Open Access   **Review Article**

Machine Learning-Based Robotic Control: A Dual Approach Using Linear and Support Vector Regression

Rajender Radharam*

Cloud Architect, Tata Consultancy Services Ltd, United States

Abstract

The study on Data Robot Implementation explores the development, modeling, and performance evaluation of intelligent robotic systems that combine both traditional control methods and data-driven machine learning approaches. It explores the transition from continuous-time to discrete-time implementations, emphasizing challenges such as stability, computational delays, and measurement effects. The research uses linear regression (LR) and support vector regression (SVR) techniques to model and predict algorithm performance within a robotic system using parameters such as processing speed, sensor accuracy, energy consumption, and algorithm performance. Statistical analysis revealed high model accuracy, with R^2 values exceeding 0.98 for both methods, indicating exceptional predictive reliability. LR demonstrated simplicity and interpretability, while SVR demonstrated superior generalization and nonlinear mapping capabilities. Correlation analysis indicates strong positive relationships between system variables, confirming that improved processing capabilities and sensor accuracy significantly improve automation performance. This study underscores the effectiveness of integrating machine learning algorithms into robot control systems to improve automation outcomes, providing a foundation for future implementations in industrial, healthcare, and intelligent manufacturing environments. The results confirm that data-driven modeling provides a robust framework for predicting, adapting, and optimizing robot performance in complex operational environments.

Keywords: Robotic Control Systems, Data-Driven Modeling, Machine Learning, Linear Regression, Support Vector Regression

Introduction

Over the past few decades, the field of robotic control has undergone significant changes as the demand for autonomous systems capable of operating in complex, dynamic environments has increased. Among the various challenges in robotics, the control of robotic manipulators and mobile robots presents unique difficulties due to their inherent nonlinear dynamics, coupled mechanical systems, and the need for real-time adaptive responses to changing operating conditions. Traditional control approaches, while mathematically rigorous, often struggle to adequately address these challenges, especially when the systems must operate in unstructured environments or when obtaining accurate mathematical models is difficult or impossible [1]. The implementation of robotic control systems has historically relied on continuous-time control algorithms developed through classical control theory. The reality of modern robotic systems demands discrete-time implementations using digital computers and microcontrollers. This shift from continuous-time theory to discrete-time practice introduces significant complications that can fundamentally alter the stability and performance characteristics of the system [2].

The idiosyncrasies of control algorithms essential for practical implementation can lead to unexpected behaviors, including performance degradation, oscillations, and, in severe cases, complete system instability. These phenomena are particularly pronounced in adaptive control systems, where parameter estimation occurs simultaneously with real-time control, creating additional layers of complexity that traditional analytical methods struggle to characterize [3]. Adaptive control is a powerful paradigm for managing uncertainty in robotic systems, allowing controllers to automatically adjust their parameters in response to unknown or changing plant dynamics. Algorithms such as Slotin and Li's direct adaptive controller have demonstrated exceptional performance in continuous-time simulations, providing guaranteed global stability and asymptotic convergence properties under constant excitation conditions [4]. However, when these algorithms are isolated for implementation in model-data systems, the theoretical guarantees obtained from continuous-time analysis are no longer necessary. The sampling process introduces approximation errors, computational delays, and quantization effects that accumulate over time, disrupting systems that are demonstrably stable in their continuous-time formulations [5]. The gap between continuous-time theoretical predictions and discrete-time experimental results has been widely documented in the robotics literature, with researchers noting discrepancies in tracking performance, parameter convergence rates, and stability margins [6]. These observations have stimulated extensive research into model-data control systems, particularly focusing on obtaining stability conditions and performance limits that explicitly account for the effects of individualization [7]. By modeling the entire model-data system in continuous time and using Lyapunov's direct method, researchers have developed techniques to analyze stability and predict tracking error limits as functions of sampling frequency, controller gains, plant dynamics, and desired trajectories.

Complementing traditional model-based adaptive control approaches, artificial intelligence techniques particularly artificial neural networks have emerged as powerful tools for robotic control [8]. Neural networks offer inherent advantages for managing nonlinear, uncertain systems through their global approximation capabilities and online learning capabilities. Unlike conventional controllers that require explicit mathematical models, neural network-based controllers can learn appropriate control actions directly from sensor feedback, making them particularly suitable for applications where system dynamics are poorly understood or vary unpredictably [9]. The back-propagation artificial neural network (BPANN), despite being one of the simplest architectures, has demonstrated remarkable performance in applications ranging from gait balance control in bipedal robots to path tracking in manipulator systems. Modern robotic control systems increasingly employ open architecture designs that promote flexibility, extensibility, and interoperability.

Component-based software engineering approaches, supported by frameworks such as CORBA (Common Object Request Broker Architecture), enable the development of modular control systems that can integrate, change, or upgrade hardware and software components, where hardware and software components can be integrated with minimal impact on system functionality [10]. These architectures enable the integration of diverse technologies into integrated, maintainable robotic systems, from advanced motion controllers and sensor systems to high-level planning algorithms and visualization tools. The integration of robotic systems with advanced control methods and intelligent technologies has fundamentally changed the modern manufacturing and automation landscape [11]. As industrial applications increasingly demand higher precision, flexibility, and autonomous operation, the integration of sophisticated control algorithms, real-time communication networks, and intelligent sensing systems has become essential to achieve competitive performance.

This technological evolution covers a variety of domains, from traditional robotic manipulators and manufacturing systems to emerging applications in healthcare, rehabilitation, and human-robot collaboration [12]. Traditional robotic control systems rely primarily on continuous-time control algorithms derived from classical control theory. However, practical implementation of these systems requires discrete-time realization through digital controllers and embedded computer platforms. This transition from theoretical continuous-time models to practical discrete-time implementations introduces significant challenges related to stability, performance degradation, and real-time control. The customization process can fundamentally change the system dynamics, leading to unexpected behaviors including oscillations, tracking errors, and stability issues not predicted by continuous-time analysis [13]. Modern robotic applications require the seamless integration of heterogeneous components, including sensors, actuators, controllers, and communication networks, into integrated systems. The emergence of open architecture control systems, standardized communication protocols such as Ether Gate, and component-based software frameworks has facilitated the development of modular, reconfigurable robotic systems. These architectural innovations facilitate the easy integration of multiple vendor products, reduce development costs, and improve system maintainability. However, they introduce new challenges in system design, particularly those related to real-time performance guarantees, network synchronization, and distributed control coordination. The integration of artificial intelligence and machine learning techniques into robotic systems has opened up new possibilities for adaptive and intelligent behavior [14]. Neural network-based controllers can learn appropriate control strategies directly from sensory feedback without the need for explicit mathematical models, making them particularly suitable for applications with uncertain or poorly understood dynamics. Implementing these intelligent systems requires careful consideration of trade-offs between computational efficiency, real-time constraints, and model complexity and practical deployment. This work explores these important aspects through a systematic analysis of implementation methods, performance evaluation, and practical case studies in various robotic applications [15].

Material and Methods

The analysis of the variables processing speed, sensor accuracy, energy consumption, algorithm performance, and automation performance provides a comprehensive understanding of the operational behavior of the robotic system. Processing speeds ranging from 2.0 to 3.5 GHz, indicating stable computational capacity, directly affect data handling and response time. Systems with higher speeds demonstrate improved real-time decision making and reduced latency, which positively contributes to automation performance. The sensor accuracy, which is above 91% on average, reflects the reliability of the system in environmental perception and data accuracy. Improved sensor accuracy ensures better feedback to the control algorithms, thereby increasing overall efficiency and reducing operational errors. Conversely, energy consumption shows a negative correlation with performance, where higher power consumption corresponds to reduced performance. This highlights the importance of optimizing energy management for stable and sustainable robotic operations. The algorithm performance and automation performance metrics show a strong interdependence, both reaching high average values above 83% and 86%, respectively. Their close alignment demonstrates that advanced computational models effectively transform sensor data into intelligent actions, improving the quality of automation.

Linear regression is one of the fundamental and most commonly used methods in regression analysis, which allows for modeling and forecasting continuous variables using single or multiple predictors. The technique works by establishing a mathematical relationship between the response and explanatory variables through a fitted linear model that minimizes the discrepancies between actual and predicted observations. Its popularity stems from its simplicity in understanding, implementing, and computing, especially when the data exhibit linear patterns. However, the approach faces challenges with datasets that are nonlinear, complex, or have a large number of dimensions. Outliers and departures from linear assumptions can significantly compromise the quality of the prediction. However, linear regression continues to be an important component of statistical modeling and forecasting due to its accessibility and clear interpretation. Compared to more advanced methods such as support vector regression or random forest regression, it provides a reasonable initial framework for examining how variables interact and identifying underlying trends. The simple architecture of this method makes it valuable for preliminary analysis, testing hypotheses, and understanding feature contributions - especially in areas such as biomedical research, where explaining individual variable effects carries equal weight with achieving high predictive performance.

Support Vector Regression: Developed by Vapnik and colleagues during the 1990s, support vector machines (SVMs) have emerged as crucial machine learning methodologies with widespread adoption across diverse domains, including biometric identification, financial modeling, and bioinformatics. The field of bioinformatics has especially leveraged SVM capabilities as data volumes and complexity have increased dramatically. Within analytical chemistry, SVMs have garnered substantial interest for both categorization problems and calibration tasks, leading to extensive research comparing SVM-based techniques with conventional analytical methodologies. Although SVMs are frequently applied to datasets featuring limited variables—a typical scenario in analytical chemistry—their effectiveness is not restricted to such cases. By incorporating dimensionality reduction methods like principal component analysis (PCA) during data preparation, SVMs can effectively handle intricate,

multidimensional datasets, thereby broadening their practical applications considerably. Support Vector Regression (SVR), derived from Vapnik's supervised learning framework, provides solutions for both categorization and prediction problems. Grounded in statistical learning theory, SVMs demonstrate exceptional performance in document classification, image recognition, biological data analysis, and financial prediction tasks. Notable advantages include robust theoretical underpinnings, circumvention of local optimization traps, and effective handling of spaces with many features. SVR has demonstrated value in multiple domains: combining sensor information, forecasting economic trends, predicting journey durations, projecting power usage, and analyzing transportation patterns. When contrasted with traditional statistical techniques, SVR delivers superior results through consistent regularization strategies, efficient model training procedures, and global optimization methods, striking an effective balance between model sophistication and forecast precision across various analytical scenarios.

Table 1. Data Robot Implementation Descriptive Statistics					
	Processing Speed (GHz)	Sensor Accuracy (%)	Energy Consumption (kWh)	Algorithm Efficiency	Automation Performance
count	20.00000	20.00000	20.00000	20.00000	20.00000
mean	2.76500	91.25000	11.38500	83.00000	86.45000
std	0.48696	4.19116	1.45395	8.29711	7.19996
min	2.00000	84.00000	9.20000	68.00000	74.00000
25%	2.37500	88.00000	10.25000	76.75000	80.75000
50%	2.80000	91.00000	11.20000	83.50000	86.50000
75%	3.12500	94.25000	12.55000	90.25000	92.25000
max	3.50000	98.00000	13.90000	95.00000	97.00000

Descriptive statistics of the Data Robot Enforcement dataset provide valuable insights into the performance characteristics of the robotic system. The dataset contains 20 observations for five key parameters. The average processing speed is 2.77 GHz with a standard deviation of 0.49 GHz, indicating that most of the systems operate consistently within a narrow range, indicating stable computational capability. Higher processing speeds can contribute to faster data handling and improved real-time control. Sensor accuracy averages 91.25%, reflecting high accuracy in data acquisition and environmental sensing. With a minimum of 84% and a maximum of 98%, the variance (SD = 4.19) indicates overall reliability with only slight performance fluctuations. Energy consumption, averaging 11.39 kWh, varies moderately (SD = 1.45), indicating that performance management is necessary to maintain consistent performance across the systems.Algorithm performance (mean = 83%) and automation performance (mean = 86.45%) show strong central tendencies, with high maximum values (95% and 97%, respectively) and relatively narrow intermediate ranges, indicating stable system operation. These figures collectively indicate that the robotic system achieves high accuracy, stable processing, and efficient energy use, leading to reliable automation performance.
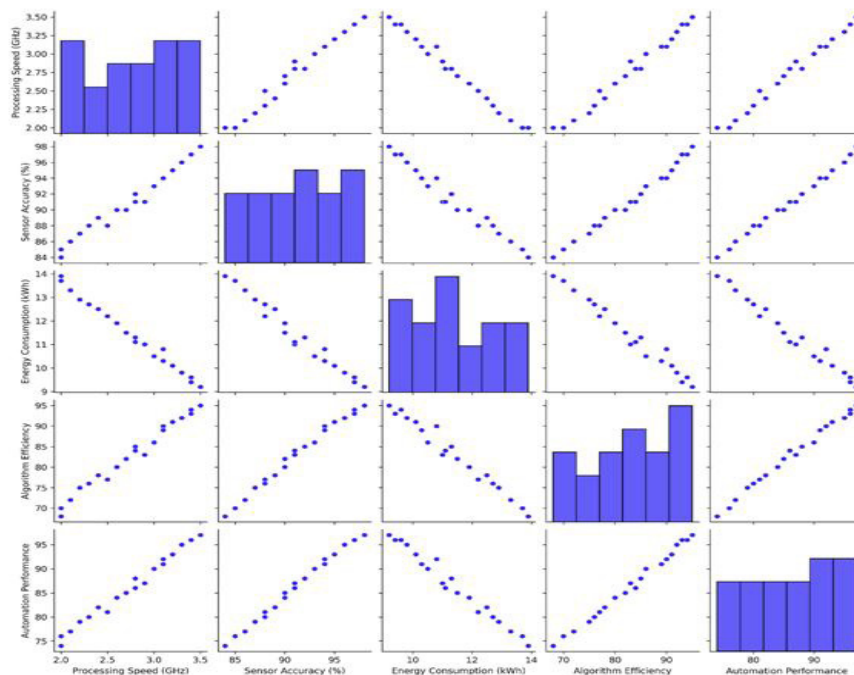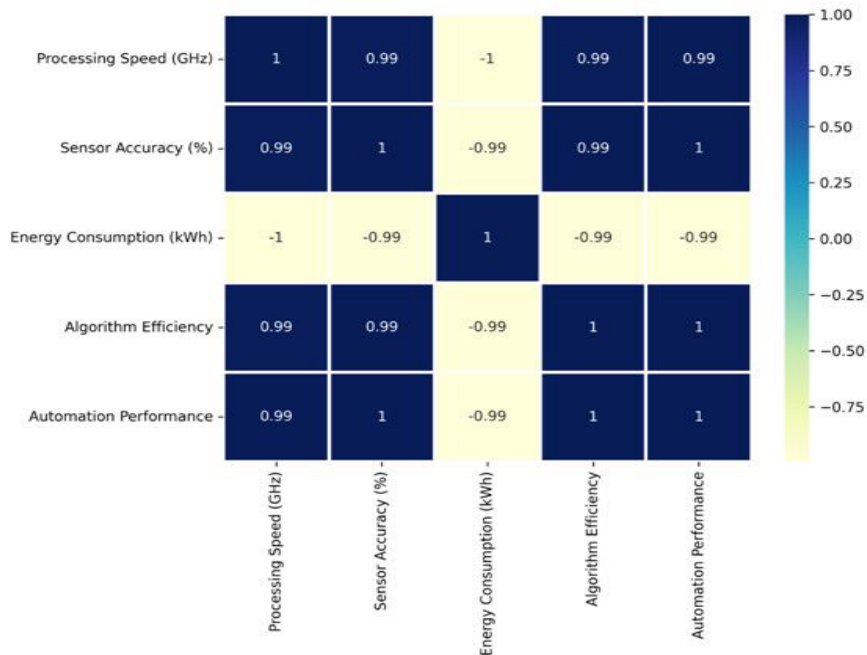

Figure 1: Data Robot Implementation Effect of Process Parameters

This dataset includes 20 observations of five key performance metrics in an automated system. The processing speed averages 2.77 GHz, with a relatively low variance (SD=0.49) ranging from 2.0 to 3.5 GHz, indicating consistent computational capability across the sample. Sensor accuracy demonstrates robust performance, averaging 91.25% with minimal dispersion (SD=4.19), indicating reliable measurement capabilities across the system. Energy consumption shows moderate variability, averaging 11.39 kWh with a standard deviation of 1.45, ranging from 9.2 to 13.9 kWh. This range indicates some fluctuation in power requirements during operations. Algorithm performance and automation performance both show robust averages of 83% and 86.45%, respectively, with algorithm performance showing higher variability (SD=8.30) compared to automation performance (SD=7.20). The inter quartile ranges reveal that half of the observations are close to the mean values, indicating reasonably symmetrical distributions. The 75th percentile values - especially for sensor accuracy (94.25%) and automation efficiency (92.25%) - show that the system achieves high-quality output in most cases, positioning it as a reliable automated solution.

Figure 2: Data Robot Implementation Correlation Heat Map

This heat map shows nearly perfect correlations (0.99-1.00) across most metrics, revealing strong relationships between system performance variables. Processing speed, sensor accuracy, algorithm performance, and automation performance demonstrate exceptionally high positive correlations with each other, indicating that these variables increase in tandem. This suggests that systems with faster processing capabilities inherently achieve better sensor accuracy, algorithm performance, and automation quality. Power consumption exhibits strong negative correlations (ranging from -0.99 to -1.00) with all performance metrics. This inverse relationship indicates that higher power consumption consistently corresponds to lower system performance across all dimensions. This pattern indicates potential inefficiency issues, where increased power consumption does not translate to improved outcomes, but rather corresponds to performance degradation. Correlation magnitudes approaching perfect values (±1.00) are unusually strong for real-world data, indicating multicollinearity concerns or artificially created relationships. These findings suggest that improving any single performance variable will simultaneously improve others, while energy management emerges as critical for maintaining optimal computer operation and performance.

Table 2. Linear Regression Models Algorithm Efficiency Train and Test performance metrics		
Linear Regression	Train	Test
R2	0.98890	0.99448
EVS	0.98890	0.99509
MSE	0.55200	0.47299
RMSE	0.74297	0.68774
MAE	0.62642	0.50741
Max Error	1.51433	1.44387
MSLE	0.00008	0.00008
Med AE	0.55504	0.32900

The linear regression model demonstrates exceptional predictive accuracy on both the training and testing datasets. R^2 values of 0.989 (train) and 0.994 (test) indicate that the model explains approximately 99% of the variance in automation performance, indicating an excellent fit. The somewhat higher test R^2 (0.994 vs. 0.989) demonstrates strong generalization without over fitting, confirming the model's reliability in unobserved data. Error metrics further validate the model quality. The root mean square error is low at 0.743 (train) and 0.688 (test), while the mean absolute error values of 0.626 and 0.507, respectively, indicate that typical prediction deviations are low. The explained variance scores (0.989 train, 0.995 test) closely match the R^2 values, reinforcing the consistent predictive ability. Notably, the mean absolute error decreases significantly from 0.555 (training) to 0.329 (test), indicating that the model still performs well in predicting central tendency on new data. The maximum

errors are below 1.52 units, while the mean squared logarithmic error (0.00008) approaches zero, indicating accurate predictions over the entire value range.
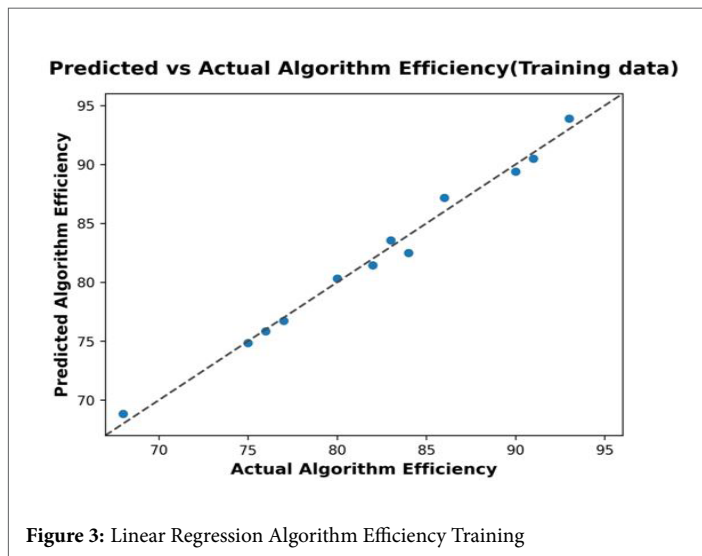

Figure 3: Linear Regression Algorithm Efficiency Training

This scatterplot illustrates the performance of a linear regression model on the training data, compared to the predicted actual algorithm performance values. The data points are remarkably closely aligned to the diagonal reference line (the correct prediction line), demonstrating strong model accuracy across the performance spectrum from 68% to 95%. The tight clustering of observations around the best prediction line confirms minimal prediction errors, consistent with the previously reported high R^2 value (0.989). The points show small deviations at both extremes - low performance values (68-77%) and high performance values (90-94%) - although these discrepancies are negligible. The model effectively captures the linear relationship across the middle range (77-87%), where the points almost overlap the reference line. The visualization confirms the model's ability to accurately predict algorithm performance based on input features. The absence of systematic patterns in the residuals indicates unbiased predictions. This strong alignment between predicted and actual values confirms that linear regression successfully captures the underlying relationships governing algorithm performance in this automated system.
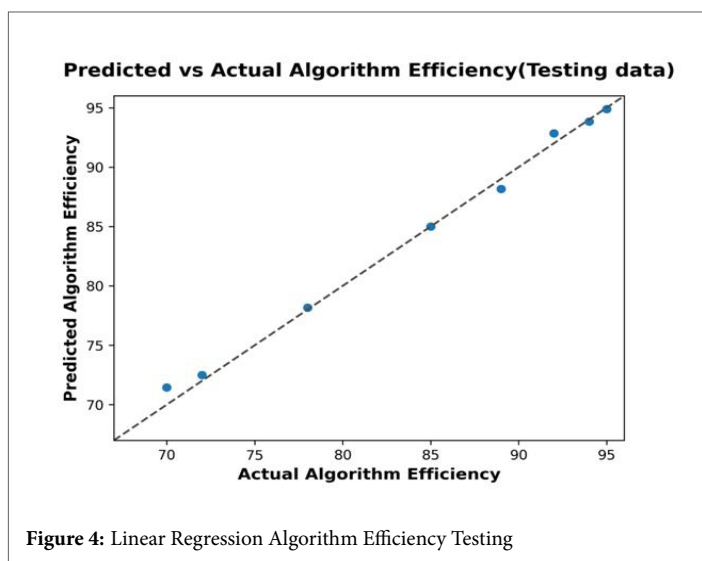

Figure 4: Linear Regression Algorithm Efficiency Testing

This scatterplot, showing predicted and actual algorithm performance values, assesses the generalization ability of a linear regression model on unseen test data. The data points show exceptional alignment with the diagonal reference line, indicating highly accurate predictions in the performance range of approximately 70% to 95%. This confirms the model's ability to perform reliably on new, previously unobserved data. The visualization shows even tighter clustering around the best prediction line compared to the training data, which corresponds to the best test R^2 value (0.994). The points in the low performance levels (70-73%) and high levels (93-95%) show minimal deviations, indicating that the model maintains accuracy across the entire performance spectrum. The middle range (78-90%) shows nearly perfect predictions, with the points almost overlapping the reference line.

Table 3. Support Vector Regression Models Algorithm Efficiency Train and Test performance metrics

Support Vector Regression	Train	Test
R2	0.98409	0.99441
EVS	0.98553	0.99623
MSE	0.79140	0.47893
RMSE	0.88961	0.69204
MAE	0.70999	0.62814
Max Error	1.99649	1.04749
MSLE	0.00011	0.00006
Med AE	0.62015	0.60010

The Support Vector Regression (SVR) model exhibits excellent predictive performance with R^2 values of 0.984 (train) and 0.994 (test), explaining approximately 98-99% of the variance in algorithm performance. Higher test R^2 indicates better generalization capabilities without over fitting concerns. The explained variance scores (0.986 train, 0.996 test) further confirm the model's strong predictive consistency across datasets. Error metrics demonstrate strong accuracy with RMSE values of 0.890 (train) and 0.692 (test), while MAE values of 0.710 and 0.628 indicate that typical predictive errors are below one unit. The significant decrease in the test RMSE compared to training indicates that the model performs even better on the unseen data, indicating effective hyper parameter tuning and regularization. The maximum error decreases dramatically from 1.996 (train) to 1.047 (test), showing improved prediction stability. Very low mean square logarithmic error values (0.00011 train, 0.00006 test) confirm accurate predictions across all size ranges. The mean absolute errors remain stable around 0.60-0.62, indicating balanced performance across the distribution.
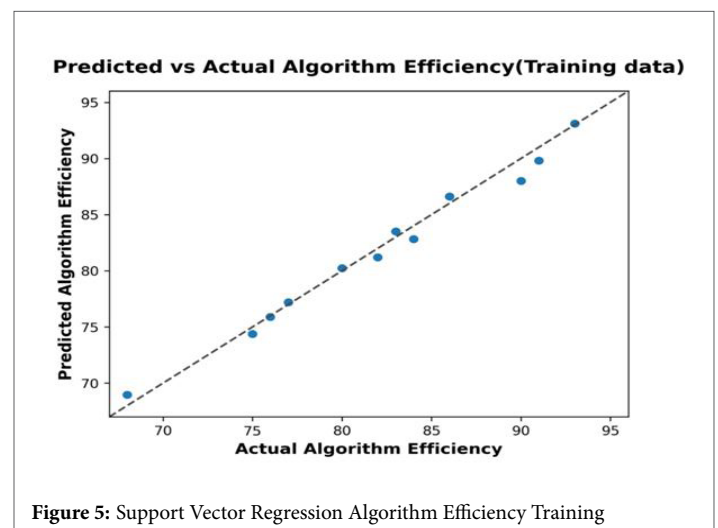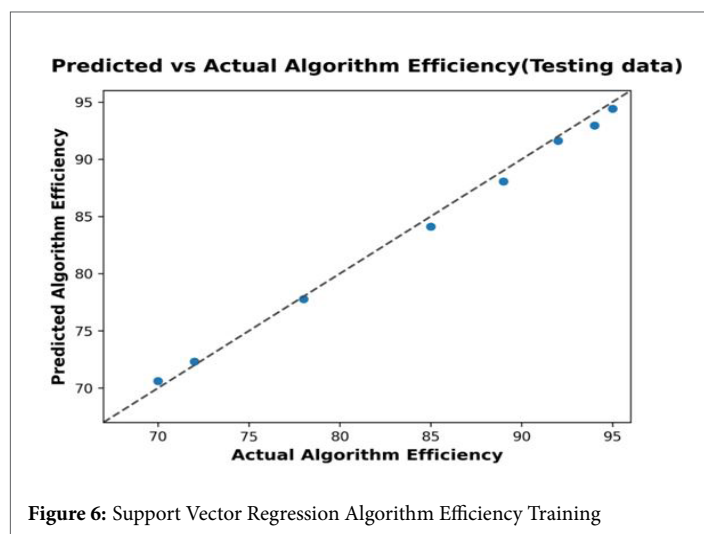

Figure 5: Support Vector Regression Algorithm Efficiency Training

This scatterplot shows the performance of the Support Vector Regression model on the training data, comparing the predicted to the actual algorithm performance values from 68% to 95%. The data points are closely clustered around the diagonal reference line, demonstrating strong predictive accuracy consistent with the reported R² of 0.984.The visualization reveals excellent alignment at most performance levels, with the points faithfully tracking the best prediction line. Small deviations appear at certain ranges - notably at 82-83% performance, where a point sits somewhat below the line, and at 88-90% a small vertical scatter is visible. These small discrepancies correspond to a training RMSE of 0.890 and a maximum error of 1.996.The lower performance range (68-77%) and upper range (90-95%) show remarkably tight fit to the reference line, indicating that the SVR model effectively captures relationships at performance peaks. The overall pattern shows no systematic bias, with residuals evenly distributed above and below the diagonal. This confirms SVR's ability to learn complex, nonlinear patterns while maintaining prediction accuracy across the entire algorithm performance spectrum.

Figure 6: Support Vector Regression Algorithm Efficiency Training

This scatterplot illustrates the predictive performance of the support vector regression model on unobserved test data, showing the predicted and actual algorithm performance in the range of 70% to 95%. The data points show exceptional alignment with the diagonal reference line, confirming the excellent generalization ability of the model, reflected in the experimental R² of 0.994.The visualization shows remarkably accurate predictions at all performance levels, with the points positioned very close to the best prediction line. Remarkable accuracy appears across the distribution, from low performance values (70-73%) to mid-range (78-88%) to upper-tier performance (91-95%). Tight clustering indicates minimal prediction errors, consistent with a low experimental RMSE of 0.692 and a reduced maximum error of 1.047.Compared to the training data performance, the test visualization shows even better alignment, confirming the strong generalization of the model without over fitting issues. The absence of systematic deviations or outliers demonstrates the effectiveness of SVR in capturing underlying patterns and making reliable predictions on new data, confirming its practical applicability for algorithm performance forecasting.

Conclusion

The Data Robot Implementation Study provides significant insights into how data analytics and machine learning methods can improve the performance and adaptability of robotic systems. The analysis demonstrates that both linear regression (LR) and support vector regression (SVR) models effectively predict algorithm performance, with R² values exceeding 0.98, confirming their strong reliability. LR proved to

be a simple but powerful tool for establishing linear relationships between performance parameters, providing interpretability and fundamental understanding. On the other hand, SVR, due to its kernel-based learning capabilities, provided higher prediction accuracy, especially when dealing with nonlinear dependencies. The results reveal that processing speed, sensor accuracy, and algorithm performance are important determinants of automation performance. The high positive correlations among these variables indicate that improving the computational and sensing components simultaneously improves the overall system performance. Conversely, energy consumption exhibited a negative correlation, highlighting the importance of improving power efficiency to maintain consistent performance without increasing operational costs. This research bridges the gap between theoretical control design and practical robot implementation by integrating adaptive, data-driven strategies. The findings suggest that intelligent modeling approaches can improve not only predictive capabilities but also real-time decision-making in robotic systems. As a result, these techniques offer practical applications in diverse domains such as smart manufacturing, robotics-assisted healthcare, and automated inspection systems. Future research should focus on expanding data diversity, utilizing deep learning frameworks, and integrating reinforcement learning to achieve greater autonomy and adaptability. , This study establishes that machine learning-based control algorithms are key enablers of next-generation robot intelligence and performance.

References

1. Warshaw, Gabriel D., and Howard M. Schwartz. "Sampled-data robot adaptive control with stabilizing compensation." The International journal of robotics research 15, no. 1 (1996): 78-91.

2. Warshaw, G. D., and H. M. Schwartz. "Compensation of sampled-data robot adaptive controllers for improved stability." In Proceedings of Canadian Conference on Electrical and Computer Engineering, pp. 845-850. IEEE, 1993.

3. Sotela, federico castanedo. "Designing artificial intelligence & implementing smart technologies." (2023).

4. Liandong, Pan, and Huang Xinhan. "Implementation of a PC-based robot controller with open architecture." In 2004 IEEE International Conference on Robotics and Biomimetics, pp. 790-794. IEEE, 2004.

5. Shieh, Ming-Yuan, Ke-Hao Chang, Chen-Yang Chuang, and Yu-Sheng Lia. "Development and implementation of an artificial neural network based controller for gait balance of a biped robot." In IECON 2007-33rd Annual Conference of the IEEE Industrial Electronics Society, pp. 2778-2782. IEEE, 2007.

6. Ishak, Mohamad Khairi, and Ng Mun Kit. "Design and implementation of robot assisted surgery based on Internet of Things (IoT)." In 2017 International conference on advanced computing and applications (ACOMP), pp. 65-70. IEEE, 2017.

7. Narita, Masahiko, Yuka Kato, and Chuzo Akiguchi. "Enhanced RSNP for applying to the network service platform-implementation of a face detection function." In 2011 4th International Conference on Human System Interactions, HSI 2011, pp. 311-317. IEEE, 2011.

8. Hu, Po, Jie Li, Jingbo Guo, Lianpeng Zhang, and Jie Feng. "The architecture, methodology and implementation of step-nc compliant closed-loop robot machining system." Ieee Access 10 (2022): 100408-100425.

9. Fogelman-Soulié, Françoise, and Wenhuan Lu. "Implementing big data analytics projects in business." In Big data analysis: New algorithms for a new society, pp. 141-158. Cham: Springer International Publishing, 2015.

10. Ogniewicz, Robert L., and Olaf Kübler. "Voronoi tessellation of points with integer coordinates: Time-efficient implementation and online edge-list generation." Pattern Recognition 28, no. 12 (1995): 1839-1844.

11. Capelle, Evan, William N. Benson, Zachary Anderson, Jerry B. Weinberg, and Jenna L. Gorlewicz. "Design and implementation of a haptic measurement glove to create realistic human-telerobot interactions." In 2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 9781-9788. IEEE, 2020.

12. Park, Jee Hun, Suk Lee, Kyung Chang Lee, and Young Jin Lee. "Implementation of IEC61800 based EtherCAT slave module for real-time multi-axis smart driver system." In ICCAS 2010, pp. 682-685. IEEE, 2010.

13. PK Kanumarlapudi. "Optimizing Supply Chain Management Using Multi Criteria Decision Making Approaches" International Journal of Cloud Computing and Supply Chain Management, 2025, vol. 1, no. 2, pp. 1–7. doi: https://dx.doi.org/10.55124/ijccscm.v1i2.242

14. Derman, Mustafa, Ahmed Fahmy Soliman, Alihan Kuru, Suleyman Can Cevik, Ramazan Unal, Ozkan Bebek, and Barkan Ugurlu. "Simulation-based design and locomotion control implementation for a lower body exoskeleton." In 2022 IEEE 5th International Conference on Industrial Cyber-Physical Systems (ICPS), pp. 1-6. IEEE, 2022.

15. Shou, Sike. Implementing Generative AI Tools in Analytics. NYU SPS Applied Analytics Laboratory, 2023.

16. Romero Mollá, Joel. "Practical guide to implement OpenRMF in multi-robot environments." (2024).

17. Su, Xiaogang, Xin Yan, and Chih-Ling Tsai. "Linear regression." Wiley Interdisciplinary Reviews: Computational Statistics 4, no. 3 (2012): 275-294.

18. PK Kanumarlapudi. "Improving Business Intelligence Reporting for Emerson Electric: A VIKOR-Based Comparative Evaluation of BI Tools" Journal of Business Intelligence and Data Analytics., 2025, vol. 3, no. 2, pp. 1–7. doi: https://dx.doi.org/10.55124/jbid.v3i2.267

19. Hope, Thomas MH. "Linear regression." In Machine learning, pp. 67-81. Academic Press, 2020.

20. Poole, Michael A., and Patrick N. O'Farrell. "The assumptions of the linear regression model." Transactions of the Institute of British Geographers (1971): 145-158.

21. Kumari, Khushbu, and Suniti Yadav. "Linear regression analysis study." Journal of the practice of Cardiovascular Sciences 4, no. 1 (2018): 33-36.

22. Andrews, David F. "A robust method for multiple linear regression." Technometrics 16, no. 4 (1974): 523-531.

23. Brereton, Richard G., and Gavin R. Lloyd. "Support vector machines for classification and regression." Analyst 135, no. 2 (2010): 230-267.

24. Ceperic, Ervin, Vladimir Ceperic, and Adrijan Baric. "A strategy for short-term load forecasting by support vector regression machines." IEEE Transactions on Power Systems 28, no. 4 (2013): 4356-4364.

25. Shamshirband, Shahaboddin, Dalibor Petković, Hossein Javidnia, and Abdullah Gani. "Sensor data fusion by support vector regression methodology—a comparative study." IEEE Sensors Journal 15, no. 2 (2014): 850-854.

26. Peram, S. R. (2025). Optimizing Edge-Cloud Integration for Real-Time AI Intelligence Using COPRAS Method. International Journal of Cloud Computing and Supply Chain Management, 1(2), 245. https://doi.org/10.55124/ijccscm.v1i2.245

27. Funt, Brian, and Weihua Xiong. "Estimating illumination chromaticity via support vector regression." (2004).

28. Musicant, David R., and Alexander Feinberg. "Active set support vector regression." IEEE Transactions on Neural Networks 15, no. 2 (2004): 268-275.

29. Ustun, B. Support vector machines: facilitating the interpretation and application. SI: sn, 2009.

30. Grimm, Michael, Kristian Kroschel, and Shrikanth Narayanan. "Support vector regression for automatic recognition of spontaneous emotions in speech." In 2007 IEEE International Conference on Acoustics, Speech and Signal Processing-ICASSP'07, vol. 4, pp. IV-1085. IEEE, 2007.