


Article

A Machine Learning-Based Energy Management Agent for Fine Dust Concentration Control in Railway Stations

Kyung-Bin Kwon ¹ , Su-Min Hong ², Jae-Haeng Heo ², Hosung Jung ³  and Jong-young Park ^{3,*} 

¹ Department of Electrical and Computer Engineering, The University of Texas at Austin, 2501 Speedway, Austin, TX 78712, USA

² Raon Friends, 267 Simin-daero, Dongan-gu, Anyang-si 14054, Gyeonggi-do, Republic of Korea

³ Korea Railroad Research Institute, 176 Cheoldobangmulgwan-ro, Uiwang-si 16105, Gyeonggi-do, Republic of Korea

* Correspondence: jypark@krii.re.kr; Tel.: +82-31-460-5731

Abstract: This study developed a reinforcement learning-based energy management agent that controls the fine dust concentration by controlling facilities such as blowers and air conditioners to efficiently manage the fine dust concentration in the station. To this end, we formulated an optimization problem based on the Markov decision-making process and developed a model for predicting the concentration of fine dust in the station by training an artificial neural network (ANN) based on supervised learning to develop the transfer function. In addition to the prediction model, the optimal policy for controlling the blower and air conditioner according to the current state was obtained based on the ANN to which the Deep Q-Network (DQN) algorithm was applied. In the case study, it is confirmed that the ANN and DQN of the predictive model were trained based on the actual data of Nam-Gwangju Station to converge to the optimal policy. The comparison between the proposed method and conventional method shows that the proposed method can use less power consumption but achieved better performance on reducing fine dust concentration than the conventional method. In addition, by increasing the value of the ratio that represents the compensation due to the fine dust reduction, the learned agent achieved more reduction on the fine dust concentration by increasing the power consumption of the blower and air conditioner.

Keywords: Deep Q-network; energy management; particulate matter; reinforcement learning; supervised learning



Citation: Kwon, K.-B.; Hong, S.-M.; Heo, J.-H.; Jung, H.; Park, J.-y. A Machine Learning-Based Energy Management Agent for Fine Dust Concentration Control in Railway Stations. *Sustainability* **2022**, *14*, 15550. <https://doi.org/10.3390/su142315550>

Academic Editors: Antonio Caggiano, Damien Guilbert and Phatiphat Thounthong

Received: 22 September 2022

Accepted: 16 November 2022

Published: 22 November 2022

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2022 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Fine dust is an air pollutant designated by the International Cancer Institute under the World Health Organization (WHO) as a group 1 carcinogen that has been confirmed to cause cancer in humans [1]. Fine dust can be divided into fine dust with a diameter of less than 10 μm (PM 10) and ultra-fine dust (PM 2.5) with a diameter of less than 2.5 μm according to its size, and the accumulation of ultra-fine dust can rapidly deteriorate indoor air quality [2]. In particular, in the case of urban railway stations, various pollutants such as wear particles from the railway, congestion of users, scattering by train wind, and maintenance work of tunnels exist together [3,4]. Therefore, it is vulnerable to air pollution including fine dust due to its difficulty in ventilation. To this end, it is possible to reduce the concentration of fine dust by using a blower or air conditioner to simultaneously circulate the air inside and outside the station through the filter [5]. In this regard, studies [6,7] to predict the concentration of fine dust in stations and research [8] on the establishment of a reduction control system are being actively conducted in order to manage the concentration of fine dust in stations.

However, the change in fine dust concentration depends on the control of the blower and air conditioner depending on the depth, congestion, and structure of each station [9]. Considering this, it is necessary to directly model the interaction between the fine dust

reduction facility and the environment. These problems can be solved by applying machine learning (ML) that can find optimal policies without directly building an interaction model with the environment. In this condition, the power cost increases as the air conditioner and blower are controlled to reduce fine dust, so a ML-based energy management agent is needed to consider the power cost and the resulting fine dust reduction. Here, we refer the controller that manages the fine dust with blowers and air conditioners as “the agent”, as it refers to the learner and decision maker based on RL [10].

Machine learning (ML) means “an algorithm that can perform requested tasks by using and analyzing data to perform specific tasks” [11]. ML can be largely divided into supervised learning, which uses given data and labels to predict unknown states or values, unsupervised learning, which finds useful patterns in the data itself, and reinforcement learning (RL), which maximizes the long-term benefits of agents’ interactions with the environment [12]. Among them, supervised learning mainly generates predictive models, and RL is used to find optimal policies in control problems.

There have been several research that predict the fine dust concentrations in various places, as summarized in Table 1. In [13], PM 10 concentration level has been predicted based on the 9 years of data in Ankara, Turkey with ML algorithms including LASSO, SVR and ANN. In [14], long-term spatially continuous monthly PM 2.5 level has been predicted through ML algorithms with aerosol optical depth (AOD) data derived from satellite images. On the other hand, there have been research on comparing several ML and statistical models for predicting indoor air quality [15] or evaluating different ML approaches such as Multiple Additive Regression Trees (MART), Deep Feedforward Neural Network (DFNN) and Long Short-Term Memory (LSTM) when forecasting PM 2.5 concentration levels [16]. In [17], several learning algorithms including Support Vector Machine (SVM), AdaBoost (AdB) and Multilayer Perceptron (MLP) are used to forecast CO₂ levels. Similarly, Artificial Neural Network (ANN), Random Forest (RF), SVM and LSTM models are applied to predict air quality in [18,19]. In [20], a hybrid deep learning model that combines Convolution Neural Network (CNN) and LSTM is developed to predict the PM 2.5 concentration level to be used in the early warning and control management.

Table 1. Machine learning-based literature summary.

Reference	Subject	Algorithm
[13] Bozdag et al., 2020	PM 10 level prediction	LASSO, SVR, ANN
[14] Xu et al., 2018	PM 2.5 level prediction	SVM, LASSO, RF
[15] Wei et al., 2019	Air quality control	ANN
[16] Karimian et al., 2019	PM 2.5 level prediction	MART, DFNN, LSTM
[17] Taheri et al., 2021	CO ₂ level prediction	SVM, AdB, MLP
[18] Kang et al., 2018	Air quality prediction	ANN, RF
[19] Janarthanan et al., 2021	Air quality prediction	SVR, LSTM
[20] Du et al., 2019	PM 2.5 level prediction	CNN, LSTM
[21] Kwon et al., 2021	PM 2.5, PM 10 level control	DQN

The recent works have provided state-of-the-art techniques on forecasting fine dust concentration, but the research area has not been extended to combine the fine dust concentration prediction model to an agent that controls the fine dust level with energy facilities such as blowers and air conditioners. To this end, we need to adopt not only the supervised learning for the prediction model, but also the RL for training the agent.

RL has the advantage of finding optimal policies in situations where the agent’s behavior and its interaction with a given environment are unknown [12]. Therefore, if ML is used to construct fine dust management problems in stations where uncertainty exists, we can find optimal policies directly using uncertainty without modeling as a separate probability distribution.

In this regard, ref. [21] established an energy management agent based on RL as a previous study. In order to build an agent, a linear transition function and a compensation

function were developed by selecting an element in a linear relationship with the concentration of fine dust in history as a component of the current state, and based on this, an agent based on the Deep-Q network (DQN) algorithm was developed. However, when developing the transfer kernel, based on the assumption that it is linear, elements that do not form a linear relationship were excluded from the current state. Accordingly, there was a problem in that the accuracy of the transfer kernel for predicting the concentration of fine dust in the station decreased. As a summary, here are the drawbacks of the previous research:

- Most previous works focused on predicting the fine dust concentration level or air quality with various, but it does not consider the control problem to resolve the issue
- Though there was a paper that developed an RL-based agent, it considered linear mapping in the fine dust concentration, which has limited accuracy

In this study, to solve this problem, we propose a method to learn an artificial neural network (ANN) based on supervised learning instead of a linear transfer kernel and use it as a transfer kernel and connect it with the existing DQN-based agent model. The contribution of the paper is as follows:

- Developing a RL-based agent that controls the fine dust concentration in the railway station using DQN method
- Developing artificial neural network (ANN) that predicts the changes in fine dust concentration in the stations to train the agent with a small amount of data
- Combining ANN prediction model and DQN agent to train the agent with offline learning

To this end, in Section 2, system modeling based on the Markov decision-making process was constructed. In Section 3, a model for predicting changes in the concentration of fine dust in stations according to the control of fine dust reduction facilities was developed using an ANN based on supervised learning. In Section 4, the DQN-based agent was developed using the ANN developed in Section 3 as a transfer kernel, and the optimal policy was obtained through this. In Section 5, the performance of the agent learned through a case study was analyzed based on the actual data of Nam-Gwangju Station, and in Section 6, the conclusion of this study was described.

2. System Modeling Based on Markov Decision Process

In order to build a RL-based energy management agent, we first assumed Markov properties, and based on this, we constructed a system modeling based on the Markov Decision Process [10]. The Markov decision process can be configured by defining State, Action, Transition kernel, Reward, and Discount factor. Depending on the system, each can be defined as follows.

First, the state can be defined by time (t), the concentration of fine dust inside and outside the station ($I_t^{(1)}, I_t^{(2)}; O_t^{(1)}, O_t^{(2)}$), humidity (H_t^i, H_t^o), and temperature (T_t^i, T_t^o). In this case, the fine dust concentration can be expressed as a concentration of fine dust (PM 2.5) with a diameter of less than 2.5 μm and a concentration of fine dust with a diameter of less than 10 μm . That is, the state at time can be expressed as Equation (1):

$$s_t = \{t, I_t^{(1)}, I_t^{(2)}, O_t^{(1)}, O_t^{(2)}, H_t^i, H_t^o, T_t^i, T_t^o\} \quad (1)$$

Next, the action is a choice made based on a policy in a given state, where it refers to the power usage of the blower and the air conditioner. Assuming there are a total of K blowers and L air conditioners, the behavior at time is as Equation (2), where v and w indicate the power usage of the blower and air conditioner, respectively.

$$a_t = \{v_t^{(1)}, \dots, v_t^{(K)}, w_t^{(1)}, \dots, w_t^{(L)}\} \quad (2)$$

Here, each action can be a discrete value or a continuous value depending on the control method.

Next, the transition kernel means the probability of moving to the next state s_{t+1} when an action a_t is performed in the current state s_t . By the Markov property, as in Equation (3), the probability of moving to s_t can be expressed as a conditional probability for the current state s_t , meaning that it is not affected by any previous state [22].

$$\Pr(s_{t+1} | \{s_\tau\}_{\tau=1}^t) = \Pr(s_{t+1} | s_t) \quad (3)$$

Since it cannot be operated in practice, trial and error cannot be carried out to find the optimal policy of the blower and air conditioner like the general RL method. Therefore, based on the existing data, an artificial neural network was constructed to predict the concentration of fine dust in the station according to the control of the blower and air conditioner, and this was used as a transfer kernel. More details on this are described in Section 3.

Reward refers to the reward obtained when an action a_t is taken in a state s_t . The reward r_t at time t is expressed as a function of s_t and a_t . In this paper, the power consumption of the blower and air conditioner and the reduction in the concentration of fine dust in the station are considered as reward. When the electricity price at time t is given by p_t , the total electricity cost c_t is as in Equation (4):

$$c_t = p_t \left(\sum_{k=1}^K v_t^{(k)} + \sum_{l=1}^L w_t^{(l)} \right) \quad (4)$$

Then, the amount of reduction in PM 2.5 and PM 10 fine dust concentrations due to the control of the blower and air conditioner through the use of power above can be expressed as $\Delta_t^{(1)} = i_t^{(1)} - i_{t-1}^{(1)}$, $\Delta_t^{(2)} = i_t^{(2)} - i_{t-1}^{(2)}$ respectively. As a result, the reward function can be expressed as Equation (5):

$$r_t(s_t, a_t) = \rho(\Delta_t^{(1)} + \Delta_t^{(2)}) - c_t \quad (5)$$

In this case, ρ represents the ratio between the reward due to the reduction in the concentration of fine dust and the total power cost; the larger the value of ρ , the larger the reward is, due to the reduction in the concentration of fine dust.

Lastly, the discount factor γ means the ratio between the present reward and the future reward and is determined as a value in the range $(0, 1)$. As the value become smaller, it implies that the present reward is considered more valuable than the value of the future reward. In this paper, because finite time is considered, it is set to $\gamma = 1$.

3. Model for Predicting the Concentration of Fine Dust

As discussed above, a supervised learning-based prediction model using an ANN was developed to predict the change in the concentration of fine dust in the station according to the control of the blower and air conditioner.

The ANN of the predictive model takes the elements and actions of the current state as input values, and fine dust concentrations (PM 2.5, PM 10) in the station of the next time as output values. In this case, if the input vector value in the k -th layer is X_k and the output vector value in the subsequent $(k + 1)$ -th layer is Y_{k+1} , Y_{k+1} can be calculated as follows [23].

$$Y_{k+1} = \sigma(X_k^T W_k + b_k) \quad (6)$$

Here, W_k and b_k denote a weight matrix and a bias value between the k -th layer and the $(k + 1)$ -th layer, respectively, and $\sigma(\cdot)$ denotes an activation function of the $(k + 1)$ -th layer. In supervised learning, the forward propagation algorithm is performed through the process of Equation (6), and the final predicted value Y is output from the last output layer,

and this is compared with the actual value O . In this case, the loss function l is defined as the following mean-squared error (MSE) value [24].

$$l = \|Y - O\|_2^2 \quad (7)$$

After the loss function is calculated as in Equation (7), a backward propagation algorithm is then performed to update the weights. In order to improve prediction accuracy, the value of l must be minimized, so each element w_i of the weight matrix is updated using gradient descent. That is, the value of w_i in the $(n + 1)$ -th iterative learning can be calculated as in Equation (8).

$$w_i^{n+1} = w_i^n - \alpha \frac{\partial l}{\partial w_i^n} \quad (8)$$

Therefore, each element of the weight matrix is updated in a direction to decrease the value of the loss function l , and through iterative learning, the value of the loss function l approaches the minimum value. This means that the difference between the predicted value of the ANN model and the actual value is minimized.

4. Development of Energy Management Agent

4.1. Deep Q-Network Based Agent Development

Based on the reward function and discount factor defined in Section 2, the objective function is to find the optimal policy π for controlling the blower and air conditioner that maximizes the expected total discounted reward, as indicated in Equation (9).

$$J(\pi) = \max_{\pi} E_{\pi} \left[\sum_{t=1}^T \gamma^t r_t \right] \quad (9)$$

Specifically, when the discount factor is considered, the optimal policy can be defined as a policy that maximizes the average of the sum of the reward r_t for each time period from time $t = 1$ to $t = T$. The above optimization problem can be transformed into the following problem by parameterizing the policy π as a parameter θ .

$$J(\theta) = \max_{\theta} E_{\pi_{\theta}} \left[\sum_{t=1}^T \gamma^t r_t \right] \quad (10)$$

That is, instead of directly obtaining the optimal policy, by parameterization, the policy can be obtained by expressing the policy as a parameter θ and finding the optimal θ . Since this paper uses the Deep-Q Network (DQN) algorithm based on ANN, the parameter θ refers to the weight matrix of the Q-network [25]. In the DQN algorithm, the ANN predicts the approximate value $Q(s_t, a_t)$ of the Q function for each action a_t with respect to the input value s_t .

$$Q(s_t, a_t) = E_{\pi_{\theta}} \left[\sum_{\tau=t}^T \gamma^{(\tau-t)} r_{\tau}(s_{\tau}, a_{\tau}) | s_t, a_t \right] \quad (11)$$

$Q(s)$ in Equation (11) means the expected value of the total reward after executing the action a_t in the state s_t at time t . Therefore, based on the above value, the optimal policy π can be defined as selecting an action that maximizes the expected value of the total reward.

$$\pi : a_t = \arg \max_{a'} Q(s_t, a') \quad (12)$$

Meanwhile, the Q function satisfies the Bellman equation in Equation (13) [26].

$$Q^*(s_t, a_t) = r_t(s_t, a_t) + \gamma E_{s_{t+1}} \left[\max_{a_{t+1}} Q^*(s_{t+1}, a_{t+1}) | s_t, a_t \right] \quad (13)$$

As the predicted value of the Q-network that predicts the Q function value is more accurate, the difference between the left side and the right side of Equation (13) decreases. Therefore, it is possible to construct a Q-network that predicts the Q function value by finding the optimal parameter θ value to minimize the loss function of Equation (14) below.

$$L(\theta) = E_{s_t, a_t, s_{t+1}} \left[\left(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta') - Q(s_t, a_t; \theta) \right)^2 \right] \quad (14)$$

The parameters θ and θ' mean the parameters of the train network and the target network, respectively, and in the algorithm applied in this paper, the fixed target network method was applied to solve the problem of convergence instability during the optimization process [27]. This is a method in which the parameter θ' of the target network is fixed at θ^- value instead of being applied every time during repeated learning like the parameter θ of the Train network when updating, and θ^- is updated once in every N_0 updates. Since the parameter θ' on the right side is fixed while the value of the left side of Equation (13) is updated to reduce the value of the loss function, convergence occurs more stably.

To minimize Equation (14), the optimal parameter θ can be obtained by applying the gradient descent method. The gradient obtained by partial differentiation of the loss function with respect to the parameter θ is as follows.

$$\nabla L(\theta) = E_{s_t, a_t, s_{t+1}} \left[-2 \left(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-) - Q(s_t, a_t; \theta) \right) \nabla_{\theta} Q(s_t, a_t; \theta) \right] \quad (15)$$

Since it is difficult to obtain the expected value for all $\{s_t, a_t, s_{t+1}\}$ combinations, instead, as in Equation (16), a trajectory is constructed through sampling based on the current policy, and the average value is calculated for the approximate value of the slope.

$$\nabla \hat{L}(\theta) = \left(-\frac{2}{|\psi|} \right) \sum_{t \in \psi} \left[\left(r_t + \gamma \max_{a_{t+1}} Q(s_{t+1}, a_{t+1}; \theta^-) - Q(s_t, a_t; \theta) \right) \nabla_{\theta} Q(s_t, a_t; \theta) \right] \quad (16)$$

In this case, the ϵ -greedy method was applied to ensure sufficient exploration at the beginning of the algorithm. The ϵ -greedy method selects the optimal action defined in Equation (13) with a probability of $(1 - \epsilon)$, and selects a random action with a probability of ϵ [10]. The value of ϵ decreases as the iterative learning progresses, so the final policy follows Equation (13). In addition, the experience replay method was applied [28]. First, the experience replay method stores the sampling result in the memory $\Phi = \{(s_t, a_t, s_{t+1}, r_t)\}$, and when calculating the loss function, randomly selects the samples stored in the memory Φ to create mini-batch Ψ and calculates Equation (14). This makes it possible to efficiently obtain the slope of the loss function by utilizing the previous sample without the need to make a new sample each time the μ is updated.

4.2. DQN-Based Energy Management Agent Using ANN Prediction Model

Finally, we develop the DQN-based energy management agent, which works as an agent to determine the control of energy facilities based on the given state information. Here, the agent is trained through the process shown in Figure 1, which utilizes the fine dust concentration prediction model developed in Section 3 as a transition function.

First, for the current state s_t given at time t , the action a_t is determined through the DQN algorithm. Next, s_t and a_t are used as input values of the station's fine dust concentration prediction model. This value again becomes the input value of DQN at time $t + 1$. The whole process that combines ANN prediction model and DQN-based energy management agent is depicted in Figure 2 as a flowchart.

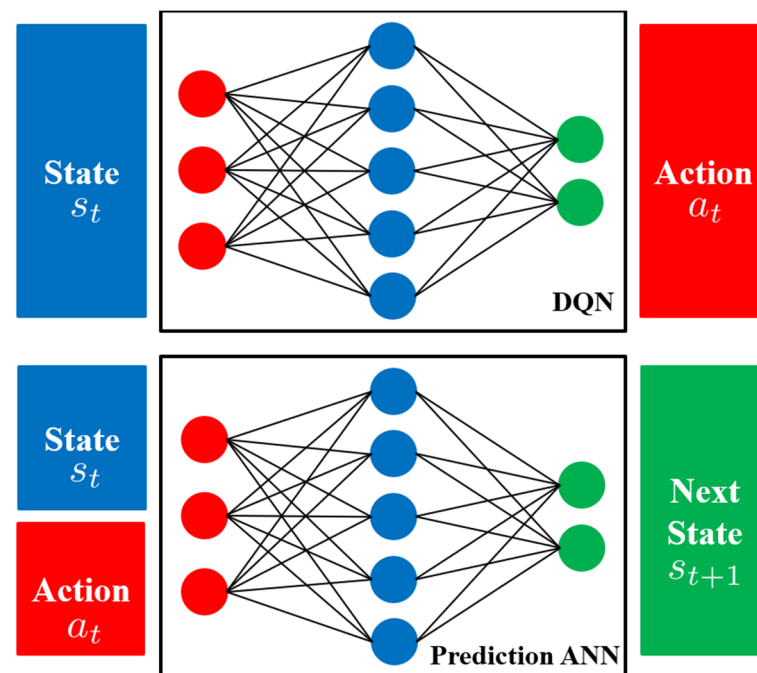


Figure 1. Energy management agent operation process.

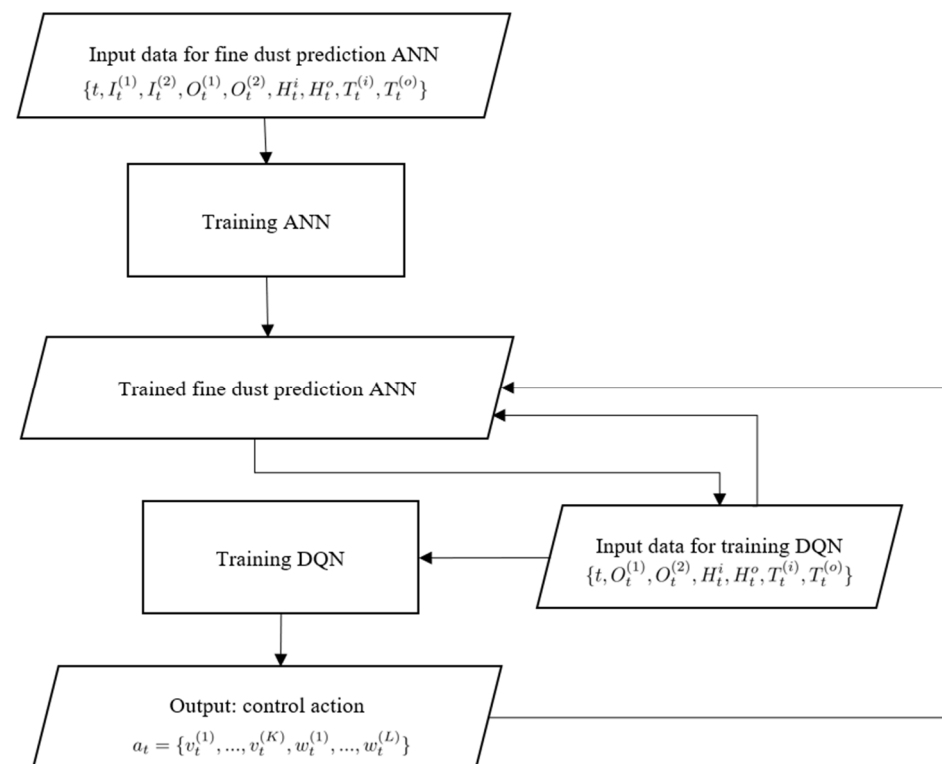


Figure 2. Flowchart of the process combining ANN prediction model and DQN-based energy management agent.

Finally, the blower and air conditioner control algorithm using the fine dust concentration prediction model in the station and the DQN method is shown in Algorithm 1 (DQN-based energy management agent optimal operation algorithm using ANN prediction model).

Algorithm 1: Machine learning-based optimal control of energy management agent

```

1  Hyperparameter: Discounting factor  $\gamma = 1$ , learning rate  $\eta > 0$ ,  $\epsilon$ -greedy
   coefficient  $\kappa \in (0, 1)$ , mini-batch size  $|\Phi|$ , target network update interval  $N_0$ , maximum
   number of iterations for ANN  $N^E$ , maximum number of iterations for DQN  $N^Q$ 
2  Inputs: Exploration time horizon  $T$ 
3  Initialize: Initial parameter of prediction ANN  $W_0$ ,  $\epsilon$ -greedy probability
    $\kappa \in (0, 1]$ , Replay memory  $\Phi = \emptyset$ , initial  $n = 0$ , initial  $\theta'$ , initial target network
   parameters  $\theta^- = \theta'$ 
4  (1) Concentration of fine dust prediction ANN
5  while  $n^E \leq N^E$  do
6      for  $t = 0, \dots, T$  do
7          Compute prediction of concentration of fine dust  $Y_{t+1}$  with Equation (6) using
           the data  $s_t, a_t$  as inputs
8          Compute  $\ell$  by Equation (7) with measured data  $Q_{t+1}$ .
9          Update  $W$  by Equation (8)
10     end
11     Update iteration index:  $n^E \leftarrow n^E + 1$ 
12 end
13 (2) Deep Q-Network Training
14 while  $n^Q \leq N^Q$  do
15     for  $t = 0, \dots, T$  do
16         Select a random action  $a_t$  with probability  $\epsilon$ ; otherwise,
            $a_t^* = \operatorname{argmax}_{a_t} Q(s_t, a_t; \theta')$ 
17         Compute  $s_{t+1}$  by using  $s_t$  and  $a_t$  as inputs of the prediction ANN.
18         Compute  $r_t$  with Equation (5).
19         Store the tuple  $(s_t, a_t, s_{t+1}, r_t)$  in  $\Phi$ 
20         Select a random mini-batch  $\emptyset$  with size  $|\Phi|$  from  $\Phi$ .
21         Compute the gradient estimate of loss function based on Equation (15).
22         Parameter updates:  $\theta' \leftarrow \theta' - \eta \hat{\nabla} \mathcal{L}(\theta')$ 
23         if  $t/N_0$  is integer then
24             target network parameter update  $\theta^- \leftarrow \theta'$ .
25         end
26         Update  $\epsilon$ :  $\epsilon = \kappa \epsilon$ .
27     end
28     Update iteration index:  $n^Q \leftarrow n^Q + 1$ 
29 end

```

5. Case Study

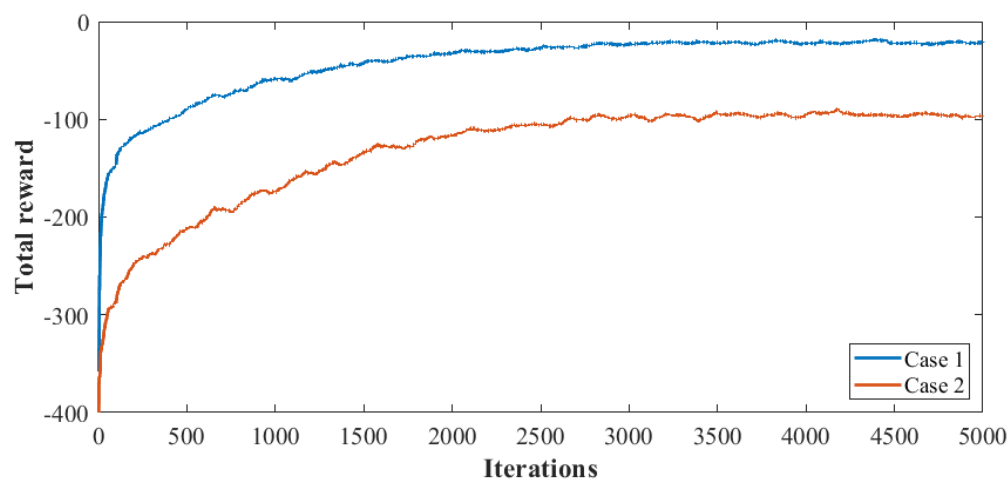
In order to prove the effectiveness of the algorithm presented in Algorithm 1, a case study was conducted based on the data of Nam-Gwangju Station as in the case study in [8]. First, the definition of Equation (1) was used for the current state. Next, in the case of action, it was assumed that three operation modes could be selected for three blowers ($K = 3$) and two operation modes could be selected for two air conditioners ($L = 2$). Accordingly, the total number of selectable actions was set to $3^2 \times 2^2 = 108$. Accordingly, the ANN input nodes of the fine dust prediction model in the station consisted of a total of 14 nodes including 9 nodes for current state and 5 nodes for actions, and output nodes composed of 2 nodes; in the case of DQN, there are 9 input nodes and 5 output nodes. The ANN required for the predictive model and DQN was constructed using Python and Keras packages and then trained [29]. Table 2 shows the hyperparameter settings of the predictive model and DQN. Based on the algorithm presented in Algorithm 1, the prediction model and DQN algorithm were applied using Tensorflow and Keras based on Python.

Table 2. Hyperparameters of predictive model and DQN.

Model	Hyperparameter	Value
Prediction ANN	Number of nodes in hidden layers	(32, 16)
	Mini-batch size	64
	Number of iterations	300
	Loss function	MSE
	Optimizer	ADAM
DQN	Number of nodes in hidden layers	(256, 128)
	Activation function	ReLU
	Learning rate	0.001
	Optimizer	ADAM
	Mini-batch size	64
	Number of iterations	5000

Learning was conducted using data for a month, which updated every 15 min ($T = 2880$). In order to compare the results according to the change in the ρ value, which represents the ratio between the compensation due to the reduction in fine dust concentration and the total power cost, the results were compared by setting the ρ value to 1 in Case 1 and the ρ value to 5 in Case 2. In Case 3, we have adopted conventional control methods based on the thresholds of fine dust concentrations; if PM 2.5 or PM 10 concentrations exceed $24 \mu\text{g}/\text{m}^3$, the agent will turn on three blowers. When the concentrations exceed $48 \mu\text{g}/\text{m}^3$, the agent will additionally operate two air conditioners.

The learning results conducted using the ANN are as follows: Figures 3 and 4 show the changes in the logarithmic value of the total reward and loss function during the learning process for each case. As can be seen from the two figures, as the learning progresses, the value of the loss function converges to 0 and it can be seen that the total reward increases. This means that as we update the parameters of DQN so that the value of the loss function decreases, the predicted value of the Q function of DQN becomes more accurate. Then, for each state, the value of the total reward also increases because it follows the policy that determines the action to maximize the Q value. This means that it converges to the optimal policy through RL using the DQN method.

**Figure 3.** Changes in the total reward value during the learning process.

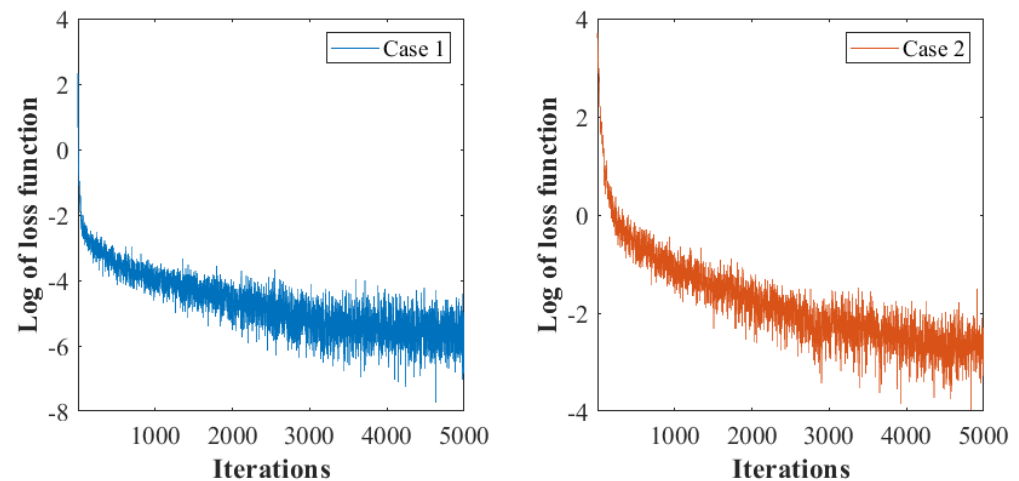


Figure 4. Change of log value of loss function during training process.

Subsequently, based on the learned ANN model, a simulation was conducted based on data for 3 days. As a result, the changes in the concentrations of PM 2.5 and PM 10 in the station for Cases 1, 2 and 3 were shown in Figures 5 and 6. As seen in both figures, Case 1 and 2 with proposed method achieve better performance on decreasing both PM 2.5 and PM 10 concentration compared to Case 3. In addition, it can be seen that Case 2 maintains a lower concentration of fine dust in the station compared to Case 1, because by setting the ρ value of Case 2 larger, the reward for the reduction of fine dust is made larger. This can be confirmed by the fact that the power consumption of Case 2 is larger than that of Case 1 in Figure 7, which shows the power consumption according to the control of the blower and air conditioner. This is because, as the ρ value of Case 2 is set to be larger, the blower and air conditioner are controlled in a direction to further reduce fine dust by increasing the power consumption. Moreover, we can see in Figure 7 that Case 3 has smaller power consumption than Case 1 but has larger consumption than Case 2. It implies that the control policy of Case 3 cannot use energy facilities efficiently, whereas Case 1 and 2 construct a model-free policy with a data-driven method that gives better performance with less power consumptions.

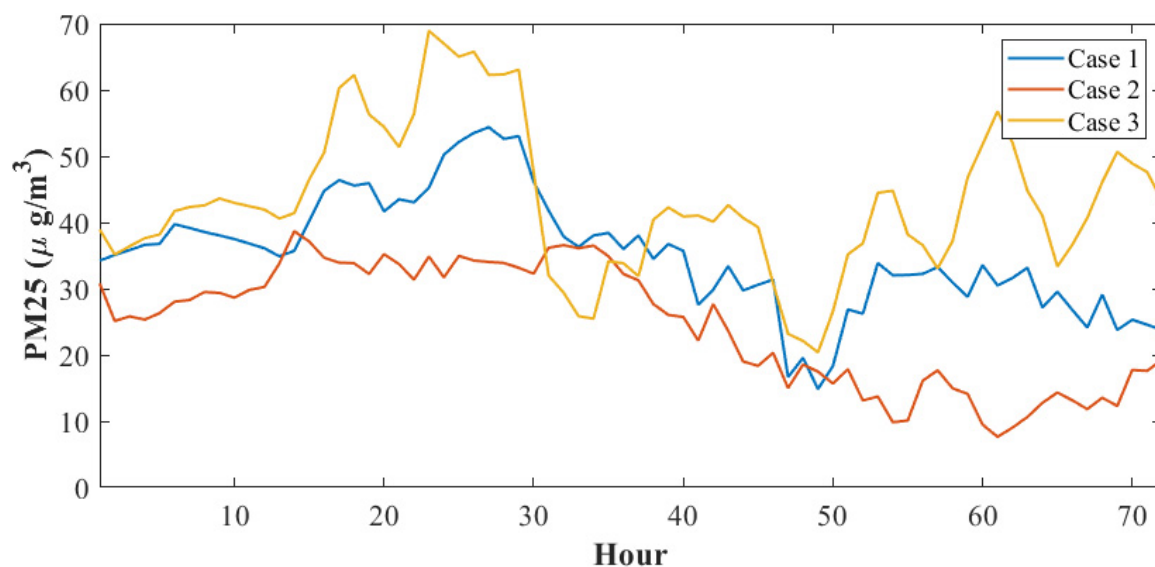


Figure 5. Changes in PM 2.5 concentration in station according to blower and air conditioner control.

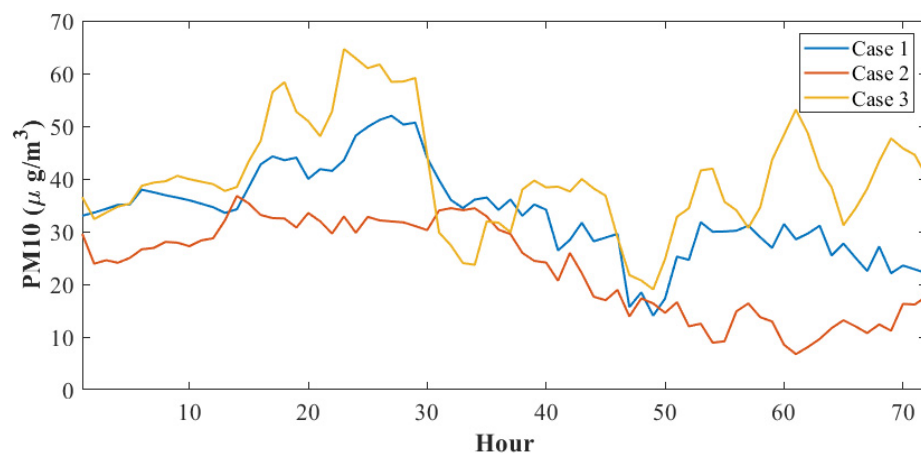


Figure 6. Changes in PM 10 concentration in station according to blower and air conditioner control.

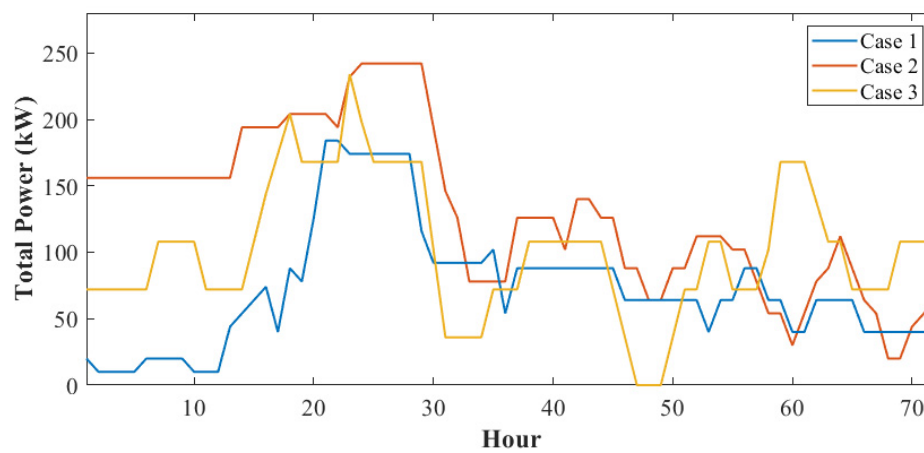


Figure 7. Changes in power consumption according to blower and air conditioner control.

To observe the result with more detail, Table 3 shows the average, minimum and maximum value of PM 2.5, PM 10 concentration levels and total power consumption of three cases. We can see that Case 2 has a least fine dust concentrations on minimum, maximum and average values, even though it has more power consumption compared to other cases. When comparing Case 1 and Case 3, Case 1 has smaller fine dust concentrations, even though it uses less average power than Case 3, which proves the effectiveness of the proposed method.

Table 3. Minimum, maximum and average values of fine dust concentration and total power consumption for three cases.

Category	Value	Case 1	Case 2	Case 3
PM 2.5 concentration	Minimum	14.85	7.64	20.38
	Maximum	54.37	38.67	68.91
	Average	35.25	24.56	43.27
PM 10 concentration	Minimum	14.06	6.76	19.03
	Maximum	51.97	36.77	64.65
	Average	33.49	23.08	40.44
Total power consumption	Minimum	10	20	0
	Maximum	184	242	234
	Average	72	130	102

6. Conclusions

In this paper, we developed a RL-based energy management agent to control PM 2.5 and PM 10 concentrations in stations using supervised learning of ANN and DQN algorithm. To this end, a Markov decision-making model was constructed in which the concentration of fine dust in the station and the time, temperature, and humidity that change it were set as the current state, and the control of the blower and air conditioner as an action. In order to predict the change in the concentration of fine dust in the station according to the control of the blower and air conditioner, an artificial neural network based on supervised learning was constructed and learned and used as a transfer kernel. Then, after constructing an artificial neural network based on the DQN algorithm to control the blower and air conditioner according to the current state, we developed an agent that controls the blower and air conditioner according to the optimal policy according to the current state.

In the case study, using actual data measured at Nam-Gwangju Station, the agent showed better performance by reducing the fine dust concentration while using the power efficiently than the conventional method. In addition, as the ratio between compensation for fine dust reduction and total electricity cost increases, the power consumption of blowers and air conditioners increases to further reduce the fine dust concentration in the station. It implies that we can adjust the level of operations by setting the value for fine dust concentration reduction. We believe the contribution of this paper leads to one more step toward the sustainable power system, with the development of new control techniques for the air-quality control facilities that efficiently manage the fine dust concentration in the station while minimizing the power consumed from the facilities. Existing future research directions open up on adding an energy storage for reducing more on operation cost and improving the performance by training multiple DQNs on each time period. As a way of improving the performance of RL, state-of-the-art algorithms such as Aquila Optimizer (AO) [30], Smell Agent Optimization (SAO) [31], African Vultures Optimization Algorithm (AVOA) [32] and Chameleon Swarm Algorithm (CSA) [33] can be applied.

Author Contributions: Methodology, J.-H.H.; Software, S.-M.H.; Writing—original draft, K.-B.K. and J.-y.P.; Project administration, H.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by a grant from R&D Program (Virtualization-based railway station smart energy management and performance evaluation technology development, PK2203E1) of the Korea Railroad Research Institute.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Yeo, M.J.; Kim, Y.P. Trends of the PM10 Concentrations and High PM10 Concentration Cases in Korea. *J. Korean Soc. Atmos. Environ.* **2019**, *35*, 249–264. [\[CrossRef\]](#)
2. Back, J.-M.; Yee, S.-W.; Lee, B.-H.; Kang, D.-H.; Yeo, M.-S.; Kim, K.-W. A Study on the Relationship between the Indoor and Outdoor Particulate Matter Concentration by Infiltration in the Winter. *J. Arch. Inst. Korea Plan. Des.* **2015**, *31*, 137–144. [\[CrossRef\]](#)
3. Querol, X.; Moreno, T.; Karanasiou, A.; Reche, C.; Alastuey, A.; Viana, M.; Font, O.; Gil, J.; de Miguel, E.; Capdevila, M. Variability of levels and composition of PM10 and PM2.5 in the Barcelona metro system. *Atmos. Meas. Tech.* **2012**, *12*, 5055–5076. [\[CrossRef\]](#)
4. Moreno, T.; Pérez, N.; Reche, C.; Martins, V.; de Miguel, E.; Capdevila, M.; Centelles, S.; Minguillón, M.; Amato, F.; Alastuey, A.; et al. Subway platform air quality: Assessing the influences of tunnel ventilation, train piston effect and station design. *Atmos. Environ.* **2014**, *92*, 461–468. [\[CrossRef\]](#)
5. Lim, H.; Yin, T.; Kwon, Y. A Study on the Optimization of the Particulate Matter Reduction Device in Underground Subway Station. In Proceedings of the Spring Conference of the Korean Institute of Industrial Engineers, Gwangju, Republic of Korea, 10 April 2019; p. 3786.

6. Park, S.; Lee, Y.; Yoon, Y.; Oh, M.; Kim, M.; Kwon, S. Prediction of Particulate Matter (PM) using Machine Learning. In Proceedings of the Korea Society for Railway Conference, Jeju, Republic of Korea, 3 May 2018; pp. 499–500.
7. Kim, Y.; Kim, B.; Ahn, S. Application of spatiotemporal transformer model to improve prediction performance of particulate matter concentration. *J. Intell. Inform. Syst.* **2022**, *28*, 329–352.
8. Kim, J.; Lee, K.; Bae, J. Construction of real-time Measurement and Device of reducing fine dust in Urban Railway. In Proceedings of the Korea Society for Railway Conference, Online, 7 July 2020; pp. 101–102.
9. Lee, Y.; Kim, Y.; Lee, H.; Kim, Y.J.; Kim, B.H. Analysis of the Correlation between the Concentration of PM 2.5 in the Out-side Atmosphere and the Concentration of PM 2.5 in the Subway Station. *J. Korean Soc. Atmos.* **2022**, *38*, 1–12. [\[CrossRef\]](#)
10. Sutton, R.S.; Barto, A.G. *Reinforcement Learning: An Introduction*, 2nd ed.; The MIT Press: Cambridge, MA, USA, 2017.
11. Kim, M.S. Research & Trends for Converged AI Technology based on Unsupervised Reinforcement Learning. *J. Korean Soc. Comp. Inform.* **2020**, *28*.
12. Michalski, R.S.; Carbonell, J.G.; Mitchell, T.M. *Machine Learning: An Artificial Intelligence Approach*, 1983rd ed.; Springer: Berlin/Heidelberg, Germany, 2013.
13. Bozdağ, A.; Dokuz, Y.; Gökçek, O.B. Spatial prediction of PM10 concentration using machine learning algorithms in Ankara, Turkey. *Environ. Pollut.* **2020**, *263*, 114635. [\[CrossRef\]](#)
14. Xu, Y.; Ho, H.C.; Wong, M.S.; Deng, C.; Shi, Y.; Chan, T.-C.; Knudby, A. Evaluation of machine learning techniques with multiple remote sensing datasets in estimating monthly concentrations of ground-level PM2.5. *Environ. Pollut.* **2018**, *242*, 1417–1426. [\[CrossRef\]](#)
15. Wei, W.; Ramalho, O.; Malingre, L.; Sivanantham, S.; Little, J.C.; Mandin, C. Machine learning and statistical models for pre-dicting indoor air quality. *Indoor Air* **2019**, *29*, 704–726. [\[CrossRef\]](#)
16. Karimian, H.; Li, Q.; Wu, C.; Qi, Y.; Mo, Y.; Chen, G.; Zhang, X.; Sachdeva, S. Evaluation of different machine learning ap-proaches to forecasting PM2.5 mass concentrations. *Aerosol Air Qual. Res.* **2019**, *19*, 1400–1410. [\[CrossRef\]](#)
17. Taheri, S.; Razban, A. Learning-based CO2 concentration prediction: Application to indoor air quality control using de-mand-controlled ventilation. *Build. Environ.* **2021**, *205*, 108164. [\[CrossRef\]](#)
18. Kang, G.K.; Gao, J.Z.; Chiao, S.; Lu, S.; Xie, G. Air Quality Prediction: Big Data and Machine Learning Approaches. *Int. J. Environ. Sci. Dev.* **2018**, *9*, 8–16. [\[CrossRef\]](#)
19. Janarthanan, R.; Partheeban, P.; Somasundaram, K.; Elamparithi, P.N. A deep learning approach for prediction of air quality index in a metropolitan city. *Sustain. Cities Soc.* **2021**, *67*, 102720. [\[CrossRef\]](#)
20. Du, S.; Li, T.; Yang, Y.; Horng, S.J. Deep Air Quality Forecasting Using Hybrid Deep Learning Framework. *arXiv* **2018**, preprint. arXiv:1812.04783. [\[CrossRef\]](#)
21. Kwon, K.-B.; Hong, S.; Heo, J.-H.; Jung, H.; Park, J.-Y. Reinforcement Learning-based HVAC Control Agent for Optimal Control of Particulate Matter in Railway Stations. *Trans. Korean Inst. Electr. Eng.* **2021**, *70*, 1594–1600. [\[CrossRef\]](#)
22. Norris, J.R. *Markov Chains*; Cambridge University Press: Cambridge, UK, 1997.
23. Minsky, M.; Papert, S.A. *Perceptrons: An Introduction to Computational Geometry*; MIT Press: Cambridge, MA, USA, 1987.
24. Bishop, C.M. *Neural Networks for Pattern Recognition*; Clarendon: Oxford, UK, 1995.
25. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Graves, A.; Antonoglou, I.; Wierstra, D.; Riedmiller, M. Playing Atari with Deep Reinforcement Learning. *arXiv* **2013**, arXiv:1312.5602.
26. Recht, B. A Tour of Reinforcement Learning: The View from Continuous Control. *Annu. Rev. Control. Robot. Auton. Syst.* **2019**, *2*, 253–279. [\[CrossRef\]](#)
27. Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A.A.; Veness, J.; Bellemare, M.G.; Graves, A.; Riedmiller, M.; Fidjeland, A.K.; Ostrovski, G.; et al. Human-level control through deep reinforcement learning. *Nature* **2015**, *518*, 529–533. [\[CrossRef\]](#)
28. Lin, L.-J. Self-improving reactive agents based on reinforcement learning, planning and teaching. *Mach. Learn.* **1992**, *8*, 293–321. [\[CrossRef\]](#)
29. Keras. Available online: <https://github.com/fchollet/keras> (accessed on 27 August 2021).
30. Abualigah, L.; Yousri, D.; Elaziz, M.A.; Ewees, A.A.; Al-qaness, M.A.; Gandomi, A.H. Aquila Optimizer: A novel me-ta-heuristic optimization Algorithm. *J. Comput. Ind. Eng.* **2021**, *157*, 107250. [\[CrossRef\]](#)
31. Salawudeen, A.T.; Mu’azu, M.B.; Sha’aban, Y.A.; Adedokun, A.E. A Novel Smell Agent Optimization (SAO): An extensive CEC study and engineering application. *Knowl. Based Syst.* **2021**, *232*, 107486. [\[CrossRef\]](#)
32. Abdollahzadeh, B.; Gharehchopogh, F.S.; Mirjalili, S. African vultures optimization algorithm: A new nature-inspired metaheuristic algorithm for global optimization problems. *Comput. Ind. Eng.* **2021**, *158*, 107408. [\[CrossRef\]](#)
33. Braik, M.S. Chameleon Swarm Algorithm: A bio-inspired optimizer for solving engineering design problems. *Expert Syst. Appl.* **2021**, *174*, 114685. [\[CrossRef\]](#)