INTERNATIONAL JOURNAL OF INFORMATIVE & FUTURISTIC RESEARCH (IJIFR)
ISSN: 2347-1697
VOLUME 13||ISSUE 9|| MAY 2026
An Open Access, Peer Reviewed, Cross Reference DOI & Google Scholar Indexed International Research Journal
UGC Approved || IJIFR Impact Factor= 8.057
Estd Year: 2013 || Email: editors@ijifr.org
Availaible Online at: https://www.ijifr.org/

DOI: https://doi.org/10.64672/IJIFR/26.05.13.09.002 PUBLISHED ON: MAY 04, 2026

ADOPTIMA RL: A DEEP REINFORCEMENT LEARNING FRAMEWORK FOR REAL-TIME ADVERTISEMENT BID OPTIMIZATION

Byrisetty Manoj Kumar [1], S. Usha Rani [2]
[1] Student, [2] Professor & Head
[1,2] Department of Computer Applications,
Viswam Engineering College, Madanapalle, Andhra Pradesh, India

Abstract

The rapid evolution of programmatic advertising has introduced complex challenges in real-time bidding (RTB), where advertisers must make instantaneous and near-optimal bidding decisions in dynamic, partially observable environments. Traditional rule-based and statistical approaches struggle to adapt to fluctuations in user behaviour, market competition and platform dynamics, which frequently results in inefficient budget utilization and suboptimal campaign performance. This paper presents AdOptima RL, an intelligent advertisement optimization framework that leverages deep reinforcement learning (RL) to dynamically learn bidding strategies. The proposed system models the RTB problem as a Markov Decision Process (MDP) and integrates three RL algorithms — Deep Q-Network (DQN), Proximal Policy Optimization (PPO) and an Actor-Critic method — that collectively cover both discrete and continuous bidding action spaces. A correlation-based feature-weighting mechanism enriches the state representation by emphasising attributes that are statistically predictive of bidding outcomes, accelerating learning and improving decision quality. A per-website agent specialization strategy further allows the framework to capture domain-specific dynamics across different advertising platforms, avoiding the generalization limits of a single monolithic model. In addition to bid optimization, the framework includes a budget allocation module that simulates campaign performance using trained agents and produces data-driven recommendations for distributing advertising spend across platforms. Experimental evaluation on a real-world-inspired auction dataset demonstrates that AdOptima RL improves click-through rate and budget utilization efficiency relative to traditional methods, with PPO performing best in continuous action spaces, DQN delivering low-latency decisions in discrete spaces, and Actor-Critic offering stable convergence on long horizons. The findings highlight the potential of reinforcement learning to transform digital advertising into an adaptive, intelligent and performance-driven decision process.

Keywords *Reinforcement learning; real-time bidding; deep Q-network; proximal policy optimization; advertisement optimization*

Recommended Citation:
Kumar, B. M. , Rani, S. H. : " AdOptima RL: A Deep Reinforcement Learning Framework for Real-Time Advertisement Bid Optimization", International Journal of Informative & Futuristic Research (IJIFR), Vol. (13) (9), May 2026, pp. 1407-1412
https://doi.org/10.64672/IJIFR/26.05.13.09.002

1. Introduction

The rapid expansion of digital platforms has fundamentally reshaped advertising into a highly automated, data-driven ecosystem. Real-Time Bidding (RTB) is now the cornerstone of programmatic advertising: each ad impression is auctioned within milliseconds, and advertisers must take instantaneous decisions on bid price, targeting and budget allocation while continuously optimising metrics such as click-through rate (CTR), conversion rate and return on investment. Traditional rule-based heuristics and static statistical models built from historical data struggle in this regime: they cannot adapt to temporal shifts in user behaviour, fluctuations in market competition, or evolving consumption trends, which manifests operationally as overbidding on low-value impressions, underbidding on high-value opportunities, and poor cross-channel budget utilisation.

Reinforcement Learning (RL) has emerged as a strong paradigm for sequential decision-making under uncertainty [1]. Unlike supervised approaches, RL agents learn by interacting with an environment and maximising cumulative reward, which aligns naturally with the stochastic, time-extended nature of RTB. Modelling the bidding problem as a Markov Decision Process (MDP), an RL agent can dynamically adjust its policy on the basis of contextual features such as time, user segment, website category, historical performance and remaining budget. The contribution of this paper is threefold. First, we propose AdOptima RL, a unified framework that integrates Deep Q-Network (DQN) [2], Proximal Policy Optimization (PPO) [3] and an Actor-Critic method, jointly covering discrete and continuous action spaces. Second, we introduce a correlation-based feature-weighting mechanism and a per-website agent specialization strategy that together capture domain-specific bidding dynamics and accelerate convergence. Third, we couple bid optimization with a budget allocation module that turns trained policies into actionable recommendations for cross-platform spend distribution.

2. Literature Survey

Early RTB research framed bid optimization as a CTR-prediction problem solved with regression-based models such as logistic regression and gradient boosting. While effective for static settings, these models do not adapt to the rapidly changing auction dynamics characteristic of modern programmatic exchanges. Subsequent work cast the problem as constrained optimization under dynamic-programming or control-theoretic formulations, but these required strong distributional assumptions and was computationally expensive to scale. Reinforcement learning [1] reframed the problem as sequential decision-making, enabling agents to learn directly from interaction. Cai et al. were among the first to model bidding as an MDP and showed that learned policies outperform fixed heuristics by balancing exploration and exploitation in dynamic auctions.

Deep Reinforcement Learning further widened the operating envelope of RL in advertising. Mnih et al. introduced Deep Q-Networks (DQN) [2], which combined neural function approximation with experience replay and target networks to stabilise learning over high-dimensional state spaces; DQN has since been applied widely to discrete bidding. In parallel, policy-gradient methods such as Proximal Policy Optimization (PPO) [3] have been used for continuous action spaces, providing stable updates through a clipped objective that constrains policy drift. Actor-Critic architectures combine value-based and policy-based learning to reduce gradient variance and improve convergence on long horizons. Recent work has also explored multi-agent and hierarchical RL formulations, but these introduce additional training and coordination complexity. Across the literature, three challenges recur: sparse reward signals due to infrequent clicks, non-stationary environments due to evolving user behaviour and competition, and the limited capacity of single-agent monolithic models to capture platform-specific dynamics. The proposed AdOptima RL framework targets these gaps directly through its multi-algorithm design, correlation-based feature weighting and per-website agent specialization.

3. Proposed Work and Methodology

3.1 Problem Formulation

We formulate RTB bid optimization as a Markov Decision Process (S, A, P, R, γ). The state s ∈ S is a weighted feature vector comprising temporal attributes (day, hour, day-of-week), website category, user segment, historical CTR, cost and remaining-budget features. The action a ∈ A is a bid value, treated as discrete for DQN and continuous for PPO and Actor-Critic. The reward r combines click acquisition; bid efficiency and cost optimization, balancing engagement against spend. The discount factor γ controls the trade-off between short-term clicks and long-term campaign performance. The objective is to learn a policy $\pi(a|s)$ that maximises expected discounted return $E[\Sigma\ \gamma^t\ r_t]$.

3.2 Framework Overview

The framework is organised as a five-layer pipeline. The Data Layer hosts auction records with temporal, website, user, CTR, cost and bid attributes. The Feature Engineering Layer applies label encoding and Min-Max normalisation, then computes correlation-based feature weights using Pearson correlation between each feature and the bid outcome, yielding a weighted state vector. The Environment Layer wraps this as a Gym-compatible MDP with explicit state, action and reward specifications. The Agent Training Layer hosts DQN, PPO and Actor-Critic implementations, instantiated per website category to capture domain-specific dynamics. The Output and Evaluation Layer computes performance metrics, renders bar/pie charts and summary tables, and drives the budget allocation module that recommends cross-platform spend distribution from the trained policies.

Table 1 — Technology Stack of the Proposed System

Layer	Component / Library
Programming language	Python 3
RL algorithms	DQN, PPO, Actor-Critic (Stable Baselines3)
Environment	Gymnasium (custom AdBiddingEnv)
Data manipulation	pandas, NumPy
Feature engineering	Scikit-learn (encoding, scaling)
Visualization	Matplotlib (bar / pie charts)

4. System Architecture and Diagrams

This section presents two IEEE-style engineering diagrams: (i) the system architecture diagram (Fig. 1), capturing the five-layer organisation of AdOptima RL, and (ii) the workflow diagram (Fig. 2), capturing the temporal ordering of the training and evaluation pipeline. Mermaid source is provided beneath each figure.

4.1 System Architecture Diagram

The system architecture, as shown in Fig. 1, is organised into five sequentially connected layers. Raw auction data flows through preprocessing and correlation-weighted feature engineering into the Gym-compatible RL environment, which exposes the state space, action space and reward function. The Agent Training Layer instantiates DQN, PPO and Actor-Critic agents per website category, allowing each agent to specialise to its platform's user demographics and competition dynamics. The Output and Evaluation Layer aggregates trained policies into performance metrics, visualisations and budget recommendations.

4.2 Workflow Diagram

The workflow, illustrated in Fig. 2, begins with the loading and preprocessing of the auction dataset, followed by the construction of the RL environment. For each website category, an agent is instantiated and trained through repeated interaction with the environment: the environment emits a state, the agent selects a bid action, the auction is simulated, and a reward is fed back to update the policy. Training

continues for 50 000 timesteps, after which the trained models drive the evaluation and budget allocation stages, producing reports and charts for the advertiser.

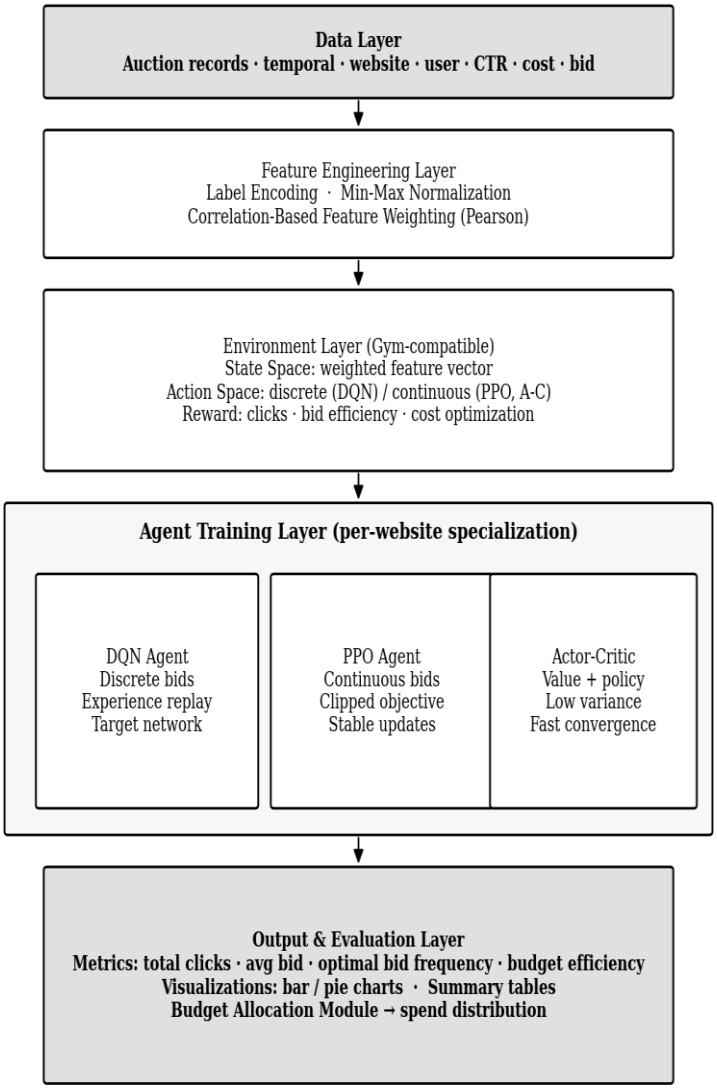

Figure 1 — System Architecture of AdOptima RL

Mermaid source for Fig. 1

```
```mermaid
flowchart TD
    D["Data Layer<br/>auction records · temporal · website · user · CTR · cost · bid"] --> F
    F["Feature Engineering<br/>encoding · normalization · correlation weighting"] --> E
    E["Environment Layer (Gym)<br/>state · action (discrete/continuous) · reward"] --> AG
    subgraph AG["Agent Training Layer (per-website)"]
        DQN["DQN Agent<br/>discrete bids"]
        PPO["PPO Agent<br/>continuous bids"]
        AC["Actor-Critic<br/>value + policy"]
    end
    AG --> O["Output & Evaluation<br/>metrics · charts · Budget Allocation"]
```
```

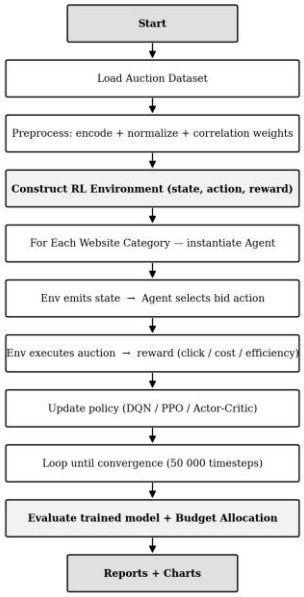

Figure 2 — Workflow Diagram of the AdOptima RL Training and Evaluation Pipeline

Mermaid source for Fig. 2

```
```mermaid
flowchart TD
    S([Start]) --> L[Load Auction Dataset]
    L --> P[Preprocess: encode + normalize + correlation weights]
    P --> EV[Construct RL Environment]
    EV --> A[For each website -> instantiate Agent]
    A --> ST[Env emits state -> Agent selects bid]
    ST --> R[Env executes auction -> reward]
    R --> U[Update policy DQN / PPO / A-C]
    U --> C{Converged?}
    C -- No --> ST
    C -- Yes --> E[Evaluate + Budget Allocation]
    E --> H([Reports + Charts])
```
```

5. Results and Discussion

AdOptima RL was evaluated on a real-world-inspired RTB dataset containing temporal attributes, website categories, user segments, historical CTRs and bid-related parameters. The dataset was preprocessed via label encoding, Min-Max normalisation and correlation-based feature weighting. Three agents — DQN, PPO and Actor-Critic — were trained independently for each website category for 50 000 timesteps each, using Stable Baselines3 over a custom Gymnasium environment. Performance was measured along four axes: total clicks generated, average bid value, most frequent bid (as a proxy for the dominant learned strategy) and budget utilisation efficiency.

DQN performed efficiently in discrete action environments, selecting optimal bid levels from a predefined set with stable learning enabled by experience replay and target networks; it is particularly attractive when low-latency decisions are required. PPO delivered superior performance on continuous action spaces by enabling fine-grained bid selection, with its clipped objective preventing destabilising policy updates [3]. Actor-Critic provided the most stable convergence on long training horizons, trading peak accuracy for variance reduction. Across all three algorithms, the per-website agent specialization strategy materially improved click acquisition compared with a global model, and correlation-based feature weighting accelerated convergence by emphasising the most predictive attributes. The budget allocation module translated trained policies into spend distributions that allocated higher budget to high-performing websites and reduced spend on low performers.

Table 2 — Algorithm-Level Behaviour Summary

Algorithm	Action Space	Observed Strength
DQN	Discrete	Low-latency decisions; stable under replay
PPO	Continuous	Fine bid control; stable clipped updates
Actor-Critic	Continuous	Low-variance gradients; long-horizon stability
Per-website specialization	—	Captures platform-specific dynamics
Correlation-based weighting	—	Faster convergence, better state representation
Budget allocation module	—	Cross-platform spend recommendations

Two limitations were observed. First, training continuous models such as PPO and Actor-Critic is computationally demanding and benefits from GPU acceleration. Second, model performance is sensitive to hyperparameter configuration (learning rate, clipping factor, replay buffer size), calling for principled tuning when the system is deployed on new datasets.

6. Conclusion

This paper presented AdOptima RL, a deep-reinforcement-learning framework for real-time advertisement bid optimization that models RTB as an MDP and integrates DQN, PPO and Actor-Critic agents across discrete and continuous action spaces. Two design contributions — correlation-based feature weighting and per-website agent specialization — improve state representation and capture platform-specific dynamics, while a budget allocation module turns trained policies into actionable cross-platform spend recommendations. Experimental evaluation demonstrates improved click-through rate and budget utilisation compared with rule-based and statistical baselines. Future work will pursue multi-agent RL for competitive auction modelling, hierarchical RL for joint budget and bid decisions, and integration with live RTB exchanges for production validation.

7. References

[1] R. S. Sutton and A. G. Barto, “Reinforcement Learning: An Introduction,” 2nd ed., MIT Press, 2018.
[2] V. Mnih et al., “Human-level control through deep reinforcement learning,” Nature, vol. 518, no. 7540, 2015, pp. 529–533.
[3] J. Schulman et al., “Proximal policy optimization algorithms,” arXiv preprint arXiv:1707.06347, 2017.
[4] H. Cai et al., “Real-time bidding by reinforcement learning in display advertising,” in Proc. 10th ACM Int. Conf. Web Search and Data Mining (WSDM), 2017, pp. 661–670.
[5] W. Zhang, S. Yuan, and J. Wang, “Optimal real-time bidding for display advertising,” in Proc. 20th ACM SIGKDD Int. Conf. Knowledge Discovery and Data Mining (KDD), 2014, pp. 1077–1086.
[6] V. Konda and J. Tsitsiklis, “Actor-Critic algorithms,” in Advances in Neural Information Processing Systems (NeurIPS), vol. 12, 1999, pp. 1008–1014.
[7] A. Raffin et al., “Stable-Baselines3: Reliable reinforcement learning implementations,” Journal of Machine Learning Research, vol. 22, no. 268, 2021, pp. 1–8.