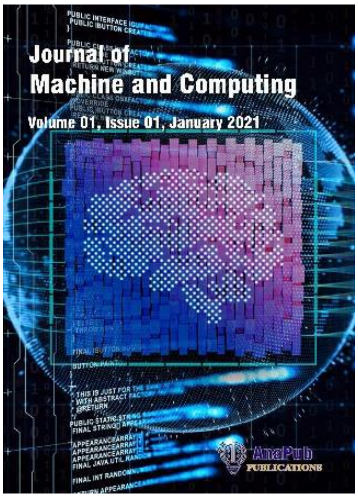
Journal Pre-proof



Robust Ensembled Model Based Reinforcement Learning for better Tomato yield in Greenhouse

Gandhimathi S, Senthilkumaran B, Jude Moses Anto Devakanth J, Nithya K V and Jayaprakash Chinnadurai

DOI: 10.53759/7669/jmc202505090

Reference: JMC202505090

Journal: Journal of Machine and Computing.

Received 13 March 2024

Revised form 30 October 2024

Accepted 15 March 2025

Please cite this article as: Gandhimathi S, Senthilkumaran B, Jude Moses Anto Devakanth J, Nithya K V and Jayaprakash Chinnadurai, "Robust Ensembled Model Based Reinforcement Learning for better Tomato yield in Greenhouse", Journal of Machine and Computing. (2025). Doi: https://doi.org/10.53759/7669/jmc202505090.

Robust Ensembled Model Based Reinforcement Learning for better Tomato yield in Greenhouse

[1]S. Gandhimathi, [2]B.Senthilkumaran, [3]J.Jude Moses Anto Devakanth, [4] K. V. Nithya, [5]Jayaprakash Chinnadurai

[1]Department of Computer Science, Valluvar College of Science and Management (Autonomous), Karur 639003.

[2,5]Department Of Computer Science And Engineering, School Of Computing ,Vel Tech Rangarajan Dr.Sagunthala R&D Institute of Science and Technology ,Chennai, Tamilnadu, India

[3]Department of in Computer Applications ,Madanapalle Institute of Technology & Science, Kadiri Road,Angallu Madanapalle - 517325, Andhra Pradesh, India.

[4]Department of Artificial Intelligence and Data Science, M. Kumarasamy College of Engineering ,Karur, Tamilnadu, India.

gandhimathiphd@gmail.com, skumaran.gac16@gmail.com, drjudede[illegible]j@gmail.com, kvnithyaphd@gmail.com, dr.jayaprakash.cs@gmail.com,

Corresponding Author: B.Senthilkumaran

Abstract

The usage of autonomous greenhouses has become essential in meeting the food demands of the world's expanding population. Finding the right optimization strategy to sustain growth, yield, and profit is one of the major issues with greenhouse production. Addressing this issue effectively requires a combination of advanced technologies, data-driven insights, and innovative management practices. The overarching goal is to maximize crop yield and quality while minimizing input costs and environmental impact. The automated optimizations, which are often implemented using reinforcement learning algorithms, encounter issues with sample efficiency and robustness due to the time-consuming nature of the real-world simulation. Therefore, the goal of this research is to solve these issues by combining the SAC and Q learning algorithms to create an ensemble method. To properly optimize the worst-case inefficient samples, a discrete randomization and dropout module is included. The problem of sample efficiency and resilience is treated as a Mismatch Markov Decision Optimisation problem. The suggested model outperforms the current methods in handling the robustness and sample efficiency issues, according to an experimental evaluation. Additionally, this improvement increased production and maximized net profit.

Keywords: Mismatch Markov decision optimization problem (MMDP), Discrete Randomization, Dropout, Ensembled Q and SAC Reinforcement algorithm (EQSACRL)

1. Introduction

The population of the globe is predicted to rise to 9.7 billion by 2050 [1], having reached 8.0 billion in November 2022. It is anticipated that this figure will peak in the 2080s. The need for food will rise by 56%, according to the meta-analysis published in [2], placing a heavy burden on agricultural land. Due to factors including urbanization, climate change, and restrictions on arable land, the amount of agricultural land is decreasing [3]. As a result, it is crucial to use technical breakthroughs to meet the world's expanding demand for food, and science-based farming has become increasingly popular in recent years. Greenhouse-based agricultural production is one invention that addresses crop production from a scientific standpoint in terms of yield and nutrition. This greenhouse-based agricultural production is powered by autonomous operations in terms of monitoring and control.

With the assistance of the greenhouse, vegetable production as well as the land required to get the yield are very low [4]. Thus, spatial efficiency is great with greenhouse-based agricultural production. However, one of the common concerns with greenhouse-based agricultural production is the amount of emissions. India produced its national climate agreement [5] after the Paris Agreement [6] in 2015, which aims for more than 40% reduction of greenhouse gas emissions by 2030 and more than 90% reduction by 2050. The Indian greenhouses account for an average emission of 2.3 tCO_2e per capita annually, which gives an alarming indication that the Indian greenhouse emissions are increasing, and this necessitates the need to optimize greenhouse agriculture. Thus, the need for current greenhouse agriculture is to increase food production with less energy and emissions.In addition to improving the nation's ecology, optimized greenhouse production can have a positive financial impact on individual farmers. Energy sources account for more than 20% of overall earnings, and by 2021, their price will have increased by 200%. Therefore, having the best greenhouse possible is also essential to increasing individual turnover.

Considering the problem of energy accountability, the net profit for an individual can be increased only with better climate controllers inside the greenhouse, and there are several efforts in the literature for the optimization of climate controllers in the greenhouse. More recently, the usage of artificial intelligence techniques and algorithms for the optimization of greenhouses has become common. Wageningen University & Research is organizing its 4th Autonomous Greenhouse Challenge (AGC) based on the results obtained for other crops like lettuce and cherry tomatoes. Based on the excellent results, outperforming humans was observed when the lettuce crops were grown virtually and physically using fully autonomous algorithms. The approaches adopted by various researchers for the virtual growing of greenhouse plants are based on model predictive control and reinforcement learning [7, 8]. For the optimization of the tomato plants, various control experiments are performed as mentioned in [9] and [10]. To make decisions about crop production in an autonomous manner, as there is a shortage of well-trained crop managers. Thus, artificial intelligence has been used for various activities like pest detection [16], disease detection [11], weed detection [14], stress detection [15], harvesting [13], counting [12], etc. Though significant progress has been made in other detections, plant production control holds limitations in terms of not providing an optimum temperature, light, and other factors in the greenhouse [17]. Consistent efforts are made by various researchers, like the one mentioned in [18]. There is continuous improvement in the optimization of the reinforcement algorithms, which has shown a significant increment in crop production as well as net profit.The problem arises when the optimized algorithms are used in real-time. There are certain practical difficulties associated with the adoption of these reinforcement algorithms in real-time, and they are as follows:

- Due to its potential to harm machines and wipe out crops, sample efficiency is the most frequent issue encountered when trying to train a reinforcement algorithm in the real world. Because of the current sampling efficiency, simulations resort to reinforcement tactics.
- Robustness is the primary problem in real-world greenhouse optimization since disruptions during the training phase might impact the algorithm's effectiveness, creating a gap between simulation and reality.
- The most realistic model for greenhouse simulation and the issues with a single model for all kinds of greenhouses are not considered in this study since they call for greater financial outlays.

Motivated by the reasons outlined in Section 1, the primary goal of this work is to apply the reinforcement learning algorithm to enhance the autonomous greenhouse control system. Sample efficiency and robustness are the two main issues, as discussed in Section 1. If they are optimized using the optimization processes in a simulative setting, they lose their practical value. Therefore, the primary goal of this effort is to close the gap that exists between the simulation and the real world. By making the agents in the reinforcement learning algorithm more efficient, this gap is closed. This is accomplished by exposing the agent to higher performance, which increases its robustness. Expanding the agent operating state is taken into consideration because it is an expensive endeavour to construct a more accurate greenhouse model that performs well in both simulated and real-world environments.

This approach of expanding the agent operating condition is favored because it requires less time and money and is more in line with real-world requirements. As a result, this work primarily addresses the gap between simulation and reality while taking sample efficiency and robustness issues into consideration. This work's focus is limited to tomato plants that are grown in greenhouses. The tomato plant was chosen for adoption because of its rapid growth and widespread distribution around the world. In light of these issues, the research sought to provide answers to the following queries to solve the issues raised:

- Does the reinforcement learning algorithm provide better robustness than the existing methods?
- Do we have an option to make the reinforcement learning algorithm robust both in the training and evaluation conditions?
- What is the contribution of the reinforcement algorithm to getting a better yield?
- For the identification of answers to these questions, a greenhouse simulation environment is used, and some of the novel contributions of this work include:
- With the assistance of the gymnasium interface for reinforcement learning, a parametrized greenhouse simulator is created.
- We exhibit how the implementation of our ensemble model with randomization and dropout can handle the worst-case performance.
- We show how robustness and sample efficiency problems for worst-case performance can be improved using randomization and dropout.

The rest of the paper is structured as follows: Section 2 gives the relevant background about growing tomatoes in a greenhouse, greenhouse control, reinforcement learning, and related works. Section 3 briefs the architecture of the greenhouse simulative environment and the optimization method algorithms for better yield. Section 4 gives an idea of the implementation, important findings, and result discussion. Finally, Section 5 discusses the key findings and gives directions for future research.

2. Related works

This section provides an overview of the literature. The purpose of this study is to learn about the latest advancements in automated greenhouse operations, the relevant history of tomatoes grown in greenhouses, and the latest reinforcement algorithms used to optimize greenhouse operations.

Because of the increased need for food items during these years, greenhouse-based production is greater; thus, the main goal of greenhouses is to produce their maximum yield. Good plant development is the key to achieving maximum production, and photosynthesis is the single process that makes this possible [20]. Only under the ideal conditions of light, temperature, and carbon dioxide (Co2) can photosynthesis take place.During the early days, as mentioned in [21], the greenhouse was controlled using three approaches. The first approach requires manual involvement for switching on and off the valve positions of the light, temperature, etc. The second and third approaches used the PID and fuzzy mechanisms for controlling these valves autonomously. Along with this, the outside weather also plays a pivotal role while performing the optimization of the greenhouses. For example, the cloudy weather requires artificial lighting. For performing these operations automatically, recent advancements in artificial intelligence techniques [22, 23] have played a major role. Two approaches are used to apply AI to greenhouse control. The first perspective is utilizing digital twins to create a simulation environment [25–26]. When deciding which optimal values should be set for the highest yield, this twining is helpful. The twins essentially interact between real and virtual greenhouses [27], which makes this feasible. The expense of this testing is negligible since the parametric variables from the real greenhouse are sent straight into the virtual greenhouse. Since several situations have been verified, it is also possible to find an appropriate model that can manage the yield in addition to cost.

As discussed in Section 1, the problems faced while trying to use RL algorithms for optimizing greenhouse control come in the form of sample efficiency and robustness, and in this section, we will review the various approaches used for handling these RL algorithm problems. Model-based approaches are commonly used to handle the sample efficiency problem, as mentioned in [28–31]. These methods have made significant improvements in dealing with sampling efficiency. However, these works are not robust enough because of their effect on the control policy while dealing with high-dimensional tasks. Due to the larger variation between the simulator and the real-world environment, performance is affected. Thus, to address the concern with control policy uncertainties, they are introduced into the transition models, which helps in figuring out the parameters that are generated using the sample inefficiency.

Another popular strategy used for solving the sample efficiency and robustness problem is the use of ensemble learning. More robustness is achieved with the joining of more than one model since, in our method, we are going to adopt the ensemble technique with additional randomization and dropout. Let us review some of the ensemble approaches used so far. Sample-efficient ensembled reinforcement learning (SEERL) [32] outputs the model framework based on the ensembled policies. SEERL aims for the selection of better policies, and that is obtained with sequential training of the policies. For a converged policy, an increment in the learning rate is done, for which the new policy is obtained, and from that, a selection of optimized policies is obtained. Since our strategy is also to perform ensembling, this is different from the SEERL because we expect the ensembling to produce a better outcome in new environments. So, the aim is to achieve diversity among the policies, and this comes through the penalty. We expect the policy to be well-adapted to the new environment. Another approach using the Q-ensemble called SUNRISE [33] can better explore the environment. Traditional RL algorithms like Soft Actor-Critic [34] and Soft Q-Learning [35] are used in the literature for the policy optimization of the autonomous greenhouse.After understanding the state of the art of reinforcement learning in an autonomous greenhouse, it is important to improve the robustness of the greenhouse. Though our primary objective is not to implement a new RL algorithm, this work does not concentrate on building the implementation; instead, the focus is on understanding the effectiveness of these algorithms in solving the problem of sample efficiency and robustness. We followed these ideas and incorporated a dropout and randomization module to achieve robustness.

3. Proposed System

This work considers the problem of autonomous greenhouse control as a mismatch Markov decision optimization problem (MMDP). The concept of model mismatch is included in the Markov Decision Optimisation problem because the agent needs to be evaluated in simulated and real environments. Using this, we can identify the gap between the training and evaluation environments. This section gives the preliminaries and notations of the considered problem and the approach to solving it.Notations and Preliminaries:

Markov Decision Process (MDP) is denoted by a tuple $MDP = (S, A, P, r, \gamma, \rho_0)$ where S denotes the state with dimension $S \in \mathbb{R}^{d_s}$ and A denotes the action with dimension $A \in \mathbb{R}^{d_a}$. The transitioning in S at state s to take action a is represented as a transition probability P (s, a) denoting the state action pair. The reward function is mapped to this state action pair with [0,1] as discount factor. Thus reward $r: (s, a) \in [0,1]$. H denotes the episode of interaction $H = \{1,2, \dots \dots \dots H\}$. Suppose if the initial state s_0 then the state $s \in s_h$ after taking the action $a \in A$. After the action is encountered the environment will change from s_h to s_{h+1} and that is denoted as state s' with the probability $P(s'|s, a)$. So, this means from the episode H it changes to H+1. Each state's policy is a distribution over action A and there exists a deterministic policy for the agent.

The reward is thus expected following the policy on the state and suppose if the state $s \in s_h$ then the reward is notated using the tern $V_M^\pi(s)$. Thus, the reward $V_M^\pi(s)$ with transition P and policy π is obtained using the equation 1 as a expectation over the probabilistic transition function.

$$V_M^\pi(s) = \text{expectation} \left[\sum_{h'=h}^{H} r(s_{h'}, \pi(s_{h'})) \right) | s_h = s \dots \dots \dots \dots \dots \dots \dots \dots (1)$$

The goal is to find a policy π' such that

$$\pi' = argmax_\pi V_M^\pi(s_0) \dots \dots \dots \dots \dots \dots \dots \dots (2)$$

Since our problem is to identify the gap between the simulation and real environment, we expect the environment to be different at training and evaluation. Thus this Markov Decision process is modelled randomly as a mismatch MDP and hence the training transition and testing transition is different denoted as$MDP^* = (S, A, P^*, r, H)$ for evaluation and $MDP^s = (S, A, P^s, r, H)$ for training. The assumption here is only the transition are different so during the training time only P^s samples are given to the agent and this state action pair generates a sample $(s', r(s, a))$. For the transfer of training to testing a policy is defined that wanted both the environment to be similar. Hence a constraint is added to definition of the perturbation set. The constraint function C aims to map P^* transition function to be located in C (P^s). Hence the perturbation is obtained by subtracting the P from C(P) where P is the transition of training and C(P) is the transition of evaluation and this is obtained using the equation (3)

$$Per(P) = (P + C(P) \dots \dots \dots \dots \dots \dots \dots . (3)$$

Where C(P)=[-1,1] so that P contains the elements 0 and P^* will be the neighbour values of P^s

Since we aim for the robustness, the policy should be considered for the worst-case environment and that is defined using the equation (4)

$$V'^\pi_M(s) = \min \left(V_M^\pi(s) \right) \dots \dots \dots \dots \dots . . (4)$$

The goal is to find the optimal policy π' such that

$$\pi' = argmax_\pi V'^\pi_M(s_0) \dots \dots \dots \dots \dots \dots \dots \dots \dots . (5)$$

And the learning is better when the policy π' satisfies the condition in equation (6)

$$Err(\pi') \leq 0 \geq 1 - (0,1) \dots \dots \dots \dots \dots . . (6)$$

The learning goal thus is the optimal policy for MDP^s performs very poor for MDP^* thus concluding a robust learning for the evaluation stage.

Formulation:

We expect that the problem framed as MMDP needs to find a methodology that provides crop yield with less cost in both real and simulated environments. Though several factors frame this optimization problem, only the observable parameters are considered. Based on these, the formulations for the state space, action space, reward function, and transition function are defined below:

3.1 State Space

The state here is a 4 variable tuple that holds different values that determine the growth of the tomato plant. The variables that are considered are tabulated in Table I.

Table I. State Space variables that makes a transition with the changes from action variables

Name	min	max
Planting days	0	365
Greenhouse temperature	-25	90
Greenhouse carbon	400	1000
Greenhouse wind	0	25

3.2 Action Space

The action space variables are the ones like temperature, light, and CO2, and any changes in the values of these variables make changes with regard to the state variables. Table II presents the action space variables.

Table II. Action Space variables whose change affects the state transition

Name	Minimum value	Maximum value
Temperature set point	10	30
Carbon-di-oxide set point	400	1000
Light on time	0	20
Light off time	0	20

3.3 Reward Function

The reward function R is the net profit that is obtained by subtracting the cost of spending in terms of the labour cost, resource consumption etc from the cost obtained with the total yield of the crop.

3.4 Transition Function

The transition value is unknown and the results are obtained based on the simulator results. This is basically a function that interchanges the state transitions to rewards.

4. Methodology

The overall architecture for solving the greenhouse MMDP problem is presented in Figure 1. As shown in Figure 1, the green house challenge dataset is used for the training and evaluation. In the simulation environment the ensembled optimization strategy is evaluated with the setting of the action variables. Thus, the optimization method is basically a predictive model learned from the greenhouse data. The robustness is measured by setting the simulation environment with unknown set of values.

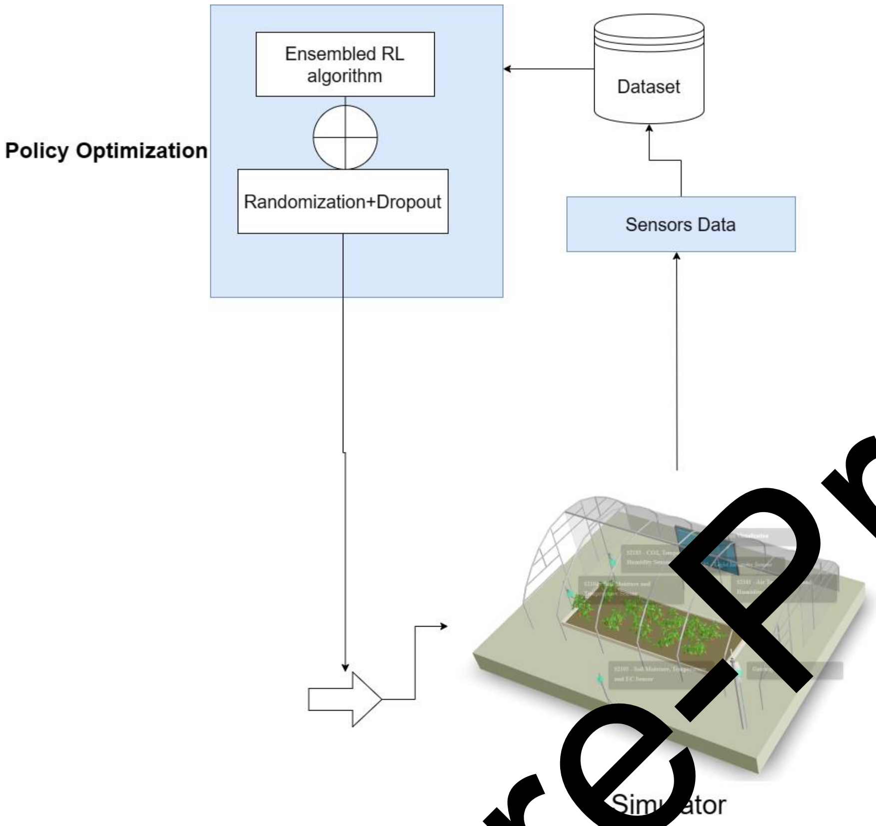


Figure 1. Overall architecture of the proposed greenhouse control system

4.1 Simulator

Simulator plays an important role in solving the MDP optimization problem. The brief summary of the simulation environment is presented in Table III. As mentioned in Table III, the simulation is supplied with details of the climate, weather, net profit, plant growth and reward function.

Table III. Overall components that are implemented as a part of the Simulation

Model Name	Components	Function
Climate	Heater Screen Light Ventilation CO2	Computing Green house State using the weather and action variable input setting
Weather	Weather Dataset	Local Real weather data is considered for the optimization of green house
Plant	Disease Extreme heat Freezing	Factors that limit the growth of the plant
Netprofit and Reward	On and Off-Peak usage price	Consumption of the input components for the calculation of profit

Under the simulation environment, the agent gets transformed to a new state based on the components of the climate model and weather data through which the growth of the plant and the consumption details of the greenhouse are computed which then collectively combined to form the new state and the new reward.

4.2 Discrete Randomization

Since the weather is updated hourly, but the climate and plant development data need to be changed every minute, discrete randomization is introduced to the simulation. Discrete randomization is used inside the simulation environment of the model to account for this, prevent differential equation mistakes, and minimize action per episode. All it takes to do this is to simulate many time scales. The climate and plant models are updated fifteen times for a single step, requiring one hour and fifteen minutes for every step.

4.3 Ensembled Policy Optimization algorithm

Though the RL algorithms are meant to learn policies well, they face a problem in terms of training. The reduction in training efficiency causes a less robust model because of the higher number of interactions with the environment. To alleviate this problem, an ensemble RL algorithm is proposed along with the dropout mechanism in algorithm 1. The simulator created resembles the actual tomato growth environment and through the greenhouse dataset, the simulation can now generate samples for the entire growth life cycle. These samples are fed as an input to the ensemble RL algorithm for optimizing the policies.

Using the greenhouse challenge data, the model is pretrained, and this dataset is referred to as $Data_{real}$ and this data includes the greenhouse and growth parameters. For the data samples to be more realistic and resemble real-world data, prior knowledge is used for adding restrictions such as the maximum level of light, temperature, etc. The reason this algorithm is proposed is to improve its robustness and address the sample inefficiency, which is its adaptability to work under varied scenarios. Utilising the idea proposed in [36], an ensemble approach is proposed using the Q-learning and actor critic methods. The reason to adopt an ensemble approach is to avoid the problems of sample efficiency and robustness so that the algorithm can adapt to the new environment with very little tuning. Thus, using the ensemble approach, the uncertainties in the greenhouse environment can be captured, thus increasing the growth of the plant as well.

The fine-tuning here happens through the different sets of samples, and these samples are generated using the Bernoulli distribution with parameters $p \in (0,1]$, then each of the sample data along with binary masking$h_{i,j}$. Thus, the subset samples are generated using the equation mentioned in (7).

$$Data_{subset} = h_{i,j} \Theta Data_j | Data_j \in Data_{real} \dots\dots\dots\dots\dots\dots..(7)$$

Now these data from the sub sets are completely new data and this is used for the fine tuning denoted as $M = \{M_{\phi_1}, M_{\phi_2}, M_{\phi_3}, \dots\dots\dots\dots\dots\dots. M_{\phi_N}\}$ and every member is a neural network that is probabilistic in nature and hence a Gaussian distribution is used to parametrize it. The output of the probabilistic neural network is $\mu_{\emptyset}, \sigma_{\emptyset}$ and the objective function of the model is given in equation (8)

Now these data from the subsets are completely new, and this is used for the fine-tuning, denoted as $M = \{M_{\phi_1}, M_{\phi_2}, M_{\phi_3}, \dots\dots\dots\dots\dots\dots. M_{\phi_N}\}$,every member is a neural network that is probabilistic in nature, and a Gaussian distribution is used to parametrize it. The output of the probabilistic neural network is $\mu_{\emptyset}, \sigma_{\emptyset}$, and the objective function of the model is given in equation (8)

$$loss_{\phi} = \sum_{t=1} [\mu_{\emptyset i}(s_t, a_t) - s_{t+1}]\sigma_{\emptyset i}^{-1}(s_t, a_t)) - s_{t+1}] + \log|\sigma_{\emptyset i}(s_t, a_t) \dots\dots\dots\dots (8)$$

In the proposed method, the simulator is trained to have an environment similar to the real environment. So the optimization samples are the ones that are pretrained using the simulator environment and the derived subset samples. Thus, at the training phase, policies aim to maximize the reward, irrespective of the environment. To achieve this, the policy divergence is achieved if the policies produce Gaussian distributions for actions, and thus the divergence policy is given using equations (9) and (10).

$$policy_{\pi i,\pi j} = concave(||expectation(\pi_j(a|0)] - epectation(\pi_i(a|0)||2] \dots\dots\dots\dots\dots\dots (9)$$

$$policy_{\pi i,\pi j}(0) = concave(||expectation(\pi_j(a|0)] - epectation(\pi_i(a|0)||2] \dots\dots\dots\dots\dots. (10)$$

The concave function used here is sigmoid, and the reason for considering this as sigmoid is because the model need not worry about how different the policies are. Moreover, the distribution of trajectories is not cumulative, and the training is carried out in parallel, hence the objective function defined in equation (8) gets modified to include the ensembled members as shown in equation (11).

$$loss_{M_{\phi_i}(div)=loss_{M_{\phi_i}}+}\ \delta policy_{\pi i,\pi j}) \dots \dots \dots \dots \dots \dots \dots (11)$$

Where δ is the hyperparameter that capture the variation between the policies.As a result of the policy divergence we have attained, policy learning proceeds smoothly in each model. Additionally, our strategy incorporates a dropout mechanism to prevent significant performance fluctuation across the models. The dropout discussed here, which draws influence from [37], seeks to exclude samples that offer an excessive reward to focus exclusively on the lowest-performing examples. As a result, the model becomes more reliable and appropriate for use in actual environments by guaranteeing plant output even under the most adverse circumstances. Algorithm 1 presents the overall optimization method used in this proposed system.

Algorithm 1: Robust Ensembled Reinforcement Learning Algorithm (Usual Reinforcement learning algorithm instructions are not included here to highlight the difference between this approach and usual algorithm.)

Main function ()

Load original environment initialize hyperparameters, $Data_{real}$ and policy $\pi \in \{\pi_{\theta 1}, \pi_{\theta 2, \dots\dots\dots\dots} \pi_{\theta n}\}$

for N_{epoch} iterations do

Take action in the greenhouse using policy$\pi_{\theta 1}, \pi_{\theta 2, \dots\dots\dots\dots} \pi_{\theta n}$ and add samples to $Data_{real}$

Mask $Data_{real}$ into

$$Data_{subset} = h_{i,j} \Theta Data_j | Data_j \in Data_{real}$$

for N_{train} iterations do

Load the pretrained model and train on

$$Data_{subset} = \{Data_1, Data_2, \dots \dots \dots \dots \dots Data_N\}$$

Get the pretrained ensemble model = $\{M_{\phi_1}, M_{\phi_2}, M_{\phi_3}, \dots \dots \dots \dots \dots \dots . M_{\phi_N}\}$

While t<T do

for every $\pi_{\theta i} \in \pi$ do

$$loss_{M_{\phi_i}} = \sum_{t=1}^{N} [\mu_{\phi i}(s_t\ , a_t) - s_{t+1}\ ]\sigma_{\phi i}^{-1}(s_t, a_t)) - s_{t+1}\ ] + \log\ |\sigma_{\phi i}(s_t, a_t)$$

end for

end while

Fine tuning

For the new environment

While t<T' do

$$policy_{\pi i,\pi j} = concave(||expectation(\pi_j(a|0)] - epectation(\pi_i(a|0)||2]$$

$$loss_{M_{\phi_i}(div)=loss_{M_{\phi_i}}+}\ \delta policy_{\pi i,\pi j})$$

end while

function dropout ()

Perform selection on model M with policy π_θ and get samples into batch $B^{\pi_\theta, M}$

Optimize policy π_θ in$Data_{real}$and $Data_{subset}$

end main

As mentioned in algorithm 1, the new environment basically gets trained with new policy obtained from the ensembled members thus improving the sample efficiency. The predictions for this ensemble model collection is obtained by choosing a model with probability mentioned in equation (12)

$$P[M = M_{\emptyset i} | i \sim P_m \, , i \ \in \{1,2 \ldots \ldots \ldots \ldots .. N\} \ldots \ldots \ldots \ldots \ldots \ldots \ldots .12$$

Then every model in M interact with varied policies and generates more growth samples on which the usual policy gradient approach is adopted for updating.

5. Results and Discussion

This section aims to give answers to the questions that were framed in Section 1.2, as below:

- Does the reinforcement learning algorithm provide better robustness than the existing methods?
- Do we have an option to make the reinforcement learning algorithm robust both in the training and evaluation conditions?
- What is the contribution of the reinforcement algorithm to getting a better yield?
- To answer these queries, the proposed system implementation details and the corresponding results are analysed.

5.1. Dataset

The greenhouse challenge dataset includes the details of six independent greenhouses, including the weather data, indoor greenhouse climate, resource consumption parameters, quality details of the tomatoes, and analysis data. Along with this, the price for the cost and net profit are also available. Based on these values, there are two datasets $Data_{real}$ and $Data_{subset}$ of tomato planting incorporated in the simulation experiment.

5.2. Performance Analysis of the Model

Analyzing the ensemble algorithm's impact in the greenhouse setting is crucial to comprehend how the reinforcement algorithm contributes to a higher yield. The prediction improves with increasing learning. The training curves and the model's efficiency with and without the drop-out mechanism are validated to comprehend the ensemble model's learning efficiency. Two versions of the training have been conducted: one with a sample dropout of 0.6 and the other without the dropout, with a value of 1. Figure 2, Figures 3, and Figures 4 demonstrate the convergence capacity of the method. As was noted, variation was reduced with the help of the drop-out mechanism. When this ensemble model is compared to the baseline SAC and PPO algorithms, it is shown that the ensemble model performs better than the baseline algorithms. The ensemble model's lower variance indicates that it places more emphasis on the worst-case scenarios. Better sample efficiency is the cause of this improved performance, and as a result, even with small sample sizes, learning is improved. Thus, this satisfies our goal of increasing sampling efficiency while using fewer training examples.

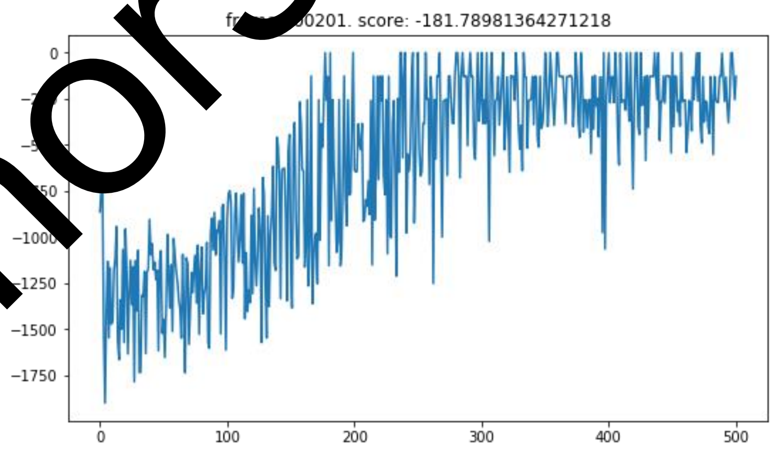


Figure 2. Learning curve showing the ensembled algorithm average score

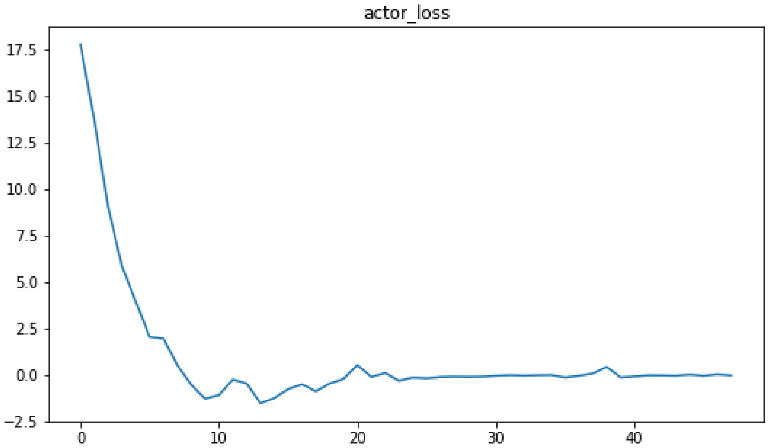


Figure 3. The actor loss value dynamics of the proposed ensembled algorithm.

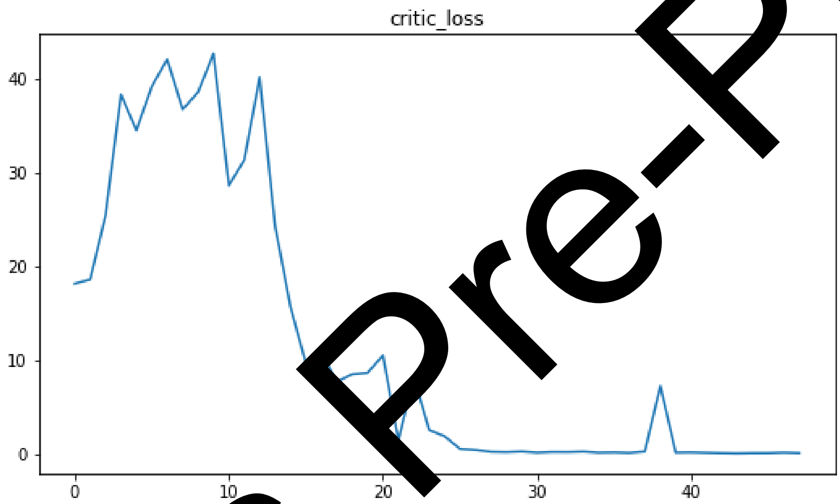


Figure 4. The critic loss value dynamics of the proposed ensembled algorithm

Figure 3 and Figure 4 demonstrate a steady reduction in the critic loss and the maximum reward, respectively, confirming high-quality Q network optimization. As Figure 2 illustrates, learning has occurred more effectively with lower values thanks to the ensemble model's assurance that the reward is maximized. Figure 5 illustrates the outcomes of a comparison study using PPO and SAC, which was done to compare the performance of the ensemble method with the baseline.

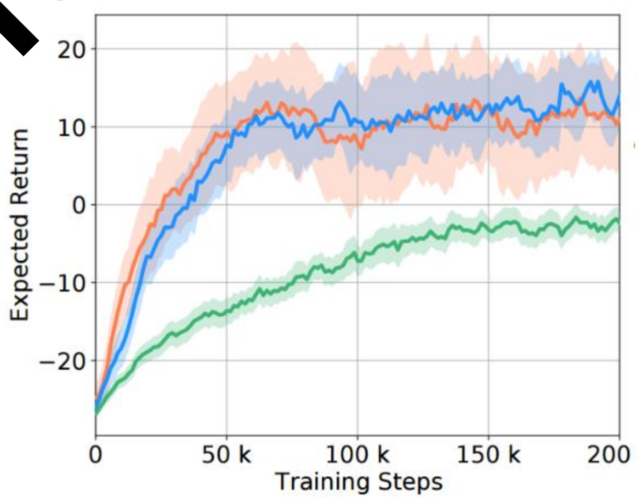


Figure 5. Comparative curve showcasing the rewards for the PPO, SAC, and ensemble learning

The solid lines show the mean of the trails with different seed values, and the shaded region denotes the standard deviation of the plot. Green indicates the performance of SAC, orange indicates the performance of PPO, and blue shows the ensemble algorithm.This performance analysis allows us to conclude that training performs better with fewer samples, and a comparison with baseline algorithms demonstrates the robustness of this approach over the state-of-the-art reinforcement algorithms, thereby providing an answer to question 1 under section 1.2.

5.3. Robustness Analysis

This study is done to understand the robustness of the proposed approach both at the training and testing phase and thus giving justification to the question 2 under section 1.2. In the proposed system, dropout mechanism is the one added for ensuring the robustness. So the benefits of dropout is analyzed with different environment. The different environment is achieved with the inclusion of outside weather conditions. This is manually setting the parameters like outside solar radiation (Iglob), greenhouse humidity and temperature. Since these parameters do have direct relationship to the growth of the plants these are set to represent the different environment and observe the model's behaviour. Since this setting up parameter takes up newer values that are not a part of the dataset, this is considered to be a new environment and the parameters here are anomalous parameters. These experimental results are tabulated in Table IV and Figure 6. The Table IV shows the crops weights and retention rate under the new environment.

Table IV. Robustness Analysis with new environment

Parameter	Crop Weight		Crop Retention Rate	
	Without Dropout	With dropout	Without dropout	With Dropout
Air temperature (35,40) for maximum and (-2,10) for minimum	32.78	46.23	80.32%	87.62%
Air Humidity (75)	39.63	42.3[illegible]	81.32%	89.43%
Solar Radiation (null)	42.17	49.26	74.23%	86.21%

Based on the findings shown in Table IV, the ensemble approach that incorporates the dropout has a higher crop weight and retention rate. As a consequence, the suggested model is sufficiently resilient to manage the updated set of anomalous parameters. Analysis was done on the net profit that resulted from setting the dropout parameter p to various values, such as 0.8, 0.5, etc. Figure 6 presents the findings as a heatmap.

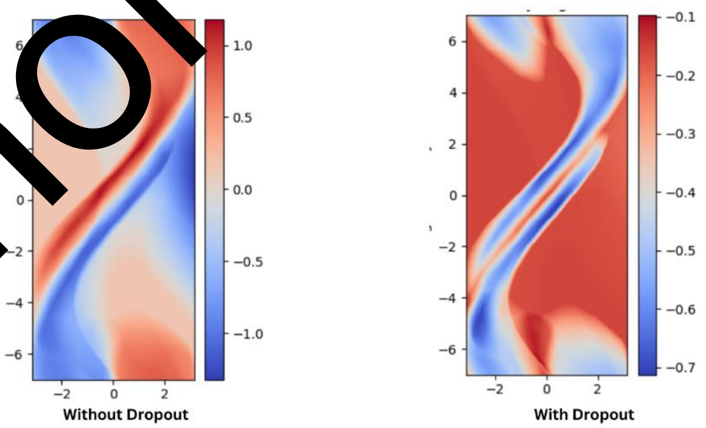


Figure 6. Heat maps showing the robustness performance under varied environments with different drop-out values

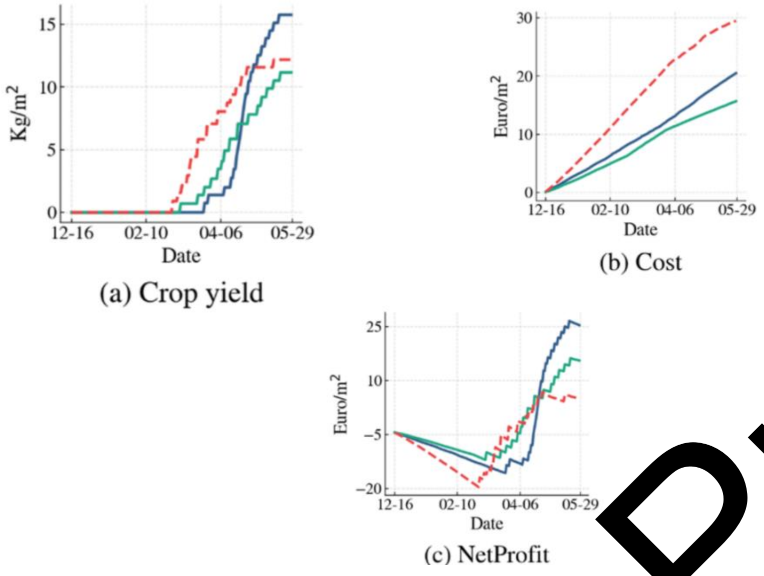


Figure 7. Performance comparative analysis of the different algorithmic approaches

As seen in Figure 7, the ensemble approach shows superior performance in terms of yield and profit; however, the cost imposed to get this yield is slightly higher in the proposed approach. The main reason for this could be because the reinforcement learning algorithms considered the finer details. There is a direct relationship between yield and profit, denoting the ensemble approach taking care of short-term and long-term optimization.

6. Conclusion

An ensembled reinforcement learning model is presented in this study to address sample efficiency and robustness issues in tomato production under greenhouse conditions. Specifically, our approach learns the various conditions using the greenhouse challenge tomato dataset, and even in brand-new situations, the policy is optimized. The evaluation of the experiment is completed and shown to address the questions posed in Section 1.2. Different experimental configurations are constructed for each topic, and the outcomes are assessed to demonstrate the robustness and efficacy of the suggested approach in addressing the sample efficiency issue.The results indicate that the yield-to-cost ratio is comparatively lower, which paves the way for further advancements in this area of study. Therefore, in our next effort, we want to enhance this algorithm's ability to generalize with a larger range of crops. To more effectively balance return and cost, a new set of algorithmic combinations must be assessed. The online RL approaches need to be evaluated in a real greenhouse, and their impact on training time and cost has to be examined.

Conflict of Interest: The authors declare no conflicts of interest(s).

Data Availability Statement: The Datasets used and /or analysed during the current study available from the corresponding author on reasonable request.

Funding: No fundings.

Consent to Publish: All authors gave permission to consent to publish.

References

[1] J. Guo et al., "Revolutionizing Agriculture: Real-Time Ripe Tomato Detection With the Enhanced Tomato-YOLOv7 System," in IEEE Access, vol. 11, pp. 133086-133098, 2023, doi: 10.1109/ACCESS.2023.3336562.

[2] S. K. Reddy et al., "Early Sensing of Tomato Brown Rugose Fruit Virus in Tomato Plants via Electrical Measurements," in IEEE Sensors Letters, vol. 6, no. 5, pp. 1-4, May 2022, Art no. 1500304, doi: 10.1109/LSENS.2022.3161595.

[3] Y. -P. Huang, T. -H. Wang and H. Basanta, "Using Fuzzy Mask R-CNN Model to Automatically Identify Tomato Ripeness," in IEEE Access, vol. 8, pp. 207672-207682, 2020, doi: 10.1109/ACCESS.2020.3038184.

[4] T. Kojić, M. Simić, M. Pojić and G. M. Stojanović, "Detecting Freshness of Fruit and Vegetable Without and With Edible Protein-Based Foil," in IEEE Sensors Journal, vol. 22, no. 16, pp. 15698-15705, 15 Aug.15,2022, doi:10.1109/JSEN.2022.3188388.

[5] https://unfccc.int/sites/default/files/NDC/202208/India%20Updated%20First%20Nationally%20Determined

%20Contrib.pdf

[6] UURL:https://treaties.un.org/pages/ViewDetails.aspx?src=TREATY&mtdsg_no=XXVII-7-d&chapter=27&clang=_en. 1

[7] Silke Hemming, Feije de Zwart, Anne Elings, Isabella Righini, and Anna Petropoulou. Remote Control of Greenhouse Vegetable Production with Artificial Intelligence-Greenhouse Climate, Irrigation, and Crop Production. Sensors, 19(8), April 2021.

[8] Iqbal Jebril, M. Premkumar, Ghaida Muttashar Abdulsahib, S. R. Ashokkumar, S. Dhanasekaran, Osamah Ibrahim Khalaf, and Sameer Algburi, "Deep Learning based DDoS Attack Detection in Internet of Things: An Optimized CNN-BiLSTM Architecture with Transfer Learning and Regularization Techniques", Infocommunications Journal, Vol. XVI, No 1, March 2024, pp. 2-11., https://doi.org/10.36244/ICJ.2024.1.1.

[9] Ramírez-Arias, A.; Rodríguez, F.; Guzmán, J.L.; Berengue, M. Multiobjective hierarchical control architecture for greenhouse crop growth. Automatica 2020, 48, 490–498.

[10] Elings, A.; Heinen, M.; Werner, B.E.; de Visser, P.; van den Boogaard, H.A.G.M.; Gieling, T.H.; Marcelis, L.F.M. Feed-forward control of water and nutrient supply in greenhouse horticulture: Development of a system. ActaHortic. 2020, 654, 195–202.

[11] karthick Perumal, V., Supriyaa, T., Santhosh, P. R., & Dhanasekaran, [illegible] (2024). CNN Based Plant Disease Identification Using Pynq Fpga. Systems and Soft Computing, 200088.

[12] Rahnemoonfar, M.; Sheppard, C. Deep count: Fruit counting based on deep simulated learning. Sensors 2019, 17, 905.

[13] Bac, C. W. (2015). Improving obstacle awareness for robotic harvesting of sweet-pepper. [internal PhD, WU, Wageningen University]. Wageningen University. https://doi.org/10.18174/327202 [13] Nieuwenhuizen, A.T.; Kool, J.; Suh, H.K.; Hemming, J. Automated spider mite damage detection on tomato leaves in greenhouses. ActaHortic. 2020, 1268, 165–172.

[14] Suh, H.K.; IJsselmuiden, J.; Hofstee, J.W.; van Henten, E.J. Transfer learning for the classification of sugar beet and volunteer potato under field conditions. Biosyst. Eng. 2018, 174, 50–65.

[15] Marcelis, L.F.M.; Elings, A.; de Visser, P.H.B.; Heuvelink, E. Simulating growth and development of tomato crop. ActaHortic. 2019, 821, 101–110.

[16] Morimoto, T.; Hashimoto, [illegible] AI approaches to identification and control of total plant production systems. Control Eng. Pract. 2019, [illegible] 555–567.

[17] Byunghyun Ban and Soobin Kim. Control of nonlinear, complex and black-boxed green house system with reinforcement learning. In 2017 International Conference on Information and Communication Technology Convergence (ICTC), pages 913–918, Jeju, October 2017. IEEE. URL: http://ieeexplore.ieee.org/document/8190813/, doi:10.1109/ICTC.2017.8190813. 2, 10.

[18] 43. [illegible]ddi, V. [illegible]., Gopal, S. K., Mohammed, A. S., Dhanasekaran, S., & Naruka, M. S. (2024, March). Examine the role of generative AI in enhancing threat intelligence and cyber security measures. In 2024 2nd International Conference on Disruptive Technologies (ICDT) (pp. 537-542). IEEE

[19] John [illegible] Preece and Paul E. Read. The Biology of Horticulture: An introductory textbook. John Wiley & Sons, 2005. [illegible], 6, 7, 27, 83

[20] Pedro Ponce, Arturo Molina, Paul Cepeda, Esther Lugo, and Brian MacCleery. Greenhouse Design and Control. Taylor & Francis Ltd, 09 2014. doi:10.1201/b17391. 3, 8, 9, 10

[21] Ming-Hui Huang and Roland T Rust. Artificial intelligence in service. Journal of Service Research, 21(2):155–172, 2018.

[22] KirtanJha, AalapDoshi, Poojan Patel, and Manan Shah. A comprehensive review on automation in agriculture using artificial intelligence. Artificial Intelligence in Agriculture, 2:1–12, 2019.

[23] Priyan, S. V., Dhanasekaran, S., Karthick, P. V., & Silambarasan, D. (2024). A new deep neuro-fuzzy system for Lyme disease detection and classification using UNet, Inception, and XGBoost model from medical images. Neural Computing and Applications, 36(16), 9361-9374

[24] WouterJacobus Peter Kuijpers. Model Selection and Optimal Control Design for Automatic Greenhouse Climate Control. PhD thesis, Mechanical Engineering, March 2021. Proefschrift. 2, 10

[25] Zhicheng An, Xiaoyan Cao, Yao Yao, Wanpeng Zhang, Lanqing Li, Yue Wang, ShihuiGuo, and DijunLuo. A simulator-based planning framework for optimizing autonomous greenhouse control strategy. Proceedings of the International Conference on Automated Planning and Scheduling, 31(1):436–444, May 2021. URL: https://ojs.aaai.org/index.php/ICAPS/article/view/15989. 10

[26] FH de Zwart. Simulatiemodelkaspro, 2022. URL: https://www.wur.nl/nl/show/Simulatiemodel-KASPRO-1.htm. 10

[27] Lu Wang, Xiaofeng He, and DijunLuo. Deep reinforcement learning for greenhouse climate control. In 2020 IEEE International Conference on Knowledge Graph (ICKG), pages 474–480. IEEE, 2020

[28] Ganesh Chandrasekaran, S. Dhanasekaran, C. Moorthy & A. Arul Oli (2024) Multimodal sentiment analysis leveraging the strength of deep neural networks enhanced by the XGBoost classifier, Computer Methods in Biomechanics and Biomedical Engineering, DOI: 10.1080/10255842.2024.[illegible]13066

[29] Kurtland Chua, Roberto Calandra, Rowan McAllister, and Sergey Levine. Deep Reinforcement Learning in a Handful of Trials using Probabilistic Dynamics Models. Advances in Neural Information Processing Systems, 2018-Decem(NeurIPS):4754–4765, 2018. ISSN10495258.

[30] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. When to trust your model:Model-based policy optimization. In Advances in Neural Information Processing Systems,pages 12519–12530, 2019.

[31] RohanSaphal, BalaramanRavindran, DheevatsaMudigere, [illegible]Avancha, and Bharat Kaul. SEERL: sample efficient ensemble reinforcement learning. CoRR, abs/2001.0[illegible], 2020

[32] Kimin Lee, Michael Laskin, AravindSrinivas, and Pieter Abbeel. SUNRISE: A simple unified framework for ensemble learning in deep reinforcement learning. CoRR, abs/2007.04938, 2020.

[33] TuomasHaarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. CoRR, abs/1702.08165, 201[illegible]

[34] TuomasHaarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. CoRR, abs/1801.01290, 2018

[35] Richard S. Sutton. Dyna, an integrated architecture for learning, planning, and reacting. ACM SIGART Bulletin, 2(4):160–163, 1991. ISSN 0163-5719. doi: 10.1145/122344.122377.

[36] Aviv Tamar, Yonatan Glassner, and ShieMannor. Optimizing the CVaR via sampling.Proceedings of the National Conference on Artificial Intelligence, 4:2993–2999, 2015.

[37] G Parameswaran and K Siva[illegible]rasath. Arduino based smart drip irrigation system using internet of things. International Journal of Engineering Science and Computing, 6(5):5518–5521, 2016.