CHAPTER 1

From Rules to Learning

To see clearly, one must zoom out before diving in

1.1 THE RISE AND FALL OF EXPERT SYSTEMS

In the early decades of **Artificial Intelligence (AI), expert systems** were seen as a promising way to automate reasoning. Popular from the 1970s to the late 1980s, these systems relied on handcrafted if-then rules to replicate expert knowledge in narrow domains. For a time, they showed potential and attracted interest.

However, expert systems faced serious limitations. The knowledge acquisition bottleneck (manually extracting rules from experts) was slow and incomplete. As rule bases grew, systems became brittle, hard to maintain, and unable to adapt to new data or changing environments. Their lack of scalability and flexibility led to declining enthusiasm.

By the early 1990s, expert systems had largely fallen out of favor, giving way to **Machine Learning (ML).** ML systems learn patterns directly from data, making them more adaptable, robust, and suitable for complex tasks. The shift from rule-based AI to data-driven learning reflects a deeper realization: in many real-world problems, writing all the rules is simply not feasible.

The limitations of expert systems are evident in real-world problems like controlling autonomous vehicles. These systems must operate in dynamic, unpredictable environments, navigating traffic, pedestrians, weather, and countless edge cases. Trying to cover every case with rules is both unrealistic and naive. In contrast, ML offers a more practical and scalable solution by enabling systems to learn from data and adapt to new, unseen scenarios without relying on handcrafted logic.

1.2 ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING

AI and ML are not the same thing. ML is a subfield of AI. Table 1.1 gives a description of the differences between AI and ML from different aspects.

So, what is ML? As illustrated in Figure 1.1, it can be split into **supervised, unsupervised,** and Reinforcement Learning (RL). This classification is widely accepted. However,

DOI: 10.1201/9781003634409-1

TABLE 1.1 Artificial Intelligence in Contrast to Machine Learning

Aspect	Artificial Intelligence	Machine Learning
Scope	Broad field of intelligent systems	Subfield of AI focused on learning from data
Approach	Can be rule-based, knowledge-based, or data-driven	Data-driven learning approach
Applications	Expert systems, planning, robotics, natural language processing	Neural networks, decision trees, RL
Goal	Mimic human intelligence	Generate intelligence by learning from data and generalizing

the next level of categorization, such as how to structure RL, is more debatable. Splitting RL into value-based and policy-based methods aligns well with the structure of this book, but alternative classifications exist. For example, one could differentiate methods based on whether a Neural Network is used or not.

The rest of this book is about RL, so it will not be explained in this section. However, the other two types of ML, supervised and unsupervised learning, deserve some attention.

Supervised learning involves training a model on labeled data, where the input-output pairs are explicitly provided. The model learns to map inputs to correct outputs based on these examples. Common tasks include classification (e.g., spam detection, image recognition) and regression (e.g., predicting stock prices). Algorithms include linear regression, decision trees, and neural networks.

Neural Networks are computational models inspired by the human brain, consisting of interconnected neurons (nodes) organized in layers. Each neuron processes inputs and passes signals through activation functions to learn complex patterns. Deep neural networks, or deep learning, refer to multi-layered neural networks with multiple hidden layers, allowing them to capture high-level abstractions in data. They excel in tasks like image recognition and natural language processing.

Unsupervised learning deals with unlabeled data, where the model identifies patterns, structures, or relationships without predefined outputs. It is commonly used for clustering,

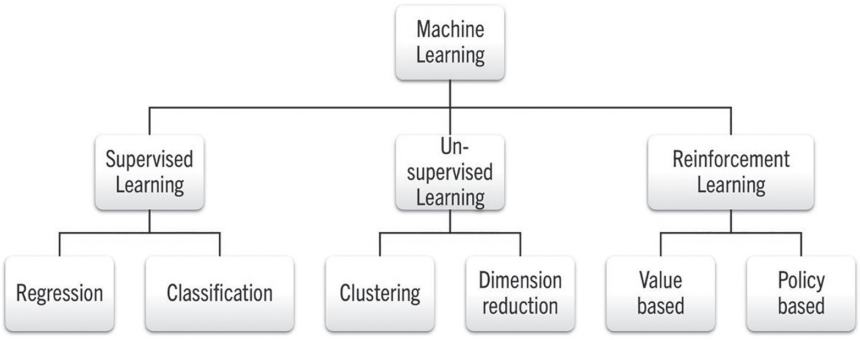

FIGURE 1.1 Machine Learning variants.

dimensionality reduction, and anomaly detection. Faulty sensor reading in an engine is an example of anomaly detection. Algorithms include K-means clustering and autoencoders.

RL is a way of learning through trial and error, where a virtual decision-maker—an agent—interacts with its surroundings and receives feedback from its actions. The goal is to teach the agent to take judicious decisions, ones that lead to consequences aligned with a predefined objective.

RL is ideal for problems that involve making sequential decisions in uncertain environments, where each action influences future outcomes. Uncertainty can come from unpredictable changes in the environment, incomplete information, or randomness in outcomes. Instead of relying on fixed rules, RL agents learn through trial and error, gradually improving their strategy to maximize long-term rewards.

1.3 ADJACENT FIELDS

As an ML method, RL is not well suited for problems where decisions have no impact on future outcomes. When the relationship between inputs and outputs is clear and deterministic, supervised learning is often a more efficient and straightforward choice than RL. For example, tasks such as image classification or predicting housing prices are best addressed with supervised learning, since each input has a direct output.

A related field to RL is control theory, which is a common course for many engineering students. The main challenge in control theory is to ensure that a system follows a reference value. For example, if a car is supposed to maintain a certain speed, the control system automatically adjusts the throttle, even in the presence of disturbances such as wind or road gradients. In this context, simple methods like Proportional–Integral–Derivative (PID) control remain unsurpassed in terms of stability and simplicity. PID controllers are robust, easy to implement, and effective across a wide range of systems. RL, on the other hand, can function as an outer loop that determines which reference value is most appropriate under the prevailing conditions. In this way, the two approaches complement each other. Classical control handles the microscale, providing precise low-level regulation, while RL handles the macroscale, focusing on strategy and long-term planning. A useful analogy is that classical control corresponds to the car keeping within its lane with the help of a good steering servo, while RL is the driver who decides which lane is the smartest to stay in to reach the destination as quickly and safely as possible.

Optimal control is another adjacent field. Two examples within this field are Dynamic Programming (DP) and Model Predictive Control (MPC). MPC is widely employed in industrial control systems, power systems, and robotics, where accurate models of the system dynamics are available. In environments that are stable and predictable, methods such as MPC or DP are often more suitable than RL.

1.4 HISTORY OF REINFORCEMENT LEARNING

The history of RL is summarized in Figure 1.2. The figure is not exhaustive; only major milestones are included. The milestone "Real world impact" is exemplified by ChatGPT, which uses RL from human feedback to improve its responses. The techniques in the figure will be explained throughout the book.

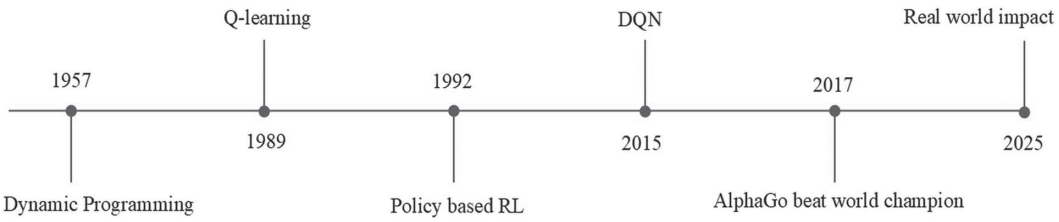

FIGURE 1.2 History of RL. DQN denotes Deep Q-Network. "Real world impact" reflects multiple achievements.

The early foundations of RL emerged from multiple disciplines, including psychology and control theory, forming the basis of modern decision-making models. One of the earliest concepts was introduced by Herbert Robbins in 1952 [1], where he formulated the multi-armed bandit problem, a fundamental framework for decision-making under uncertainty. The central question in the paper is how to design sequential experiments where an agent must choose between multiple options (arms) to maximize cumulative rewards.

Richard Bellman's book *Dynamic Programming* [2], published in 1957, introduced a powerful mathematical framework for solving sequential decision-making problems, where decisions at one stage influence future outcomes. The key innovation was the principle of optimality, which states that an optimal policy has the property that, regardless of the initial state, the remaining decisions must constitute an optimal strategy with respect to the subsequent states.

Dynamic programming remains widely used across various fields, including control theory and economics, particularly when the model is well-defined, and the goal is to determine an optimal sequence of actions. Its effectiveness lies in its ability to break down complex decision-making problems into manageable subproblems, solving them recursively to achieve optimal solutions.

Richard S. Sutton's 1984 Ph.D. thesis, *Temporal Credit Assignment in Reinforcement Learning* [3], focused on the fundamental challenge of credit assignment in sequential decision-making—how an agent determines which past actions led to future rewards. Sutton explored Temporal Difference (TD) learning, a method that updates value estimates based on differences between successive predictions rather than waiting for the final outcome.

A key contribution of this work was the development and formalization of TD learning, which later became a core component of modern RL. He also established theoretical connections between TD learning and DP. His thesis laid the groundwork for subsequent breakthroughs, including Q-learning, influencing the development of modern RL. Sutton's insights remain foundational in RL research today.

Q-learning was first introduced by Chris Watkins in his Ph.D. thesis from 1989, *Learning from Delayed Rewards* [4], at the University of Cambridge. In this work, Watkins proposed a model-free RL algorithm that enables an agent to learn optimally without requiring prior knowledge of the environment's dynamics. The core idea behind Q-learning is to estimate the Q-value function, which represents the expected cumulative reward for taking a specific action in each state and following the optimal policy thereafter. The algorithm

updates these Q-values iteratively using the Bellman equation, incorporating a learning rate and a discount factor to balance immediate and future rewards.

RL has seen significant advancements beyond its early foundations, particularly with the development of policy gradient methods, deep RL, and model-based approaches. In the late 1990s and early 2000s, policy gradient, deep learning, and actor-critic methods gained attention.

Williams proposed REINFORCE in 1992 [5], a family of algorithms that optimize policies using stochastic gradient ascent. The key idea is to update policy parameters in a direction that maximizes expected rewards by estimating the gradient directly from sampled trajectories. This work laid the foundation for modern policy optimization methods.

In 1983, Barto, Sutton, and Anderson introduced the actor-critic architecture [6], a framework that remains popular in RL. The authors proposed a learning system where an actor selects actions, and a critic evaluates them. However, it was not until the 1990s that the actor-critic architecture was refined and widely applied.

In 2016, Asynchronous Advantage Actor-Critic (A3C) [7] was introduced. A core innovation of the paper was the Advantage Actor-Critic (A2C) framework, which introduced the advantage function to enhance policy gradient updates. Instead of using raw returns or action-value estimates directly, the advantage function refines learning by using the value estimate provided by the critic. The consequence is an actor guided by the critic.

In 2015, the scientific journal Nature published a paper proposing deep neural network as a function approximator [8], allowing RL to scale to large state spaces. Instead of maintaining a Q-table, Deep Q-Network (DQN) parameterized the Q-function with a neural network, enabling the incorporation of continuous and high-dimensional state spaces.

In the thesis *Reinforcement Learning and Simulation-Based Search* [9], D. Silver developed fundamental principles for simulation-based planning, with a particular focus on Monte Carlo Tree Search. He demonstrated how RL can be effectively combined with search algorithms to tackle complex decision-making problems, especially in the domain of games.

1.5 REAL-WORLD APPLICATIONS OF REINFORCEMENT LEARNING

RL is transforming industrial automation, humanoid robotics, and autonomous drones by enabling robots to learn complex tasks through experience rather than manual programming. In industrial automation, RL is used to optimize robotic arms for assembly, welding, and material handling, allowing them to adapt to changing production needs and improve efficiency. Humanoid robotics benefits from RL in locomotion, object manipulation, and human interaction, enabling robots to perform dynamic movements, balance, and handle objects in unstructured environments. In autonomous drones, RL enhances navigation, obstacle avoidance, and delivery optimization, allowing drones to operate in complex environments with minimal human intervention.

FIGURE 1.3 Quadrupedal robot. The dog-like robot weighs 9 kg and has 12 actuated joints (three per leg), making it well suited for dynamic locomotion research. It measures approximately 24 cm in height and 40 cm in length. The figure is reproduced from reference [10].

One example of RL applied in robotics is from MIT[1] (2022) [10]. The paper presents an RL-based framework that enables high-speed locomotion in a quadrupedal robot. The robot, illustrated in Figure 1.3, is capable of agile movements such as running, jumping, and flipping. Once trained, the robot exhibited a range of learned locomotion skills, including fast acceleration, stable running, recovery from perturbations, and adaptive foot placement. These skills emerged without explicit programming, demonstrating RL's ability to discover effective movement strategies.

RL is transforming autonomous vehicles by enabling real-time decision-making, adaptive driving strategies, and enhanced navigation in dynamic and uncertain environments. Unlike traditional rule-based approaches, RL allows autonomous vehicles to learn from experience and refine their behavior through trial and error, improving their ability to handle complex traffic scenarios. RL is useful in motion planning and path optimization, where algorithms dynamically adjust routes based on traffic conditions and unforeseen obstacles. Additionally, RL is applied to traffic light control and intersection management, helping to reduce congestion and optimize urban traffic flow, contributing to safer and more efficient transportation systems.

Work related to path planning has been performed by Swedish researchers [11]. The work integrates Monte Carlo tree search-based path planning with motion trajectory optimization to enhance the coordination of fully automated vehicles in confined environments. The proposed approach ensures that vehicles navigate efficiently while minimizing energy consumption. By optimizing both route selection and movement trajectories, the framework enhances operational productivity and multi-vehicle coordination, reducing conflicts and improving efficiency in space-constrained settings. A key feature of the path planning is its ability to dynamically adjust routes to ensure the effective use of charging infrastructure. Monte Carlo tree search is a search-based decision-making technique often

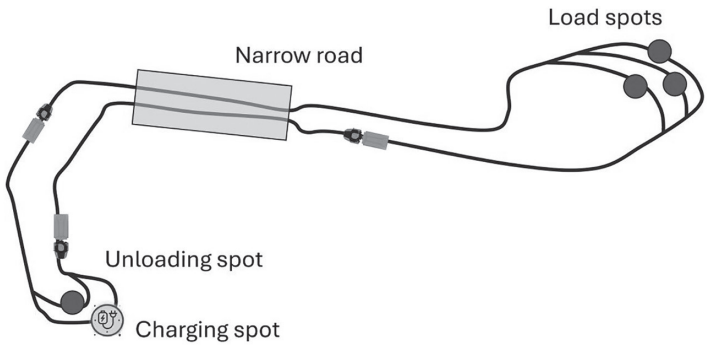

FIGURE 1.4 Monte Carlo tree search-based path planning. Vehicle coordination is essential; for example, two vehicles are not allowed to meet in the narrow road section. Path planning must also determine whether a vehicle should charge. The figure is adapted from reference [11].

used in RL, allowing vehicles to explore and evaluate multiple possible paths before selecting the optimal one. An example scenario, using the proposed path planner,[2] is illustrated in Figure 1.4.

Traditional financial models often rely on static strategies or predefined rules, which struggle to adapt to rapidly changing markets. RL-powered agents optimize trade execution by learning how to buy, sell, or hold assets based on real-time market fluctuations, helping traders maximize profits while minimizing transaction costs. The book *Foundations of RL with Applications in Finance* [12] provides a comprehensive introduction to RL techniques with a strong focus on financial applications. The book explores how RL can be used to solve key financial problems, including dynamic asset allocation, derivatives pricing and hedging, optimal trade execution, and market-making strategies.

RL is applied in medicine by enabling personalized treatment plans, optimized drug discovery, intelligent diagnostics, robotic-assisted surgeries, and efficient hospital resource management. One example of how RL is applied in medicine is AlphaFold [13]. It is an AI system developed by DeepMind that significantly has advanced protein structure prediction by accurately determining 3D protein structures based solely on their amino acid sequences. In 2020, AlphaFold 2 demonstrated remarkable performance in the Critical Assessment of Structure Prediction challenge, achieving near-experimental accuracy, a breakthrough that has transformed drug discovery, disease research, and synthetic biology. One of the key innovations in AlphaFold's success is the use of RL and Monte Carlo tree search to improve structure prediction. In 2024, AlphaFold was awarded the Nobel prize.

RL is playing an increasingly important role in energy systems and smart grids, optimizing power generation, distribution, and consumption while integrating renewable energy sources. One promising application of RL in this domain is Vehicle-to-Grid (V2G) technology, where electric vehicles act as energy storage units, supplying power back to the grid when needed.

The problem is surprisingly challenging due to imperfect system models and the random power activations caused by frequency-stabilization services.[3] As a result, classic

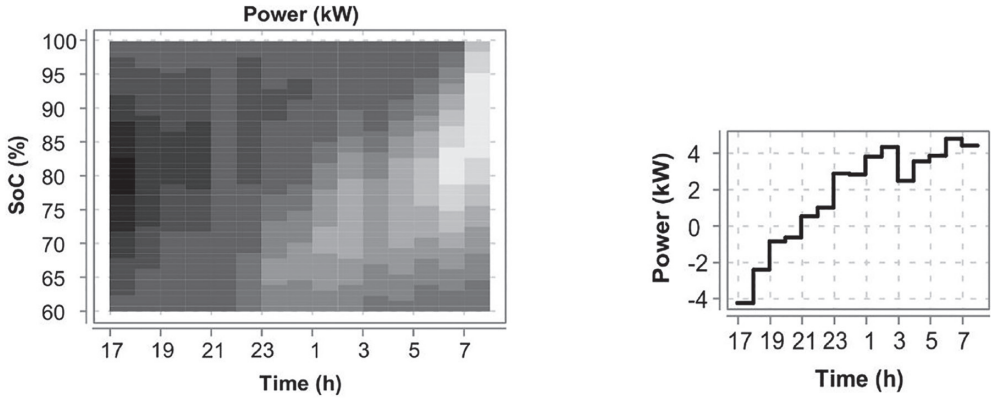

FIGURE 1.5 Vehicle-to-Grid charging strategy agent. Charge-discharge policy (left) and power-time trajectory (right). Positive power, light areas in the left plot, denotes charging, while negative power, dark areas in the left plot, corresponds to discharging. The measure SoC stands for state of charge. The plots are from [19].

optimization methods such as linear programming are insufficient; instead, an artificial agent can be trained to handle the wide range of possible future battery state-of-charge trajectories.

Some papers proposing RL for charge-discharge strategies of electric vehicles are given in references [14–19]. Figure 1.5 is from reference [19], it shows the result of training an agent able to perform optimal charge and discharge of a mid-sized electric car. Most of the charging, buying electricity, is performed at night when electricity is cheaper.

RL has become a crucial component in natural language processing, particularly in training large language models like ChatGPT. Traditional natural language processing models rely on supervised learning, where they learn from human-annotated datasets, but RL enables models to refine their responses based on feedback, improving fluency, coherence, and alignment with human preferences. More about this can be found in reference [20].

Finally, the remarkable story of RL in games will be told. Old-style chess engines rely on brute-force search and handcrafted evaluation functions, whereas AlphaGo, developed by DeepMind, introduced a self-play RL approach combined with Monte Carlo tree search to evaluate positions and select optimal moves. By training solely through self-play, AlphaGo developed human-like intuition and novel strategies, outperforming top chess engines and human grandmasters. Similarly, Go, with its immense search space of 10^{170} possible board positions, presented a greater challenge for AI. The number 10^{170} is much larger than the estimated number of atoms in the observable universe. It is not practical to have rules determining what to do for every possible board position.

DeepMind's AlphaGo revolutionized gameplay by combining deep learning, RL, and Monte Carlo tree search, achieving superhuman performance by defeating world champion Lee Sedol in 2016. Further advancements led to AlphaGo Zero, which discarded human training data and learned purely from self-play, demonstrating unprecedented strategic

depth and adaptability. More about AlphaGo can be found in references [21] and [22]. Additionally, there are videos available on YouTube showcasing the tournament between Lee Sedol and AlphaGo.

The RL success in games is not limited to board games. In 2022, Nature published the paper "Outracing champion Gran Turismo drivers with deep reinforcement learning" [23]. This study detailed the development of an AI agent capable of outperforming professional human drivers in the high-fidelity racing simulator Gran Turismo.

These were just a few examples of real-world RL applications; additional examples are provided in Appendix H.

NOTES

1. Massachusetts Institute of Technology.
2. Work related to RL-supported path planning has twice received the Best Patent Award at the Volvo Group.
3. Random power activations in V2G occur because the vehicle provides frequency stabilization. When the grid frequency deviates from 50 Hz, the operator sends a regulation signal that instructs the EV to charge or discharge. These deviations happen unpredictably due to fluctuating load and renewable generation, so the power requested from the vehicle appears random, continuously switching between charging and discharging.

REFERENCES

1. H. Robbins, "Some Aspects of the Sequential Design of Experiments", Bulletin of the American Mathematical Society, vol. 58, pp. 527–535, 1952.
2. R. E. Bellman, *Dynamic Programming*. Princeton, NJ, USA: Princeton University Press, 1957.
3. R. S. Sutton, *Temporal Credit Assignment in Reinforcement Learning*, Ph.D. dissertation, University of Massachusetts, Amherst, MA, USA, 1984.
4. C. J. C. H. Watkins, *Learning from Delayed Rewards*, Ph.D. dissertation, University of Cambridge, Cambridge, U.K., 1989.
5. R. J. Williams, "Simple Statistical Gradient-Following Algorithms for Connectionist Reinforcement Learning", Machine Learning, vol. 8, no. 3–4, pp. 229–256, 1992.
6. A. G. Barto, R. S. Sutton, and C. W. Anderson, "Neuronlike Adaptive Elements that Can Solve Difficult Learning Control Problems", IEEE Transactions on Systems, Man, and Cybernetics, vol. 13, no. 5, pp. 834–846, 1983.
7. V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous Methods for Deep Reinforcement Learning", Proceedings of the 33rd International Conference on Machine Learning (ICML), pp. 1928–1937, 2016.
8. V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, and D. Hassabis, "Human-Level Control Through Deep Reinforcement Learning", Nature, vol. 518, no. 7540, pp. 529–533, 2015.
9. D. Silver, *Reinforcement Learning and Simulation-Based Search*, Ph.D. dissertation, University of Alberta, Edmonton, AB, Canada, 2009.
10. G. B. Margolis, G. Yang, K. Paigwar, T. Chen, and P. Agrawal, "Rapid Locomotion via Reinforcement Learning", in Proc. Robotics: Science and Systems (RSS), New York City, NY, USA, 2022.
11. S. Kojchev, J. Hellgren, R. Hult, and J. Fredriksson, "Combined Path and Trajectory Planning for Energy-Efficient Coordination of Automated Vehicles in Confined Areas", Proceedings of the European Control Conference, Stockholm, Sweden, 2024.

12. A. Rao, *Foundations of RL with Applications in Finance*, Boca Raton, FL, USA: Taylor & Francis Group, 2022.
13. J. Jumper, R. Evans, A. Pritzel et al., "Highly Accurate Protein Structure Prediction with AlphaFold", Nature, vol. 596, pp. 583–589. 2021.
14. Y. Zhang, K. Li, C. Du, W. Cai, Y. Lu, and Y. Feng, "Learning-Based Scheduling of Integrated Charging-Storage-Discharging Station for Minimizing Electric Vehicle Users' cost", Journal of Energy Storage, vol. 81, Art. no. 110474, 2024.
15. M. Qiu, J. Ye, and G. Strbac, "A Deep Reinforcement Learning Method for Pricing Electric Vehicles with Discrete Charging Levels", IEEE Transactions on Smart Grid, vol. 11, no. 2, pp. 1175–1185, 2020.
16. F. Zhang, Q. Yang, and D. An, "CDDPG: A Deep-Reinforcement-Learning-Based Approach for Electric Vehicle Charging Control", IEEE Internet of Things Journal, vol. 8, no. 11, pp. 8816–8826, 2021.
17. H. Li, Z. Wan, and H. He, "Constrained EV Charging Scheduling Based on Safe Deep Reinforcement Learning", IEEE Transactions on Smart Grid, vol. 11, no. 3, pp. 2427–2439, 2020.
18. L. Ren and M. Yuan, "Deep Reinforcement Learning for Continuous Electric Vehicles Charging Control with Dynamic User Behaviors", IEEE Transactions on Industrial Informatics, vol. 17, no. 11, pp. 7681–7692, 2021.
19. J. Hellgren and Daniel Jung, "Cost Evaluation of Vehicle-to-Grid Technology in Sweden Using Learning-Based Trading Strategy", in *ICACRS 2025*, IEEE Conference, Pudukkottai, India, 2025.
20. T. Kaufmann, P. Weng, V. Bengs, and E. Hüllermeier, "A Survey of Reinforcement Learning from Human Feedback", arXiv preprint arXiv:2312.14925, 2023.
21. D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis, "Mastering the Game of Go with Deep Neural Networks and Tree Search", Nature, vol. 529, no. 7587, pp. 484–489, Jan. 2016.
22. D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, and D. Hassabis, "Mastering the Game of Go Without Human Knowledge", Nature, vol. 550, no. 7676, pp. 354–359, 2017.
23. P. Wurman, S. Barrett, K. Kawamoto, J. MacGlashan, K. Subramanian, T. J. Walsh, R. Capobianco, A. Devlic, F. Eckert, F. Fuchs, L. Gilpin, P. Khandelwal, V. Kompella, H. Lin, P. MacAlpine, D. Oller, T. Seno, C. Sherstan, M. D. Thomure, H. Aghabozorgi, L. Barrett, R. Douglas, D. Whitehead, P. Dürr, P. Stone, M. Spranger, and H. Kitano, "Outracing Champion Gran Turismo Drivers with Deep Reinforcement Learning", Nature, vol. 602, no. 7896, pp. 223–228, 2022.