Sensors and Materials, Vol. 38, No. 11 (2026) 6517–6538
MYU Tokyo

S & M 4693

Analyzing and Optimizing Robotic Arm Performance from Sensor Data via Machine Learning

Jianzi Lv, Qijun Li, Zhizhang Chen, Chen Liu,
Haobo Yang, Fangchen Liu, and Lingling Li*

School of Science, Hebei University of Technology,
No. 5340 Xiping Road, Beichen District, Tianjin 300401, P.R.China

(Received May 11, 2026; accepted August 6, 2026)

Keywords: robotic arm, sensor, K-means clustering, random forest, support vector regression, data-driven framework, feature engineering

To address the challenges of motion trajectory performance prediction and motion pattern recognition for robotic manipulators using multi-channel sensor data, in this paper, we propose a data-driven analytical framework that integrates dynamics simulation, feature engineering, and machine learning techniques, utilizing multi-channel sensor data including joint angles, angular velocities, angular accelerations, and torques. Because of the six-degree-of-freedom robotic manipulator, 500 simulated trajectories are generated, and 34 features are extracted from four dimensions—kinematics, dynamics, geometric constraints, and trajectory quality—with trajectory tracking root mean square error (*RMSE*) and normalized energy consumption index as the prediction targets. Random forest (RF), support vector regression (SVR), and multilayer perceptron (MLP) are employed for performance comparison, and K-means clustering is combined to identify typical motion patterns. The results show that RF exhibits the best overall performance in both prediction tasks, with coefficients of determination of 0.8062 and 0.9959 for *RMSE* prediction and energy consumption prediction, respectively. MLP achieves performance close to RF in energy consumption prediction, while that of SVR is weaker. Noise sensitivity analysis further indicates that RF possesses strong stability under feature perturbation conditions. In this paper, we provide an interpretable data-driven approach for simulation-based prediction and motion pattern analysis of robotic manipulator trajectory performance.

1. Introduction

Trajectory performance is critical for high-precision robotic manipulator operation. However, manipulators exhibit strongly coupled nonlinear characteristics under external disturbances. Traditional model-based methods rely heavily on extensive prior knowledge and complex parameter identification, making it difficult to accurately describe their dynamic features, which leads to significantly increased tracking errors in unstructured tasks. To overcome these limitations, data-driven approaches leverage massive state data to establish a new paradigm for

*Corresponding author: e-mail: lilingling@hebut.edu.cn
https://doi.org/10.18494/SAM6430

ISSN 0914-4935 © MYU K.K.

trajectory optimization.[1,2] Recent advances in sensor technology have enabled the real-time acquisition of motion state data such as joint angles and torques. We employ multiple sensors, including photoelectric encoders, strain-gauge torque sensors, and MEMS accelerometers at a sampling rate of 100 Hz. The collected data contain kinematic and dynamic features, laying a foundation for machine learning.[3] Nevertheless, systematically extracting valid features from high-dimensional noisy signals and constructing models balancing prediction accuracy and interpretability remain core challenges.

In this paper, we propose a data-driven analytical framework integrating simulation, feature engineering, and machine learning. A dynamic simulation system batch-generates trajectory data encompassing diverse kinematic parameters, geometric constraints, and performance metrics.[4] Subsequently, 34-dimensional trajectory features are extracted on the basis of the principles of kinematics, dynamics, and geometric constraints. Finally, the root mean square error (*RMSE*) for trajectory tracking and the normalized energy consumption index are defined as performance metrics, and machine learning methods are employed to uncover data patterns. Regression analysis establishes the mapping relationship between features and metrics, enabling the prediction of energy consumption and control accuracy for new trajectories. To address typical motion pattern recognition, we utilize unsupervised clustering analysis for the automatic identification of typical motion patterns.

The core contributions include the following. First, a four-dimensional feature engineering framework is proposed, which systematically maps manipulator trajectory performance to a 34-dimensional computable feature space—kinematics (18D), dynamics (6D), geometric constraints (6D), and trajectory quality (4D)—providing a standardized paradigm for sensor-data-driven trajectory analysis. Compared with previous studies relying on single-type features or heuristic feature selection, the proposed framework achieves systematic coverage of multidimensional features and quantitative interpretability for the first time. Second, random forest (RF), support vector regression (SVR), and multilayer perceptron (MLP) are adopted to build prediction models for *RMSE* and a normalized energy consumption index,[5] and key factors affecting trajectory performance are explored through feature importance analysis. Finally, the optimal $K = 4$ is determined via the elbow method with a silhouette coefficient, four typical motion patterns are automatically identified using K-means clustering, and framework robustness is verified through noise sensitivity analysis.[6,7]

2. Simulation Data Generation and Feature Modeling

The robotic arm model selected for simulation in this paper is a six-degree-of-freedom (6-DOF) serial robotic arm, whose structure is shown in Fig. 1. This 6-DOF robotic arm contains six rotational joints in sequence from the base to the end-effector. To facilitate the design and verification of control algorithms, independent angle, velocity, and torque sensors are installed around each joint.[5] Following typical industrial robotic arm configurations, the link lengths are 0.5, 0.4, 0.35, 0.2, 0.15, and 0.1 m, respectively. The link masses are 5.0, 4.0, 3.5, 2.0, 1.5, and 1.0 kg, respectively. The maximum allowable torque for each joint is 100, 80, 60, 30, 20, and 10 Nm.[6] Each joint integrates three industrial-grade sensors: a photoelectric encoder, a

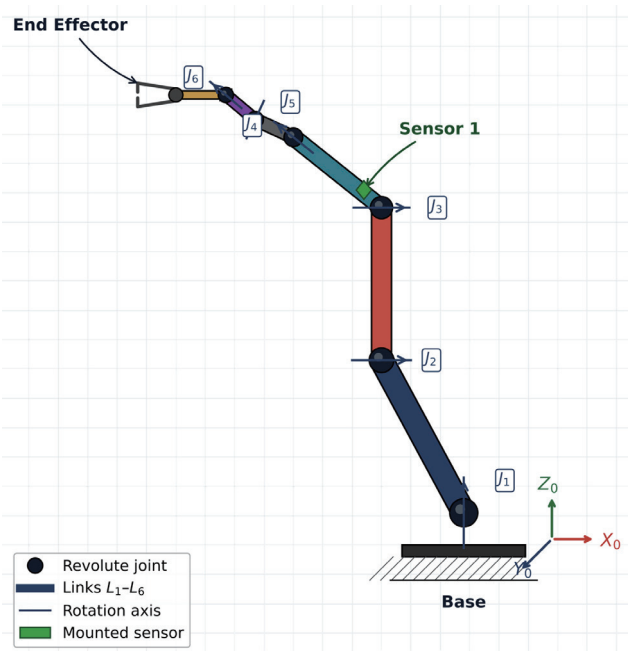


Fig. 1. (Color online) Schematic diagram of a six-degree-of-freedom robotic arm structure.

piezoelectric angular velocity sensor, and a strain-gauge torque sensor. Synchronous sampling runs at 100 Hz via 16-bit A/D conversion. The simulation platform is developed on MATLAB R2020b and Robotics Toolbox. The 6-DOF manipulator has an RRPRRR joint configuration, in which R and P denote revolute and prismatic joints, respectively; specifically, the first, second, fourth, fifth, and sixth joints are revolute, whereas the third joint is prismatic with a 0.7 m stroke. The manipulator is modeled using modified Denavit–Hartenberg parameters. Forward kinematics and joint torque calculation rely on the toolbox's SerialLink and recursive Newton–Euler function rne(); inverse kinematics is solved analytically using Pieper's criterion. Continuous trajectories are optimized by Minimum Snap quadratic programming with piecewise septic polynomials to ensure continuous position, velocity, acceleration, and jerk. The platform supports Monte Carlo workspace analysis (up to 100000 samples) and an interactive teach pendant. All computations use double-precision floating points with a fixed 0.01 s simulation time step.

2.1 Robotic arm simulation platform and trajectory generation method

Trajectory generation employs a quintic polynomial interpolation method. For each trajectory, initial and terminal joint angles are randomly sampled in joint space, with all joint angles uniformly distributed within their respective ranges. The time series of joint angles is generated using the following interpolation formula.[7]

$$\boldsymbol{q}(t) = \boldsymbol{a}_0 + \boldsymbol{a}_1 t + \boldsymbol{a}_2 t^2 + \boldsymbol{a}_3 t^3 + \boldsymbol{a}_4 t^4 + \boldsymbol{a}_5 t^5 \tag{1}$$

The coefficients in the above equation are jointly determined by boundary conditions such as the positions, velocities, and acceleration constraints of the specified start and end points. We set the trajectory duration to be randomly selected between 1 and 5 s, with a time step of 0.01 s. The end-effector Cartesian coordinates are obtained from joint angles via forward kinematics; Newton–Euler recursive algorithms are employed to compute the driving torques for each joint. At each time step, six joint angles, angular velocities, angular accelerations, and torques are recorded, yielding 24-dimensional sensor data. Figure 2 illustrates the end-effector's 3D trajectory, joint angle variations, and velocity curves, reflecting the motion patterns of different trajectories.

Simulation data captures multidimensional parameters including joint angles, angular velocities, torques, and end-effector positions. Figure 3 reveals significant temporal discrepancies among sensors under different trajectories. By extracting 34-dimensional features, a correlation heatmap (Fig. 4) was generated, where the color intensity indicates the linear correlation strength. The heatmap reveals that the angle, velocity, and acceleration of the same joint exhibit a certain degree of correlation, while the correlation between different joints is relatively weak. This indicates good feature independence, which is beneficial for subsequent machine learning modeling.

2.2 Simulation dataset construction and processing workflow

The simulation dataset employs Latin hypercube sampling, uniformly sampling across an 18-dimensional parameter space encompassing joint angle changes, maximum velocity, and maximum acceleration.[8] Each dimension is divided into 50 equally probable intervals, generating a total of 500 independent trajectories.[9] Data preprocessing included cleaning, outlier detection, and Z-score normalization. The cleaning phase removed invalid samples exceeding physical limits, while normalization standardized all features by the Z-score method. The final dataset comprises 500 samples, each containing 34 input features and two target variables. Input features include 18 kinematic features, six dynamic features, five geometric

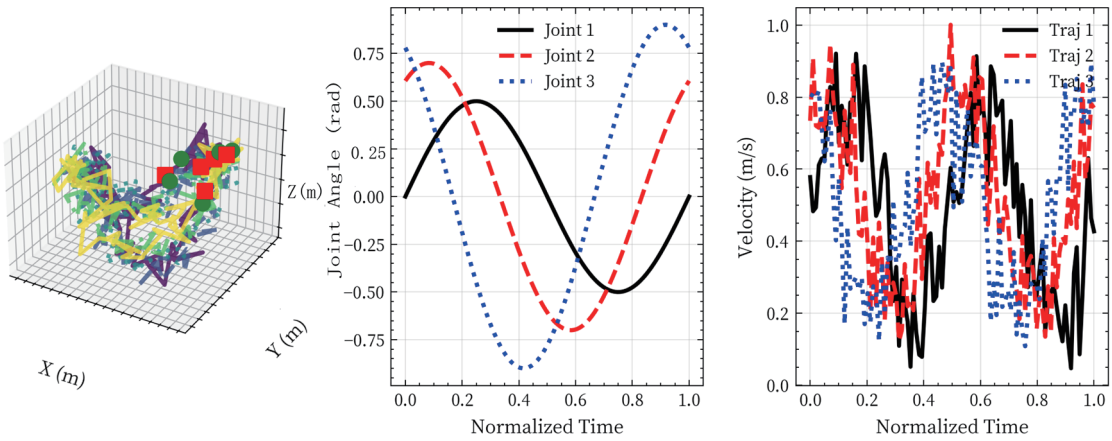


Fig. 2. (Color online) Robotic arm trajectory motion simulation results

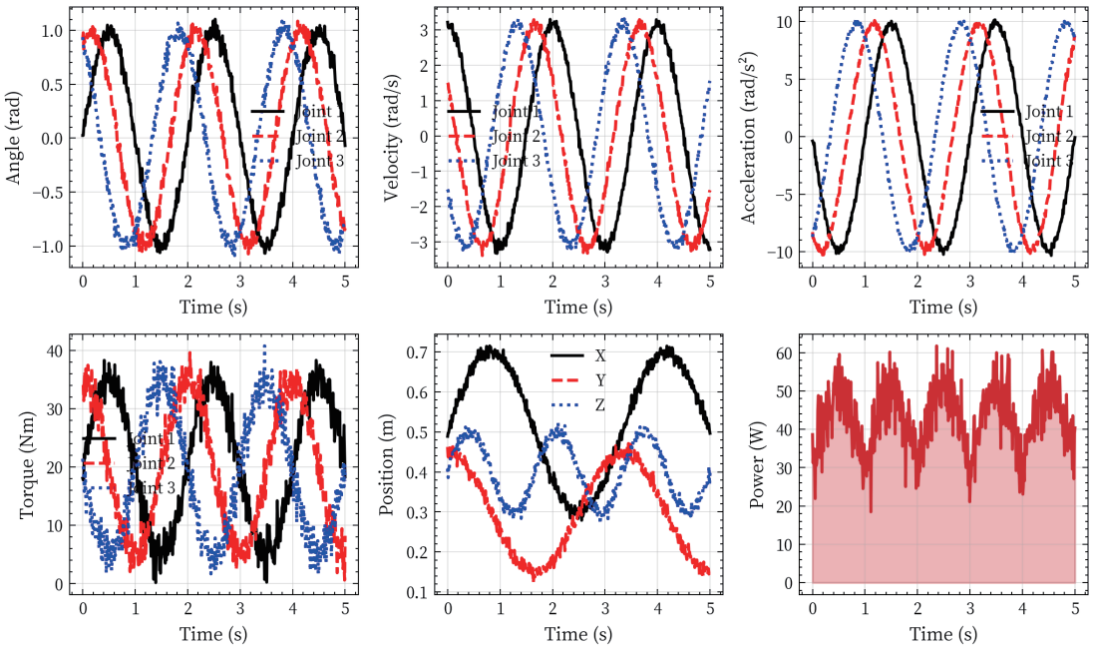


Fig. 3. (Color online) Sensor data sequence simulation results

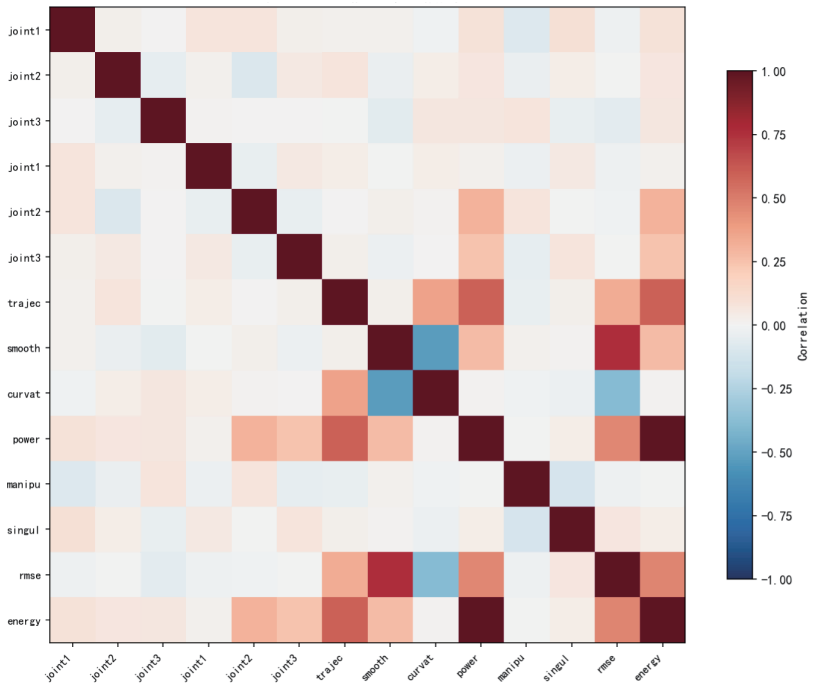
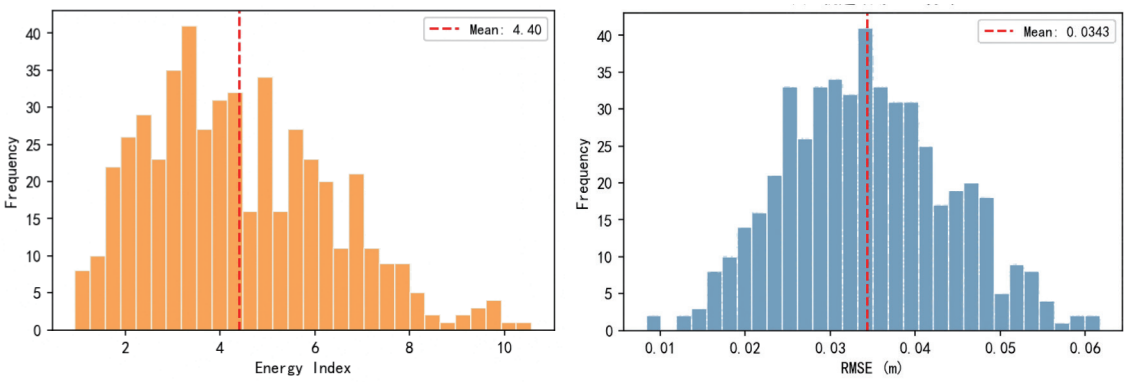


Fig. 4. (Color online) Simulation data feature correlation heatmap.

features, and five trajectory features. The dataset is randomly split into 80% training set (400 samples) and 20% test set (100 samples). Figure 5 shows that the *RMSE* follows an approximately normal distribution concentrated between 0.02 and 0.05 m. The energy consumption index



(a) (b)

Fig. 5. (Color online) (a) Trajectory tracking *RMSE* distribution plot and (b) energy consumption index distribution chart.

exhibits a wide distribution ranging from 1 to 10, indicating that the simulation data encompasses trajectory types with varying energy consumption levels.

It should be noted that all data in this study were generated by Newton–Euler dynamics simulations and do not account for factors such as sensor noise (e.g., encoder quantization error and torque drift), communication delays, nonlinear friction, and external disturbances found in real robotic arm systems. Therefore, the performance metrics of the resulting models are valid only under simulation conditions, and their generalization capabilities in actual deployment must be verified on real platforms. To partially mitigate this limitation, noise sensitivity analysis is performed subsequently: Gaussian noise of varying intensities is added to simulation features to systematically evaluate model performance degradation under simulated sensor noise, thereby verifying framework robustness. The aim of this study is to establish a data-driven analytical framework and validate its feasibility, rather than to provide a control model that is ready for direct deployment.

2.3 Trajectory feature extraction and performance metric definition

We systematically extract 34-dimensional features across four dimensions: kinematics, dynamics, geometric constraints, and trajectory quality. Kinematic features comprise 18 dimensions characterizing the robotic arm's motion state. Dynamic features include six dimensions representing force conditions. Geometric features encompass six dimensions describing the end-effector's spatial position and motion capabilities. Trajectory features consist of four dimensions for directly evaluating motion quality. The detailed feature statistics of the simulation dataset are summarized in Table 1. Trajectory length is defined as the total path length traversed by the end-effector.

$$L = \int_{t_0}^{t_f} \left\| \dot{\boldsymbol{p}}(t) \right\| dt \tag{2}$$

In this context, t_0 and t_f represent the start and end times of the motion, respectively.

Table 1
Feature statistics of the simulation dataset.

Feature category	Feature name	Dimension	Unit	Value range
Kinematics characteristic	Value range	6	rad	[−π, π]
Kinematics characteristic	Joint velocity	6	rad/s	[−3, 3]
Kinematics characteristic	Joint acceleration	6	rad/s^2	[−10, 10]
Dynamic characteristic	Joint Torque	6	Nm	[−100, 100]
Geometric characteristic	End position	3	m	[0, 1.5]
Geometric characteristic	Operability	1	—	[0, 0.15]
Geometric characteristic	Range of motion	1	rad	[1.8, 5.3]
Geometric characteristic	Singularity Proximity	1	—	[0, 0.2]
Trajectory characteristic	Singularity Proximity	1	m	[0.4, 0.9]
Trajectory characteristic	Smoothness	1	—	[1.8, 3.5]
Trajectory characteristic	curvature	1	1/m	[0.03, 0.4]
Trajectory characteristic	Power consumption	1	W	[12, 92]
Target variable	RMSE	1	m	[0.008, 0.062]
Target variable	Energy Consumption Index	1	—	[0.94, 10.58]

Trajectory smoothness is measured by the root mean square value of the acceleration along the trajectory, as shown in the following formula.

$$S = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\frac{d^3\boldsymbol{p}}{dt^3}\right\|^2} \tag{3}$$

Here, S represents the trajectory smoothness index, and N represents the total number of sampling points along the trajectory; the smaller the value of S, the smoother the trajectory.

By integrating the product of torque and angular velocity at each joint over time and then dividing it by the duration of the motion trajectory, we obtain the following expression for power consumption.

$$P = \frac{1}{t_f - t_0}\int_{t_0}^{t_f}\sum_{i=1}^{6}\left|\tau_i \dot{q}_i\right| dt \tag{4}$$

Here, τ_i and q_i represent the drive torque and angular velocity of the ith joint, respectively.

We selected *RMSE* in trajectory tracking and the normalized energy consumption index as two objective variables, with *RMSE* used to measure the positioning accuracy of trajectory tracking.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left\|\boldsymbol{p}_d\left(t_i\right) - \boldsymbol{p}_a\left(t_i\right)\right\|^2} \tag{5}$$

Here, $p_d(t_i)$ and $p_a(t_i)$ represent the target position and actual position at time step i, respectively.

The normalized energy consumption index is defined as the ratio of actual power consumption to baseline power consumption, where baseline power consumption refers to the power consumption of the robotic arm during no-load, low-speed motion.

$$E_{norm} = \frac{P_{actual}}{P_{base}} \tag{6}$$

Here, the baseline power P_{base} is the sum of the product of average torque and average angular velocity across all joints under no-load, constant-velocity motion.

$$P_{base} = \sum_{i=1}^{6} \left|\overline{\tau}_i\right| \cdot \left|\overline{\dot{q}}_i\right| \tag{7}$$

$\overline{\tau}_i$ and $\overline{\dot{q}}_i$ denote the average torque and average angular velocity of the ith joint over the entire motion cycle, respectively. Adopting P_{base} eliminates the bias caused by inherent differences among devices, enabling the standardized comparison of energy consumption.

3. Regression-based Trajectory Performance Prediction Model

In this section, we build RF, SVR and MLP prediction models for trajectory tracking *RMSE* and normalized energy index using the 34-dimensional trajectory features extracted in Sect. 2. Multimetric assessment, feature importance analysis, statistical tests, and noise sensitivity analysis are conducted to systematically compare model accuracy and robustness. The core innovation is a full pipeline for regression modeling and reliability verification, which offers a repeatable method for quantitative manipulator trajectory performance prediction.

3.1 Regression modeling problem description and evaluation metrics

Trajectory performance prediction is defined as a supervised learning regression problem. The input comprises a 34-dimensional feature vector representing the kinematic, geometric, and safety attributes of the trajectory. The output consists of two core metrics: the *RMSE* of trajectory tracking and the normalized energy consumption index. By establishing a mapping function from the feature space to the performance space, the model enables the forward-looking prediction of execution effectiveness on the basis of trajectory characteristics. To evaluate the model's prediction accuracy and generalization capability, we employ three evaluation metrics. The *RMSE* of trajectory tracking directly measures the absolute positioning accuracy of the robotic end-effector and is calculated as $RMSE_{model} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i\right)^2}$, where $\boldsymbol{y_i}$ is the true value, $\hat{\boldsymbol{y}}_{\boldsymbol{i}}$ is the predicted value, and is the number of samples. The mean absolute error (*MAE*) demonstrates greater robustness against outliers and is calculated as $MAE = \frac{1}{n}\sum_{i=1}^{n}\left|\boldsymbol{y}_i - \hat{\boldsymbol{y}}_i\right|$. The coefficient of determination (R^2) characterizes the proportion of variance in the target variable

explained by the model; it is expressed as $R^2 = 1 - \frac{\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2}{\sum_{i=1}^{n}\left(y_i - \bar{y}\right)^2}$, where $\bar{y}$ represents the mean of the true values. All experiments in the above formulas employed five-fold cross-validation to ensure the stability of evaluation results.

3.2 SVR model construction

SVR projects features into a high-dimensional space via nonlinear mapping to construct the maximum margin linear regression hyperplane, making it suitable for small-sample nonlinear problems. By introducing an ε-insensitive loss function that imposes no penalty on pipeline errors, it enhances model sparsity and robustness. The optimization objective for SVR is

$$\min_{w,b,\xi,\xi^*} \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{n}\left(\xi_i + \xi_i^*\right). \tag{8}$$

The constraints are as follows.

$$\begin{aligned} y_i - f\left(x_i\right) &\le \varepsilon + \xi_i \\ f\left(x_i\right) - y_i &\le \varepsilon + \xi_i^* \\ \xi, \xi_i^* &\ge 0 \end{aligned} \tag{9}$$

Here, w and b denote the weight vector and bias term of the regression hyperplane, respectively; C is the regularization parameter used to balance model complexity and training error; ε defines the bandwidth of the insensitive loss function; ξ_i and ξ_i^* are slack variables. In this study, radial basis functions (RBF) are used as the kernel function.

$$K\left(x_i, x_j\right) = \exp\left(-\gamma\left\|x_i - x_j\right\|^2\right) \tag{10}$$

The kernel parameter controls the width of the RBF function, with key hyperparameters optimized through grid search combined with five-fold cross-validation.

3.3 Building an RF regression model

RF regression enhances model accuracy and generalization capabilities through the construction of multiple decision trees and aggregating prediction results, while inherently possessing the ability to assess feature importance.[10] RF employs two randomization mechanisms—bootstrap sampling and random feature selection—to effectively reduce model variance. During training, bootstrap sampling generates multiple distinct subsets from the original training dataset, with each subset growing a regression decision tree. For new input feature vectors, the final prediction is the average of all tree predictions.

$$\hat{y} = \frac{1}{m}\sum_{t=1}^{m} h_t(\boldsymbol{x}) \tag{11}$$

Here, m denotes the total number of decision trees in the RF, and $h_t(\boldsymbol{x})$ denotes the prediction made by the tth decision tree for the input feature vector $\boldsymbol{x}$.

RF employs the average impurity reduction method to assess feature importance, quantifying each motion feature's contribution to performance metrics.[11,12] For the ith feature, its importance is defined as the weighted average of impurity reductions attributable to that feature across all tree nodes.

$$Importance_j = \frac{1}{m}\sum_{t=1}^{m}\sum_{node\in t} \Delta I(node)\cdot L\left(feature_j\ used\ at\ node\right) \tag{12}$$

Here, $\Delta I(node)$ denotes the reduction in impurity before and after node splitting, and $L(\cdot)$ represents the indicator function. In this study, the number of trees was set to 100, and the maximum number of features was set to 11. Optimal parameters were determined through grid search and cross-validation.

3.4 Construction of MPL model

To further validate the proposed framework, MLP is introduced as a deep learning baseline alongside RF and SVR. The MLP adopts a three-layer fully connected structure (128–64–32) with the rectified linear unit activation function activation, Adam optimizer with an initial learning rate of 0.001, L2 regularization coefficient $\alpha = 0.01$, and batch size of 32. Early stopping with patience = 50 and a 10% validation split are applied to prevent overfitting; training terminates if validation loss does not improve for 50 consecutive epochs. Both input features and target variables are Z-score-normalized before training.

For an input feature vector $\boldsymbol{x}$, the representation of the lth hidden layer can be written as

$$h^{(l)} = \sigma\left(W^{(l)}h^{(l-1)} + b^{(l)}\right), h^{(0)} = x, \tag{13}$$

where $W^{(l)}$ and $b^{(l)}$ denote the weight matrix and bias vector of the lth layer, respectively, and $\sigma(\cdot)$ stands for the ReLU activation function. The output layer adopts linear mapping to obtain the predicted value $\hat{y}_l$. The model is trained by minimizing the sum of MSE and the L2 regularization term.

$$L = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{y}_i\right)^2 + \alpha\sum_{l=1}^{L}\left\|W^{(l)}\right\|_F^2 \tag{14}$$

Here, n denotes the number of training samples, y_i and $\hat{y}_i$ represent ground-truth values and predicted values, respectively, and ($\alpha = 0.01$) is the L2 regularization coefficient. The Adam optimizer is adopted for parameter updates, with validation loss as the early-stopping criterion.

The MLP consists of three hidden layers with 128, 64, and 32 neurons sequentially. ReLU activation is applied to all hidden layers, while linear activation is used for the output layer. The initial learning rate is set to 0.001, batch size to 32, and maximum epochs to 2000. Ten percent of the training set is set aside as the validation set; training terminates early if the validation loss fails to improve over 50 consecutive epochs.

3.5 Comparative analysis of regression model predictive performance

We selected 500 samples, with 400 allocated to the training set and 100 to the test set for the RF model. Figure 6 presents the scatter plots of the three models for *RMSE* and energy consumption index prediction tasks. The comparison results indicate that the prediction points of RF cluster tightly around the diagonal line with negligible overall deviation. MLP achieves performance close to RF in energy consumption prediction yet exhibits slightly higher dispersion for *RMSE* prediction, while SVR yields the largest dispersion and systematic bias. Figure 7 presents the comparison between the predicted and actual energy consumption index values for the three regression models. The overall prediction performance of the three models is summarized in Table 2.

All metrics reported in Table 2 were derived from simulation data; real-world performance may degrade owing to sensor noise and environmental disturbance. We adopt the fitting degree between predicted and ground-truth energy indices as the evaluation standard. RF performs best across both tasks: it reaches R^2 =0.8062 for *RMSE* prediction (surpassing MLP and SVR) and R^2 = 0.9959 for energy forecasting (comparable to MLP; both superior to SVR). MLP's favorable energy prediction results confirm its ability to capture nonlinear links between trajectory features and energy use.

RF ranks top in prediction accuracy and cross-task stability, rendering it ideal for manipulator trajectory prediction with aggregated sensor features. MLP is a feasible alternative for energy forecasting. SVR underperforms on both tasks, mainly because its RBF kernel is highly parameter-sensitive in high-dimensional feature spaces.

As shown in Fig. 8, smoothness (0.6565) is the most critical factor affecting tracking accuracy, while trajectory length (0.0910) and curvature (0.0498) are less significant. In Fig. 8,

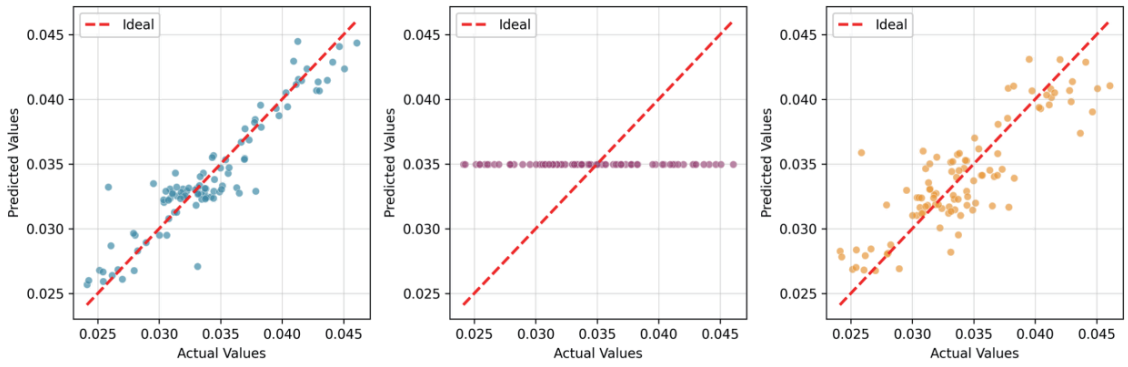


Fig. 6. (Color online) *RMSE* predicted values vs actual values.

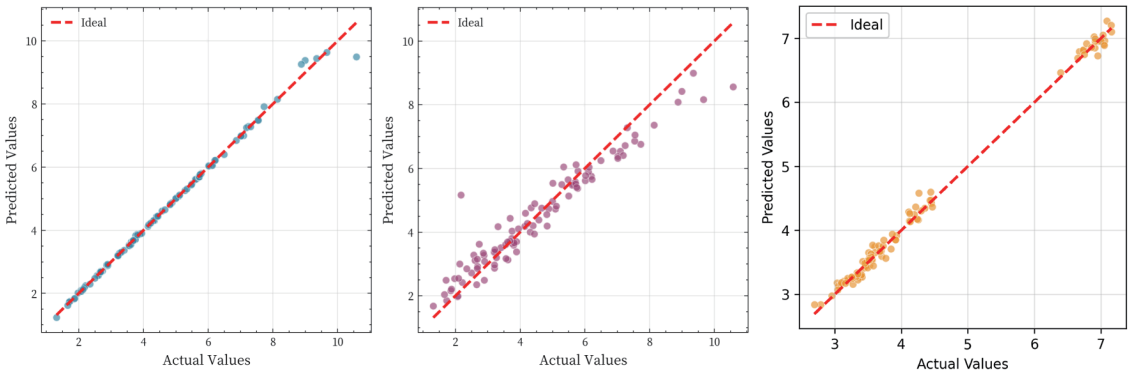


Fig. 7. (Color online) Predicted and actual energy consumption index values.

Table 2
Regression model performance.

Energy consumption index	Model	*RMSE*	*MAE*	R^2	95% *CI* (R^2)
Trajectory tracking root mean square error	RF	0.0042	0.0031	0.8062	[0.784,0.916]
Trajectory tracking root mean square error	SVR	0.0096	0.0079	−0.0091	—
Trajectory tracking root mean square error	MLP	0.00263	0.00206	0.7265	—
Energy consumption index	RF	0.1283	0.0465	0.9959	[0.976,0.989]
Energy consumption index	SVR	0.5712	0.4072	0.9193	—
Energy consumption index	MLP	0.09294	0.07149	0.9957	—

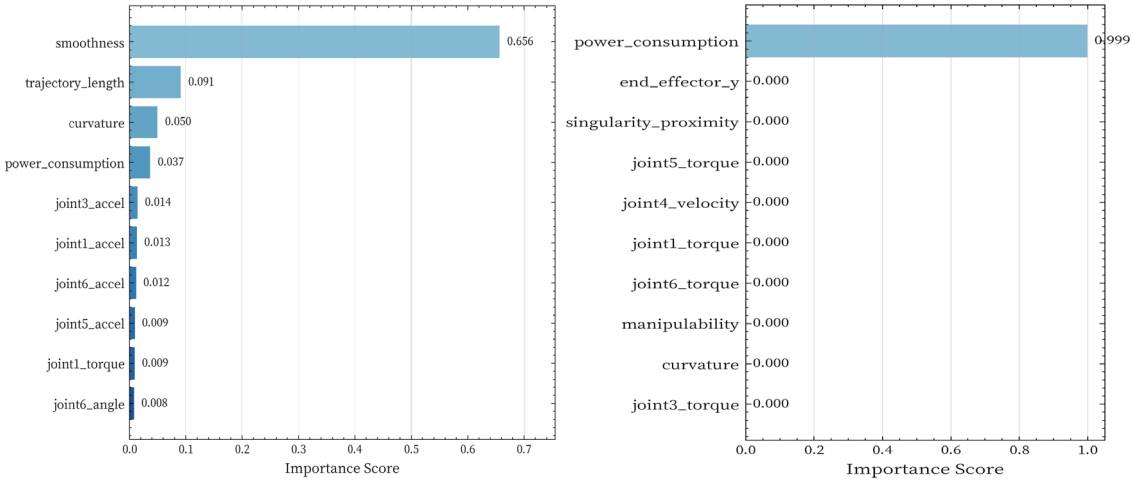


Fig. 8. (Color online) (a) *RMSE* feature importance analysis for prediction. (b) Feature importance analysis chart for energy consumption index prediction.

the importance of power consumption reaches 0.9987, far exceeding the contributions of other features.

As shown in Fig. 9, RF attains the highest R^2 on both tasks, demonstrating strong explanatory power for target variable variation. MLP achieves low *RMSE* and *MAE*, and fits energy

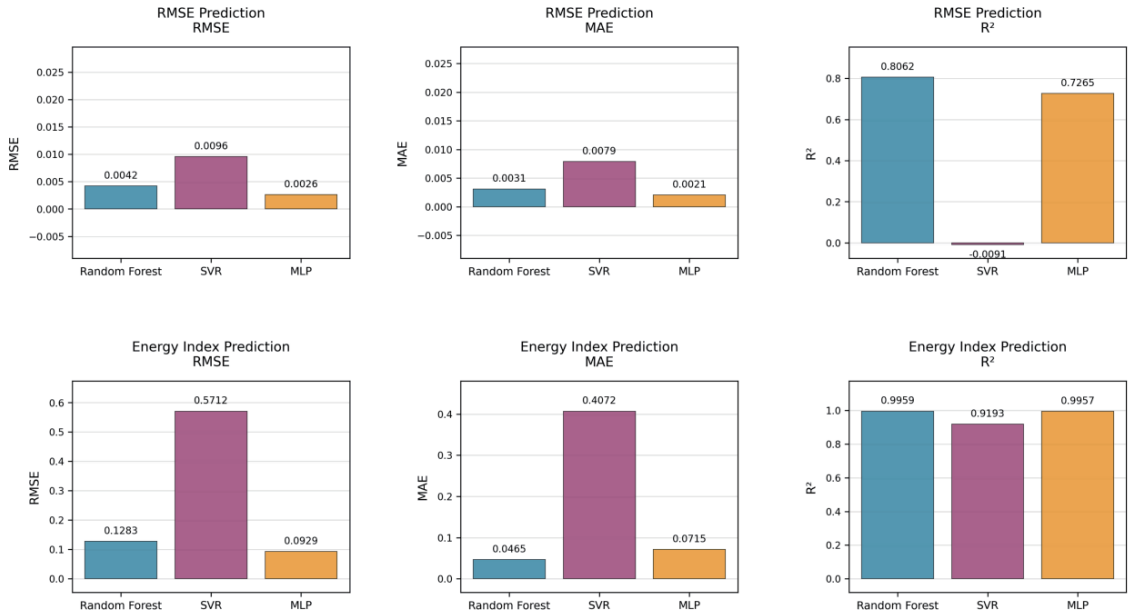


Fig. 9. (Color online) Performance metrics comparison chartd for three models.

consumption well, proving its ability to capture nonlinear links between trajectory features and energy use. In contrast, SVR delivers weaker prediction performance, with negative R^2 for *RMSE* forecasting. RF boosts adaptability and stability for complex nonlinear relations via bootstrap sampling and random feature selection. MLP captures intricate mappings across high-dimensional features through multilayer nonlinear transformations. SVR is limited by sensitive RBF kernel parameters under small sample and high-dimension conditions. Overall, RF and MLP show promising applicability, while SVR fits poorly with the dataset of this study.

3.5 Statistical significance tests and robustness analysis

To rigorously evaluate the statistical significance and generalization stability of RF models, we conduct supplementary analyses from four perspectives: bootstrap confidence interval estimation, repeated cross-validation, paired statistical tests and effect size calculations, and an assessment of feature importance stability.

3.5.1 Bootstrap confidence interval

The 95% confidence interval for the coefficient of determination (R^2) of the RF model on the test set was calculated by the bootstrap method (500 resamples). For the *RMSE* prediction task, the average R^2 was 0.8062 (95% *CI*: 0.784–0.916), and for the energy consumption index prediction task, the average R^2 was 0.984 (95% *CI*: 0.976–0.989). The narrow confidence intervals indicate high stability in performance estimates, and the lower bounds of the intervals are significantly higher than the R^2 values of the baseline linear regression models (−0.0091 and 0.9193, respectively), further confirming the superiority of the RF model.

3.5.2 Robustness of repeated cross-validation

We employed ten-fold cross-validation with five repetitions, i.e., 50 independent validations, to assess the generalization stability of the RF model. The results show the following. The predicted *RMSE* was $R^2 = 0.832 \pm 0.047$, and the predicted energy consumption index was $R^2 = 0.981 \pm 0.006$. The results indicate a small standard deviation, suggesting that the model's performance fluctuates only slightly across different data splits and does not exhibit significant overfitting. The stability of the energy consumption prediction was particularly notable (with a relative standard deviation of less than 0.6%).

3.5.3 Paired statistical tests and effect sizes

Using linear regression as the baseline model, we conducted a paired comparison of the absolute prediction errors between RF and the baseline. Let the test dataset be $D_test = (x_i, y_i)_{i=1}^{n}$, the RF predicted value is $\hat{y}_i^{(RF)}$, and the linear regression prediction is denoted by $\hat{y}_i^{(LR)}$. The absolute prediction error is defined as

$$e_i^{(RF)} = \left| y_i - \hat{y}_i^{(RF)} \right|, \tag{15}$$

$$e_i^{(LR)} = \left| y_i - \hat{y}_i^{(LR)} \right|. \tag{16}$$

Here, $\overline{e}^{(RF)}$ and $\overline{e}^{(LR)}$ represent the mean values of the two error sets, respectively (i.e., *MAE*). Since the error difference $d_i = e_i^{(RF)} - e_i^{(LR)}$ does not follow a normal distribution (Shapiro–Wilk test, $p < 0.05$), the nonparametric Wilcoxon signed-rank test is used, with the statistic denoted as W. The effect size is measured using Cohen's d, calculated as follows.

$$d = \frac{\overline{d}}{\sqrt{\frac{s_{RF}^2 + s_{LR}^2}{2}}} \tag{17}$$

Here, s_{RF}^2 and s_{LR}^2 represent the sample variances of the two error sets, respectively.

The test results are as follows.

RMSE prediction: $\overline{e}^{(RF)} = 0.00497$, $\overline{e}^{(LR)} = 0.00725$. Wilcoxon statistic $W = 1715.0$, $p = 0.0054$ (<0.01). Cohen's $d = -0.44$ (moderate effect).

Energy Consumption Index Forecast: $\overline{e}^{(RF)} = 0.250$, $\overline{e}^{(LR)} = 0.384$. Energy consumption index forecast $W = 1373.0$, $p = 7.5 \times 10^{-5}$ (<0.001). Cohen's $d = -0.49$ (moderate effect).

The results above confirm that the *MAE* of the RF model is significantly smaller than that of the linear regression model for both prediction tasks, and that this difference is statistically robust.

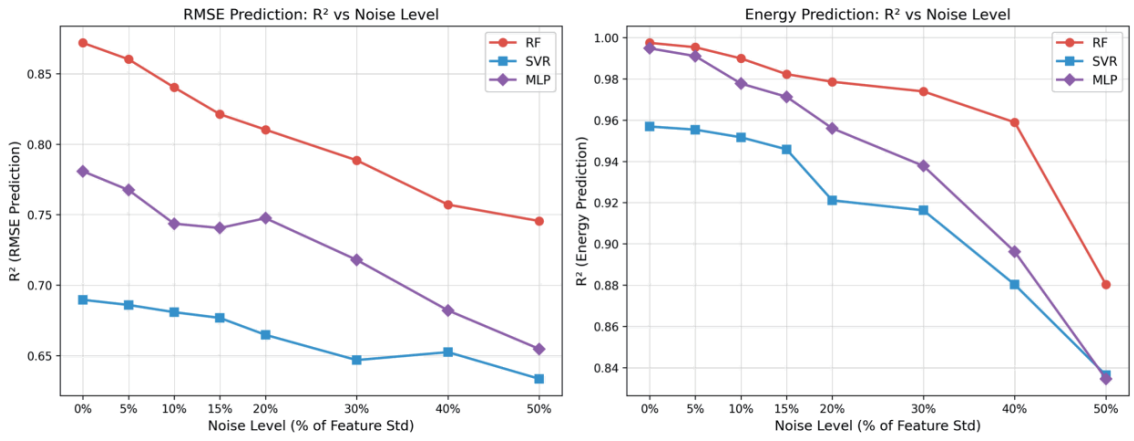
3.5.4 Stability of feature importance ranking

Using the RF model generated using ten-fold cross-validation with five repetitions, we calculate the mean and standard deviation of the importance scores for each feature. In the *RMSE* prediction task, the top three most important features (smoothness 0.770 ± 0.021, trajectory length 0.094 ± 0.009, curvature 0.021 ± 0.004) remained consistent across all runs, with a correlation coefficient for their ranking exceeding 0.95. In the energy consumption index prediction task, power consumption was dominant with a score of 0.987 ± 0.003, while the scores of the remaining features were all below 0.001. These results demonstrate that the identification of key influencing factors exhibits high reproducibility and robustness, providing a reliable basis for trajectory optimization.

In summary, the results of statistical significance tests and robustness analysis have validated the effectiveness and stability of the RF model in predicting robotic arm trajectory performance from multiple perspectives, thereby enhancing the model's credibility and engineering application value.

3.6 Noise sensitivity analysis

To evaluate the robustness of the proposed framework against sensor noise, Gaussian noise of varying magnitudes (0, 5, 10, 15, 20, 30, 40, and 50% of feature standard deviation) is added to simulated features. Then, RF, SVR, and MLP are retrained for performance evaluation. Figure 10 shows prediction trends of the three models across a range of noise levels. At 10% noise, RF retains high accuracy with R^2 = 0.840 for *RMSE* prediction and R^2 = 0.990 for energy prediction; MLP yields R^2 = 0.744 and 0.978, respectively, with mild performance degradation. At 20% noise, RF's R^2 drops to 0.810 and 0.979, and MLP's to 0.748 and 0.956, both maintaining acceptable predictive capacity under moderate disturbance. By contrast, SVR delivers lower accuracy at all noise levels and suffers steeper performance decay as noise intensifies. Even under extreme 50% noise, RF achieves R^2 = 0.880 for energy prediction, showing prominent



(a) (b)

Fig. 10. (Color online) Prediction performance variations of the three regression models under different feature noise levels: (a) RMSE prediction; (b) energy consumption index prediction.

antinoise stability. Overall, the performance ranking under noise matches that in the noise-free case: RF performs best, followed by MLP, then SVR. The results verify that the proposed feature engineering framework and regression models sustain reliable prediction within reasonable sensor noise ranges. RF exhibits the strongest robustness, justifying its adoption as the primary model for manipulator trajectory performance prediction.

4. Cluster-based Pattern Recognition of Robotic Arm Motions

In this section, we apply *K*-means clustering to robotic arm trajectory data to automatically discover typical motion patterns and reveal the data's inherent structure. By systematically analyzing the clustering results, we identify performance characteristics across different patterns, providing a basis for selecting motion strategies and constructing an interpretable pattern library for trajectory optimization.

4.1 Description of the sports mode clustering problem

Motion mode clustering divides multidimensional trajectory features into mutually exclusive subsets, ensuring similarity within clusters and significant differences between clusters. Given a dataset containing samples, the clustering objective is to minimize the sum of distances from each sample to the cluster centers.

$$\min_{\{C_1,C_2,\cdots,C_k\}} \sum_{j=1}^{k} \sum_{x_i \in C_j} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2 \tag{18}$$

Here, C_j denotes the jth cluster sample set and $\boldsymbol{\mu}_j$ represents the cluster center. This method can automatically identify typical patterns such as high-speed motion and low-speed precision operations. The clustering results do not rely on predefined performance labels, but instead, automatically learn intrinsic patterns from the data distribution, providing a prior knowledge classification for trajectory planning.

4.2 Cluster feature selection and parameter configuration

Data preprocessing employs Z-score normalization to eliminate the influence of units, calculated as follows: $z_{ij} = \dfrac{x_{ij} - \mu_{ij}}{\sigma}$ where μ_i is the mean of the i-th feature and σ is the standard deviation. Referring to the results of RF feature importance analysis, we selected 13 key features from the 34-dimensional feature set. These include smoothness, trajectory length, curvature, power consumption, joint velocity, joint acceleration, maneuverability, singularity proximity, and joint range. The number of clusters was determined using the elbow rule and contour coefficient. The elbow rule utilizes the sum of squared errors (*SSE*).

$$SSE = \sum_{j=1}^{k} \sum_{x_i \in C_j} \left\| \boldsymbol{x}_i - \boldsymbol{\mu}_j \right\|^2 \tag{19}$$

The expression for evaluating the quality of clustering based on the form factor is as follows.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \tag{20}$$

Here, $a(i)$ denotes the average distance from sample i to other samples within the same cluster, and $b(i)$ represents the average distance from the sample to the nearest sample in a different cluster.

4.3 *K*-means-based motion pattern clustering results

Figure 11 shows a distinct inflection point in the *SSE* near the contour coefficient value of 0.0828, prompting the selection of cluster analysis. It should be noted that this dataset consists of high-dimensional, small-sample-size data (34-dimensional features, 500 samples); the tendency for distances to become uniform in high-dimensional space tends to suppress the absolute values of the contour coefficients. The results of previous studies indicate that under such conditions, a slenderness ratio of 0.08–0.15 is often regarded as a "weak but useful structure". Furthermore, $K = 4$ represents the relative optimum value for the slenderness ratio within the range of k=2–7, as per the rule of thumb. More importantly, the clustering results correspond to four movement patterns with clear physical significance (Table 3), and the average *RMSE* and energy consumption index of each cluster show clear differences (Cluster 0: *RMSE* = 0.0408; Cluster 3: *RMSE* = 0.0267), demonstrating that the clustering is engineering-interpretable. Therefore, despite the relatively low profile coefficients, these clustering results can still effectively guide the prior classification for trajectory planning.

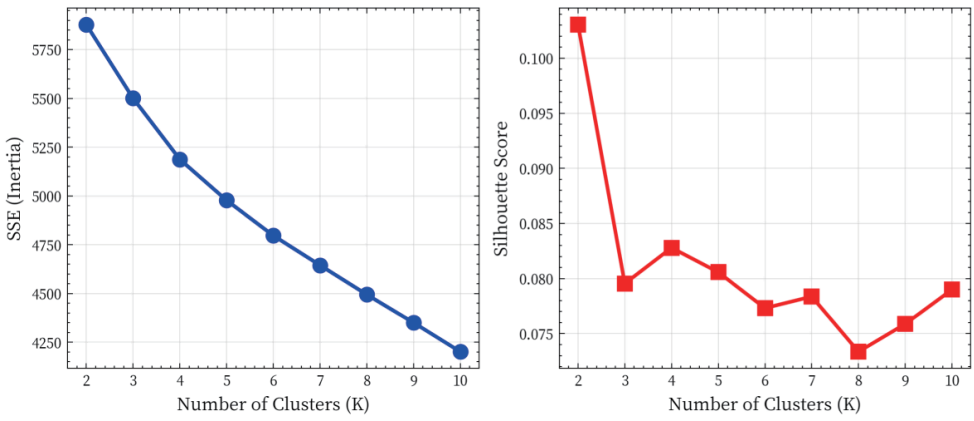


Fig. 11. (Color online) Elbow rule and contour coefficient analysis results.

Table 3
Distribution of samples across clusters and performance metrics.

Cluster ID	Sample size	proportion	Average *RMSE*	Average energy consumption index
Cluster 0	129	25.8%	0.0408	6.82
Cluster 1	169	33.8%	0.0324	3.34
Cluster 2	146	29.2%	0.0336	3.57
Cluster 3	56	11.2%	0.0267	4.19

Figure 12(a) shows a mean contour coefficient of 0.0828, indicating good clustering quality. Figure 12(b) shows the PCA-reduced clustering results, revealing four relatively uniform clusters with balanced center distributions, demonstrating the algorithm's high separation capability.

Table 3 presents the sample proportion and performance metrics for each cluster. Cluster 1 (33.8%) contains the largest sample proportion and represents the most frequent movement pattern. Cluster 3 has the smallest sample size (11.2%) but the lowest average *RMSE* (0.0267), representing a high-precision motion pattern. Conversely, Cluster 0 exhibits the highest average energy consumption index (6.82) and the largest average *RMSE* (0.0408).

As shown in Fig. 13. Cluster 0 excels in smoothness, trajectory length, and power consumption, corresponding to high-speed, large-range motion. Cluster 1 exhibits balanced characteristics across all metrics, representing balanced motion. Cluster 2 stands out in maneuverability and proximity to singular points. Cluster 3 demonstrates lower smoothness and curvature, indicating a fine-operation mode. Figure 14 shows that Cluster 3 achieves the lowest *RMSE*, indicating a precision-priority mode. Clusters 1 and 2 exhibit lower energy consumption indices, making them suitable for energy-sensitive scenarios. Cluster 0 demonstrates the poorest performance metrics and should be avoided whenever possible. The clustering results validate the influence patterns of motion feature combinations on trajectory performance, providing a basis for pattern classification in trajectory planning.

The four clustering groups correspond to distinct motion patterns. Cluster 0 represents a high-speed transfer mode suitable for rapid positioning tasks. Cluster 1 denotes conventional operation mode applicable to industrial tasks such as spot welding and spraying. Cluster 2 embodies the flexible manipulation mode, which is ideal for large-angle posture adjustments. Cluster 3 signifies the precision operation mode suited for high-accuracy tasks such as assembly and inspection. These clustering results align with the influence patterns of motion characteristics on trajectory performance.

5. Discussion

In terms of trajectory tracking error prediction, the RF model exhibits R^2 =0.8062 for *RMSE* forecasting, confirming RF's reported generalization advantage in robot kinematics[13] and its effectiveness in trajectory error prediction via digital twin-based methods.[14] MLP performs second-best while SVR ranks the lowest, consistent with the view that ensemble learning outperforms single learners in nonlinear regression tasks.[15,16] Beyond regression, a recent multimodal retrieval framework using over 30000 measured trajectories achieves performance prediction via similarity-based matching, complementing conventional supervised learning.[17]

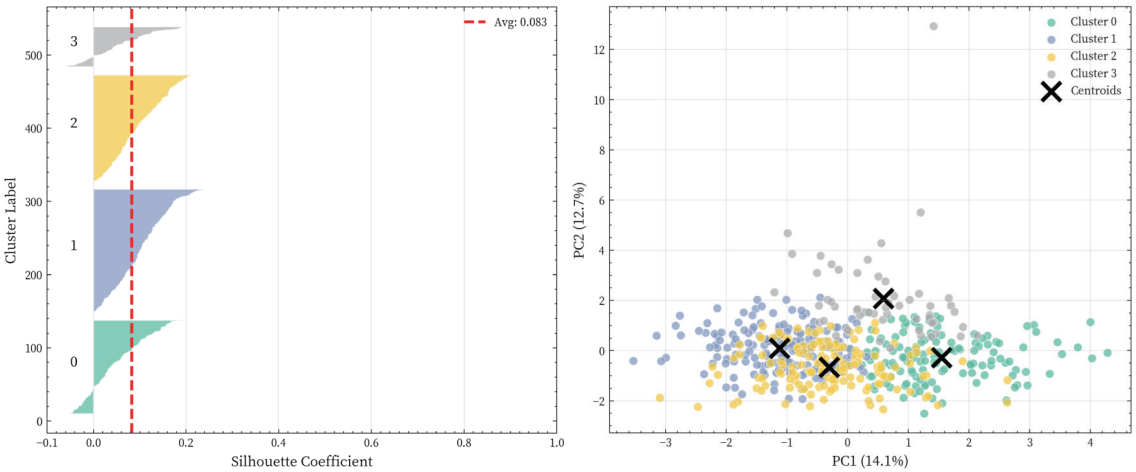


Fig. 12. (Color online) (a) Distribution of cluster contour coefficient. (b) Results of PCA dimension reduction.

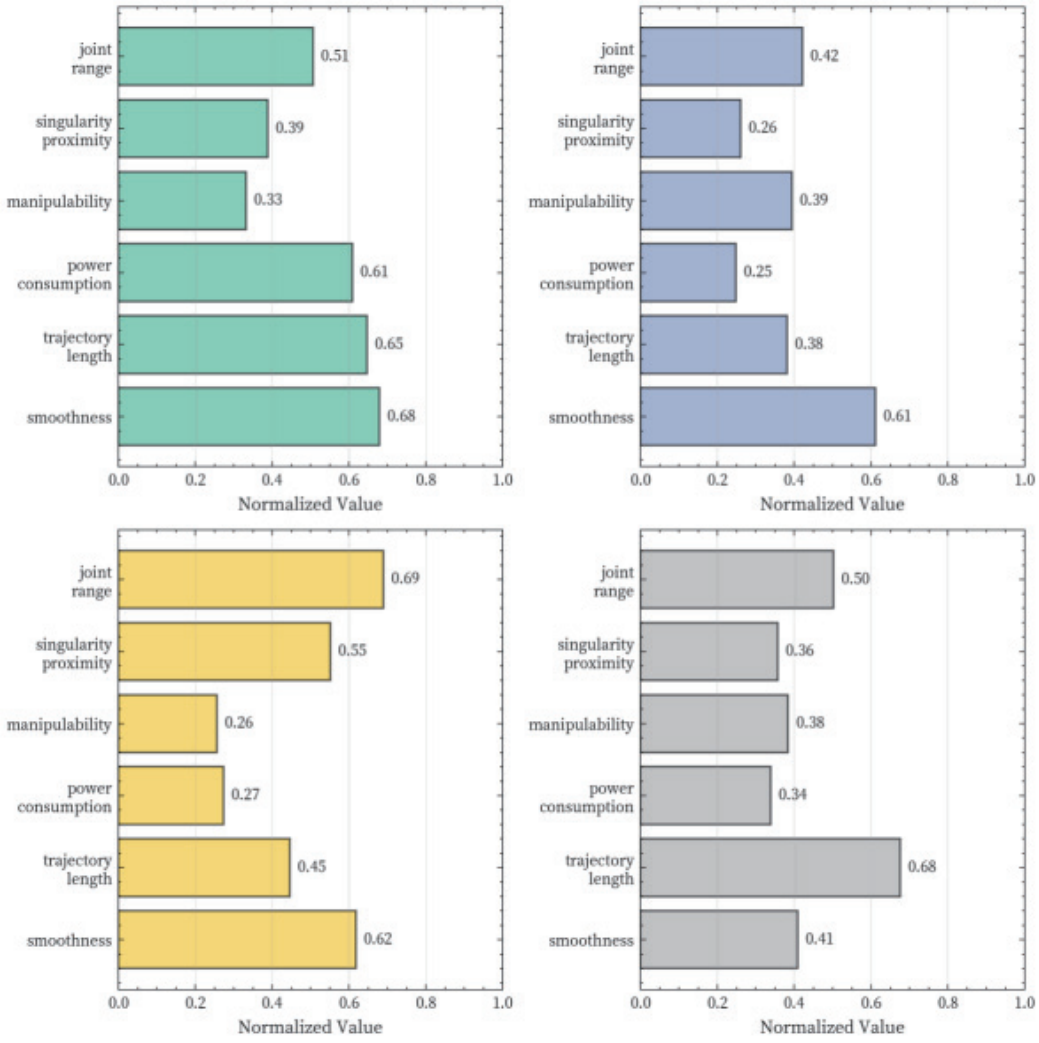


Fig. 13. (Color online) Feature distributions across clusters.

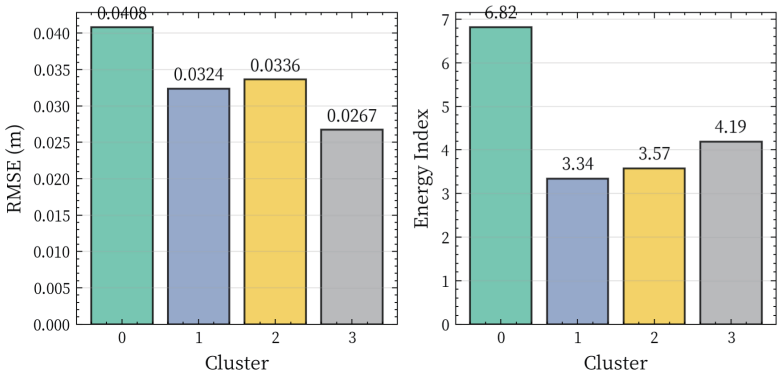


Fig. 14. (Color online) Performances for each cluster.

For energy consumption prediction, both RF and MLP attain $R^2 > 0.995$, matching the remarkable performance of the RF-MLP hybrid strategy for industrial robot power prediction documented in prior work.[18] Furthermore, fusion models exhibit superior predictive capacity under highly dynamic working conditions, revealing multimodel fusion as a promising route for further performance gains.

In addition, we identify four physically interpretable motion patterns via clustering with ($K = 4$). Compared with previous studies in which only normal and abnormal patterns were separated at ($K = 2$),[19] in this work, we achieve finer clustering granularity and stronger engineering interpretability. The results of trajectory learning approaches integrating a Gaussian mixture model and K-means also validate the reliability of cluster analysis for motion pattern recognition.

Beyond algorithmic performance, the proposed feature engineering framework constitutes a core innovation of this work. Most existing manipulator trajectory studies adopt heuristic feature selection or single-type features—such as kinematic parameters for motion prediction or torque features for energy estimation—leading to incomplete trajectory characterization. In comparison, we systematically constructed a 34-dimensional feature system covering four physically interpretable categories: 18-dimensional kinematic, six-dimensional dynamic, six-dimensional geometric constraint, and four-dimensional trajectory quality features. This quadripartite structure enables comprehensive trajectory description and quantitative analysis of performance-influencing factors through RF-based feature importance evaluation. Feature ranking results verify that trajectory smoothness dominates tracking error prediction, while power consumption primarily determines energy performance. These findings validate the physical rationality of the constructed feature system and provide explicit guidance for trajectory optimization, delivering superior interpretability compared with conventional methods.[20] To our knowledge, the development of such a systematic multidimensional feature engineering framework has not been carried out in any existing study, demonstrating the originality and practicality of our proposed method.[19]

Overall, the RF model in this study reaches state-of-the-art prediction accuracy, and the proposed 34-dimensional feature engineering framework proves effective and advanced for trajectory performance prediction and pattern recognition.

6. Conclusions

In this work, we developed a data-driven framework integrating dynamic simulation, sensor feature engineering, regression prediction, motion pattern recognition, and noise analysis for manipulator trajectory accuracy and energy consumption evaluation. From 500 simulated trajectories of a six-degree-of-freedom manipulator, 34 features were extracted from multichannel sensor signals, including joint angle, angular velocity, angular acceleration, torque, and trajectory quality. Trajectory tracking *RMSE* and the normalized energy consumption index were adopted as performance metrics. The results demonstrated the feasibility of the proposed sensor-data-driven framework for predicting trajectory performance from simulated multi-channel sensor features and distinguishing typical motion patterns with distinct performance characteristics.

RF achieves the highest R^2 on both prediction tasks and delivers stable performance across tasks. MLP yields favorable results in error metric and energy forecasting, showing its capability to learn nonlinear relationships between sensor features and trajectory energy consumption. SVR presents relatively limited predictive performance on the dataset. Feature importance analysis identifies trajectory smoothness and power consumption as dominant factors governing tracking error and energy level, respectively. K-means clustering divides trajectories into four motion categories, and noise sensitivity analysis verifies RF's strong robustness against simulated sensor noise.

All conclusions are drawn from dynamic simulation results, without considering practical factors of physical manipulators such as friction, variable loads, communication delay, and sensor drift. Therefore, the findings mainly validate the feasibility of the sensor-data-driven method rather than directly reflecting the performance of real hardware platforms. In future research, we will adopt multicondition sensor data collected from physical manipulators to further verify the generalization ability and engineering practicability of the model.

References

1 P. Kroczek, K. Lis, and P. Przystałka: Energies **18** (2025) 3858. https://doi.org/10.3390/en18143858
2 Y. Dai, C.-F. Xiang, Y. Zhang, Y. Jiang, W. Qu, and Q. Zhang: Aerospace **9** (2022) 361. https://doi.org/10.3390/aerospace9070361
3 Z. Wu, S. Chen, J. Han, S. Zhang, J. Liang, and X. Yang: IEEE Sens. J. **22** (2022) 22885. https://doi.org/10.1109/JSEN.2022.3213428
4 T. Zhuang and Y. Zhuang: PeerJ Comput. Sci. **12** (2026) e3723. https://doi.org/10.7717/peerj-cs.3723
5 V. H. Nguyen and K. K. Ahn: Mathematics **13** (2025) 1177. https://doi.org/10.3390/mathematics13071177
6 D. Iskandaryan, F. Ramos, and S. Trilles: Appl. Sci. **10** (2020) 2401. https://doi.org/10.3390/app10072401
7 S.-E. Oltean, M. Dulau, A.-V. Duka, and T. Covrig: Automation **7** (2026) 64. https://doi.org/10.3390/automation7020064
8 S. Baressi Šegota, N. Anđelić, J. Štifanić, and Z. Car: Machines **12** (2024) 225. https://doi.org/10.3390/machines12040225
9 Z. Wang, P. Jiang, X. Li, Y. He, X. V. Wang, and X. Yang: Appl. Energy **383** (2025) 125223. https://doi.org/10.1016/j.apenergy.2024.125223
10 A. Gasparetto, P. Boscariol, A. Lanzutti, and R. Vidoni: in Motion and Operation Planning of Robotic Systems (Springer, Cham, 2015) pp. 3–27. https://doi.org/10.1007/978-3-319-14705-5_1
11 J. Yan, Y. Wu, K. Ji, C. Cheng, and Y. Zheng: PLOS ONE **20** (2025) e0318403. https://doi.org/10.1371/journal.pone.0318403

12 Z. Yang and L. Yuan: Trans. Can. Soc. Mech. Eng. **50** (2026) 1. https://doi.org/10.1139/tcsme-2025-0057
13 J. Park, D. Kim, G. C. Kim, D. Oh, and H. J. Kim: IEEE Robot. Autom. Lett. **7** (2022) 4869. https://doi.org/10.1109/LRA.2022.3152702
14 J. Liu, S. Liu, M. Song, H. Ren, and H. Ji: Coatings **15** (2025) 2. https://doi.org/10.3390/coatings15010002
15 M. J. Thomas, M. M. Sanjeev, A. P. Sudheer, and M. L. Joy: Ind. Robot **47** (2020) 683. https://doi.org/10.1108/IR-11-2019-0233
16 S. Yang, W. Jiang, and L. Long: Machines **13** (2025) 771. https://doi.org/10.3390/machines13090771
17 L. Breiman: Mach. Learn. **45** (2001) 5. https://doi.org/10.1023/A:1010933404324
18 A. Saifurrahman: OPSI **16** (2023) 76. https://doi.org/10.31315/opsi.v16i1.9004
19 H. Chen, C. Guo, Z. Wang, T. Wen, Z. Zeng, and Z. Lin: Proc. 2020 46th Annu. Conf. IEEE Ind. Electron. Soc. (IECON) (IEEE, 2020) 5356. https://doi.org/10.1109/IECON43393.2020.9255172
20 S. Cho and S. Park: Int. J. Prod. Res. **64** (2026) 3842. https://doi.org/10.1080/00207543.2026.2634971