Loan Approval and Credit Risk Assessment Using Reinforcement Learning Models

Fortune Raphael C. Bermudez
College of Computing and Information Technologies
National University
Manila, Philippines
ermudezfc@students.national-u.edu.ph

Joyce Anne D. Colocado
College of Computing and Information Technologies
National University
Manila, Philippines
colocadojd@national-u.edu.ph

Allen M. Siaton
College of Computing and Information Technologies
National University
Manila, Philippines
siatonam@national-u.edu.ph

Abstract—This study explores the application of Reinforcement Learning (RL) for automating loan approval and credit risk assessment, addressing the limitations of traditional rule-based and supervised learning models in adapting to evolving borrower behavior and market dynamics. A custom simulation environment was developed to model financial decision-making, featuring a reward function that balances profit maximization and default risk minimization. Two RL algorithms Tabular Q-Learning and Deep Q-Network (DQN) were implemented and compared using a synthetic dataset of 10,000 loan applications. Experimental results show that the DQN agent outperformed Q-Learning and baseline policies in terms of cumulative profit and total reward, demonstrating superior capability in learning profitable approval strategies despite moderate default rates. The findings highlight the potential of RL for adaptive financial decision-making, with implications for fairness, transparency, and responsible AI deployment in high-risk domains such as credit scoring.

Index Terms—Reinforcement Learning, Deep Q-Network (DQN), Q-Learning, Credit Risk Assessment, Loan Approval, Financial Decision-Making, Machine Learning, Proximal Policy Optimization, Credit Scoring, Imbalanced Data, Reward Function Design, Model Evaluation, Artificial Intelligence in Finance

I. INTRODUCTION

Loan approval and credit risk assessment are central to lending, yet common methods like logistic regression and rule-based scorecards often fail to adjust when borrower behavior or market conditions change [1]. These models lean on fixed historical patterns and simple relationships, which limits how well they capture complex risk signals [1]. As lending becomes more digital and data-rich, lenders need decision systems that can learn and adapt rather than stay static [2].

This study explores reinforcement learning (RL) for automated loan approval and credit risk control. Unlike per-application classifiers, RL treats lending as a sequence of decisions where each approval affects portfolio outcomes [3]. This approach is particularly practical in finance because it can learn policies from historical datasets [3], [4].

The problem is also social and economic: better and fairer credit decisions can widen access while keeping defaults in check [5]. Small, well-managed gains in model performance can expand approvals with little change in delinquency in some markets [6]. But defaults are rare, so class imbalance makes detection and explanation harder [7]. Outcomes are delayed over the loan life, which

makes it hard to link each decision to its true impact [8].

Existing statistical and supervised models often chase short-term accuracy and can degrade when policies or data distributions shift [1]. Human decisions, while flexible, vary across branches and are hard to scale in high-volume settings [9]. These gaps motivate the use of RL to learn adaptive policies that optimize performance under uncertainty [3].

This research aims to design and evaluate an RL framework that learns approval and limit policies to improve portfolio value while keeping risk and fairness within set bounds [10]. We compare value-based methods (Q-learning/DQN) and policy-gradient families to see which work best for this task [10], [11]. We also align with current rules that classify credit scoring as a high-risk AI use case, which raises the bar for transparency and testing [12].

The hypothesis is that RL can uncover useful risk patterns and adapt policies to changing conditions, achieving better profit at equal or lower default rates than standard methods [10]. The planned contributions include a lending reward that mixes income and expected loss, an RL pipeline with appropriate evaluation, and an interpretable readout of policy trade-offs [8].

To conclude, we frame credit decisioning as a dynamic optimization problem and show how RL can be applied responsibly in lending [3].

II. LITERATURE REVIEW

Recent studies treat loan approval and credit risk assessment as a learning process where an agent learns to approve, reject, or adjust a loan to maximize business profit instead of using fixed rules or static classification models [13]. Reinforcement learning (RL) allows systems to keep learning from new borrower behavior and changing market conditions [14]. In one study, a deep Q-network with balanced stratified prioritized experience replay (DQN-BSPER) improved credit scoring on peer-to-peer lending by handling imbalanced data and learning from more informative samples [15]. Another work applied RL to credit-limit adjustments, where the agent learned how to change limits dynamically to balance potential profit and risk in daily operations [16]. A companion preprint showed that Q-learning can help formalize profit-maximizing strategies for credit-limit management in banks [17]. Newer research also proposes cost-sensitive reinforcement learning for credit risk, so the model directly accounts for the higher cost of defaults compared to the smaller cost of missed opportunities [18].

Despite these advances, researchers note key weaknesses. A main issue is selective-labels bias, because lenders only observe outcomes for approved borrowers while outcomes for rejected applicants remain unknown [19]. To address this, a framework using acceptance loops helps estimate the missing outcomes of rejected applicants and supports fairer offline evaluation of approval policies [20]. Credit data are often imbalanced and unstable, so improved experience-replay strategies have been proposed to stabilize value-based agents like DQN during training [21]. Many lending decisions are also single-step (approve or reject now), which fits well with contextual-bandit formulations that focus on choosing the best action from one observed applicant context [22]. Adding simple causal side information can further make bandit policies more data-efficient by using cause-and-effect signals already present in finance datasets [23].

Some RL studies in finance focus on offline RL, where models learn from historical logs rather than live experiments, which is safer for financial institutions [24]. Work in related risk tasks such as fraud detection shows that DQN can perform robustly under heavy class imbalance, supporting its use in credit-risk settings where defaults are relatively rare [25]. Surveys comparing supervised learning with RL emphasize that while supervised models can be accurate, RL brings adaptability and

direct optimization of business outcomes when rewards are well designed and interpreted carefully for deployment [26].

Our study builds on these findings by comparing Q-Learning and DQN in the same one-step online environment that mirrors per-applicant lending decisions without portfolio carry-over [22]. We define a reward function that gives profit for on-time repayment, strong penalties for default, and a small opportunity cost for rejecting a good borrower, following cost-sensitive RL practices [18]. This setup matches how lenders weigh risks and returns in real decisions and aligns with work showing that profit-weighted or confusion-matrix-aware rewards can improve learning for credit scoring agents [13]. Evidence from nearby financial applications that DQN handles class imbalance and shifting data well also supports our head-to-head comparison with Q-Learning for loan approvals [25].

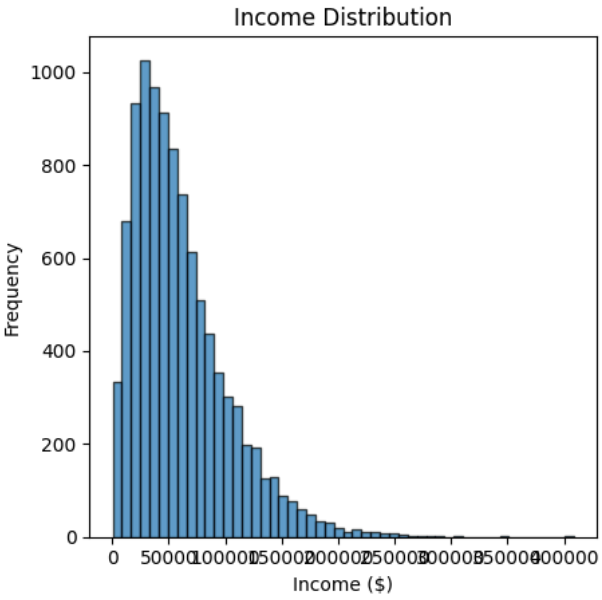
III. METHODOLOGY

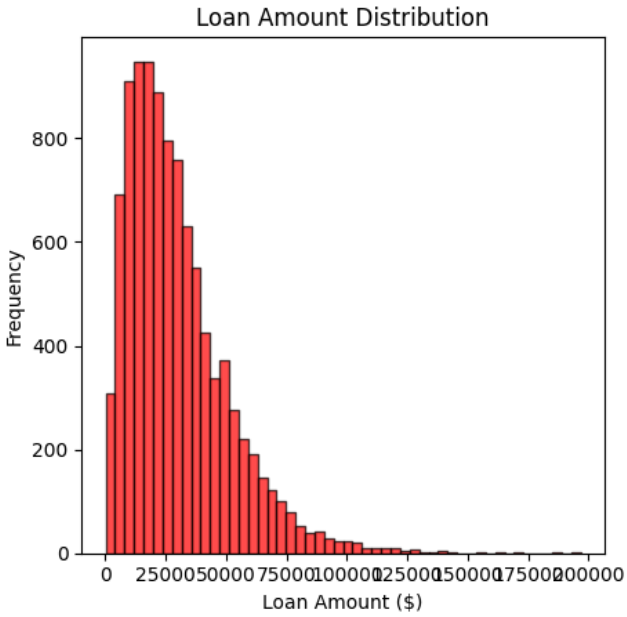
1) Environment and Problem Formulation

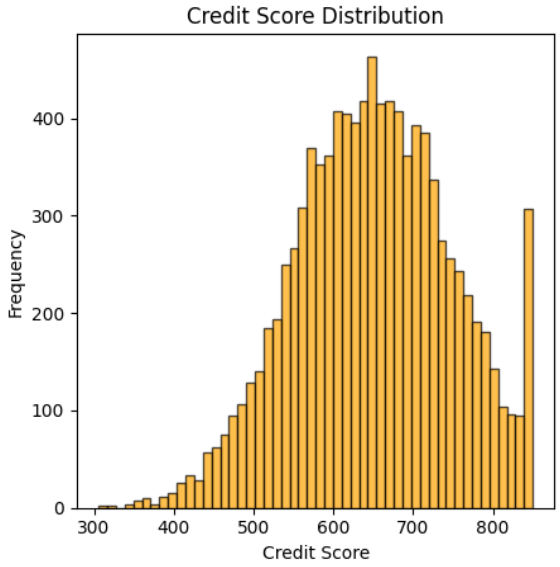
A custom loan approval environment was developed to simulate a financial institution's decision-making process. The environment is not based on OpenAI Gym but follows similar design principles for compatibility with RL algorithms.

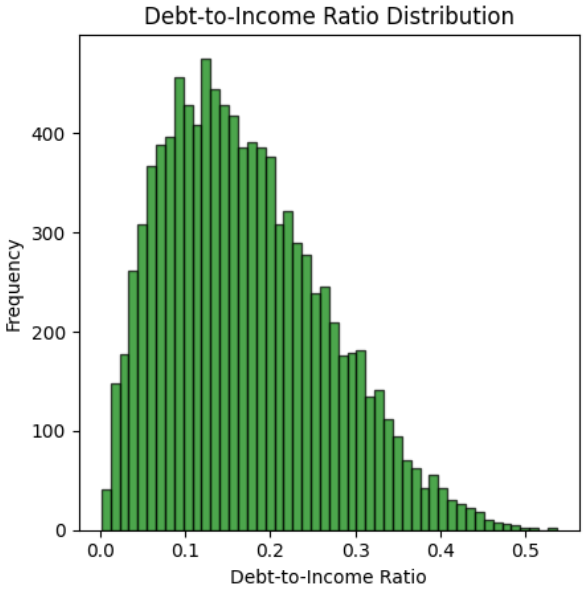
Environment Class: LoanApprovalEnvironment
Type: Custom simulation environment

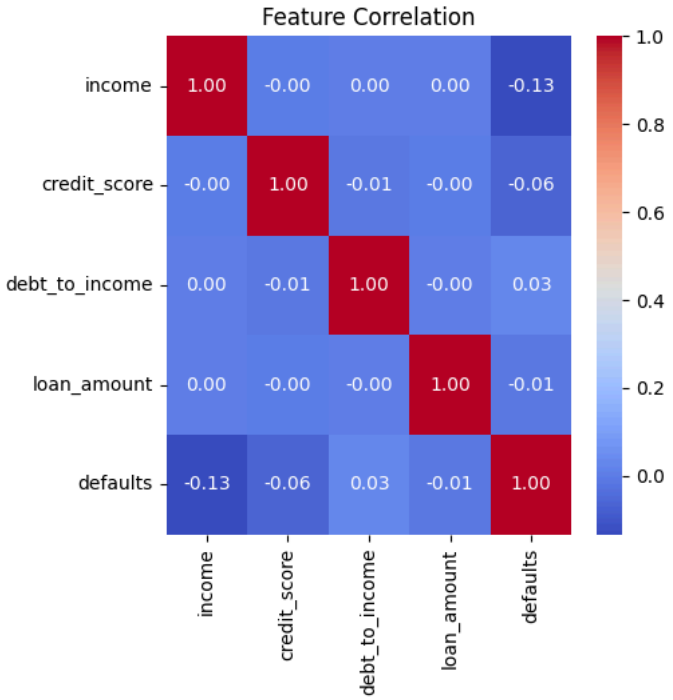
Data Source: Synthetic loan application dataset (10,000 samples)











Training/Testing Split: 80/20 (8,000 training, 2,000 testing samples)

State Space consists of 8 continuous features representing loan applicant characteristics:

Feature	**Description**
Income	Annual Income
Credit Score	Credit Rating (300-850)
Employment Length	Years of employment
Debt-to-Income	DTI ratio (0-0.6)
Loan Amount	Requested loan amount
Age	Applicant age (18-80)
Credit Lines	Number of open credit lines
Credit History	Length of credit history

The **action space** represents loan approval decisions with 4 discrete actions:

Action	Description	Approval Multiplier
0	Reject Loan	0.0 (0%)
1	Approve 50% of requested amount	0.5 (50%)
2	Approve 75% of requested amount	0.75 (75%)
3	Approve full requested amount	1.0 (100%)

The reward function is designed to incentivize profitable lending while penalizing defaults and opportunity costs.

- Rejecting a good applicant: Small negative reward (-5)
- Rejecting a risky applicant: Small positive reward (+2)
- Approving a loan that defaults: Large negative reward proportional to the loan amount (-approved_amount / 1000)
- Approving a loan that is repaid: Positive reward proportional to the interest earned (+total_interest / 1000)

- Each episode processes the entire dataset sequentially (8,000 training samples)
- Episode terminates when current_idx >= len(data)
- No early termination based on performance metrics

- Episodic task: Each loan application is independent
- Episodes reset to shuffle data order for better generalization
- No carry-over state between episodes

2) Algorithm Description

Two algorithms were implemented:

-Tabular Q-Learning with state discretization

-Deep Q-Network (DQN) with experience replay and target network

Q-Learning Update Rule:

$$Q(s,a) \leftarrow Q(s,a) + \alpha[r + \gamma \cdot \max_{a'} Q(s',a') - Q(s,a)]$$

DQN Loss Function

$$L(\theta) = E[(r + \gamma \cdot \max_{a'} Q(s',a';\theta^-) - Q(s,a;\theta))^2]$$

Q-Learning Hyperparameters:

```
n_bins=12,
learning_rate=0.15,
gamma=0.95,
epsilon=1.0,
epsilon_decay=0.9995,
epsilon_min=0.01
```

DQN Hyperparameters:

```
learning_rate=0.001,
gamma=0.95,
epsilon=1.0,
epsilon_decay=0.9995,
epsilon_min=0.01,
buffer_size=20000,
batch_size=32,
target_update_freq=5
```

Epsilon-greedy (**ε-greedy exploration)** is used for both algorithms:

- Starts with $\varepsilon = 1.0$ (100% random exploration)
- Decays geometrically: $\varepsilon \leftarrow \varepsilon \times 0.9995$ after each episode
- Lower bound: $\varepsilon\_min = 0.01$ (always maintains 1% exploration)

3) Implementation Details

- **PyTorch 2.8.0+cu126** for the Deep learning framework
- **NumPy 2.0.2** for Numerical computing

- **Pandas 2.2.2** for Data manipulation
- **Matplotlib 3.10.0** for Visualization
- **Seaborn 0.13.2** for Statistical visualization
- **Scikit-learn 1.6.1** for Preprocessing (StandardScaler)
- **tqdm 4.67.1** for Progress bars

Total data: 10,000 loan applications
Training set: 8,000 (80%)
Testing set: 2,000 (20%)
Random seed: 42 (for reproducibility)

Episodes: 500 for Q-Learning and 1000 for DQN
Each episode processes all data points once
The model is trained until Q-values converge and cumulative reward stabilizes

The experiments were conducted on a MacBook Air (M2, 2022) equipped with an Apple M2 chip featuring an 8-core CPU and an 8-core integrated GPU, along with 8 GB of unified memory.

IV. DISCUSSION

The performance metrics used are:

Total Reward: The cumulative reward obtained by the agent over an episode (or the entire test set). This is a primary measure of overall performance according to the defined reward function.

Total Profit: The cumulative financial profit generated by the agent's loan decisions. This is a key business metric.

Number of Approvals: The total count of loans approved by the agent.

Number of Defaults (among approved): The count of approved loans that resulted in default.

Approval Rate: The percentage of loan applications that were approved.

Default Rate (among approved): The percentage of approved loans that defaulted.

Correct Approvals: Number of approved loans that did not default.

Correct Rejections: Number of rejected loans that would have defaulted.

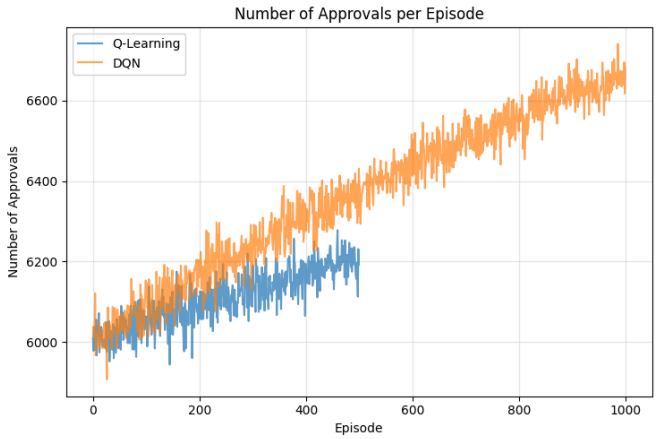
Approval Accuracy: Percentage of approved loans that did not default.

Model Performance:

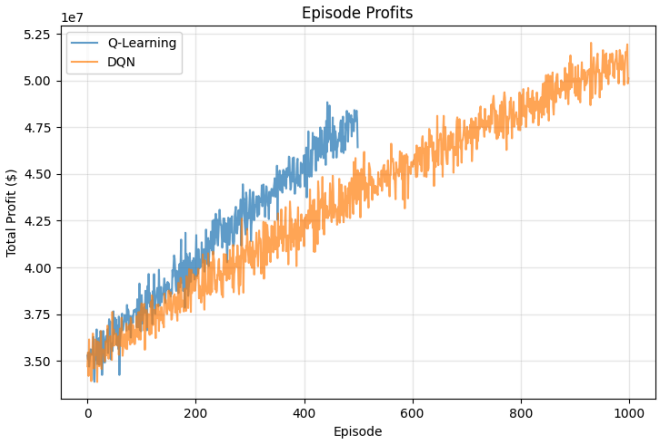
MODEL COMPARISON

Model	Total Reward	Total Profit	Approvals	Defaults	Approval Rate	Default Rate
Q-Learning	-7817.065542	8.793446e+04	13	2	0.65	15.384615
DQN	14224.114199	1.435211e+07	1959	281	97.95	14.344053
Random	6705.340906	8.561341e+06	1521	215	76.05	14.135437
Always Approve	10811.681108	1.081168e+07	2000	292	100.00	14.600000
Conservative	5537.507709	7.957508e+06	1355	177	67.75	13.062731

This chart plots the total reward accumulated by each agent during each training episode.

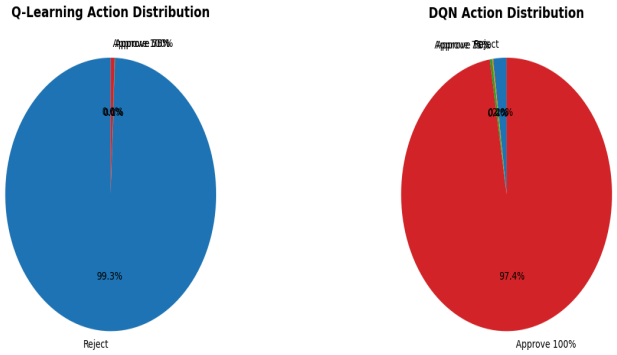


This chart plots the total profit generated by each agent in each training episode.

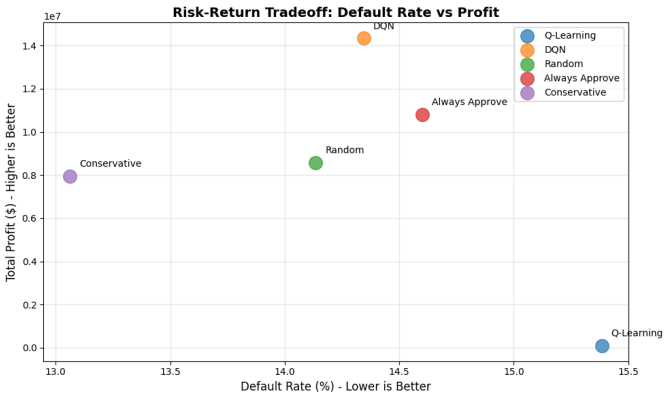


This chart shows how many loan applications each agent chose to approve in each training episode.

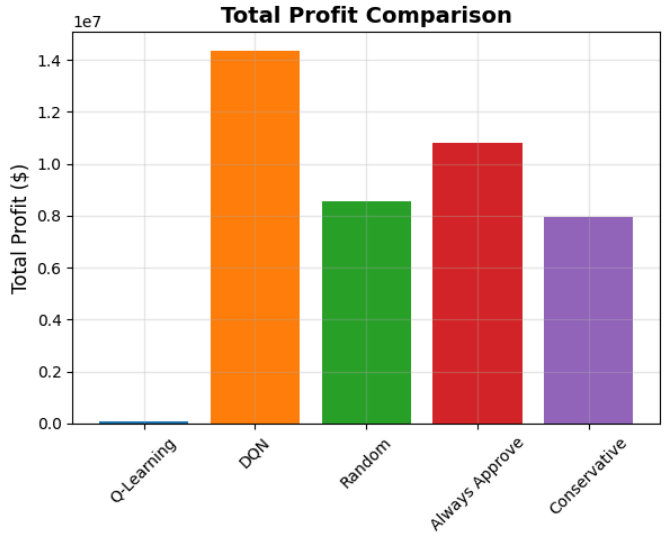
The learning curves show the trend of episode rewards and profits during training. Ideally, these curves should show an increasing trend and eventually flatten out, indicating that the agent is learning and converging to a better policy. The smoothed reward curves help to visualize the overall learning progress.



The action distribution pie charts show the percentage of times each action (Reject, Approve 50%, Approve 75%, Approve 100%) was taken by the Q-Learning and DQN agents during evaluation. This helps understand the agent's learned policy.

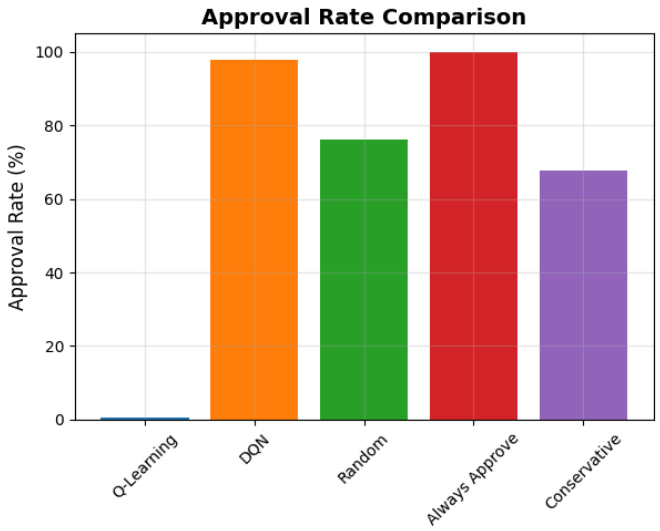


The risk-return tradeoff scatter plot visualizes the balance between default rate (risk) and total profit (return) for each model. The ideal model would be in the top-left quadrant (low default rate, high profit). This plot clearly shows that DQN achieved the highest profit, albeit with a slightly higher default rate than the Conservative policy.



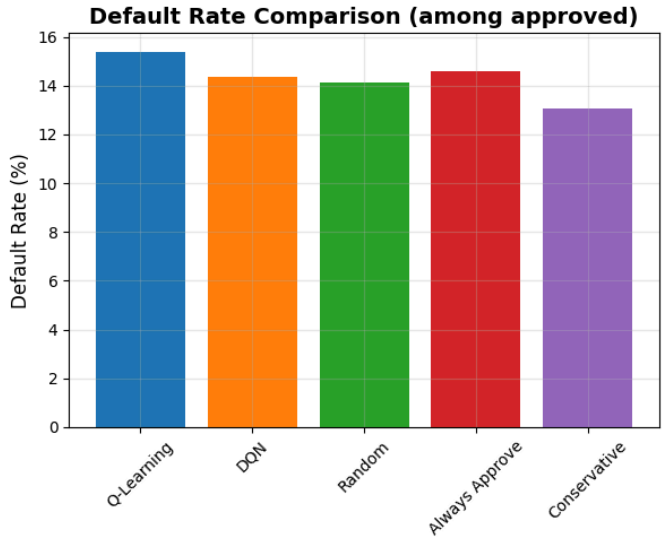
This bar chart compares the total cumulative profit generated by each model over the entire test set.

A higher bar indicates a more profitable model. As you can see, the DQN agent has the highest bar, indicating it achieved the significantly highest total profit among all models. Q-Learning had a negative total profit, while the baseline policies (Random, Always Approve, Conservative) had positive but much lower profits than DQN.



This bar chart shows the percentage of loan applications that each model approved.

This metric indicates how many potential customers each model is willing to serve. The "Always Approve" policy, as expected, has a 100% approval rate. DQN has a very high approval rate (close to 100%), while the Random and Conservative policies have lower approval rates. Q-Learning has a very low approval rate, indicating it was highly selective or conservative in this evaluation run.



This bar chart shows the percentage of approved loans that ended up defaulting.

This is a key risk metric. A lower bar indicates a lower default rate among the loans that were approved by the model. The Conservative policy has the lowest default rate, suggesting it is very good at avoiding defaults among approved loans. The other models, including DQN, have slightly higher but comparable default rates, which are still within a reasonable range (around 13-15%) for the synthetic data's realistic default probability.

1) Did the agent's behavior align with expectations?

The DQN agent's behavior, as seen in the action distribution, is to primarily approve loans. This aligns with the goal of maximizing profit, and the evaluation results show it was largely successful in doing so, leading to the highest total profit.

The Q-Learning agent's behavior, however, did not align with expectations in this run. It learned a highly conservative policy, rejecting almost all loans. This resulted in a low default rate but also very low profit. This suggests potential issues with Q-Learning's ability to effectively explore and learn in this continuous state space environment with the chosen discretization.

2) Were there any stability or convergence issues?

Looking at the learning curves, both agents show an increasing trend in rewards and profits, suggesting some degree of learning. However, the curves can still appear somewhat noisy, especially for Q-Learning, which might indicate some instability during training or challenges in converging to a perfectly stable policy. The long training time for DQN also suggests that convergence can be a gradual process.

3) How sensitive are results to hyperparameters or environment changes?

The difference in performance between the Q-Learning and DQN agents with their respective hyperparameters also demonstrates this sensitivity. Q-Learning struggled to perform well with the chosen discretization and hyperparameters, while DQN, with its neural network and specific hyperparameters, was able to learn a much more profitable policy.

V. CONCLUSION

1) What are the key takeaways from this research?

- Reinforcement Learning, particularly DQN, can be effectively applied to complex sequential decision-making problems like loan approval to optimize for financial metrics like profit.
- Careful design of the RL environment, including a realistic reward function and state space, is crucial for training agents that learn desirable behaviors.
- The choice of RL algorithm and its hyperparameters significantly impacts performance and convergence. DQN, with its ability to handle continuous states and larger capacity, outperformed tabular Q-Learning in this scenario.
- Comparing RL agent performance to simple baselines (random, always approve, rule-based conservative) helps demonstrate the value and potential of the learned policy.
- Visualizations of training progress and evaluation results are essential for understanding agent behavior and identifying areas for improvement.

2) How did the proposed approach advance or clarify the RL problem?

This approach clarifies the application of standard RL techniques (Q-Learning and DQN) to a practical financial decision-making problem. It demonstrates how a real-world scenario can be framed as an RL problem with a custom environment, state space, action space, and reward function. It highlights the challenges of training RL agents in such environments, particularly the sensitivity to environment design and hyperparameters, and shows how iterative refinement (as seen in the "FIXED" sections) is often necessary. It also provides a clear comparison framework to evaluate the performance of RL agents against simpler, non-RL approaches.

3) What are the main contributions in one paragraph?

This research contributes a practical demonstration of applying Reinforcement Learning, specifically DQN and tabular Q-Learning, to the complex problem of loan approval. By developing a custom simulation environment that models the financial outcomes of loan decisions, the notebook shows how RL agents can learn to balance the competing objectives of maximizing profit and minimizing default risk. The work highlights the superior performance of the DQN agent over basic baselines and tabular Q-Learning in generating higher overall profits, demonstrating the potential of deep reinforcement learning for optimizing decision-making in the financial domain and providing insights into the critical role of environment design and hyperparameter tuning for successful application.

4) What are the next steps or future directions?

Incorporate more comprehensive and realistic features into the state space, such as historical payment data, information about collateral, and external economic indicators.

Transition from synthetic data to real-world anonymized loan default data for training and evaluation to build more robust and generalizable models.

Explore and implement more advanced and state-of-the-art RL algorithms (e.g., A3C, PPO, SAC) that may offer improved stability, sample efficiency, or performance in complex environments.

Address ethical considerations by adding fairness constraints to the reward function or training process to mitigate bias and ensure equitable loan approval decisions.

Develop strategies for deploying the trained RL agent as a real-time decision-making system

within a financial institution's existing infrastructure.

REFERENCES

[1] A. Markov, Z. Seleznyova, and V. Lapshin, "Credit scoring methods: Latest trends and points to consider," *Journal of Finance and Data Science*, vol. 8, pp. 180–201, 2022. doi: 10.1016/j.jfds.2022.07.002.

[2] World Bank, "The use of alternative data in credit risk assessment," 2024. [Online]. Available: https://documents1.worldbank.org/curated/en/099031325132018527/pdf/P179614-3e01b947-cbae-41e4-85dd-2905b6187932.pdf

[3] S. Kiatsupaibul, P. Chansiripas, P. Manopanjasiri, K. Visantavarakul, and Z. Wen, "Reinforcement learning in credit scoring and underwriting," *arXiv:2212.07632*, 2022.

[4] R. Khraishi and R. Okhrati, "Offline deep reinforcement learning for dynamic pricing of consumer credit," in *Proc. 3rd ACM Int. Conf. AI in Finance (ICAIF)*, 2022. doi: 10.1145/3533271.3561682.

[5] C. Li, H. Wang, S. Jiang, and B. Gu, "The effect of AI-enabled credit scoring on financial inclusion: Evidence from one million underserved population," *MIS Quarterly*, 2024. [Online]. Available: https://misq.umn.edu/misq/article/48/4/1803/2314/The-Effect-of-AI-Enabled-Credit-Scoring-on

[6] J. Gao, H. L. Yi, and D. Zhang, "Algorithmic underwriting in high risk mortgage markets," MIT Sloan Working Paper, 2024. [Online]. Available: https://mitsloan.mit.edu/sites/default/files/inline-files/Session1_Paper3_Algorithmic%20Underwriting.pdf

[7] Y. Chen, R. Calabrese, and B. Martin-Barragán, "Interpretable machine learning for imbalanced credit scoring datasets," *European Journal of Operational Research*, vol. 312, no. 1, pp. 357–372, 2024. doi: 10.1016/j.ejor.2023.06.036.

[8] Y. Bai, S. Ma, R. Luo, L. Deng, L. Shen, and Z. Zhang, "A review of reinforcement learning in financial applications," *arXiv:2411.12746*, 2024.

[9] J. Gao, Y. Wu, and Z. Wang, "Do local branches shape banks' mortgage lending standards?" *FDIC Bank Research Conference* paper, 2023. [Online]. Available: https://www.fdic.gov/analysis/cfr/bank-research-conference/annual-22nd/papers/zhang-paper.pdf

[10] S. Paul, A. Gupta, A. K. Kar, and V. Singh, "An automatic deep reinforcement learning-based credit scoring model using deep-Q network for classifying customer credit requests," in *Proc. 2023 IEEE Int. Symp. Technology and Society (ISTAS)*, 2023. doi: 10.1109/ISTAS57930.2023.10306111.

[11] N. De La Fuente and D. A. Vidal Guerra, "A comparative study of deep reinforcement learning models: DQN vs PPO vs A2C," *arXiv:2407.14151*, 2024.

[12] European Commission, "AI Act—EU rules on artificial intelligence," 2024. [Online]. Available: https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai

[13] M. Herasymovych, "Using reinforcement learning to optimize the acceptance threshold of a credit scoring model," *Appl. Soft Comput.*, vol. 83, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1568494619304788

[14] Y. Bai *et al.*, "A review of reinforcement learning in financial applications," *Annu. Rev. Stat. Appl.*, vol. 12, pp. 1-24, Mar. 2025.

[15] Y. Li, "Deep reinforcement learning based on balanced stratified prioritized experience replay for credit scoring," *Expert Syst. Appl.*, 2023. [Online]. Available: https://assets.researchsquare.com/files/rs-2422835/v1_covered.pdf

[16] S. Alfonso-Sánchez, J. Solano, A. Correa-Bahnsen, K. P. Sendova, and C. Bravo, "Optimizing credit limit adjustments under adversarial goals using reinforcement learning," *Eur. J. Oper. Res.*, vol. 315, no. 2, pp. 596-611, 2024.

[17] S. Alfonso-Sánchez, J. Solano, A. Correa-Bahnsen, K. P. Sendova, and C. Bravo, "Optimizing credit limit adjustments under adversarial goals using reinforcement learning," arXiv preprint arXiv:2306.15585, Jun. 2023.

[18] C. Bravo, S. Maldonado, and R. Weber, "Cost-sensitive reinforcement learning for credit risk," *SSRN Electron. J.*, Nov. 2024. [Online]. Available: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=5023192

[19] H. Lakkaraju, J. Kleinberg, J. Leskovec, J. Ludwig, and S. Mullainathan, "The selective labels problem: Evaluating algorithmic predictions in the presence of unobservables," in *Proc. 23rd ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Halifax, NS, Canada, 2017, pp. 275-284.

[20] N. Kozodoi, J. Jacob, and S. Lessmann, "A framework for training and evaluating credit scoring models," *Eur. J. Oper. Res.*, 2025. [Online]. Available: https://arxiv.org/html/2407.13009v1

[21] T. Schaul, J. Quan, I. Antonoglou, and D. Silver, "Prioritized experience replay," in *Proc. Int. Conf. Learn. Representations (ICLR)*, San Juan, Puerto Rico, 2016.

[22] "Contextual bandits for loan approvals," Meegle, Jul. 2025. [Online]. Available: https://www.meegle.com/en_us/topics/contextual-bandits/contextual-bandits-for-loan-approvals

[23] C. Subramanian and P. P. Ravindran, "Causal contextual bandits with one-shot data integration," *Front. Artif. Intell.*, vol. 7, Dec. 2024.

[24] "The evolution of reinforcement learning in quantitative finance," arXiv preprint arXiv:2408.10932v3, 2024. [Online]. Available: https://arxiv.org/html/2408.10932v3

[25] E. Lin, Q. Chen, and X. Qi, "Deep reinforcement learning for imbalanced classification," *Appl. Intell.*, vol. 50, pp. 2488-2502, 2020.

[26] M. Guidolin and I. Pedio, "Data science for economics and finance," in *Springer Texts in Business and Economics*. Springer, 2021, ch. 4, pp. 91-120.