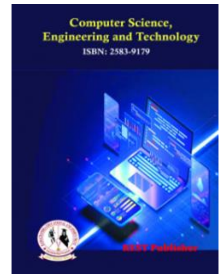
Computer Science, Engineering and Technology

Vol: 3(2), June 2025

REST Publisher; ISSN: 2583-9179

Website: https://restpublisher.com/journals/cset/

DOI: https://doi.org/10.46632/cset/3/2/24



Machine Learning for Robot Precision: Predicting End-of-Operation Errors Using Regression Techniques

[1]C Murugamani, [2]Anantharamaiah Vengala, [3]P. Jyothi, [4]Thangiah Sathishkumar

[1]Bhoj Reddy Engineering College for Women, Hyderabad, India.
[2]K L E F, Vaddeswaram, Andhra Pradesh, India.
[3]VNR Vignana Jyothi Institute of Engineering and Technology, Hyderabad, India.
[4]Hyderabad Institute of Technology and Management, Hyderabad, India.
*Corresponding author Email: murugamanija@gmail.com

Abstract: *This study provides an in-depth evaluation of predictive modeling techniques used to estimate end-effector error in robotic systems, comparing the performance of four machine learning models: linear regression, multilayer perceptron (MLP), decision tree regression, and support vector regression (SVR). The research focuses on addressing key challenges in precision control by analyzing how operational parameters such as joint angles, motor torque, sensor delay, and battery voltage affect the accuracy of end-effector positioning. A dataset consisting of 20 observations with joint angles ranging from 20° to 95°, motor torques ranging from 1.0 to 2.2 Nm, sensor delays ranging from 7 to 16 milliseconds, and battery voltages ranging from 22.3 to 25.1 volts was used. A rigorous testing approach was adopted using several performance metrics including R-squared (R^2), mean squared error (MSE), root mean squared error (RMSE), and mean absolute error (MAE). Linear regression gave the best results, scoring 0.9838 on training and 0.9594 on test data, indicating a strong linear relationship between input features and end-product error. The MLP model also performed well, capturing nonlinear patterns with R^2 values of 0.98075 (training) and 0.94832 (test). The decision tree regression model showed a perfect fit on the training data (R^2 = 1.0000), while it showed overfitting, leading to a decrease in performance on the test data (R^2 = 0.9240). SVR provided a balanced effect with strong generalization, showing R^2 values of 0.9606 and 0.9501 on the training and test datasets, respectively. Correlation analysis identified strong positive correlations between the final-effector error and motor torque (0.97), joint angle 1 (0.93), and sensor delay (0.89). Conversely, battery voltage had a strong negative correlation (-0.95), indicating that higher voltage improves position accuracy. These insights underscore the importance of efficient power usage and timely sensor feedback in achieving high-precision robot control.*

Key words: *Robotics, End-effector error, Machine learning, Predictive modeling, Linear regression, multi-layer perceptron, Decision tree regression, Support vector regression, Robot precision, Joint angles, Motor torque.*

1. INTRODUCTION

From a purely theoretical concept, robotics has emerged as a major part of contemporary life, influencing fields such as healthcare, manufacturing, education, and culture. The term "robotics" encompasses the multidisciplinary study of the design, construction, operation, and use of robots. These largely autonomous machines are increasingly shaping how we live and what we expect from technology.[1] Central to the advancement of robotics is the creation of intelligent systems that are capable of learning and adapting. Reinforcement learning, a branch of artificial intelligence, has shown great potential in this area. Unlike traditional methods that follow fixed instructions, reinforcement learning allows robots to improve their performance by interacting with their surroundings. This approach is particularly useful in complex, real-world situations where pre-established programming is flawed. However, challenges arise due to the complexity of physical systems, the high cost of real-world experimentation, and the difficulty of accurately simulating dynamic environments. To address these challenges, techniques such as integrating prior knowledge, using simulations for initial training, and creating abstract representations are necessary.[2] A key focus in the field of robotics is identifying the grand challenges that will shape its future direction. According to Science Robotics, these challenges include developing innovative materials, energy-efficient robotics systems, biologically inspired designs, swarm robotics, and advanced human-robot interaction. Of particular note, the emphasis on social and medical robotics underscores the need for ethical, sustainable, and empathetic integration of robots into everyday human environments.[3] Healthcare is one of the fastest growing and impacting areas in the field of robotics. Robots are increasingly central to applications ranging from computer-assisted surgery to geriatric care. Systems such as the da Vinci surgical robot have transformed medical practice by enabling minimally invasive surgeries with improved precision and shorter recovery times. However, concerns remain about the gap between theoretical feasibility and real-world application. Many healthcare robotics systems are at the prototype stage or in limited experimental use, and issues related to ethics, patient safety, and preserving human-centered care continue to be debated. Despite these obstacles, the promise of robotics to reduce the pressure on strained healthcare systems is driving innovation and investment.[4] Medical robotics derives significant benefits from the broader field of computer-integrated surgery (CIS), where robotic tools assist in planning, real-time surgical assistance, and post-operative assessment. By combining imaging technologies, artificial intelligence, and robotic algorithms, CIS improves surgical precision while minimizing invasiveness. This integration highlights the effective collaboration between computational tools and physical implementation, reinforcing the important role of robotics in advancing precision medicine.[5] Beyond healthcare, robotics is having a significant impact on education. Innovative programs have introduced robotics to students as early as elementary school, encouraging skills such as problem-solving, programming, and logical reasoning. These efforts not only prepare students for a future career in technology, but also encourage teamwork and creative thinking. When integrated into a variety of subjects such as math, science, and language arts, robotics serves as a powerful cross-curricular tool that enriches the overall educational experience.[6] Robotics also has significant cultural significance. As robots become more common in homes, workplaces, and public settings, they are beginning to shape and reflect social norms and values. The field of cultural robotics explores these two-way interactions—how robots can represent cultural beliefs and how human interactions with robots influence emerging social norms. It explores complex questions surrounding identity, empathy, and the evolving relationship between humans and machines, especially as robots take on roles that were once considered uniquely human, such as caregivers, companions, or creative partners.[7] Robotics has been rapidly adopted in industry, particularly in sectors such as food processing and manufacturing. Robots are now used for a variety of tasks, from planting and harvesting to inspecting product quality and handling packaging. Technologies such as vision systems, automated picking machines, and collaborative robots have significantly improved efficiency and precision. In the food industry, robotic systems must also comply with strict hygiene regulations, which necessitates careful design and material selection. Despite ongoing challenges, the benefits, including higher productivity and reduced reliance on manual labor, make robotics a valuable investment for food producers worldwide.[8] Despite their many advantages, robotic systems face many technical hurdles. An important area of ongoing research is execution monitoring – the process of verifying that a robot's actions are consistent with its intended objectives. Failures can stem from hardware problems, unpredictable environmental conditions, or flawed system models. Robust execution monitoring enables the detection and diagnosis of such problems in real time, allowing robots to adjust their behavior or stop operations before significant errors occur. This functionality is especially important for autonomous robots operating in dynamic systems, such as drones, delivery bots, or mobile service assistants. At the same time, ethical considerations play a major role in the evolution of robotics.[9] As robots increasingly participate in social roles and decision-making processes, concerns about accountability, privacy, and security become more urgent. In domains such as healthcare and social assistance, decisions made by autonomous systems can have serious consequences. This underscores the need for collaboration between developers, ethicists, and policymakers to develop clear standards and guidelines that promote responsible innovation. Recognizing ethics as a fundamental challenge in robotics reflects a broader understanding that technological progress must be guided by societal values.[10] Looking ahead, robotics is poised for rapid and transformative growth. Advances in machine learning, materials engineering, and human-computer interaction will continue to expand the capabilities of robotics systems. However, with this progress, it is critical to maintain a critical perspective on the broader social, cultural, and ethical implications. As robots become increasingly embedded in everyday environments, whether in hospitals, classrooms, workplaces, or homes, the ultimate goal is not just to improve functionality, but to ensure that these technologies contribute positively and equitably to human life.[11]

2. MATERIALS AND METHOD

Linear Regression is a traditional and widely accepted statistical method that assumes a direct linear relationship between input variables and output. Its simplicity makes it easy to interpret and computationally efficient, making it very advantageous in many engineering contexts. When predicting end-effector error, linear regression works best if the underlying relationships between features and the target are approximately linear and the data is low in noise. The performance of a model is typically assessed using metrics such as R-squared (R^2), mean squared error (MSE), and mean absolute error (MAE). High R^2 values combined with low error metrics indicate a strong fit and reliable predictions. A significant advantage of linear regression is its transparency, as the coefficients provide direct insight into how each feature affects the prediction, which aids in understanding and decision-making. However, a major drawback is its inherent linear assumption, which limits its accuracy when the relationships are nonlinear or very complex. It can also be sensitive to multicollinearity, where related features can distort the model.

Multi-Layer Perceptron A form of artificial neural network, the multi-layer perceptron (MLP), offers a more advanced approach capable of modeling complex and nonlinear relationships. Unlike linear regression, MLP consists of multiple layers of interconnected neurons with nonlinear activation functions, allowing it to learn complex mappings between inputs and outputs. This adaptability makes MLP particularly useful for robotics tasks, where system behaviors such as joint angles, motor torque, and sensor latency are rarely linear. However, MLP models require high computational effort and careful tuning, including choosing the architecture (number of layers and neurons) and using regularization techniques to avoid overfitting. Training typically uses back propagation to iteratively update the weights to reduce errors. Although MLPs often achieve high accuracy on training data by capturing complex patterns, thorough validation is necessary to ensure good generalization. While MLPs perform better than linear regression in handling nonlinearities, they are less interpretable, often considered "black box" models.

Decision Tree regression is a flexible, nonparametric method that iteratively divides data into regions with similar outputs. It handles nonlinear relationships and feature correlations well without requiring assumptions about the data distribution. Decision trees are easy to interpret visually because their structure resembles logical decision rules. They often achieve perfect fits on training data by closely following the data points, which leads to fitting more than meaningful patterns - capturing noise. This leads to better training metrics, but poor performance, especially on unseen data, indicates poor generalization. Overfitting can be reduced by using ensemble techniques such as random forests or. Despite this limitation, decision trees are popular for their descriptiveness and ability to work with a wide variety of data types.

Support Vector Regression (SVR) balances bias and variance by extending the principles of support vector machines to regression tasks. SVR fits a function within a tolerance margin (epsilon) by relying on support vectors—data points near the edge boundaries—to define the model. This approach helps prevent overfitting better than decision trees or MLPs, especially when the data contains noise or outliers. SVR accommodates both linear and nonlinear relationships through kernel functions such as the radial basis function (RBF), which allows for flexible modeling of complex patterns. In predicting end-effector error, SVR often provides consistent performance across training and test sets, maintaining high R^2 and low error metrics. This robustness and ability to generalize well make SVR well suited for precision-oriented engineering applications. However, SVR can be computationally demanding on large datasets and requires careful hyperparameter tuning to balance model complexity and edge width.

This dataset provides measurements of six important parameters from a robotic system: joint angle 1, joint angle 2, motor torque, sensor delay, battery voltage, and the resulting end-effector error, each expressed in their respective units. Examining these values reveals significant patterns related to the accuracy and functionality of the robotic arm. Joint angle 1 varies from 20° to 95°, while joint angle 2 ranges from 25° to 90°, covering a wide range of arm positions. Motor torque values range from 1.0 to 2.2 Nm, reflecting the varying amounts of force applied by the motors depending on the arm position and operational requirements. Sensor delay varies from 7 MS to 16 MS, indicating variations in the timing of sensor feedback and control signals, which can affect the responsiveness of the system. Battery voltage ranges from 22.3 V to 25.1 V, reflecting changes due to battery load or state of charge during operation. The end-effector error, which measures position accuracy in millimeters, varies from 0.28 mm to 1.5 mm. Higher errors are typically observed when the joint angle 1 is large and the motor torque exceeds 2.0 Nm, in which cases the errors reach up to 1.5 mm. In contrast, lower errors are associated with smaller joint angles, reduced motor torque, shorter sensor delays, and higher battery voltages - in situations with angles ranging from 20° to 40° and voltages close to 25 V, the errors decrease below 0.4 mm. This indicates that accuracy decreases with more extreme joint angles and higher motor loads, whereas higher battery voltage seems to improve performance by stabilizing motor operation. Sensor delays also plays a role in the error magnitude; longer delays, especially those above 13 MS, are consistent with increased end-effector errors. This underscores the importance of immediate sensor feedback to maintain precise control and stabilization.

3. ANALYSIS AND DISSECTION

TABLE 1. Descriptive Statistics

	Joint Angle 1 (°)	Joint Angle 2 (°)	Motor Torque (Nm)	Sensor Delay (MS)	Battery Voltage (V)	End-Effector Error (mm)
count	20.0000	20.0000	20.0000	20.0000	20.0000	20.0000
mean	58.2500	58.4500	1.5000	11.2000	23.7900	0.7795
std	22.4168	18.9667	0.3613	2.7644	0.9182	0.3902
min	20.0000	25.0000	1.0000	7.0000	22.3000	0.2800
25%	41.5000	47.2500	1.2000	9.0000	22.9500	0.4375
50%	59.0000	57.5000	1.4500	11.0000	23.9500	0.7650
75%	72.7500	70.5000	1.7250	13.2500	24.5250	0.9625
max	95.0000	90.0000	2.2000	16.0000	25.1000	1.5000

Table 1 summarizes descriptive statistics for six parameters associated with a robotic or electromechanical system: joint angle 1, joint angle 2, motor torque, sensor delay, battery voltage, and end-effector error. Each parameter is derived from a dataset of 20 observations, representing a modest sample size for statistical evaluation. The joint angles, recorded in degrees, show significant variability. Joint angle 1 has a mean value of 58.25° with a standard deviation of 22.42°, while joint angle 2 shows a nearly identical mean of 58.45°, but with slightly lower variability (standard deviation of 18.97°). The spread of values—from 20° to 95° for joint angle 1 and from 25° to 90° for joint angle 2—reflects a wide range of motion across different cases. The motor torque, measured in Newton-meters (Nm), exhibits a mean of 1.5 Nm and a standard deviation of 0.36 Nm. This indicates a relatively stable torque output, with a minimum of 1.0 Nm and a maximum of 2.2 Nm, indicating little fluctuation in motor load or performance. The sensor latency, expressed in milliseconds (MS), has a mean latency of 11.2 MS and a standard deviation of 2.76 MS. The data range from 7 MS to 16 MS reveals little variation that could be due to system latency or delays in data processing. The battery voltage appears to be very stable, with a mean of 23.79 V and a standard deviation of 0.92 V. The voltage range from 22.3 V to 25.1 V is within the typical operating limits for battery-powered devices, indicating reliable power delivery throughout the measured tests.

TABLE 2. Linear Regression Models End-Effector Error (mm) Train and Test performance metrics

Linear Regression	Train	Test
R2	0.9838	0.9594
EVS	0.9838	0.9791
MSE	0.0022	0.0062
RMSE	0.0468	0.0789
MAE	0.0418	0.0684
Max Error	0.0853	0.1327
MSLE	0.0007	0.0032
Med AE	0.0394	0.0539

Table 2 presents the evaluation of the linear regression model used to estimate the end-effector error (in millimeters) based on key performance metrics for the training and testing datasets. The findings highlight the model's strong fit, high accuracy, and consistently low error rates across both sets. The R-squared (R^2) values are 0.9838 for training and 0.9594 for testing, meaning that the model accounts for 98.4% of the variance in the training data and 95.9% in the testing data. These values reflect excellent model performance and generalization, with minimal signs of overfitting. Likewise, the explained variance score (EVS) is equally high, recording 0.9838 for training and 0.9791 for testing, confirming the model's ability to capture the variance in the output. The model also demonstrates low prediction errors. The mean squared error (MSE) for training is 0.0022 and for testing is 0.0062, while the root mean squared error (RMSE) is 0.0468 mm and 0.0789 mm, respectively. These measurements confirm that the model's predictions closely match the actual end-effector errors. The mean absolute error (MAE) is 0.0418 mm for the training data and 0.0684 mm for the testing data, further emphasizing the prediction consistency. Although the maximum error increases slightly from 0.0853 mm in training to 0.1327 mm in testing, this increase is within acceptable limits for practical applications. The mean square logarithmic error (MSLE) values are low, indicating minimal logarithmic discrepancies, while the mean absolute error (Med AE) also shows small differences – 0.0394 mm for training and 0.0539 mm for testing – highlighting the reliability and consistency of the model in predicting the end-effector error.

TABLE 3. Multi-layer Perceptron Models End-Effector Error (mm) Train and Test performance metrics

Multi-layer Perceptron	Train	Test
R2	0.98075	0.94832
EVS	0.98111	0.94978
MSE	0.00261	0.00792
RMSE	0.05113	0.08899
MAE	0.04435	0.07167
Max Error	0.09181	0.14208
MSLE	0.00101	0.00269
Med AE	0.04054	0.07600

Table 3 shows the evaluation metrics of the Multi-Layer Perceptron (MLP) model used to predict the end-effector error in millimeters, considering both the training and test datasets. The results suggest that the model achieves strong predictive accuracy and demonstrates good generalization ability between training and unseen data. The coefficient of determination (R^2) scores Are 0.98075 for the training set and 0.94832 for the test set, indicating that the model captures more than 98% of the variance during training and almost more than 95% of the variance during testing. Although there is a small drop in performance on the test data, the difference is negligible, indicating effective generalization with limited overfitting. The explained variance score (EVS) follows a similar pattern, recording 0.98111 for training and 0.94978 for testing, underscoring the consistent ability of the model to capture the variance in the target variable. The error metrics support the accuracy of the model. The mean square error (MSE) increases slightly from 0.00261 (training) to 0.00792 (test), and the root mean square error (RMSE) increases from 0.05113 mm to 0.08899 mm. These values indicate a small increase in prediction error with new data, but the overall deviation is within the acceptable range. The mean absolute error (MAE) and mean absolute error (Med AE) reflect low error levels, with values of 0.04435 mm and 0.04054 mm for training and 0.07167 mm and 0.076 mm for testing, respectively. These metrics highlight the stability and reliability of the model. The maximum error between training and testing increases from 0.09181 mm to 0.14208 mm, which is expected in real-world conditions.

TABLE 4. Decision Tree Regression Models End-Effector Error (mm) Train and Test performance metrics

Decision Tree Regression	Train	Test
R2	1.0000	0.9240
EVS	1.0000	0.9320
MSE	0.0000	0.0117
RMSE	0.0000	0.1079
MAE	0.0000	0.0850
Max Error	0.0000	0.2500
MSLE	0.0000	0.0032
Med AE	0.0000	0.0650

Table 4 summarizes the performance metrics of a decision tree regression model developed to estimate the estimated end-effector error in millimeters on both the training and test datasets. The results demonstrate an excellent fit on the training data and very strong accuracy on the test data, although there is evidence of overfitting. The training set achieves a perfect R-squared (R^2) value of 1.0000, meaning that the model fully accounts for the variance in the training data. This is further supported by an explained variance score (EVS) of 1.0000, which confirms that the model's predictions accurately match the actual results during training. In contrast, the test set yields an R^2 of 0.9240 and an EVS of 0.9320 – still robust, but representing a drop in performance. This difference indicates that the model may be overfitting the training data, capturing specific patterns or noise that does not generalize well to the unseen data. The error measurements reinforce this interpretation. The training error values – mean square error (MSE), root mean square error (RMSE), mean absolute error (MAE), mean square logarithmic error (MSLE), and mean absolute error (Med AE) – are all zero, indicating perfect accuracy on the training data. However, for the test set, the MSE increases to 0.0117 and RMSE to 0.1079 mm, indicating a significant increase in prediction error. Similarly, the MAE and Med AE increase to 0.0850 mm and 0.0650 mm, respectively, reflecting moderate average deviations from the true values. The maximum error observed in the test is 0.2500 mm, which is the largest of all the models analyzed –

highlighting occasional large deviations in the prediction. Despite this, the mean square logarithmic error (MSLE) remains low at 0.0032, indicating that the model handles scaling differences well.

TABLE 5. Support Vector Regression Models End-Effector Error (mm) Train and Test performance metrics

Support Vector Regression	Train	Test
R2	0.9606	0.9501
EVS	0.9607	0.9701
MSE	0.0053	0.0076
RMSE	0.0731	0.0874
MAE	0.0613	0.0664
Max Error	0.1005	0.1557
MSLE	0.0017	0.0038
Med AE	0.0677	0.0391

Table 5 shows the estimation results of a support vector regression (SVR) model developed to estimate end-effector error in millimeters, using both training and test datasets. The SVR model achieves solid predictive accuracy, demonstrating reliable performance and effective generalization, with only minor signs of overfitting. The R-squared (R^2) value for the training set is 0.9606, meaning that the model accounts for 96.06% of the variance in the training data. For the test set, the R^2 is slightly lower at 0.9501, indicating even higher predictive power. The small difference between these values indicates that the model generalizes well and does not overfit the training data. The explained variance score (EVS) aligns closely, recording 0.9607 for training and a slightly improved 0.9701 for test, further confirming the consistent model performance. Error-related metrics also support this conclusion. The mean square error (MSE) is relatively low - 0.0053 for training and 0.0076 for testing - indicating small average deviations between the predicted and actual values. Similarly, the root mean square error (RMSE) values are moderate, 0.0731 mm and 0.0874 mm for training and testing, respectively, indicating minimal prediction errors. The mean absolute error (MAE) remains low, 0.0613 mm for training and 0.0664 mm for testing. In particular, the mean absolute error (Med AE) is lower in the testing phase (0.0391 mm) compared to training (0.0677 mm), indicating more stable predictions on the unseen data. Although the maximum error increases from 0.1005 mm in training to 0.1557 mm in testing, this increase is within the acceptable range, especially for precision-oriented tasks.

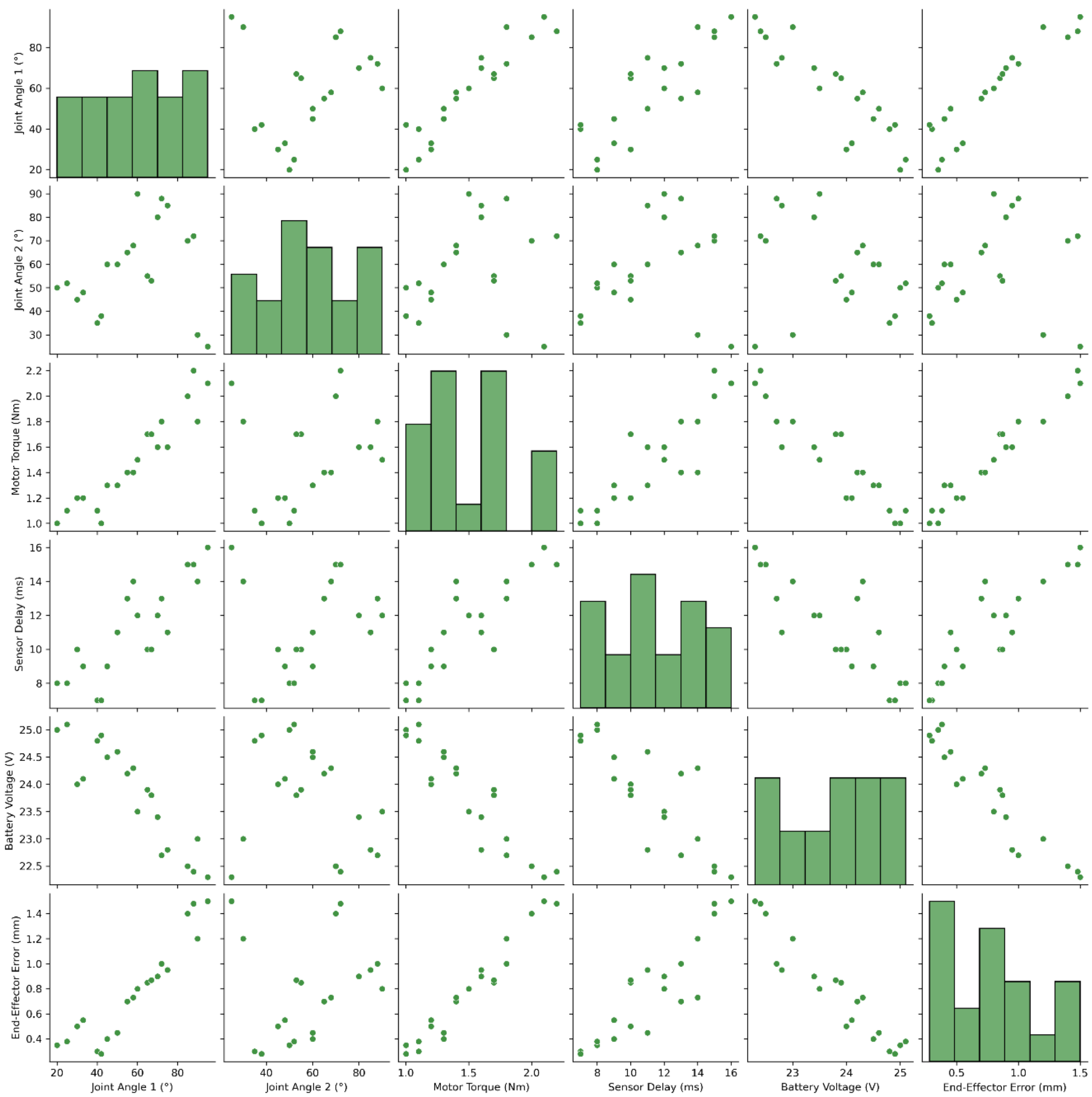


FIGURE 1. Effect of Process Parameters

This pair diagram provides a detailed visual overview of the relationships between six process variables: joint angle 1, joint angle 2, motor torque, sensor delay, battery voltage, and end-effector error. Along the diagonal, the histograms show the distribution of each variable, with some, such as joint angle 1 and motor torque, following an approximately normal distribution, while others, such as end-effector error, show a skewed pattern. Below the diagonal, the scatter plots illustrate the relationships between the variable pairs. In particular, there is a strong linear relationship between motor torque and joint angle 1, sensor delay, and end-effector error, indicating that these factors are closely linked within the robot system. In contrast, joint angle 2 shows weak or negligible relationships with most other parameters. End-effector error appears to be particularly responsive to changes in joint angle 1, motor torque, and battery voltage. The negative relationship observed with battery voltage suggests that increasing voltage will help reduce the error, a factor that is important for improving the accuracy of the system.

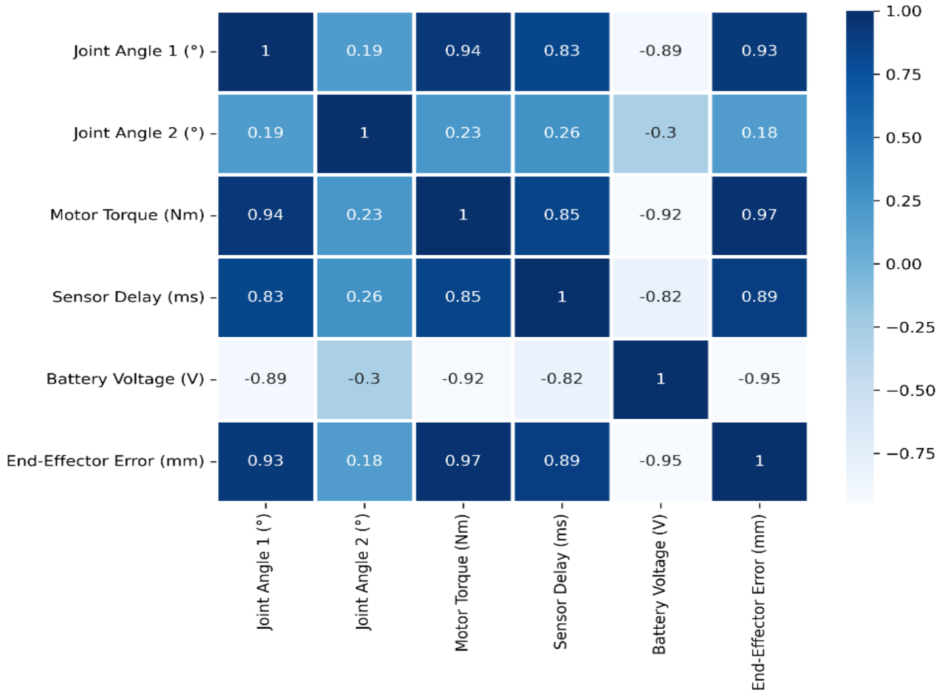


FIGURE 2. Correlation heatmap

The heat map shows the magnitude and direction of the linear relationships between the variables using Pearson correlation coefficients. The end-effector error shows strong positive correlations with motor torque (0.97), joint angle 1 (0.93), and sensor delay (0.89), while it shows a strong negative correlation with battery voltage (−0.95), confirming the patterns observed in the pair diagram. These relationships indicate that increases in motor torque and sensor delay are associated with higher error values, while higher battery voltage tends to reduce the error. Joint angle 2 shows weak correlations with all other variables, indicating that it has minimal impact on the performance of the system. In addition, the strong positive correlations between joint angle 1, motor torque, and sensor delay highlight the possibility of multicollinearity, which may need to be addressed during predictive modeling to ensure accurate and reliable results.

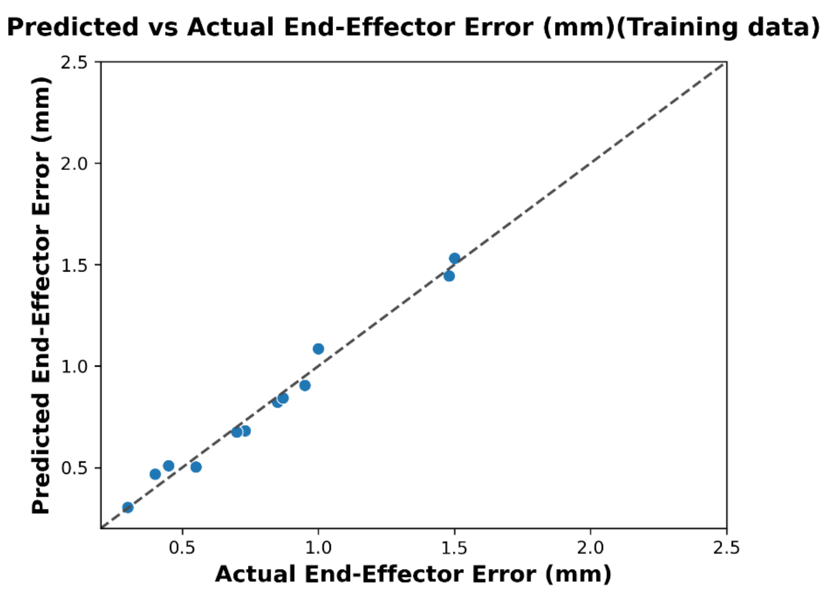


FIGURE 3. Linear Regression End-Effector Error (mm) Training

The scatter plot comparing the predicted and actual end-effector error for the training data shows a strong alignment between the model's outputs and the actual values. Most of the points closely follow the diagonal line (y = x), indicating that the model effectively captures the relationships in the training set. This indicates that the linear regression learned well from the model data, due to the strong correlations between important variables such as motor torque, joint angle 1, and sensor delay. However, some small deviations occur at higher error values, where the model slightly underestimates the actual errors. This may indicate nonlinear patterns that the linear regression approach does not fully address. Despite this, the overall training performance of the model is strong, indicating that the selected features are appropriate and that the model shows low bias in fitting the training data.

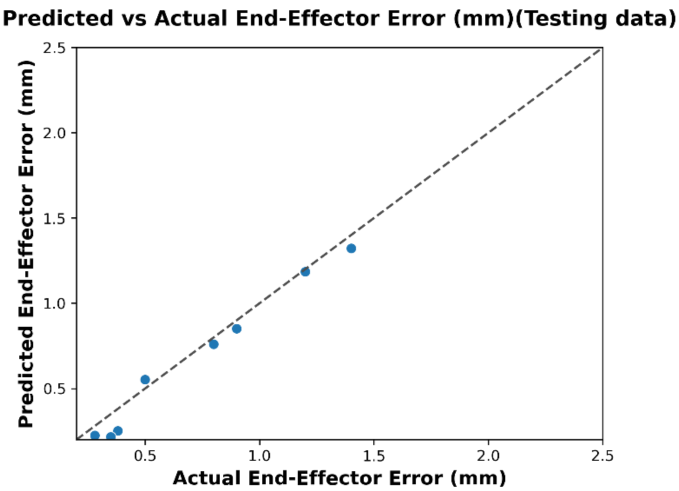


FIGURE 4. Linear Regression End-Effector Error (mm) Testing

The plot for the experimental data similarly shows a strong alignment between the predicted and actual values, confirming the model's ability to generalize well to new data. Most of the points are close to the ideal diagonal line, with only small deviations, indicating that the model has avoided overfitting during training. The pattern closely matches what is observed on the training data, further supporting the model's reliability when applied to unseen samples. While the discrepancies are minimal at low error values, there is some underestimation at moderate error levels. Nevertheless, the overall high accuracy indicates that the key variables identified – particularly motor torque and battery voltage – act as useful predictors of end-effector error.

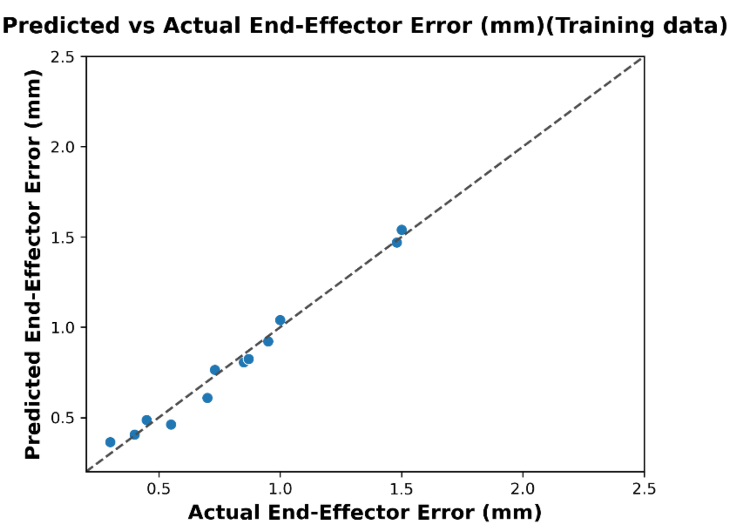


FIGURE 5. Multi-layer Perceptron End-Effector Error (mm) Training

Figure 5 depicts the relationship between the actual and predicted end-effector error (in millimeters) on the training dataset using a multi-layer perceptron (MLP) model. The scatterplot compares these values, with the dashed diagonal line representing the best case where the predictions match the actual measurements exactly. The data points closely align with this line, indicating that the MLP model successfully captures the complex nonlinear patterns in the training data. The predictions show minimal deviation from the actual error values, especially within the lower to mid-range error spectrum (0.3–1.5 mm). This close fit reflects the robust accuracy of the MLP during training, benefiting from its ability to learn complex relationships between input features and output. Unlike linear models, MLP uses multiple hidden layers and nonlinear activation functions, which helps model not only the linear relationship between motor torque and end-effector error, but also the nonlinear relationships found among variables with weak correlations, such as joint angle 2 or sensor delay. The absence of significant outliers or consistent over- or under-predictions indicates that the model is well-trained and has appropriate regularization and network architecture. However, caution is necessary to ensure that this level of performance is maintained for new, unobserved data, as neural networks can easily be over-trained without proper validation strategies.

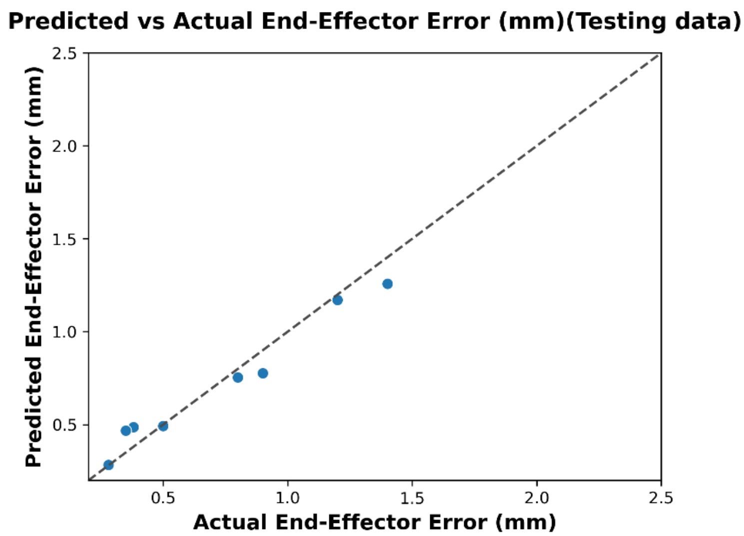


FIGURE 6. Multi-layer Perceptron End-Effector Error (mm) Testing

Figure 6 shows a comparison between the predicted and actual end-result errors for the test dataset using the Multi-Layer Perceptron (MLP) model. Most of the data points are close to the diagonal reference line (y = x), indicating strong predictive performance. The tight clustering around this line indicates that the MLP model generalizes effectively to new, unseen data, accurately capturing underlying patterns beyond the training set. While small deviations occur at high error values, the overall predictions show minimal bias and low variance. This demonstrates the robustness and suitability of the model for accurate regression tasks in applications such as robotics.

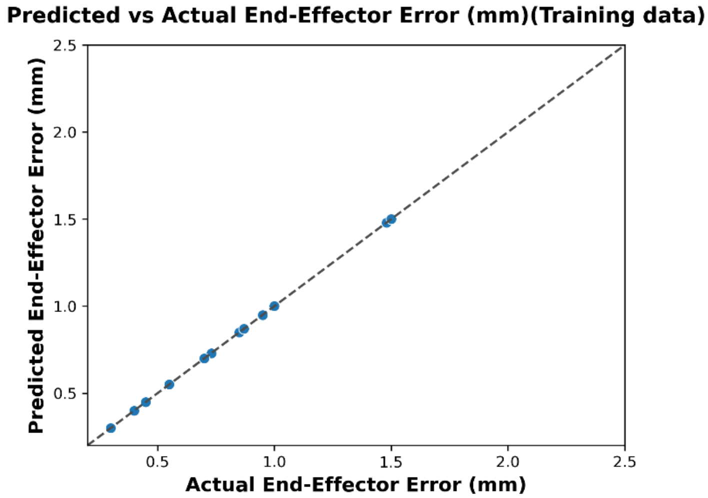


FIGURE 7. Decision Tree Regression End-Effector Error (mm) Training

Figure 7 depicts the performance of the decision tree regression model on the training dataset. Most of the points align almost perfectly with the diagonal line, indicating a very accurate fit to the training data. This almost perfect fit indicates that the model may be overfitting, meaning that it has learned not only the underlying patterns but also the noise within the training set. Although this results in very low training error, it raises potential concerns about how well the model will perform on unseen data, as overfitting can limit generalization.

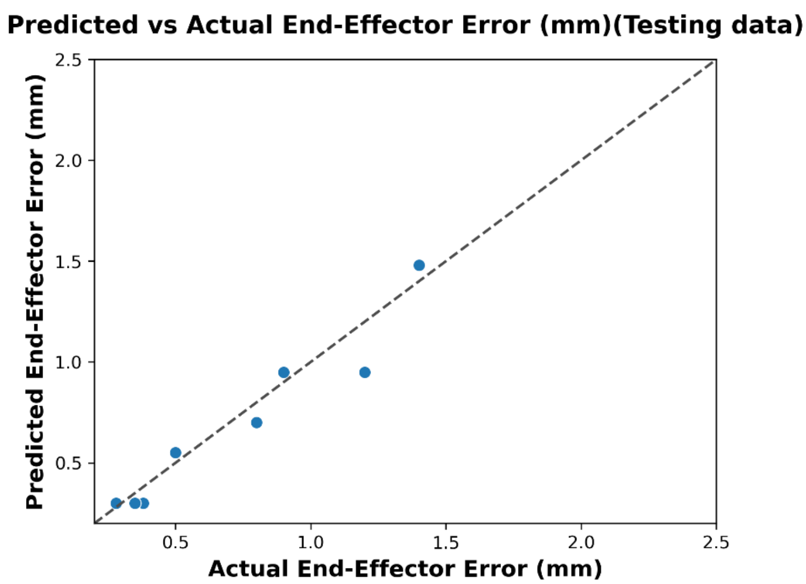


FIGURE 8. Decision Tree Regression End-Effector Error (mm) Testing

Unlike its performance on the training data, the decision tree model shows considerable variation when evaluated on the test dataset (Figure 8). While some predicted values closely match the actual measurements, many points deviate significantly from the best-fit diagonal line. This variation confirms the previous concern of overfitting, as the model struggles to maintain accuracy on new, unseen data. The results emphasize the model's vulnerability to variations in the data and its limited ability to generalize beyond the training set.

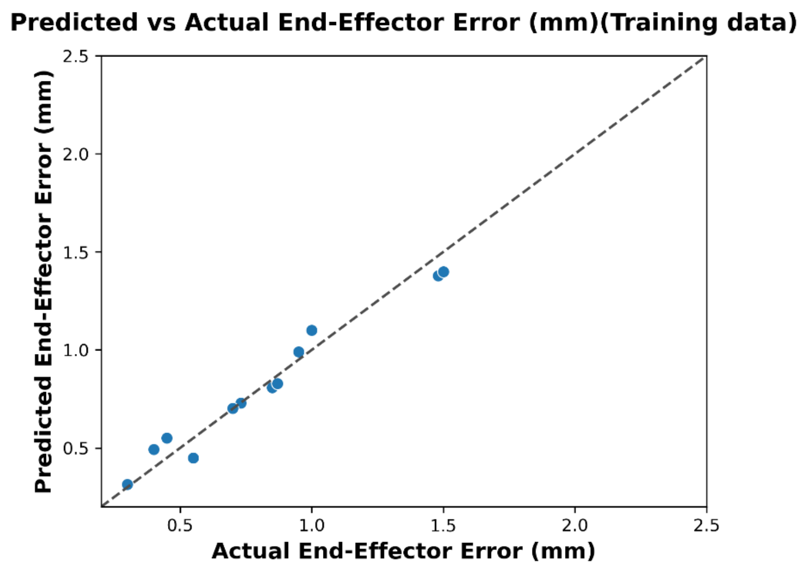


FIGURE 9. Support Vector Regression End-Effector Error (mm) Training

Figure 9 illustrates the performance of a support vector regression (SVR) model on the training data. The data points closely follow the diagonal line, indicating accurate predictions at various end-point error values. Unlike decision trees, SVR uses an edge-based method to help prevent overfitting. The tight clustering of points near the diagonal demonstrates the model's ability to learn effectively, achieving a good balance between bias and variance without having to memorize the training data.

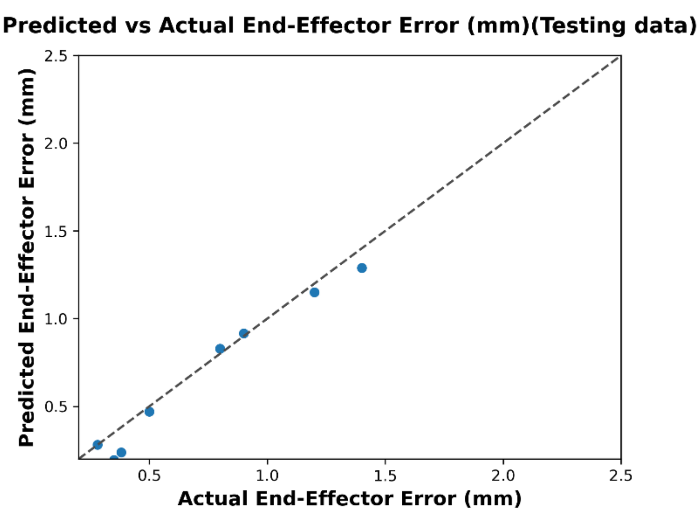


FIGURE 10. Support Vector Regression End-Effector Error (mm) Testing

Figure 10 depicts the performance of the SVR model on the test dataset, where the predicted values closely follow the best-fit diagonal line. This close clustering indicates the model's strong ability to generalize to new, unseen data. The similarity in performance between the training and test sets reinforces the robustness of the model. SVR effectively manages data variability while maintaining high prediction accuracy, making it a reliable choice for precision tasks that demand consistent performance under changing conditions.

4. CONCLUSION

This comprehensive study effectively evaluated four machine learning techniques for predicting end-effector error in robotic systems, providing valuable insights for applications that require high accuracy. The comparison highlights that each approach has unique strengths and limitations, with performance often affected by the complexity of the interactions between data characteristics and operational parameters. Linear regression stood out for its high predictive accuracy and low computational requirements. With R^2 values of 0.9838 for training and 0.9594 for testing, it confirmed largely linear relationships between variables such as motor torque, joint angles, and sensor latency. Its simplicity and interpretability make it suitable for engineering environments where understanding the influence of each parameter is essential for system tuning and optimization. The Multi-Layer Perceptron (MLP) model demonstrated the ability of neural networks to capture complex, nonlinear interactions. Although computationally more demanding, MLP achieved strong performance on both training and test sets, making it a solid choice for more sophisticated robotic systems that incorporate nonlinear behavior. In addition to achieving perfect accuracy on training data, decision tree regression exposed a key challenge in machine learning – overfitting. The drop in test accuracy (R^2 drops to 0.9240) emphasizes the need for rigorous validation and regularization techniques to ensure generalization in robotics-related prediction tasks. Support Vector Regression (SVR) achieved a strong balance between accuracy and generalization, maintaining consistent performance across datasets. Its resilience to noise and ability to effectively handle outliers, environmental variability, and data imperfections make it well-suited for real-world robotics applications where there is typically a high level of uncertainty. This study identified battery voltage as a critical determinant of end-effector accuracy, with a strong negative correlation (-0.95), indicating that higher voltage levels significantly improve position accuracy. This insight has direct implications for power management and maintenance practices. Additionally, positive correlations with motor torque, joint angles, and sensor latency provide practical guidance for improving robotic performance and developing predictive maintenance strategies.

REFERENCE

[1]. Kober, Jens, J. Andrew Bagnell, and Jan Peters. "Reinforcement learning in robotics: A survey." The International Journal of Robotics Research 32, no. 11 (2013): 1238-1274.

[2]. Yang, Guang-Zhong, Jim Bellingham, Pierre E. Dupont, Peer Fischer, Luciano Floridi, Robert Full, Neil Jacobstein et al. "The grand challenges of science robotics." Science robotics 3, no. 14 (2018): eaar7650.

[3]. Samani, Hooman, Elham Saadatian, Natalie Pang, Doros Polydorou, Owen Noel Newton Fernando, Ryohei Nakatsu, and Jeffrey Tzu Kwan Valino Koh. "Cultural robotics: The culture of robotics and robotics in culture." International Journal of Advanced Robotic Systems 10, no. 12 (2013): 400.

[4]. Scaradozzi, David, Laura Sorbi, Anna Pedale, Mariantonietta Valzano, and Cinzia Vergine. "Teaching robotics at the primary school: an innovative approach." Procedia-Social and Behavioral Sciences 174 (2015): 3838-3846.

[5]. Maibaum, Arne, Andreas Bischof, Jannis Hergesell, and Benjamin Lipp. "A critique of robotics in health care." AI & society (2022): 1-11.

[6]. Taylor, Russell H., and Dan Stoianovici. "Medical robotics in computer-integrated surgery." IEEE transactions on robotics and automation 19, no. 5 (2003): 765-781.

[7]. Taylor, Russell H., and Dan Stoianovici. "Medical robotics in computer-integrated surgery." IEEE transactions on robotics and automation 19, no. 5 (2003): 765-781.

[8]. Pettersson, Ola. "Execution monitoring in robotics: A survey." Robotics and Autonomous Systems 53, no. 2 (2005): 73-88.

[9]. Shah, Jay, Arpita Vyas, and Dinesh Vyas. "The history of robotics in surgical specialties." American journal of robotic surgery 1, no. 1 (2014): 12.

[10]. Iqbal, Jamshed, Zeashan Hameed Khan, and Azfar Khalid. "Prospects of robotics in food industry." Food Science and Technology (Campinas) 37, no. 2 (2017): 159-165.

[11]. Hocking, Ronald R. "Developments in linear regression methodology: 1959–l982." Technometrics 25, no. 3 (1983): 219-230.

[12]. Kavitha, S., S. Varuna, and R. Ramya. "A comparative analysis on linear regression and support vector regression." In 2016 online international conference on green engineering and technologies (IC-GET), pp. 1-5. IEEE, 2016.

[13]. Davies, Patrick L. "Aspects of robust linear regression." The Annals of statistics (1993): 1843-1899.

[14]. Marill, Keith A. "Advanced statistics: linear regression, part I: simple linear regression." Academic emergency medicine 11, no. 1 (2004): 87-93.

[15]. Pal, Manoranjan, Premananda Bharati, Manoranjan Pal, and Premananda Bharati. "Introduction to correlation and linear regression analysis." Applications of regression techniques (2019): 1-18.

[16]. Jou, I-Chang, Shih-Shien You, and Long-Wen Chang. "Analysis of hidden nodes for multi-layer perceptron neural networks." Pattern recognition 27, no. 6 (1994): 859-864.

[17]. Turkoglu, Bahaeddin, and Ersin Kaya. "Training multi-layer perceptron with artificial algae algorithm." Engineering Science and Technology, an International Journal 23, no. 6 (2020): 1342-1350.

[18]. Abd-elaziem, Ayman H., and Tamer HM Soliman. "A multi-layer perceptron (mlp) neural networks for stellar classification: A review of methods and results." International Journal of Advances in Applied Computational Intelligence 3, no. 10.54216 (2023).

[19]. Pratama, Mahardhika, Choiru Za'in, Andri Ashfahani, Yew Soon Ong, and Weiping Ding. "Automatic construction of multi-layer perceptron network from streaming examples." In Proceedings of the 28th ACM international conference on information and knowledge management, pp. 1171-1180. 2019.
[20]. Pratama, Mahardhika, Choiru Za'in, Andri Ashfahani, Yew Soon Ong, and Weiping Ding. "Automatic construction of multi-layer perceptron network from streaming examples." In Proceedings of the 28th ACM international conference on information and knowledge management, pp. 1171-1180. 2019.
[21]. Kim, Kyoungok, and Jung-sik Hong. "A hybrid decision tree algorithm for mixed numeric and categorical data in regression analysis." Pattern Recognition Letters 98 (2017): 39-45.
[22]. Chen, Mu-Ming, and Mu-Chen Chen. "Modeling road accident severity with comparisons of logistic regression, decision tree and random forest." Information 11, no. 5 (2020): 270.
[23]. Zhang, Chen, Andrew Kotz, Kenneth Kelly, and Luke Rippelmeyer. "Development of heavy-duty vehicle representative driving cycles via decision tree regression." Transportation Research Part D: Transport and Environment 95 (2021): 102843.
[24]. Arista, Artika. "Comparison Decision Tree and Logistic Regression Machine Learning Classification Algorithms to determine Covid-19." Sinkron: jurnal dan penelitian teknik informatika 6, no. 1 (2021): 59-65.
[25]. Tamir, Tariku Sinshaw, Gang Xiong, Zhishuai Li, Hao Tao, Zhen Shen, Bin Hu, and Heruye Mulugeta Menkir. "Traffic congestion prediction using decision tree, logistic regression and neural networks." Ifac-PapersOnline 53, no. 5 (2020): 512-517.
[26]. Paniagua-Tineo, A., S. Salcedo-Sanz, C. Casanova-Mateo, E. G. Ortiz-García, M. A. Cony, and E. Hernández-Martín. "Prediction of daily maximum temperature using a support vector regression algorithm." Renewable Energy 36, no. 11 (2011): 3054-3060.
[27]. Long, Nanye, Daniel Gianola, Guilherme JM Rosa, and Kent A. Weigel. "Application of support vector regression to genome-assisted prediction of quantitative traits." Theoretical and applied genetics 123 (2011): 1065-1074.
[28]. Cheng, Kai, Zhenzhou Lu, Yicheng Zhou, Yan Shi, and Yuhao Wei. "Global sensitivity analysis using support vector regression." Applied Mathematical Modelling 49 (2017): 587-598.
[29]. Quan, Quan, Zou Hao, Huang Xifeng, and Lei Jingchun. "Research on water temperature prediction based on improved support vector regression." Neural Computing and Applications (2022): 1-10.
[30]. Luo, Xiangang, Xiaohui Yuan, Shuang Zhu, Zhanya Xu, Lingsheng Meng, and Jing Peng. "A hybrid support vector regression framework for streamflow forecast." Journal of Hydrology 568 (2019): 184-193.

