ARTICLE

doi: https://dx.doi.org/10.55124/csdb.v2i2.249

SciForce

International Journal of Computer Science and Data Engineering

Journal homepage: www.sciforce.org

Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making

Akhilesh Reddy Eppa*

*Sr. AI Engineer, Master's in Big Data Systems at Arizona State University, Maryland, USA

ARTICLE INFO

Article history:
Received:20250303
Received in revised form: 20250305
Accepted20250308
Available online: 20250312

Keywords:
Reinforcement Learning;
Autonomous Systems;
Machine Learning;
Sensor Accuracy;
Processing Power;
Training Episodes;
Average Reward.

ABSTRACT

Reinforcement Learning (RL), a powerful paradigm for decision-making in autonomous systems, has enabled agents to learn the optimum policies through trial and error in dynamic environments. This paper explores the integration of RL within autonomous systems, emphasizing how various machine learning algorithms Linear Regression (LR), Random Forest Regression (RFR), and Support Vector Machines (SVM) can assist in predicting agent performance and optimizing training processes. These algorithms are employed to analyze key input parameters, including sensor accuracy (%), processing power (GHz), and the number of training episodes (#), to determine their influence on the agent's learning efficiency and overall performance. The effectiveness of an RL agent is often measured using an evaluation metric such as the average reward (R), which quantifies the long-term benefits obtained by following a learned policy. By leveraging predictive modeling techniques, we aim to establish correlations between input parameters and RL performance, helping to refine system design and resource allocation. Sensor accuracy has a direct influence on decision-making processes and is essential in assessing the dependability of state information.. Processing power influences the speed and complexity of model updates, affecting convergence rates. The number of training episodes determines the agent's exposure to various environmental states, influencing its ability to generalize learned behaviors.

This study employs a hybrid approach where RL agents are trained in simulated environments, and machine learning models are used to analyze performance trends. LR provides a simple yet interpretable linear relationship between parameters and average reward, while RFR captures complex nonlinear interactions and enhances prediction robustness. SVM, known for its strong generalization capabilities, further refines decision boundaries in high-dimensional spaces. By comparing these approaches, we derive insights into which factors most significantly impact RL performance and how predictive models can be leveraged to improve autonomous system efficiency. The findings show that the average reward is significantly impacted nonlinearly by sensor accuracy., highlighting the need for high-fidelity sensing in autonomous applications. Processing power influences real-time adaptability, while an optimal number of training episodes ensures sufficient learning without excessive computational overhead. The findings demonstrate that integrating supervised learning techniques with RL not only aids in understanding system behavior but also provides a foundation for adaptive optimization strategies in real-world applications. Future research will concentrate on extending these techniques to more complex, multi-agent environments and exploring meta-learning approaches for enhanced adaptability.

*Corresponding author. e-mail: akhieppa@gmail.com

Introduction

We investigate the problem of enhancing users' comprehension and confidence in deep Reinforcement Learning (RL) systems. To assess the efficacy of this XAI system, we created a user interface and ran a test with actual users. The findings show that when the AI system provides an explanation, user approval and confidence are significantly higher than when it does not.[1] It is anticipated that mobile robots and other autonomous systems would be crucial in demanding settings like space and in sectors like industrial production and transportation. Usually, different algorithms are used to map the environment, locate the robot on the map, calculate a path to the objective, and then carry out the trajectory. Adari et al. has been published for their work on applying artificial neural networks to fiber-reinforced polymer composites, evaluated using the ARAS method.[2] These approaches do have certain drawbacks, though, such as the presumption that there are no transparent or dynamic objects in the surroundings. They can have a significant computational overhead and don't learn from mistakes or real-world experiences.

As a result, other strategies, such as deep reinforcement learning, are being investigated for autonomous navigation. However, there are a number of difficult design choices that must be made when applying deep reinforcement learning to particular tasks. The necessity of a structured methodology in the creation of deep reinforcement learning systems has not been well addressed by prior research. [3] Adaptive autonomy has been enabled by using machine learning. Based on the quantity and caliber of feedback on the system or job, machine learning is classified as supervised, unsupervised, or reinforcement. The feedback information given to learning algorithms is a labeled training data set. In unsuspended learning, the algorithm is not given any feedback information, and the goal is to group the sample sets according to how similar the input samples are. RL is a goal-oriented learning technique where a create a policy that maximizes a long-term payoff. At every level, an RL agent gets evaluation data on how effective. All work done in fields like psychology, computer science, economics, and so on is included in the term "RL." Approximate DP (ADP) is a more recent version of RL. [4] Almost all of these fields employ machine learning and artificial intelligence approaches in addition to conventional control design methods. The various tiers of Motion Planning, including control, trajectory planning, and strategic choices, are the subject of additional study. This article discusses Deep Reinforcement Learning (DRL), one of the many approaches that have been created in machine learning itself.

The study explains the fundamentals of DRL and offers insight into the hierarchical motion planning problem. [4]Alight decrease in processing power would allow it to be redirected toward accelerating the flight, which would save a substantial amount of energy due to shorter flight durations. Because energy and operating voltage have a quadratic relationship, reducing the onboard processor's supply voltage is a potent way to compute energy-efficiently within the allocated Swap budgets. [6]A

current field of study in computer vision and control systems is autonomous driving. Many companies in the sector, like Google, Tesla, NVIDIA, Uber, and Baidu, are dedicated to developing state-of-the-art autonomous driving cars because they have the Potential to truly make people's lives better. On the other hand, several games have successfully used the deep reinforcement learning technique. In particular, picture attributes extracted from raw images are often used to characterize state spaces in vision control systems. Compared to situations where the controller has just discrete and constrained action spaces, deep reinforcement learning methods perform worse when used in autonomous driving systems. For instance, certain Atari games, including Enduro and Space Invaders, only have four actions. Despite the high-dimensionality of its spaces, the rules and board states in the game Go are fairly simple to visualize. In many cases, visual issues are easily overcome, and agents only need to focus on optimizing the policy with limited action spaces. However, state spaces and input images from the surroundings for autonomous driving include extremely complex backgrounds and internal items, like people, that can change dynamically and exhibit unpredictable behavior.

These include a variety of challenging visual tasks like depth estimation, picture comprehension, and object detection. More significantly, in order to stay safe and avoid colliding with things in such challenging situations, our controller must respond appropriately and quickly. [7]Autonomous robotic systems are frequently evaluated utilizing evolutionary search-based approaches. However, these methods frequently use computationally demanding simulator-based models to evaluate test cases. [8] AIoT systems can attain ambient intelligence by utilizing reinforcement learning (RL), which provides a set of methodologies for addressing the closed-loop difficulty of processing sensory inputs to make control decisions.

Agents interact directly with their environment to develop appropriate rules that relate states and actions. These learning agents must perceive the current environmental state (such as room temperature) and take appropriate actions (such as adjusting a thermostat) to influence future states while optimizing immediate rewards (e.g., maintaining a desired temperature) and maximizing long-term benefits over time. Notably supervised learning, in that it needs the agent to experiment to identify which actions produce the largest long-term reward. [9]The attacker's purpose in the game is to alter sensor data in order to change the optimal safe distance between automobiles, which could increase the likelihood of a collision or halt traffic. [10]One effective controller for autonomous robots is reinforcement learning. Since it can now accomplish tasks automatically by trial and error, it no longer needs prior knowledge or behavior. However, to accomplish complex tasks,

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

a lot of tries are needed. Thus, the assignment that can be completed with the only simple robots can be considered real. In light of these considerations, a number of techniques have been put out to increase. The autonomy, which is the most crucial aspect of reinforcement learning when applied to robots, is lost in the methods that employ Māori knowledge. Learning time is reduced by using the model. Although autonomy is maintained in this architecture, learned environmental knowledge cannot be applied to other tasks because the model is task-specific. [11]

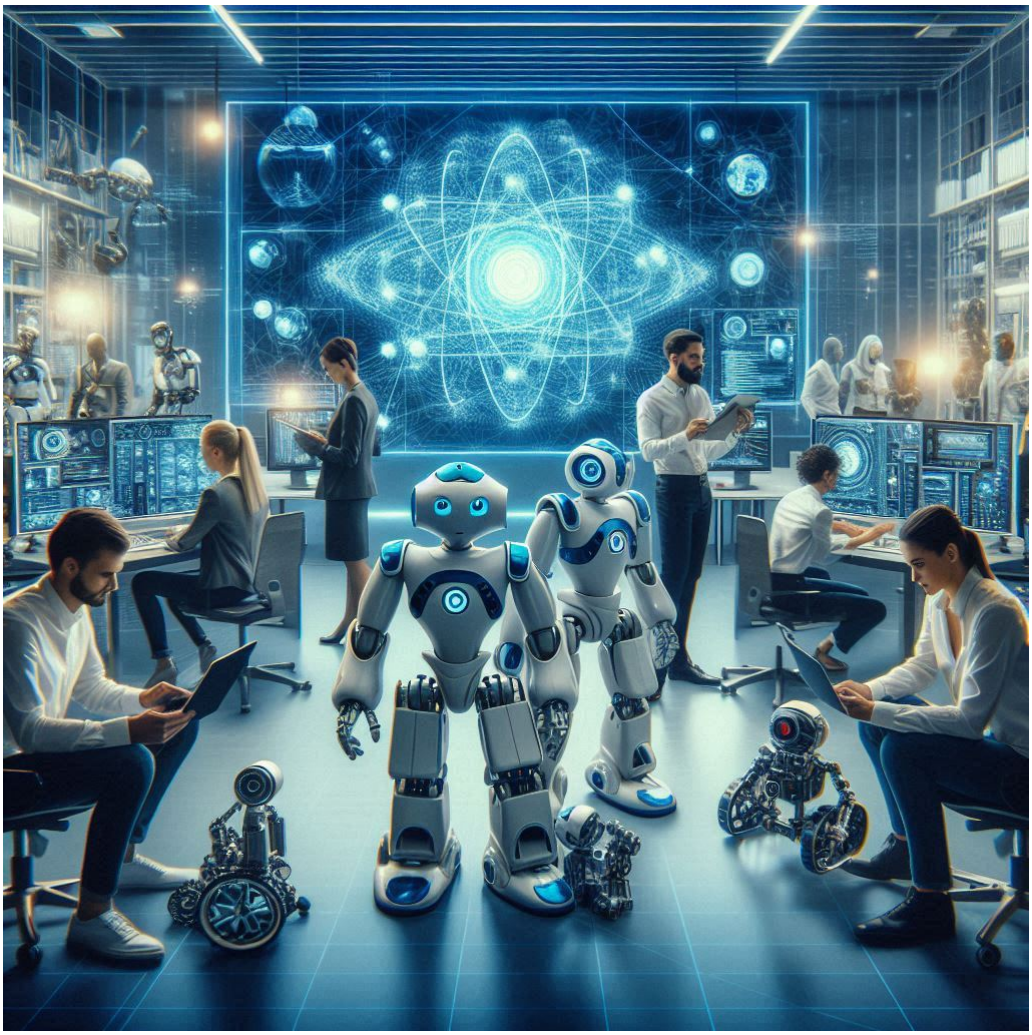


FIGURE 1. Reinforcement Learning in Autonomous Systems

The potential for autonomous vehicles to reduce traffic congestion and increase fuel efficiency and safety is significant. They are an important trend in contemporary mobility and a crucial part of the intelligent transportation networks of the future. With the driving behavior of an experienced driver as the learning aim, interactions between autonomous vehicles and traffic environments are modeled in this paper. Road geometry is incorporated into the stochastic. Vagvala, et al. presented a study on big data adaptation through distributed cloud systems, emphasizing the scalability and real-time processing capabilities of modern data infrastructures. [12] The automated braking system uses sensor-acquired information about the barrier to determine possible collision is identified. Four braking levels make up the action space of the braking control, which is framed as an optimization problem inside a Markov decision process (MDP) framework: 1) no braking, 2) mild braking, 3) moderate braking, and 4) forceful braking. In order to weigh the advantages of swiftly leaving the risk area against the possible harm from a collision, a reward function was created. The DQN model is trained to deal with situations like a person crossing a city street. According to experimental data, the control agent maintains optimal braking behavior in a variety of unpredictable

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

settings, thereby averting crashes. [13]Unlike traditional rule-based or predictive models, this system is fully data-driven and does not rely on preset guidelines. the proposed approach achieves a 16.3% reduction in energy consumption compared to conventional binary control methods.The real-time EMS model is trained and tested using actual commute data from Southern California. It employs a deep Q-network, integrating Q-learning with a deep neural network, to make optimal control decisions in continuous driving environments. Evaluation results confirm that the approach consistently outperforms traditional binary control techniques, achieving an average fuel savings of 16.3%. Future research will focus on further testing with real-world driving data and implementation on vehicle platforms. [14]Using simply camera images for AUV control can be energy-intensive, despite the fact that many deep learning systems educate artificial agents to make decisions using video images. Directly collecting rewards from video frames is really tough. [15] One of the most important and difficult problems for intelligent cars functioning in dynamic transportation scenarios is autonomous decision making.

A study evaluates machine learning techniques for predicting outcomes in 3D printing of composite materials. Ramancha compares multiple models to enhance production reliability, making it a valuable contribution to AI-powered manufacturing processes. [16] One of the biggest challenges in mobile robotics is still creating clever and reliable autonomous navigation systems. With the use of expert demonstrations, inverse reinforcement learning (IRL) provides an effective method for teaching robots how to carry out particular tasks without requiring the reward function to be manually specified. The majority of current IRL methods are used in experiments with relatively limited state spaces and use the assumption that the expert policy is optimum and deterministic. But in problems involving autonomous navigation, the state spaces are usually enormous, making it difficult for protests to travel to every state. In the meanwhile, the expert policy can be stochastic and suboptimal. Neural networks can easily provide an explicit policy representation, even for stochastic expert rules. Demonstrate the robot's autonomous navigation skills by progressing from small-scale experiments to completely unknown tasks. [17]

MATERIAL AND METHODS

Material

Input Parameters:

Sensor Accuracy (%): This measures how accurate the sensors (such as LIDAR, radar, cameras, etc.) are in detecting objects, obstacles, and the environment. A higher sensor accuracy percentage indicates better detection capability, which is critical for autonomous systems to make safe and informed decisions. The higher the accuracy, the better the system can understand its surroundings and react to environmental shifts.

Processing Power (GHz): This refers to the computational power of the onboard processor (CPU or GPU) in an autonomous system. The number of cycles per second that the processor can manage is expressed in gigahertz (GHz). Higher processing power allows faster data processing, enabling the autonomous system to analyze and react to data in real-time, crucial for decision-making and control tasks in dynamic environments.

Training Episodes (#): This is the number of iterations (episodes) the reinforcement learning model has gone through during training. the autonomous system learns throughout each episode. More training episodes allow the system to experience diverse situations and improve its decision-making capabilities, leading to a more refined and robust model.

Evaluation Parameter:

Average Reward (R): This is a performance metric that reflects how well the autonomous system is performing in the task. In reinforcement learning, agents receive rewards for completing specific actions that align with the goal. The average reward represents the mean value of all rewards accumulated during testing or after a set number of actions. A higher average reward indicates better performance, typically meaning the system is effectively achieving its objectives, such as navigating the environment safely and efficiently.

Machine Learning Algorithms

1. Linear Regression

Linear Regression (LR) is a popular statistical and machine learning method that is frequently used to simulate how variables relate to one another. One variable tends to rise along with the other in a positive connection. A negative association, on the other hand, occurs when one variable rises while the other falls. By measuring the statistically significant correlation between one or more variables, linear regression examines these relationships.

The variables in a linear regression model can be categorized into two types. The variable that is reliant on the by y, is the target value that the model seeks to predict or estimate. The independent variables, denoted by x1, x2, ..., xn, represent factors that influence or explain the behavior of the dependent variable.

The simplest form of linear regression is represented by the equation:

$$y = c + mx$$

where m is the line's slope and c is the y-intercept. With the intercept c showing the value of y when x is zero and the slope

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

m showing how much y varies for every unit change in x, this equation shows a straight-line relationship between x and y. The best-fitting line through a collection of data points is described by this equation.

In statistics, the linear regression equation is often written as:

$$y = \beta 0 + \beta 1\ x1$$

Here, β0 is the intercept, and β1 symbolises a basic linear regression model's slope. . The coefficients β0 and β1 are estimated using statistical methods that reduce the discrepancy between the observed and anticipated values of y, like least squares.

For more complex models, linear regression can involve multiple independent variables. When multiple predictors are included in the model, the general equation becomes:

$$y = \beta 0 + \beta 1\ x1 + \beta 2\ x2 + ... + \beta n\ xn$$

This equation is known as multiple linear regression, where x1, x2, ..., xn are the independent variables, and each of the corresponding coefficients β1, β2, ..., βn represents the change in y associated with a one-unit change in each of the respective predictors, assuming the other predictors are held constant.

Linear regression seeks to find the best-fitting line (or hyperplane in the case of many variables) that minimizes the error between the regression equation's projected values and the observed values of y. This inaccuracy is often quantified using a metric known as the residual sum of squares (RSS), which is the sum of the squared disparities between observed and anticipated values.

A basic statistical method for modelling and examining relationships between variables is linear regression. Whether it's used for predicting future values, estimating the strength of relationships, or understanding how different factors influence an outcome, linear regression provides valuable insights into the dynamics of various phenomena. By finding the optimal coefficients that most accurately depict how the dependent and independent variables are related, linear regression remains a core tool in both statistics and machine learning.

2. Random Forest Regression

Random forest regression (RFR) is a powerful approach to predictive modeling in supervised machine learning. It falls under the category of ensemble methods and is based on decision tree algorithms. Random forest effectively generates a bunch of decision trees, each trained on a different dataset subset. The random forest enhances prediction accuracy by averaging the outputs of various trees, while lowering the computational costs associated with storing, training, and generating predictions with multiple separate models. This makes random forest very useful for regression problems, where it is frequently used to forecast continuous values.

The random forest approach works by creating a "forest" of many decision trees. Every tree in the woods is built independently, and the random forest's overall projection is the average of all the individual tree forecasts. The trees themselves are built using the bagging technique, also known as bootstrapping, which entails training each tree on a random subset of the data. Because each tree in the forest is exposed to a variety of data, this strategy reduces variance and over fitting, hence enhancing the model's generalizability.

In a random forest model, the decision trees are typically trained in parallel, making it a highly efficient process that can be distributed across multiple computing resources. This parallelism is a key advantage of random forest, as it allows the algorithm to take advantage of modern computational power, particularly when working with big datasets.

The output of the random forest regression model is produced by taking the average of each individual tree. Mathematically, this can be expressed as:

$$\text{Random Forest Prediction} = \frac{1}{K}\sum_{k=1}^{k} h_k\,(x)$$

where Kis the overall quantity of separate regression trees constructed using the bootstrap samples, and $h_k(x)$symbolises the forecast provided by the k-th regression tree for vector x as input . This aggregation of predictions from multiple trees helps to smooth out errors and make the final prediction more robust.

One of the key benefits of using random forests is that they are relatively quick to train, especially when in contrast to alternative machine learning models. This is mostly because of how parallel the decision trees in the forest. Additionally, random forests are known for their high accuracy, which stems from their ability to combine the predictions of many diverse models and minimize errors through averaging.

The mean squared error (MSE) for out-of-bag (OOB) data is a frequently used metric to assess a random forest model's effectiveness. The data points that are not chosen for a particular decision tree's bootstrap sample are known as "out-of-bag data," and they can be utilised to validate the model. The OOB dataset's MSE is determined by:

$$\text{MSE}_{\text{OOB}} = \frac{1}{n}\sum_{i=1}^{n} (y_i - \hat{y}_i, \text{OOB})^2$$

In this equation, $y_i$represents the true worth of the i -th data point, and y_i, OOB is the predicted value for the i- th data point based on the aggregation of all the decision trees in the forest.

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

The sum of squared errors is averaged over all the data points in the OOB dataset to compute the MSE.

Another important metric for evaluating the random forest model is the coefficient of determination, often denoted as R^2, which shows the extent to which the model can account for the variation in the data. . For the OOB dataset, the R^2 value is calculated as:

$$R^2_{OOB} = 1 - \frac{MSC_{OOB}}{Var(y)}$$

where $Var(y)$is the output parameter's overall variance. y . The R^2value gives information about the percentage of the target variable's volatility that the model can account for. A higher R^2shows that the model fits the data better, whereas a lower R^2implies that a large portion of the variance cannot be explained by the model. .

In summary, random forest regression is a highly effective and efficient ensemble method for predictive modeling. By building a forest of decision trees, it reduces overfitting and variance, leading to more accurate predictions. The use of bagging and parallelism enables random forest to train quickly and scale well with large datasets. With metrics like MSE and R^2, the performance of a random forest model can be assessed, providing a reliable measure of its predictive power. Random forest continues to have gained popularity in machine learning applications because to their accuracy, efficiency, and resilience.

3. Support Vector Machines

Support Vector Machines (SVMs) are supervised machine learning algorithms that are widely used for regression, classification, and outlier detection. The SVM works by identifying the optimal hyperplane for dividing data points into discrete classes. The basic goal is to maximize the gap between data points from various classes while minimizing classification errors. Support Vector Machines (SVM) are based on the principle of determining the optimum hyper plane to partition data points in a higher-dimensional space. This makes the data linearly separable when it is not in its native space.

The primary purpose of SVM is to identify a hyper plane that maximizes the distance between classes. A hyper plane is a decision border that separates data points into two groups during a binary classification operation. This would depict both a two-dimensional line and a three-dimensional plane. SVM, however, can function in any number of dimensions. The SVM algorithm is designed to find the highest margin, or the greatest distance between the hyper plane and the closest data points in each class. These nearest points, also known as support vectors, are important since they determine the hyper plane's location and direction. Binary classification

Let's consider the case of a binary classification problem where the goal is to classify data points into one of two classes: +1 and -1. Given a training dataset $\{(x^1, y^1), (x^1x2, y^2), \dots, (x^n, y^n)\}$, where x^i represents a feature vector and $y^i \in \{-1, +1\}$is the corresponding class label, we aim to find a hyperplane defined by the equation:

$$w \cdot x + b = 0$$

where:

w is the weight vector perpendicular to the hyper plane.

b is the bias term.

x represents any point in the input space.

The objective is to maximize the margin $\frac{1}{||\omega||}$, which corresponds to minimizing $||\omega||$, subject to the constraint that each data point is correctly classified. For any data point x_i, the classification rule is:

$$yi(w \cdot xi + b) \geq 1$$

This ensures that all data points are correctly classified with a margin of at least 1 from the hyperplane. Thus, the optimization problem becomes:

$$\min_{w\,b} \frac{1}{2}||w||^2 \text{subject to } yi(w \cdot xi + b) \geq 1, i = 1,2, \dots, n$$

Kernel Trick

SVM use the kernel trick to translate data into a higher-dimensional space where it becomes linearly separable when the data is not linearly separable in its original space. SVM use a kernel function to calculate the transformation rather than doing so explicitly. $K(x, x')$that calculates the higher $-$ dimensional space$'$s inner product

$$K(x, x') = \phi(x) \cdot \phi(x')$$

where $\phi(x)$ is the mapping function. Common kernel functions include:

Linear kernel: $K(x, x') = x \cdot x'$

Polynomial kernel: $K(x, x') = (x \cdot x' + c)^d$

Radial Basis Function (RBF) kernel: $K(x, x') = \exp(-\gamma \parallel x - x' \parallel)^2$

SVM is computationally efficient because of the kernel trick, which enables it to function in higher-dimensional spaces without explicitly calculating the converted coordinates.

Support Vectors Support vectors are the data points closest to the hyper plane. These details are essential for establishing the optimal hyper plane. If the support vectors are removed, the position and orientation of the hyper plane could change

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

significantly. Therefore, SVM focuses on these critical points to make decisions about the hyper plane. In the equation for classification, when figuring out the best decision boundary, the support vectors are crucial.

SVM for Regression (SVR)In addition to classification, Regression issues, in which the objective is to predict a continuous output rather than classifying data points. This is called Support Vector Regression (SVR). In SVR, the objective is to find a function that keeps the margin between the data points and the regression line while departing from the actual values by no more than a given amount (or hyper plane in higher dimensions). The optimization problem for SVR is slightly different but follows the same principles of maximizing margins while minimizing errors.

RESULT AND DISCUSSION

TABLE 1.Reinforcement Learning in Autonomous Systems

Sensor Accuracy (%)	Processing Power (GHz)	Training Episodes (#)	Average Reward (R)
85	2.5	500	200
90	3	1000	300
80	1.8	800	180
95	2.2	600	240
88	2.8	700	260
92	3.5	1200	320
86	2	900	210
89	2.7	750	270
91	3.1	1100	310
84	1.9	550	190
87	2.3	950	220
93	3.2	1300	330
82	1.7	600	160
96	3.6	1500	350
83	2.1	500	170
94	3.3	1400	340
89	2.6	1000	290
88	2.9	1250	310
85	2.4	600	200
92	3.4	1600	360
81	1.5	400	150
97	3.8	2000	400
84	2	750	180
86	2.2	500	190

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

90	3	1000	280
93	3.4	1800	390
82	1.6	700	160
91	3.2	1300	300
95	3.7	1700	370
88	2.5	900	250

Table 1 presents a summary of various parameters relevant to reinforcement learning in autonomous systems. Each row corresponds to a different scenario, showcasing the relationship between sensor accuracy, processing power, training episodes, and the average reward obtained by the system. Sensor accuracy, ranging from 80% to 97%, represents the effectiveness of the system's sensors in detecting and responding to the environment. Processing power, indicated in GHz, varies from 1.5 GHz to 3.8 GHz, showing the computational capacity available for training and decision-making.

Training episodes, from 400 to 2000, refer to the number of iterations the system undergoes to learn optimal behavior. The average reward (R), spanning from 150 to 400, reflects the system's performance, with higher rewards indicating better overall results. Notably, there appears to be a trend where higher sensor accuracy and increased processing power generally correlate with more training episodes and higher average rewards. For instance, scenarios with sensor accuracy of 96% and processing power of 3.8 GHz yield the highest reward of 400, indicating that greater resources contribute to improved system performance. Conversely, lower sensor accuracy and reduced processing power result in lower rewards, highlighting the importance of optimized hardware and efficient training processes in reinforcement learning systems.

TABLE 2. Descriptive Statistics

	Sensor Accuracy (%)	Processing Power (GHz)	Training Episodes (#)	Average Reward (R)
count	30.000000	30.000000	30.000000	30.000000
mean	88.533333	2.663333	995.000000	262.666667
std	4.732378	0.672865	429.584804	75.563781
min	80.000000	1.500000	400.000000	150.000000
25%	85.000000	2.125000	625.000000	192.500000
50%	88.500000	2.650000	925.000000	265.000000
75%	92.000000	3.200000	1287.500000	317.500000
max	97.000000	3.800000	2000.000000	400.000000

Table 2 presents the descriptive statistics of the dataset, offering a summary of key metrics across four parameters: sensor accuracy, processing power, training episodes, and average reward. For each parameter, we can observe the

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

following Sensor Accuracy has a mean of 88.53% and a standard deviation of 4.73%. The minimum accuracy recorded is 80%, while the maximum is 97%. The interquartile range (IQR) shows that 50% of the data falls between 85% and 92%, indicating a slight concentration around the higher accuracy levels. Processing Power has an average of 2.66 GHz with a standard deviation of 0.67 GHz. The range spans from 1.5 GHz (min) to 3.8 GHz (max), suggesting a moderate variation in processing power across the data. The IQR between 2.13 GHz and 3.2 GHz shows that most systems tend to have processing power in this range. Training Episodes has an average of 995, with a substantial spread as indicated by the standard deviation of 429.58. The range spans from 400 to 2000 episodes, with the IQR between 625 and 1287 episodes, indicating a wide range of training durations across systems. Average Reward has a mean of 262.67, with a standard deviation of 75.56, demonstrating a moderate variability. The IQR lies between 192.5 and 317.5, showing that most systems yield rewards within this range, with the highest recorded reward reaching 400. Overall, the data demonstrates a broad variability in the performance and configuration of autonomous systems, with higher sensor accuracy and processing power generally correlating with higher rewards.

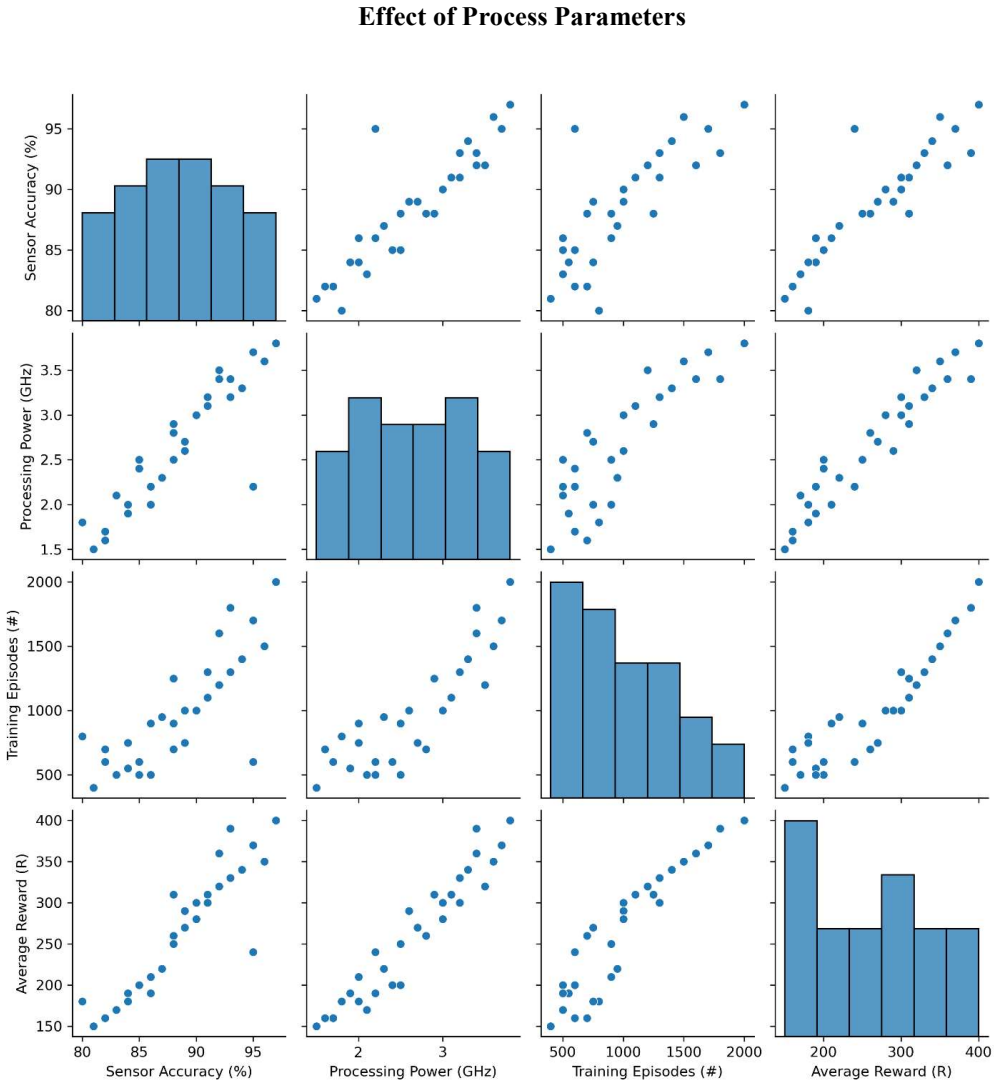


FIGURE 2 . Scatter plot of the various Reinforcement Learning in Autonomous Systems process parameters

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

The scatter plot matrix in Figure 2 visualizes the relationships among various reinforcement learning parameters in autonomous systems, including Sensor Accuracy (%), Processing Power (GHz), Training Episodes (#), and Average Reward (R). Each diagonal subplot represents the histogram of a specific parameter, showing its distribution. The off-diagonal scatter plots illustrate pair wise relationships between these parameters. From the scatter plots, a positive correlation is evident between Sensor Accuracy and Average Reward, suggesting that higher accuracy contributes to better learning outcomes. Similarly, Processing Power exhibits a strong positive correlation with Training Episodes and Average Reward, indicating that higher computational capability enhances learning efficiency. The number of Training Episodes also positively influences the Average Reward, implying that extended training generally results in improved performance. These insights suggest that optimizing Sensor Accuracy, Processing Power, and the number of Training Episodes can significantly enhance the reinforcement learning process in autonomous systems. The data patterns highlight key dependencies that can guide improvements in system design and training strategies for better decision-making and adaptation in dynamic environments.

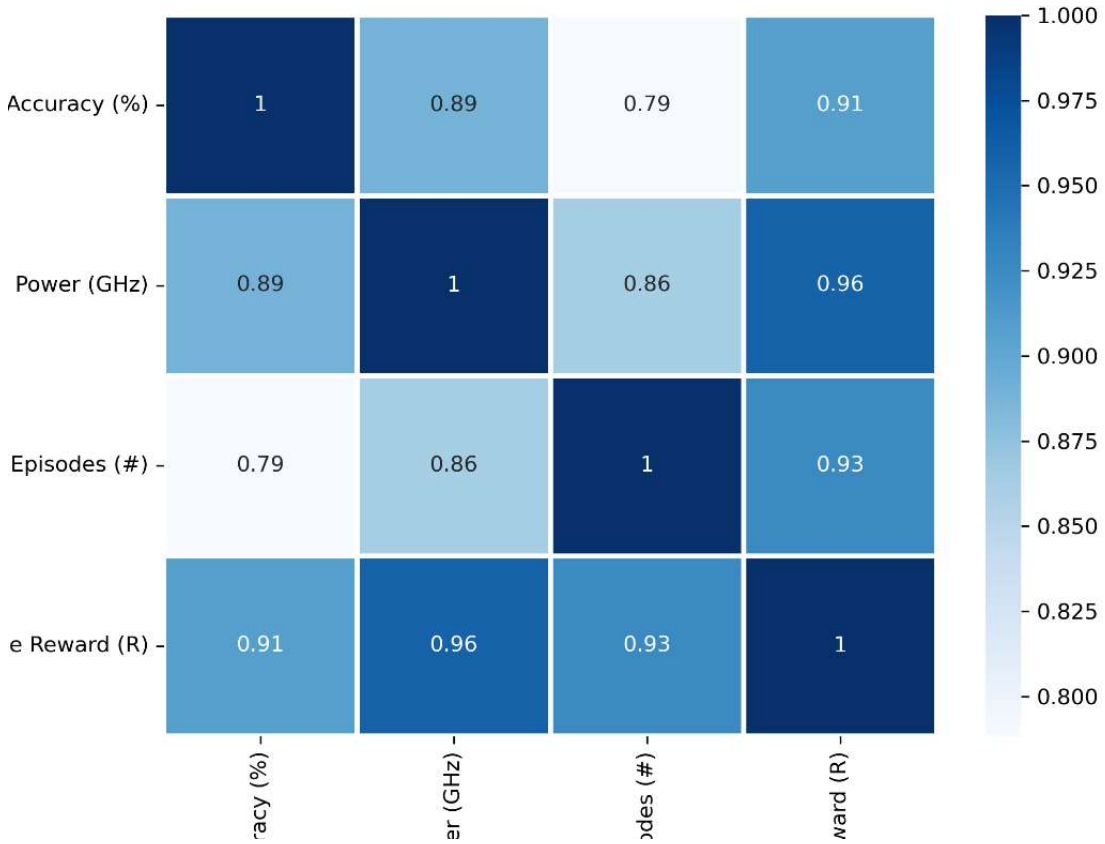


FIGURE 3. Correlation heat map between the process parameters and the responses

Figure 3 presents a correlation heat map illustrating the relationships between various process parameters and response variables in reinforcement learning for autonomous systems. The heat map quantitatively represents the correlation coefficients, ranging from -1 to 1, where values closer to 1 indicate a strong positive correlation, and values near 0 suggest little to no correlation. From the heat map, Sensor Accuracy (%) shows a high positive correlation with Average Reward (R) (0.91), indicating that more accurate sensors contribute significantly to better learning outcomes. Additionally, Sensor Accuracy has a strong correlation with Processing Power (0.89), suggesting that systems with greater computational power tend to support higher sensor accuracy.

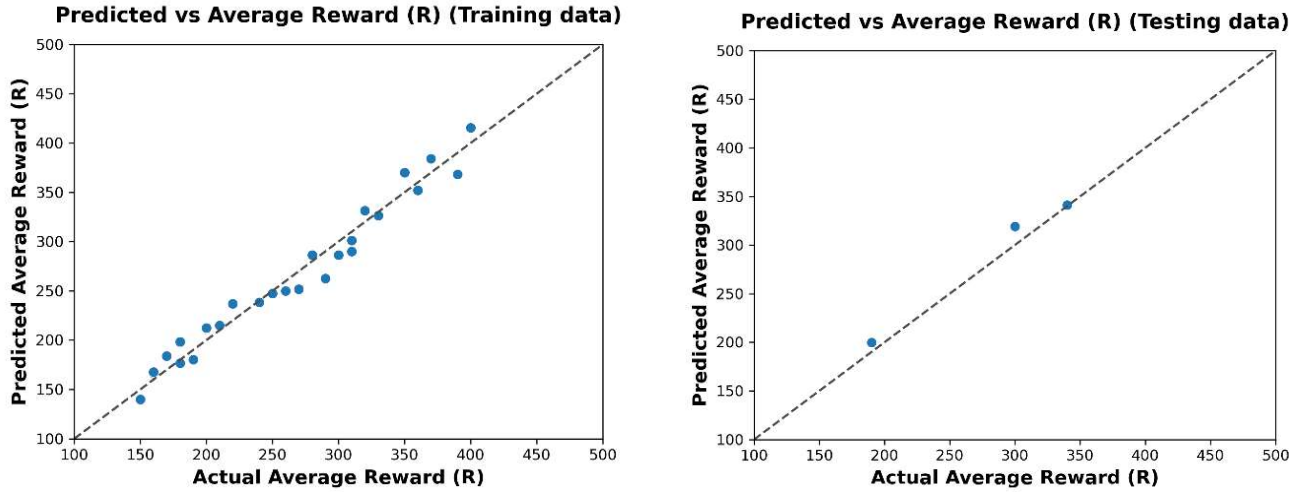
Processing Power (GHz) has the highest correlation with Average Reward (0.96), implying that increased processing capabilities lead to improved reinforcement learning

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

performance. Furthermore, Processing Power also correlates strongly with Training Episodes (0.86), suggesting that higher computational capacity allows for more training iterations, ultimately enhancing learning efficiency. Training Episodes (#) and Average Reward (R) are also highly correlated (0.93), demonstrating that longer training durations yield better reinforcement learning outcomes. This relationship highlights the importance of extensive training in optimizing system performance.

Additionally, Training Episodes exhibit a moderate correlation with Sensor Accuracy (0.79), suggesting that accuracy improvements benefit from prolonged learning. these strong positive correlations emphasize the interdependence of computational power, sensor accuracy, training duration, and learning performance. Optimizing these parameters can significantly enhance reinforcement learning efficiency in autonomous systems, ensuring better decision-making and adaptability in dynamic environments.

Linear Regression



a) b)

FIGURE 4. Predictive performance of the linear regression predictive model in Reinforcement Learning in Autonomous Systems (a) train; (b) test.

Figure 4 presents the predictive performance of a linear regression model in reinforcement learning for autonomous systems, using training data. (a) The scatter plot compares the actual average reward (R) with the predicted average reward (R). The dashed diagonal line represents the ideal scenario where predicted values perfectly match actual values. From the plot, the data points are closely aligned along the diagonal, indicating that the linear regression model achieves a strong fit to the training data. This suggests a high correlation between the input features (such as sensor accuracy, processing power, and training episodes) and the predicted reward. The model effectively captures the underlying pattern in the data, showing minimal deviations from the ideal line.

However, slight deviations from the diagonal line can be observed, which may indicate minor prediction errors due to inherent noise in the data or limitations of linear regression in capturing complex, nonlinear relationships. Despite this, the overall performance suggests that the model generalizes well within the training set. This analysis confirms that linear regression is a useful tool for approximating reward functions in reinforcement learning, especially when relationships between parameters are primarily linear. Future improvements could involve nonlinear models such as decision trees or neural networks for better handling of complex dependencies. The figure (b) presents a scatter plot comparing the predicted and actual average rewards (R) for testing data in a reinforcement learning-based autonomous system. The x-axis represents the actual average reward, while the y-axis denotes the predicted average reward obtained from a linear regression model. The dashed diagonal line serves as the ideal reference line, where perfect predictions would lie, indicating a 1:1 correspondence

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

between predicted and actual values. Observing the plotted points, the model demonstrates a reasonable predictive performance, as the points are relatively close to the diagonal line. However, some deviations exist, particularly in lower and mid-range values, suggesting minor prediction errors.

These discrepancies could result from limitations in the linear regression model, potential over fitting to training data, or inherent stochasticity in reinforcement learning environments. Despite these minor errors, the model generally captures the trend of the actual rewards, making it a useful tool for estimating expected rewards in autonomous system decision-making. This evaluation highlights the effectiveness of linear regression in predicting performance metrics in reinforcement learning scenarios, although more sophisticated models, such as neural networks or ensemble learning techniques, could enhance predictive accuracy further.

Random Forest Regression

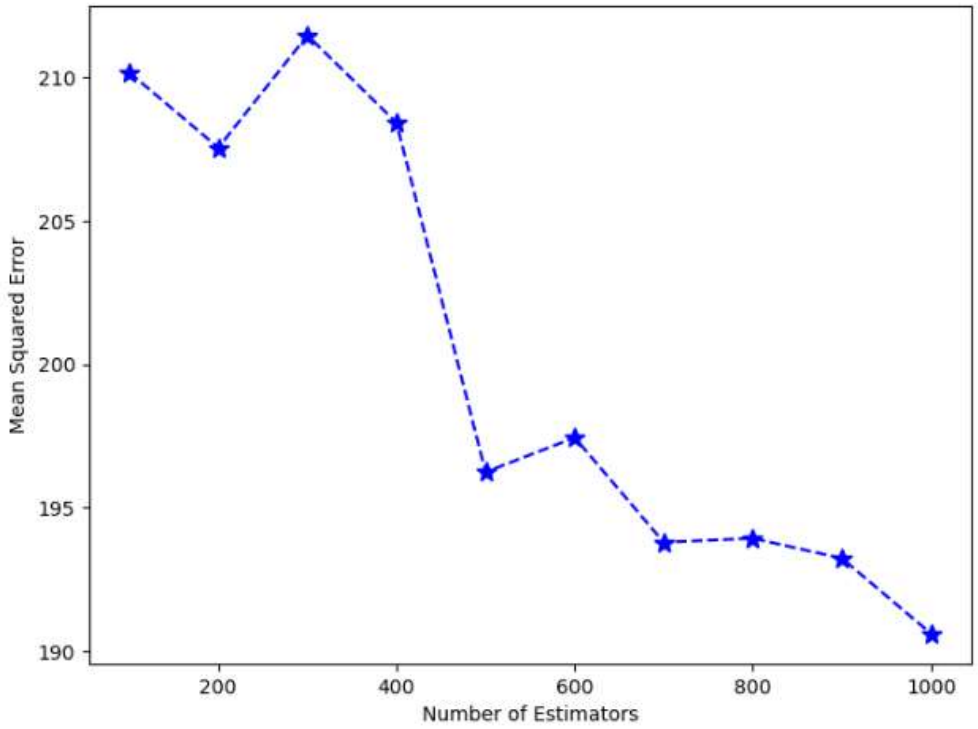


FIGURE 5. Effect of number of repressor in random forest regression on Number of Estimators vs Mean Squared Error

The figure 5 illustrates the relationship between the number of estimators in a random forest regression model and the corresponding mean squared error (MSE). The x-axis represents the number of estimators (trees) in the random forest, while the y-axis denotes the MSE, a key metric for evaluating model performance. The plot shows a general decreasing trend in MSE as the number of estimator's increases, suggesting that a larger number of trees improves model accuracy by reducing prediction error. Initially, MSE fluctuates slightly for lower values of estimators, indicating some instability in performance. However, beyond approximately 400 estimators, there is a significant drop in MSE, followed by a more gradual decline as the number of estimator's approaches 1000. This trend aligns with the expectation that increasing the number of trees enhances predictive accuracy by reducing variance and preventing over fitting to training data. The diminishing returns observed at higher estimator values suggest that beyond a certain point, additional trees contribute marginally to error reduction. Therefore, an optimal number of estimators must be selected to balance computational efficiency and model performance. This analysis highlights the importance of tuning hyper parameters in random forest regression to achieve an optimal trade-off between accuracy and efficiency.

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

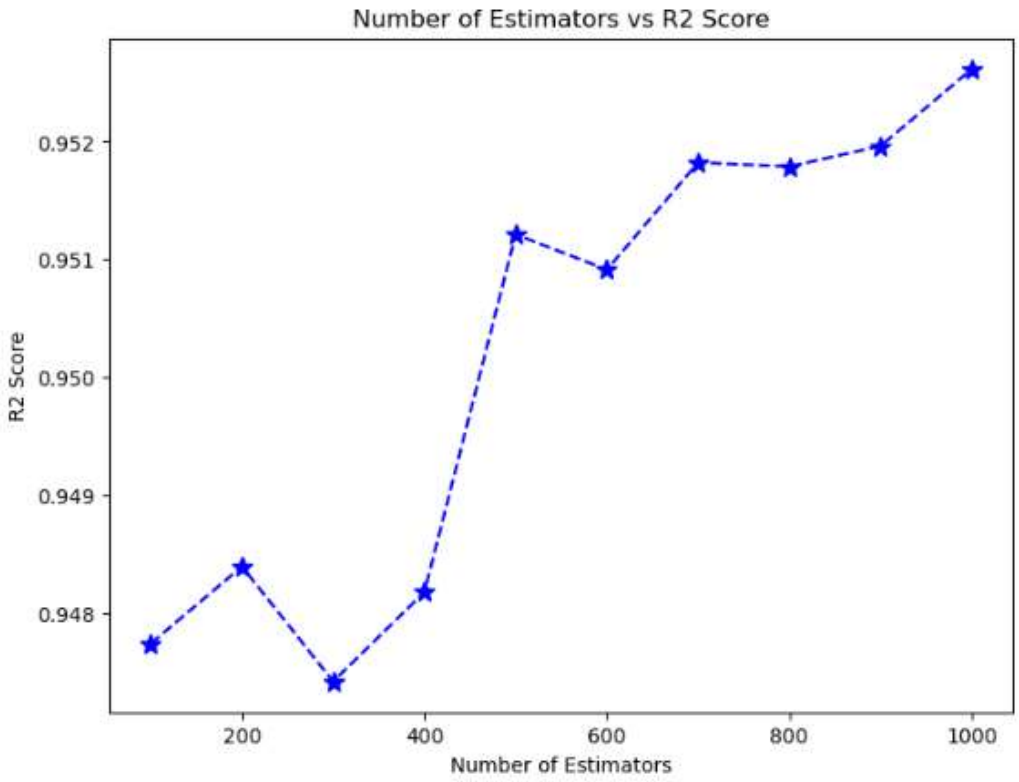


FIGURE 6. Effect of number of repressor in random forest regression on Number of Estimators vs Mean Absolute Error

The figure 6 presents the relationship between the number of estimators in a random forest regression model and the R^2 score, a key metric that measures how well the model explains the variance in the data. The x-axis represents the number of estimators (trees), while the y-axis denotes the R^2 score, which ranges between 0 and 1, with higher values indicating better model performance. The plot shows a general increasing trend, suggesting that as the number of estimator's increases, the model's predictive power improves. Initially, the R^2 score exhibits fluctuations, indicating some instability when the number of estimators is low. However, after approximately 400 estimators, there is a significant rise in R^2, followed by a steady improvement as more estimators are added. Beyond 600 estimators, the improvements become more gradual, with diminishing returns in predictive performance. This trend suggests that increasing the number of trees enhances the model's ability to generalize but only up to a certain point. Beyond an optimal number of estimators, additional trees provide marginal improvements while increasing computational cost. Therefore, selecting an appropriate number of estimators is crucial to balancing model accuracy and efficiency in random forest regression.

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

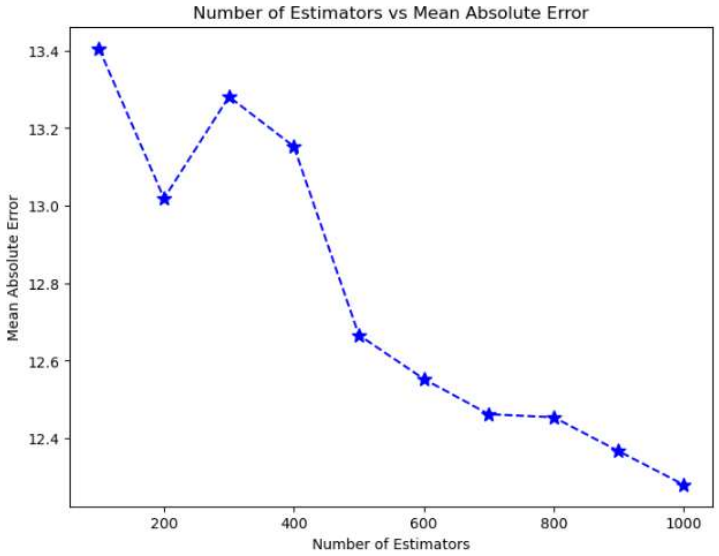
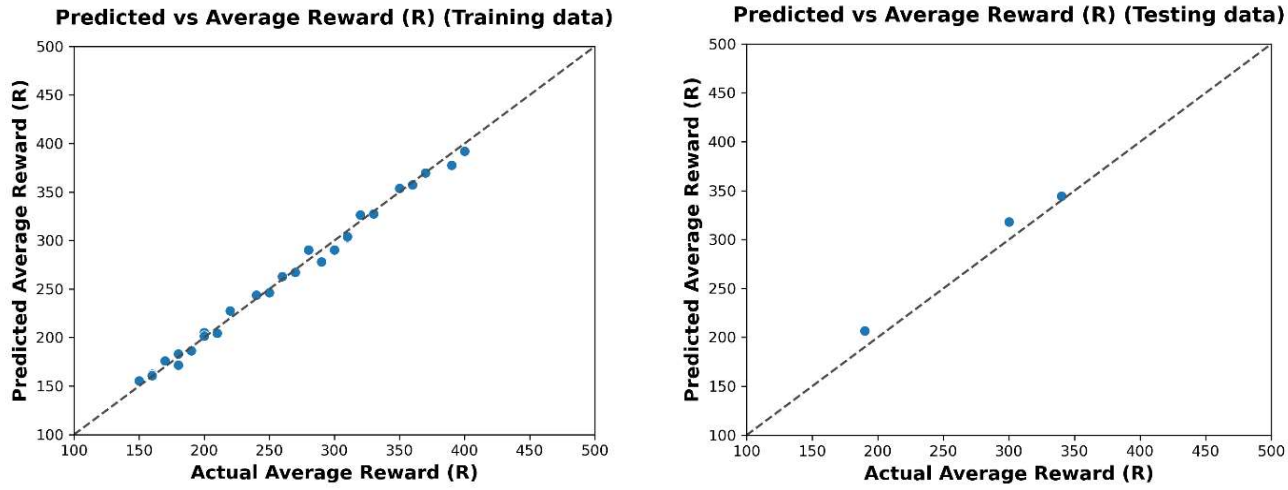


FIGURE 7. Effect of number of repressor in random forest regression on Number of Estimators vs R2 Score

The provided 7 graph illustrates the relationship between the number of estimators and the mean absolute error (MAE) in a random forest regression model. As the number of estimators increases, the MAE tends to decrease, indicating improved model performance. Initially, with a lower number of estimators (e.g., around 200), the MAE is relatively high, suggesting that the model is not yet fully stable and may suffer from high variance. As the number of estimators increases beyond 400, a noticeable decline in MAE occurs, signifying better predictive accuracy and reduced variance. This trend continues, with diminishing returns, as the number of estimators approaches 1000. The fluctuation in MAE at lower estimator values may be due to insufficient ensemble averaging, which stabilizes as more trees are added. The graph suggests that increasing the number of estimators improves model performance up to a point, beyond which the gains become marginal. This aligns with the principle that a larger ensemble reduces over fitting and variance in predictions, ultimately enhancing generalization. However, excessive estimators can lead to computational inefficiency without significant performance improvement. In practical applications, an optimal number of estimators should be chosen to balance accuracy and computational cost.



a) b)

FIGURE 8. Predictive performance of the random forest regression predictive model inReinforcement Learning in Autonomous Systems a) train b) test

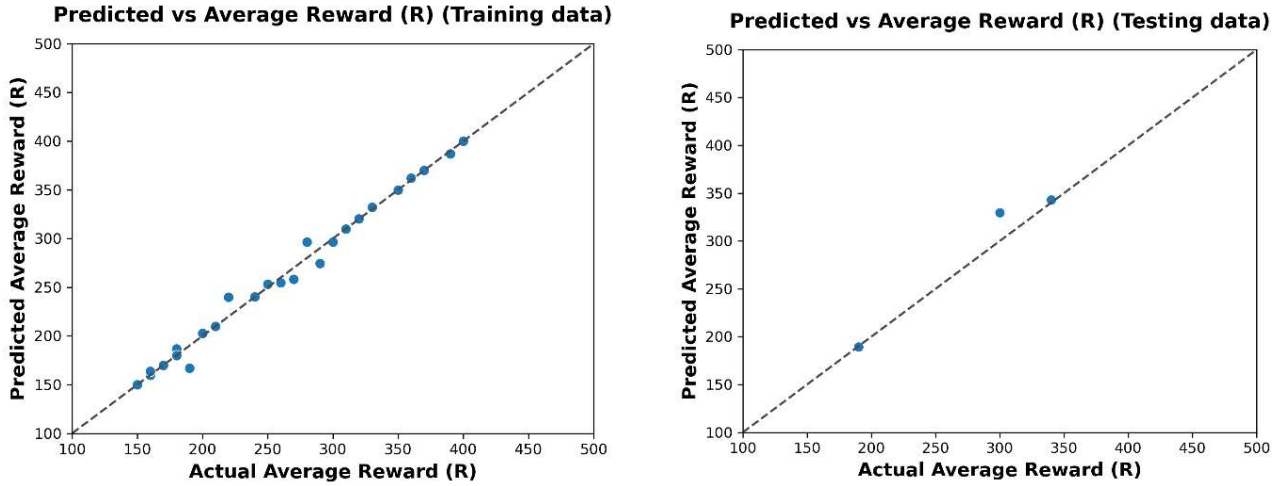
Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

The provided a) graph illustrates the predictive performance of a random forest regression model in reinforcement learning for autonomous systems using training data. The plot compares the actual average reward (x-axis) with the predicted average reward (y-axis). The data points align closely with the dashed diagonal line, indicating a strong correlation between predicted and actual values. This suggests that the model effectively captures the underlying patterns in the training data, leading to highly accurate predictions. The minimal deviation from the diagonal line implies low prediction error, signifying that the model has learned well from the training dataset.

However, while high accuracy on training data is desirable, it is essential to evaluate the model on unseen test data to assess its generalization ability. Over fitting could be a concern if the model performs exceptionally well on training data but poorly on new data. In reinforcement learning applications for autonomous systems, precise reward prediction is crucial for making optimal decisions and improving system performance. The high correlation observed in this graph suggests that the random forest model can serve as a reliable predictor for estimating rewards, ultimately aiding in more effective learning and decision-making in autonomous environments. The provided b) graph illustrates the predictive performance of a random forest regression model in reinforcement learning for autonomous systems using testing data. The plot compares the actual average reward (x-axis) with the predicted average reward (y-axis). Unlike the training data performance, where the predictions closely followed the diagonal line, this test data plot shows a much sparser distribution with only a few data points. While some predictions align relatively well with the actual values, others deviate more noticeably from the ideal diagonal line, suggesting reduced model accuracy on unseen data. his discrepancy may indicate potential over fitting, where the model has learned the training data patterns well but struggles to generalize to new data.

The small number of test samples further limits the ability to fully assess the model's performance, making it crucial to evaluate on a larger test set. In reinforcement learning for autonomous systems, reliable reward prediction is essential for effective decision-making and long-term performance improvements. If the model does not generalize well, it may lead to suboptimal policies or incorrect estimations of rewards. To enhance generalization, techniques such as hyper parameter tuning, feature selection, or increasing the training data diversity could be considered to improve robustness in real-world scenarios.

Support Vector Machines



a) b)

FIGURE 8. Predictive performance of the random forest regression predictive model in Reinforcement Learning in Autonomous Systems a) train b) test

This scatter plot (Figure 8) a) demonstrates the predictive performance of a random forest regression model in the context of reinforcement learning, specifically showing the relationship between predicted and actual average rewards during training. The x-axis represents the actual average reward (R) values, while the y-axis shows the predicted average reward values generated by the model. The plot reveals a strong linear correlation between predicted and actual values, as evidenced by the points closely following the dashed diagonal line (which represents perfect prediction where predicted = actual). The data

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

points span from approximately 150 to 400 reward units, showing a consistent prediction accuracy across different reward magnitudes.

The tight clustering of points around the diagonal line suggests that the random forest model has achieved high predictive accuracy with minimal variance in its predictions. The model appears to perform particularly well in the middle range of rewards (between 250-350), where the points almost perfectly align with the ideal prediction line. There's slightly more scatter at the lower and higher ends of the reward spectrum, which is common in machine learning models as they typically have less training data in these extreme regions. Overall, this visualization indicates that the random forest regression model has successfully captured the underlying patterns in the reinforcement learning system's reward structure and can make reliable predictions about expected rewards during training. This scatter plot b) shows the testing performance of the random forest regression model on unseen data, evaluating how well the model generalizes beyond its training examples. The graph plots predicted average rewards against actual average rewards (R), with a dashed diagonal line representing perfect predictions. The testing data contains notably fewer points compared to the training plot, which is typical as testing sets are usually smaller than training sets. The data points are distributed across three main regions: one around 200, another around 300, and one around 350 on the reward scale.

These points generally align well with the diagonal line, suggesting that the model maintains good predictive accuracy on new, unseen data. The presence of accurate predictions on test data indicates that the random forest model has avoided over fitting and successfully learned generalizable patterns from the training data. The model appears to maintain consistent performance across different reward levels, though with slightly less data to evaluate compared to the training set. This sparse but accurate distribution of test points suggests the model is robust and reliable for practical applications in predicting rewards for autonomous systems. The close alignment with the diagonal line, particularly in the middle range (around 300-350), demonstrates that the model's predictions remain reliable when faced with new scenarios not seen during training.

TABLE 3. Regression Model Performance Metrics (Training Data)

Data	Symbol	Model	R2	EVS	MSE	RMSE	MAE	Max Error	MSLE	MedAE
Train	LR	Linear Regression	0.968202	9.68E-01	180.0341	13.41768	11.78375	27.53653	0.002766	1.12E+01
Train	RFR	Random Forest Regression	0.993174	9.93E-01	38.64644	6.216626	5.318951	12.3	0.000588	5.15E+00
Train	SVR	Support Vector Regression	0.988722	9.89E-01	63.85022	7.990633	4.549631	23.14915	0.00131	2.12E+00

Table 3 presents the performance metrics of three regression models (Linear Regression, Random Forest Regression, and Support Vector Regression) on the training data. The key metrics provided include R^2, Explained Variance Score (EVS), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), Maximum Error (MaxError), Mean Squared Logarithmic Error (MSLE), and Median Absolute Error (MedAE). Linear Regression (LR): The model achieves a high R^2 value of 0.968, indicating that it explains approximately 96.8% of the variance in the data. The MSE of 180.03 and RMSE of 13.42 suggest a moderate error in predictions. The MAE of 11.78 reflects the average absolute deviation, and the MaxError of 27.54 indicates the largest discrepancy between predicted and actual values. The MSLE and MedAE are relatively small, showing that the errors are well-distributed. Random Forest Regression (RFR): This model performs exceptionally well, with an R^2 of 0.993, suggesting 99.3% of the variance is explained. It has a much lower MSE (38.65) and RMSE (6.22), indicating very accurate predictions. The MAE (5.32) and MaxError (12.3) are also low, indicating fewer large errors compared to LR. Support Vector Regression (SVR): SVR also shows strong performance, with an R^2 of 0.989, indicating 98.9% variance explained. The MSE and RMSE are moderate (63.85 and 7.99), with an MAE of 4.55 and a higher MaxError of 23.15. Random Forest Regression outperforms the other two models in terms of accuracy and error

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

metrics, followed by SVR, with Linear Regression showing the least favorable results. This study evaluates machine learning techniques for predicting outcomes in 3D printing of composite materials. Mishra, et al. Introduce a hybrid MOORA method for reverse logistics provider selection, supporting circular economy models.[20]

TABLE 4. Regression Model Performance Metrics (Testing Data)

Data	Symbol	Model	R2	EVS	MSE	RMSE	MAE	Max Error	MSLE	MedAE
Test	LR	Linear Regression	0.961968	9.87E-01	152.9714	12.36816	10.01099	19.09648	0.002071	9.62E+00
Test	RFR	Random Forest Regression	0.948399	9.91E-01	207.5517	14.40666	13.01889	18.20667	0.003492	1.65E+01
Test	SVR	Support Vector Regression	0.927185	9.55E-01	292.8799	17.11373	11.01116	29.49082	0.00294	2.92E+00

Table 4 presents the performance metrics of the three regression models (Linear Regression, Random Forest Regression, and Support Vector Regression) on the testing data, providing insight into how well these models generalize to unseen data. Linear Regression (LR): The model shows strong performance with an R^2 of 0.9619, meaning it explains about 96.2% of the variance in the testing data. The MSE (152.97) and RMSE (12.37) indicate moderate prediction errors. The MAE of 10.01 reveals that the average absolute deviation is relatively low, while the MaxError of 19.1 indicates that the largest prediction error is smaller compared to other models. The MSLE and MedAE are also reasonably small, suggesting good model fit with some outliers. Random Forest Regression (RFR): This model has an R^2 of 0.9484, explaining 94.8%a of the variance. Although it performs well, the MSE (207.55) and RMSE (14.41) are higher than those of Linear Regression, indicating slightly more prediction error. The MAE of 13.02 and MaxError of 18.21 suggest that the model occasionally makes larger errors, but still performs relatively well. Support Vector Regression (SVR): SVR exhibits the lowest R^2 value of 0.9272, explaining 92.7% of the variance. The MSE (292.88) and RMSE (17.11) are the highest among the models, reflecting greater prediction errors. The MAE (11.01) and MaxError (29.49) are also larger, indicating that SVR has a tendency to make more significant errors in its predictions. Linear Regression performs the best on the testing data, followed by Random Forest Regression, with Support Vector Regression showing the least favorable results in terms of error metrics.

CONCLUSION

In order for autonomous systems to adjust and operate at their best in changing contexts, reinforcement learning, or RL, has become an essential approach. The effectiveness of the learned policies and the overall system performance are determined by the average reward (R), the main evaluation parameter. When it comes to autonomous systems, reinforcement learning algorithms are essential for figuring out the best course of action through experimenting and interacting with the environment. The number of training episodes, processing power, and sensor precision are some of the variables that affect how efficiently the learning process works. Increased sensor precision improves state observations' dependability and lowers decision-making uncertainty. The learning algorithm's efficiency is directly impacted by processing power, which is expressed in GHz and determines the speed and complexity of calculations.

The quantity of training episodes is essential for enabling the RL agent to efficiently explore and take advantage of its surroundings, which eventually raises the average rewards. When it comes to managing intricate interactions between variables and identifying non-linear patterns that affect the reward function, Support Vector Machines (SVMs), especially in regression settings (SVR), are excellent. Applying these techniques aids in optimizing system settings and enables a greater comprehension of the elements influencing RL performance. Experiments show that RL results in autonomous systems are much improved by adjusting input parameters. Increased sensor accuracy results in more accurate depictions of

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

the surroundings, which lowers incorrect actions and enhances the quality of decisions. Real-time adaptation is made possible by faster training and inference times brought about by increased processor capacity. More training episodes enable more thorough investigation, which results in improved policies and better average rewards. However, after a given number of training cycles, where additional learning gains become negligible, declining returns could be seen. Implementing RL in autonomous systems is difficult, despite its benefits. Particularly in complicated contexts that need for deep reinforcement learning (DRL) structures, computational demands might be significant. Furthermore, it is still difficult to guarantee generalization across many contexts because policies that have been trained in controlled environments might not necessarily translate well to practical implementations.

Another crucial element is striking a balance between exploration and exploitation, since too much exploration might result in less-than-ideal performance and too much exploitation can produce local optima. A possible paradigm for improving autonomy in intelligent systems is provided by reinforcement learning, which is backed by machine learning algorithms. RL models can perform better, as seen by greater average rewards, by maximizing training episodes, processing power, and sensor precision. Enhancing sample efficiency, cutting down on processing overhead, and creating adaptable techniques for dynamic and unpredictable. The development of autonomous systems will be fueled by the ongoing improvement of RL techniques, which will increase their capability, effectiveness, and adaptability in practical applications.

REFERENCES

1. Druce, Jeff, Michael Harradon, and James Tittle. "Explainable artificial intelligence (XAI) for increasing user trust in deep reinforcement learning driven autonomous systems." arXiv preprint arXiv:2106.03775 (2021).
2. V K, Adari., Vinay Kumar, Ch., Srinivas, G., Kishor K , A., & Praveen Kumar, K. (2024). Artificial neural network in fibre-reinforced polymer composites using the ARAS method. SOJ Materials Science and Engineering, 10(1), 1-11.
3. Hillebrand, Michael, Mohsin Lakhani, and Roman Dumitrescu. "A design methodology for deep reinforcement learning in autonomous systems." Procedia Manufacturing 52 (2020): 266-271.
4. Kiumarsi, Bahare, Kyriakos G. Vamvoudakis, Hamidreza Modares, and Frank L. Lewis. "Optimal and autonomous control using reinforcement learning: A survey." IEEE transactions on neural networks and learning systems 29, no. 6 (2017): 2042-2062.
5. Aradi, Szilárd. "Survey of deep reinforcement learning for motion planning of autonomous vehicles." IEEE Transactions on Intelligent Transportation Systems 23, no. 2 (2020): 740-759.
6. Wan, Zishen, Nandhini Chandramoorthy, Karthik Swaminathan, Pin-Yu Chen, Vijay Janapa Reddi, and Arijit Raychowdhury. "BERRY: Bit Error Robustness for Energy-Efficient Reinforcement Learning-Based Autonomous Systems." In 2023 60th ACM/IEEE Design Automation Conference (DAC), pp. 1-6. IEEE, 2023.
7. Wang, Sen, Daoyuan Jia, and Xinshuo Weng. "Deep reinforcement learning for autonomous driving." arXiv preprint arXiv:1811.11329 (2018).
8. Humeniuk, Dmytro, Foutse Khomh, and Giuliano Antoniol. "Reinforcement learning informed evolutionary search for autonomous systems testing." ACM Transactions on Software Engineering and Methodology 33, no. 8 (2024): 1-45.
9. Lei, Lei, Yue Tan, Kan Zheng, Shiwen Liu, Kuan Zhang, and Xuemin Shen. "Deep reinforcement learning for autonomous internet of things: Model, applications and challenges." IEEE Communications Surveys & Tutorials 22, no. 3 (2020): 1722-1760.
10. Ferdowsi, Aidin, Ursula Challita, Walid Saad, and Narayan B. Mandayam. "Robust deep reinforcement learning for security and safety in autonomous vehicle systems." In 2018 21st International Conference on Intelligent Transportation Systems (ITSC), pp. 307-312. IEEE, 2018.
11. Ito, Kazuyuki, Akio Gofuku, Yoshiaki Imoto, and Mitsuo Takeshita. "A study of reinforcement learning with knowledge sharing for distributed autonomous system." In Proceedings 2003 IEEE International Symposium on Computational Intelligence in Robotics and Automation. Computational Intelligence in Robotics and Automation for the New Millennium (Cat. No. 03EX694), vol. 3, pp. 1120-1125. IEEE, 2003.
12. Vagvala, Prabhakar. "Cloud-Driven Big Data Adaptation Leveraging Distributed Systems." Journal of Data Science and Information Technology, vol. 1, 2024, pp. 32–41. www.sciforce.org.
13. Chae, Hyunmin, Chang Mook Kang, ByeoungDo Kim, Jaekyum Kim, Chung Choo Chung, and Jun Won Choi. "Autonomous braking system via deep reinforcement learning." In 2017 IEEE 20th International conference on intelligent transportation systems (ITSC), pp. 1-6. IEEE, 2017.
14. Qi, Xuewei, Yadan Luo, Guoyuan Wu, Kanok Boriboonsomsin, and Matthew J. Barth. "Deep reinforcement learning-based vehicle energy efficiency autonomous learning system." In 2017 IEEE Intelligent Vehicles Symposium (IV), pp. 1228-1233. IEEE, 2017.

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249

15. Carlucho, Ignacio, Mariano De Paula, Sen Wang, Yvan Petillot, and Gerardo G. Acosta. "Adaptive low-level control of autonomous underwater vehicles using deep reinforcement learning." Robotics and Autonomous Systems 107 (2018): 71-86.
16. Ramancha, Nitesh Kumar. "Leveraging Machine Learning for Predictive Modeling in 3D Printing of Composite Materials: A Comparative Study."International Journal of Intellectual Advancements and Research in Engineering Computations (IJIARREC), vol. 11, no. 4, Oct.–Dec. 2023, pp. 39–58. https://doi.org/10.61096/ijiarec.v11.iss4.2023.39-58..
17. Xia, Chen, and Abdelkader El Kamel. "Neural inverse reinforcement learning in autonomous navigation." Robotics and Autonomous Systems 84 (2016): 1-14.
18. Rodriguez-Galiano, Victor, Manuel Sanchez-Castillo, M. Chica-Olmo, and M. J. O. G. R. Chica-Rivas. "Machine learning predictive models for mineral prospectivity: An evaluation of neural networks, random forest, regression trees and support vector machines." Ore Geology Reviews 71 (2015): 804-818.
19. Ao, Yile, Hongqi Li, Liping Zhu, Sikandar Ali, and Zhongguo Yang. "The linear random forest algorithm and its advantages in machine learning assisted logging regression modeling." Journal of Petroleum Science and Engineering 174 (2019): 776-789.
20. Dewi, Christine, and Rung-Ching Chen. "Random forest and support vector machine on features selection for regression analysis." Int. J. Innov. Comput. Inf. Control 15, no. 6 (2019): 2027-2037.
21. Mishra, Mahesh Kumar. "Hybrid Approach: The MOORA Method for Reverse Logistics Provider Selection." SOJ Materials Science & Engineering, vol. 10, no. 2, 2024, pp. 1-10.
22. Ben Ishak, Anis. "Variable selection using support vector regression and random forests: A comparative study." Intelligent Data Analysis 20, no. 1 (2016): 83-104.
23. Teles, Germanno, Joel JPC Rodrigues, Ricardo AL Rabelo, and Sergei A. Kozlov. "Comparative study of support vector machines and random forests machine learning algorithms on credit operation." Software: Practice and Experience 51, no. 12 (2021): 2492-2500.
24. Aljahdali, Sultan, and Syed Naimatullah Hussain. "Comparative prediction performance with support vector machine and random forest classification techniques." International journal of computer applications 69, no. 11 (2013).
25. Ahmad, Iftikhar, Mohammad Basheri, Muhammad Javed Iqbal, and Aneel Rahim. "Performance comparison of support vector machine, random forest, and extreme learning machine for intrusion detection." IEEE access 6 (2018): 33789-33795.
26. Naghibi, Seyed Amir, Kourosh Ahmadi, and Alireza Daneshi. "Application of support vector machine, random forest, and genetic algorithm optimized random forest models in groundwater potential mapping." Water Resources Management 31 (2017): 2761-2775.
27. Shataee, Shaban, Syavash Kalbi, Asghar Fallah, and Dieter Pelz. "Forest attribute imputation using machine-learning methods and ASTER data: comparison of k-NN, SVR and random forest regression algorithms." International journal of remote sensing 33, no. 19 (2012): 6254-6280.

Citation: Akhilesh R. Eppa, "Optimizing Autonomous Systems through Reinforcement Learning: The Role of Linear Regression, Random Forest, and Support Vector Machines in Decision Making" International Journal of Computer Science and Data Engineering., 2025, vol. 2, no. 2, pp. 1–20. doi: https://dx.doi.org/10.55124/csdb.v2i1.249