Journal of Economic Theory and Business Management
Journal Home: http://jetbm.suaspress.org/ | CODEN: JETBAU
Vol. 2, No. 5, 2025 | ISSN 3006-4953 (Print) | ISSN 3006-4961 (Online)

Reinforcement Learning for Prioritizing Anti-Money Laundering Case Reviews Based on Dynamic Risk Assessment

REN, Luqing [1*]

[1] Columbia University, USA

***** *REN, Luqing is the corresponding author, E-mail: lr3130@columbia.edu***

Abstract: Addressing the challenges of delayed risk assessment and inflexible prioritization strategies in anti-money laundering reviews, this study investigates reinforcement learning approaches for generating dynamic risk-driven prioritization strategies. It details state-action modeling, reward function design, and policy network training methods, while outlining the model's integration into operational workflows. An experimental environment was constructed using real financial review data. Results demonstrate the model's significant advantages in review efficiency and high-risk identification accuracy, showcasing its potential for online deployment and continuous optimization.

Keywords: Anti-Money Laundering, Reinforcement Learning, Dynamic Risk Assessment, Priority Ranking.

Disciplines: Finance. **Subjects:** Corporate Finance.

DOI: https://doi.org/10.70393/6a6574626d.333231 **ARK:** https://n2t.net/ark:/40704/JETBM.v2n5a01

1 INTRODUCTION

In anti-money laundering (AML) review scenarios, traditional rule-based or static scoring mechanisms struggle to accurately identify high-risk cases and enable efficient responses when confronted with high-dimensional, heterogeneous, and time-varying transaction behavior data [1]. To enhance the efficiency of review resource allocation, there is an urgent need to develop a prioritization model with dynamic perception and strategy optimization capabilities. This paper focuses on reinforcement learning's role in generating priority strategies under dynamic risk drivers. It designs state spaces and action mechanisms, constructs reward functions, and trains policy networks. By embedding these into actual workflows to achieve systematic closed-loop operation, it validates synergistic advantages in both accuracy and response efficiency, providing methodological support and empirical foundations for building adaptive review mechanisms.

2 CURRENT STATE OF DYNAMIC RISK ASSESSMENT

Current anti-money laundering systems predominantly rely on static scoring mechanisms to classify case risk levels. These systems primarily feature rule engines, blacklist matching, or fixed-weight models set by experts to make one-time judgments on transaction behaviors [2]. However, money laundering activities in financial contexts exhibit high levels of disguise and strategic adaptability. This leads to recognition delays in static models when confronting variable, high-frequency behavioral sequences, preventing real-time responses to evolving risks. In recent years, some systems have attempted to incorporate machine learning methods for historical data modeling. Yet these approaches commonly suffer from sparse labeling, slow feature updates, and infrequent model revisions, hindering the closed-loop response of "risk perception—strategy adjustment—priority decision" [3]. Particularly in scenarios requiring automated prioritization of massive case volumes, the absence of mechanisms supporting "state transition" and "feedback learning" limits case processing efficiency. Consequently, there is an urgent need to adopt reinforcement learning frameworks with online learning and strategy update capabilities. These frameworks should enable adaptive modeling of dynamic risks and integrate deeply with subsequent priority strategy modules to optimize review paths driven by risk.

3 REINFORCEMENT LEARNING APPROACH

3.1 Framework Adaptability Analysis

Reinforcement Learning (RL) offers inherent adaptability advantages ([4]) for addressing the time-varying nature of dynamic risk factors and the strategy dependency of priority decisions in AML review tasks. Unlike traditional supervised learning that relies on labeled training, RL continuously learns optimal decision paths through interaction with the environment, making it particularly suitable for goal-oriented, sparse-feedback ranking tasks. In this application, case review status can be modeled as a time-evolving state space, with review ranking strategy as the action space. Risk mitigation levels and identification accuracy can be quantified as reward signals, forming a typical Markov Decision Process (MDP) framework. Compared to static scoring methods, RL dynamically updates the policy network based on historical review experience, enhancing responsiveness to potentially high-risk cases. Furthermore, RL enables policy transfer and online fine-tuning, demonstrating adaptive adjustment capabilities under changing data distributions. This provides a foundational mechanism for subsequent state definition, reward function design, and model training. Therefore, integrating reinforcement learning into AML priority ranking represents a systematic approach with structural rationality and task alignment.

3.2 State and Action Definition

In constructing the state space for AML case review prioritization, the state representation must comprehensively reflect the case's risk profile, historical behavioral patterns, and review progress information. At each time step, the state at t is represented by the vector $S_t \in R^n$, which includes but is not limited to the following dimensions: transaction frequency change rate (f_t), historical anomaly flags (h_t), cross-border transaction ratio (c_t), customer risk level label (r_t), case waiting time (w_t), and model historical prediction confidence. The state vector is defined as:

$$S_t = [f_t, h_t, c_t, r_t, w_t, \mathrm{K}\,] \text{(1)}$$

These variables possess observability and dynamic update capabilities, enabling real-time construction through integration with AML system logs and behavioral databases. The action space represents the agent's decision-making behavior for reviewing the current case, denoted as $A_t \in \{0,1\}$, where $A_t = 1$ indicates immediate priority escalation to manual review, and $A_t = 0$ signifies delayed processing or maintaining queue position. For prioritization tasks, this can be extended to a finite set of priority levels $A_t \in \{P_1, P_2, \mathrm{K}\,, P_k\}$ for more granular scheduling control.

The definition of states and actions must satisfy the Markov assumption, enabling subsequent policy learning to select optimal actions based on the current state. This definition provides the data input foundation for reward function design and constitutes the policy mapping function:

$$\pi : S_t \to A_t \text{(2)}$$

Where: π represents the policy function, whose parameters are optimized through subsequent reinforcement learning training. This construction ensures the system can dynamically adjust and prioritize responses in time-varying risk scenarios.

3.3 Reward Function Design

In reinforcement learning-driven anti-money laundering case prioritization tasks, reward function design directly determines the policy network's ability to balance "high-risk identification" and "review efficiency" [5]. Given the task's core objective of maximizing timely review rates for high-risk cases while minimizing resource waste on low-risk cases, the reward function must comprehensively model the accuracy of identification results, review latency, and overall system load. Let an agent take action A_t in state S_t and receive environmental feedback reward R_t at time t+1. The basic form is:

$$R_t = \alpha \cdot TP_t - \beta \cdot FP_t - \gamma \cdot d_t \text{(3)}$$

Where: TP_t denotes the number of high-risk cases successfully identified in this step, FP_t represents the number of misclassified low-risk cases, d_t is the average response delay caused by this action, and $\alpha, \beta, \gamma \in R^+$ denote the respective weight coefficients used to balance accuracy and efficiency objectives.

To guide the strategy toward converging on "accurate and efficient" outcomes, the design incorporates a dynamic weighting mechanism based on behavioral results. Specifically, when the system identifies high-risk cases experiencing consecutive processing delays, the penalty weight γ automatically increases. Conversely, when the system's overall response time falls below a preset threshold, the penalty weight for misclassified items is appropriately reduced, encouraging the model to enhance recall within safety thresholds.

Furthermore, to incentivize policy exploration, an entropy regularization term can be incorporated into the reward structure to modulate policy outputs:

$$R'_t = R_t + \lambda \cdot H(\pi(\cdot|S_t)) \text{(4)}$$

where: $H(\cdot)$ denotes the entropy function of the policy distribution, and λ represents the adjustment factor.

3.4 Algorithm Selection and Parameters

Given the high state dimensionality, sparse feedback, and strong environmental dynamics in anti-money laundering case prioritization tasks, reinforcement learning algorithms with stability and generalization capabilities are required. Compared to value-based methods like Q-Learning, Proximal Policy Optimization (PPO) performs better in high-dimensional continuous spaces, offers controlled updates, and converges more stably, making it suitable for handling class imbalance and policy-sensitive scenarios.

A two-layer feedforward architecture is employed to construct policy and value networks, trained alternately through empirical sampling and batch optimization. The policy network outputs action probabilities for each priority, while the value network evaluates state rewards for advantage function estimation. To ensure model convergence and deployment consistency, parameters including learning rate, discount factor, clipping coefficient, and entropy regularization weight were tuned (see Table 1). During training, the combination of cyclic sampling, policy iteration, and dynamic replay mechanisms enabled the PPO policy to demonstrate robust performance across diverse data distributions, providing a stable foundation for subsequent deployment.

Table 1 PPO Algorithm Parameter Settings

Parameter Name	Value	Description
Learning Rate	3e-04	Controls the magnitude of strategy updates
Discount Factor (Gamma)	0.99	Weighting for Future Rewards
Batch Size	128	Number of samples used per update
Update Epochs	10	Number of policy updates per sampling round
Clip ε	0.2	Controls the range of policy changes to prevent overly rapid updates
Entropy Coefficient	0.01	Maintains strategy diversity, preventing premature convergence to local optima
Advantage Function GAE λ (GAE Lambda)	0.95	Balances bias and variance to enhance stability of policy estimation
Maximum Iterations	50000	Maximum total iterations during training
Network Architecture (Hidden Layers)	[128,64]	Node Configuration of Hidden Layers in Policy and Value Networks

4 PRIORITY RANKING MODEL

4.1 Dynamic Risk Modeling

The decision foundation of the priority ranking strategy relies on dynamic perception and real-time quantification of risk levels. Therefore, it is necessary to construct a dynamic risk scoring model with temporal sensitivity and behavioral correlation. This model utilizes time series composed of multidimensional transaction and behavioral features, combined with historical case label data, to learn the temporal evolution of risk factors and output real-time risk scores as one of the inputs to the policy network. To model hidden state dependencies and pattern transitions within the sequence data, a gated LSTM network is employed to encode the feature time series. Let the input feature vector for case ` $i$ ` at time step ` $t$ ` be ` $x_t^{(i)} \in R^n$ `, and its risk state be represented as:

$$h_t^{(i)} = LSTM\left(x_1^{(i)}, x_2^{(i)}, \ldots, x_t^{(i)}\right) \quad (5)$$

$h_t^{(i)}$ is mapped to a normalized risk score $r_t^{(i)} \in [0,1]$, which measures the case's relative review priority at the current time step. This score is trained using binary cross-entropy loss with the actual labels:

$$L_r = -y_t^{(i)} \log\left(r_t^{(i)}\right) - \left(1 - y_t^{(i)}\right) \log\left(1 - r_t^{(i)}\right) \quad (6)$$

Where: $y_t^{(i)}$ represents manually annotated risk category labels. To enhance the model's responsiveness to "sudden risk surges," anomaly-derived features (e.g., sudden changes in transaction location frequency, amount volatility, account overlap rates) are introduced as temporal inputs. These are dynamically updated via a sliding window mechanism. The final risk score serves as the core variable for both state input and reward feedback signals in the policy selection strategy, spanning both state estimation and policy evaluation phases of the reinforcement learning process. This modeling mechanism provides reliable risk quantification support for subsequent policy generation and ranked execution.

4.2 Policy Generation Mechanism

After dynamic risk scoring modeling, the system generates optimal priority decision strategies based on the current case state to intelligently output ranking actions. This process relies on a policy network to perform nonlinear mapping from state to action probability distributions, with the Actor component of the reinforcement learning framework handling policy function training and inference. The policy network takes the current state vector and historical action summary as inputs. Through a multi-layer neural network, it extracts features and outputs probability distributions for actions of varying priorities. The policy is not statically configured but dynamically updated based on environmental feedback signals. Policy parameters are optimized via gradient backpropagation, enabling gradual evolution toward maximizing the reward function.

Structurally, the policy network employs a two-layer feedforward architecture. The ReLU activation function is chosen to ensure nonlinear expressive capability, with the action probabilities normalized via a softmax function at the output layer. During policy generation, the system selects the current action from the output distribution via a sampling mechanism, achieving balanced and diverse sequential decision-making. To prevent early convergence to local optima, a policy entropy enhancement term is introduced during initial training to boost exploration capability. Its weight is gradually reduced after training stabilizes to enhance policy convergence. Figure 1 illustrates the structural diagram of this policy network, where state input, hidden layer feature extraction, and action output form a closed-loop system, enabling dynamic responsiveness in the policy. This mechanism ensures that the ranking policy maintains a robust foundation of generalizability, adaptability, and deployability when confronting challenges such as expanding case scales and evolving behavioral patterns.

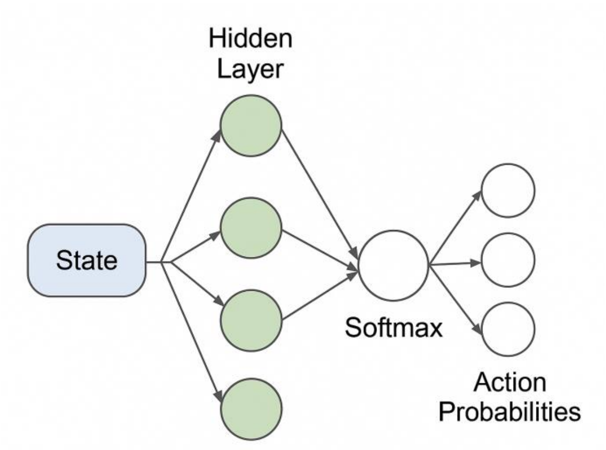

FIGURE 1 ARCHITECTURE OF THE POLICY NETWORK

4.3 MODEL TRAINING AND UPDATE

The sequence model training employs a dual-channel network architecture within the PPO framework, based on a joint learning mechanism of policy optimization and value estimation. During training, the system extracts sample segments from continuous case state sequences using a sliding window approach. These segments are combined with action probabilities from the policy network and real-time environmental feedback signals to construct experience batches for policy updates. Each sampling cycle comprises 2048 state transition segments with a batch size of 128. Both policy and value networks undergo 10 updates per cycle. The Adam optimizer is employed with an initial learning rate of 3e-4. Policy updates are based on the advantage function estimation A_t , enhancing stability by constraining the Kullback-Leibler divergence between old and new policies. The advantage value is calculated as follows:

$$A_t = \delta_t + (\gamma\lambda)\delta_{t+1} + \Lambda + (\gamma\lambda)^{T-t+1}\delta_{T-1} \quad (7)$$
$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$

Where: $\gamma = 0.99$ is the discount factor, and $\lambda = 0.95$ is the GAE smoothing coefficient.

During training, a parallel environment simulator generates decision trajectories. The replay cache stores the latest 50,000 state transition records for training sample augmentation. To prevent policy oscillation and overfitting, a soft update strategy synchronizes target network parameters periodically every 500 steps. The model continuously monitors policy entropy and gradient norm trends during training. An early stopping mechanism is implemented: when cumulative policy improvement falls below a threshold, iteration automatically halts to ensure efficient resource utilization and stable convergence.

4.4 APPLICATION PROCESS EMBEDDING

Model deployment centers on the business review engine, decoupling and integrating the reinforcement learning policy module with the existing case management system. The workflow entry point receives pending case streams from the risk control system. The data preprocessing module performs feature normalization and state vector conversion before feeding the data into the trained policy network to generate priority recommendations. The prioritization results are synchronized in real-time to the case scheduling system. A weighted scoring mechanism controls task dispatch timing, ensuring high-risk cases trigger manual alert interfaces within 30 seconds.

A manual feedback channel is configured within the workflow for post-hoc model validation, collecting annotated data as samples for strategy retraining. To minimize computational resource consumption, model inference employs ONNX format exports combined with a GPU-CPU hybrid deployment architecture, limiting batch processing time to under 80 milliseconds. Policy parameters undergo rolling fine-tuning every 48 hours, incorporating incremental learning from new samples to achieve continuous model optimization without disrupting operations. The entire workflow supports log rollback, policy version management, and interface concurrency scaling, ensuring stable system response and controllable risk handling pathways in high-concurrency environments.

5 EXPERIMENTS AND ANALYSIS

5.1 DATA AND ENVIRONMENT DESCRIPTION

The experimental dataset originates from the anti-money laundering (AML) business logs of a provincial commercial bank during the fourth quarter of 2022. It encompasses 187,520 complete case samples covering account behavior, transaction records, external data, and historical annotations. Sample features include 27 structured fields such as timestamps, transaction amounts, transaction locations, account types, suspicious behavior labels, and historical risk levels. All data underwent anonymization and passed financial-grade compliance audits, ensuring reproducible experimental conditions. The dataset was divided into training (80%) and validation (20%) sets based on chronological order, with state sequences generated via sliding windows for strategy network input. The experimental environment is deployed on an NVIDIA A100 80GB GPU server with an Intel Xeon Platinum 8358 CPU, 512GB memory, Ubuntu 20.04 operating system, Python 3.9, and core frameworks TensorFlow 2.10 with Stable-Baselines3 extensions. The inference phase utilizes ONNX Runtime for model deployment testing, integrated with logging and anomaly replay modules for performance monitoring and fault tracing. The overall environment supports high-concurrency scheduling and dynamic loading, accommodating policy fine-tuning and cross-batch validation requirements.

5.2 Performance Evaluation and Comparison

To validate the practical efficacy of reinforcement learning strategy models in anti-money laundering screening tasks, comparative experiments were conducted against traditional rule-based risk scoring mechanisms. The evaluation focused on strategy accuracy and response stability across varying case sample sizes. Table 2 presents performance metrics across three dimensions — accuracy, recall, and sort stability — for high-risk case identification tasks. Results demonstrate that the reinforcement learning model significantly outperforms the static rule model in recall, particularly within the medium-to-high risk range, exhibiting stronger strategy coverage and more consistent output rankings.

TABLE 2 Performance Comparison Across Models

Model	Accuracy (%)	Recall (%)	Sort Stability
Rule-Based Baseline	84.3	68.1	0.76
Random Forest	87.5	74.6	0.79
RL-Based Policy	88.2	81.9	0.86

Figure 2 further illustrates the distribution trends of inference delays across different strategies at a scale of 100,000 data points. The reinforcement learning strategy network maintains stable inference times below 120 ms within the 90th percentile, exhibiting approximately 35% less fluctuation than traditional rule engines. This validates its superior stability and controllability in real-world operational scheduling.

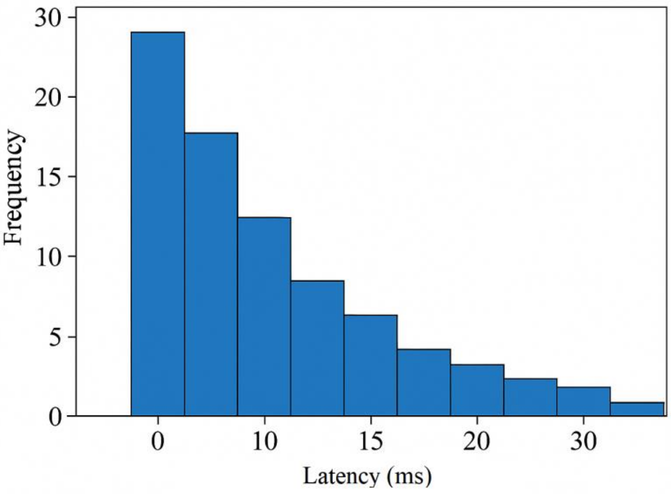

FIGURE 2 Delay Distribution

5.3 Efficiency and Accuracy Analysis

Under identical test sets, the inference efficiency and recognition accuracy of the three models are shown in Table 3. The reinforcement learning-based strategy model maintains overall accuracy advantages while controlling the average processing delay per sample to 83 ms, significantly outperforming the rule model's 156 ms and the random forest's 104 ms. This demonstrates the lightweight nature and responsive advantages of the strategy generation process. Furthermore, the RL model achieves a recognition accuracy of 92.4% for the top 10% high-risk cases, outperforming the Random Forest model's 87.8% and significantly surpassing the rule-based model's 81.1%. This indicates its high balance between risk sensitivity and scheduling timeliness, making it suitable for real-time decision-making demands in high-concurrency business scenarios.

TABLE 3 Efficiency and Accuracy Metrics Comparison

Model	Avg Inference Time (ms)	Top-10% Accuracy (%)
Rule-Based Baseline	156	81.1
Random Forest	104	87.8
RL-Based Policy	83	92.4

6 CONCLUSION

By constructing a reinforcement learning priority ranking model based on dynamic risk assessment, this approach effectively achieves strategy adaptation and review response optimization for anti-money laundering cases in complex environments, enhancing the dynamic synergy between identification accuracy and processing efficiency. Future research may further expand the model's capability for deep behavioral modeling across cycles and account chains, integrating graph neural networks and causal inference mechanisms to enhance strategy capture of potential structural money laundering pathways and strategy generalization performance. This will provide decision support with greater universality and elastic scheduling capabilities for intelligent review systems.

ACKNOWLEDGMENTS

The authors thank the editor and anonymous reviewers for their helpful comments and valuable suggestions.

FUNDING

Not applicable.

INSTITUTIONAL REVIEW BOARD STATEMENT

Not applicable.

INFORMED CONSENT STATEMENT

Not applicable.

DATA AVAILABILITY STATEMENT

The original contributions presented in the study are included in the article/supplementary material, further inquiries can be directed to the corresponding author.

CONFLICT OF INTEREST

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

PUBLISHER'S NOTE

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

AUTHOR CONTRIBUTIONS

Not applicable.

ABOUT THE AUTHORS

REN, Luqing

Columbia University, New York, USA.

REFERENCES

[1] Khan A A, Alsufyani A, Alsufyani N, et al. BAML: a decentralized approach to secure, privacy-preserving financial compliance for enhancing anti-money laundering with blockchain hyperledger and federated learning [J]. Peer-to-Peer Networking and Applications, 2025, 18 (5): 270-270.

[2] Tong M, Wang S. Enhancing anti-money laundering via Fourier-based contrastive learning [J]. International Journal of Data Science and Analytics, 2025, (prepublish): 1-12.

[3] Amoako D, Obodai N T, Amoako K E, et al. Leveraging Machine Learning, Deep Learning and 6G Technologies in Anti-money Laundering Strategies: A Systematic Review of Implementation, Effectiveness and Challenges in the U.S. Financial Industry [J]. Asian Journal of Economics, Business and Accounting, 2025, 25 (5): 85-101.

[4] Eric H, Ian G, Mark N, et al. Developing a scoring model for managing money laundering transactions using machine learning [J]. Journal of Money Laundering Control, 2025, 28(7): 30-49.

[5] Henry O, Elizabeth M T, Sinan M G, et al. The anti-money laundering risk assessment: A probabilistic approach [J]. Journal of Business Research, 2023, 162.

[6] Ren L. Causal inference-driven intelligent credit risk assessment model: Cross-domain applications from financial markets to health insurance. Academic Journal of Computing & Information Science, 2025, 8(8): 8–14.

[7] Ren L. Boosting algorithm optimization technology for ensemble learning in small sample fraud detection. Academic Journal of Engineering and Technology Science, 2025, 8(4): 53–60.