Proceedings of the 6th International Conference on Computing and Data Science
DOI: 10.54254/2755-2721/64/20241354

Application and comparative analysis of adaptive strategies in multi-armed bandit algorithms

Jin Zhou

School of Advanced Technology, Xi'an Jiaotong Liverpool University, 215028, China

Jin.Zhou22@student.xjtlu.edu.cn

Abstract. This study explores the application and comparative analysis of adaptive strategies in multi-armed bandit algorithms, specifically focusing on the ε-greedy algorithm, the Upper Confidence Bound (UCB) algorithm, and the Thompson sampling algorithm. By designing and implementing a series of experiments, the research identifies Thompson sampling as the most effective method, despite its greater reward fluctuation, highlighting its superior adaptability in uncertain environments. The comparative analysis reveals that each algorithm possesses distinct advantages and drawbacks, necessitating a strategic selection based on the specific context of the application. This paper emphasizes the importance of adaptive strategies in optimizing decision-making processes in stochastic settings and underscores the need for further exploration into more sophisticated and optimized adaptive strategies to enhance algorithmic performance and efficiency. Through a meticulous examination of these algorithms, the research contributes valuable insights into the dynamic field of machine learning, offering a foundation for future advancements in adaptive strategy optimization.

Keywords: Multi-armed bandit, Adaptive strategies, Exploration and exploitation, Algorithm performance comparison.

1. Introduction

Navigating the complexities of uncertainty and rapid decision-making environments, the efficient allocation of resources to optimize long-term returns remains a central challenge within the fields of machine learning and operations research. The multi-armed bandit (MAB) problem [1] presents an ideal framework to simulate such scenarios, where each "arm" represents a potential action choice, each with its unique unknown reward distribution. Adaptive strategies, as a key instrument to address this challenge, are designed to maximize cumulative rewards by dynamically adjusting the choice logically, balancing the acts of exploring unknown "arms" and exploiting known information.

This paper begins by revisiting the foundational concepts and classical algorithms of the multi-armed bandit, including the ε-greedy algorithm, Upper Confidence Bound (UCB) algorithm [2], and Thompson sampling, among others. We further discuss the role of adaptive strategies in algorithmic performance, especially in diverse reward environments, such as static reward distributions and dynamically changing contexts. Additionally, we introduce several novel adaptive strategies and demonstrate their application and performance in a variety of test environments through a series of theoretical analyses and experimental comparisons [3].

DOI: 10.54254/2755-2721/64/20241354

In practical applications, ranging from online advertising placements to the allocation of resources in clinical trials, for example, application in breast cancer screening [4], multi-armed bandit algorithms have proven to be a powerful tool. Therefore, a deep understanding of the efficacy of different adaptive strategies is crucial for designing efficient decision-making systems. Through comparative analysis, this paper aims to provide clear guidance for researchers and practitioners in selecting or designing strategies that cater to specific application needs. In the following chapters, we will detail our methodology, experimental design, and the insights and conclusions obtained, all of which further affirm the effectiveness and necessity of adaptive strategies in solving the multi-armed bandit problem.

2. Research Background

2.1. Introduction to the Multi-Armed Bandit Problem

The multi-armed bandit problem [5] epitomizes the challenge of optimal decision-making in uncertain environments. It is akin to a gambler choosing which slot machine to play—a metaphor for any situation requiring a choice between multiple uncertain options. Success hinges on the strategy's ability to balance exploration of new possibilities against the exploitation of known profitable outcomes.

In practical terms, multi-armed bandit algorithms are pivotal in fields where choices must be continually assessed and adapted based on evolving data. From optimizing web content to personalizing medical treatments, these algorithms apply adaptive strategies, like epsilon-greedy, Upper Confidence Bound (UCB), and Thompson Sampling to incrementally improve decision-making and maximize long-term rewards [6].

Understanding and analyzing these strategies illuminates their effectiveness and guides the development of advanced systems capable of nuanced decision-making across various applications.

2.2. The Importance of Adaptive Strategies

In the landscape of algorithmic decision-making, adaptive strategies play a pivotal role. They navigate the tension between exploration, the act of seeking new avenues for reward, and exploitation, the practice of optimizing from known profitable options. This balance is not static; as conditions and available data change, so too must the approach. Adaptive strategies are the tools that algorithms use to recalibrate this balance in real-time.

Their importance cannot be overstated in rapidly changing environments such as financial markets or complex adaptive systems like the internet, where user behavior and preferences evolve continuously [7]. Here, the ability to adjust strategies based on incoming data can mean the difference between staying relevant and falling behind. In digital advertising, for instance, adaptive strategies ensure that the most effective advertisements reach the right audience, maximizing return on investment. In personalized medicine, they could mean adapting treatment plans to a patient's changing condition, potentially improving outcomes.

2.3. Research Motives and Objectives

The impetus for this research stems from the imperative to enhance decision-making capabilities in situations laden with uncertainty. With multi-armed bandit algorithms being integral to a spectrum of applications, from online recommendation systems to adaptive clinical trials, the adaptive strategies they employ necessitate a thorough examination. The research is propelled by a fundamental question: How can we tailor these strategies to optimize efficiency and effectiveness across diverse operational landscapes?

The primary objective of this study is to conduct an in-depth analysis and comparison of adaptive strategies within multi-armed bandit algorithms. The aim is to assess their performance, particularly how they negotiate the trade-off between exploration and exploitation in varying conditions. The research seeks to understand the implications of strategy adaptability, especially in terms of reaction to

DOI: 10.54254/2755-2721/64/20241354

fluctuating reward scenarios, and to examine the computational pragmatism of these strategies concerning resource demands and scalability. Additionally, the study endeavors to develop a cohesive framework that guides the strategic choice, aligning with particular environmental and contextual demands of applications. Through this comprehensive approach, the research aspires to inform future algorithmic innovation and identify avenues for advancing the efficacy of adaptive decision-making tools.

3. Fundamentals of Multi-Armed Bandit Algorithms

3.1. Background Knowledge

The Multi-Armed Bandit (MAB) algorithm [8] is a strategy to balance the exploration versus exploitation dilemma, commonly used for optimizing resource allocation in decision-making processes. This term is derived from casino slot machines, which feature multiple levers, each providing a certain reward when pulled. In the context of the multi-armed bandit problem, each lever represents a choice or action, with potentially uncertain rewards for each choice.

Basic Principles: (1) Exploration: Refers to attempting arms that have not been fully leveraged, to gather more information about these arms. Exploration helps the algorithm understand the reward distribution of each arm, thereby finding the best arm over the long term. (2) Exploitation: Involves choosing the known best arm to maximize immediate reward. Using existing information to choose the best option can improve rewards in the short term but might miss out on better choices.

The core challenge of the multi-armed bandit algorithm is to balance exploration and exploitation to maximize total rewards. If the algorithm is too biased towards exploitation, it may get trapped in local optima due to lack of exploration [9]; conversely, if it leans too much towards exploration, it might miss out on gaining more rewards due to not sufficiently utilizing known information.

3.2. Algorithm Implementation

The multi-armed bandit problem is a decision-making challenge where you have to choose among multiple options, each with an unknown probability distribution of rewards. The goal is to maximize the total reward. Here, this study introduces three common algorithms to solve this problem: the ε-Greedy algorithm, the Upper Confidence Bound (UCB) algorithm, and the Thompson Sampling algorithm.

Each of these algorithms has its unique approach to addressing the exploration-exploitation trade-off. The ε-Greedy algorithm does this by occasionally choosing a random action, allowing for exploration of less-favored options. In addition, the UCB algorithm makes its selection based on a calculated confidence interval, favoring options that have either been less explored or have shown higher rewards. Thompson Sampling, a probabilistic approach, chooses an action based on the likelihood that a given action is optimal, which is computed from a statistical model of the rewards. This model is continuously updated with each action taken, making it highly adaptive and responsive to new information.

Algorithm 1 ε-Greedy Algorithm

TRAIN(X, ε)

Initialize:

N_arms ← number of actions in X

Counts ← array of zeros with length N_arms

DOI: 10.54254/2755-2721/64/20241354

```
    Values ← array of zeros with length N_arms
    for each round t = 1, 2, ..., T do
        if random() < ε then
            Action ← choose a random action from X
        else
            Action ← argmax(Values)
        end if
        Reward ← take action Action and observe reward
        Update Counts and Values for Action:
        Counts[Action] ← Counts[Action] + 1
        Values[Action] ← Values[Action] + (Reward−Values[Action])/Counts[Action]
    end for
    return Values
PREDICT(X)
    Action ← argmax(Values) # Choose the action with the highest estimated value
    return Action
```

Algorithm 2 Upper Confidence Bound (UCB) Algorithm

```
TRAIN(X, N, C)
Initialize:
    N_arms ← number of actions in X
    Counts ← array of zeros with length N_arms
    Values ← array of zeros with length N_arms
    TotalCounts ← 0
    for each round t = 1, 2, ..., T do
        for each arm i = 1 to N_arms do
            if Counts[i] == 0 then
                UCBValues[i] ← ∞
            else
                AverageReward ← Values[i]/Counts[i]
                Delta ← √(2 log(TotalCounts)/Counts[i])
                UCBValues[i] ← AverageReward + C * Delta
            end if
        end for
        ChosenArm ← argmax(UCBValues)
        Reward ← take action ChosenArm and observe reward
        Counts[ChosenArm] ← Counts[ChosenArm] + 1
        TotalCounts ← TotalCounts + 1
        Values[ChosenArm] ← Values[ChosenArm] + (Reward -
Values[ChosenArm])/Counts[ChosenArm]
    end for
    return Values, Counts
PREDICT(Values, Counts)
    ChosenArm ← argmax(Values/Counts)
    return ChosenArm
```

Algorithm 3 Thompson Sampling Algorithm

```
TRAIN(X, N)
```

DOI: 10.54254/2755-2721/64/20241354

```
    Initialize:
        Successes ← array of zeros with length N_arms
        Failures ← array of zeros with length N_arms
        for each round t = 1, 2, ..., T do
            for each arm i = 1 to N_arms do
                BetaSample[i] ← sample from Beta distribution with parameters Successes[i] + 1
and Failures[i] + 1
            end for
            ChosenArm ← argmax(BetaSample)
            Reward ← take action ChosenArm and observe reward
            if Reward is success then
                Successes[ChosenArm] ← Successes[ChosenArm] + 1
            else
                Failures[ChosenArm] ← Failures[ChosenArm] + 1
            end if
        end for
        return Successes, Failures
    PREDICT(Successes, Failures)
        BetaMeans ← compute mean of Beta distributions for each arm
        ChosenArm ← argmax(BetaMeans)
        return ChosenArm
```

The ε-Greedy algorithm [10] is a straightforward approach that selects the best-known option most of the time (precisely, a proportion of $1-\epsilon$) to exploit the current knowledge, and chooses a random option with a probability of ε for exploration. This algorithm is easy to implement but requires careful selection of the ε value to balance exploration and exploitation. Typically, the ε value is decreased over time to increase the rate of exploitation.

The UCB algorithm makes decisions based on the confidence interval of each option. For each option, it calculates an upper confidence bound that considers both the average reward and the uncertainty associated with that option [11]. The algorithm then selects the option with the highest upper confidence bound. The core advantage of the UCB algorithm is that it automatically balances exploration and exploitation, as options with high uncertainty (i.e., less explored) or high average rewards are naturally prioritized. With increasing trials, the algorithm gradually focuses on the optimal option while maintaining exploration of potentially better options.

Thompson Sampling is a probabilistic approach that builds a model for the reward distribution of each option and decides the next action based on these models [12]. At each trial, the algorithm draws a sample for each option and selects the option with the highest sample value. As more data accumulates, the models of reward distribution become more accurate, allowing Thompson Sampling to better identify the optimal option. Thompson Sampling is particularly suited for high-uncertainty situations because it dynamically adjusts the balance between exploration and exploitation by continuously updating the probabilistic models of rewards.

Summary:

(1) The ε-Greedy algorithm makes decisions through a fixed proportion of random exploration and exploitation, simple but requires adjustment of the ε value.

(2) The UCB algorithm automatically balances exploration and exploitation by selecting options based on the upper bound of their confidence intervals.

(3) Thompson Sampling dynamically adjusts decisions by updating the probabilistic models of rewards, especially suitable for environments with high uncertainty.

DOI: 10.54254/2755-2721/64/20241354

4. Applications of Adaptive Strategies in Multi-Armed Bandit Algorithms

In the context of multi-armed bandit problems, adaptive strategies are crucial, as the algorithm must continually adjust its behavior in an ever-changing environment to maximize rewards or meet other performance metrics. These strategies involve assessing the value of each action and updating it over time based on newly acquired information. Here are several applications of adaptive strategies in multi-armed bandit algorithms.

4.1. ϵ-Greedy Algorithm

The ϵ-greedy strategy makes an adaptive trade-off between exploration and exploitation, exploiting the currently known best option most of the time (choosing the arm with the highest estimated reward) and exploring (selecting an arm at random) with a probability of ϵ. This ϵ value can be adaptively adjusted over time, for instance, decreasing the frequency of exploration as the algorithm learns and less information is needed.

4.2. Upper Confidence Bound (UCB) Algorithm

The UCB strategy adaptively chooses the arm with the highest confidence interval, reflecting a balance between exploitation and exploration. It takes into account the average reward of the arm and the number of times the arm has been chosen. As an arm is chosen more frequently, our confidence in it increases, adaptively reducing the likelihood of its selection unless its performance proves to be worthwhile.

4.3. Thompson Sampling

Thompson Sampling uses a Bayesian approach to adaptively update the estimated distribution of rewards for each arm. Whenever an arm is chosen and a reward is returned, the parameters of its reward distribution are updated, enabling the algorithm to adapt to the latest reward information and adjust future action selections accordingly.

The common thread among these strategies is that they all use past actions and reward outcomes to update their estimates of arm values, thereby adaptively adjusting future selections. In this way, multi-armed bandit algorithms can continuously learn and improve their performance in uncertain environments.

5. Comparative Analysis of Adaptive Strategies

5.1. Comparative Study of Experimental Design

The experiment has been meticulously structured to conduct a comprehensive analysis of movie ratings data. It unfolds through a sequence of well-defined stages, each crafted to methodically address a distinct aspect of the data processing and analysis workflow. The initial phase involves the careful acquisition and curation of the relevant datasets, ensuring that the foundation for the analysis is robust and reliable. Following this, a detailed preparation process is undertaken, wherein the datasets are systematically structured into a format amenable to intricate analysis. This is succeeded by an integration phase, where disparate pieces of information are skillfully merged to create a unified source, setting the stage for a nuanced exploration of the ratings data. Subsequently, the data undergoes a transformation phase, designed to segregate and refine the information further, allowing for an insightful genre-based evaluation. The culmination of this process is the analytical phase, where statistical techniques are employed to distill the data into meaningful insights, culminating in a rich understanding of the movie ratings landscape.

(1) Data Loading: Two data sets are loaded; one containing movie information and another containing user ratings.

(2) Data Preparation: Data sets are loaded into Pandas DataFrames with specified columns and character encoding.

(3) Data Merging: The movie and ratings data are merged using “MovieID” as a key.

DOI: 10.54254/2755-2721/64/20241354

(4) Data Transformation: The "Genres" column is split and exploded to create separate rows for each genre.

(5) Analysis: Calculate the average rating for each genre by grouping the merged data by "Genres".

(6) Visualization or Further Analysis (Implied): Possible visualization and statistical analysis as suggested by imported libraries.

Below is the Python code used for the analysis:

Algorithm 4 Analysis of Movie Ratings by Genre

1: Import the pandas, numpy, scipy.stats, and matplotlib libraries
2: Load the movie data from ml-1m/movies.dat
3: Load the ratings data from ml-1m/ratings.dat
4: Merge the movie and ratings data on MovieID
5: Split the Genres field into separate rows for each genre
6: Calculate the average rating for each genre
7: Reset the index of the resulting dataframe = 0

Additionally, the python code sets up the simulation for an experiment related to the multi-armed bandit problem, where each arm represents a movie genre. Here's an elaboration of the code's role in the experimental design:

(1) Number of Rounds:

n rounds = 1000 establishes the number of trials the algorithm will perform, which simulates decision-making over a set number of iterations and provides substantial data for subsequent analysis.

(2) Number of Arms:

n arms = len(genre avg rating) calculates how many distinct options (movie genres) are present, which the algorithm will consider during the experiment.

(3) Rewards for Each Arm:

rewards = genre avg rating [Rating].values assigns the average rating of each genre as the reward, setting up the basis for the optimization process within the algorithm.

(4) Reward Normalization:

Normalizing rewards between 0 and 1 is crucial for certain algorithms like Thompson Sampling, which require a standardized scale for the reward distribution.

(5) Exploration Rate:

epsilon = 0.1 configures the exploration probability for the ε-Greedy algorithm, guiding the algorithm on how frequently it should explore less certain options versus exploiting the best-known option.

This setup is pivotal for running simulations that assess the performance of different bandit strategies in maximizing expected rewards based on the movie genres' ratings data.

Algorithm 5 Setup for Multi-Armed Bandit Experiment

1: Define the default number of rounds: n_rounds <- 10000
2: Compute the number of arms (genres): n_arms <- length of genre_avg_rating
3: Assign rewards for each genre: genre_avg_rating['Rating']
4: Normalize the rewards to a scale of 0 to 1 for Thompson Sampling:
    min_reward <- minimum of rewards
    max_reward <- maximum of rewards
    normalized_rewards <- (rewards - min_reward) / (max_reward - min_reward)
5: Calculate the maximum possible reward: max_possible_reward <- max_reward x n_rounds
6: Set the exploration rate for ε-Greedy: ε <- 0.1 ε=0

Algorithm 6 Epsilon-First Strategy for Multi-Armed Bandit

Require: eps, rounds, n_arms, rewards
1: n_explore <- int(eps x rounds)
2: n_exploit <- rounds - n_explore
3: arm_rewards <- array of zeros(n_arms)
4: arm_counts <- array of zeros(n_arms)

DOI: 10.54254/2755-2721/64/20241354

5: rewards_over_time <- empty list
6: for i = 1 to n_explore do
7: Select a random arm
8: Update rewards and counts for that arm
9: Record the reward over time
10: end for
11: for i = 1 to n_exploit do
12: Select the best arm based on current knowledge
13: Update the total reward and record the reward over time
14: end for
15: return total_reward, rewards_over_time =0

Algorithm 7 Upper Confidence Bound (UCB) Algorithm
1: Function UCB(rounds, n_arms, rewards)
2: Initialize array arm_reward to zeros of size n_arms
3: Initialize array arm_counts to zeros of size n_arms
4: Initialize total_reward to 0
5: Initialize list rewards_over_time to empty
6: for r in 1 to rounds do
7: if r < n_arms then
8: Select each arm once, set arm to r - 1
9: else
10: Calculate UCB values for each arm
11: Select arm with the highest UCB value
12: end if
13: Observe reward for selected arm
14: Update arm_rewards and arm_counts for selected arm
15: Increment total_reward by observed reward
16: Append observed reward to rewards_over_time list
17: end for
18: return total_reward, rewards_over_time =0

Algorithm 8 Thompson Sampling Algorithm
1: Function Thompson_Sampling(rounds, n_arms, normalized_rewards)
2: Initialize vectors alpha and beta with ones of size n_arms
3: Initialize total_reward to 0
4: Initialize list rewards_over_time to empty
5: for each iteration do
6: Sample from Beta distribution for each arm using alpha and beta
7: Choose arm with the highest sample
8: Update alpha and beta with the reward and 1 - reward
9: Scale the return to the original scale for total reward calculation
10: Update total_reward and append actual reward to rewards_over_time list
11: end for
12: return total_reward, rewards_over_time =0

5.2. Performance Comparison of Different Strategies

This section delves into the intricacies of the experimental design and methodology underpinning the triad of algorithms under scrutiny. This study embarks on a comparative analysis of the performance metrics across disparate strategies, subsequently distilling the empirical evidence into a comprehensive statistical discourse.

DOI: 10.54254/2755-2721/64/20241354

Figure 1. Rewards Over Time for Epsilon-First (ETC) Algorithm

As shown in Figure 1, the Epsilon-First algorithm exhibits reward fluctuations during the initial exploration phase, followed by a stable increase during the exploitation phase. The total accumulated reward of 4027.84 indicates that the algorithm effectively utilizes the known information to maximize rewards. A regret of 47.34 suggests that there is some loss in performance compared to the optimal strategy. The variance of the rewards being 0.0246 indicates a high consistency in the results.

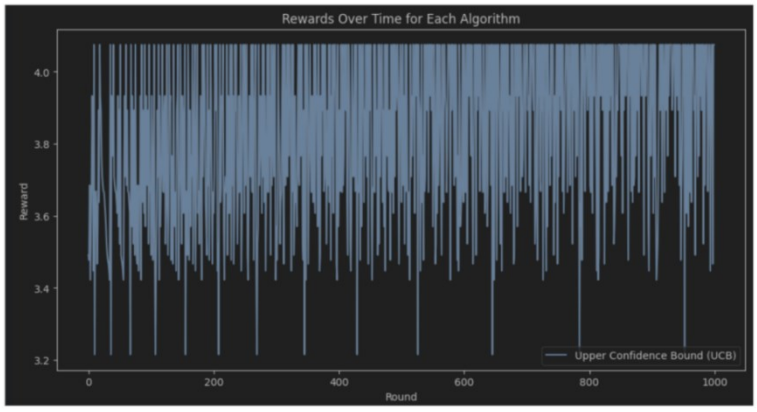

Figure 2. Rewards over Time for Upper Confidence Bound (UCB) Algorithm

As illustrated in Figure 2, the Upper Confidence Bound (UCB) algorithm demonstrates variability in rewards throughout the 1000 rounds, indicative of its inherent exploration-exploitation trade-off. The total accumulated reward of 3829.02 reflects the algorithm's proficiency in balancing between the exploration of new arms and the exploitation of known ones. A regret of 246.17 points to a substantial difference from the theoretically optimal performance, implying potential room for improvement in the algorithm's strategy. The variance of the rewards, at 0.0522, signals a moderate consistency in the obtained rewards, suggesting that while the algorithm is making informed decisions, the outcomes of those decisions still exhibit some variability.

DOI: 10.54254/2755-2721/64/20241354

Figure 3. Rewards over Time for Thompson Sampling (TS) Algorithm

As demonstrated in Figure 3, the Thompson Sampling algorithm exhibits an initial exploration phase with considerable variability in rewards, which stabilizes as the algorithm progressively learns and exploits the most rewarding arms.

The total accumulated reward of 4058.54 is indicative of the algorithm's effective exploitation of the arms based on the updated probabilistic models, hinting at a successful balance between exploration and exploitation. A regret of 16.65 implies a minimal performance gap compared to the optimal strategy, showcasing the efficiency of Thompson Sampling in maximizing the cumulative reward. Furthermore, the low variance of 0.00736 in the rewards reflects a high level of consistency in the outcomes, emphasizing the algorithm's ability to make reliable decisions over the course of 1000 rounds.

5.3. Experimental Results and Statistical Analysis

(1) Thompson Sampling algorithm yields the highest total accumulated rewards and lowest regret, suggesting it balances exploration and exploitation effectively but with higher variance, indicating a more explorative and less consistent performance.

(2) The UCB algorithm performs better than ETC in terms of rewards but has higher regret than Thompson Sampling, with moderate variance, indicating a balance between consistency and value.

(3) ETC has the lowest accumulated rewards and variance, which might point to a more conservative approach with less exploration after the initial phase.

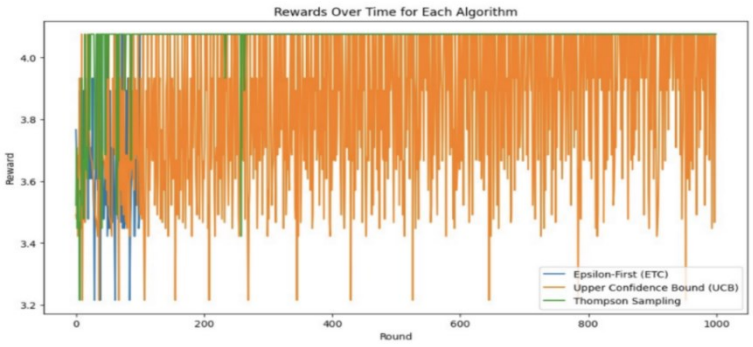

Figure 4. Rewards compared over time for three algorithms

Overall, Thompson Sampling appears to be the best performing algorithm in this specific scenario, albeit with greater fluctuation in rewards (Figure 4).

DOI: 10.54254/2755-2721/64/20241354

6. Conclusion

This study explores the application effectiveness of ε-Greedy, Upper Confidence Bound (UCB), and Thompson Sampling adaptive strategies in the multi-armed bandit problem. Through extensive simulation experiments, we have conducted an in-depth analysis of the performance of these strategies under various reward distributions. Here is a summary of the research findings, limitations, as well as future research directions for these strategies.

During the experiment, the ε-Greedy algorithm simplifies the trade-off between exploration and exploitation through a fixed proportion of random exploration but may lack flexibility in dynamic environments [13]. The UCB algorithm [14] balances exploration and exploitation by calculating confidence upper bounds and often achieves better performance, especially when the reward distributions are relatively stable. Thompson Sampling [15], on the other hand, uses probabilistic models to provide a richer information basis for decision-making, making it more adaptable in the face of uncertainty. Through the design and execution of various experiments, the study demonstrates that Thompson sampling outperforms other methods in effectiveness, despite experiencing higher variability in rewards. This outcome underscores its superior adaptability in scenarios marked by uncertainty, setting it apart as a particularly effective approach within the context of adaptive strategies.

Future studies should focus on the development of novel adaptive strategies by merging the strengths of existing algorithms. This approach aims at enhancing the ability to navigate diverse and dynamic reward landscapes more efficiently. Additionally, there is a need for an expanded theoretical analysis to offer more robust performance guarantees under a variety of conditions. Such investigations would enrich our understanding of algorithmic behavior and effectiveness. The practical application of these strategies in real-world scenarios, including online advertising, financial market analysis, and recommendation systems, should be pursued to tailor algorithm performance to the unique demands of these fields. Furthermore, exploring the integration of deep learning models with adaptive strategies presents a promising avenue for elevating decision-making quality in complex environments. This comprehensive approach will not only advance the theoretical framework but also significantly impact practical applications, paving the way for more sophisticated and adaptable solutions in the field of machine learning. Future research will continue to advance the field, expanding the boundaries of the application of multi-armed bandit algorithms across various domains.

References

[1] K. Sasaki, T. Mihana, K. Kanno, M. Naruse and A. Uchida, Experiment on Decision Making for Multi-Armed Bandit Problem Using Chaos and Low Frequency Fluctuations in Laser Network, 2022 Conference on Lasers and Electro-Optics Pacific Rim (CLEO-PR), Sapporo, Japan, 2022, pp. 1-2, doi: 10.1109/CLEO-PR62338.2022.10432044.

[2] S. Garbar, Invariant description of UCB strategy for multi-armed bandits for batch processing scenario, 2020 24th International Conference on Circuits, Systems, Communications and Computers (CSCC), Chania, Greece, 2020, pp. 75-78, doi: 10.1109/CSCC49995.2020.00021.

[3] X. Liu, M. Derakhshani, S. Lambotharan and M. van der Schaar, Risk-Aware Multi-Armed Bandits With Refined Upper Confidence Bounds, in IEEE Signal Processing Letters, vol. 28, pp. 269-273, 2021, doi:10.1109/LSP.2020.3047725.

[4] L. Song, W. Hsu, J. Xu and M. van der Schaar, Using Contextual Learning to Improve Diagnostic Accuracy: Application in Breast Cancer Screening, in IEEE Journal of Biomedical and Health Informatics, vol. 20, no. 3, pp. 902-914, May 2016, doi: 10.1109/JBHI.2015.2414934.

[5] N. Yoshida, T. Nishio, M. Morikura and K. Yamamoto, MAB-based Client Selection for Federated Learning with Uncertain Resources in Mobile Networks, 2020 IEEE Globecom Workshops (GC Wkshps, Taipei, Taiwan, 2020, pp. 1-6, doi: 10.1109/GCWkshps50303.2020.9367421.

[6] P. Landgren, V. Srivastava and N. E. Leonard, Distributed cooperative decision-making in multiarmed bandits: Frequentist and Bayesian algorithms, 2016 IEEE 55th Conference on Decision and Control (CDC), Las Vegas, NV, USA, 2016, pp. 167-172, doi: 10.1109/CDC.2016.7798264.

[7] D. Nguyen, L. Howard, H. Yan and F. Lin, Adaptive Sequence Learning: Contextual Multi-armed Bandit Approach, 2023 IEEE Intl Conf on Dependable, Autonomic and Secure Computing, Intl Conf on Pervasive Intelligence and Computing, Intl Conf on Cloud and Big Data Computing, Intl Conf on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Abu Dhabi, United Arab Emirates, 2023, pp. 0751-0756. doi: 10.1109/DASC/PiCom/CBD-Com/Cy59711.2023.10361498.

[8] E. N. Mambou and I. Woungang, Bandit Algorithms Applied in Online Advertisement to Evaluate Click-Through Rates, 2023 IEEE AFRICON, Nairobi, Kenya, 2023, pp. 1-5, doi: 10.1109/AFRICON55910.2023.10293356.

[9] X. Yu and W. Ouyang, Mean-Variance Pure Exploration of Bandits, 2023 6th International Conference on Artificial Intelligence and Big Data (ICAIBD), Chengdu, China, 2023, pp. 165-170, doi: 10.1109/I-CAIBD57115.2023.10206292.

[10] G. K. Shojaee and H. R. Mashhadi, Optimistic initial value analysis in a greedy selection approach to MAB problems, 2017 7th International Conference on Computer and Knowledge Engineering (ICCKE), Mashhad, Iran, 2017, pp. 419-424, doi: 10.1109/ICCKE.2017.8167915.

[11] W. B. Sghaier, H. Gassara, F. Rouissi and F. Tlili, Adaptive UCB MAB Algorithm forOptimizing Relay Selection in Narrowband Power Line Communication, 2023 IEEE Tenth International Conference on Communications and Networking (ComNet), Hammamet, Tunisia, 2023, pp. 1-6, doi: 10.1109/ComNet60156.2023.10366719.

[12] S. Ye and S. Wang, An Improved Thompson Sampling Method for Dynamic Spectrum Access in Non-Stationary Environments, 2023 IEEE Applied Sensing Conference (APSCON), Bengaluru, India, 2023, pp. 1-3, doi: 10.1109/APSCON56343.2023.10101010.

[13] S. V. S. Santosh and S. J. Darak (2024) Multiarmed Bandit Algorithms on Zynq System-on-Chip: Go Frequentist or Bayesian?, in IEEE Transactions on Neural Networks and Learning Systems, vol. 35, no. 2, pp.2602-2615, doi: 10.1109/TNNLS.2022.3190509.

[14] Y. Shi and Y. Mei, Efficient Sequential UCB-based Hungarian Algorithm for Assignment Problems, 2022 58th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, USA, 2022, pp. 1-8. doi: 10.1109/Allerton49937.2022.9929380.

[15] Z. Zhou and B. Hajek, Improving Particle Thompson Sampling through Regenerative Particles, 2023 57th Annual Conference on Information Sciences and Systems (CISS), Baltimore, MD, USA, 2023, pp. 1-4. doi:10.1109/CISS56502.2023.10089647.