Jurnal Ilmiah Teknik Elektro Komputer dan Informatika (JITEKI)
Vol. 11, No. 2, June 2025, pp. 154-168
ISSN: 2338-3070, DOI: 10.26555/jiteki.v11i2.30690

Enhancing Soybean Fertilization Optimization with Prioritized Experience Replay and Noisy Networks in Deep Q-Networks

Alfian Fakhrezi[1], Gelar Budiman[1], Doan Perdana[2]
[1]Department of Electrical Engineering, School of Electrical Engineering, Telkom University, Bandung 40257, Indonesia
[2]Department of Electrical Engineering, School of Electrical Engineering, Muhammadiyah University of Surakarta, Surakarta 57169, Indonesia

ARTICLE INFO

Article history:
Received February 10, 2025
Revised March 21, 2025
Accepted April 18, 2025

Keywords:
Deep Q-Network;
Fertilizatio;
Noisy Network;
Prioritized Experience Replay;
Precision farming;
Soil nutrient management

ABSTRACT

This study focuses on the optimization of reinforcement learning in the Deep Q Network algorithm. This is achieved using the prioritized experience replay algorithm and Noisy Network optimization. The main goal is to optimize fertilization so that it can adapt to its environment and avoid over-fertilization. This study uses the prioritized experience replay algorithm and Noisy Network optimization to create an agent in RL that is able to explore and exploit optimally so that it can improve the precision of fertilization in soybeans. This methodology includes several steps, including data preparation, creating an environment that matches real-world conditions, and validating changes in soil nutrient conditions. The RL model was trained with PER and NN, with performance evaluated using cumulative reward, convergence speed, action distribution, and Mean Squared Error (MSE). The main results of the study show that DQN-PER NN achieves the highest cumulative reward, approaching 600,000 in 1000 episodes, outperforming standard DQN, A2C, and PPO. It also converges faster at episode 230, indicating superior adaptability. In addition, the results of this study indicate that the model that has been created is able to recommend a dose of SP36 fertilizer of 150 kg/ha, urea fertilizer of 100 kg/ha, and KCL fertilizer of 125 kg/ha. Compared with the A2C and PPO methods, the dose of urea fertilizer is reduced by 14%, KCL fertilizer is reduced by 33%, while for SP36 the difference is 23%. In Conclusion this model effectively distributes actions based on environmental conditions, which supports sustainable agriculture. In conclusion, the integration of PER and NN into DQN significantly improves exploration and decision making, and optimizes soybean fertilization. This model not only improves harvest efficiency but also encourages sustainable agricultural practices.

Corresponding Author:
Alfian Fakhrezi, Department of Electrical Engineering, School of Electrical Engineering, Telkom University, Bandung 40257, Indonesia
Email: alfian.fakhrezi@gmail.com

1. INTRODUCTION

Agriculture has always been essential for humanity and remains the foundation of both national and global economies [1], [2]. Food security is a major concern, with a potential crisis predicted in the next 50 years due to factors like climate change and soil nutrient depletion [3]. Recent reports highlight an alarming rise in food crises and malnutrition since 2024, emphasizing the urgent need for sustainable agricultural solutions[3]. Indonesia, as an agricultural country, plays a crucial role in global food security, particularly in soybean production. Despite its potential, soybean farming in Indonesia struggles to compete with major producers like Brazil and the U.S. due to lower yields and suboptimal farming techniques [4]. Then one of the important factors in agriculture is fertilization fertilization is crucial for soybean growth, but improper application can harm the environment and plants. Soybeans require essential nutrients like nitrogen, phosphorus, and potassium, which support chlorophyll synthesis, root growth, and nutrient uptake for higher yields and stress toleranc [5]-[8]. Proper fertilization improves bean size, quality, and resistance to pests while promoting sustainable agriculture and increasing farmers' incomes [9], [10]. High temperatures during flowering and filling can reduce seed weight, alter chemical composition, and affect seedling emergence and vigor [11]. However, environmental factors such as temperature, rainfall, and solar radiation also influence soybean growth. High temperatures can reduce seed weight and affect seed quality, while rainfall and solar radiation impact productivity in various ways [10]-[12]. To enhance productivity, technological advancements such as artificial intelligence can be integrated into agricultural practices [13]-[15].

Given these complexities, advanced technologies such as artificial intelligence (AI) offer promising solutions to enhance precision in fertilization management. In [14] discusses reinforcement learning for irrigation management systems in rice and focuses on efficient water use through sensor nodes, but does not specifically discuss rice irrigation or weather forecasting. In addition, there is research related to fertilizer management by introducing the RL framework to optimize crop yields based on fertilizer management [15], [16], [17]. Several studies have used RL for fertilization, such as Dueling DDQN and PPO, which showed increased nitrogen efficiency. However, this method still has weaknesses in exploring actions and does not consider the needs of phosphorus and potassium in tropical agricultural systems such as in Indonesia [18]. Standard practice agents performed well but were less adaptive. Evaluations were conducted over several /years of trials, showing a strong correlation between fertilizer application and nitrogen uptake. Dueling DDQNs showed the best strategy in the long run, although results varied depending on the number of applications [19].

There are other studies related to fertilization using Gym-DSSAT [20], and CropGym [15] both studies discuss nitrogen fertilization management in corn and wheat crops in Europe [21].

Several previous studies have discussed Reinforcement Learning for agriculture that discusses fertilization and irrigation management using Proximal policy optimization (PPO) and Deep Q-Network (DQN) [14], [16] the results obtained outperform the policies recommended by experts, in addition, there is also nitrogen management in corn and wheat [20], but in soybean planting, farmers need three main fertilizers, namely Urea, KCL and Potassium for optimal growth. In addition, the study [22] uses Deep Q-Network as an algorithm for the system. The results obtained by the algorithm have shortcomings in terms of action distribution and optimal fertilizer amounts. In addition, these studies use subtropical environmental data such as Europe [16]. Although various studies have applied RL to fertilizer management, most of them still focus on nitrogen only and use subtropical plant-based environments. In addition, standard RL methods often face challenges in action exploration and learning stability, which can lead to suboptimal fertilizer recommendations.

To overcome the limitations of exploration and learning stability, this study proposes the use of PER and NN in Deep Q-Network (DQN). PER allows the agent to focus more on experiences that are more influential on learning, while NN enhances action exploration by adapting to environmental changes. The PER optimization algorithm is a technique in RL to replay important experiences more often, This technique aims to improve learning efficiency by focusing RL on experiences that are most likely to improve decision making [23], [24]. PER uses mathematical reasoning to assign priority scores based on temporal difference errors (TD errors) which measure how surprising or unexpected a transition is [25], [26]. In Reinforcement Learning, especially Deep Q-Network, the $\epsilon -$ greedy policy is used, which often leads to inefficient exploration because it produces random behavior without considering the conditions and situations of the environment and agents and it is also often difficult to determine the right $\epsilon -$ greedy [27], [28]. In this case, Noisy Networks (NN) is a method of providing noise to the weights and biases of a neural network to train the neural network so that the agent can automatically adapt to environmental conditions over time, in short, NN reduces randomness and encourages choices during training [29].

The purpose of this study is to develop a DQN model using PER and NN methods for accurate and efficient soybean fertilization. By achieving this, we aim to improve the efficiency of soybean multifertilization based on Reinforcement Learning and to analyze the actions taken by the RL agent. It can be a recommendation for farmers to achieve better yields. this study is expected to improve the efficiency of soybean fertilization and provide more accurate data-based recommendations for farmers, thereby increasing crop yields and agricultural sustainability.

2. MATERIAL AND METHODS

2.1. Dataset

This study uses data from two main sources: an NPK, pH, and moisture sensor providing time-series data from tests in Colomadu and Karang Anyar, and meteorological data from the Karang Anyar Meteorology and

Geophysics Agency, including rainfall and temperature [30], [31]. Additional information from journals helps define soil mass, fertilizer content, solar radiation, and planting time. After processing, this data is used to build the RL environment, including state, reward calculation, and actions. The steps taken in the preprocessing process are as follows:

1. Data Collection: The data includes daily time series records from sensors with data collection on 1 ha of land. (N, P, K in ppm, pH, and Moisture) and meteorological data (rainfall, max/min/average temperature). Additional variables such as solar radiation, fertilizer content, soil mass, and planting period were gathered from literature.
2. Data Cleaning: Missing values were handled using **interpolation or imputation**, **outliers** were identified using **statistical methods (e.g., Z-score, IQR)** and either removed or adjusted, and **irrelevant data** was eliminated to maintain dataset integrity.
3. Normalization and Standardization: Features were **normalized (0–1 range)** to prevent magnitude dominance, **standardized (mean = 0, std = 1)** for models sensitive to normal distributions, and the appropriate technique was chosen based on variable characteristics.
4. Data Merging: Sensor and meteorological data were **aligned based on timestamps**, **granularity mismatches** were resolved using **resampling techniques**, and inconsistencies in recording frequency were adjusted to maintain synchronization across variables.

Merging sensor data with BMKG data based on the appropriate time series per day. Processed data would define the state in this RL environment, including variables like N, P, K, pH, humidity, temperature, and rainfall. It is combined with a reward function, probably aimed at improving crop yield or fertilizer efficiency. This will help in building the RL environment by defining state space, action space, and reward. The environment so defined will enable the agent to learn from its experiences and arrive at better decisions over some time.

2.2. Method

2.2.1. Deep Q-Network

Q-learning is a popular reinforcement learning algorithm that operates independently of the environment and is classified as an off-policy method, as it updates values based on actions from a different policy [15], [16], [20]. In Deep Q-Networks (DQN), the action value depends on the maximum Q-value and its corresponding reward [18]. This approach can be combined with Q-learning to solve problems effectively, and since artificial neural networks (ANN) can directly process input values, state discretization is not required [32], [33]. Deep learning (DL) is commonly used in the agent block to enhance state stability and provide reliable transition functions [34], [35]. To address data correlation issues, experience replay captures and reuses past experiences from memory when training the neural network [36]. The optimal action value is determined by the target Q and the prediction Q, which are defined as the loss function in the relationship (1).

$$L(\theta) = E\left[\left(r + \gamma . \max Q\left(s_{t+1}, a_{t+1} | \theta'\right) - \left(s_t, a_t | \theta\right)\right)^2\right] \tag{1}$$

After the sampling process to minimize the loss function, the next step is to update the state function and subsequent actions based on the previously defined function. The formulation of this next function is described by the Bellman equation in equation (2).

$$Q(s_t, a_t) = Q(s_t, a_t) + [r + \max Q\left(s_{t+1}, a_{t+1}\right) - Q(s_t, a_t)] \tag{2}$$

The parameter θ is used for weighting the target value and the predicted output value of the Q-network. Alpha (α) is the learning rate that determines how much new information replaces old information. Reward (r) is obtained after choosing an action, while the discount factor (γ) has a value between 0 and 1, which determines the current value of the future reward. The value $maxQ(s_{t+1}, a_{t+1})$ represents the Q value of the next state-action pair chosen to obtain optimal future benefits [37], [38]. In the initialization of DQN parameters, some important settings include the maximum number of steps per episode (C), the maximum number of episodes, the size of the replay experience memory, the learning rate (α), the discount factor (γ), and a small value for random memory transitions (P).

2.2.2. Prioritized Experience Replay (PER)

Prioritized Experience Replay (PER) enhances Reinforcement Learning by prioritizing experiences in the replay buffer based on Temporal Difference (TD) error, which measures the discrepancy between predicted and actual rewards. Experiences with larger TD errors are given higher priority, as they provide more informative learning signals.: TD error is usually denoted by δ and is calculated by the formula [39]-[42]:

$$\delta = r + \gamma \cdot \max_{a'} Q\,(s', a') - Q(s, a) \tag{3}$$

where r is the reward, γ\gammaγ is the discount factor, $(Q(s, a))$ represents the current Q-value, and $(\max_{a'} Q\,(s', a'))$ denotes the highest expected Q-value in the next state. PER assigns sampling priorities using [43]:

$$\text{priority} = (|\delta| + \epsilon)^{\alpha} \tag{4}$$

δ calculation based on (3) [37]. Once the priorities are calculated, the sampling probability P(i) for experience i in the buffer is determined by:

$$P(i) = \frac{\text{priority}_i}{\sum_j \text{priority}_j} \tag{5}$$

This approach improves learning efficiency by focusing on high-value experiences, accelerating convergence, and enhancing decision-making in complex environments [44].

2.2.3. Noisy Network (NN)

Noisy Networks is a technique in Reinforcement Learning (RL) that enhances an agent's exploration by adding noise to the neural network [27]. This method replaces traditional approaches like epsilon-greedy, which require manual tuning of exploration parameters such as epsilon. With Noisy Networks, exploration happens automatically as noise is directly added to the weights and biases in specific layers of the neural network [45]. The main goal of Noisy Networks is to help agents explore more adaptively and efficiently. This technique allows agents to take different actions at each step without explicitly using a random exploration mechanism. The added noise influences the network's output, enabling the agent to adjust its decisions based on the environment's context [27]. In Noisy Networks, the added noise consists of learnable components. For example, the weights and biases in the network are expressed as a combination of the learned mean ϕ and standard deviation σ.The equations for weight (w) and bias (b) in Noisy Networks can be written as follows [27], [45]:

$$w = \mu_w + \sigma_w \cdot \epsilon_w \tag{6}$$

$$b = \mu_b + \sigma_b \cdot \epsilon_b \tag{7}$$

ε is random noise drawn from a standard distribution (usually a Gaussian distribution). The values of μ and σ are learned during training, allowing the agent to adjust the noise level accordingly for the environment.

2.2.4. Reinforcement Learning

This research applies Reinforcement Learning to optimize soybean fertilization by designing an environment that closely resembles real conditions. The agent selects fertilization actions for urea, SP36, and KCL based on a policy and target Q-network, aiming to maximize rewards. The state space represents environmental conditions, while rewards are determined by factors such as soil moisture, pH, nutrient deficiencies (N, P, K), and evapotranspiration influenced by temperature, soil moisture, and solar radiation. The overall framework is illustrated in Fig. 1 and will be explained in detail in Fig. 1.

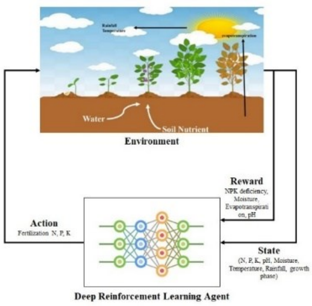


Fig. 1. Reinforcement Learning Framework

Components in reinforcement learning are agents and environments; Environment is a system where agents interact to learn to make optimal decisions**.** In the environment consists of state space, then action space, termination condition, and rewards. all components are explained below:

1. State Space

State is a description of the condition of the environment that helps the agent in understanding the current situation so that it can choose the right action. While the state space is in the form of variables related to the growth system of soybean plants. in general, the state space equation is:

$$s = \{s \in R^n \mid s = [s_1, s_2, \dots, s_n]\} \tag{8}$$

In this Reinforcement Learning, the state space used consists of 10 variables that represent the environmental conditions of soybean growth. These variables shown in Table 1.

Table. 1. State Space Variable

State variable	Description	Unit
N (Nitrogen)	Nitrogen content in the soil	Part per million
P (Phosphorus)	Phosphorus content in the soil	Part per million
K (Potassium)	Potassium content of the soil	Part per million
pH	The acidity of the soil	-
Moisture	Soil moisture	%
temperature max	Daily maximum temperature	°C
temperature min	Daily minimum temperature	°C
temperature avg	Daily average temperature	°C
rainfall	Daily rainfall	Mm
phase plant	Growth phases of soybean plants	0 to phase 5

Based on the states above, the equation used for this state space is as follows:

$$s = \{N, P, K, pH, Moisture, Suhu\_max, Suhu\_min, Suhu\_avg, fase\} \tag{9}$$

2. Action Space

Action space is the set of actions that can be taken by the agent at each step in the environment. There are 3 action spaces in this research because it is in accordance with real conditions in the field, namely 0 for no fertilization, action 1 for urea or nitrogen fertilization, action 2 for fertilization for phosphorus or SP36, action 3 for potassium or KCL fertilization, and action 4 for combination of urea Sp36 and KCL, the equation is as follows:

$$A = \{0, 1, 2, \dots, 4\} \tag{10}$$

Calculation of actions 1, 2, and 3 based on shortages [46] :

$$\mathrm{fn} \times \mathrm{pc} = \mathrm{nud} \times \frac{\mathrm{sm}}{10^6} \tag{11}$$

fn is the fertilizer amount in kg per hectare, pc is the percentage of content this is the content used in the fertilizer, nud is the nutrient needed, sm is the soil mass and 106 is the multiplier vactor for m2 to hectare.

3. Reward

Rewards are given immediately after the agent takes an action, transitioning from state **s** to state **s'**. The total rewards received after each action are multiplied by a discount factor to prevent infinite reward accumulation. In a Reinforcement Learning (RL)-based fertilization system, the reward function plays a crucial role in guiding the agent toward optimal decisions. It provides feedback based on the actions taken and the environmental conditions observed. By designing an effective reward function, the agent can learn to maximize fertilization efficiency and improve crop yields. In this study, rewards are given not only for achieving the target levels of nitrogen (N), phosphorus (P), and potassium (K) but also for maintaining soil and plant health.. Rewards for soil nutrients are as follows:

$$reward.apk = \begin{cases} 0, & if action = 1\ condition (N < target_N \land P < target_P \land K < target_K) \\ 10, & if action = 1\ condition (N > target_N \land P > target_P \land K > target_K) \\ 10, & if\ action = 1\ condition\ (N < target_N \\ 1, & if\ action = 1\ condition\ (N > target_N \\ 10, & if\ action = 2\ condition\ (N < target_N \\ 1, & if\ action = 2\ condition\ (N > target_N \\ 10, & if\ action = 3\ condition\ (N < target_N, \\ 1, & if\ action = 3\ condition\ (N > target_N \\ 10, & if\ action = 4\ condition\ (N < target_N \\ 1, & if\ action = 4\ condition\ (N > target_N \\ 8, & if\ action = 4\ condition\ (N = 0 \land P = 0 \land K = 0) \end{cases} \tag{12}$$

The next reward is based on the journal [40] that evapotranspiration (ET) has an effect on the evaporation of soil nutrients and soil moisture, therefore a reward is made based on ET using the Hargreaves Method, namely:

$$ET_0 = 0.0023 \cdot \left(T_{\text{avg}} + 17.8\right) \cdot (T_{\max} - T_{\min})^{0.5} \cdot R_a \tag{13}$$

The Reward function is:

$$reward\ ET\ = \begin{cases} 2, & if\ 3 < ET < 5 \\ -2, & if\ ET < 3\ or\ ET > 5 \end{cases} \tag{14}$$

Solar radiation (**Ra**) is set to a default value of **0.0820 MJ/m²/day**, based on [47], which represents typical solar radiation in Indonesia. Meanwhile, the **maximum, minimum, and average daily temperatures** are taken from the state after the agent takes an action.

The reward function in this study focuses on **two key aspects: Action and Evaporation**. These aspects are designed to encourage optimal fertilization strategies that ensure balanced nutrient supply while considering environmental factors. The total reward is calculated as follows:

$$\text{total reward} = \text{clip}(\text{reward\_npk} + \text{et\_reward}, -10{,}10) \tag{15}$$

With this reward system, the model is expected to be able to learn fertilization decisions that are responsive to plant conditions, thereby supporting efficient crop yield improvements.

4. Termination Condition

   In this study the termination conditions will be to achieve the desired nutrient or environmental targets where plants can grow optimally with sufficient nutrient intake. There are Plant Growth Reaches Optimal Phase [48], Optimal Nutrient Conditions [49] and Maximum Day Limit in One Fertilization Cycle [50].

2.2.5. Proposed Method

A reinforcement learning framework is used by the Deep Q-Network (DQN) machine learning model uses a reinforcement learning framework to optimize fertilization practices in soybean crop systems . DQN maps states and actions into Q function values which quantify the relative importance of an action in a state using artificial neural networks. The intricacy of agricultural dynamics can be modeled using the neural network design. To improve learning stability, the experience replay approach saves and retrieves prior encounters. By carefully combining exploration and exploitation, DQN allows the model to update fertilization guidelines in response to continuous environmental interaction. By iteratively creating the best fertilization strategies, DQN seeks to increase soybean yields.

In Fig. 2, this mechanism is shown. The agent use historical data samples and break the association between data in order to support convergence and stability in neural network training. This is accomplished by using a uniform random sampling technique to retrieve data for the minibatch and by saving data in the database throughout the reinforcement learning process. Additionally, to avoid model uncertainty resulting from initial conditions, the state of each episode is randomly initialized during training. then **Fig. 3** above shows that t is the time the agent executes a process or action, s is the state, or the state of the environment, St is the condition at a specific moment, r is the reward, and πis the action policy that the agent executes by Q-Value. Play again the minibatch is a storage location based on Priority experience playback using Temporal Difference Error for training noisy neural networks, whereas the buffer is a storage location for action, reward, state, and future state.

In Fig. 2 and Fig. 3, t is the time the agent performs an action or process, s is the state, namely the condition in the environment, St is the condition at a certain time, r is the reward, π is the action policy that the agent performs based on Q-Value, Replay Buffer is a storage place for action, reward, state and next state, minibatch is a storage place based on Priority experience replay using Temporal difference Error for training noisy Neural networks.

With an agent, the entire working environment procedure is shown in Fig. 3. the workflow of the agent in this system. The state is first initialized and stored in the experience replay buffer, which holds up to 64 samples. Once the buffer exceeds this size, a mini-batch is randomly selected for training. The neural network, consisting of an input layer (S, a, R, S*), two hidden layers with 64 neurons each, and an output layer generating Q-values, updates the agent's policy. The agent selects an action, which is then processed by the environment to update the state and reward. This cycle continues until the termination condition is met. Table. ***2*** presents the DQN algorithm's pseudocode with PER and Noisy Layers.

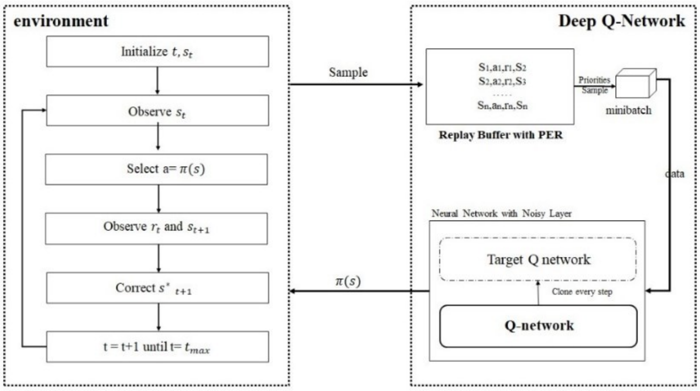


Fig. 2. Proposed Method

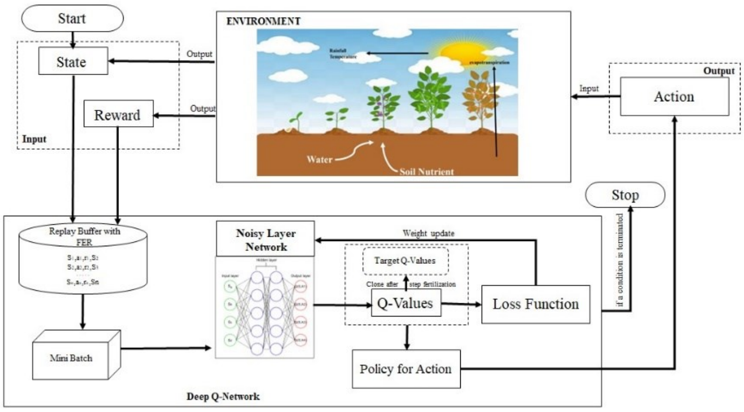


Fig. 3. System process flow

Table. 2. Pseudocode DQN-PER NN

Algorithm 1: DQN – PER NN

Initialize replay memory D with prioritized replay buffer to capacity N
Initialize action-value function Q with Noisy Layers in network and random weights θ
Initialize target action-value function Q with weights θ − = θ
Set initial exploration rate

For episode = 1, ..., M do
    Collect the environmental condition and initialize state s
    For t = 1, ..., T do
        Select action at = argmax(Q(st ,at ;θ)) using the noisy Q-network for automatic exploration
        Execute action at in the environment
        Observe reward rt and next state st+1
        Calculate TD error $\delta = \left|rt + \gamma \cdot max\left(Q(st+1, a; \theta -)\right) - Q(st, at; \theta)\right|$
        Store $(st, at, rt, st+1)$ in prioritized replay buffer D with priority δ α
        If size of D > batch size:
            Sample minibatch of experiences (sj ,aj ,rj ,sj+1) from D with probability proportional to priority
            Set target y j : $y\,j\ = rj\ if\ sj+1$ is terminal, otherwise y j = rj + γ · $max\left(Q(sj+1, a; \theta -)\right)$
            Compute importance-sampling weights for each sample
            Calculate loss as weighted MSE: loss = mean(weights·(y j −Q(sj ,aj ;θ))2)
            Perform a gradient descent step on loss with respect to θ
            Update priorities in D based on new TD errors for sampled transitions
        Every C steps:
            Clone weights of Q-network to target Q-network, θ − = θ
            Decay exploration rate (optional if epsilon-greedy is combined):
                ε = max(εmin, ε · εdecay)
            Update state st to st+1
    End for
End for

3. RESULTS AND DISCUSSION

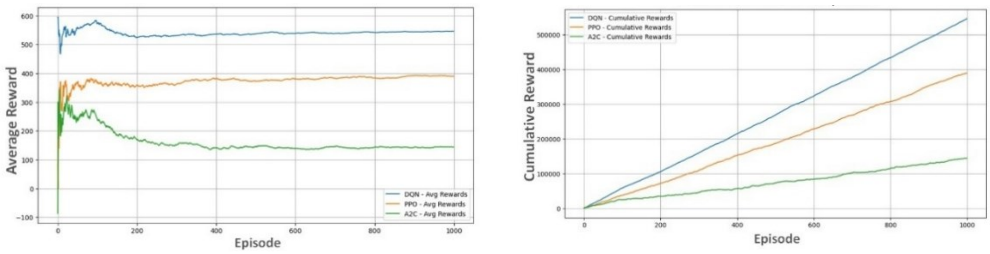
The study presents results through metric evaluation, curve analysis, and yield prediction based on fertilization. To assess DQN's performance, we compare it with A2C and PPO. DQN uses Q-values for discrete actions, while A2C and PPO rely on policy-based methods for handling environmental variations [13]. This comparison helps identify each algorithm's strengths in optimizing cumulative rewards, yield outcomes, convergence speed, and computational efficiency.

3.1. Evaluation Metrics

In this journal, the evaluation each model is assessed based on the accumulated rewards obtained over 1000 episodes with hyperparameters buffer size 15000, Batch size 32, Gamma 0.9, Learning rate, 0.0001, TAU 0.01, epsilon end 0.02 and epsilon decay 0.995, which indicates how well each model optimizes the fertilization strategy in this scenario.

3.1.1. Cumulative Reward

In Fig. 4 (b), The results show that DQN achieves the highest cumulative rewards compared to PPO and A2C, reaching around 590,000 in the last episode. This indicates that DQN effectively learns and applies fertilization strategies, making it a stable and suitable choice. PPO also performs well, achieving around 390,190 cumulative rewards, though still lower than DQN. Meanwhile, A2C shows the lowest performance, with a flatter reward curve, suggesting that it struggles to optimize fertilization effectively. This may be due to the actor-critic approach, which requires more training or tuning to perform optimally.



(a) Average Reward (b) Cumulative Reward

Fig. 4. Reward

3.1.2. Convergence Time

The average rewards of three reinforcement learning algorithms DQN, PPO, and A2C are compared in the graph below, in Fig. 4 (a). Each algorithm has a unique convergence pattern that indicates how quickly and well it adapts to a particular learning environment. The most promising results are shown by DQN or Deep Q-Network. DQN reaches convergence and stabilizes at a reward value of around 600 in the first 100 to 150 episodes. After this initial stage, its rewards are consistent and barely change. DQN's excellent performance further indicates that this algorithm can effectively adapt to its environment and create a successful policy. Proximal Policy Optimization, or PPO, performs very well, although slightly worse than DQN. PPO stabilizes at an average reward of around 550 after taking 150–200 events to reach convergence. The benefits offered by PPO increase gradually before plateauing. Although the reward of PPO is slightly lower than that of DQN, its stability suggests that PPO also learns an efficient policy, although perhaps not as well as DQN. Fig. 4 shows the best results of hyperparameter tuning, while the results of testing other hyperparameters are shown in the Table 3.

A2C is stable at episode 800 with its small reward only in the range of 150–200. This implies that A2C cannot adapt effectively to this environmental incentive system. Given its inability to maximize rewards efficiently in the same setting, A2C may not be the best choice, as indicated by its low reward stability. All things considered, DQN appears to be the best algorithm for this setting due to its fast convergence and higher reward rate.

3.2. DQN PER NN Result

From the evaluation in Fig. 4 (a) and Fig. 4 (b), DQN outperforms A2C and PPO in terms of reward acquisition and convergence speed. However, DQN has certain limitations such as falling into suboptimal actions. In this work, the enhancement in DQN is introduced by incorporating PER and Noisy Layers. The results of the combined methods are explained.

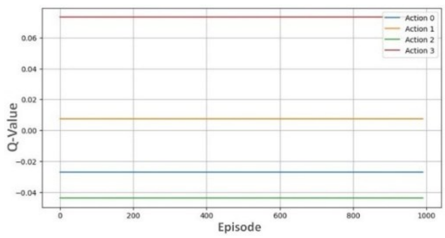
Table 3. Hyperparameter Result

Buffer Size	Batch size	Gamma	Learning Rate	TAU	Epsilon end	Epsilon decay	Average Reward	Convergence time
5000	32	0.9	0.001	0.01	0.01	0.995	500	850
10000	32	0.95	0.0005	0.005	0.01	0.995	460	600
20000	32	0.99	0.0001	0.01	0.01	0.995	390	400
5000	64	0.9	0.001	0.01	0.05	0.99	530	529
10000	64	0.95	0.0005	0.005	0.05	0.99	551	903
20000	64	0.99	0.0001	0.01	0.05	0.99	521	870
5000	128	0.9	0.001	0.01	0.01	0.995	498	763
10000	128	0.95	0.0005	0.005	0.01	0.995	581	900
15000	32	0.9	0.0001	0.01	0.02	0.995	600	125
15000	64	0.95	0.0005	0.005	0.02	0.995	421	894
5000	32	0.98	0.001	0.01	0.01	0.995	489	650

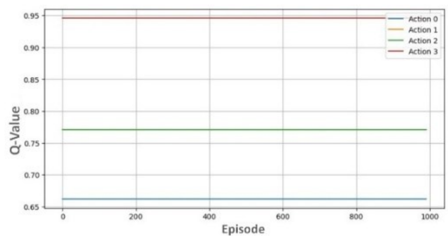
3.2.1. Q – Value

The distribution of Q-values in the DQN algorithm across 1000 training episodes without Prioritized Experience Replay (PER) and Noisy Layer is displayed in Fig. 5. It is evident that each action's Q-value is comparatively constant from the start and barely varies over the course of the episodes. This overly quick stability suggests that the agent might not have done enough exploring or gained knowledge from a wide range of experiences, which could impede the process of determining the best course of action. The agent is unable to concentrate on more instructive encounters and has less exploratory freedom when PER and Noisy Layer are absent. Without the Noisy Layer, exploration just uses the epsilon greedy setting without any noise variation in the network, and without PER, the agent learns from encounters at random without giving priority to the most important ones. In the meantime, the DQN algorithm's Q value variations with PER and Noisy Layer are displayed in Fig. 6**.**

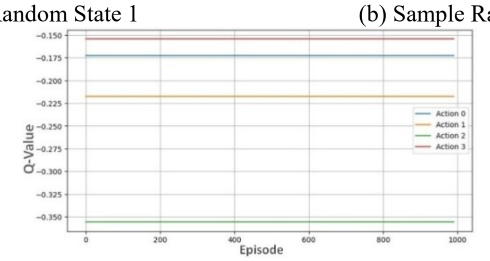
Fig. 6 shows the distribution of Q-values for the DQN algorithm using Prioritized Experience Replay (PER) and Noisy Layer for three different states (Tracked State 1, 2, and 3) over 1000 training episodes. It can be seen from each graph that the Q-values for each action (Action 0, Action 1, Action 2, and Action 3) have significant fluctuations, especially at the beginning of training, which gradually become more stable over episodes. These fluctuations highlight the active exploration early in the learning process, which is caused by the Noisy Layer.



(a) Sample Random State 1



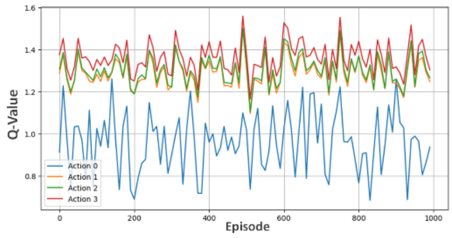
(b) Sample Random State 2



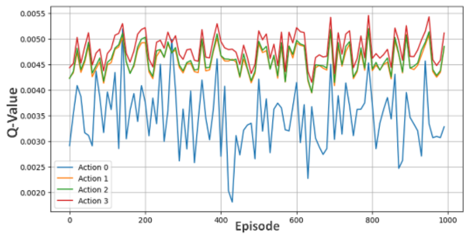
(c) Sample Random State 3

Fig. 5. Change of Q in DQN algorithm

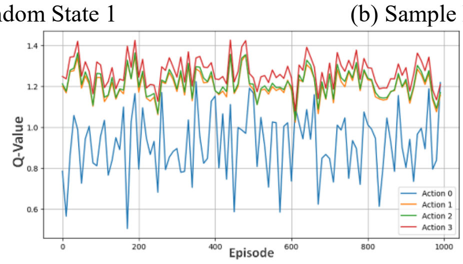
PER helps the agent focus more on experiences with high damping levels so that the agent can adapt to the environment faster and reach the ideal Q faster.It can be observed from the graph that Action 3-the red line-has consistently greater Q in respect to other actions-the agent chooses it constantly as the best among the actions in the current state. This more stable and dynamic Q-value distribution shows that agents with PER and Noisy Layer can make more diverse data explorations and usages, reducing risks of being stuck in suboptimal patterns. Without a doubt, this DQN is much more stable and explorative compared to the Q distribution from the standard DQN algorithm sans PER and Noisy Layer. PER enables the agents to be more focused on important experiences. On the other hand, it encourages natural exploration by Q-value variation. That avoids suboptimal patterns for the agents and, as a consequence, accelerates the convergence of the process towards an optimal adaptiveness of Q-distribution in the environment.



(a) Sample Random State 1



(b) Sample Random State 2



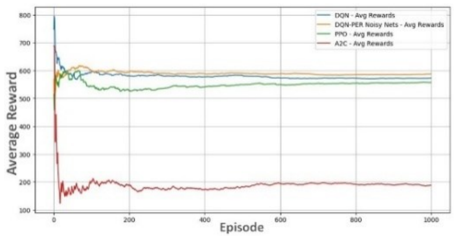
(c) Sample Random State 3

Fig. 6. Change of Q in DQN with PER and Noisy Layer algorithm

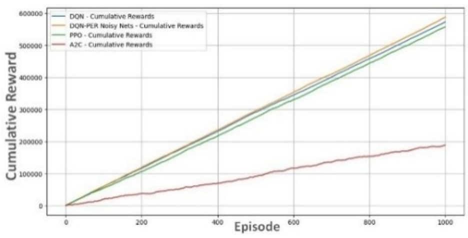
3.2.2. Generalization new data

In this model validation, sed data from other SI-Soil test results, which were more or less still in Karanganyar. The results that we compared were the average reward, cumulative reward, and MSE which are shown in the Fig. 7.

In Fig. 7 is a Cumulative Rewards Comparison graph, it can be seen that DQN-PER and NN have the highest cumulative reward acquisition compared to other models. Followed by DQN in Blue, then PPO in Green. DQN with PER and Noisy Nets show a stable increase in cumulative rewards, Then in the Average reward graph it can also be seen that the designed model is able to achieve convergence in 230 episodes, in contrast to the regular DQN algorithm which converges in episode 600 followed by the PPO algorithm which is more or less the same in episode 600 just converged. While A2C converges in episode 800. The MSE obtained in the proposed model is 2500, which is lower than the comparative method which is more than 3000.



(a) Average Reward



(b) Cumulative Reward Reward

Fig. 7. Reward on New Data

3.3. Action Distribution

Based on the action distribution in Fig. 8, each reinforcement learning method has a different fertilization pattern. The DQN-PER NN model tends to choose SP36 (action 2) and a combination of fertilizers (action 4) as the main strategy, indicating that the fertilizer combination is more optimal with a balance according to the soil conditions.

The standard DQN predominantly selects SP36 and KCL (actions 2 and 3), with the combination action (action 4) becoming more frequent towards the middle and end of the simulation. A2C demonstrates a more balanced distribution of actions across all fertilizer types, suggesting a more flexible fertilization strategy. In contrast, PPO exhibits a strong preference for actions 1 (urea), 2 (SP36), and 4 (combination) while maintaining a more stable decision pattern throughout the simulation period. Notably, all models favor the combination strategy (action 4), reinforcing its effectiveness in optimizing fertilization. Additionally, no model selects action

0 (no fertilization), indicating that each method prioritizes continuous fertilization to maximize agricultural yields.

The results show that the model can accurately provide fertilizer recommendations based on the nutrient conditions present in the soil, demonstrating its ability to adapt to real-world agricultural environments. By considering key soil parameters such as nitrogen (N), phosphorus (P), potassium (K), pH, moisture, and climatic factors, the model ensures that fertilizer recommendations match actual crop needs at different growth stages. Furthermore, the model's decision-making process mirrors real-world agronomic practices, where fertilizer distribution is dynamically adjusted based on environmental conditions.

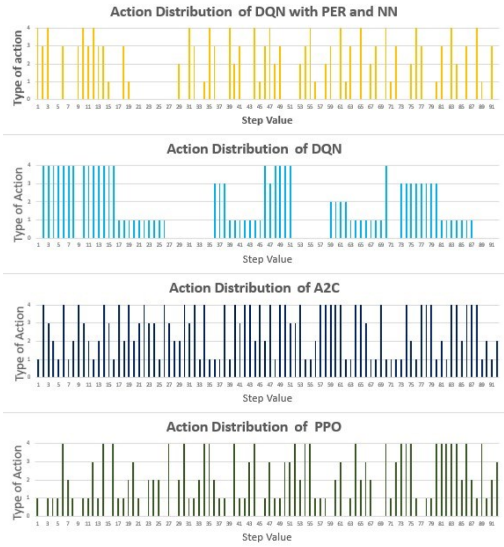
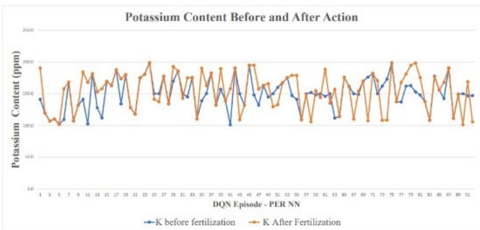
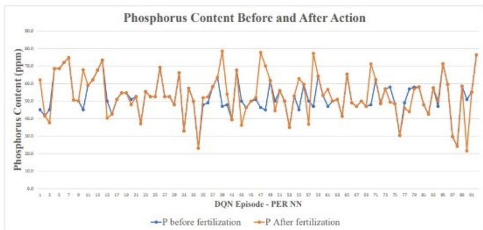


Fig. 8. Action Distribution

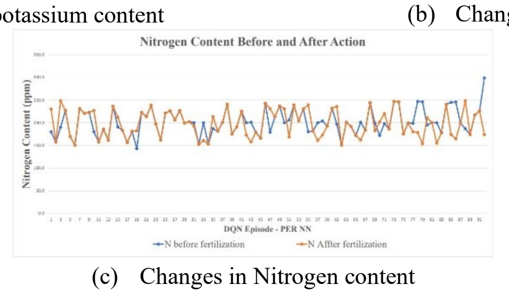
In Fig. 9, illustrates that DQN-PER NN effectively increases nitrogen, phosphorus, and potassium levels after fertilization. However, periodic declines in nutrient levels are observed, which could be attributed to over-fertilization, nutrient leaching, or interactions between soil properties and environmental conditions. This raises concerns about potential inefficiencies in nutrient utilization and environmental sustainability. A deeper analysis is required to determine whether these fluctuations are a result of excessive fertilizer application, soil absorption capacity, or external factors such as rainfall and irrigation. Furthermore, the study lacks a comparative evaluation against conventional fertilization practices, making it unclear how the proposed RL-based fertilization strategy aligns with real-world agricultural constraints and farmer preferences. Addressing these limitations would enhance the interpretability and practical relevance of the findings.



(a) Changes in potassium content



(b) Changes in Phosphorus content



(c) Changes in Nitrogen content

Fig. 9. Changes in nutrients after action

3.4. Fertilizer Result

The developed model enables efficient fertilization recommendations, as shown in Fig. 10. When tested on new data, DQN-PER NN provides optimal fertilizer amounts, leading to higher yields while using less fertilizer. This makes it both cost-effective and environmentally friendly. The bar chart shows that while all methods offer different recommendations, DQN-PER NN balances efficiency and productivity. It suggests 100 kg/ha of Urea, 150 kg/ha of SP36, and 125 kg/ha of KCL—lower than conventional methods. Additionally, the predicted harvest yield is 1.95 tons/ha, higher than DQN, A2C, and PPO. These results confirm that DQN-PER NN optimizes fertilization better than other methods. This approach not only maximizes yields but also promotes sustainable resource use. By recommending effective fertilization, it helps farmers achieve higher productivity while using inputs responsibly.

(a) Fertilizer Amount Recommendation (b) Crop Yield Prediction

Fig. 10. Crop Yield Prediction and Fertilizer Recommendation

4. CONCLUSION

This study proposes and explores the application of Reinforcement Learning (RL) in a soybean fertilization recommendation system by integrating Deep Q-Network (DQN) with Prioritized Experience Replay (PER) and Noisy Networks (NN), demonstrating superior cumulative rewards and faster convergence compared to standard DQN, A2C, and PPO. The system optimally recommends 150 kg/ha of SP36, 100 kg/ha of urea, and 125 kg/ha of KCL—lower than the fertilization rates suggested by PPO (175 kg/ha urea) and A2C (160 kg/ha urea)—while still achieving a soybean yield of 1.95 tons/ha, higher than conventional methods. However, several limitations must be addressed, including the high computational cost of PER and Noisy Networks, which may hinder real-world adoption in resource-constrained agricultural settings. The model's generalizability beyond soybean farming remains uncertain, as different crops and regions may require extensive recalibration due to variations in soil composition, climate, and nutrient demands. While the study suggests expanding the dataset with weather parameters, soil properties, and plant growth dynamics, it does not assess how these additions might impact model complexity and interpretability. Future research should explore transfer learning for cross-crop adaptability, irrigation integration for optimized resource use, and long-term ecological impacts to ensure sustainable fertilization practices. Additionally, while reducing fertilizer use can mitigate environmental effects, excessive optimization for short-term yield gains may have unintended consequences on soil health and nutrient cycling, requiring further investigation. By addressing these challenges, RL-based fertilization systems can be refined to support precision agriculture and promote sustainable food production.

Acknowledgments

The research conducted as partial support was given by Advance Creative Network Laboratory, Faculty of Electrical Engineering, Telkom University. The author would like to express his gratitude for the support given by Electrical Engineering, Telkom University

REFERENCES

[1] M. N. O. Sadiku, T. J. Ashaolu, and S. M. Musa, "International Journal of Scientific Advances Big Data in Agriculture", [Online]. Available: www.ijscia.com.
[2] M. Kijima and K. Fujita, "Effectiveness of Reward Functions for Deep Reinforcement Learning in Chick-Feeding System," in *15th International Congress on Advanced Applied Informatics Winter (IIAI-AAI-Winter)*, pp. 236–241, 2023, https://doi.org/10.1109/IIAI-AAI-Winter61682.2023.00051.
[3] H. Elver. *Global Food Systems in Crisis*. pp. 332–347, 2024, https://doi.org/10.4324/9781032699028-24.
[4] F. Kahar, I. Fadah, E. Wahyudi, and A. Wardhono, "Does Soybean in Indonesia Still Competitive?: A Policy Analysis Matrix Approach," *Jurnal Ekonomi dan Studi Pembangunan*, vol. 15, p. 89, Nov. 2023, https://doi.org/10.17977/um002v15i22023p089.

[5] F. Hellal and M. Abdelhamid, "Nutrient Management practices for enhancing Soybean (Glycine max L.) production," *Acta Biolo Colomb*, vol. 18, pp. 3–14, May 2013. https://dialnet.unirioja.es/servlet/articulo?codigo=9770226.
[6] H. Shaaibu and H. M. Rabiu, "Effects Of Npk And Poultry Manure Rates On Growth Performance And Yield Of Soybean (Glycine Max L. Merril)," *Fudma Journal Of Sciences*, vol. 7, no. 5, pp. 64–70, Oct. 2023, https://doi.org/10.33003/fjs-2023-0705-1228.
[7] O. Cahyono and S. Minardi, "Effect of Fast Dissolved Phosphorus Fertilizer on the Growth, Seed Product, and Phosphorus Uptake Efficiency of Soybean (Glycine max L.)," *Agrivita*, vol. 44, no. 1, pp. 21–30, 2022, https://doi.org/10.17503/agrivita.v44i1.3002.
[8] U. Mutammimah, S. Minardi, S. Suryono, O. Cahyono, and S. Sudadi, "Efektivitas Pupuk Organik pada Nitrogen, Fosfor, dan Produksi Kedelai di Tanah Masam," *Agrista: Jurnal Ilmiah Mahasiswa Agribisnis UNS*, vol. 4, no. 1, pp. 221–230, 2020, https://www.neliti.com/publications/365797/efektivitas-pupuk-organik-pada-nitrogen-fosfor-dan-produksi-kedelai-di-tanah-mas.
[9] G. Giusti, Y. María, and G. Almeida, "Environmental impacts assessment of maize, soybean, and wheat production in the southwest of São Paulo State: alternative scenarios for the substitution of chemical fertilization," *Revista Engenharia na Agricultura - REVENG*, vol. 30, pp. 328–346, Oct. 2022, https://doi.org/10.13083/reveng.v30i1.13864.
[10] N. Thenveettil, R. Bheemanahalli, K. Reddy, W. Gao, and K. Reddy, "Temperature and elevated CO2 alter soybean seed yield and quality, exhibiting transgenerational effects on seedling emergence and vigor," *Front Plant Sci*, vol. 15, Jul. 2024, https://doi.org/10.3389/fpls.2024.1427086.
[11] E. Lopez, S. Alvarez Prado, J. Rotundo, and J. Gerde, "Unravelling the environmental drivers determining the residual fraction of soybean seed," *Crop Sci*, vol. 65, no.1, p. e21406 Nov. 2024, https://doi.org/10.1002/csc2.21406.
[12] N. Vennampally *et al.*, "Envirotype-based delineation of environmental effects and genotype × environment interactions in Indian soybean (Glycine max, L.)," *Sci Rep*, vol. 14, May 2024, https://doi.org/10.1038/s41598-024-62613-y.
[13] Y. Dong, Z. Fu, Y. Peng, Y. Zheng, H. Yan, and L. Xinxing, "Precision fertilization method of field crops based on the Wavelet-BP neural network in China," *J Clean Prod*, vol. 246, p. 118735, Oct. 2019, https://doi.org/10.1016/j.jclepro.2019.118735.
[14] M. Chen *et al.*, "A reinforcement learning approach to irrigation decision-making for rice using weather forecasts," *Agric Water Manag*, vol. 250, p. 106838, 2021, https://doi.org/https://doi.org/10.1016/j.agwat.2021.106838.
[15] R. Tao *et al.*, *Optimizing Crop Management with Reinforcement Learning and Imitation Learning*. 2023. https://doi.org/10.24963/ijcai.2023/691.
[16] M. Madondo *et al.*, "A SWAT-based Reinforcement Learning Framework for Crop Management," *arXiv preprint*, 2023. [Online]. Available: https://arxiv.org/abs/2302.04988.
[17] Md. A. Jabed and M. A. Azmi Murad, "Crop yield prediction in agriculture: A comprehensive review of machine learning and deep learning approaches, with insights for future research and sustainability," *Heliyon*, vol. 10, no. 24, p. e40836, 2024, https://doi.org/10.1016/j.heliyon.2024.e40836.
[18] J. Liu, L. Lu, X. Zong, and B. Xie, "Nonlinear relationships in soybean commodities Pairs trading-test by deep reinforcement learning," *Financ Res Lett*, vol. 58, p. 104477, 2023, https://doi.org/10.1016/j.frl.2023.104477.
[19] M. Vogt and B. Rosman, "Analyzing Reinforcement Learning Algorithms for Nitrogen Fertilizer Management in Simulated Crop Growth," in *Proceedings of 43rd Conference of the South African Institute of Computer Scientists and Information Technologists*, vol. 85, pp. 146–157, 2022, https://doi.org/10.29007/1v4x.
[20] M. G. J. Kallenberg, H. Overweg, R. van Bree, and I. N. Athanasiadis, "Nitrogen management with reinforcement learning and crop growth models," *Environmental Data Science*, vol. 2, p. e34, 2023, https://www.cambridge.org/core/journals/environmental-data-science/article/nitrogen-management-with-reinforcement-learning-and-crop-growth-models/358749FAFAA4990B1448DAB7F48D641C.
[21] Z. Wang, K. Jha, and S. Xiao, "Continual Reinforcement Learning for Intelligent Agricultural Management under Climate Changes," *Computers, Materials and Continua*, vol. 81, no. 1, pp. 1319–1336, 2024, https://doi.org/10.32604/cmc.2024.055809.
[22] A. Din, M. Y. Ismail, B. Shah, M. Babar, F. Ali, and S. U. Baig, "A deep reinforcement learning-based multi-agent area coverage control for smart agriculture," *Computers and Electrical Engineering*, vol. 101, p. 108089, 2022, https://doi.org/10.1016/j.compeleceng.2022.108089.
[23] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized Experience Replay," 2016. [Online]. Available: https://arxiv.org/abs/1511.05952.
[24] D. Panda, O. Turner, S. Das, and M. Abusara, "Prioritized experience replay based deep distributional reinforcement learning for battery operation in microgrids," *J Clean Prod*, vol. 434, p. 139947, Nov. 2023, https://doi.org/10.1016/j.jclepro.2023.139947.
[25] A. Mitra, G. J. Pappas, and H. Hassani, "Temporal Difference Learning with Compressed Updates: Error-Feedback meets Reinforcement Learning," *arXiv preprint*, 2024. [Online]. Available: https://arxiv.org/abs/2301.00944.
[26] K. Asadi, S. Sabach, Y. Liu, O. Gottesman, and R. Fakoor, "TD Convergence: An Optimization Perspective," arXiv *preprint*, 2023. [Online]. Available: https://arxiv.org/abs/2306.17750.
[27] S. Han, W. Zhou, J. Lu, J. Liu, and S. Lü, "NROWAN-DQN: A stable noisy network with noise reduction and online weight adjustment for exploration," *Expert Syst Appl*, vol. 203, p. 117343, 2022, https://doi.org/10.1016/j.eswa.2022.117343.

[28] A. Choudhari, D. Bhoyar, and W. P. Badole, "Design of an IoT based Monitoring System Utilizing Deep Reinforcement Learning and Ensemble Learning for Precision Agriculture," *International Conference on Sustainable Computing and Smart Systems (ICSCSS)*, pp. 845-852, 2024, https://doi.org/10.1109/ICSCSS60660.2024.10625266.
[29] H. Chen *et al.*, "Reinforcement learning-based assimilation of the WOFOST crop model," *Smart Agricultural Technology*, vol. 9, p. 100604, 2024, https://doi.org/10.1016/j.atech.2024.100604.
[30] A. Upadhyaya and Rahul, "Optimizing Plant Health with Q-Learning: A Deep Reinforcement Learning Approach," in *2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT)*, pp. 1–5, 2024, https://doi.org/10.1109/ICCCNT61001.2024.10725377.
[31] D. Elavarasan and P. M. D. Vincent, "Crop Yield Prediction Using Deep Reinforcement Learning Model for Sustainable Agrarian Applications," *IEEE Access*, vol. 8, pp. 86886–86901, 2020, https://doi.org/10.1109/ACCESS.2020.2992480.
[32] W. Su, M. Gao, X. Gao, X. Zhu, and D. Fang, "Adaptive decision-making with deep Q-network for heterogeneous unmanned aerial vehicle swarms in dynamic environments," *Computers and Electrical Engineering*, vol. 119, p. 109621, 2024, https://doi.org/10.1016/j.compeleceng.2024.109621.
[33] Y. Lu, X. Yu, Z. Hu, and X. Wang, "Convolutional neural network combined with reinforcement learning-based dual-mode grey wolf optimizer to identify crop diseases and pests," *Swarm Evol Comput*, vol. 94, p. 101874, 2025, https://doi.org/10.1016/j.swevo.2025.101874.
[34] S. Deivasigamani, A. J. M. Rani, R. Natchadalingam, P. Vijayakarthik, G. B. S. Kumar, and P. C. S. Reddy, "Crop Yield Prediction Using Deep Reinforcement Learning," in *2023 Second International Conference on Trends in Electrical, Electronics, and Computer Engineering (TEECCON)*, pp. 137–142, 2023, https://doi.org/10.1109/TEECCON59234.2023.10335861.
[35] W.-T. Sung and I. G. T. Isa, "A Deep Reinforcement Learning Approach in IoT-based Smart Plantation Monitoring System," in *2023 International Automatic Control Conference (CACS)*, 2023, pp. 1–6. https://doi.org/10.1109/CACS60074.2023.10326165.
[36] C. Zhang, M. Yao, W. Chen, S. Zhang, D. Chen, and Y. Wu, "Gradient Descent Optimization in Deep Learning Model Training Based on Multistage and Method Combination Strategy," *Security and Communication Networks*, vol. 2021, 2021, https://doi.org/10.1155/2021/9956773.
[37] R. S. Alonso, I. Sittón-Candanedo, R. Casado-Vara, J. Prieto, and J. M. Corchado, "Deep Reinforcement Learning for the management of Software-Defined Networks in Smart Farming," in *2020 International Conference on Omni-layer Intelligent Systems (COINS)*, pp. 1–6, 2020, https://doi.org/10.1109/COINS49042.2020.9191634.
[38] H. M. Yassine, K. Roufaida, V. P. Shkodyrev, M. Abdelhak, L. Zarour, and R. Khaled, "Intelligent Farm Based on Deep Reinforcement Learning for optimal control," in *2022 International Symposium on iNnovative Informatics of Biskra (ISNIB)*, pp. 1–6, 2022, https://doi.org/10.1109/ISNIB57382.2022.10076088.
[39] M. Ramicic and A. Bonarini, "Correlation minimizing replay memory in temporal-difference reinforcement learning," *Neurocomputing*, vol. 393, pp. 91–100, 2020, https://doi.org/10.1016/j.neucom.2020.02.004.
[40] L. Wang and X. Wang, "Enhanced Deep Reinforcement Learning Strategy for Energy Management in Plug-in Hybrid Electric Vehicles with Entropy Regularization and Prioritized Experience Replay," *Energy Engineering*, vol. 121, no. 12, pp. 3953–3979, 2024, https://doi.org/https://doi.org/10.32604/ee.2024.056705.
[41] H. Hassani, S. Nikan, and A. Shami, "Improved exploration–exploitation trade-off through adaptive prioritized experience replay," *Neurocomputing*, vol. 614, p. 128836, 2025, https://doi.org/10.1016/j.neucom.2024.128836.
[42] C. Wang *et al.*, "Prioritized sum-tree experience replay TD3 DRL-based online energy management of a residential microgrid," *Appl Energy*, vol. 368, p. 123471, 2024, https://doi.org/10.1016/j.apenergy.2024.123471.
[43] M. Paniri, M. B. Dowlatshahi, and H. Nezamabadi-pour, "Ant-TD: Ant colony optimization plus temporal difference reinforcement learning for multi-label feature selection," *Swarm Evol Comput*, vol. 64, p. 100892, 2021, https://doi.org/10.1016/j.swevo.2021.100892.
[44] J. Gao, X. Li, W. Liu, and J. Zhao, "Prioritized Experience Replay Method Based on Experience Reward," in *International Conference on Machine Learning and Intelligent Systems Engineering (MLISE)*, pp. 214–219, 2021, https://doi.org/10.1109/MLISE54096.2021.00045.
[45] J. Park, J. Choi, S. Nah, and D. Kim, "Distributional and hierarchical reinforcement learning for physical systems with noisy state observations and exogenous perturbations," *Eng Appl Artif Intell*, vol. 123, p. 106465, 2023, https://doi.org/10.1016/j.engappai.2023.106465.
[46] P. Yadav, A. Mandal, and K. Dhakal, "Comparative Study Of Evaluation Of Soil Fertility Status In Rice Zone, Morang," *Tropical Agroecosystems*, vol. 2, no. 1, pp. 12-25, Apr. 2021, https://doi.org/10.26480/taec.01.2021.12.25.
[47] M. A. Budiyanto and M. H. Lubis, "Physical reviews of solar radiation models for estimating global solar radiation in Indonesia," *Energy Reports*, vol. 6, pp. 1206–1211, 2020, https://doi.org/10.1016/j.egyr.2020.11.053.
[48] M. A. Raza *et al.*, "Sugarcane/soybean intercropping enhances crop productivity, nutrient uptake, and net economic return with reduced inputs," *Field Crops Res*, vol. 314, p. 109428, 2024, https://doi.org/10.1016/j.fcr.2024.109428.
[49] G. Verma *et al.*, "Productivity, soil health, and carbon management index of soybean-wheat cropping system under double zero-tillage and natural-farming based organic nutrient management in north-Indian plains," *Science of The Total Environment*, vol. 917, p. 170418, 2024, https://doi.org/10.1016/j.scitotenv.2024.170418c.
[50] Y. Liu *et al.*, "An integrated tool for cost-effectively applying nutrient management practices for corn, soybeans, and wheat," *Science of The Total Environment*, vol. 955, p. 177110, 2024, https://doi.org/10.1016/j.scitotenv.2024.177110.

BIOGRAPHY OF AUTHORS

Alfian Fakhrezi, born in Majalengka, West Java, Indonesia, is an enthusiastic person with high dedication in the fields of signal processing, agriculture and machine learning. In 2022, he successfully obtained a Bachelor of Computer Engineering from Telkom University. Currently, Alfian is pursuing a Master of Electrical Engineering at Telkom University, focusing on signal processing. For professional or academic inquiries, Alfian can be contacted via email at Alfian.fakhrezi@gmail.com

Gelar Budiman, was born in 1978 in Bandung, Indonesia's West Java. He graduated from Telkom Technology College (STTTelkom) in 2005 with a Bachelor of Engineering and a Master of Electrical Engineering. In 2021, he earned his PhD from Bandung Institute of Technology's School of Electrical and Informatics Technology. From 2008 to the present, he has been a lecturer at the Telkom Technology Institute (ITTelkom) in Telecommunication Engineering. Since 2018, he has served as an assistant professor. Signal processing, information concealment, computer vision, artificial intelligence, wireless communication, quantum signal processing, and security are among his specialties. He has written nine articles in prestigious international journals, seven in Scopus International Proceedings, nine in intellectual property rights, and one book. He is also an IEEE member.
He can be contacted at email: gelarbudiman@telkomuniversity.ac.id.

Doan Perdana, received bachelor's and master's degrees at Telkom University, Bandung, and he earned his doctorate at the University of Indonesia, Depok. Indonesia. He has authored some high-prestigious journal and conference papers. He is also a Lecturer with Telkom University, which is one of the best private universities in Indonesia. His research interests include Telecommunication Engineering, computer network, and Network Engineering.
He can be contacted at email: doanperdana@ums.ac.id.