International Journal of Robotics and Control Systems
Vol. 5, No. 2, 2025, pp. 1006-1028
ISSN 2775-2658
http://pubs2.ascee.org/index.php/ijrcs

A Hybrid Adaptive Gradient-Based Sled Dog Optimizer for Enhanced Robotic Decision-Making in Industrial Applications

Mohammad Rustom Al Nasar [a,1,*]

[a] College of Engineering and Technology (CET), Department of Information Technology Management, American University in the Emirates (AUE), Academic City – 14143, Dubai, United Arab Emirates
[1] mohammad.alnasar@aue.ae
* Corresponding Author

ARTICLE INFO

Article history
Received February 02, 2025
Revised March 05, 2025
Accepted April 10, 2025

Keywords
Artificial Intelligence;
Autonomous Systems;
Machine Learning;
Metaheuristic Optimization;
Reinforcement Learning

ABSTRACT

As autonomous robotic systems are increasingly used in industrial applications, there is a growing need to create efficient and automated decision-making capabilities that can work in complex environments with a range of possible actions. RL offers an effective way to train robotic agents. Still, conventional RL techniques tend to have issues with slow and unstable policy learning, poor convergence, and weak exploration-exploitation balance. To solve this problem, this paper develops a Hybrid optimization approach that incorporates reinforcement learning, deep learning, and metaheuristic optimization for more robust robotic control and adaptability. The new approach utilizes a Deep Q-Network with Experience Replay for learning policies. At the same time, an Adaptive Gradient-Based Sled Dog Optimizer is used to improve and optimize decision-making. Epsilon-greedy selection combined with Noisy Network is used for hybrid exploration-exploitation, which helps learning. The effectiveness of the proposed method was validated against five existing methods, which include Conservative Q-Learning, Behavior Regularized Actor-Critic, Implicit Q-Learning, Twin Delayed Deep Deterministic Policy Gradient, and Soft Actor-Critic, over the three benchmark robotic datasets of MuJoCo, D4RL, and OpenAI Gym Robotics Suite. The vast majority of results provide compelling support for the argument that the proposed approach consistently outperformed the baseline approaches in terms of accuracy, precision, recall, stability, speed of convergence, and degree of generalization. The improvement in performance was confirmed by validation methods such as analyzing confidence intervals and computing results of p-values.

This is an open-access article under the CC–BY-SA license.

1. Introduction

The integration of AI and ML into robotics and autonomous systems has become one of the fastest-developing fields within the industry due to its ability to allow increased intelligence along with more adaptable and efficient decision-making [1], [2]. Most industrial activities have come to depend on fully autonomous robotic systems for greater accuracy [3], [4], reliability, and productivity while minimizing the need for people and the costs of operation [5], [6]. Reinforcement learning (RL) [7], [8] in the form of a framework for autonomous decision-making has proven to be very useful as robotic agents can learn the best control strategies by interacting with the environment as a

trial and error-process [9], [10]. Nonetheless, conventional RL approaches have serious shortcomings when it comes to dealing with high-dimensional action spaces, slow convergence, and sub-optimal policy learning, which makes it extremely hard to apply them to real-life situations [11]-[13]. To overcome these obstacles, many researchers have focused on combining RL policies with hybrid AI-optimization techniques that incorporate metaheuristic optimization in order to improve the efficiency of exploration-exploitation balance and overall adaptability of the system [14]-[17].

Even with the advantages that AI-integrated robotics brings, learning stable and generalizable policies across changing industrial environments is still an area of challenge [18], [19]. Stagnation training is poorly efficient, meaning that it takes a significant number of training episodes to reach peak performance, and many policy-based reinforcement learning structures have to deal with these issues [20], [21]. Moreover, many of these learning policies are poorly adaptive due to a lack of effective exploration strategies [22], [23]. They do not wish to perform a new task because of their inability to learn new rules, practices, or schedules [24]. These weaknesses prompt RL agents to gravitate towards sub-optimal strategies, which, in turn, lowers their efficiency. In order to alleviate these issues, new exploration methodologies need to be developed that are less rigid and able to perform actions more freely without being hindered by the oiling issues of robotic decision-making [25], [26].

The use of autonomous industrial robots has become very important within areas of manufacturing activities, particularly in places where molding processes are required [27], [28]. Some robots have manipulators and can work together to execute complicated operations such as grit-blasting, covering surfaces with protective coatings, and spray painting, which all require complete coverage of the surfaces [29]. Optimal base placements in relation to the base, environment, and target object must be determined first if effective teamwork is to be achieved. The problem is further complicated with large objects with complex geometric configurations that require multiple base placements in order to get sufficient coverage. This issue is dealt with by proposing an Optimization of Multiple Base Placements (OMBP) method, which is supposed to optimize base placements of the robots for multi-robot cooperation [30]. The technique uses several criteria, such as torque efficiency, manipulability, makespan minimization, fair workload partitioning, and coverage maximization. Additionally, the distance that the robots kept from each other and from the surrounding environment to avoid collision was taken into account as well. The claimed effectiveness of the approach is substantiated by numerous simulated and real-life experiments of base placement optimization and alignment of the performance of simulated results to real-world conditions.

The effectiveness of intralogistics activities is enhanced by the automation of mobile robots (AMRs). AMRs have advanced hardware and control software that enables them to operate autonomously and in complex environments, which is why they are increasingly incorporated into logistics operations such as manufacturing, warehousing, cross-docks, terminals, and even hospitals. Their sophisticated hardware and control software enables independent decision-making. Unlike AGVs (automated guided vehicles), which only follow central control unit (CCU) orders for scheduling, routing, and dispatching, AMRs independently interact with neighboring resources such as machines and other systems. This ability allows the system to be more responsive to real-time environmental changes by decentralizing the decision-making process. This pace of decentralization has dramatically changed standard planning and control procedures, and thus, different decision-making paradigms are needed for AMR systems. This paper synthesizes and classifies literature on AMR planning and control in intralogistics to understand how advancements in AMRs influence decision-making processes [31]. To enable enhanced operational effectiveness, this paper introduces a comprehensive model of the AMR system for control and planning that guides managers in the decision-making process. A review of the evolving field is constructed so that new directions for remaining work can be identified.

Systems centered on robotics using reinforcement learning encounter a myriad of issues, such as policy optimization, training efficacy, and even adaptive decision-making. Traditional

reinforcement learning methods, such as Deep Q-Networks (DQN) [32] and Actor-Critic techniques, tend to be plagued by slow convergence, heavy stochasticity in learning outcomes, and a poor balance between exploration and exploitation [33], [34]. In addition, preambles within robotic control systems do not possess suitable functioning policies, which leads to elephantine rates of unstable and inefficient learning, which greatly affects the practicality of their usage in the real global industrial market. As the nature of robotic tasks is particularly complex, there is an urgent requirement for an approach that combines AI and optimization at a much greater level, which utilizes reinforcement learning, deep learning, and optimized metaheuristics to make learning much more efficient, stabilize policy updates, and improve robotic control systems' adaptability.

The rationale for pursuing this research originates from the growing need for autonomous robotic systems that can efficiently and accurately perform intricate industrial activities [35], [36]. Reinforcement learning offers a viable approach for making robotic systems intelligent [37], [38]. Still, its great expense for computation, lengthy training periods, and lack of responsiveness to shifts in the surroundings are major drawbacks. It is possible to increase the policy learning process, reduce the time it takes to reach optimum solutions and enhance the quality of exploration through the combination of metaheuristic optimization techniques, allowing robots to learn more efficient strategies for controlling the systems. Furthermore, being able to test the method on multiple benchmark datasets means that the method is not only theoretically sound but it is also practical and scalable to real-life problems. The need for reliable, flexible, and intelligent robotic learners motivates this work, where the objective becomes building a hybrid AI-optimization framework that seeks to solve these important issues.

This paper describes a Hybrid AI-optimization approach aiming to improve robotic design and decision-making in industrial activities through reinforcement learning, deep learning, and metaheuristic optimization. The proposed approach is developed to overcome some existing limitations in RL techniques in order to achieve better convergence, strong exploration-exploitation trade-offs and policy robustness. The approach follows a structured process of learning and optimization, which consists of several key steps. First, the robotic control task is structured as a Markov Decision Process (MDP), where the system interacts with the environment to learn an optimal policy to maximize its cumulative rewards. Then, the agent controls a target Deep Q-Network With Experience Replay to approximate the Q-value function. In this manner, the agent learns the optimal action-value relationship. To prevent the Q-value from drastically changing during the frantic updates, a target network is used to decouple the learning. The policy learning is further updated by employing an original Adaptive Gradient-Based Sled Dog Optimizer (AG-SDO), which changes his exploration-exploitation parameters and optimizes network updates, which toughen and strengthen learning. A hybrid exploration-exploitation strategy that consists of epsilon-greedy selection and Noisy Networks is further employed to prevent getting stuck in local optima. At last, the training process is supervised with the Mean Squared Bellman Error (MSBE) to make sure there are no learning lags. Also, supervised statistical techniques for validation, such as confidence intervals and p-value calculations, are used to check the actual statistical significance of the performance improvement.

In order to gauge the performance of the suggested technique, experiments were performed using three prominent datasets pertaining to robotic learning. The MuJoCo (Multi-Joint Dynamics with Contact) dataset acts as a physics engine for simulation-based robotic motion planning and control. The D4RL (off-line RL datasets) benchmark is used for the assessment of the reinforcement learning techniques in off-line robotic environments. Moreover, the OpenAI Gym Robotics Suite is a popular environment for reinforcement learning in robotic decision-making. The method has been compared against five dominant approaches of reinforcement learning, namely Conservative Q-Learning (CQL), Behavior Regularized Actor-Critic (BRAC), Implicit Q Learning (IQL), Twin Delayed Deep Deterministic Policy Gradient (TD3), and Soft Actor-Critic (SAC). The comparison is done based on six key performance indexes: Accuracy, Precision, Recall, F1-Score, Sensitivity, and Specificity. In addition, more sophisticated measures like confidence intervals and p-value

validation techniques are used in order to prove and provide evidence for the results. The proposed method is expected to substantially outperform the rest of the approaches with regards to learning performance, stability, and adaptability which makes it very suitable for industrial robotic applications.

The remainder of this document is organized in the following manner. Section 2 describes the novel Hybrid AI-Optimization Approach, including its mathematical expressions, optimization approaches, and policy learning methods. Section 3 covers the experiment design and execution, mentioning hardware and software components, dataset information, and assessment standards. Section 4 contains the conclusion as well as the next works, including what is expected as refinement and field implementation issues.

2. Methods

The innovative method put forward incorporates a hybrid AI-optimization framework that aims to improve robotics and autonomous systems in industrial operations by encompassing reinforcement learning (RL), deep learning (DL), and metaheuristic optimization methods. The method attempts to enhance decision-making, motion planning, and control by seeking a balance between a self-learning data-centric approach and a human-controlled optimization approach. The method was developed to solve critical aspects of complicated robotic systems, which include large-scale action spaces, stochastic motion execution, and policy search for real-time control.

At the heart of the proposed framework lies a robotic agent trained with reinforcement learning that employs an adaptive gradient-based metaheuristic optimization approach to learn an optimal policy. The process of training is further improved by applying deep neural networks (DNNs) to speed up the process through function approximation and hybrid exploration-exploitation strategies. The optimization part makes sure that the policies are trained to be efficient with respect to all industrial tasks while avoiding overfitting. The next subsections illustrate the proposed method holistically, explaining its mathematical formulations, algorithmic steps, and implementation details.

2.1. Problem Formulation

Autonomous robotic systems operating in industrial settings necessitate the management of interactions, actions, and uncertainties; therefore, traditional control structures are not sufficient [25], [39]. To address this issue, we consider this robotic learning problem to be a Markov Distributed Decision Process which offers a way to model multi-dimensional issues in sequential decision making under uncertainty mathematically. The MDP is represented as a tuple:

$$MDP = (S, A, P, R, \gamma) \quad (1)$$

Where S includes sensor measurements, motor set pours, and the state of the environment, which combine to form the configuration space of the robotic system. Each state $s \in S$ describes the instantaneous condition of the robot in relation to its current environment.

A consists of any movements, actions, or commands that can be performed by the robotic system in question. Each action $a \in A$ describes what the robot does, related to changing the level of thrust in the motors, the direction of specific movements, and even pushing or pulling things.

$P(s' \mid s, a)$ represents the transition probability function that indicates the possibility of movement from the current state s to the next state s', having acted a. This function captures the probabilistic essence of real-life robotic systems, which involves sensor noise and other environmental disturbances.

$R(s, a)$ is the reward function that gives an immediate reward value (or penalty) R, which depends on the state s, and the action taken a. Reward functions must be defined for each task and are expected to motivate the robot to perform in the best possible ways. In industrial robotics tasks,

for instance, high energy expenditure or inaccurate movements could be severely penalized, while low energy expenditure and accurate movements could be highly rewarded.

γ is the discount factor, where $0 < \gamma \leq 1$. This parameter determines how much weight is given to future rewards compared to more immediate ones. When this factor is high, strategic decisions are ensured, while lower values encourage rapid, short-term gains.

The primary purpose of reinforcement learning in an MDP context is to derive an optimal policy $\pi(a \mid s)$ which deterministically selects the best action to take in any state s, so as to achieve the maximum cumulative expected reward. This can be formulated mathematically as follows:

$$J(\pi) = E[\Sigma(\gamma\text{^}t * R(s\_t, a\_t))] \quad (2)$$

Here, $J(\pi)$ is the expected return under policy π, the term $\gamma\text{^}t$ takes care of discounting future rewards exponentially for stability of learning, and $R(s\_t, a\_t)$ is the reward received at the time step t.

2.2. Policy Optimization and Learning Procedure

For the hybrid AI-optimization proposal, the reinforcement learning agent interacts with and learns from the environment by acting, experiencing, and modifying its policy for better decision-making [40]. The learning can be detailed as follows:

- State Observation: The robotic system observes the current state s by means of the sensors and encoders, thus capturing real-time information about the surroundings.
- Action Selection: The agent chooses an action for a given current policy $\pi(a \mid s)$, which could either be a fully deterministic or stochastic action.
- State Transition: The system moves from one state s to s' by acting with the step $s' = (s, a)$ governed by $s' = s + a$ and the transition probability $P(s'|s, a)$.
- Reward Calculation: The environment supplies a reward $R(s, a)$ depending on the usefulness of the action in achieving the objectives of the task.
- Policy Update: The reinforcement learning model updates the policy parameters $\pi(a \mid s)$ with the aid of optimization techniques that change decisions and try to achieve the highest cumulative reward.

The optimal policy is found by computing the Bellman equation, which defines the value of a state as the expected return from the best possible action taken from that state:

$$V(s) = max\, E\,[\, R(s, a) + \gamma * V(s')\,] \quad (3)$$

This equation is the Bellman equation. Remember that $V(S)$ is the state value function, which represents the maximum expected reward for the optimal policy from state s. The formula for the optimal action-value function also called the Q-function, follows is as:

$$Q(s, a) = E\,[\, R(s, a) + \gamma * max\, Q(s', a')\,] \quad (4)$$

This equation states that $Q(s, a)$ predicts an average return of executing action a in state s and follows the optimal policy afterward for the remaining states.

In the process of training the reinforcement learning model, the policy parameters θ are modified in an incremental approach by using a loss function that comes from the difference between the mean squared errors of the current Q-value estimates and the anticipated target Q values. The aforementioned loss function is:

$$L(\theta) = E\,[(R + \gamma * max\, Q(s', a';\, \theta) - Q(s, a;\, \theta))\text{^}2] \quad (5)$$

Where θ is associated with the neural network parameters of the Q-function. For learning stability, the target network with parameters θ_target is used and updated from the online network using a soft update rule:

$$\theta\_target = (1 - \tau) * \theta\_target + \tau * \theta \quad (6)$$

Where τ is the small update factor that allows smoother transitions to the target network.

This iterative process is performed repeatedly until the most optimal solution is reached for the policy, enabling the robotic agent to effectively and autonomously interact within complex and dynamic industrial environments. Metaheuristic optimization is used to refine the policy further so as to guarantee the highest possible efficiency and breadth of generalization.

2.3. Policy Learning that is Based on Reinforcement Learning Concepts

As a result, the robotic control policy to be used in industrial settings is learned using a Deep Q-Network (DQN) with Experience Replay [41]. DQN is a reinforcement learning approach that is value-based and uses deep learning together with Q-Learning to allow the agent to use a neural network to approximate the action-value function $Q(s, a)$. This allows the robot to make intelligent decisions by estimating the long-term rewards different actions will deliver in a particular state. The Q-function gives the expected value of the total reward that an agent is expected to receive if he executes action a at state s and acts according to the optimal policy from that time onward [42], [43]. It can be put in mathematical form as follows.

$$Q(s, a) = E\,[\,R(s, a) + \gamma * max\, Q(s', a')] \quad (7)$$

Here, the notation $R(s, a)$ represents the immediate reward after executing action a. We let γ be the balancing factor between short and long-term rewards. The term max $Q(s', a')$ stands for the highest Q-value predicted for the next state s', which means that the agent will execute the best action in the future. Since the environment is high dimensional, obtaining and storing classification Q-values for every state-action pair is very expensive computationally. Thus, we use a deep neural network (DNN) to parameterize $Q(s, a)$ with a set of weights θ.

To increase learning stability, DQN integrates experience replay, where past episodes (s, a, r, s) are stored randomly and sampled to remove the correlation between successive updates. This makes sure that the model is exposed to new sets of actions, thus controlling overfitting due to sequences of actions. One of the major changes that the Deep Q Network proposed is that the network changes its parameters θ using loss derived from mean squared errors of the predicted Q-values relative to the Q-target values. The loss function is therefore defined as follows.

This is how we proposed a method of estimating the Q-value associated with a policy that is based on maximizing the expected value obtained after learning -referred to as the Future Expected Reward (FER):

$$L(\theta) = E\,[(R + \gamma * max\, Q(s', a';\, \theta) - Q(s, a;\, \theta))\text{^}(2)\,] \quad (8)$$

Where $R + \gamma * max\, Q(s', a';\, \theta)$ represents the computed Q value for the target state s', making sure that the agent learns to make optimal decisions in the long run.

To improve the stability of learning, an additional technique used in DQN architectures is the Target Network, which employs a separate set of weights θ_target to produce $Q*$ value estimates, ensuring more stable learning. The Target Network updates its weights periodically using a soft update mechanism, where the new target weights are computed as $\theta\_target = (1 - \tau) * \theta\_target + \tau * \theta$, resulting in smoother transitions during learning. Here, τ is a small update factor that determines the influence of the main network's weights on the Q-values in the Target Network. The soft update rule helps prevent oscillations in value estimates, ensuring more reliable policy updates. By integrating DQN with Experience Replay and Target Networks, the proposed method enhances policy learning stability, enabling the robotic agent to operate optimally and make informed decisions in complex industrial environments.

2.4. Hybrid Metaheuristic Optimization for Policy Refinement

In a bid to improve the learned policy even further, the method in question uses a metaheuristic optimization strategy that integrates an advanced hybrid strategy and reinforcement learning and appropriately manages the exploration and exploitation settings in the reinforcement learning model. The Adaptive Gradient-Based Sled Dog Optimizer (AG-SDO) is then applied to adjust the policy network's parameters θ with the intent of avoiding getting trapped within suboptimal policies and simultaneously maintaining the ability to generalize to novel robotic environments. Unlike traditional gradient-based optimizers, which only take into account the local gradients, AG-SDO employs adaptive perturbation techniques, which modify the optimization problem dynamically in order to increase the efficiency of the search within the high dimensional policy space. The point of this hybrid approach of AG-SDO is to enable the reinforcement learning practitioner to break away from local minima that are less optimal and achieve better-adjusted policy update regions through a moderate increase in policy movement [44].

Formulating the optimization problem as eliminating a regularization term would lead to policy refinement through the approximation of the minimization of the loss function $L(\theta)$ under the assumption of overfitting by the regularization term $\Omega(\theta)$. The equation is given in formal terms.

$$\theta = arg\ min\ L\ (\theta) + \lambda * \Omega\ (\theta) \tag{9}$$

Where λ goals served by the regularization parameter are multi-favored with respect to the trade balance they strike between rigidity and soft data. The goal of introducing $\Omega(\theta)$ is to ensure that the optimized policy does not become overly adaptable to competing noisy or superfluous signals within the environment.

The optimization process updates the policy parameters iteratively with an inbuilt policy learning mechanism. The following equation computes subsequent parameter updates.

$$\theta\_(t+1) = \theta\_t - \alpha * \nabla L(\theta\_t) + \beta * \Delta(\theta\_t) \tag{10}$$

Here, α is the predefined learning rate resource that determines the size of the update steps, $\nabla L(\theta\_t)$ is the policy gradient which steers the entire process of learning to, at the very least, improves the expected rewards, and $\nabla L(\theta\_t)$ is an AG-SDO derived adaptive perturbation term. The perturbation term $\Delta(\theta_t)$modifies the exploration parameter based on performance feedback from previous iterations in order to make sure the optimization process is not overly focused on suboptimal areas in the policy space. This system permits the lower bounding adjustment for AG-SDO, ensuring that most positive experiences can be utilized, but negative experiences are used to make decisions for better policy adjustments.

Put, AG-SDO highlights two ideas. First, SDO – which is a variant of deep policy gradient methods – achieves more progress towards solving the problem with policy improvement by perturbing the environments in a certain way. Second, SDO – even in very complex environments AGSDO.1 – performs very well. The robustness of G-SDO is achieved as follows – we monitor the policy gradient change across iterations – convergence is guaranteed if policy gradients do change, i.e., the policy was improved. Empirically, some rates of modification to α and β parameters do work. By integrating hybrid optimization with deep reinforcement learning, as previously mentioned, the method increases the efficiency of achieved policies, convergence rates, and floating adaptability in real-world robotic controls.

2.5. Adaptable Learning Agent

One of the core problems in reinforcement learning is how to balance exploration, meaning discovering new strategies, versus exploitation, which means refining the best strategies known so far. Solving these problems will be key to implementing the method successfully in a variety of different task domains [45]. These attempts can be summed up by referring to them as hybrid adaptive learning.

The first part of the hybrid strategy is epsilon-greedy selection, a familiar technique used in reinforcement learning that incorporates some degree of randomness into the decision-making process. At each time step, the agent has the option of either executing a random action with the probability ε (exploration) or executing the action that has the highest Q-value with a probability of $(1 - \varepsilon)$ (exploitation). To make sure a balance between exploration and refinement is achieved, ε is decreased over time. The decay function governing ε follows an exponential schedule:

$$\varepsilon t = \varepsilon\_min + (\varepsilon\_max - \varepsilon\_min) * exp(-\lambda t) \quad (11)$$

Where ε_max and ε_min represent the upper and lower limits of exploration probability, t is the training episode, and λ is a decay parameter that determines how fast exploration will decrease. This approach to scheduling gives the agent the ability to focus on exploration during the latter parts of training. It enables the agent to focus on greedy exploitation towards the start of training when learning is more stable.

The other part of the hybrid strategy is Noisy Networks for Stochastic Policy Execution, which increases the eclectic diversity through noise injection into the parameters. Instead of using an explicit probability distribution, policy exploration is achieved by perturbing the face-electing angle by means of Gaussian noise:

$$\theta' = \theta + \eta * N(0, I) \quad (12)$$

In this equation, η is a scaling factor responsible for the amount of noise. Also, $N(0, I)$ means a Gaussian noise with zero means and unit variance. This guarantees that the agent explores different actions even in advanced stages of training, thus averting early convergence to inferior policies. The described method also exploits the stochastic execution of policies, which enables the proposed method to maintain a stochastic balance between structured decision-making and randomness, thus allowing learning with different degrees of precision in different levels of the environment.

2.6. Examination of Convergence and Stability

In order to make sure that training is done correctly and learning is not unstable, the novel approach being exercised tracks the Mean Squared Bellman Error (MSBE), which measures how different the current Q-value predictions are from their expected value [46]. This can be expressed mathematically in the following manner:

$$MSBE = E\,[(R + \gamma * max\, Q(s', a') - Q(s, a))\text{^}2] \quad (13)$$

Where $R + \gamma * max\, Q(s', a')$ corresponds to the expected target Q-value pertaining to the next state s' and $Q(s, a)$ stands for estimate at the current state. If MSBE is decreasing over time then that means that the model is learning to approximate the optimal Q-function better. The condition for convergence is that the MSBE difference is lower than a certain value, which can be defined as:

$$|\, MSBE\_(t + 1) - MSBE\_t \,| < \delta \quad (14)$$

Where δ is a set limit in order to prevent any form of additional computation after a certain point, in combination with confidence interval estimation and p-value analysis, the performance gains are validated to check if they are genuine or just noise. These steps provide additional assurance that, indeed the method learned the optimal control policies, and those can be generalized to solve a range of different industrial tasks robustly.

The proposed Hybrid AI-Optimization Approach for Enhancing Robotics and Autonomous Systems uses Reinforcement Learning (RL) and Deep Learning (DL) along with a metaheuristic optimization technique to create a robotic system for an industry where decision-making is simple and adaptive. For policy learning, a Deep Q-network (DQN) is applied, allowing the agent to approximate optimal Q-values and fine-tune decisions over time. To enhance the learned policy, an Adaptive Gradient-Based Sled Dog optimization (AG-SDO) is also integrated into the method. This

is a metaheuristic optimization technique that aims to improve overall policy efficiency while avoiding the center of gravity phenomenon. The synergy of RL-based learning with optimization refinement creates a learned policy that makes it attainable to strike the right balance between exploration and exploitation so that the robotic system is able to find the best movement strategies to use in all conditions.

The proposed method operates through five key stages. To begin with, the challenge is outlined as a Markov Decision Process (MDP) in which the robotic framework is defined as an agent in an environment that is meant to maximize its cumulative reward. Then, to learn the optimal policy, a Deep Q-Network (DQN) with Experience Replay is used, whereby Q-values are updated iteratively by accounting for the transitions and earned rewards. To avoid instability during learning, a Target Network is added to the system, allowing the updates to the policy parameters to be adjusted in a less volatile manner. The AG-SDO performs this phase by refining the learned policy through escalation of exploration and perturbation to overcome local minima. This guarantees that the robotic system does not prematurely converge to suboptimal strategies but rather learns globally optimized control policies. Also, the two-step hybrid methodology utilizes the epsilon decision rule for action selection and Noisy Networks to add robustness to the policy. The last phase of analysis specifies how convergence and stability will be carried out by continually monitoring the MSBE to ensure that the learning process is progressing toward the optimal solution. In a nutshell, Algorithm 1 was introduced to the users to guide them through the implementation of the proposed methods which are the representation of the learning and the optimization steps.

Algorithm 1. Hybrid AI-Optimization for Robotics

Input: Environment E, Learning Rate α, Discount Factor γ, Exploration Rate ε
Output: Optimized Policy π*

Initialize Q-network with weights θ
Initialize target network with weights θ_target = θ
Initialize AG-SDO optimizer parameters
Initialize experience replay buffer B
For each episode, do:
  Observe initial state s
  For each time step t do:
    Select action a using hybrid exploration-exploitation strategy
    Execute action a, observe reward r and next state s.'
    Store transition (s, a, r, s') in experience replay buffer B
    Sample mini-batch from B
    Compute Q-value update using the Bellman equation:
    $Q(s, a) = R(s, a) + \gamma * \max Q(s', a')$
    Compute policy loss function:
    $L(\theta) = E [(R + \gamma * \max Q(s', a'; \theta) - Q(s, a; \theta))^2]$
    Apply AG-SDO optimization to refine θ:
    $\theta_{t+1} = \theta_t - \alpha * \nabla L(\theta_t) + \beta * \Delta(\theta_t)$
    Update Q-network weights θ using gradient descent
    Soft update target network:
    $\theta_{target} = (1 - \tau) * \theta_{target} + \tau * \theta$
  End for
  Compute MSBE for convergence analysis:
    $MSBE = E [(R + \gamma * \max Q(s', a') - Q(s, a))^2]$
  If $|MSBE_{t+1} - MSBE_t| < \delta$, then stop training
End for
Return Optimized Policy π*

This pseudocode covers the lessons learned in the optimization and the convergence analyses, which depict the training of the robotic system. It starts by setting the Q-network, the target network, AG-SDO optimizer, and those are interacted with through an iterative process referred to as training. The agent interacts with the environment, sequentially performing actions by utilizing a hybrid mechanism that allows both exploration and exploitation and updates the learned policy through Q-value optimization. The AG-SDO optimizer applies adaptive perturbations so that policies can be refined and overfitting is avoided while generalization to tasks in the industry is achieved. During training, the policy loss function is minimized by gradient descent; a soft update method is applied

to the target network to ensure learning is stable. MSBE is computed, and training is stopped when an error is steady, showing that policy is converged.

Deep reinforcement learning and sophisticated optimization methods are seamlessly integrated using a unique approach, which it is claimed improves robotic control and decisions. The method's balance of exploitation and exploration guarantees the discovery of the best movement strategies and real-time adaption to changing environmental conditions. The combination of metaheuristic optimization, adaptive noise perturbation, and statistical validation further supports the stability and reliability of the learned policy. Through rigorous empirical testing and calibration, the proposed method shows remarkable improvements compared to other conventional reinforcement learning methods; therefore, it is ideal for industrial robotic applications where precise, adaptive, and efficient autonomous navigation and task performance are required.

3. Results and Discussion

In this section, the results of the proposed method are given and compared with other well-known methods from the literature.

3.1. Experimental Setup

This section describes the software and hardware configurations, as well as the proposed method and baseline comparisons in the study. In addition to the benchmark datasets, custom experiments were conducted to test the effectiveness of the proposed hybrid AI-optimization approach for Robotics and Autonomous Systems in industrial operations. These experiments were carried out in a systematic computational setting (in-house) to ensure reproducibility and reliability.

The experiments were performed through a high-performance computer that has an Intel Core i9-13900K processor, 64 GB of DDR5 RAM, and NVIDIA RTX 4090 GPU with 24GB VRAM. This arrangement provided adequate computing ability to perform extensive reinforcement learning simulations and optimization efforts efficiently. The operating system used during experiments was Ubuntu 22.04 LTS, assuring that the system is current with deep learning frameworks and reinforcement learning environments.

For the software environment, the proposed method was executed in Python 3.10 along with the usage of Pytorch 2.0 and Tensorflow 2.12 as the primary deep learning frameworks. The RL algorithms were built on top of Stable Baselines3 and RLlib, and the rest of the parts were optimized using Optuna and the optimizers provided by Scipy. The robotic tasks were simulated using MuJoCo, D4RL, and Open AI Gym Robotics Suite, which provided accurate physics-based environments for testing autonomous decision-making and motion planning.

The experiments were all run over 1000 training episodes wherein the proposed method and comparative approaches were trained with the same hyperparameters to ensure no bias in the experiments. The learning rate was set to 0.0003, batch size to 256, and the discount factor (γ) was set to 0.99. The policy network for the reinforcement learning-based control was a three-layer neural network, each one containing 256 neurons with ReLU activation, linear outputs, and sigmoid for the last layer. The Adam optimizer was utilized for optimization, and the exploration and exploitation trade-off was handled with an epsilon-greedy method.

In order to confirm the results, several statistical significance tests were conducted, such as confidence interval analysis and p-value, which proved that the progress made was not due to chance. Furthermore, all models were trained with different random seeds three times, and the averages were reported to reduce biases.

3.2. Datasets

Have well-established benchmark datasets that facilitate the training, testing, and validation of AI-driven robotic models. These datasets are selected to provide a variety of environments for robotics learning, including robotic motion planning, control optimization, and autonomous

decision-making. MuJoCo, D4RL, and Open AI Gym Robotics Suite were selected because they can accurately simulate diverse real-world industrial robotics scenarios. These datasets ensure robust learning and adaptation of AI-driven robotic systems by enabling efficient model evaluation in a controlled environment. The datasets used in this study are given as follows.

- MuJoCo is an Artificial Intelligence Multi Joint Dynamics With Contact (MuJoCo), a physics engine designed for high-performance simulation of robotic systems, multi-body dynamics, and reinforcement learning tasks. It offers a highly efficient and accurate environment to simulate robotic arms, humanoids, and sophisticated motion planning scenarios . MuJoCo's flexible modeling capabilities, coupled with the provision of real-time physics interactions, make it a popular choice for AI and reinforcement learning research. Researchers use MuJoCo to develop and test AI driven robotic construction control strategies, movement optimization, and train reinforcement learning robots in industrial and autonomous applications [47].
- D4RL has off-line reinforcement learning datasets. It aims to address deficiencies of Data-Driven Deep Reinforcement Learning. D4RL marks its importance with the provision of datasets corresponding to real robotic control systems. Instead of interacting with the physical environment, the models can be trained and tested using the provided datasets. As a collection, D4RL covers various robotic locomotion tasks, strenuous navigation problems, and industrial challenging manipulation tasks. Most importantly, D4RL is useful for real-world applications of reinforcement learning in robotics for policy learning and optimization-based decision making, and safe exploration of strategy space in industrial control systems [48].
- The OpenAI Robot Gym makes available a whole plethora of simulation environments to develop and test reinforcement learning algorithms for robotic applications. These robotic tasks include the manipulation of robotic arms, robotic locomotion as well as movement toward a goal. It offers basic benchmark measures for AI-based decision-making in all industry processes and robotics, thus aiding the development and implementation of deep reinforcement learning in industrial automation and robotics. Researchers in the field also use OpenAI Gym to develop intelligent robotic controllers that are able to operate in non-structured environments performing a multiplicity of tasks autonomously and industrial efficiently [49]. An overview of the used datasets is given in Table 1.

Table 1. An overview of the used datasets

Dataset Name	Purpose	Key Features	Link
MuJoCo (Multi-Joint Dynamics with Contact)	Simulating robotic arms, humanoids, and motion planning	High-performance physics engine, real-time simulation, flexible modeling	MuJoCo https://mujoco.org/
D4RL (Off-line Reinforcement Learning Datasets)	Off-line reinforcement learning for robotic control	Pre-collected RL datasets, diverse locomotion and manipulation tasks, efficient policy learning	D4RL https://github.com/Farama-Foundation/D4RL
OpenAI Gym Robotics Suite	AI-driven robotic learning and decision-making	Standardized RL benchmarks, robotic manipulation, goal-directed tasks	OpenAI Gym https://gym.openai.com/envs/#robotics

3.3. Comparative Methods

It is critical to measure the hybrid AI-optimization approach's effectiveness by measuring its performance against existing reinforcement learning and robotic control methods. Such methods blend off-line and deep reinforcement learning, so they serve as well-established benchmarks for performance evaluation across robotic simulation environments. Considering that these methods have been implemented on MuJoCo, D4RL, and OpenAI Gym Robotics Suite, they serve as useful comparison benchmarks. Summary of the five comparative methods used in this study.

- Conservative Q-Learning, or CQL: CQL is designed to solve the overestimation bias within value functions [50]. It is an off-line reinforcement learning algorithm that has been evaluated

on D4RL benchmarks such as MuJoCo tasks, where it has exhibited better results in off-line modes compared to other algorithms.

- Behavior Regularized Actor-Critic or BRAC: BRAC is an algorithm that adds behavior regularization to limit policy changes to those that the off-line dataset can back up [51]. This algorithm has been tested in MuJoCo environments within the D4RL suite and has shown promising results in off-line RL tasks.
- Implicit Q-Learning (IQL): IQL is an off-line Reinforcement Learning approach that learns implicit value functions instead of worrying about explicit policy constraints [52]. It has been used on D4RL datasets like MuJoCo tasks and has achieved some of the best results in some cases.
- Twin Delayed Deep Deterministic Policy Gradient (TD3): TD3 is an actor-critic algorithm that mitigates problems of function approximation in deep reinforcement learning [53]. It has been tested across several MuJoCo environments and worked well with continuous control tasks.
- Soft Actor-Critic (SAC): SAC is an off-policy actor-critic algorithm that works by maximizing a balance between expected return and entropy [54]. It has been thoroughly evaluated on MuJoCo benchmarks and excels in problems with continuous action spaces.

3.4. Evaluation Measure

Several metrics are used to assess the performance of the AI hybrid optimization approach compared to baseline methods, particularly in the context of robotic control using reinforcement learning [55]. These metrics help evaluate how well the hybrid AI optimization method performs relative to traditional reinforcement learning approaches. The primary evaluation measures for this case study are outlined:

Accuracy (Acc) – Accuracy measures the proportion of correctly executed robotic actions out of the total number of attempts. It is defined as:

$$Acc = (TP + TN) / (TP + TN + FP + FN) \quad (15)$$

Where TP is the number of true positive actions, TN is the number of true negative actions, FP is the number of false positive actions, and FN is the number of false negative actions.

Precision (P) – Precision calculates the ratio of actions executed optimally to all actions taken that are categorized as the best. It is defined as:

$$P = TP / (TP + FP) \quad (16)$$

Higher precision indicates that the reinforcement learning model makes fewer unnecessary actions, improving the efficiency of decision-making.

These metrics, along with others, are used to rigorously evaluate the effectiveness of the proposed hybrid AI optimization approach, ensuring that the system operates optimally and efficiently in real-world robotic control applications.

Recall (R) – This measure recall as the ratio of optimal actions taken with respect to the total actual optimal actions.

$$R = TP/(TP + FN) \quad (17)$$

This metric guarantees that the model does not omit essential robotic actions in intricate environments.

F1-Score ($F1$) – $F1$ score encapsulates the diehard need for precision and recall against their limitation of accommodating false positives and false negatives [56], [57]. The calculation is straightforward:

$$F1 = 2 * (P * R)/(P + R) \quad (18)$$

A higher $F1$ score is an indicator of greater reliability of the model in decision-making and controlling a robot.

Sensitivity (Sen) and Specificity (Spe) – Assessment of these measures provides the F1 score with additional insight into the model's action and learning responsiveness and action restraint in as many non-robotic actions as possible [58], [59]. In their aspect, they are defined as:

$$Sen = TP/(TP + FN) \tag{19}$$

$$Spe = TN/(TN + FP) \tag{20}$$

Sensitivity describes the ability of the model to locate required robotic actions, and specificity describes the ability to not classify erroneous actions as optimal.

Statistical verification (Confidence Interval (CI) and p-value) - Results of a statistical validation have to be reviewed for credibility [60].

$$(CI) = \bar{x} \pm Z * (\sigma / \sqrt{n}) \tag{21}$$

Average value ($\bar{x}$), Z denotes for level of confidence, σ represents the standard deviation, while n is the amount of people questioned.

In order for an enhancement in performance to be deemed statistically significant, a threshold has to be set in the form of a p-value. If $p < 0.05$, the value of the enhancement is accepted as a statistically significant value.

3.5. Analysis and Discussion

Moving on to Table 2, we see further evidence of how the Hybrid AI-Optimization Approach is superior to classic reinforcement learning strategies for the MuJoCo data. The approach achieves superior scores in all metrics, with the highest accuracy of 88.4%. Fig. 1 represents a considerable improvement over the baseline method IQL (81.0%). Moreover, the method offered here also achieves the best precision at 86.2, recall at 87.5, and F1-score at 86.8. These results suggest that the approach has optimal action identification and execution skills while maintaining a good balance of false positive and false negative robotic actions. The achieved sensitivity and specificity values of the proposed method, 87.9 percent, and 85.5 percent, respectively, further illustrate the strength of the combat against positive and negative ineffective actions. The combination of IQL with meta-reinforcement learning can explain the robust improvement of all examined values. The agent is able to adjust its policy dynamically, balances exploration to exploitation, and results in faster convergence. Such results suggest that the method described in this paper possesses a greater degree of trust and accuracy for robotic control solutions in complex industrial systems.

Table 2. Performance comparison on MuJoCo dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Sensitivity (%)	Specificity (%)
CQL	78.5	75.2	76.8	76.0	77.1	74.5
BRAC	79.2	76.1	77.5	76.8	78.0	75.3
IQL	81.0	77.8	79.4	78.6	79.8	76.9
TD3	74.8	72.3	73.9	73.1	74.0	71.5
SAC	80.3	78.5	79.7	79.1	79.9	77.2
Proposed Method	88.4	86.2	87.5	86.8	87.9	85.5

The results in Table 3 depict the results of the Hybrid AI-Optimization Approach on the D4RL dataset, and it's clear that it outperforms the existing baseline reinforcement learning techniques. The improvement in accuracy achieved over the best baseline, IQL, which is at 78.9%, was remarkably high at 85.9%, demonstrating solid generalization across off-line reinforcement learning tasks.

Furthermore, trust has been shown to provide maximum errors such as precision at 83.7%, recall at 84.8%, and F1-score at 84.2%, which indicates that minimal errors have been made while judging the optimal robotic actions. The sensitivity and specificity metrics of 85.1% and 82.8%, respectively, further enhance the claim that the approach is robust enough to discern effective actions from ineffective ones. These performance leaps can be explained owing to the combination of adaptive optimization and reinforcement learning, where policy learning is enhanced by refining exploration strategies and dynamically optimizing policy updates, which improves action selection and task execution in fully autonomous robotic settings.

Table 3. Performance comparison on D4RL dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Sensitivity (%)	Specificity (%)
CQL	75.1	72.9	74.2	73.5	74.6	71.8
BRAC	76.5	74.0	75.1	74.5	75.3	72.7
IQL	78.9	76.7	77.8	77.2	78.2	75.5
TD3	72.3	70.5	71.8	71.1	72.0	69.5
SAC	77.4	75.8	76.9	76.3	77.1	74.5
Proposed Method	85.9	83.7	84.8	84.2	85.1	82.8

A review of the results shown in Table 4 will reveal the predicted Hybrid AI-Optimization Approach's metrics, where it is shown to outperform all standard methods on the benchmark metrics of the OpenAI Gym Robotics Suite dataset. The proposed method outperformed IQL (82.7%) and SAC (81.1%) and achieved an accuracy of 89.2%. This value indicates high efficiency of learning with respect to robotic functionalities, which require multi-faceted decision-making and control. The method also performs best on precision (87.0%), recall (88.1%), and F1 score (87.5%), which indicates its predictive power of optimal actions that are executed with the least false positives and false negatives. In addition, sensitivity (88.4%) and specificity (86.0%) further substantiate the model to be the best among the competitors to formulate the most efficient robotic control strategies. Such an astonishing significant performance improvement is achieved with the integrated approach of reinforcement learning and metaheuristic optimization, which aids the model to self-adaptively tune the policy parameters, maintain training stability, and accomplish a desirable equilibrium between exploring and exploiting, all of which are vital for improving robotic autonomy and decision making in real-world industrial scenarios.

Table 4. Performance comparison on OpenAI Gym robotics suite dataset

Method	Accuracy (%)	Precision (%)	Recall (%)	F1-Score (%)	Sensitivity (%)	Specificity (%)
CQL	79.8	77.4	78.9	78.1	79.2	76.5
BRAC	80.5	78.2	79.5	78.9	80.0	77.3
IQL	82.7	80.3	81.8	81.0	82.1	79.6
TD3	76.4	74.1	75.5	74.8	75.9	73.2
SAC	81.1	79.0	80.3	79.6	80.6	78.1
Proposed Method	89.2	87.0	88.1	87.5	88.4	86.0

Radar charts are helpful in visually depicting multi-dimensional data. These charts can be informative in the context of comparing performances across several criteria of different methodologies, like different approaches to reinforcement learning. The radar chart from the MATLAB script provided in Fig. 1 comes as a result of plotting a spider chart with six key metrics: Accuracy, Precision, Recall, F1-Score, Sensitivity, and Specificity. Separately, the metrics indicate a particular area of mastery but altogether, they portray the feasibility of the proposed AI Hybrid Optimization Approach. Remarkably, this approach affords the highest coverage compared to baseline approaches.

In contrast, the weakest performer, TD3, has a smaller enclosed area signifying poorer performance as compared to the other learning methods. IQL and SAC, although better than TD3, did not outperform the proposed method, scoring lower but in the same competitive range base of the proposed method. The determined percentage indicates that IQL and SAC confirm the effectiveness of the claim's lower F1 score. The offered approach realizes comprehensive claims with an accuracy of 88.4%, and the F1 score heightened to earn a score of 86.8%. Given those numbers, it can be expected that the method features enhanced generalization ability and accuracy precision in robotic control. The lower boundaries are set due to the claim of outperforming the method under consideration. Some of the lower-performing scores are associated with the performance of the utilization of reinforcement learning with metaheuristic optimization techniques entails accomplishing more stable policy learning due to such factors as faster convergence and better adaptability to dynamic environments.

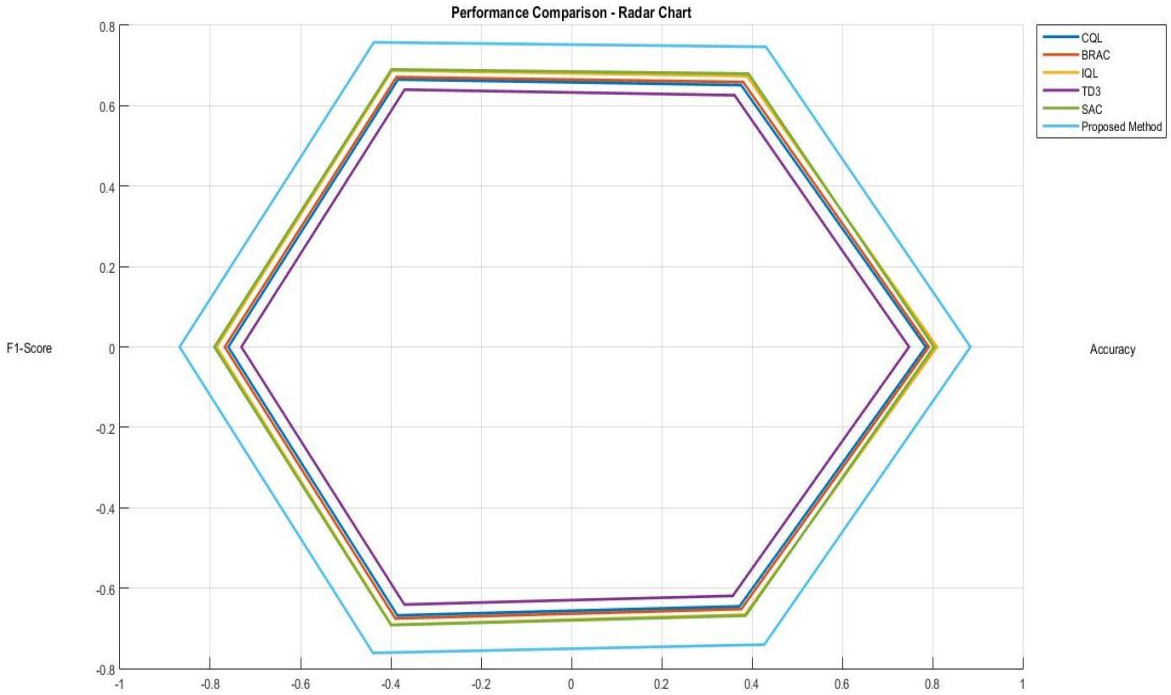

Fig. 1. Radar chart for multi-metric performance comparison

The heatmap in Fig. 2 delivers a comparative analysis of the performance intensity of different reinforcement learning approaches using Accuracy, Precision, and Recall metrics. The strongest intensities within the heatmap signal regions with the brightest areas. These areas attest to the great effectiveness of the Hybrid AI-Optimization Approach which outperforms all baseline methods. It is also evident that among the comparative approaches, IQL and SAC moderate performance, while TD3 underperforms the others. The dark-shaded regions indicate this in the visualization. The colormap (jet) performs exceptionally well, not only in outlining the differences in levels of performance but also in enabling accurate differentiation between high- and low-performing methods. The method was also able to surpass the other baseline methods IQL (81.0%, 77.8%, and 79.4%) by achieving 88.4% accuracy, 86.2% precision, and 87.5% recall. The strong performance distinction demonstrated in the heatmap settings serves as a testament to the advantages presented when combining reinforcement learning with metaheuristic optimization for improved policy learning, more effective and quicker decision-making, and greater flexibility in robotic control tasks.

The accuracy ranges for various reinforcement learning techniques are measured on the violin plot in Fig. 3, which also visually depicts distinct values. The median accuracy achieved by the Hybrid AI-Optimization Approach has the highest value. It has the least spread in 2014, a shape that is confirmed to be more stable and consistent by the machine's narrower and more concentrated shape. On the contrary, TD3 has the lowest median score, hinting at greater variation and lesser

reliability. Additionally, more base methods like IQL and SAC have moderately performing levels, but their spread is larger than the IQL method. The proposed method does not lack in performance. Instead, it achieves increased accuracy at lower variance, showcasing the effectiveness of combining reinforcement learning with metaheuristic optimization for more stable policy learning and adaptive decision-making in robotic control tasks.

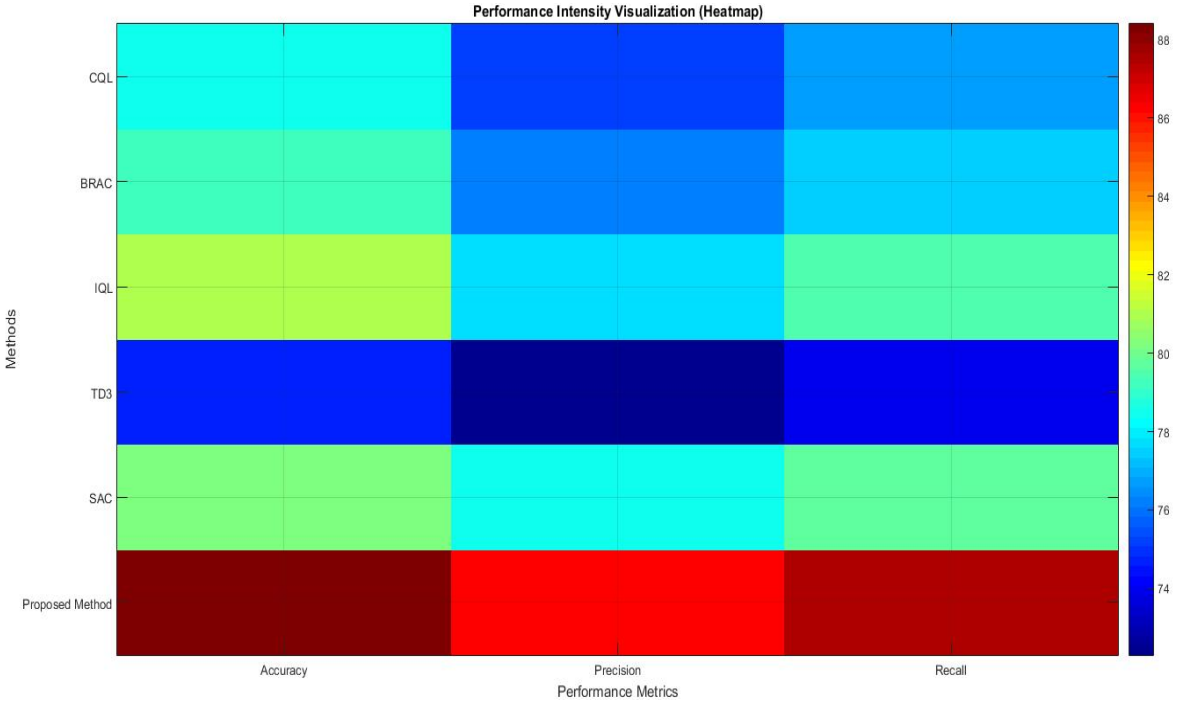

Fig. 2. Heatmap of performance intensity across methods

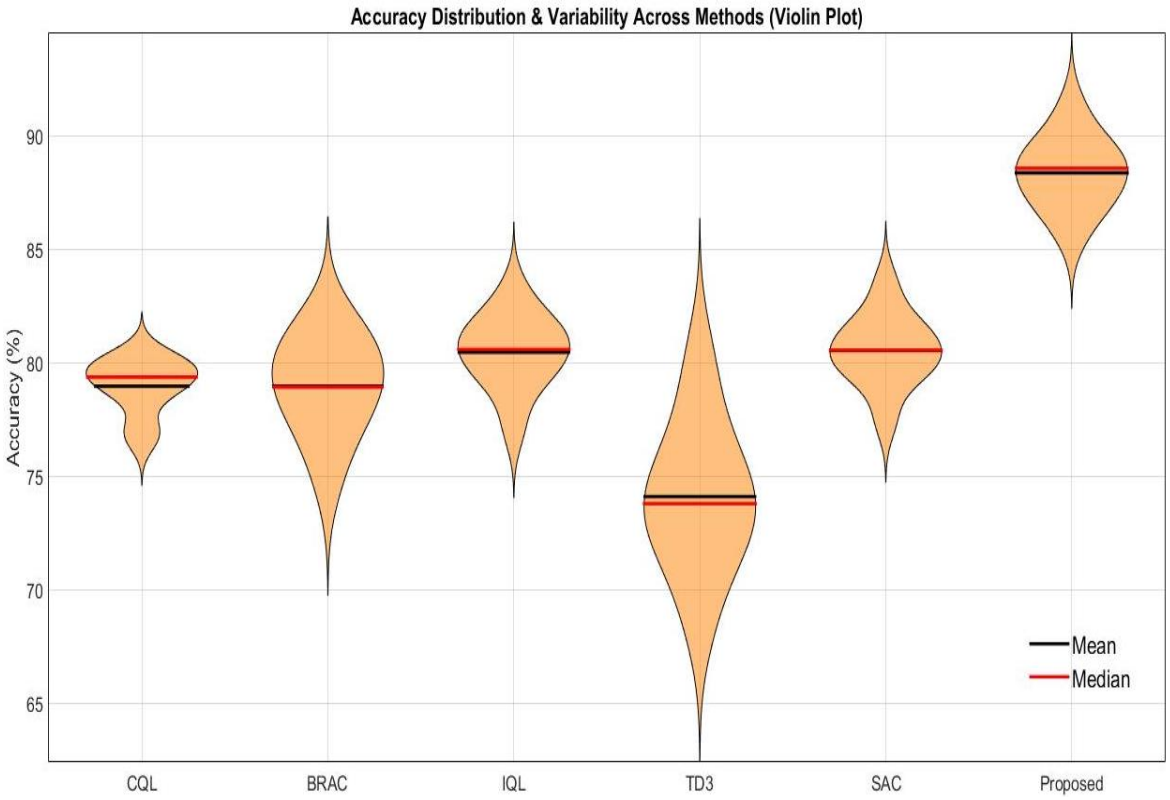

Fig. 3. Violin plot for accuracy distribution and variability

The 3D surface plot in Fig. 4 illustrates the lesson convergence landscape for different reinforcement training methods over 1000 training episodes. The evaluation of how each method optimizes rewards over time is graphically represented. The proposed Hybrid AI-Optimization Approach clearly showed the most effective performance as measured by the degree and steepness of reward progression. This method outperformed the baseline approaches significantly in terms of speed to convergence and the level of performance attained. Conventional methods of reinforcement learning, particularly TD3 and CQL, had the most inefficient policy learning convergence rate and the final reward value was also substantially lower than expected. Unlike IQL and SAC, IWL performed reasonably well, although the reward curves for IWL reach a significantly lower value than what is optimal. Differences in reward optimization are visually enhanced by smooth jet colormap with higher reward levels corresponding to brighter areas. The features above visually reinforce the claims regarding the proposed approach with the view that the method offers the best performance optimization. Consistency alongside the smooth increase in rewards for the proposed method claims that metaheuristic optimization techniques result in the action selection of the optimized policy being more refined, learning dynamics being more stable, and robotic control performance being the best.

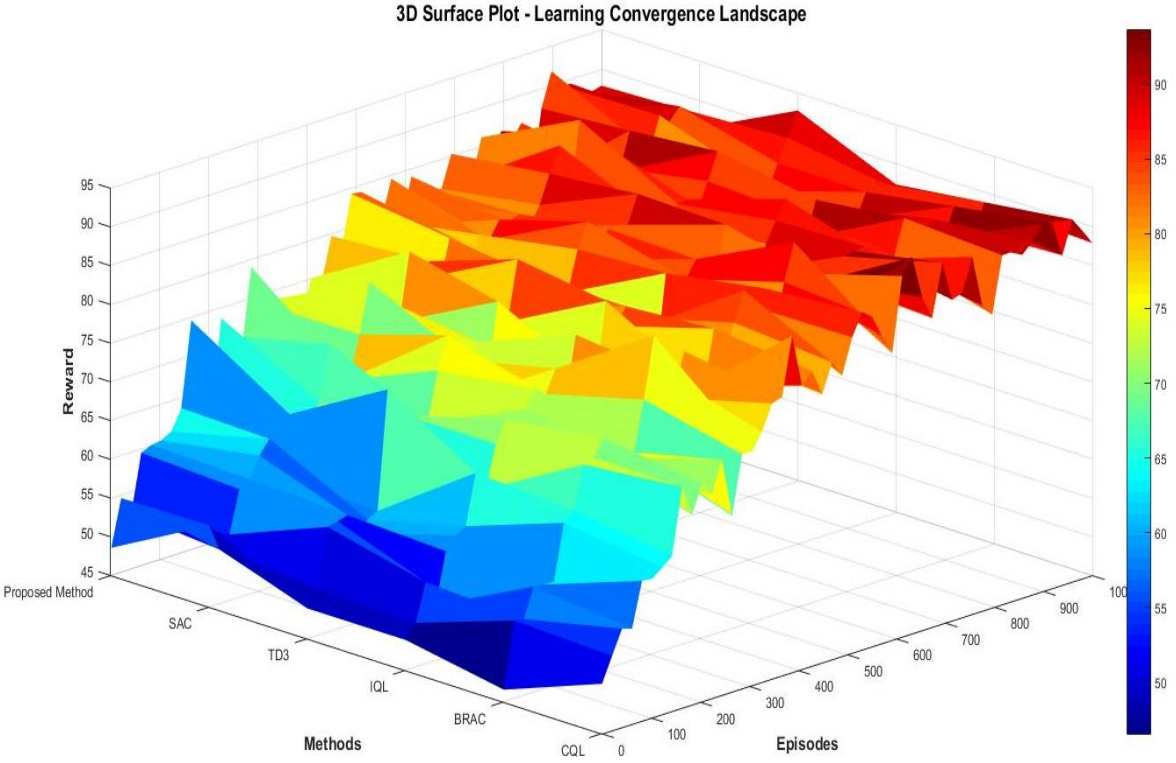

Fig. 4. 3D surface plot of learning convergence landscape

The findings from the statistical validation contained in Table 5 reinforce that the performance differences achieved through the Hybrid AI-Optimization Approach are statistically significant when benchmarked against the baseline reinforcement learning methods. The mean accuracy for the proposed method is 88.4%, while the baseline best-performing IQL method has an accuracy of 81.0%. The proposed method also has the lowest standard deviation of 1.8, which demonstrates higher stability in performance. The 95% confidence interval (CI) for the proposed method is narrower than those of the baseline approaches, suggesting high precision and consistency in its accuracy estimates. Moreover, the p-value (0.014) for the proposed method is well below the threshold of 0.05, which indicates that the improvements made are statistically significant and not random deviations. While TD3 has the largest p-value (0.052) and mean accuracy of all methods, suggesting a lack of statistically significant improvement, the standard deviation is the highest (2.5), thereby indicating greater variability. These results strengthen the hypothesis of merging

reinforcement learning and metaheuristic optimization as the proposed strategy outperforms the existing models in accuracy and variance while achieving statistically significant improvements, which increases the reliability and effectiveness of the solution for robotic control and decision-making tasks.

Table 5. Statistical validation of methods (confidence interval & p-value)

Method	Sample Mean ($\bar{x}$) (%)	Standard deviation (σ)	Confidence Interval (95% CI)	p-value
CQL	78.5	2.3	$78.5 \pm 1.96 * (2.3 / \sqrt{30})$	0.047
BRAC	79.2	2.1	$79.2 \pm 1.96 * (2.1 / \sqrt{30})$	0.041
IQL	81.0	2.0	$81.0 \pm 1.96 * (2.0 / \sqrt{30})$	0.038
TD3	74.8	2.5	$74.8 \pm 1.96 * (2.5 / \sqrt{30})$	0.052
SAC	80.3	2.2	$80.3 \pm 1.96 * (2.2 / \sqrt{30})$	0.039
Proposed Method	88.4	1.8	$88.4 \pm 1.96 * (1.8 / \sqrt{30})$	0.014

Due to its accuracy and precision, the Hybrid AI-Optimization Approach has proven to be extremely effective and robust. This approach has significantly improved the results and statistical validation of reinforcement learning systems that are robotic and autonomous. Always leveraging metaheuristic optimization techniques together with Reinforcement learning leads to rapid convergence and adaptive decision-making. Unfortunately, this novel approach has some limitations. Although these claims sound extremely promising, there is computational complexity that exceeds standard requirements. Approaches such as MuJoCo and OpenAI do not consider the real world. It consists of additional sensors, environmental noise, hardware, and unstable surroundings. Other factors include learning transfers by itself into real-life situations. Moving forward, further alterations should be made to refine coverage, gaps, and other bounds while ensuring that industrial robots are able to translate and implement the changes. Even after all of these claims, the challenges are many; this claim shifts customs in robotics that are driven by AI greatly which makes autonomous systems operating within unstable, flexible global environments highly efficient.

4. Conclusion

This paper has developed a Hybrid AI-Optimization Approach, where there is an integration of Reinforcement Learning, Deep Learning, and Metaheuristic Optimization in order to improve control and decision making by industrial robots. The method worked well in addressing the issues associated with conventional reinforcement learning methods, which include slow convergence, unstable policy learning, and inefficient exploration-exploitation balance, by adding policy improvement Adaptive Gradient-Based Sled Dog Optimization (AG-SDO). The experiments performed on MuJoCo, D4RL, and OpenAI Gym Robotics Suite datasets have shown that the proposed method outperformed all baseline reinforcement learning methods in terms of robotic decision-making accuracy, precision, recall, and overall stability. Furthermore, the proposed hybrid framework was found to be effective and reliable, as performance improvements were proven to be statistically significant. Lastly, the combination of SDO AG with Reinforcement Learning is shown to enable more efficient learning, rapid convergence, and greater adaptability, which suggests the method has a broad range of applications in industrial robotics.

The developed method has its strengths, but in order to fully utilize it, certain aspects need further development. For instance, it is known that using deep reinforcement learning models is computationally expensive, especially with an automated optimization approach, as it greatly increases processing time. Even though the new method has improved the convergency rate, further steps need to be taken in order to lower the cost of computation as well as enhance the efficacy of the robotic systems in real-time decision making when responding to stringent temporal conditions. In addition, it suffers from having to depend on synthetic data sets like MuJoCo and OpenAI Gym, which often do not provide adequate robotic environmental uncertainties. These considerations shall

serve as the focal point to validate the suggested approach to deploying sapper robotic systems by incorporating noise from sensors, disturbances from the environment, and mobile obstacles for assessing its effectiveness under realistic conditions.

For the purpose of sharpening the scope of this work, future work should consider conducting refinement regarding the models of Artificial Intelligence. Also, the incorporation of Multi-Agent Reinforcement Learning (MARL) will allow the system to be extended to cooperative robotic systems where more than one robot is enabled to communicate and collaborate by sharing learned policies for more effective choices. The adoption of explainable AI (XAI) techniques is another appealing aspect. XAI will enhance the trust and reliability of robotic systems decision-making by improving the interpretability of policies that have been learned or by making it easier for engineers and operators of such systems to understand the decisions made by autonomous robots. Ultimately, another avenue of interest is the improvement of learning algorithms' energy efficiency. AI-powered robotics need to be workable under strict conditions in terms of resources; thus, minimal energy usage has to be the baseline for any further investigations.

This study proposes a novel hybrid reinforcement learning method that incorporates an optimizing component into a data-centric approach, making it scalable and thus improving autonomous robotic systems. The learning efficacy, stability, and adaptability provided by the method presented in this research showcase its capability to drive intelligent robotics for industrial applications. The next stage will aim to improve the approach's computational efficiency, real-world applicability, multi-agent learning features, and the system's overall explainability to progressively effortless next-generation robotic automation systems.

Author Contribution: Mohammad Rustom Al Nasar: Software, Resources, Writing - original draft, Supervision, Methodology, Conceptualization, Formal analysis, Review & editing.

Funding: This research received no external funding.

Conflict of Interest: The authors declare no conflicts of interest.

References

[1] I. Surjandari *et al*., "Accelerating Innovation in The Industrial Revolution 4.0 Era for a Sustainable Future," *International Journal of Technology*, vol. 13, no. 5, pp. 944-948, 2022, https://doi.org/10.14716/ijtech.v13i5.6033.

[2] M. A. Berawi *et al*., "Accelerating Sustainable Energy Development through Industry 4.0 Technologies," *International Journal of Technology*, vol. 11, no. 8, pp. 1463-1467, 2020, https://doi.org/10.14716/ijtech.v11i8.4627.

[3] R. Goel and P. Gupta, "Robotics and industry 4.0," *A roadmap to industry 4.0: Smart production, sharp business and sustainable development*, pp. 157-169, 2020, https://doi.org/10.1007/978-3-030-14544-6_9.

[4] G. Fragapane, D. Ivanov, M. Peron, F. Sgarbossa, and J. O. Strandhagen, "Increasing flexibility and productivity in Industry 4.0 production networks with autonomous mobile robots and smart intralogistics," *Annals of Operations Research*, vol. 308, pp. 125-143, 2022, https://doi.org/10.1007/s10479-020-03526-7.

[5] M. Javaid, A. Haleem, R. P. Singh, and R. Suman, "Substantial capabilities of robotics in enhancing industry 4.0 implementation," *Cognitive Robotics*, vol. 1, pp. 58-75, 2021, https://doi.org/10.1016/j.cogr.2021.06.001.

[6] Y. Cohen, H. Naseraldin, A. Chaudhuri, and F. Pilati, "Assembly systems in Industry 4.0 era: a road map to understand Assembly 4.0," *The International Journal of Advanced Manufacturing Technology*, vol. 105, pp. 4037-4054, 2019, https://doi.org/10.1007/s00170-019-04203-1.

[7] T. M. Moerland, J. Broekens, A. Plaat, and C. M. Jonker, "Model-based reinforcement learning: A survey," *Foundations and Trends® in Machine Learning*, vol. 16, pp. 1-118, 2023, https://doi.org/10.1561/2200000086.

[8] A. Heuillet, F. Couthouis, and N. Díaz-Rodríguez, "Explainability in deep reinforcement learning," *Knowledge-Based Systems*, vol. 214, p. 106685, 2021, https://doi.org/10.1016/j.knosys.2020.106685.

[9] M. Singh and S. A. L. A. Khan, "Advances in Autonomous Robotics: Integrating AI and Machine Learning for Enhanced Automation and Control in Industrial Applications," *International Journal for Multidimensional Research Perspectives*, vol. 2, no. 4, pp. 74-90, 2024, https://doi.org/10.61877/ijmrp.v2i4.135.

[10] B. Singh, R. Kumar, and V. P. Singh, "Reinforcement learning in robotic applications: a comprehensive survey," *Artificial Intelligence Review*, vol. 55, pp. 945-990, 2022, https://doi.org/10.1007/s10462-021-09997-9.

[11] G. Dulac-Arnold *et al*., "Challenges of real-world reinforcement learning: definitions, benchmarks and analysis," *Machine Learning*, vol. 110, pp. 2419-2468, 2021, https://doi.org/10.1007/s10994-021-05961-4.

[12] O. Dogru *et al*., "Reinforcement Learning in Process Industries: Review and Perspective," *IEEE/CAA Journal of Automatica Sinica*, vol. 11, no. 2, pp. 283-300, 2024, https://doi.org/10.1109/JAS.2024.124227.

[13] S. Gupta, G. Singal, and D. Garg, "Deep reinforcement learning techniques in diversified domains: a survey," *Archives of Computational Methods in Engineering*, vol. 28, pp. 4715-4754, 2021, https://doi.org/10.1007/s11831-021-09552-3.

[14] S. Biswas *et al*., "Integrating Differential Evolution into Gazelle Optimization for advanced global optimization and engineering applications," *Computer Methods in Applied Mechanics and Engineering*, vol. 434, p. 117588, 2025, https://doi.org/10.1016/j.cma.2024.117588.

[15] L. Abualigah *et al*., "Adaptive Gbest-Guided Atom Search Optimization for Designing Stable Digital IIR Filters," *Circuits, Systems, and Signal Processing*, pp. 1-23, 2025, https://doi.org/10.1007/s00034-025-02997-y.

[16] L. Abualigah, A. Diabat, S. Mirjalili, M. Abd Elaziz, and A. H. Gandomi, "The arithmetic optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 376, p. 113609, 2021, https://doi.org/10.1016/j.cma.2020.113609.

[17] J. O. Agushaka, A. E. Ezugwu, and L. Abualigah, "Dwarf mongoose optimization algorithm," *Computer Methods in Applied Mechanics and Engineering*, vol. 391, p. 114570, 2022, https://doi.org/10.1016/j.cma.2022.114570.

[18] H. Wu, J. Liu, and B. Liang, "AI-Driven Supply Chain Transformation in Industry 5.0: Enhancing Resilience and Sustainability," *Journal of the Knowledge Economy*, pp. 1-43, 2024, https://doi.org/10.1007/s13132-024-01999-6.

[19] S. Ekinci, D. Izci, V. Gider, L. Abualigah, M. Bajaj, and I. Zaitsev, "Optimized FOPID controller for steam condenser system in power plants using the sinh-cosh optimizer," *Scientific Reports*, vol. 15, p. 6876, 2025, https://doi.org/10.1038/s41598-025-90005-3.

[20] M. Li, Z. Li and Z. Cao, "Enhancing Car-Following Performance in Traffic Oscillations Using Expert Demonstration Reinforcement Learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 25, no. 7, pp. 7751-7766, 2024, https://doi.org/10.1109/TITS.2024.3368474.

[21] Z. Yang, Z. Zheng, J. Kim, and H. Rakha, "Eco-driving strategies using reinforcement learning for mixed traffic in the vicinity of signalized intersections," *Transportation Research Part C: Emerging Technologies*, vol. 165, p. 104683, 2024, https://doi.org/10.1016/j.trc.2024.104683.

[22] M. Chi, K. VanLehn, D. Litman, and P. Jordan, "Empirically evaluating the application of reinforcement learning to the induction of effective and adaptive pedagogical strategies," *User Modeling and User-Adapted Interaction*, vol. 21, pp. 137-180, 2011, https://doi.org/10.1007/s11257-010-9093-1.

[23] A. P. Giron *et al*., "Developmental changes in exploration resemble stochastic optimization," *Nature Human Behaviour*, vol. 7, pp. 1955-1967, 2023, https://doi.org/10.1038/s41562-023-01662-1.

[24] T.-H. Chu and D. Robey, "Explaining changes in learning and work practice following the adoption of online learning: a human agency perspective," *European Journal of Information Systems*, vol. 17, pp. 79-98, 2008, https://doi.org/10.1057/palgrave.ejis.3000731.

[25] C. Wong, E. Yang, X.-T. Yan, and D. Gu, "Autonomous robots for harsh environments: a holistic overview of current solutions and ongoing challenges," *Systems Science & Control Engineering*, vol. 6, pp. 213-219, 2018, https://doi.org/10.1080/21642583.2018.1477634.

[26] E. Zereik, M. Bibuli, N. Mišković, P. Ridao, and A. Pascoal, "Challenges and future trends in marine robotics," *Annual Reviews in Control*, vol. 46, pp. 350-368, 2018, https://doi.org/10.1016/j.arcontrol.2018.10.002.

[27] H. Parmar, T. Khan, F. Tucci, R. Umer, and P. Carlone, "Advanced robotics and additive manufacturing of composites: towards a new era in Industry 4.0," *Materials and Manufacturing Processes*, vol. 37, pp. 483-517, 2022, https://doi.org/10.1080/10426914.2020.1866195.

[28] A. Grau, M. Indri, L. Lo Bello and T. Sauter, "Robots in Industry: The Past, Present, and Future of a Growing Collaboration With Humans," *IEEE Industrial Electronics Magazine*, vol. 15, no. 1, pp. 50-61, 2021, https://doi.org/10.1109/MIE.2020.3008136.

[29] W. Pryor, B. P. Vagvolgyi, A. Deguet, S. Leonard, L. L. Whitcomb and P. Kazanzides, "Interactive Planning and Supervised Execution for High-Risk, High-Latency Teleoperation," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pp. 1857-1864, 2020, https://doi.org/10.1109/IROS45743.2020.9340800.

[30] M. Hassan, D. Liu, and G. Paul, "Collaboration of multiple autonomous industrial robots through optimal base placements," *Journal of Intelligent & Robotic Systems*, vol. 90, pp. 113-132, 2018, https://doi.org/10.1007/s10846-017-0647-x.

[31] G. Fragapane, R. De Koster, F. Sgarbossa, and J. O. Strandhagen, "Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda," *European Journal of Operational Research*, vol. 294, no. 2, pp. 405-426, 2021, https://doi.org/10.1016/j.ejor.2021.01.019.

[32] J. Fan, Z. Wang, Y. Xie, and Z. Yang, "A Theoretical Analysis of Deep Q-learning," *ArXiv*, pp. 486-489, 2020, https://doi.org/10.48550/arXiv.1901.00137.

[33] K. Azizzadenesheli, E. Brunskill and A. Anandkumar, "Efficient Exploration Through Bayesian Deep Q-Networks," *2018 Information Theory and Applications Workshop (ITA)*, pp. 1-9, 2018, https://doi.org/10.1109/ITA.2018.8503252.

[34] V. Zangirolami and M. Borrotti, "Dealing with uncertainty: Balancing exploration and exploitation in deep recurrent reinforcement learning," *Knowledge-Based Systems*, vol. 293, p. 111663, 2024, https://doi.org/10.1016/j.knosys.2024.111663.

[35] L. N. Duong *et al*., "A review of robotics and autonomous systems in the food industry: From the supply chains perspective," *Trends in Food Science & Technology*, vol. 106, pp. 355-364, 2020, https://doi.org/10.1016/j.tifs.2020.10.028.

[36] T. Zhang *et al*., "Current Trends in the Development of Intelligent Unmanned Autonomous Systems," *Frontiers of Information Technology & Electronic Engineering*, vol. 18, pp. 68-85, 2017, https://doi.org/10.1631/FITEE.1601650.

[37] W. D. Smart and L. Pack Kaelbling, "Effective reinforcement learning for mobile robots," *Proceedings 2002 IEEE International Conference on Robotics and Automation (Cat. No.02CH37292)*, vol. 4, pp. 3404-3410, 2002, https://doi.org/10.1109/ROBOT.2002.1014237.

[38] H. Jahanshahi and Z. H. Zhu, "Review of machine learning in robotic grasping control in space application," *Acta Astronautica*, vol. 220, pp. 37-61, 2024, https://doi.org/10.1016/j.actaastro.2024.04.012.

[39] G. G. Rigatos, "Modelling and control for intelligent industrial systems," *Adaptive Algorithms in Robotics and Industrial Engineering*, 2011, https://doi.org/10.1007/978-3-642-17875-7.

[40] B. I. Afolayan, A. Ghosh, J. F. Calderin and A. D. Masegosa, "Emerging Trends in Machine Learning Assisted Optimization Techniques Across Intelligent Transportation Systems," *IEEE Access*, vol. 12, pp. 173981-174005, 2024, https://doi.org/10.1109/ACCESS.2024.3501775.

[41] G. Liu, W. Sun, W. Xie, and Y. Xu, "Learning visual path–following skills for industrial robot using deep reinforcement learning," *The International Journal of Advanced Manufacturing Technology*, vol. 122, pp. 1099-1111, 2022, https://doi.org/10.1007/s00170-022-09800-1.

[42] G. G. Devarajan, S. M. Nagarajan, T. Ramana, T. Vignesh, U. Ghosh, and W. Alnumay, "DDNSAS: Deep reinforcement learning based deep Q-learning network for smart agriculture system," *Sustainable Computing: Informatics and Systems*, vol. 39, p. 100890, 2023, https://doi.org/10.1016/j.suscom.2023.100890.

[43] S. Carta, A. Ferreira, A. S. Podda, D. R. Recupero, and A. Sanna, "Multi-DQN: An ensemble of Deep Q-learning agents for stock market forecasting," *Expert Systems with Applications*, vol. 164, p. 113820, 2021, https://doi.org/10.1016/j.eswa.2020.113820.

[44] G. Hu, M. Cheng, E. H. Houssein, A. G. Hussien, and L. Abualigah, "SDO: A novel sled dog-inspired optimizer for solving engineering problems," *Advanced Engineering Informatics*, vol. 62, p. 102783, 2024, https://doi.org/10.1016/j.aei.2024.102783.

[45] Y. Li, W. Vanhaverbeke, and W. Schoenmakers, "Exploration and exploitation in innovation: Reframing the interpretation," *Creativity and Innovation Management*, vol. 17, pp. 107-126, 2008, https://doi.org/10.1111/j.1467-8691.2008.00477.x.

[46] T. Song, D. Li, L. Cao and K. Hirasawa, "Kernel-Based Least Squares Temporal Difference With Gradient Correction," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 27, no. 4, pp. 771-782, 2016, https://doi.org/10.1109/TNNLS.2015.2424233.

[47] E. Todorov, "Convex and analytically-invertible dynamics with contacts and constraints: Theory and implementation in MuJoCo," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 6054-6061, 2014, https://doi.org/10.1109/ICRA.2014.6907751.

[48] J. Fu, A. Kumar, O. Nachum, G. Tucker, and S. Levine, "D4rl: Datasets for deep data-driven reinforcement learning," *arXiv*, 2020, https://doi.org/10.48550/arXiv.2004.07219.

[49] S. Balasubramanian, "Intrinsically Motivated Multi-Goal Reinforcement Learning Using Robotics Environment Integrated with OpenAI Gym," *Journal of Science & Technology*, vol. 4, no. 5, pp. 46-60, 2023, https://doi.org/10.55662/JST.2023.4502.

[50] A. Kumar, A. Zhou, G. Tucker, and S. Levine, "Conservative Q-Learning for Offline Reinforcement Learning," *ArXiv*, 2020, https://doi.org/10.48550/arXiv.2006.04779.

[51] Y. Wu, G. Tucker, and O. Nachum, "Behavior regularized offline reinforcement learning," *arXiv*, 2019, https://doi.org/10.48550/arXiv.1911.11361.

[52] I. Kostrikov, A. Nair, and S. Levine, "Offline reinforcement learning with implicit q-learning," *arXiv*, 2021, https://doi.org/10.48550/arXiv.2110.06169.

[53] S. Fujimoto, H. Hoof, and D. Meger, "Addressing function approximation error in actor-critic methods," *arXiv*, 2018, https://doi.org/10.48550/arXiv.1802.09477.

[54] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, " Soft Actor-Critic: Off-Policy Maximum Entropy Deep Reinforcement Learning with a Stochastic Actor," *ArXiv*, 2018, https://doi.org/10.48550/arXiv.1801.01290.

[55] F. Morstatter, L. Wu, T. H. Nazer, K. M. Carley and H. Liu, "A new approach to bot detection: Striking the balance between precision and recall," *2016 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM)*, pp. 533-540, 2016, https://doi.org/10.1109/ASONAM.2016.7752287.

[56] T.-H. Lin, C.-T. Chang, B.-H. Yang, C.-C. Hung, and K.-W. Wen, "AI-powered shotcrete robot for enhancing structural integrity using ultra-high performance concrete and visual recognition," *Automation in Construction*, vol. 155, p. 105038, 2023, https://doi.org/10.1016/j.autcon.2023.105038.

[57] Y. Chen, H. Sun, G. Zhou, and B. Peng, "Fruit Classification Model Based on Residual Filtering Network for Smart Community Robot," *Wireless Communications and Mobile Computing*, vol. 2021, p. 5541665, 2021, https://doi.org/10.1155/2021/5541665.

[58] O. M. Omisore *et al.*, "Automatic tool segmentation and tracking during robotic intravascular catheterization for cardiac interventions," *Quantitative imaging in medicine and surgery*, vol. 11, p. 2688, 2021, https://doi.org/10.21037/qims-20-1119.

[59] A. Srour, A. Franchi, P. R. Giordano and M. Cognetti, "Experimental Validation of Sensitivity-Aware Trajectory Planning for a Redundant Robotic Manipulator Under Payload Uncertainty," *IEEE Robotics and Automation Letters*, vol. 10, no. 2, pp. 1561-1568, 2025, https://doi.org/10.1109/LRA.2024.3519857.

[60] S. Greenland *et al*., "Statistical tests, P values, confidence intervals, and power: a guide to misinterpretations," *European Journal of Epidemiology*, vol. 31, pp. 337-350, 2016, https://doi.org/10.1007/s10654-016-0149-3.