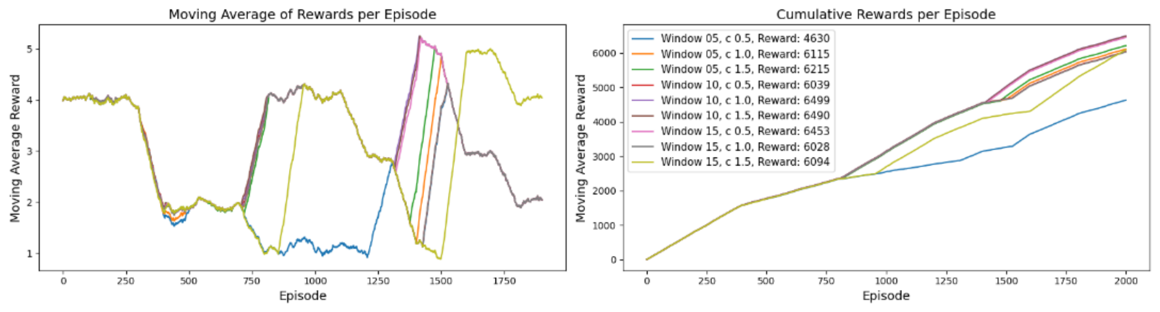
By Shania Muganga and Sara Ahmadi

Part I: Multi-Armed Bandits:

In this report we will tackle the problem of selecting which medicine to choose whilst considering changing effectiveness over time. The problem is like the Multi-Armed Bandit (MAB) scenario because the agent must choose between multiple options, in this case medicines, without state transitions and the rewards are independent of previous actions. The challenge with this task lies with the changing effectiveness of medicine over time meaning non-stationary rewards were the reward distribution randomly changes so the agent must adapt over time. This makes the exploit-exploration trade-off harder because finding the right balance with the trade-off is harder when the past becomes outdated quickly.

To handle our scenario where the effectiveness of medicine can change unpredictably, we developed a MAB algorithm using Sliding Window with Upper Confidence Bound (UCB). This method maintains a window of recent rewards for each arm, ensuring that the most relevant data is used for decision-making. Within this window, UCB values are computed to balance exploration and exploitation, and actions are taken based on these values. The window size and the exploration parameter 'c' balance using recent rewards and exploring new options. In our implementation, we test various window sizes and 'c' values across 2000 episodes. By adjusting these parameters, the algorithm can effectively respond to non-stationary rewards, aiming to minimize regret and maximize total rewards over time.

Figure 1: Sliding Window UCB Agent Performance



The right graph (Figure 1) shows that some window sizes and exploration parameters perform better than others by adapting more effectively and selecting higher-reward arms. Notably, the combination of window size 10 and ‘c’ = 1.0 adapts best to the changing rewards, which shift approximately few hundred episodes; this is evident from the quick adaptation to changes seen in the right graph, and achieves the highest cumulative reward as shown in the left graph. In contrast, the combination of window size 5 and ‘c’ = 0.5 (blue line) struggles to adapt, often failing to choose the optimal arm when rewards change, evident at around 750 episodes where it picks a lower-rewarding arm.

Due to the unpredictability of medicine effectiveness and other factors, there is significant randomness, making it hard to find a single ideal window size and ‘c’ parameter. Therefore, we would have tested multiple runs with the same parameters in different environments to average out the rewards and reduce randomness if we had had time. To visualize these changes, we use the same random seed for each parameter setting, resulting in the exact same mean reward changes across all runs, as this consistency helps interpret the results from the plots.

Figure 2: Comparison of Sliding Window UCB with Other Bandit Algorithms

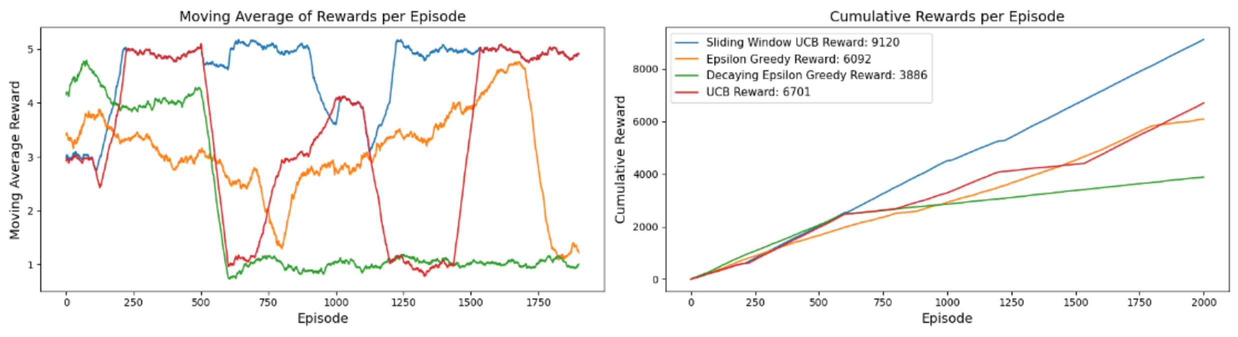


Figure 2 shows that the Sliding Window UCB algorithm achieves higher cumulative rewards of 9120 compared to UCB: 6701, Epsilon Greedy: 6092, and Decaying Epsilon Greedy: 3886. This is due to its focus on recent data, allowing for more relevant decision-making. Epsilon Greedy and Decaying Epsilon Greedy struggled due to their static exploration approaches, failing to adapt swiftly to changes. While UCB performed better than the Epsilon Greedy methods, it was still outperformed by Sliding Window UCB as it does not account for non-stationary environments where reward distributions change over time.

Overall, the Sliding Window UCB algorithm has great adaptive nature, in addition it uses recent reward information to update its confidence bounds, which enables it to respond dynamically to changes in the environment. This results in more consistent selection of optimal actions and higher cumulative rewards. The performance differences underscore the importance of using adaptive methods in environments with non-stationary reward distributions. By running these comparisons, we confirm that Sliding Window UCB is more effective in minimizing regret and maximizing rewards in dynamic scenarios, making it a superior choice over traditional MAB algorithms in such contexts.