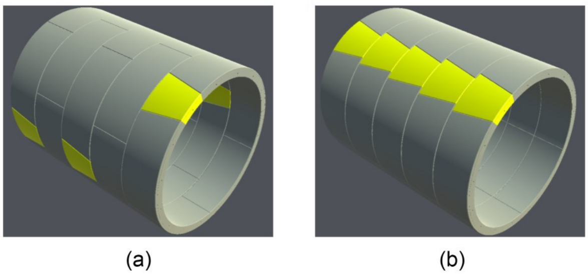

Fig. 5 Schematic diagram of two assembly methods: (a) staggered-joint assembly; (b) continuous-joint assembly

To achieve staggered-joint assembly, construction experience dictates that for 16-point segments, the possible assembly point i_{n+1} of the current segment and the assembly point i_n of the previous segment must satisfy Eq. (1). Similarly, for 10-point segments, the corresponding relationship is defined by Eq. (2).

$$i_{n+1}=\begin{cases} i_n+2+3t, & i_n+2+3t\leqslant 16, \\ i_n+2+3t-16, & i_n+2+3t>16, \end{cases} \tag{1}$$

$$i_{n+1}=\begin{cases} i_n+1+2t, & i_n+1+2t\leqslant 10, \\ i_n+1+2t-10, & i_n+1+2t>10, \end{cases} \tag{2}$$

where $t \in \{1, 2, 3, 4, 5\}$, chosen at random.

For 16-point segments, considering the quality of the tunnel, the topmost and bottommost assembly points (16 and 8) are usually not permitted for selection.

3 Methodology

Segment typesetting in shield tunneling is a highly complex issue that is dependent mainly on empirical knowledge. This process involves considering the thrust cylinder stroke differences in both the vertical ($U_{v,n}$) and horizontal ($U_{h,n}$) directions, the shield tail gaps in the upper ($T_{u,n}$), lower ($T_{d,n}$), left ($T_{l,n}$), and right ($T_{r,n}$) directions, and the assembly point (i_n) of the previous segment, in order to select the optimal assembly point (i^*_{n+1}) of the current segment. We used ANN and transfer learning to resolve this challenge.

Fig. 6 presents the proposed research flowchart. Due to the insufficiency of on-site data, we created a dataset using simulation and manual labeling methods. This dataset was used for the training and evaluation of the ANN model. Considering the widespread use of the 16-point segment, we initially developed an ANN model for 16-point segment typesetting. A Bayesian optimization algorithm was implemented for the tuning of hyperparameters. Building on this initial model, transfer learning was then used to establish typesetting models for other segment types, exemplified by the 10-point segment. After the performance of the models was evaluated through test sets, the models were applied to actual construction sites.

3.1 Dataset

The poses of both the shield machine and the previous segment ring can be determined through experience and construction specifications. Then, constraint parameters such as thrust cylinder stroke differences and shield tail gaps are calculated using spatial analytical geometry and coordinate transformation methods. Ultimately, the dataset is obtained through manual labeling.

3.1.1 Generation of the shield machine and segment ring poses

To facilitate the calculation of constraint parameters, five coordinate systems were established on the front shield, tail shield, and the previous segment ring.

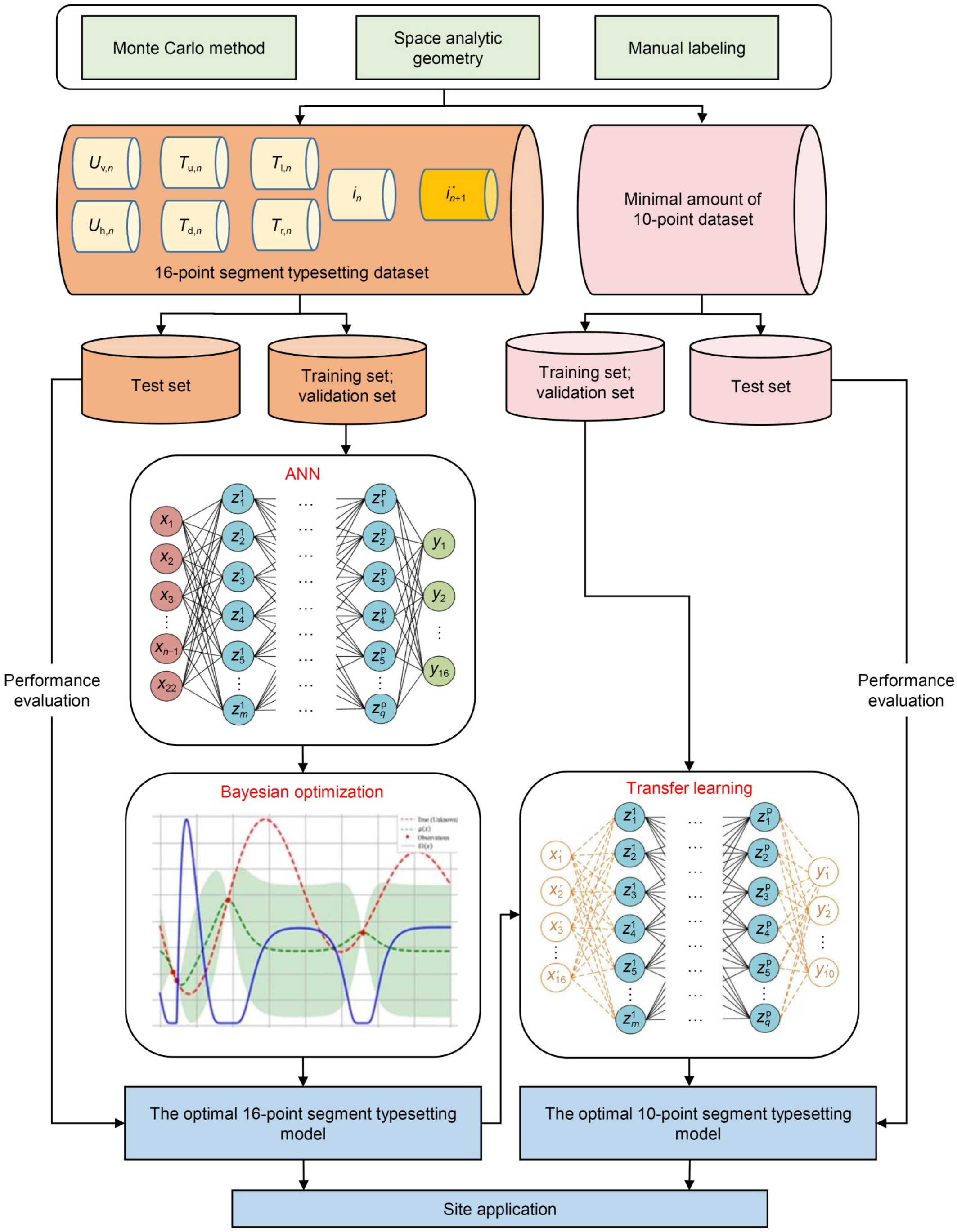

Fig. 6 Flowchart of the proposed method

The configuration of the coordinate systems is shown in Fig. 7. The origin of the DTA coordinate system, O_A-$X_AY_AZ_A$, is located on the DTA, with the *x*-axis coinciding with the DTA and the *yoz* plane perpendicular to the DTA. The front shield coordinate system, O_B-$X_BY_BZ_B$, has its origin at the center of the cutter head, with the *x*-axis aligned with the front shield axis and pointing in the direction the shield machine advances. The *y*-axis and *z*-axis are positioned within the cutter head plane. Similarly, the tail shield coordinate system O_C-$X_CY_CZ_C$ was established with its origin at the center of the front face of the tail shield. The segment coordinate system 1 (O_D-$X_DY_DZ_D$) has its origin at the center of the front face of the previous segment ring, with the y-axis and z-axis located on the front face. By rotating coordinate system O_D-$X_DY_DZ_D$ around the y-axis by an angle θ so that the x-axis aligns with the segment ring axis, the segment coordinate system 2 (O_E-$X_EY_EZ_E$) is derived.

According to construction specifications and engineering experience, the poses of the front shield relative to the DTA (x_B, y_B, z_B, α_B, β_B, γ_B), the tail shield relative to the front shield (x_C, y_C, z_C, α_C, β_C, γ_C), and the previous segment ring relative to the DTA (x_D, y_D, z_D, α_D, β_D, γ_D) are established. Taking a 6 m-class passive articulated shield machine as an example, the range of values for each pose parameter is shown in Table 1. The Monte Carlo method was used to randomly select each pose parameter within its value range, thereby obtaining sets of poses for both the shield machine and the previous segment ring.

3.1.2 Calculation of thrust cylinder stroke differences

There are four groups of thrust cylinders mounted at designated points M_k (k=1, 2, 3, 4) on the front shield (Fig. 7). The positions where the cylinders extend to push against the segment ring are denoted as S_k (k=1, 2, 3, 4). The distance between these two points can be used to calculate the stroke of the cylinders. Based on spatial analytical geometry and coordinate transformation methods (Szczotka and Wojciech, 2008), the thrust cylinder stroke differences in the vertical and horizontal directions are derived. A detailed description is provided in Section S1 of the electronic supplementary materials (ESM).

3.1.3 Calculation of shield tail gaps

The main method for measuring the tail shield gaps involves placing laser displacement sensors at the tail shield to measure the distance between the inner wall of the tail shield and the outer wall of the segment. Four displacement sensors are positioned at H_k (k=1, 2, 3, 4), with the corresponding measurement points denoted as D_k (k=1, 2, 3, 4) (Fig. 7). The distances between H_k and D_k represent the tail shield gaps. The procedure for calculating the lower tail

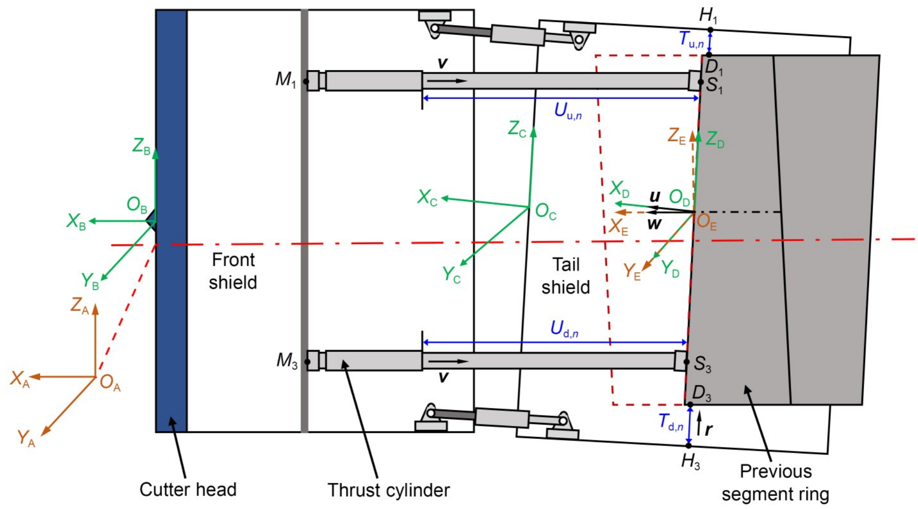

Fig. 7 Schematic diagram of the coordinate system configuration. *u* is the normal vector of the front face of the previous segment ring; *w* is the direction vector of the segment ring's axis; *v* is the direction vector of the thrust cylinder; *r* is the direction vector of the laser emitted by the displacement sensor below

Table 1 Value ranges of the relative poses of the shield machine, segment ring, and DTA

Parameter	Value range	Parameter	Value range	Parameter	Value range
x_B (mm)	0	x_C (mm)	[4680, 4880]	x_D (mm)	[6700, 6900]
y_B (mm)	[–50, 50]	y_C (mm)	[–200, 200]	y_D (mm)	[–100, 100]
z_B (mm)	[–50, 50]	z_C (mm)	[–200, 200]	z_D (mm)	[–100, 100]
α_B (°)	[–3, 3]	α_C (°)	[–3, 3]	α_D	$i_n\times360°/16$, $i_n\in[1, 7]\cup[9, 15]$
β_B (°)	[–3, 3]	β_C (°)	[–3, 3]	β_D (°)	[–3, 3]
γ_B (°)	[–3, 3]	γ_C (°)	[–3, 3]	γ_D (°)	[–3, 3]

shield gap using spatial analytical geometry is provided in detail in Section S2 of the ESM. The methods for calculating the tail shield gaps on the upper, left, and right sides are similar. Accordingly, the four tail shield gaps, $T_{u,n}$, $T_{d,n}$, $T_{l,n}$, and $T_{r,n}$, are computed.

3.1.4 Labeling and data augmentation

The assembly point of the previous segment i_n was determined during the segment ring pose generation, and the unlabeled sample data ($U_{v,n}$, $U_{h,n}$, $T_{u,n}$, $T_{d,n}$, $T_{l,n}$, $T_{r,n}$, i_n) were obtained. Subsequently, several shield machine operators selected the optimal assembly point of the current segment ring, resulting in the data being labeled.

Due to the excessive time and effort required for precise data labeling, a data augmentation method is proposed. Detailed information can be found in Section S3 of the ESM. Ultimately, a total of 11560 sets of example data were collected.

3.2 Establishment of a 16-point segment typesetting model

3.2.1 ANN model

The typical structure of an ANN consists of an input layer, hidden layers, and an output layer, which are interconnected through weights and biases. When input data are propagated forward through the network to the output layer, if the difference between the output result and the label exceeds a predefined threshold, the weights and biases are updated through backward propagation, and then the output result is predicted again. This process is repeated until the ANN can learn the complex mapping relationships between the input and output data. Fig. 8 shows the ANN structure of this application. Inputs ($U_{v,n}$, $U_{h,n}$, $T_{u,n}$, $T_{d,n}$, $T_{l,n}$, $T_{r,n}$, i_n) are fed into the network, and after forward propagation through multiple hidden layers and the output layer, the optimal current assembly point i^*_{n+1} is obtained. The assembly point is represented using one-hot encoding. Therefore, the input layer consists of 22 neurons in total, while the output layer comprises 16 neurons.

3.2.2 Hyperparameter tuning

An excellent ANN model for segment typesetting requires the appropriate selection of hyperparameters. These include the number of network layers and

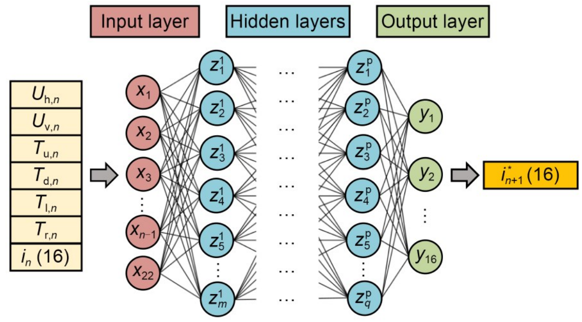

Fig. 8 Structure of the ANN model for a 16-point segment typesetting

neurons, which determine the architecture of the ANN, activation functions, which decide the type of nonlinear transmission function, optimizers, which set the strategy for backpropagation, loss functions, which define the metric for observing the difference between predicted values and label values, and the learning rate, which dictates the degree of change applied to the weights and biases with each update. The choice of these hyperparameters greatly influences the model's predictive performance, and setting reasonable parameters is a complex task.

Common methods for hyperparameter tuning include random search, grid search, and Bayesian optimization. Random search (Tarek et al., 2023) involves random sampling within the hyperparameter space and calculating the value of the objective function. It can search for continuous values and is unlikely to get stuck in local optima, but the downside is that the results can be quite variable and the model's accuracy may be compromised. Grid search (López et al., 2020) systematically goes through all possible combinations of hyperparameters to find the one that yields the best model performance. It can lead to an explosion in computational cost when dealing with numerous parameters, and is limited to discrete hyperparameter values. Bayesian optimization (Victoria and Maragatham, 2021; Li et al., 2022) not only supports the searching of continuous hyperparameter values but also uses prior knowledge during the search, offering higher search efficiency and accuracy, and is becoming the preferred method for hyperparameter optimization.

Bayesian optimization comprises two essential components: the surrogate model and the acquisition function. The surrogate model estimates the probability distribution of the target function corresponding to the hyperparameters and is constantly updated as more sample points are added. The Gaussian process (Yousefpour et al., 2024), which estimates the normal distribution of unknown sampling points within any range of values, is a commonly used surrogate model. The acquisition function in Bayesian optimization balances exploration and exploitation, thereby avoiding local optima. Based on the posterior probability distribution, it identifies the next promising sampling point. The commonly used acquisition function is expected improvement (Yan et al., 2022). This process is reiterated until either the algorithm converges or the predetermined number of iterations is reached. Bayesian optimization effectively explores the hyperparameter space, considerably enhancing the performance of the model. The hyperparameters of the ANN, tuned using Bayesian optimization, and the search space in this model are shown in Table 2.

Table 2 Search space of ANN hyperparameters

Hyperparameter	Description
Number of hidden layers	1–10
Number of neurons of hidden layers	15–40 (step=5)
Activation function in the hidden layers	ReLU, Softmax, tanh
Activation function in the output layer	ReLU, Softmax, tanh
Optimizer	SGD, Adam
Loss function	MSE, categorical cross entropy
Learning rate	0.005, 0.001, 0.0005, 0.0001

ReLU: rectified linear unit; SGD: stochastic gradient descent; MSE: mean squared error

3.3 Transfer learning

Transfer learning recognizes and shares characteristics between the source and target domains. By fine-tuning the source domain model, it can be successfully applied in a new domain, especially where data are scarce or training is costly (Chen et al., 2020). We adopted the transfer learning approach, applying the model for intelligent typesetting of 16-point segments to other types of segments. The 10-point segment serves as an example in our discussion.

The assembly points of the 10-point segment are represented by 10 neurons; therefore, the input layer consists of 16 neurons, and the output layer consists of 10 neurons. The 16-point segment typesetting model serves as the pre-trained model. Both the input and output layers are replaced to create a new model for 10-point segment typesetting (Fig. 9). The black solid lines signify frozen weights, while the brown dashed lines indicate trainable weights. A minimal amount of example data is used for training the new model, thereby implementing intelligent typesetting for the 10-point segments.

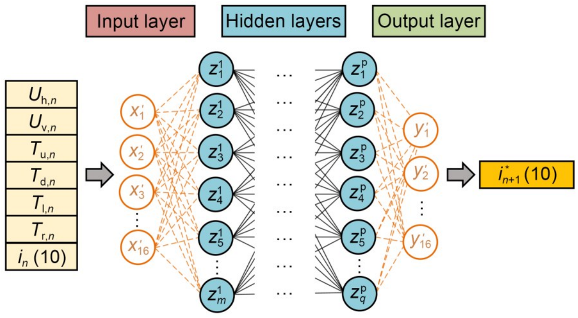

Fig. 9 Schematic diagram for constructing the 10-point segment typesetting model. References to color refer to the online version of this figure

4 Experimental results and analysis

4.1 Training and evaluation of the 16-point segment typesetting model

The 11560 sets of 16-point segment typesetting example data were divided into training, validation, and test sets in a ratio of 7:2:1. The proposed ANN model was developed using Python 3 and the TensorFlow deep learning framework, and the Bayesian optimization algorithm was implemented using the Keras Tuner library. The program ran on a computer with a 6136 CPU, 3.0 GHz frequency, and a 32 GB memory environment.

We used Bayesian optimization to search for the optimal ANN hyperparameters from Table 2, with the goal of minimizing the loss on the validation set. The preset number of iterations for each model training was 2000, with a batch size of 64 per epoch. The model with the smallest validation loss was retained throughout the iteration process. The Bayesian optimization algorithm was executed for a total of 50 iterations, and Fig. 10 shows the process of finding the optimal hyperparameters.

The procedure achieved a minimal loss value at the 24th iteration, after which it entered a state of convergence. After training, the Bayesian optimization algorithm returned the optimal hyperparameter

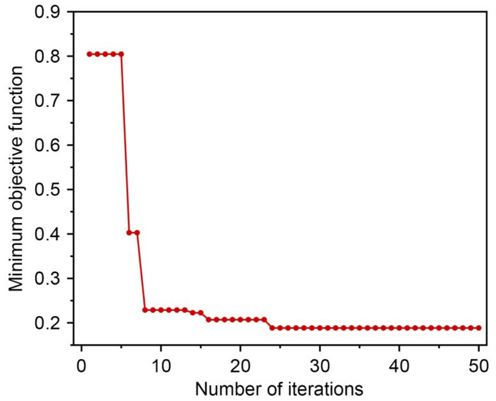

Fig. 10 Iterative convergence of the objective function

combination: a 5-layer neural network architecture with dimensions of 22, 30, 35, 15, and 16, a ReLU activation function for the hidden layers, a Softmax activation function for the output layer, an Adam optimizer for error backpropagation, and categorical cross entropy for the loss function. The learning rate was dynamically adapted during optimization using Adam's adaptive mechanism.

The trained ANN model was used to predict the data of the test set, with the ensuing predicted assembly points being compared to the labels. Based on the comparison results, each set of test examples can be classified into one of four categories: a true positive (TP), when both the prediction and the label are of the same category; a false positive (FP), when the prediction is of a specified category but the label is not; a false negative (FN), when the prediction is not of the specified category but the label is; and a true negative (TN), when neither the prediction nor the label is of the category. The accuracy (A_{cc}), precision (P), recall (R), and F1-score (F_1) were used as evaluation indicators to compare the performance of the classification model. The definitions of each indicator are shown in Eqs. (3)–(6).

$$A_{cc}=\frac{N(\mathrm{TP})+N(\mathrm{TN})}{N(\mathrm{TP})+N(\mathrm{TN})+N(\mathrm{FP})+N(\mathrm{FN})}, \tag{3}$$

$$P=\frac{N(\mathrm{TP})}{N(\mathrm{TP})+N(\mathrm{FP})}, \tag{4}$$

$$R=\frac{N(\mathrm{TP})}{N(\mathrm{TP})+N(\mathrm{FN})}, \tag{5}$$

$$F_1=\frac{2\times P\times R}{P+R}, \tag{6}$$

where N is the number of test examples in the corresponding category.

In evaluating the performance of the ANN model, three widely recognized machine learning methods were used for comparison: KNN, SVM, and DT. Each of these methods represents their respective optimal outcomes following hyperparameter tuning. In addition, the method proposed by Liu et al. (2022), which divides construction condition factors into 81 intervals and optimizes the weights within each interval, was included in the comparison. This method operates without manual intervention and has been validated in practical construction projects.

Table 3 provides a comparative assessment of the five models using the unified test set established in this study. The ANN model outperforms the other models in the task of 16-point segment typesetting in terms of all four evaluation indicators.

Table 3 Comparison of five models for 16-point segment typesetting based on the test set

Model	A_{cc}	P	R	F_1
ANN	0.9472	0.9480	0.9470	0.9473
KNN	0.8746	0.8811	0.8726	0.8740
SVM	0.9317	0.9362	0.9307	0.9319
DT	0.8106	0.8112	0.8098	0.8101
Liu et al. (2022)'s method	0.8919	0.8966	0.8904	0.8917

Fig. 11 presents the confusion matrix of the ANN model's predictions based on the test set, showing a well-balanced data distribution and high predictive accuracy across all categories. No samples were classified as points 8 or 16, indicating that the model has effectively learned to identify non-selectable points. Furthermore, the ANN model has demonstrated significant learning capacity in discerning the principles of staggered-joint assembly. These findings clearly show the powerful processing capabilities of the ANN model for 16-point segment typesetting. Note that since points 8 and 16 are non-selectable, the model effectively performs 14-class classification. However, to distinguish this from the 14-point segments, we maintained the 16-class classification framework for 16-point segment typesetting.

4.2 Transfer learning results

Using the method used for obtaining the 16-point segment typesetting dataset, 2000 examples were collected for the 10-point segments. Given the limited amount of data, the dataset was split into training, validation, and test sets in a 6:2:2 ratio to ensure

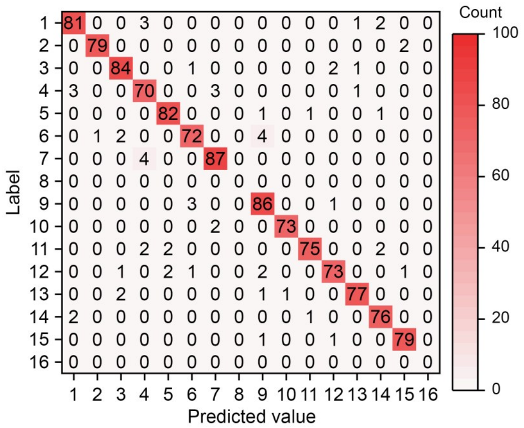

Fig. 11 Confusion matrix of the ANN model for 16-point segment typesetting based on the test set

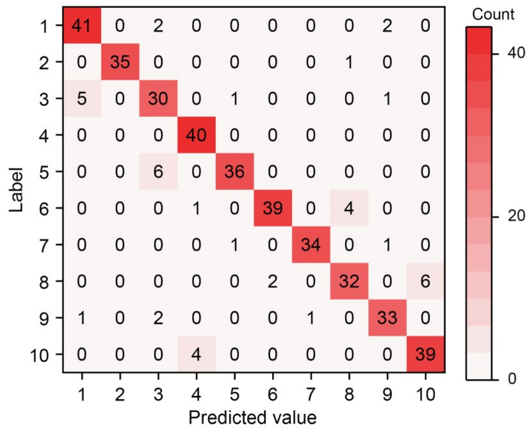

Fig. 12 Confusion matrix of the ANN model for 10-point segment typesetting based on the test set

sufficient test data for statistically valid performance evaluation.

According to the transfer learning method presented earlier, a model was developed with a network architecture with dimensions of 16, 30, 35, 15, and 10. To demonstrate the advantages of transfer learning in addressing issues with limited data, we directly trained models using ANN, KNN, SVM, and DT, in addition to the model established via transfer learning. Each model was optimized for the best performance. Table 4 compares the performance indicators of the five models on the same test set used in this study. The model obtained using the transfer learning approach significantly outperformed the other models in terms of all four metrics. This result indicates that the transfer learning method can effectively address problems similar to those of the source model in scenarios with limited data.

Table 4 Comparison of five models for 10-point segment typesetting on the test set

Model	A_{cc}	P	R	F_1
Transfer learning	0.8975	0.9007	0.8981	0.8988
ANN	0.8350	0.8389	0.8342	0.8348
KNN	0.7275	0.7480	0.7301	0.7321
SVM	0.8375	0.8446	0.8402	0.8392
DT	0.7650	0.7678	0.7649	0.7632

Fig. 12 shows the confusion matrix of the model's classification results based on the test set, demonstrating good classification ability at each assembly point. Using transfer learning methods has significantly reduced the required amount of data. The classification results of the model indicate that effective knowledge transfer from 16-point segment typesetting to 10-point segment typesetting has been successfully achieved.

4.3 Site application

We further validated the performance of the proposed method using real construction site data. Zhengzhou Metro Line 8 in China uses 16-point segments, while Chongqing Metro Line 24 in China uses 10-point segments. We collected historical construction data for 32 sets of 16-point segments from rings 1–21 on the right line and rings 80–90 on the left line. Furthermore, we obtained historical data for 35 sets of 10-point segments corresponding to rings 454–487 on the right line.

Under rushed on-site conditions, the manual segment typesetting process often leads to incomplete consideration, making it difficult to guarantee that the selected assembly point is optimal. For each set of field data, three experienced shield machine operators were tasked with conducting detailed calculations and discussions to identify viable assembly points given sufficient time. When optimization priorities for construction condition factors are comparable, the selection of assembly points may not be unique. Consequently, two alternative assembly points may exist under certain site conditions.

The two intelligent segment typesetting models developed in this study were applied to the corresponding sites. Figs. 13 and 14 compare the operators' verification results with both the historical manual typesetting and the ANN-generated results for 16-point and 10-point segments, respectively. Typesetting accuracy

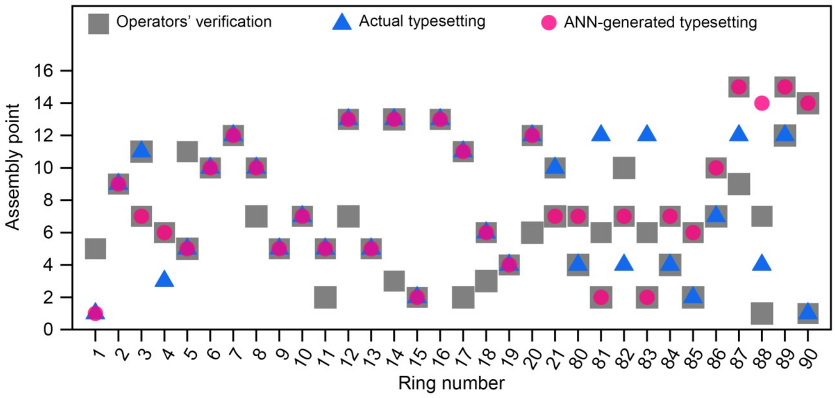

Fig. 13 Comparison of 16-point segment typesetting results

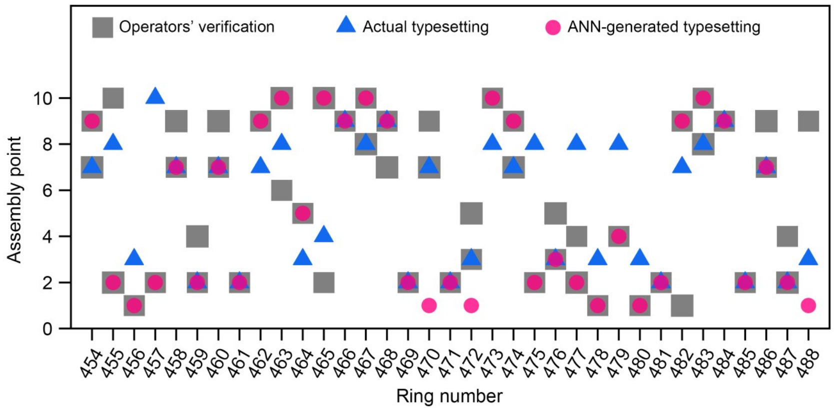

Fig. 14 Comparison of 10-point segment typesetting results

is defined by the overlap with the verification results. For the 16-point segments, only 78.13% of the historical typesetting matched the verification results, while for the 10-point segments, the value was just 57.14%. The discrepancy can be attributed to the rushed considerations of operators due to time constraints during construction and to the operators' habit of selecting assembly points alternately to the left and right when site conditions are relatively good. Representative data can be observed at the 16-point segment from rings 80–84 and at the 10-point segment from rings 475–480, where operators created sections of straight tunnels following this pattern. Although this method is convenient and simple, it does not favor the optimization of construction condition factors. In contrast, the two ANN models developed in this study showed superior accuracy, achieving 93.75% at the 16-point segment site (30/32) and 91.43% at the 10-point segment site (32/35). These models outperformed the accuracy of historical manual typesetting results by 15.62 and 34.29 percentage points, respectively.

The performance of the 16-point segment typesetting model is remarkable, and the results of the 10-point model, developed through transfer learning with limited data, are promising. These outcomes demonstrate the successful applicability of the proposed method across different segment types. Combined with an average selection time of just 30 ms per assembly point, the results indicate that the proposed method is highly accurate, efficient, and intelligent in segment typesetting. It is suitable for various segment types and provides real-time guidance for on-site segment typesetting.

5 Discussion

Segment typesetting currently relies heavily on manual on-site experience, and research focuses mainly on 16-point segments, with the accuracy of typesetting still requiring improvement. To address these issues, a highly accurate, rapid, and intelligent method for shield segment typesetting was developed by combining ANN and transfer learning. First, we analyzed the constraints influencing segment typesetting, namely, the thrust cylinder stroke differences, shield tail gaps, and the previous segment assembly point. Then, we simulated the poses of the shield machine and previous segment ring using the Monte Carlo method. Spatial analytical geometry was used to calculate these constraint parameters, and a manual labeling approach was used to create the example datasets. Next, an intelligent 16-point segment typesetting model was developed using ANN, with the model's hyperparameters optimized through Bayesian optimization. Subsequently, transfer learning was used to develop models for typesetting other segment types, with the 10-point segment used as a case study.

Compared with several commonly used machine learning methods, such as KNN, SVM, and DT, as well as a representative, well-validated segment typesetting method (Liu et al., 2022), the ANN model achieved the best performance for 16-point segment typesetting based on the test set developed in this study, with accuracy, precision, recall, and F1-score of 0.9472, 0.9480, 0.9470, and 0.9473, respectively. Similarly, the 10-point segment typesetting model obtained through transfer learning significantly outperformed the ANN, KNN, SVM, and DT models trained on limited data, achieving accuracy, precision, recall, and F1-score of 0.8975, 0.9007, 0.8981, and 0.8988, respectively. In field applications, the 16-point and 10-point models achieved accuracies of 93.75% and 91.43% based on historical real data, surpassing manual typesetting by 15.62 and 34.29 percentage points, respectively. These results showed that the proposed method not only achieved higher accuracy and had the potential to replace manual typesetting in the field, but also filled the gap in research on non-16-point segment typesetting, making it applicable to various segment types.

Although our method has made satisfactory progress in intelligent segment typesetting, there are still some limitations. First, due to the considerable workload involved in labeling sample data and the difficulty in identifying suitable field sites, we have validated the generalizability of the proposed method only on 16-point and 10-point segment sites. Further validation is needed to confirm its broader applicability. In addition, the method has not yet been tested in complex and specialized engineering scenarios, and its suitability for such conditions requires further verification.

6 Conclusions

This paper presents a highly accurate, rapid, and intelligent method for segment typesetting based on ANN and transfer learning. The proposed method outperformed the widely used machine learning methods (KNN, SVM, and DT) and a representative, well-validated segment typesetting method in accuracy on the same test set. In field applications with real data, the proposed method achieved 93.75% accuracy for 16-point segments and 91.43% accuracy for 10-point segments. The proposed method has shown superior results in segment typesetting and is suitable for various segment types, with the potential to replace manual operations in this task.

In the future, we will focus on improving and validating the generalizability of the proposed method. This includes exploring additional constraints introduced by complex, specialized engineering scenarios and enhancing the rationality of segment typesetting in such projects. The method will also undergo extensive validation in field applications with various segment types.

Acknowledgments

This work is supported by the National Key Research and Development Program of China (No. 2022YFC3802302) and the Project of Institute of Advanced Machines, Zhejiang University (No. KY202404-058). The authors give special thanks to the China Railway Construction Heavy Industry Corporation Limited for providing the application sites.

Author contributions

Rui LIU and Guoli ZHU designed the research. Rui LIU and Quanyong ZENG processed the corresponding data. Rui LIU wrote the first draft of the manuscript. Quanyong ZENG and Guoli ZHU helped to organize the manuscript. Guoli ZHU and Haibo XIE revised and edited the final version.

Conflict of interest

Rui LIU, Quanyong ZENG, Haibo XIE, and Guoli ZHU declare that they have no conflict of interest.

References

An SH, Park UY, Kang KI, et al., 2007. Application of support vector machines in assessing conceptual cost estimates.

Journal of Computing in Civil Engineering, 21(4):259-264. https://doi.org/10.1061/(asce)0887-3801(2007)21:4(259)

Chen YJ, Tong ZM, Zheng Y, et al., 2020. Transfer learning with deep neural networks for model predictive control of HVAC and natural ventilation in smart buildings. *Journal of Cleaner Production*, 254:119866. https://doi.org/10.1016/j.jclepro.2019.119866

Cui J, Zhang DL, Zhu GL, et al., 2025. Spatio-temporal data fusion-based method for working cycles identification of hydraulic excavators. *IEEE Transactions on Instrumentation and Measurement*, 74:2510814. https://doi.org/10.1109/TIM.2025.3545514

Guan XQ, Liang JY, Qian YH, et al., 2017. A multi-view OVA model based on decision tree for multi-classification tasks. *Knowledge-Based Systems*, 138:208-219. https://doi.org/10.1016/j.knosys.2017.10.004

Hu ZZ, Li B, Hu YZ, 2017. Fast sign recognition with weighted hybrid *K*-nearest neighbors based on holistic features from local feature descriptors. *Journal of Computing in Civil Engineering*, 31(5):04017034. https://doi.org/10.1061/(asce)cp.1943-5487.0000673

Li DY, Liu ZD, Xiao P, et al., 2022. Intelligent rockburst prediction model with sample category balance using feedforward neural network and Bayesian optimization. *Underground Space*, 7(5):833-846. https://doi.org/10.1016/j.undsp.2021.12.009

Li JB, Chen ZY, Li X, et al., 2023. Feedback on a shared big dataset for intelligent TBM Part II: application and forward look. *Underground Space*, 11:26-45. https://doi.org/10.1016/j.undsp.2023.01.002

Lin W, Sheil B, Zhang P, et al., 2024. Seg2Tunnel: a hierarchical point cloud dataset and benchmarks for segmentation of segmental tunnel linings. *Tunnelling and Underground Space Technology*, 147:105735. https://doi.org/10.1016/j.tust.2024.105735

Liu FY, Ding WQ, Qiao YF, et al., 2021. Compressive behavior of hybrid steel-polyvinyl alcohol fiber-reinforced concrete containing fly ash and slag powder: experiments and an artificial neural network model. *Journal of Zhejiang University-SCIENCE A*, 22(9):721-735. https://doi.org/10.1631/jzus.A2000379

Liu R, Hu JL, Zhang DL, et al., 2022. Genetic algorithm-based intelligent selection method of universal shield segment assembly points. *Applied Sciences*, 12(14):6926. https://doi.org/10.3390/app12146926

Liu R, Hu JL, Xu Z, et al., 2024. Differential evolution-based universal segment preliminary layout method for shield tunneling. The 30th International Conference on Mechatronics and Machine Vision in Practice (M2VIP), p.1-6. https://doi.org/10.1109/M2VIP62491.2024.10746025

Liu R, Liu YS, Xie Z, et al., 2025. A relative pose measurement method for multi-shield TBM adjacent shields based on multi-sensor integration. *IEEE Transactions on Instrumentation and Measurement*, 74:2534115. https://doi.org/10.1109/TIM.2025.3575988

López D, Alaíz CM, Dorronsoro JR, 2020. Modified grid searches for hyper-parameter optimization. The 15th International Conference on Hybrid Artificial Intelligence Systems, p.221-232. https://doi.org/10.1007/978-3-030-61705-9_19

Luo HB, Li LH, Chen K, 2022. Parametric modeling for detailed typesetting and deviation correction in shield tunneling construction. *Automation in Construction*, 134:104052. https://doi.org/10.1016/j.autcon.2021.104052

Na GS, 2024. One-shot heterogeneous transfer learning from calculated crystal structures to experimentally observed materials. *Computational Materials Science*, 235:112791. https://doi.org/10.1016/j.commatsci.2024.112791

Paldino GM, de Caro F, de Stefani J, et al., 2024. Transfer learning-based methodologies for dynamic thermal rating of transmission lines. *Electric Power Systems Research*, 229:110206. https://doi.org/10.1016/j.epsr.2024.110206

Shen X, Yuan DJ, Cao LQ, et al., 2022. Experimental investigation of the dynamic sealing of shield tail grease under high water pressure. *Tunnelling and Underground Space Technology*, 121:104343. https://doi.org/10.1016/j.tust.2021.104343

Su WL, Li XG, Jin DL, et al., 2020. Analysis and prediction of TBM disc cutter wear when tunneling in hard rock strata: a case study of a metro tunnel excavation in Shenzhen, China. *Wear*, 446-447:203190. https://doi.org/10.1016/j.wear.2020.203190

Szczotka M, Wojciech S, 2008. Application of joint coordinates and homogeneous transformations to modeling of vehicle dynamics. *Nonlinear Dynamics*, 52(4):377-393. https://doi.org/10.1007/s11071-007-9286-2

Tarek Z, Elshewey AM, Shohieb SM, et al., 2023. Soil erosion status prediction using a novel random forest model optimized by random search method. *Sustainability*, 15(9):7114. https://doi.org/10.3390/su15097114

Victoria AH, Maragatham G, 2021. Automatic tuning of hyperparameters using Bayesian optimization. *Evolving Systems*, 12(1):217-223. https://doi.org/10.1007/s12530-020-09345-2

Vinod V, Kleinekathöfer U, Zaspel P, 2024. Optimized multi-fidelity machine learning for quantum chemistry. *Machine Learning: Science and Technology*, 5(1):015054. https://doi.org/10.1088/2632-2153/ad2cef

Xie P, Chen K, Zhu YT, et al., 2023. Dynamic parametric modeling of shield tunnel: a WebGL-based framework for assisting shield segment assembly point selection. *Tunnelling and Underground Space Technology*, 142:105395. https://doi.org/10.1016/j.tust.2023.105395

Yan C, Du H, Kang EZ, et al., 2022. AVEI-BO: an efficient Bayesian optimization using adaptively varied expected improvement. *Structural and Multidisciplinary Optimization*, 65(6):164. https://doi.org/10.1007/s00158-022-03256-3

Ye XW, Zhang XL, Chen YB, et al., 2024. Prediction of maximum upward displacement of shield tunnel linings during construction using particle swarm optimization-random forest algorithm. *Journal of Zhejiang University-SCIENCE A*, 25(1):1-17.

https://doi.org/10.1631/jzus.A2300011
Yousefpour A, Foumani ZZ, Shishehbor M, et al., 2024. GP+: a Python library for kernel-based learning via Gaussian processes. *Advances in Engineering Software*, 195:103686. https://doi.org/10.1016/j.advengsoft.2024.103686
Zhang WC, Zhang J, Yan JR, et al., 2021. Selection of the key segment position for trapezoidal tapered rings and calculation of the range of jack stroke differences with a predetermined key segment position. *Advances in Civil Engineering*, 2021:3606389. https://doi.org/10.1155/2021/3606389
Zhang YK, Gong GF, Yang HY, et al., 2022. Towards autonomous and optimal excavation of shield machine: a deep reinforcement learning-based approach. *Journal of Zhejiang University-SCIENCE A*, 23(6):458-478. https://doi.org/10.1631/jzus.A2100325
Zhang YK, Gong GF, Yang HY, et al., 2024. From tunnel boring machine to tunnel boring robot: perspectives on intelligent shield machine and its smart operation. *Journal of Zhejiang University-SCIENCE A*, 25(5):357-381. https://doi.org/10.1631/jzus.A2300377
Zhou Y, Su WJ, Ding LY, et al., 2017. Predicting safety risks in deep foundation pits in subway infrastructure projects: support vector machine approach. *Journal of Computing in Civil Engineering*, 31(5):04017052. https://doi.org/10.1061/(asce)cp.1943-5487.0000700
Zhou Y, Wang Y, Ding LY, et al., 2018. Utilizing IFC for shield segment assembly in underground tunneling. *Automation in Construction*, 93:178-191. https://doi.org/10.1016/j.autcon.2018.05.016
Zhu X, Bao TF, Yu H, et al., 2020. Utilizing building information modeling and visual programming for segment design and composition. *Journal of Computing in Civil Engineering*, 34(4):04020024. https://doi.org/10.1061/(asce)cp.1943-5487.0000903

Electronic supplementary materials

Sections S1–S3