

Article

Prediction and Control of Hovercraft Cushion Pressure Based on Deep Reinforcement Learning

Hua Zhou , Lijing Dong * and Yuanhui Wang

College of Intelligent Systems Science and Engineering, Harbin Engineering University, Harbin 150001, China; huazhou@hrbeu.edu.cn (H.Z.); wangyuanhui@hrbeu.edu.cn (Y.W.)

* Correspondence: donglijing0520@163.com; Tel.: +86-188-4509-8912

Abstract

This paper proposes a deep reinforcement learning-based predictive control scheme to address cushion pressure prediction and stabilization in hovercraft systems subject to modeling complexity, dynamic instability, and system delay. Notably, this work introduces a long short-term memory (LSTM) network with a temporal sliding window specifically designed for hovercraft cushion pressure forecasting. The model accurately captures the dynamic coupling between fan speed and chamber pressure while explicitly incorporating inherent control lag during airflow transmission. Furthermore, a novel adaptive behavior cloning mechanism is embedded into the twin delayed deep deterministic policy gradient with behavior cloning (TD3-BC) framework, which dynamically balances reinforcement learning (RL) objectives and historical policy constraints through an auto-adjusted weighting coefficient. This design effectively mitigates distribution shift and policy degradation in offline reinforcement learning, ensuring both training stability and performance beyond the behavior policy. By integrating the LSTM prediction model with the adaptive TD3-BC algorithm, a fully data-driven control architecture is established. Finally, simulation results demonstrate that the proposed method achieves high accuracy in cushion pressure tracking, significantly improves motion stability, and extends the operational lifespan of lift fans by reducing rotational speed fluctuations.

Keywords: hovercraft; LSTM; reinforcement learning; TD3-BC; cushion pressure control



Academic Editor: José Miguel Rodrigues

Received: 17 September 2025

Revised: 19 October 2025

Accepted: 23 October 2025

Published: 28 October 2025

Citation: Zhou, H.; Dong, L.; Wang, Y. Prediction and Control of Hovercraft Cushion Pressure Based on Deep Reinforcement Learning. *J. Mar. Sci. Eng.* **2025**, *13*, 2058. <https://doi.org/10.3390/jmse13112058>

Copyright: © 2025 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

During the navigation of a fully skirted hovercraft, the stability of the cushion pressure is crucial for safe operation. However, due to the unique and complex structure of the skirts, modeling and controlling the cushion system poses significant challenges [1,2]. With the rapid advancement of artificial intelligence and the continuous improvement in computational capabilities, researchers are no longer limited to spending extensive time and effort on modeling and controlling such complex systems solely through traditional methods. Instead, deep learning-based modeling [3–5] and reinforcement learning-based control [6–8] have emerged as increasingly popular research directions. Therefore, exploring how to apply deep reinforcement learning to model and control the complex cushion system of a fully skirted hovercraft constitutes a key focus and challenge of this study.

Numerous studies have been conducted on cushion pressure control for fully skirted hovercrafts [2,9,10]. An adaptive fuzzy sliding-mode control method to regulate the cushion pressure of hovercrafts by controlling the lift fans was proposed in [9]. An adaptive neural network to adjust the parameters of the PID controller was introduced in [2], thereby

enhancing the stability control of the cushion pressure. Similarly, reference [10] utilized the DDPG reinforcement learning algorithm to tune the parameters of a PID controller for autonomous cushion height control. However, the aforementioned methods all simplify the complex hovercraft cushion system into a second-order system for controller design, thereby not only neglecting the inherent characteristics of the hovercraft cushion dynamics but also yielding control strategies with limited practical relevance for real-world cushion pressure regulation.

Traditional Reinforcement Learning (RL) relies on interaction with the real environment to make action decisions, and it has achieved preliminary applications in certain low-cost domains such as small unmanned aerial vehicles (UAVs) [11,12] and unmanned ground vehicles (UGVs) [13,14]. A multi-agent reinforcement learning method based on both extrinsic and intrinsic rewards was proposed in [12] to collaboratively control the behavior of UAVs for target encircling tasks. To address the challenge of capturing escaping targets with UGVs, Su et al. [13] decomposed the reward function into individual and cooperative components, optimizing both global and local incentives. By enhancing cooperation among the pursuing UGVs, this approach significantly improves the capture success rate. However, in applications involving large-scale equipment or specific domains where the cost of trial-and-error is prohibitively high, its practical implementation remains challenging [15]. In contrast, offline reinforcement learning eliminates the need for real-time interaction with the environment. Instead, it leverages pre-collected datasets—generated in real or simulated environments—to learn policies and make decisions, thereby mitigating safety risks and reducing training costs associated with online interaction. As a result, offline reinforcement learning has emerged as a growing research direction and has attracted broad attention from researchers and experts across disciplines [16,17].

Deep Learning (DL) models are deep neural network architectures characterized by multiple nonlinear mapping layers, which are capable of extracting features from input signals and uncovering underlying patterns [18]. The Long Short-Term Memory (LSTM) network, an enhanced variant of Recurrent Neural Networks (RNNs), has been widely adopted in numerous time series studies, including applications such as speech recognition [19], multimedia audio and video analysis [20], and road traffic flow prediction [21]. In hovercraft systems, the relationship between fan speed and chamber pressure is highly nonlinear, involving complex physical processes such as fluid dynamics and aerodynamics [2]. However, in the field of hovercraft lift system research, particularly in the use of LSTM for predicting cushion pressure, no relevant studies have been reported to date.

Based on the above analysis, this paper addresses the challenges of modeling the lift system, cushion pressure instability, and control delays in hovercraft by proposing a deep reinforcement learning-based predictive control method for cushion pressure. The main contributions of this work are summarized as follows:

- (1) An LSTM-based predictor with a temporal sliding window is proposed for hovercraft cushion pressure forecasting, effectively capturing the dynamic coupling between fan speed and chamber pressure while explicitly incorporating inherent control lag.
- (2) A novel adaptive behavior cloning mechanism is integrated into the TD3-BC framework, which dynamically balances the RL objective and historical policy constraints via an auto-adjusted weight, thereby mitigating distribution shift and policy degradation in offline settings.
- (3) A fully data-driven control architecture is established by combining the LSTM predictor with the adaptive TD3-BC algorithm, abbreviated as LSTM-TD-BC, enabling accurate cushion pressure tracking, improved motion stability, and extended lift fan operational life through reduced rotational speed fluctuations.

The remainder of this paper is structured as follows: Section 2 introduces the mathematical model of the hovercraft lift system and related preliminary knowledge. Section 3 details the proposed LSTM-TD3-BC-based cushion pressure prediction and control framework. Section 4 presents simulation results and comparative evaluations. Finally, Section 5 concludes the paper and suggests future research directions.

2. Preliminaries and Problem Formulation

2.1. Model Description

The model of the fully skirted hovercraft's cushion system comprises mathematical representations of the flow and pressure dynamics of the lift fan, the flow and pressure transfer from the booster fan to each individual chamber, and the skirt discharge characteristics and nonlinear flow-pressure balance equations, along with their corresponding solutions [1]. Owing to the structural complexity and modeling challenges associated with the cushion system of a fully skirted hovercraft, the gas within each chamber and the connecting pipelines are assumed to be incompressible. The cushion is modeled as a rectangular structure subdivided in a cross-shaped pattern. A schematic diagram of the overall cushion system layout of the fully skirted hovercraft is provided in Figure 1:

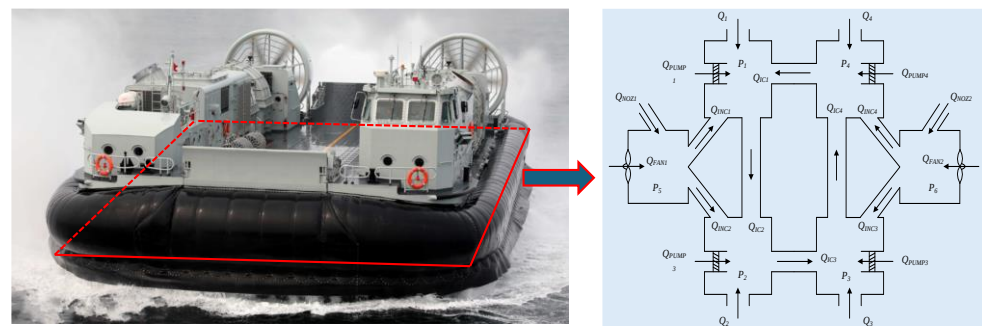


Figure 1. Schematic diagram of flow and pressure distribution in lifting system.

where $P_i (i = 1, 2, 3, 4)$ is the pressure of the four chambers, P_5 and P_6 are the pressure at the left and right side booster fans, Q_{FAN1} and Q_{FAN2} are the flow rates at the left and right side booster fans, $Q_{INCI} (i = 1, 2, 3, 4)$ is the flow rate from the booster fan to each chamber, $Q_{NOZI} (i = 1, 2)$ is the inflow flow rate of the bow nozzle, $Q_{PUMPI} (i = 1, 2, 3, 4)$ is the volume change rate of the air cushion, which is related to the navigation posture and surface characteristics of the hovercraft, $Q_{ICI} (i = 1, 2, 3, 4)$ is the flow rate between each chamber, and $Q_i (i = 1, 2, 3, 4)$ is the discharge flow rate at the bottom of the apron.

The cushion fans are positioned symmetrically on both sides of the fully skirted hovercraft, with one unit on each side. These fans operate independently yet coordinate to ensure stable motion of the vehicle. The relationship between the flow rate and pressure of the investigated booster fan can be mathematically expressed as follows:

$$\begin{cases} Q_{FAN1} = \left\{ -1280\sqrt{|P_5 - P_{50}|} \times \text{sign}(P_5 - P_{50}) - 31.6(P_5 - P_{50}) \right\} \times N_l / N_f \\ Q_{FAN2} = \left\{ -1280\sqrt{|P_6 - P_{60}|} \times \text{sign}(P_6 - P_{60}) - 31.6(P_6 - P_{60}) \right\} \times N_r / N_f \end{cases} \quad (1)$$

where N_l and N_r represent the rotational speeds of the left and right fans, N_f represents the rated speed of the fan, and P_{50} and P_{60} represent the static pressure at the raised fan.

2.2. Calculation of Air Cushion Volume Change Rate

During the navigation of a fully cushioned hovercraft, the air cushion pressure and the discharge area interact dynamically. The discharge area is determined by integrating

the leakage height along the entire circumference of the bottom apron. To address practical engineering requirements, 35 characteristic points are selected along the bottom periphery of the air cushion to serve as the basis for calculating the discharge area. Figure 2 illustrates the planar structure of the air cushion and the distribution of these characteristic points within the vessel's coordinate system.

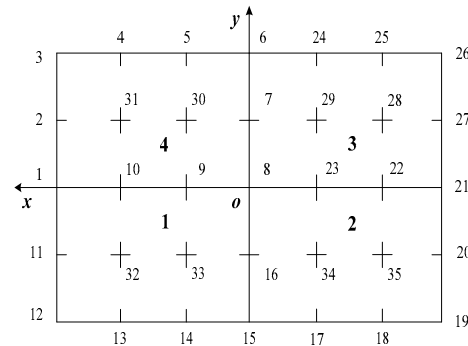


Figure 2. Distribution diagram of air cushion plane structure and bottom characteristic points of hovercraft.

For each chamber, the apron clearance on the X-axis and Y-axis is integrated on the ship's coordinate system to obtain the apron discharge area, which is expressed as follows:

$$S_i = S_{xi} + S_{yi} \quad (i = 1, 2, 3, 4) \quad (2)$$

where $S_x = \int_0^{3a} H_x(x)dx$, $S_y = \int_0^{2b} H_y(y)dy$, $H_x(x)$ and $H_y(y)$ represent the discharge height functions of the flexible apron of the hovercraft on the X and Y axes, respectively. a and b are the cubic fitting coefficients, which are fitted in the X and Y directions as follows:

$$\begin{cases} S_x = 6.398 \times (H_x(0) + H_x(a) + H_x(2a) + H_x(3a)) \\ S_y = 4.3283 \times (H_y(0) + H_y(a) + H_y(2a)) \end{cases} \quad (3)$$

Changes in the hovercraft's attitude or wave-induced motions alter the air volume within the cushion chamber, significantly affecting the air cushion pressure. The rate of change in air volume, in turn, directly influences the mass flow rate throughout the lift system. As illustrated in Figure 2, the air cushion volume is computed based on the distances TH_n ($n = 1, 2, \dots, 35$) between 35 feature points and the navigation surface. The positions of these feature points in the geodetic coordinate system are given by the following:

$$DH_n = -\theta x_n + \varphi y_n + z_n \quad (4)$$

where (x_n, y_n, z_n) is the vertical position of the feature point in the hull coordinate system, θ and φ are the longitudinal and transverse inclinations of the hovercraft, respectively. The vertical position of the feature points corresponding to the navigation surface in the geodetic coordinate system is EH_n ; therefore, the distance between the feature points and the navigation surface can be expressed as follows:

$$TH_n = DH_n - EH_n \quad (5)$$

Then, using the fitting function $OH(TH_n, n = 1, 2, \dots, 35)$, we can calculate the air cushion volume at a certain chamber (taking air cushion 1 as an example) as follows:

$$V = OH_1 \times S_{c1} \quad (6)$$

where OH_1 is the average height between the bottom of air cushion 1 and the sailing surface, given by the following equation:

$$OH_1 = C_{c1}(TH_1 + TH_8 + TH_{12} + TH_{15}) + C_{c2}(TH_{13} + TH_{14} + TH_9 + TH_{10}) + C_{c3}(TH_{11} + TH_{16} + 0.25TH_{33} + TH_{32}) \quad (7)$$

where C_{c1} , C_{c2} , and C_{c3} are approximately obtained from experiments. If S_{c1} is the horizontal projection area of air cushion 1, then the volume of air cushion 1 at a certain moment can be expressed as follows:

$$V_1 = OH_1 \times S_{c1} \quad (8)$$

If Δt is the sampling time interval and $V(old)$ is the volume of the previous time, then the rate of change in the volume of each air cushion can be obtained as follows:

$$Q_{PUMPi} = (V_i - V_i(old)) / \Delta t \quad (i = 1, 2, 3, 4) \quad (9)$$

Within the booster system, the main duct of the booster fan and the four chambers are collectively described by a set of six nonlinear continuous equations that govern the system dynamics at any given time. These equations can be expressed as follows:

$$\begin{cases} -Q_{INC1} - Q_{INC2} + Q_{NOZ1} + Q_{FAN1} = 0 \\ -Q_{INC3} - Q_{INC4} + Q_{NOZ2} + Q_{FAN2} = 0 \\ Q_{INC1} + Q_{IC1} - Q_{IC2} + Q_{PUMP1} + Q_1 = 0 \\ Q_{INC2} + Q_{IC2} - Q_{IC3} + Q_{PUMP2} + Q_2 = 0 \\ Q_{INC3} + Q_{IC3} - Q_{IC4} + Q_{PUMP3} + Q_3 = 0 \\ Q_{INC4} + Q_{IC4} - Q_{IC1} + Q_{PUMP4} + Q_4 = 0 \end{cases} \quad (10)$$

According to Bernoulli's principle, an increase in fluid velocity corresponds to a decrease in fluid pressure [22]. Based on this relationship, the flow rate in Equation (10) can be expressed in terms of the pressure in each chamber and the corresponding flow cross-sectional area as follows:

$$\begin{cases} Q_{INC1} = 589\sqrt{|P_5 - P_1|}\text{sgn}(P_5 - P_1) \\ Q_{INC2} = 589\sqrt{|P_5 - P_2|}\text{sgn}(P_5 - P_2) \\ Q_{INC3} = 589\sqrt{|P_6 - P_3|}\text{sgn}(P_6 - P_3) \\ Q_{INC4} = 589\sqrt{|P_6 - P_4|}\text{sgn}(P_6 - P_4) \end{cases} \quad (11)$$

$$\begin{cases} Q_{NOZ1} = -346\sqrt{|P_5|}\text{sgn}(P_5) \\ Q_{NOZ2} = -346\sqrt{|P_6|}\text{sgn}(P_6) \end{cases} \quad (12)$$

$$\begin{cases} Q_{IC1} = 675\sqrt{|P_4 - P_1|}\text{sgn}(P_4 - P_1) \\ Q_{IC2} = 338\sqrt{|P_1 - P_2|}\text{sgn}(P_1 - P_2) \\ Q_{IC3} = 675\sqrt{|P_2 - P_3|}\text{sgn}(P_2 - P_3) \\ Q_{IC4} = 338\sqrt{|P_3 - P_4|}\text{sgn}(P_3 - P_4) \end{cases} \quad (13)$$

$$\begin{cases} Q_1 = -S_1 \times 14.5\sqrt{|P_1|}\text{sgn}(P_1) \\ Q_2 = -S_2 \times 14.5\sqrt{|P_2|}\text{sgn}(P_2) \\ Q_3 = -S_3 \times 14.5\sqrt{|P_3|}\text{sgn}(P_3) \\ Q_4 = -S_4 \times 14.5\sqrt{|P_4|}\text{sgn}(P_4) \end{cases} \quad (14)$$

We substitute the above flow calculation expressions into Equation (10), and use the apron discharge area S_i ($i = 1, 2, 3, 4$) and chamber volume change rate Q_{PUMPi} ($i = 1, 2, 3, 4$), N_l , and N_r as action inputs. The system of equations contains six unknown pressures, which are solved using Newton's iterative method.

2.3. Markov Decision Process Modeling

The Markov Decision Process (MDP) is widely used to model reinforcement learning tasks. When an intelligent agent performs a task, it begins by interacting with the environment. Starting from a given initial state, a new state is generated according to the state transition probability. Simultaneously, the environment returns a reward signal, based on which the agent updates its action policy. This process repeats iteratively. Through multiple such interactions, the agent gradually learns an optimal policy to accomplish the designated task [23].

First, the air cushion system of the hovercraft is characterized by the following nonlinear state space model:

$$\begin{cases} \dot{X}(t) = F(X(t), U(t)) + \zeta(t) \\ Y(t) = C * X(t) \end{cases} \quad (15)$$

where $X(t) = [P_1, P_2, P_3, P_4, N_l, N_r]^T$ is the system state variable vector, and $U(t) = [\Delta N_l, \Delta N_r]^T$ is the action input, representing the incremental speed of the fan. $Y(t) = [P_1, P_2, P_3, P_4]^T$ is the output vector, and $\zeta(t)$ represents system noise and external disturbances. C is the output matrix.

Then, transform the control problem into MDP as follows:

$$M = (S, A, P, R, \gamma) \quad (16)$$

where S represents the state space, consisting of four chamber pressures and two fan speeds. A represents the action space, consisting of two incremental fan speeds. P represents the probability of state transition, approximated by the LSTM model. γ is the discount factor, which is a real number between 0 and 1. R is the reward function, and the goal of pressure prediction control in the air chamber of an air cushion vehicle is to control the pressure amplitude of the air chamber to change within a small range and with a small frequency under the action of two independent lift fans. Therefore, the reward function can be designed as follows:

$$R_t = - \left(\sum_{i=1}^4 w_i |P_i^{(t+1)} - P_{target}| + \lambda_a \sum_{j=1}^2 |U_j^{(t)}| + \lambda_b \sum_{k=1}^4 |\Delta P_k^{(t)}| \right) \quad (17)$$

where P_{target} is the target pressure of the air chamber, which is related to the actual weight and navigation conditions of the hovercraft. The term $|\bullet|$ represents absolute value. λ_a and λ_b are positive real numbers that, respectively, control the smoothness of the action and the penalty intensity of pressure changes.

3. Prediction and Control of Air Chamber Pressure in Air Cushion Vehicle Based on LSTM-TD3-BC

To enable real-time prediction and active control of the chamber pressure in a fully cushioned hovercraft, this section employs an improved LSTM network to forecast the chamber pressure. This approach is designed to capture the complex spatiotemporal dynamics inherent in the hovercraft's cushion system, thereby establishing a model foundation for subsequent reinforcement learning-based control of the cushion pressure. Building on the LSTM-based prediction model for the lift system, offline policy optimization is performed using the TD3-BC reinforcement learning algorithm. This enhances both the stability and robustness of the system while maintaining accurate control over the cushion lift pressure.

A schematic diagram of the hovercraft chamber pressure prediction and control framework based on the LSTM-TD3-BC algorithm is presented in Figure 3. The LSTM

Prediction Model serves as the environment for the RL agent, forecasting chamber pressures based on the history of system states. The TD3 Algorithm serves as the primary RL controller, generating control actions (fan speed increments) based on the current state and its Q-value estimates. The Behavior Cloning (BC) Framework regularizes the TD3 policy by constraining it towards actions demonstrated in the historical dataset, using an adaptive weight to mitigate distribution shift.

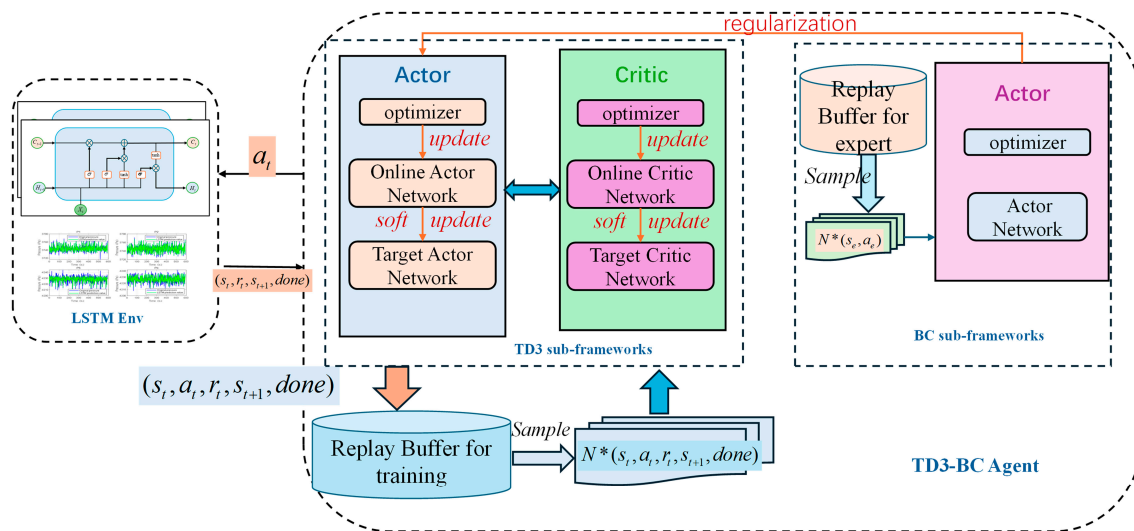


Figure 3. Principle structure diagram of LSTM-TD3-BC algorithm.

3.1. LSTM Deep Learning Model

LSTM network is an improved model of recurrent neural network (RNN). LSTM learns long-term correlated information by introducing gating mechanism to control the forgetting and flow of information, thereby alleviating the gradient explosion and vanishing problems of RNN during training. Therefore, LSTM is an ideal choice for processing time-series data [24]. The overall structure of LSTM is shown in Figure 4. The LSTM network mainly consists of four modules, namely input gate, forget gate, memory unit, and output gate, and continuously updates the states of the four modules to process the sequence data in each time step.

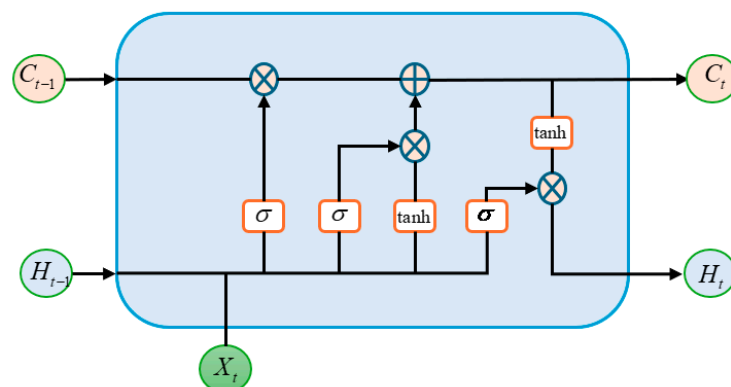


Figure 4. Schematic diagram of LSTM principle structure.

In Figure 4, X_t is the input data information at time t , H is the hidden state information, C is the memory unit state, and $\tanh$ is the hyperbolic tangent activation function. σ is a sigmoid activation function, which compresses the input value between 0 and 1 to ensure whether information is retained or forgotten.

The forget gate controls the retention and forgetting of information in the memory unit. Based on the current state and the previous hidden state, obtain an information value representing the degree of forgetting. The higher the value, the greater the probability of the corresponding information being forgotten in memory unit C_{t-1} . This mechanism allows LSTM to selectively retain long-term dependent available information, discard low value data, and effectively process complex sequence data. The calculation formula is expressed as follows:

$$F_t = \sigma(W_{xf}X_t + W_{hf}H_{t-1} + B_f) \quad (18)$$

where F_t is the output of the forget gate, W_{xf} is the weight matrix between the current input and the forget gate, W_{hf} is the weight matrix between the historical output and the forget gate, and B_f is the bias term of the forget gate.

The input gate serves to update the memory cell state by determining whether new information should be retained. It processes the previous hidden state and the current input through a sigmoid activation function, producing output values between 0 and 1, where 0 signifies insignificance and 1 indicates importance. Meanwhile, the same inputs are processed by a tanh function to generate candidate memory cell states within the range of -1 to 1 . The output of the input gate is expressed as follows:

$$I_t = \sigma(W_{xi}X_t + W_{hi}H_{t-1} + B_i) \quad (19)$$

where I_t denotes the output of the input gate, W_{xi} represents the weight matrix between the input gate and the input, W_{hi} is the weight matrix connecting the previous hidden state to the input gate, and B_i denotes the bias term of the input gate.

The candidate cell state represents the extent to which the new input at the current time step can influence the memory cell state. It is computed as follows:

$$\tilde{C}_t = \tanh(W_{xc}X_t + W_{hc}H_{t-1} + B_c) \quad (20)$$

where $\tilde{C}_t$ is the candidate cell state, W_{xc} denotes the weight matrix between the input and the candidate state, W_{hc} represents the weight matrix associated with the previous hidden state, and B_c is the bias term.

The memory cell serves as the core component of the entire LSTM network. It is capable of not only regulating the flow and update of information but also storing and transmitting data. After incorporating the information from the forget gate and the input gate, the memory cell state can be updated as follows:

$$C_t = F_t C_{t-1} + I_t \tilde{C}_t \quad (21)$$

The output gate regulates the flow of information passed to the next time step. It determines which parts of the cell state should be output Via a sigmoid activation function. The output of the gate is expressed as follows:

$$O_t = \sigma(W_{xo}X_t + W_{ho}H_{t-1} + B_o) \quad (22)$$

where W_{xo} denotes the weight matrix between the input and the output gate, W_{ho} represents the weight matrix connecting the previous hidden state to the output gate, and B_o is the bias term of the output gate. Subsequently, the hidden state H_t is obtained by multiplying the output gate with the cell state processed through a tanh activation function, as follows:

$$H_t = O_t \tanh(C_t) \quad (23)$$

3.2. TD3-BC Reinforcement Learning Algorithm

To mitigate the high interaction cost and safety risks associated with online reinforcement learning, researchers have introduced offline reinforcement learning [25]. Offline reinforcement learning trains a policy π using a pre-collected dataset. The agent executes an action a in state s according to the policy, and the expected cumulative return after following the policy π is given by the following:

$$Q_{\pi}(s, a) = E_{P, \pi|s_0=s, a_0=a} \left[\sum_{t=0}^{\infty} \gamma^t r(s_t, a_t) \right] \quad (24)$$

It represents the expected cumulative discounted return when the agent starts from state s , executes action a , and thereafter follows policy π . The expectation $E_{P, \pi|s_0=s, a_0=a}$ accounts for the inherent randomness in both the policy and the environment's state transitions.

Equation (24) is updated Via the Bellman equation as follows:

$$Q(s, a) = r(s, a) + \gamma \max_{a'} Q(s', a') \quad (25)$$

This provides a recursive decomposition of the value. It states that the Q-value for a state–action pair (s, a) can be broken down into the immediate reward $r(s, a)$ and the discounted expected value of the next state $\max_{a'} Q(s', a')$, where s' is the state resulting from taking action a in state s . This recursion is fundamental to most RL algorithms, as it enables iterative estimation and improvement of the Q-function.

Equation (25) can be solved by minimizing the mean squared Bellman error, which is defined as follows:

$$B_l = \frac{1}{2} E_{s,a,r,s'} \left[Q(s, a) - (r + \gamma \max_{a'} Q(s', a')) \right]^2 \quad (26)$$

This serves as the primary loss function for training the Critic network in value-based RL methods like TD3. By minimizing this loss, the parameters of the Q-network are adjusted so that its estimates become increasingly consistent with the Bellman Equation (25).

Essentially, the objective of offline reinforcement learning is to learn a policy that outperforms the behavior policy, thereby achieving improved performance upon deployment in real-world interactions. However, due to the issue of distribution shift, most offline RL algorithms tend to adopt conservative strategies, which often limits their ability to capture the full complexity of the policy when relying solely on behavioral cloning. Therefore, striking an effective balance between policy improvement and mitigating distribution shift constitutes a central challenge in the field of offline reinforcement learning.

TD3-BC is an offline reinforcement learning algorithm that integrates the advantages of both TD3 and BC [26]. The TD3 component employs an Actor–Critic architecture along with a replay buffer of collected experiences. It updates the policy based on the value of the Q-function, prioritizing state–action pairs with high estimated returns regardless of the current policy behavior. This flexibility enables the algorithm to leverage large, sparse rewards effectively, thereby improving the success rate. Through advanced policy exploration, it derives an optimized policy, denoted as π . The BC component consists of an expert replay buffer and an Actor network, which learns from expert demonstrations to facilitate knowledge transfer.

TD3-BC incorporates a BC regularization term to ensure that the policy optimizes the Q-function while remaining close to the actions observed in the historical dataset. This prevents the algorithm from taking overly risky actions in regions with insufficient data

coverage. Consequently, the policy optimization objective of TD3-BC is to maximize the following policy objective function:

$$L_{\pi} = E_{(s,a) \in D} \left[Q_{\pi}(s, \pi(s)) - \rho \|\pi(s) - a\|^2 \right] \quad (27)$$

where D denotes the dataset used for policy training, and ρ is the behavior regularization coefficient that controls the degree of conservatism by constraining the learned policy to remain close to the behavior policy. By default, $\|\bullet\|$ refers to the L_2 norm (Euclidean norm), which has the same meaning when used in subsequent formulas.

3.3. Design of LSTM-TD3-BC-Based Algorithm for Cushion Pressure Prediction and Control

An LSTM network is employed to model the dynamic relationship between fan speeds and chamber pressures. The input to the model is a 6-dimensional state vector X_t over a historical time window of length T , which consists of four chamber pressure values and two fan speed values, denoted as $X_t = [s_{t-T}, s_{t-T+1}, \dots, s_{t-1}]$. The output is the predicted four chamber pressure values $\hat{Y}_t$ at the subsequent time step, denoted as $\hat{Y}_t = [\hat{P}_{1t}, \hat{P}_{2t}, \hat{P}_{3t}, \hat{P}_{4t}]^T$.

To enhance the robustness of the LSTM prediction model, the Huber loss function is employed as follows:

$$L_{\delta}(Y, \hat{Y}) = \begin{cases} \frac{1}{2} \|Y - \hat{Y}\|^2 & \text{if } \|Y - \hat{Y}\| \leq \delta \\ \delta \|Y - \hat{Y}\| - \frac{1}{2} \delta^2 & \text{otherwise} \end{cases} \quad (28)$$

where δ is a positive constant.

The training process of the LSTM prediction model follows systematic data processing and deep learning optimization methodology, with key steps summarized in Algorithm 1. The procedure begins with standardized preprocessing of the raw time-series data from the air cushion vehicle lift system. The mean μ and standard deviation σ of each channel are computed to normalize the dataset into a zero-mean, unit-variance form.

Subsequently, input-output sample pairs are constructed using a temporal window slicing technique to predict the cushion pressure at the next time step. The model architecture employs a two-layer LSTM encoder, with a dropout layer (rate = 0.2) applied after the second LSTM layer to prevent overfitting. The Huber loss function is utilized to combine the advantages of both Mean Absolute Error (MAE) and Mean Square Error (MSE) loss functions, thereby enhancing the model's robustness to outliers. The Adam optimizer is adopted with a learning rate $\eta = 0.001$, and momentum coefficients $\beta_1 = 0.9$ and $\beta_2 = 0.99$. The hyperparameters in Algorithm 1 were chosen based on established practices in deep learning for time-series problems and limited empirical tuning on our validation set. For instance, the time window size $T = 20$ was selected to balance the need to capture the system's short-term dynamics with computational efficiency.

In the pressure control system of the hovercraft lift system, this paper adopts the TD3-BC algorithm as the core reinforcement learning framework. By integrating the TD3 with BC regularization, the algorithm effectively addresses the issues of extrapolation error and policy degradation in conventional offline reinforcement learning. To dynamically balance the strength of policy optimization and historical behavior imitation, and to ensure an effective trade-off between exploring novel actions and maintaining behavioral stability, this paper introduces an adaptive BC regularization method to update the objective function of the Actor network. Therefore, Equation (27) can be rewritten as follows:

$$L_{\varphi} = E_{(s,a) \sim D_r} \left[-\lambda Q_{\theta_1}(s, \pi_{\varphi}(s)) + \alpha \|\pi_{\varphi}(s) - a\|^2 \right] \quad (29)$$

where $\alpha > 0$ is BC regularization strength, and λ is the adaptive weighting coefficient defined as follows:

$$\lambda = \frac{\alpha}{E_{(s,a) \sim D_r} [\|Q(s, \pi_\varphi(s))\|] + \lambda_0} \quad (30)$$

where λ_0 is a sufficiently small constant to prevent division by zero.

Algorithm 1 Training Procedure for LSTM-Based Cushion Pressure Prediction Model

Input: Dataset $D_l = \{t, P_1, P_2, P_3, P_4, N_l, N_r\}$
Time window size $T = 20$
Hyperparameters : $\{\eta, \beta_1, \beta_2, B, E, \delta\}$
Output: Trained LSTM prediction model

- 1 data preprocessing:
- 2 $\mu, \sigma \leftarrow$ Calculate the mean and standard deviation of each channel
- 3 $D_{norm} \leftarrow (D_l - \mu) / \sigma$
- 4 $X_{train}, X_{val}, Y_{train}, Y_{val} \leftarrow$ Split training and testing sets
- 5 Initialize LSTM model M:
- 6 Input layer: $(T, 6) \leftarrow$ Time window, feature dimension
- 7 LSTM1: Unit 128
- 8 Dropout layer: 0.2
- 9 LSTM2: Unit 128
- 10 Output layer : $Dense(4) \leftarrow$ Fully connected layer, with dimensions of 4
- 11
- 12 for epoch to E do:
- 13 for batch in Batch Generator $(X_{train}, Y_{train}, B)$ do:
- 14 $\hat{Y} = M(X_{batch}) \leftarrow$ forward propagation
- 15 $L = \text{HuberLoss}(Y_{batch}, \hat{Y}; \delta) \leftarrow$ Loss calculation
- 16 $M.backward(Loss) \leftarrow$ backpropagation
- 17 $\theta \leftarrow \text{Adam}(\theta, \nabla L; \eta, \beta_1, \beta_2)$, parameters update
- 18 end for
- 19
- 20 $val_loss \leftarrow$ evaluate (M, X_{val}, Y_{val})
- 21 $best_model \leftarrow M$, Save LSTM model
- 22 end for
- 24 Return LSTM pressure prediction model M

Remark 1. The primary role of the adaptive weighting coefficient λ_0 is to dynamically balance the relative importance between the reinforcement learning objective and the behavior cloning constraint. When the policy performs poorly, indicated by unreliable and low Q-value estimates, λ_0 increases to strengthen the influence of behavioral regularization. This encourages the policy to imitate expert behaviors from the historical dataset, thereby mitigating the risk of risky actions. Conversely, when the policy performs well, λ_0 decreases to relax the behavior cloning constraint, allowing greater exploration for potentially superior strategies beyond historical behaviors.

The Critic network employs a twin Q-network architecture and is trained by minimizing the temporal difference (TD) error, which is derived from the Bellman equation:

$$L(\theta_i) = E_{(s,a,r,s') \sim D_r} \left[(Q_{\theta_i}(s, a) - y)^2 \right] \quad (31)$$

where $y = r + \gamma(1 - d)\min_{i=1,2} Q_{\theta_i}(s', \text{clip}(\pi_{\phi'}(s') + \lambda_0, -1, 1))$. The target networks are updated via a soft update mechanism to ensure training stability, thereby guaranteeing control precision in hovercraft lift pressure control task and further enhancing system robustness. The pseudo-code of the TD3-BC-based cushion pressure control algorithm is presented in Algorithm 2:

Algorithm 2 TD3-BC-Based Control for Hovercraft Cushion Pressure

Input: Historical dataset $D_r = \{S, A, R, S', done\}$
 LSTM prediction model M
 Hyperparameters : $\{\alpha, \gamma, \tau, \sigma, c, d, B\}$

Output: Optimized control strategy π_ϕ

- 1 Initialization:
- 2 Experience buffer zone $\leftarrow$ Sampling from D
- 3 Randomly initialize Actor network π_ϕ , Critic network $Q_{\theta 1}, Q_{\theta 2}$
- 4 Initialize target network : $\phi' \leftarrow \phi, \theta 1' \leftarrow \theta 1, \theta 2' \leftarrow \theta 2$
- 5
- 6 for count in maximum iterations do:
- 7 $\{(s, a, r, s', done)\} \sim B$, Batch sampling
- 8 $a' \leftarrow \pi_{\phi'}(s') + \text{clip}(N(0, \sigma), -c, c)$, Target action calculation
- 9 $Q_target \leftarrow \min(Q_{\theta 1'}(s', a'), Q_{\theta 2'}(s', a'))$, Target
- 10 Q-value calculation
- 11 Update Critic Network:
- 12 $\theta 1 \leftarrow \text{Adam}(\nabla_{\theta 1}[(Q_{\theta 1}(s, a) - y)^2])$
- 13 $\theta 2 \leftarrow \text{Adam}(\nabla_{\theta 2}[(Q_{\theta 2}(s, a) - y)^2])$
- 14 if $count/d = 0$ then:
- 15 Adaptive BC weight update:
- 16 $Q_val \leftarrow Q_{\theta 1}(s, \pi_\phi(s))$
- 17 $\lambda \leftarrow \alpha / (\text{mean}(|Q_vals|) + 1 \times 10^{-5})$
- 18 Actor loss for behavior cloning:
- 19 $L_actor \leftarrow -\lambda \text{mean}(Q_vals) + \alpha \text{MSE}(\pi_\phi(s), a)$
- 20 $\phi \leftarrow \text{Adam}(\nabla_\phi L_actor)$, Update actor
- 21 Soft update target network:
- 22 $\theta 1' \leftarrow \tau \theta 1 + (1 - \tau) \theta 1'$
- 23 $\theta 2' \leftarrow \tau \theta 2 + (1 - \tau) \theta 2'$
- 24 $\phi' \leftarrow \tau \phi + (1 - \tau) \phi'$
- 25 end if
- 26 end for
- 27 Return control strategy π_ϕ

4. Experimental Results and Discussion

To evaluate the effectiveness of the proposed algorithm, this study utilizes a fully skirted hovercraft simulation platform to generate a cushion dataset during navigation for training the LSTM prediction model. The process is further integrated with the offline reinforcement learning algorithm TD3-BC to make reinforcement learning-based control decisions for cushion pressure regulation. The experiments were conducted in the following environment: hardware setup includes an Intel Core i7-13700KF CPU and an NVIDIA RTX 4060Ti GPU; software environment consists of Python 3.9, PyTorch 1.12.1, and CUDA 11.6; the simulation platform employs PyCharm 2022 for algorithm development and Visual Studio 2022 for hovercraft simulator integration.

The experiments utilized a simulator closely mimicking a real-world hovercraft to generate the dataset, which consists of 101,922 samples. Each sample includes a 7-dimensional feature vector: $X(t) = [t, P_1, P_2, P_3, P_4, N_l, N_r]^T$. All features, except time, were normalized using min-max scaling. Input sequences were constructed by applying a sliding window method with a history of 20-time steps, and the pressure values at the next time step were used as the output. The dataset was split into training and validation sets for the LSTM model in an 8:2 ratio. Subsequently, an adaptive TD3-BC algorithm was employed, integrating offline data and the LSTM prediction model, to achieve stable cushion pressure control through reinforcement learning-based decision-making.

This study adopts a simulation-based, data-driven experimental design, which is structured into three consecutive phases: (1) data collection, (2) predictive model training, and (3) offline reinforcement learning control. The specific details are as follows:

4.1. Simulation Study on Cushion Lifting Characteristics

The cushion stability of a fully skirted hovercraft is fundamental to its safe navigation. After developing a mathematical model of the lift system, simulation was conducted to acquire pressure variation data of each air chamber under the influence of lift fan speeds and external environmental disturbances. The initial speed of the hovercraft was set to 25 knots, while the rotational speeds of the port and starboard lift fans were fixed at 2020 rpm. Partial simulation results illustrating the pressure changes in the air chambers of the cushion system are shown in Figures 5 and 6.

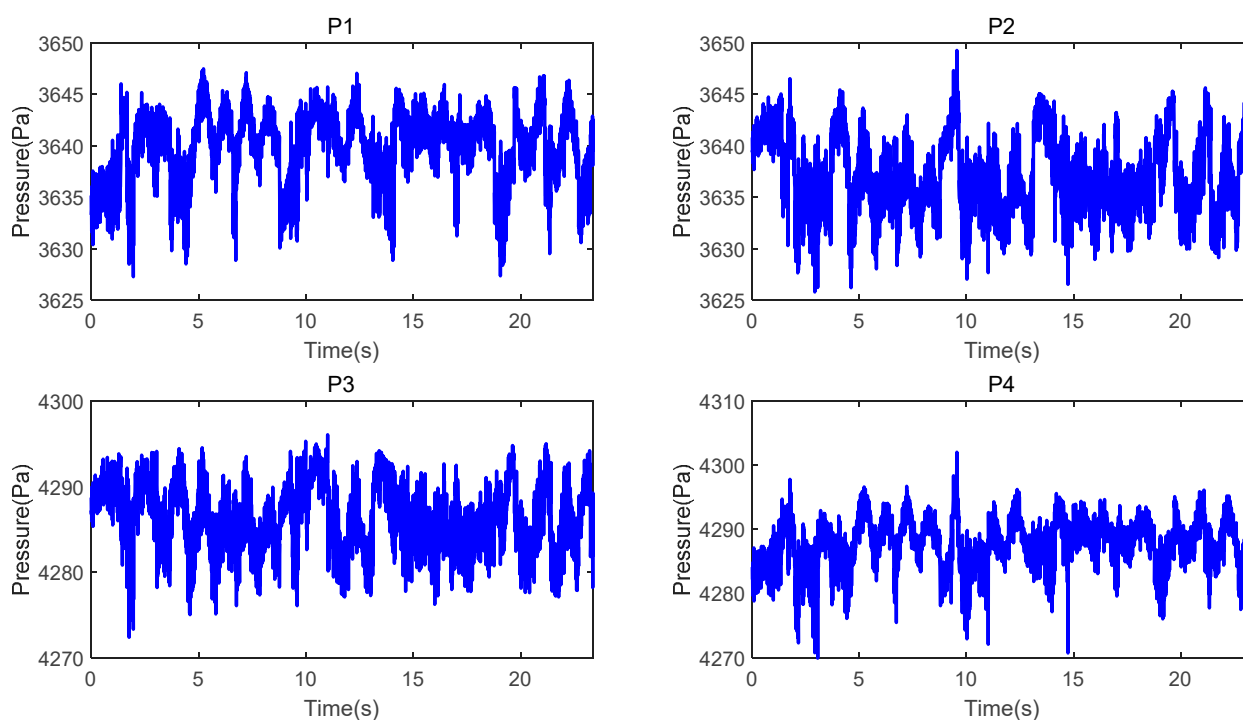


Figure 5. Pressure changes in the four chambers of the air cushion system of a hovercraft.

As can be observed from Figures 5 and 6, under constant lift fan speeds, the hovercraft exhibits significant pressure fluctuations and vibrations due to wave-induced disturbances and skirt air leakage. These effects result in considerable vertical oscillations during navigation, which not only impair crew operational safety but also compromise the hovercraft's overall seakeeping performance.

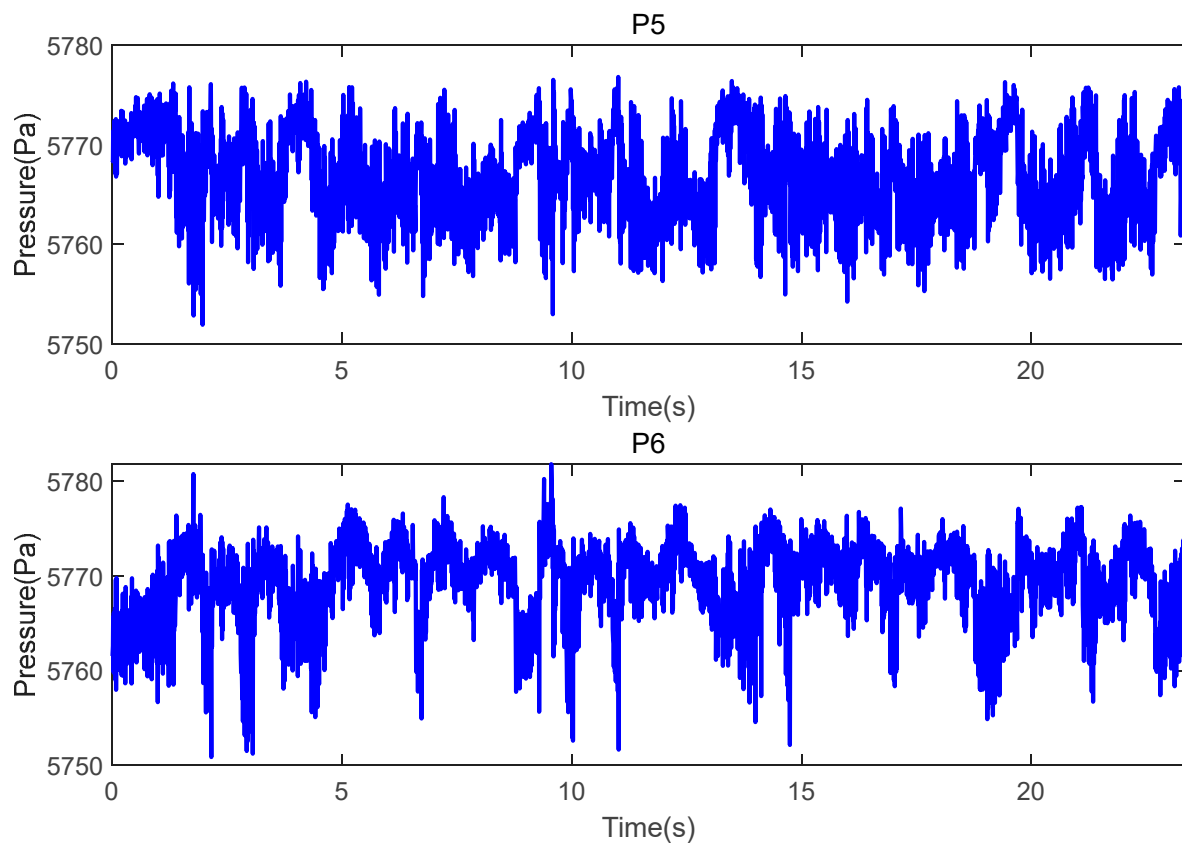


Figure 6. Pressure variation at the two lift fans of the hovercraft.

4.2. LSTM Air Cushion Pressure Prediction

To validate the effectiveness of the proposed LSTM-based cushion pressure prediction model, its parameter configuration is provided in Table 1. Using historical system states from the past 20 time steps and trained over 120 epochs on offline data, the model accurately predicts the pressure values of the four air chambers at the next time step.

Table 1. LSTM Prediction Model Settings.

Parameter Name:	Set Values	Instructions
Input dimension	(20, 6)	Time step $\times$ number of features
Output dimension	4	prediction targets (P_1 , P_2 , P_3 , P_4)
network architecture	128 $\rightarrow$ 128	with 128 hidden units per layer
Dropout	0.2	Used to prevent overfitting
Optimizer	Adam	Adaptive Moment Estimation Optimizer
Batch Size	128	Ensure training stability
Number of training epochs	120	Cooperate with the early stop method
loss function	Huber Loss	Huber Loss is more robust to outliers

The training process of the prediction model is illustrated in Figure 7. This study employs the Huber loss function, which combines the smooth convergence properties of MSE with the robustness of MAE against outliers. As can be observed, both the training loss and validation loss decrease synchronously as the number of iterations increases, converging after approximately 20 epochs without exhibiting overfitting. The stable convergence of the training process lays a foundation for obtaining a high-precision prediction model.

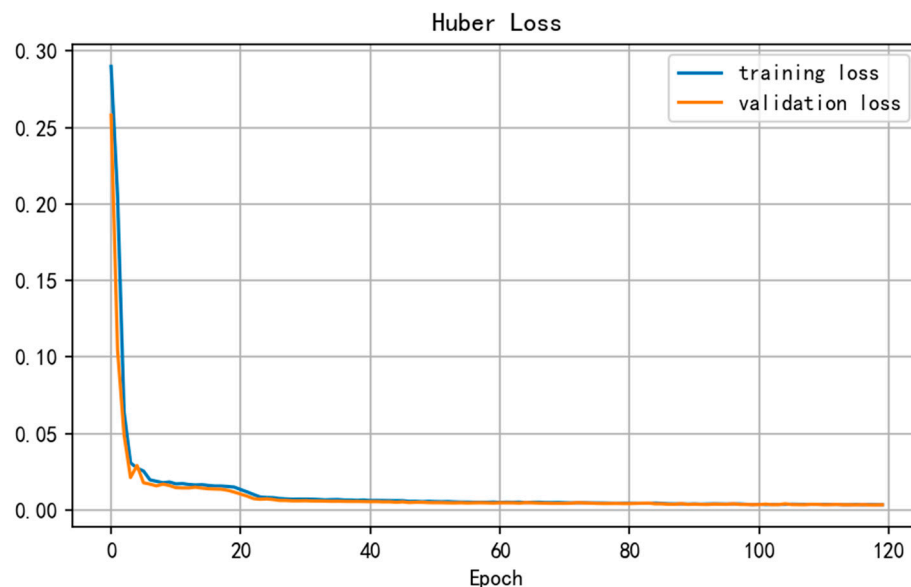


Figure 7. Training and Validation of LSTM Prediction Model Huber Loss Curve.

The pressure prediction results are shown in Figure 8. To evaluate the prediction accuracy of the LSTM algorithm for the pressure in each air chamber, the mean absolute error (MAE), root mean square error (RMSE), and coefficient of determination (R^2) were adopted as evaluation metrics, as summarized in Table 2. The proposed LSTM prediction model demonstrated outstanding performance on the test set: it achieved an average MAE of 2.2497 Pa and an average RMSE of 3.0337 Pa across the four air chambers, with a mean R^2 value of 0.7624. These results confirm the high prediction accuracy of the proposed method and establish a reliable foundation for subsequent offline reinforcement learning.

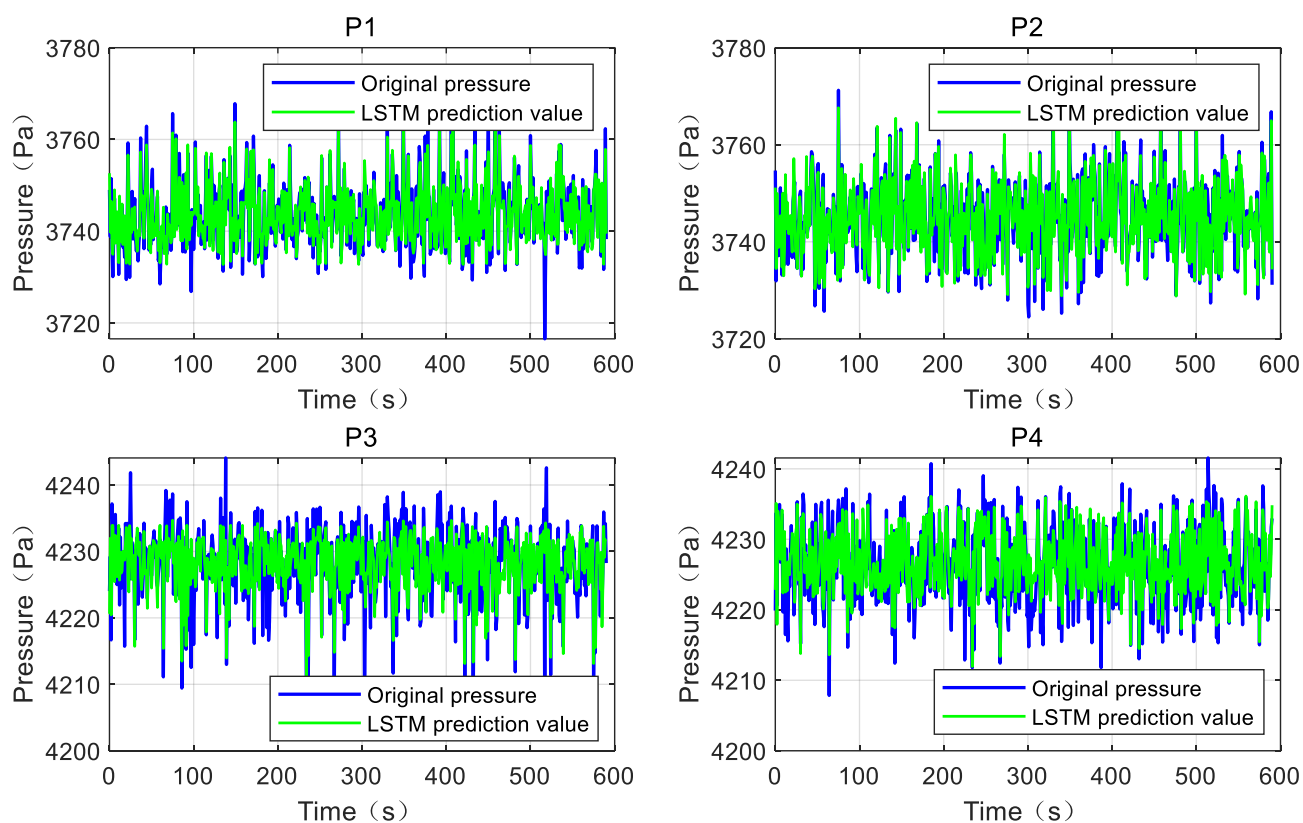


Figure 8. Comparison of pressure prediction for each chamber.

Table 2. Performance Indicators of LSTM Model for Pressure Prediction in Each Gas Chamber.

Air Chamber	MAE (Pa)	RMSE (Pa)	R^2
P_1	2.2463	2.9957	0.8302
P_2	1.8726	2.5183	0.9004
P_3	2.4950	3.5560	0.6110
P_4	2.3849	3.0647	0.7079
Average	2.2497	3.0337	0.7624

4.3. TD3-BC Reinforcement Learning Decision Control

To evaluate the performance of the TD3-BC algorithm in hovercraft cushion pressure control, this study compares the control effectiveness of the LSTM-based prediction model integrated with both the adaptively weighted TD3-BC and the conventional TD3 algorithm. We repeated the training process of the core adaptive TD3-BC algorithm multiple times with different random seeds, while keeping the dataset and LSTM prediction model fixed. All algorithms were executed under the same testing environment, and the parameter settings of the adaptive TD3-BC reinforcement learning approach are provided in Table 3.

Table 3. Adaptive TD3-BC Reinforcement Learning Parameter Settings.

Parameter Name	Set Values	Instructions
Actor Network Structure	256-256-2	Strategic Network Structure
Critic network structure	256-256-1	Value Network Structure
BC coefficient α	2.5	BC regularization strength
Discount factor	0.99	Discount rate for future rewards
Reward function parameters λ_a	0.12	Strategy smoothness parameter
Reward function parameters λ_b	0.6	Stability parameter of chamber pressure
Target network update coefficient τ	0.005	Ensure training stability
Replay buffer size	300,000	Ensure sample diversity
batch size B	256	Training batch size
Actor learning rate	3×10^{-4}	Adam optimizer learning rate
Critic learning rate	3×10^{-4}	Adam optimizer learning rate

As shown in Figure 9, compared to the uncontrolled cushion pressure and the LSTM-TD3 control algorithm, the proposed adaptively weighted LSTM-TD3-BC reinforcement learning algorithm effectively maintains the chamber pressure within the desired range with smoother adjustments. Figure 10 illustrates a comparison of lift fan speeds under different control algorithms. In contrast to the LSTM-TD3 method, the proposed adaptive LSTM-TD3-BC results in smaller fluctuations in fan speed, which not only stabilizes cushion pressure but also contributes to prolonged fan service life. The instantaneous reward curves are depicted in Figure 11. Although both methods achieve stable chamber pressure, the adaptive LSTM-TD3-BC algorithm demonstrates more consistent learning performance, owing to the effective adjustment of adaptive weights, as further supported by Figure 12.

In summary, the proposed adaptive LSTM-TD3-BC reinforcement learning-based predictive control method for hovercraft cushion pressure enables accurate prediction of chamber pressure and facilitates stable control decisions for cushion pressure regulation by leveraging historical experiential data and a reliable prediction model. Stable and autonomous control of cushion pressure is essential for the navigation of hovercrafts. Therefore, effective and smooth cushion pressure control lays a critical foundation for surface navigation control tasks. While the overall data collection was a single instance, the policy optimization and evaluation demonstrate stable and repeatable performance under

random initialization, and that future work will involve testing across a wider variety of sea states and operational conditions.

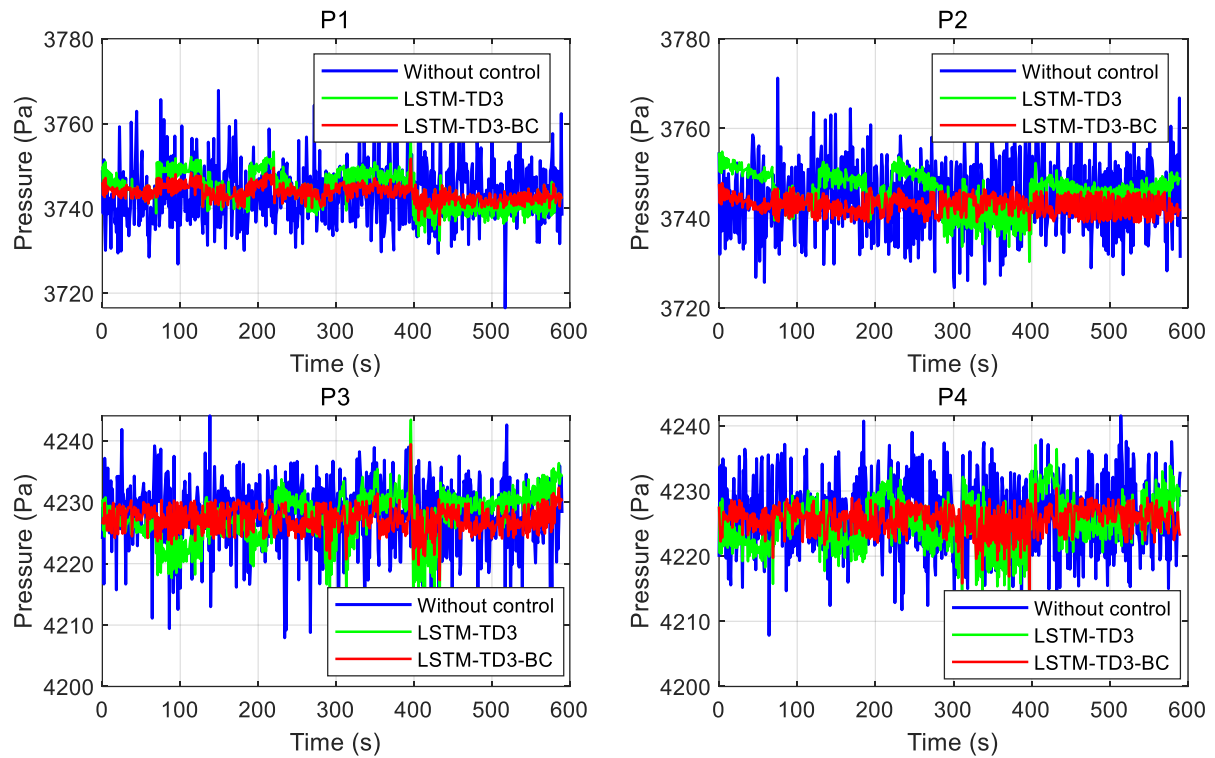


Figure 9. Comparison of chamber pressure control performance under different methods.

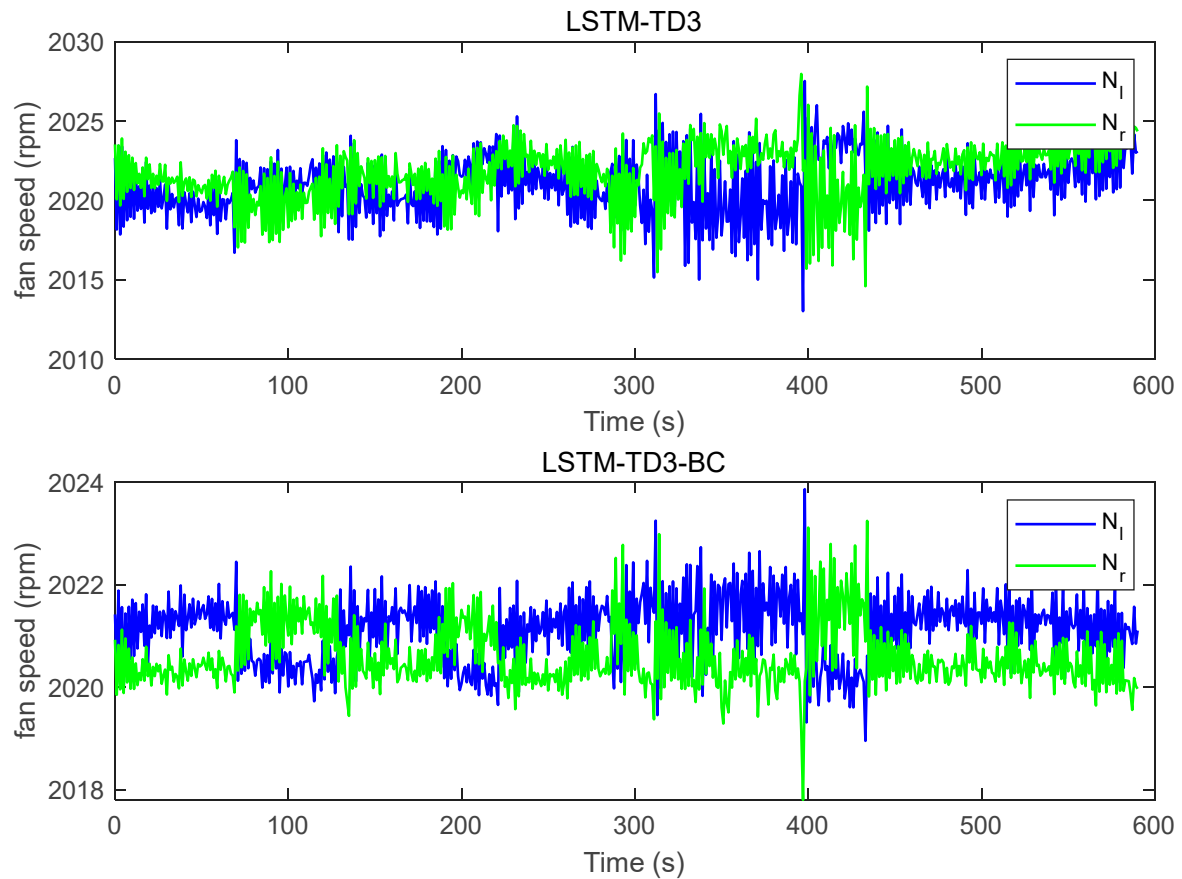


Figure 10. Changes in the speed of the booster fan under different methods.

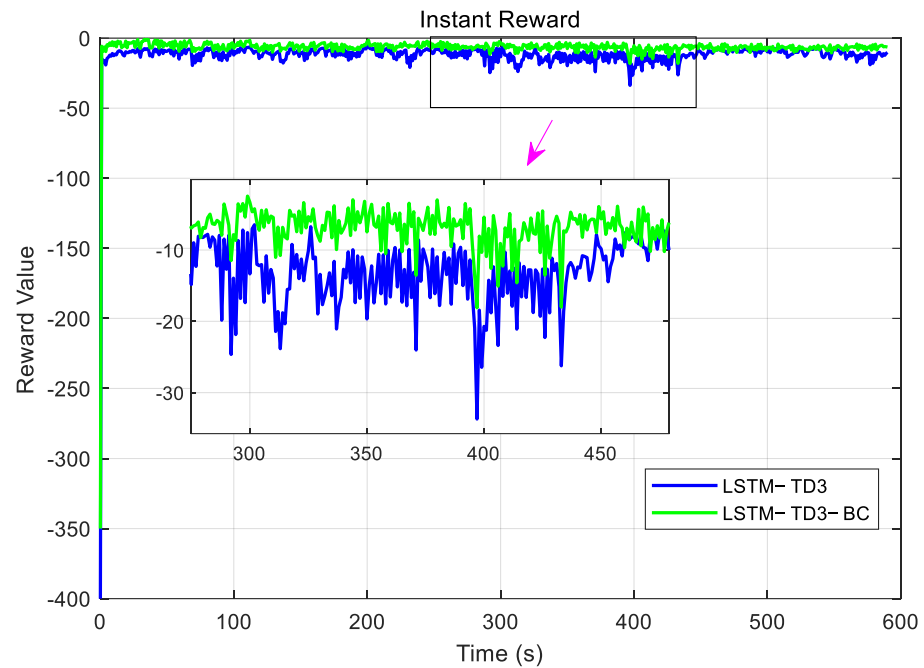


Figure 11. Instantaneous reward changes under different methods.

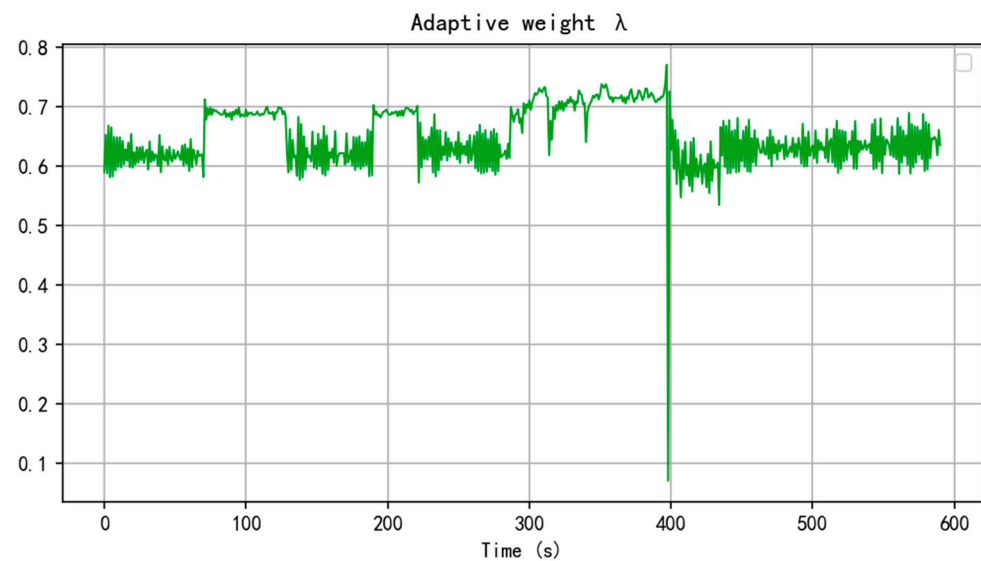


Figure 12. BC adaptive weight change.

5. Conclusions and Future Work

This paper proposed a deep reinforcement learning-based predictive control scheme for cushion pressure in fully skirted hovercraft, aiming to address the challenges of modeling complexity, pressure instability, and control delays in the lift system. Firstly, an LSTM network with a fixed-time window was employed to accurately predict chamber pressure, effectively capturing the dynamic coupling between fan speed and chamber pressure. Secondly, a novel adaptive behavior cloning mechanism was embedded into the TD3-BC framework, which dynamically balances the reinforcement learning objective and historical policy constraints through an auto-adjusted weighting coefficient, thereby effectively mitigating distribution shift and policy degradation in the offline RL setting. By integrating the LSTM prediction model with the adaptive TD3-BC offline reinforcement learning algorithm, stable cushion pressure control was achieved. Finally, simulations demonstrated that the proposed method not only improves the accuracy and robustness

of chamber pressure prediction but also achieves smooth pressure control while reducing fluctuations in lift fan speed, thus extending the service life of the equipment. Future research will focus on refining the proposed method and implementing the algorithm in real-world cushion pressure control applications for fully skirted hovercraft.

While this study demonstrates the effectiveness of the proposed method, some limitations remain. Validation was conducted solely via simulation, creating a sim-to-real gap, and control performance depends heavily on the quality of the offline dataset. Future work will focus on implementing the algorithm on a real hovercraft platform through hardware-in-the-loop testing. We will also investigate adaptive data collection strategies, extend the framework to more extreme sea conditions, and incorporate additional practical constraints such as actuator saturation and safety guarantees.

Author Contributions: Conceptualization, H.Z. and L.D.; methodology, H.Z.; software, H.Z.; validation, H.Z.; formal analysis, H.Z.; investigation, L.D. and Y.W.; resources, Y.W.; data curation, H.Z.; writing—original draft preparation, H.Z.; writing—review and editing, L.D. and Y.W.; supervision, Y.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by National Natural Science Foundation of China, Grant No. 52501438; Fundamental Research Funds for the Central Universities, Grant No. 3072025YC0403; Hainan Provincial Natural Science Foundation Youth Fund Project, Grant No. 725QN386.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to confidentiality restrictions.

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

References

1. Liu, R.Q.; Li, X.F.; Qian, H.R.; Han, D.F. Modeling and experimental study of the cushion lift dynamics system of a polar hovercraft. *Ocean. Eng.* **2023**, *284*, 115246. [\[CrossRef\]](#)
2. Rezaei, S.; Parlak, B.O.; Yavasoglu, H.A. Enhancing hovercraft energy performance: Adaptive lift fan control for optimal efficiency. *Sustain. Energy Technol. Assess.* **2024**, *71*, 103977. [\[CrossRef\]](#)
3. Ganaie, M.A.; Hu, M.H.; Malik, A.K.; Tanveer, M.; Suganthan, P.N. Ensemble deep learning: A review. *Eng. Appl. Artif. Intell.* **2022**, *115*, 105151. [\[CrossRef\]](#)
4. Yang, X.L.; Song, Z.X.; King, I.; Xu, Z.L. A Survey on Deep Semi-Supervised Learning. *IEEE Trans. Knowl. Data Eng.* **2023**, *35*, 8934–8954. [\[CrossRef\]](#)
5. Safonova, A.; Ghazaryan, G.; Stiller, S.; Main-Knorn, M.; Nendel, C.; Ryo, M. Ten deep learning techniques to address small data problems with remote sensing. *Int. J. Appl. Earth Obs. Geoinf.* **2023**, *125*, 103569. [\[CrossRef\]](#)
6. 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. [\[CrossRef\]](#)
7. Zhao, Z.J.; He, W.T.; Mu, C.X.; Zou, T.; Hong, K.S.; Li, H.X. Reinforcement Learning Control for a 2-DOF Helicopter With State Constraints: Theory and Experiments. *IEEE Trans. Autom. Sci. Eng.* **2024**, *21*, 157–167. [\[CrossRef\]](#)
8. Wallace, B.A.; Si, J.N. Reinforcement Learning Control of Hypersonic Vehicles and Performance Evaluations. *J. Guid. Control Dyn.* **2024**, *47*, 2587–2600. [\[CrossRef\]](#)
9. Zhou, H.; Wang, Y.H.; Jiyang, E.; Wang, X.L. Autonomous Control of Lift System based on Actor-Critic Learning for Air Cushion Vehicle. In Proceedings of the OCEANS Conference, Limerick, Ireland, 5–8 June 2023.
10. Wang, Q.S.; Fu, M.Y.; Wang, Y.C.; Xu, Y.J. State-constrained safety trajectory tracking control with prescribed performance for a hovercraft under arbitrary initial conditions. *Ocean. Eng.* **2024**, *309*, 118378. [\[CrossRef\]](#)
11. Hou, Y.K.; Zhao, J.; Zhang, R.Q.; Cheng, X.; Yang, L.Q. UAV Swarm Cooperative Target Search: A Multi-Agent Reinforcement Learning Approach. *IEEE Trans. Intell. Veh.* **2024**, *9*, 568–578. [\[CrossRef\]](#)
12. Chen, J.C.; Wang, Y.; Zhang, Y.; Lu, Y.T.; Shu, Q.H.; Hu, Y.J. Extrinsic-and-Intrinsic Reward-Based Multi-Agent Reinforcement Learning for Multi-UAV Cooperative Target Encirclement. *IEEE Trans. Intell. Transp. Syst.* **2025**, 1–13. [\[CrossRef\]](#)
13. Su, M.Q.; Pu, R.M.; Wang, Y.; Yu, M. A collaborative siege method of multiple unmanned vehicles based on reinforcement learning. *Intell. Robot.* **2024**, *4*, 39–60. [\[CrossRef\]](#)
14. Gan, X.L.; Huo, Z.H.; Li, W. DP-A*: For Path Planning of UGV and Contactless Delivery. *IEEE Trans. Intell. Transp. Syst.* **2024**, *25*, 907–919. [\[CrossRef\]](#)

15. Prudencio, R.F.; Maximo, M.; Colombini, E.L. A Survey on Offline Reinforcement Learning: Taxonomy, Review, and Open Problems. *IEEE Trans. Neural Netw. Learn. Syst.* **2024**, *35*, 10237–10257. [[CrossRef](#)]
16. Sharma, A.; Singh, M. Batch reinforcement learning approach using recursive feature elimination for network intrusion detection. *Eng. Appl. Artif. Intell.* **2024**, *136*, 109013. [[CrossRef](#)]
17. Wen, J.B.; Dai, H.A.; He, J.Y.; Xi, M.; Xiao, S.; Yang, J.C. Federated Offline Reinforcement Learning With Multimodal Data. *IEEE Trans. Consum. Electron.* **2024**, *70*, 4266–4276. [[CrossRef](#)]
18. Janiesch, C.; Zschech, P.; Heinrich, K. Machine learning and deep learning. *Electron. Mark.* **2021**, *31*, 685–695. [[CrossRef](#)]
19. Zhao, J.F.; Mao, X.; Chen, L.J. Speech emotion recognition using deep 1D & 2D CNN LSTM networks. *Biomed. Signal Process. Control* **2019**, *47*, 312–323. [[CrossRef](#)]
20. Nagarajan, S.M.; Devarajan, G.G.; Jerlin, M.A.; Arockiam, D.; Bashir, A.K.; Al Dabel, M.M. Deep Multi-Source Visual Fusion With Transformer Model for Video Content Filtering. *IEEE J. Sel. Top. Signal Process.* **2025**, *19*, 613–622. [[CrossRef](#)]
21. Wang, K.; Ma, C.X.; Qiao, Y.H.; Lu, X.J.; Hao, W.N.; Dong, S. A hybrid deep learning model with 1DCNN-LSTM-Attention networks for short-term traffic flow prediction. *Phys. A-Stat. Mech. Its Appl.* **2021**, *583*, 126293. [[CrossRef](#)]
22. Faroughi, S.A.; Pawar, N.M.; Fernandes, C.; Raissi, M.; Das, S.; Kalantari, N.K.; Mahjour, S.K. Physics-Guided, Physics-Informed, and Physics-Encoded Neural Networks and Operators in Scientific Computing: Fluid and Solid Mechanics. *J. Comput. Inf. Sci. Eng.* **2024**, *24*, 040802. [[CrossRef](#)]
23. Milani, S.; Topin, N.; Veloso, M.; Fang, F. Explainable Reinforcement Learning: A Survey and Comparative Review. *Acm Comput. Surv.* **2024**, *56*, 1–36. [[CrossRef](#)]
24. Yu, Y.; Si, X.S.; Hu, C.H.; Zhang, J.X. A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures. *Neural Comput.* **2019**, *31*, 1235–1270. [[CrossRef](#)] [[PubMed](#)]
25. Xu, J.; Chen, L.X.; Ren, S.L. Online Learning for Offloading and Autoscaling in Energy Harvesting Mobile Edge Computing. *Ieee Trans. Cogn. Commun. Netw.* **2017**, *3*, 361–373. [[CrossRef](#)]
26. Emerson, H.; Guy, M.; McConville, R. Offline reinforcement learning for safer blood glucose control in people with type 1 diabetes. *J. Biomed. Inform.* **2023**, *142*, 104376. [[CrossRef](#)]

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.