International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V73I3P103

Volume 73 Issue 3, 25-31, March 2025

Original Article

Reinforcement Learning with LinUCB: Comparing Reward Designs and Optimizing Alpha for Warfarin Dosage

Rishi Nandan Simhadri[1], Shantanu Awasthi[2]

[1]*BASIS Ahwatukee High School, Tempe, USA.*
[2]*Missouri Southern State University, Joplin, USA.*

[1]*Corresponding Author : rishi.cg100@gmail.com*

Received: 08 January 2025 Revised: 12 February 2025 Accepted: 04 March 2025 Published: 19 March 2025

Abstract *- Determining the appropriate dose of warfarin is a significant challenge due to the numerous factors that contribute to the proper dose of the anticoagulant, and the consequences of taking an incorrect dose can contribute to adverse side effects and have serious health consequences for the patient. Commonly used approaches to determine the initial dose of warfarin are the pharmacogenetic algorithm, the clinical algorithm, and a fixed-dose approach. This research presents the application of reinforcement learning using the LinUCB algorithm to identify the optimal warfarin dose through three major experiments. First, the authors employed lasso regression for feature selection to identify the most relevant predictors of warfarin dosage in the warfarin dataset, ensuring a more interpretable model. Second, they evaluated various reward designs, including sparse, accuracy-focused dense, time decay, and distribution-based rewards, on several metrics such as accuracy, precision, recall, and f1 score. They discovered that accuracy-based dense reward was superior in predicting optimal doses in most metrics. Third, they improved the LinUCB algorithm's accuracy and f1 score by utilizing Hyperopt to identify the optimal value of hyperparameter alpha. Using data collected by the Pharmacogenetics and Pharmacogenomics Knowledge Base (PharmGKB), this research provides Reinforcement learning as a potential approach for determining warfarin doses. The final results of this study demonstrate the prospects of Reinforcement learning to improve current personalized medicine practices in Warfarin dosage. Representing an advancement in the application of Reinforcement learning within healthcare, this work provides other options for future research aimed at optimizing medication dosages to improve patient outcomes.*

Keywords *- Reinforcement Learning, Warfarin, LinUCB, Optimizing Alpha, HyperOpt.*

1. Introduction

The appropriate dose of warfarin, a widely prescribed anticoagulant, is challenging to determine accurately due to the numerous patient-specific factors involved. Incorrect dosage can lead to severe adverse effects such as haemorrhaging or thrombosis, highlighting the critical need for precise dosing methods. While conventional approaches such as the pharmaco genetic algorithm, clinical algorithm, and fixed-dose methods exist, they often do not sufficiently account for individual patient variability, leading to suboptimal dosage recommendations.

A significant gap remains in understanding how Reinforcement Learning (RL) techniques can effectively address the complexities inherent in personalized medicine, specifically regarding warfarin dosing. This research seeks to bridge this gap by applying the LinUCB algorithm, a contextual bandit approach, to optimize warfarin dosage. We systematically investigate different reward designs—including sparse, accuracy-focused dense, time decay, and distribution-based rewards—and identify the most effective approach through rigorous evaluation. Unlike previous studies, which often rely on static algorithms or limited reward mechanisms, this paper uniquely compares multiple reward designs within the LinUCB framework and further optimizes the model using HyperOpt. This comprehensive evaluation distinctly demonstrates the potential to advance the precision and applicability of reinforcement learning for individualized warfarin dosing.

Moreover, to further enhance the performance of the RL model, we explore hyperparameter optimization techniques, specifically using HyperOpt to identify the optimal alpha value. Utilizing a publicly available dataset from the Pharmacogenetics and Pharmacogenomics Knowledge Base (Phar- mGKB), our study evaluates RL as a robust methodology for personalized warfarin dosing. By addressing the existing research gap, our work demonstrates significant potential for RL techniques to improve dosing accuracy and patient out-comes in clinical settings.

2. Related Work

Since warfarin has held a long-standing position as the primary oral anticoagulant worldwide, there has been previous work on developing the optimal model for warfarin dosing.

2.1. Estimation of the Warfarin Dose with Clinical and Pharmacogenetic Data

The International Warfarin Pharmacogenetics Consortium published its work 'Estimation of the Warfarin Dose with Clinical and Pharmacogenetic Data' in 2009. Consortium (2009) focuses on incorporating the genetic variability of the patients to determine the optimal initial dose of warfarin. The researchers used data from more than 5000 patients to develop a least-squares linear regression algorithm that predicts warfarin dosage considering both genetic and clinical factors.

This pharmacogenetic model significantly surpassed the clinical model and the fixed-dose approach.

2.2. Warfarin: Almost 60 Years Old and Still Causing Problems

Pirmohamed (2006) covers the history of warfarin, the difficulties of accurately administering warfarin, and its potential use in pharmacogenetics. Despite its long-standing position as the primary oral anticoagulant globally, warfarin is associated with significant risks and can have serious consequences if the wrong dose is prescribed. The document also highlights difficulties in accurately dosing warfarin due to several factors such as genetics, diet, alcohol intake, and more. The paper discusses the significant potential for pharmacogenetics to improve Warfarin dosage and safety by identifying key genetic factors and introducing pharmacogenetics into other therapeutic fields.

2.3. Estimation of Warfarin Dosage with Reinforcement Learning

Pinilla et al. (2020) use Reinforcement learning to model the proper dose of warfarin for patients by implementing a LinUCB bandit approach, which outperformed the baselines (fixed model of 35 mg/week doses and linear model based on patient data). In addition to the LinUCB bandit, the paper also explores online supervised learning and reward reshaping to boost performance.

2.4. Research Gap and Contribution to Our Study

This research paper directly addresses these gaps with a two-fold approach: 1) by explicitly comparing multiple research designs—sparse, accuracy-based dense, time decay, and distribution-based reward to rigorously assess their effectiveness and 2) Furthermore, unlike Pinilla et al. (2020), the authors enhance the LinUCB model performance through systematic hyperparameter optimization using HyperOpt. Thus, this work distinctly contributes to the existing literature by combining comprehensive reward design application with targeted hyperparameter optimization to demonstrate the adaptability and precision of reinforcement learning-based warfarin dosing.

3. Methodology

3.1. LinUCB: Linear Upper Confidence Bound

LinUCB, as described by Li et al. (2010), is a linear upper confidence bound algorithm designed for solving contextual multi-armed bandit problems by incorporating contextual information for each decision. LinUCB assumes a linear relationship between the expected reward and the features of the context. For a reward $r_{t,a}$, context x_t, arm a, and weight vectors $\hat{\theta}_a$, we assume $E[r_{t,a}|x_t] = x_t\hat{\theta}_a$.

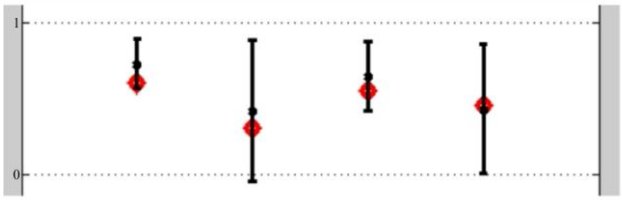


Fig. 1 LinUCB Algorithm: Confidence Interval Representation and Arm Selection. Adapted from Yoan (2019).

Figure 1 illustrates the LinUCB decision-making process by visually representing the confidence intervals for each arm at a given time. The arm selected is the one with the highest upper confidence bound, reflecting the principle of optimism in the face of uncertainty. This approach balances exploration of less certain arms and exploitation of known, high-performing arms.

For each patient, the algorithm calculates a confidence interval for each arm's expected reward. Then, it chooses the arm with the highest UCB value (estimated expected reward plus confidence bound). Context features were selected through lasso regression. The choice of α was chosen to be 0.84 by utilizing hyperparameter optimization to improve performance.

3.2. Feature Selection and Optimal Value of Al-pha

Lasso regression was employed to identify the most relevant features for predicting warfarin dosage. Alpha a was selected through hyper-parameter optimization to improve the model performance. More information about the implementation lasso regression and a optimization can be found in Sections
4.2 and 4.3 respectively.

3.3. Reward Designs

In reinforcement learning, the reward signal is responsible for determining the agent's behavior and, therefore, is a crucial element within the reinforcement learning paradigm Eschmann (2021). Essentially, a reward signal provides feedback to the algorithm, indicating the effectiveness of the actions taken in relation to the task's objectives.

Listed are the reward designs used, their definitions in the context of this methodology, and associated equations.

Sparse Reward: $R(a_t, a^*)$ returns a sparse reward of 1 if the chosen arm a_t (predicted dose) matches the true action a^* (required dose), and 0 otherwise. Sparse rewards provide clear binary signals of success or failure, which simplify learning but might slow down convergence due to limited feedback. Eschmann (2021)

$$R(a_t, a^*) = \begin{cases} 1 & \text{if } a_t = a^* \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Accuracy-Focused Dense Reward: R(at, a∗) returns a reward focused on improving accuracy. The reward is inversely proportional to the difference d(at, a∗) between the predicted and actual dose. (for example, the reward is higher for the smaller difference between the chosen arm and the true action). Eschmann (2021) and Arm Selection. Adapted from Yoan (2019).

$$d(a_t, a^*) = |a_t - a^*| \quad (2)$$

$$R(a_t, a^*) = \frac{D_{max} - d(a_t, a^*)}{D_{max}} \quad (3)$$

Time Decay Reward: $R(a_t, a^*, n)$ returns a reward based on whether the chosen arm a_t matches the actual action a^*, with the reward decreasing each trial n at a decay rate δ. Eschmann (2021).

$$R(a_t, a^*, n) = \begin{cases} \frac{r_0}{1+\delta \cdot n} & \text{if } a_t = a^* \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

Distribution-Based Reward: $R(a_t, a^*)$ returns a reward based on whether the chosen arm matches the actual action and frequency of dose category in the data. The reward is inversely proportional to the probability of the actual action $P(a^*)$ which is calculated as the count of actual required action C_{a^*}/N. The reward is normalized by multiplying a scaling factor S and the inverse of the maximum count of category counts C. Eschmann (2021)

$$R(a_t, a^*) = \begin{cases} \frac{1}{P(a^*)} \cdot \frac{S}{\max(C)} & \text{if } a_t = a^* \\ 0 & \text{otherwise} \end{cases} \quad (5)$$

3.4. Metrics

The effectiveness of the different reward designs with LinUCB was compared with several metrics. A single run means that the algorithm is run for one patient. To mitigate ordering bias, the data set was shuffled 20 times (resulting in mean metrics) that more accurately reflect the performance of the model by removing the possibility of ordering bias in the data set.

- Mean Accuracy: The ratio of correct prediction to total number of predictions at a point in each run. Specifically, the accuracy metrics were computed by comparing predictions from the LinUCB algorithm against the physician-guided, true optimal dosage from the PharmGKB dataset.
- Mean Precision: The ratio of true positive predictions to number of true positives + false positives (positive predictions)
- Mean Recall: Ratio of true positive predictions to number of true positives + false negatives in the data.
- Mean F1 score: Harmonic mean of precision and recall.

$$\frac{2 * precision * recall}{precision + recall}$$

4. Implementation

4.1. Data Preprocessing

The authors used a publicly available patient data set collected by Pharmacogenetics and Pharmacogenomics Knowledge Base (PharmGKB) for 5700 patients trained with warfarin from 21 research groups spanning 9 countries and 4 continents. There are 5528 patients with their true optimalpatient-specific Warfarin dose (initially unknown but determined through the physician-guided dose adjustment process). The dose for each patient is classified as:

1. Low: Less than 21 mg/week
2. Medium: 21 - 49 mg/week
3. High: More than 49 mg.week

To preprocess the aforementioned dataset, the authors dropped patients with no known therapeutic dosage from the dataset and additional processing described below resulted in valid data for 5406 patients.

Steps Followed:

1. Raw Dataset has 5700 patients and 66 attributes about the patient (including Patient ID).
2. Dropped patients with no therapeutic dose of warfarin and no stable dose of warfarin.
3. Dropped patients with no information on gender.
4. Imputed missing age, height, and weight with mean.
5. After imputing, age was binned into 10 groups.
6. Dropped the columns for Carbamazepine, Phenytoin, Rifampin, and Rifampicin after consolidating their values into the 'Enzyme Inducer Status' column, where 1 indicates the patient is taking any of these medications and 0 otherwise.
7. Imputed missing values for the VKORc1 SNP rs9923231 based on race and VKORC1 SNP data at rs2359612, rs9934438, or rs8050894. (see Section 4 in Supplementary Appendix 1 of Estimation of the Warfarin Dose with Clinical and Pharmacogenetic Data). If the VKORC1 genotype could not be imputed, it was treated as "unknown".
8. Dropped remaining VKORC1 genotype columns.
9. Dropped all Cyp2C9 columns except "Cyp2C9 genotypes"
10. Dropped columns that had more than 50% missing values.
11. Dropped following columns

- 'PharmGKB Subject ID'
- 'Estimated Target INR Range Based on Indication

- 'Subject Reached Stable of Warfarin'

12. Any missing cells imputed with 'unknown'
13. All the variables are converted into binary variables using one hot encoding.

Figure 2 shows a graph of the coefficients with important features.

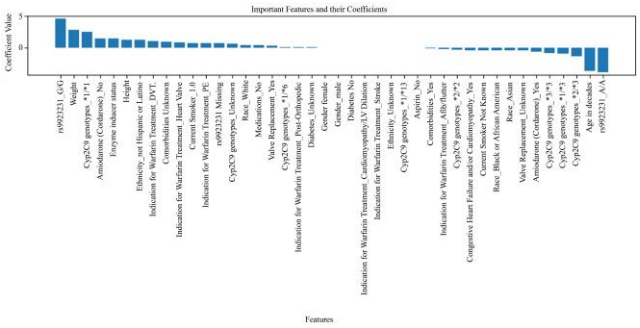


Fig. 2 Features and their Coefficients

4.2. Lasso Regression for Feature Selection

Lasso regression was employed to identify the most relevant features for predicting the optimal warfarin dosage. By applying Lasso, we effectively performed feature selection by shrinking the coefficients of less important variables to zero. We selected features with an absolute coefficient value greater than 1 to have the greatest predictive power for dosage classification while keeping low dimensionality.

It is important to remember that the dataset utilized from the Pharmacogenetics and Pharmacogenomics Knowledge Base (PharmGKB) could contain inherent biases due to demographic and clinical diversity among patients and missing values. Although extensive preprocessing, such as imputation and encoding (as done above), mitigates some biases, residual biases might persist, potentially affecting the generalizability of the results.

Feature selection using lasso regression significantly impacts model interpretability and performance. By selecting a subset of relevant predictors, we reduced model complexity, enhancing interpretability and reducing the risk of overfitting. However, this selection could inadvertently exclude clinically relevant but statistically insignificant features, which might influence model accuracy and patient outcomes. Therefore, carefully considering the trade-offs between model simplicity, interpretability, and predictive performance is critical.

From the results, we implemented the following feature set:

- Age in decades
- Gender
- Race
- Ethnicity
- Height in cm
- Weight in kg
- Enzyme Inducer Status
- Amiodarone
- Indication for Warfarin Treatment
- VKORC1 rs9923231 genotype
- Cyp2C9 genotypes

When implementing LinUCB (Section **??**), we classified the dosage based on Section**??**. Below is the dosage classification and respective counts:

- Low (0): 1146
- Medium (1): 3321
- High (2): 639

4.3. Optimal Value of Alpha

In LinUCB, the parameter α influences the exploration versus exploitation rate Fangwei (2020). In the context of multi-armed bandit problems, exploration involves selecting arms with relatively high uncertainty regarding their potential rewards (arms that have not been chosen as often in the past), while exploitation means choosing arms with high expected rewards. Note that α differs from ϵ (range from 0 to 1) used in the Epsilon Greedy Algorithm. ϵ directly determines the rate of exploration-exploitation (e.g. ϵ = 0.6 means the agent explores 60% of the time and exploits its knowledge 40%). However, α can take on any positive value and, differently, heavily influences the exploration-exploitation rate in LinUCB.

Specifically, in LinUCB, α influences the size of the confidence interval; that is, a larger α value expands the confidence interval, encouraging exploration, while lowering α narrows it, placing more weight on exploitation. The choice of α is crucial as it affects the performance of the LinUCB model. We experimented with other values of α ranging from 0 to 5 for our LinUCB model using the top-performing Accuracy Based Dense Reward (Section 4.4). The best α value found is 1.54, with a corresponding accuracy score of 0.663 and an F1 score of 0.567 (Accuracy Based Dense Reward).

4.4. Procedure for Determining Optimal Alpha

4.4.1. Objective Function

Several optimization algorithms can be used to determine the value of alpha. In this case, the goal is to maximize accuracy and F1 score (ignored precision and recall since F1 Score is a harmonic mean of both of these metrics). We used HyperOpt Bergstra et al. (2013) as the optimization interface for reasons mentioned in section 4.4.7.

The way to use Hyperopt is to describe:

- the objective function is to minimize
- the space over which to search
- a trial database
- the search algorithm to use

The objective function is a run of an experiment with some arbitrary α as its parameter, resulting in accuracy and F1 score, and returns the negative average of these two metrics. We return the combined metric as a negative

average of the metrics because HyperOpt minimizes the objective function. Returning the negative allows for the maximization of accuracy and F1 score.

$$\text{Combined Metric} = -\frac{\text{Accuracy} + \text{F1 Score}}{2}$$

4.4.2. Configuration Space

The configuration space object describes the domain over which HyperOpt is allowed to search. In the experiments in section 5, we used α as 1.0; however, α can take any positive value. Hence, the search space for α is defined as a uniform distribution between 0 and 5, so HyperOpt selects α values within this range during the optimization process.

4.4.3. Search Algorithm

The Tree-of-Parzen-Estimators (TPE) algorithm is chosen for the search through the `algo=tpe.suggest` parameter in `fmin`. TPE is a single objective Bayesian optimization technique that efficiently finds the best hyperparameters by building a probability model of the objective function.

4.4.4. Trials Database

A `Trials` object is created to track all the optimization attempts. This object stores detailed information about each trial, including the hyperparameters tested and the resulting objective function value.

4.4.5. Execution Process

The `fmin` function from HyperOpt is used to conduct the optimization. It is configured with the objective function, the defined configuration space, the TPE algorithm as the search algorithm, a maximum of 50 evaluations, and the `Trials` object to track the optimization process. The `fmin` function then executes the optimization loop, selecting a hyperparameter, evaluating the objective function, and updating its model of the search space based on the results.

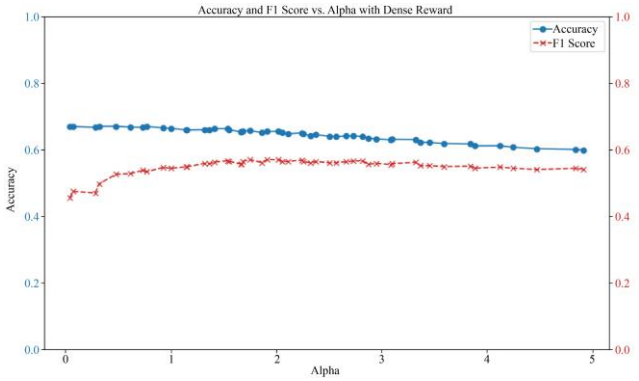


Fig. 3 Accuracy and F1 Score vs. Alpha

4.4.6. Result

After the optimization process completes, the `fmin` function returns the best α value found, which is printed out. This value represents the α that maximizes the accuracy and F1 score of the experiment. The best α value found is 2.55, with a corresponding accuracy score of 0.607 and an F1 score of 0.482. Which are not significantly different from results using an α of 1.0 (Accuracy Based Dense Reward).

4.4.6. Additional Info

Determining the optimal value of α can be done in various other ways (e.g. Genetic Algorithms, Multi-Objective Optimization, Single Objective Optimization). Hyperopt provides algorithms and software infrastructure for carrying out hyperparameter optimization for machine learning algorithms.

In this study, a single objective Bayesian optimization algorithm was chosen called Tree-of-ParzenEstimators (TPE) for its simplicity. TPE randomly tests α and its combined metric score. The α space was defined as uniformly distributed from 0 to 5, and 50 trials were conducted. To generate Figure 3, the results were sorted in ascending order of α values, and the corresponding accuracy and F1 score values were plotted.

5. Results

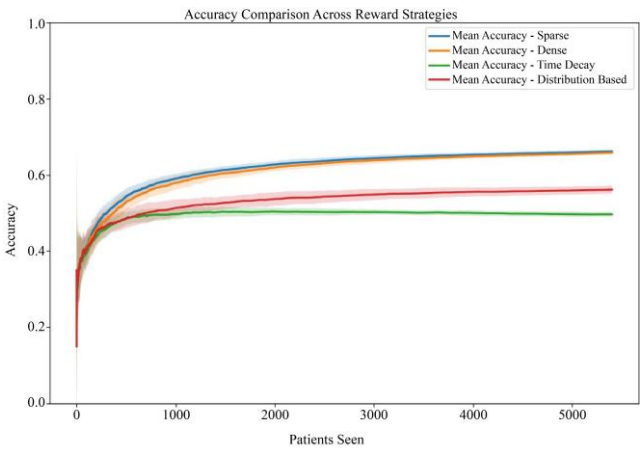
With the methods implemented as described, the next section details the experimental outcomes, clearly demonstrating the effectiveness of each reward design and optimized parameters.

5.1. Metrics Performance

Figure 4 shows the mean accuracy across all the reward strategies. The sparse and accuracy-based dense reward strategies perform similarly, achieving 66% by 5405 patients. The time decay and distribution-based reward strategies do not perform as well, achieving 50% and 56% mean accuracy, respectively.

In Figure 5, it is observed that the sparse reward and accuracy-based dense reward perform the best at 0.58%.

Figure 6 presents the recall performance of the reward strategies. From 1000 patients onward, a clear ranking of the reward strategies is evident in the following order: distribution-based, accuracy-based, dense, sparse, and time decay. Unlike its previous results, the distribution-based reward strategy performed the best at 0.64%.



Patients Seen	Sparse	Dense	Time Decay	Distribution Based
1000	0.59 (1)	0.58 (2)	0.50 (4)	0.51 (3)
2000	0.63 (1)	0.62 (2)	0.50 (4)	0.54 (3)
3000	0.64 (1)	0.64 (2)	0.50 (4)	0.55 (3)
4000	0.65 (1)	0.65 (2)	0.50 (4)	0.56 (3)
5405	0.66 (1)	0.66 (2)	0.50 (4)	0.56 (3)

Fig. 4 Accuracy comparison across reward strategies

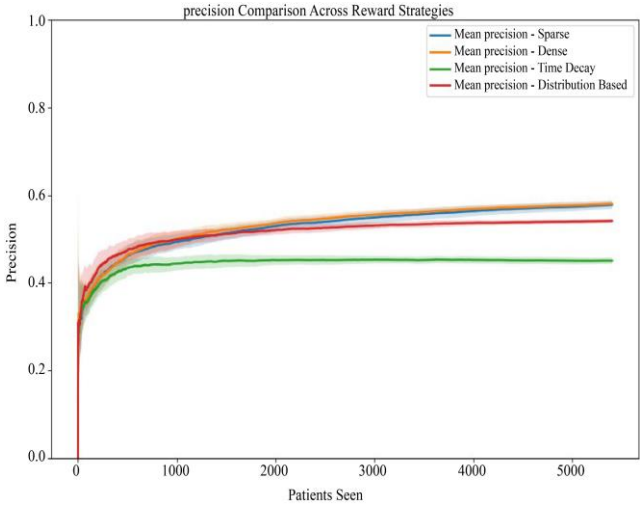
Figure 7 illustrates the comparison of the F1 score across the reward strategies. The F1 score considers both recall and precision. The accuracy-based dense reward performs the best at 0.56% F1 score, followed by the distribution-based, sparse, and time decay reward strategies.

The accuracy-based dense reward methodology consistently outperformed the competing reward designs across multiple evaluation metrics, with the exception of the recall metric, which ranked second. The consistent and high-level performance highlights the superiority of the accuracy-based dense reward in this case study.

6. Conclusion

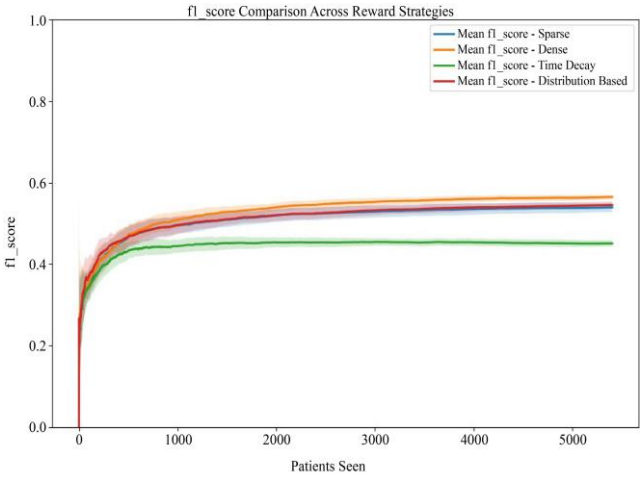
The study compares various reward designs for optimizing the initial Warfarin dose using LinUCB for training, hyper-parameter optimization, and lasso regression for feature selection. The results of the experiments demonstrate the higher performance of accuracy-based dense rewards, which increases the learning speed and accuracy of LinUCB. Among all the experiments, dense rewards seem to be the better option. From the HyperOpt experiments, the best value for al- pha is 1.54 (accuracy: 0.663 and f1 score: 0.567).

We achieved better results primarily because the accuracy-based dense reward consistently guided the model toward correct dosage predictions by providing clear and immediate feedback. Compared to other reward designs, the accuracy-based dense reward helped the model learn faster from each patient, improving accuracy. Optimizing the alpha value further enhanced these results by effectively balancing exploration and certainty in predictions.



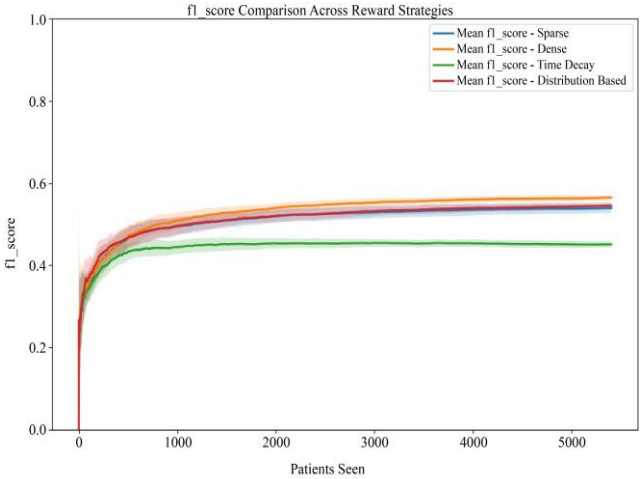
Patients Seen	Sparse	Dense	Time Decay	Distribution Based
1000	0.49 (3)	0.50 (1)	0.44 (4)	0.50 (2)
2000	0.53 (2)	0.54 (1)	0.45 (4)	0.52 (3)
3000	0.55 (2)	0.56 (1)	0.45 (4)	0.53 (3)
4000	0.56 (2)	0.57 (1)	0.45 (4)	0.54 (3)
5405	0.58 (2)	0.58 (1)	0.45 (4)	0.54 (3)

Fig. 5 Precision comparison across reward strategies



Patients Seen	Sparse	Dense	Time Decay	Distribution Based
1000	0.50 (3)	0.53 (2)	0.48 (4)	0.59 (1)
2000	0.51 (3)	0.54 (2)	0.49 (4)	0.62 (1)
3000	0.52 (3)	0.55 (2)	0.49 (4)	0.63 (1)
4000	0.52 (3)	0.55 (2)	0.49 (4)	0.64 (1)
5405	0.52 (3)	0.55 (2)	0.49 (4)	0.64 (1)

Fig. 6 Recall comparison across reward strategies



Patients Seen	Sparse	Dense	Time Decay	Distribution Based
1000	0.50 (3)	0.51 (1)	0.44 (4)	0.50 (2)
2000	0.52 (3)	0.54 (1)	0.45 (4)	0.52 (2)
3000	0.53 (3)	0.55 (1)	0.45 (4)	0.53 (2)
4000	0.54 (3)	0.56 (1)	0.45 (4)	0.54 (2)
5405	0.54 (3)	0.56 (1)	0.45 (4)	0.55 (2)

Fig. 7 F1 Score comparison across reward strategies

6.1. Future Work

Future research can aim to broaden the scope and build on the findings of this study.

- The research paper primarily focused on the LinUCB algorithm, while other Reinforcement Learning algorithms can be used, such as online supervised learning for contextual bandits, as shown in Pinilla et al. (2020).
- There is a need for exploration of the optimization of other hyperparameters to improve the performance of the Lin- UCB model. Identifying the proper adjustments to these parameters may boost performance.
- Improving the quality of the dataset could significantly enhance performance. Using a dataset that documents the patient-specific changes in dosages over time until the optimal dosage is prescribed can enable the application of algorithms such as value iteration or policy iteration.

References

[1] James Bergstra, Dan Yamins, and David D. Cox, "Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms," *Proceedings of the 12th Python in Science Conference*, pp. 1-8, 2013. [CrossRef] [Google Scholar] [Publisher Link]

[2] The International Warfarin Pharmacogenetics Consortium, "Estimation of the Warfarin Dose with Clinical and Pharmacogenetic Data," *New England Journal of Medicine*, vol. 360, no. 8, pp. 753-764, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[3] Jonas Eschmann, *Reward Function Design in Reinforcement Learning*, Reinforcement Learning Algorithms: Analysis and Applications, vol. 883, pp. 25-33, 2021. [CrossRef] [Google Scholar] [Publisher Link]

[4] Contextual Bandits Analysis of Linucb Disjoint Algorithm with Dataset, Kenneth Foo Fangwei, 2020. [Online]. Available: https://kfoofw.github.io/contextual-bandits-linear-ucb-disjoint/

[5] Lihong Li et al., "A Contextual-Bandit Approach to Personalized News Article Recommendation," *Proceedings of the 19th International Conference on World Wide Web*, Raleigh North Carolina, USA, pp. 661-670, 2010. [CrossRef] [Google Scholar] [Publisher Link]

[6] Arpita Vats, "Estimation of Warfarin Dosage with Reinforcement Learning," *arXiv*, pp. 1-7, 2020. [CrossRef] [Google Scholar] [Publisher Link]

[7] Munir Pirmohamed, "Warfarin: Almost 60 Years Old and Still Causing Problems," *British Journal of Clinical Pharmacology*, vol. 62, no. 5, pp. 509-511, 2006. [CrossRef] [Google Scholar] [Publisher Link]

[8] Yoan Russac, *Introduction to Linear Bandits*, pp. 1-38, 2019. [Google Scholar] [Publisher Link]