Khazanah Journal of Religion and Technology
Universitas Islam Negeri Sunan Gunung Djati Bandung, Indonesia
https://journal.uinsgd.ac.id/kjrt
Online ISSN: 2987-6060
Volume 3 Issue 1 (2025)
DOI: https://doi.org/10.15575/kjrt.v3i1.1706

Sharia Stock Investment Decision Making Using the Deep Recurrent Q-Network Model

Jasmein Al-baar Putri Rus'an
Department of Informatics
UIN Sunan Gunung Djati Bandung, Indonesia
1227050063@student.uinsgd.ac.id

Muhammad Zidan
Department of Informatics
UIN Sunan Gunung Djati Bandung, Indonesia
1227050079@student.uinsgd.ac.id

Muhammad Arkan Raihan
Department of Informatics
UIN Sunan Gunung Djati Bandung, Indonesia
markanraihan@gmail.com

Marleni Sukarya
Department of Informatics
UIN Sunan Gunung Djati Bandung, Indonesia
1227050069@student.uinsgd.ac.id

Sendi Ahmad Rafiudin
Department of Informatics
UIN Sunan Gunung Djati Bandung, Indonesia
sendymurphy@gmail.com

***Abstract*—** This study aims to design and evaluate a Deep Recurrent Q-Network (DRQN) agent for automated trading decision-making on Islamic stocks, training it with daily historical price data from the Indonesian Islamic Stock Index (ISSI) and integrating a Long Short-Term Memory (LSTM) layer. Although the agent successfully learns a profitable strategy during the training phase, on unseen test data, it exhibits passive behavior by only choosing the 'hold' action, resulting in zero profit—a phenomenon known as policy stagnation. This finding indicates that the used reward function implicitly encourages excessive risk aversion. The study concludes that the success of the DRQN architecture relies heavily on sophisticated reward engineering, underscoring the need for future research on dynamic and adaptive reward mechanisms to develop robust and generalizable trading agents in the complex Islamic finance domain.

Keywords- Deep Reinforcement Learning, DRQN, Islamic Finance, Islamic Stocks, Reward Function, Trading

I. INTRODUCTION

The Islamic stock market has experienced significant growth in recent years, in line with increasing public awareness of Islamic financial principles. Indexes such as the Jakarta Islamic Index (JII) and the Islamic Financial Services Index (ISSI) have become key benchmarks for evaluating the performance of Sharia-compliant investments. However, investment decision-making in the Islamic stock market still faces complex challenges, such as high volatility, limited information, and uncertain market conditions [1].

The development of artificial intelligence technology, particularly deep reinforcement learning (DRL), has opened up new opportunities in designing more adaptive and intelligent investment decision-making systems. DRL enables learning agents to acquire optimal strategies through a trial-and-error process based on environmental feedback (rewards), thus being able to respond to market dynamics in real time [2]. The Deep Recurrent Q-Network (DRQN) model, a variant of DQN that incorporates Long Short-Term Memory (LSTM), has proven effective in processing time-series data and capturing historical context in an incompletely observed market environment [3].

Previous research has explored the use of DRL for conventional stock trading, with promising results. For example, Zhou and Tang used DRQN enhanced with the ARBR sentiment indicator to generate better trading decisions in the Chinese stock market [4]. However, most of these studies have not addressed Islamic finance, leaving a research gap in the specific application of DRQN to Islamic stocks.

Although Faturohman and Nugraha have applied DRL in the context of a sharia-compliant stock portfolio, their model is actor-critic, not DRQN [1]. Therefore, there is still room for research on how DRQN can be optimized for individual sharia-compliant stock decision-making by considering historical technical data and sharia principles.

This research aims to apply the DRQN model to develop a sharia-compliant stock investment decision-making system. The model will be trained using historical stock price data from sharia-compliant indices such as the Jakarta Stock Exchange (JII), with the integration of technical indicators to enrich the observational input. By utilizing the LSTM architecture in DRQN, the model is expected to be able to recognize temporal patterns and make more informed decisions in dynamic market conditions.

This research focuses on Islamic stocks in Indonesia, using a technical approach and model performance simulations as the primary benchmarks. Fundamental analysis and market sentiment are not discussed in depth.

This study is expected to contribute to the development of artificial intelligence technology in Islamic finance, particularly in automating investment decision-making. By combining Islamic principles with DRQN's ability to understand market dynamics, this research can serve as a foundation for the development of AI-based Islamic investment recommendation systems for retail investors, investment managers, and halal fintech players.

II. RELATED WORKS

A. Conceptual Framework: DRL for Algorithmic Trading

Reinforcement Learning (RL) is a branch of machine learning in which an agent learns to make sequential decisions by interacting with an environment to maximize cumulative rewards [5], [6]. This framework is formally modeled as a Markov Decision Process (MDP), defined by a tuple of states, actions, probability transition functions, and reward functions [5]. In the context of financial trading, the agent is a trading algorithm, the environment is the stock market, states represent market information (e.g., historical prices, volume, technical indicators), actions are trading decisions (buy, sell, hold), and rewards are the resulting financial gains or losses [7], [8]. The agent's goal is to learn a policy (π), a mapping from states to actions, that maximizes the total expected reward over time [9]. DRL approaches in financial trading can be classified based on several fundamental dichotomies that influence how the agent learns and interacts with the environment [7], [10].

B. Model-Free vs. Model-Based

Model-free methods, such as DQN and its derivatives, learn optimal policies directly from *trial-and-error interactions* without attempting to build an explicit model of the market environment's dynamics [7]. This approach is popular because financial markets are highly noisy and non-stationary, making building accurate predictive models extremely difficult [10]. However, their main drawback is sample inefficiency*;* these methods require very large data volumes to converge, a significant challenge given the limited historical market data available. In contrast, [9]*model-based* methods first attempt to learn a predictive model of the environment, which can then be used for simulation and planning [11]. While more sample-efficient, their performance depends heavily on the accuracy of the learned environmental model. The tension between these two approaches has driven the development of hybrid methods [7]. The generalization failure often observed in purely *model-free approaches*, such as the one used in this study, is a predictable consequence of this fundamental challenge.

C. On-Policy vs. Off-Policy

On-policy algorithms, such as A2C and PPO, update their policies using only data generated from the latest policy version [6]. This makes their learning process more stable but less efficient in terms of data usage [10]. In contrast, *off-policy algorithms*, such as DQN and DDPG, can learn from the experience data collected by previous policies, which is stored in a *replay buffer*. [5]This ability to reuse past experience makes *off-policy methods* significantly more sample-efficient, making them a logical choice for data-constrained domains such as finance [9]. The choice of DRQN in this study, which is a derivative of DQN, aligns with the need to maximize the utility of available historical data.

Various DRL algorithms have been applied in trading, each with its own strengths and weaknesses [12]. Value-*based algorithms* such as DQN focus on learning the value of each action under certain circumstances. Policy-*based algorithms* such as A2C and PPO directly optimize a policy function [12], [13]. Meanwhile, *Actor-Critic methods* (e.g., A2C, PPO, DDPG) combine both approaches, where the 'actor' is responsible for selecting actions and the 'critic' evaluates them, resulting in a more stable and efficient learning process [7]. Some recent research even suggests that *ensemble strategies*, which combine several different DRL agents (e.g., A2C, PPO, and DDPG) and dynamically select the best one based on current market conditions, can produce more robust and superior [14]performance [15]. Table 1 presents a comparative analysis of some DRL algorithms commonly used in algorithmic trading.

Table 1. Comparative Analysis of DRL Algorithms [7], [12],[10]

Algorithm	Type	Policy Updates	Main Characteristics	Power in Trade	Weaknesses in Trading
DQN	Value-Based	Off-Policy	Experience Replay, Target Network	Sampling efficiency due to data reuse.	Tends to overestimate Q values; not suitable for continuous action spaces.
A2C	Actor-Critic	On-Policy	Synchronous updates, using the advantage function.	Stable and easy to implement.	Less efficient in sample use than off-policy methods.
PPO	Actor-Critic	On-Policy	Using clipped surrogate objective for stable policy updates.	A good balance between performance, stability and ease of implementation.	Still on-policy, so the sampling efficiency is lower than DDPG.
DDPG	Actor-Critic	Off-Policy	Using deterministic policies and replay buffers.	Effective for continuous action spaces (e.g., portfolio allocation).	Sensitive to hyperparameters and can be unstable during training.

D. Recurrent Architecture in DRL for Financial Data Analysis

Financial market data is inherently sequential and exhibits complex temporal dependencies, such as trend, momentum, and seasonal volatility [16]. Conventional artificial neural networks (*feed-forward networks*) are unable to capture these temporal relationships because they process each input independently. Therefore, *Recurrent Neural Networks* (RNNs) have become the architecture of choice for modeling financial time series data [16].

Long Short-Term Memory (LSTM) architecture, an advanced variant of RNN, is specifically designed to address the vanishing gradient problem *that* often occurs in simple RNNs. LSTM achieves this through a gating mechanism *consisting* of *input gates*, *forget gates*, and *output gates*. These gates intelligently regulate the flow of information, allowing memory cells to retain or forget information from previous time steps, thereby effectively capturing long-term dependencies [16], [17].

The Deep Recurrent Q-Network (DRQN) model is a natural evolution of the DQN architecture, where the *fully connected layers* are replaced by LSTM layers. This modification allows the agent to learn not only from the current market state but also from a sequence of previous states [18]. This is especially important in often partially observable market environments, where current information alone is insufficient to make optimal decisions. By keeping historical context in mind, DRQN agents are theoretically better able to recognize complex temporal patterns and make more informed decisions.

While DRQNs are a solid option, the research landscape continues to evolve rapidly. To remain relevant, it is important to acknowledge recent advances. For example, the *Dueling DRQN architecture* has been shown to improve performance by separating state value estimation *and* action advantage, leading to better generalization [18]. Furthermore, advanced models like *the Transformer*, with its attention mechanism, and new architectures like *xLSTM*, demonstrate superior performance compared to traditional LSTMs in financial forecasting and trading tasks [19]. Recognizing these advances positions the current study as a fundamental exploration while also providing a clear direction for future research.

E. Application of Artificial Intelligence in Islamic Finance

Islamic finance operates under a set of strict ethical principles that distinguish it from conventional finance [20]. These principles include the prohibition of interest (*riba*), excessive uncertainty (*gharar*), gambling (*maysir*), and investments in sectors considered haram (forbidden), such as alcohol, tobacco, and conventional banking [21]. The application of AI and *machine learning* (ML) in this domain offers significant opportunities to improve efficiency and compliance, but also presents unique challenges [22], [23].

One of the main applications of AI is in the automated screening of Islamic stocks. This process is traditionally performed manually and can be divided into two stages, both of which can be automated by AI:

1. Quantitative Screening: ML algorithms can automatically analyze a company's financial statements to filter based on financial ratios established by the Sharia board, such as the debt-to-total-assets ratio or the percentage of non-halal income to total revenue [24], [25]. This drastically speeds up the process and reduces human error.
2. Qualitative Screening: This is where *Natural Language Processing* (NLP) plays a crucial role. NLP models can process and analyze large volumes of unstructured data—such as annual reports, press releases, news articles, and legal documents—to detect a company's core business activities [24]. By analyzing the text, these models can identify whether a company is involved in prohibited industries or has ethical controversies that make it non-compliant with Sharia principles.

However, implementing AI in Islamic finance is not a simple technical solution; it introduces a complex set of socio-technical and ethical challenges. First, there is a scarcity of high-quality data specifically labeled for Sharia compliance, which hinders the training of accurate ML models [23]. Second, many AI models, particularly *deep learning models*, function as "black boxes," *where* their decision-making processes are opaque [20]. This potentially contradicts the Islamic principles of clarity, fairness, and

transparency in transactions [26], [27]. If an AI makes investment decisions without clear explanations, it could raise questions about its compliance and accountability. This highlights the urgent need for *Explainable AI* (XAI) in Islamic finance [28]. Finally, validating AI logic from a Sharia perspective requires close collaboration between data scientists, financial experts, and Sharia scholars. Current regulatory frameworks often lag behind the pace of technological development, creating a need for Sharia-compliant AI governance guidelines to ensure ethical and responsible implementation [29], [30].

F. Practical Challenges: Environmental Design and Reward Function

In a DRL system, the *reward function* is the most crucial component because it explicitly defines the agent's goal and guides its entire learning process. Poorly designed *reward* functions can easily lead agents to undesirable behavior, a phenomenon known as *reward hacking* [5]. In a trading context, if the *reward function* only rewards wins and heavily penalizes losses, agents might logically conclude that the safest strategy to avoid penalties is to never trade at all [10]. This behavior, which results in persistent ' *hold ' actions, is the form of reward hacking* observed in this study.

The academic literature offers several solutions to address this problem through reward shaping. *Instead* of using a simple profit-loss function, the *reward function* can be shaped using risk-adjusted metrics, such as the Sharpe Ratio or the Sortino Ratio [7]. These metrics measure not only profit but also volatility, thus encouraging agents to seek strategies that are not only profitable but also have well-managed risk [9], [31].

Recent trends in DRL research are moving away from predefined static *reward functions toward more dynamic and adaptive mechanisms. The concept of Self-Rewarding Deep Reinforcement Learning* (SRDRL) is particularly relevant. In the SRDRL framework, a separate neural network is trained to learn a dynamic *reward function* [32]. This network can be trained using expert-generated labels (e.g., *rewards* calculated based on the Sharpe Ratio or other performance metrics) and then predicts appropriate *rewards in real time* based on current market conditions [32]. This approach allows the agent to dynamically adjust its objectives, providing a more robust response to changes in market regimes and directly addressing the policy stagnation problem identified in this study. This provides strong and concrete recommendations for future work.

III. RESEARCH METHODS

A. Data collection

This study uses historical data from the Indonesian sharia-compliant stock index, the IDX Shariah Index (JKISSI), over the past five years. Data were collected daily, spanning the period from January 2020 to May 2025. Each data entry includes six key features: the opening price (Open), the highest price (High), the lowest price (Low), the closing price (Close), trading volume (Volume), and the daily percentage change (Change).

The dataset was obtained from the site Investing.com provides historical data for the IDX Shariah index (JKISSI) in CSV format. This data was selected because it represents a collection of sharia-compliant stocks in the Indonesian capital market that meet Islamic sharia principles. The selection of data from this Sharia-compliant index aims to ensure the system is relevant to the context of halal and ethical investment.

B. Pre-processing and Data Format

Before being used for training, the data was preprocessed to ensure completeness and consistency. This process included converting the volume column from a string format such as "13.5B" or "240M" to a real numeric value. Additionally, the price change percentage column was cleaned by removing the percent symbol and converted to a numeric type. The date column was converted to a datetime type and indexed for easy manipulation based on chronological order. All numeric features were then normalized to the range [0, 1] using the Min-Max Scaling method to ensure that each feature had an equal scale.

After the normalization stage, the data is arranged in a time series format using a sliding window approach. The window size used is five days, so each state or observation is formed from the last five days with six numeric features, resulting in a (5, 6) dimensional data set. This structure allows the model to study stock price dynamics over time. The dataset is then divided into two parts: 70% as training data and 30% as testing data.

C. Trading Environment Design

In this study, a custom-designed stock trading simulation environment mimics daily stock market dynamics. This environment allows a reinforcement learning agent to make buy, sell, or hold decisions based on historical price data and technical indicators.

There are three actions available to the agent at each time step: Buy (0) to purchase a stock, Sell (1) to sell the stock, and Hold (2) to take no action. Each episode begins with the agent holding no stock (holding = False). When the agent buys the stock, the current price is recorded as the buy price. If the agent then sells the stock and still holds the stock, the environment calculates the profit or loss as the percentage change between the buy and sell prices.

To ensure that agents learn to act rationally, the reward function is designed adaptively. Agents receive a positive reward for successfully selling at a profit, while sales at a loss are still rewarded negatively, but on a smaller scale. Selling without first buying is subject to a substantial fixed penalty. Similarly, excessively frequent Hold actions are subject to a small penalty to discourage passive, unprofitable behavior. Conversely, Buy actions are given a mild incentive to encourage exploration of trading opportunities.

This environment also records the total profit during each episode as a cumulative percentage of the transactions executed. The state provided to the agent at any given time reflects the last five days of price and indicator data, according to the window structure described in the previous section. The environment will terminate when the historical data is exhausted or the maximum number of steps per episode is reached.

D. DRQN Model Architecture

The reinforcement learning model used in this study is the Deep Recurrent Q-Network (DRQN). This model is an extension of the Deep Q-Network (DQN) by adding a long-term memory component through a Long Short-Term Memory (LSTM) architecture. This approach is considered more suitable for sequential and time-dependent stock market data, as LSTM is able to capture temporal patterns from previous price movements.

The DRQN model in this study was built using a layered neural network architecture consisting of one LSTM layer with 64 units as the first layer. This layer receives input in the form of a data window for the last five days, each day consisting of six numeric features: opening, high, low, closing, volume, and daily change. After the LSTM layer, the model continues with two Dense (fully connected) layers of 32 and 16 neurons, respectively, with a ReLU activation function. The output layer consists of three linear neurons that represent the Q-value for each available action: Buy, Sell, and Hold.

The model was trained using the Deep Q-Learning algorithm with a Mean Squared Error (MSE) loss function and the Adam optimizer. To improve training efficiency and stability, an experience replay technique was used, which stores every interaction between the agent and the environment (state, action, reward, next_state) in a buffer, then randomly samples from that buffer for the training process. This technique reduces correlation between sequential data and aids model generalization.

To ensure more stable training, a target network is also implemented, a copy of the main model that is periodically updated. This target model is used when calculating the target Q-value, thereby minimizing parameter oscillation and improving training convergence.

E. Training and Evaluation Procedures

The model was trained with episodic scenarios using pre-sharded historical data. Each episode began with an initial environment state, and the agent was given the opportunity to take actions at each step based on the state obtained from the last five days of data. The training process was carried out over 500 episodes, with a maximum of 150 steps per episode to maintain computational efficiency.

At the start of training, the agent takes actions using an epsilon-greedy policy, where random actions are performed with probability ε (epsilon) to encourage exploration. The initial epsilon value is set at 1.0 and is gradually decreased using a specific decay value (e.g., 0.9995) until it reaches a minimum threshold of 0.01. This ensures the agent gradually shifts from exploration to exploitation as it gains experience.

During training, every interaction between the agent and the environment (state, action, reward, next_state) is stored in a replay buffer. The model is then trained every few steps using random samples from this buffer, making the training data distribution more independent and less time-ordered. With each mini-batch training, the model is updated to minimize the difference between the predicted Q-value and the target Q-value, which is calculated using the target network. The target network itself is periodically updated with weights from the main model.

To measure training performance, cumulative reward and cumulative profit metrics are recorded for each episode, as well as the distribution of agent actions (Buy, Sell, Hold). After the training process is complete, the model is evaluated on the test data. In the evaluation phase, the epsilon value is set to zero to ensure the agent only fully exploits the learned policy.

The evaluation was performed by calculating the total profit and total reward obtained during all steps in the test data, and recording the frequency of each action taken. Furthermore, the Q-value generated by the model when making predictions on the test set was analyzed to determine decision-making tendencies. This evaluation serves as the basis for assessing the agent's ability to make optimal decisions on previously unseen data.

IV. RESULT AND DISCUSSION

After undergoing 500 training episodes, the DRQN model demonstrated its ability to learn stock trading strategies based on historical Sharia-compliant stock index data. Evaluation was conducted in two main phases: during training and after training on testing data.

A. Training Result

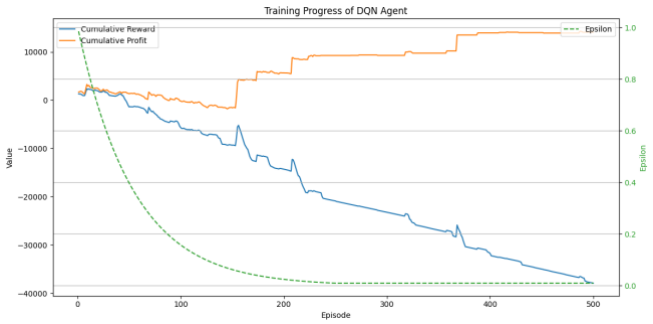

Fig. 1. Training Progress of DQN Agent

Figure 1 shows the cumulative progression of rewards and profits throughout the training process. While rewards experienced a significant downward trend, reaching large negative values, total profits showed a gradual and steady increase until the end of the 500th episode, reaching a total profit of Rp14,056.43 (in units relative to the normalized

scale). This demonstrates that rewards and profits are not always directly correlated, especially in a penalty-based reward environment like the one in this experiment.

The epsilon behavior shown in the dashed green line shows that the epsilon value has successfully decreased to near its minimum limit (0.01), which indicates that the agent has shifted from an exploration to an exploitation strategy.

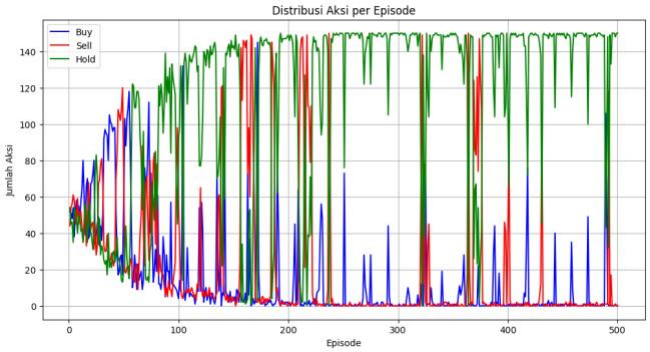

Fig. 2. Distribution of Actions per Episode

The distribution of agent actions per episode, shown in Figure 2, shows that at the beginning of training, the agent explored all action types (buy, sell, and hold) evenly. However, over time, the frequency of holding actions became increasingly dominant, especially after the 100th episode. This can be interpreted as an indication that the model tends to choose to hold positions in uncertain market conditions or when the reward from buying/selling actions is not significant enough to offset the risk.

B. Evaluation on Test Data (Test Set)

After training was complete, the model was evaluated on 380 steps of test data (the last 30% of daily data). The results are visualized in Figures 3 and 4. In Figure 3, the cumulative reward continues to decrease while the cumulative profit remains stagnant at zero. This indicates that the agent did not perform any profitable actions during the testing period.

The distribution of agent actions on the test set, as shown in Figure 4, indicates that all actions taken were hold. No buy or sell actions were taken, which explains why profits remained unchanged during the testing phase.

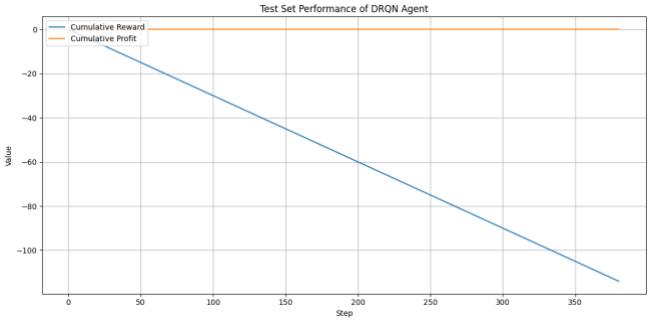

Fig. 3. Test Set Performance of DRQN Agent

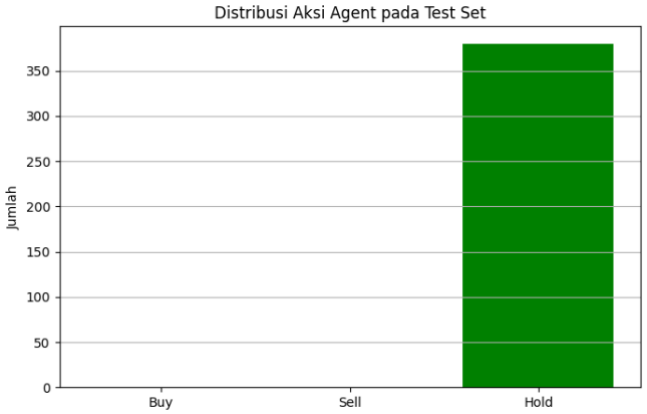

Fig. 4. Distribution of Agent Actions on the Test Set

Furthermore, the Q-values predicted by the model during the testing phase also showed a tendency to choose the action with the highest value, namely holding. The average Q-value (Avg Q) during testing ranged from -8.20 to -8.30, and the maximum Q-value (Max Q) ranged from -6.35 to -6.40. This indicates that, according to the model, all actions have negative estimated rewards, but holding is considered the "least bad" compared to buying or selling.

C. Discussion

The training results show that the DRQN agent achieved a total profit of IDR14,056.43 in 500 training episodes, with a cumulative reward trending negative (-37,915.73). Although the rewards fluctuated and included many negative values, the resulting profit remained positive. This indicates that while the agent's strategy is not always optimal in terms of short-term rewards, it can still identify profitable trading opportunities in the long run. The decrease in reward value is likely influenced by penalties imposed when the agent takes inappropriate actions, such as selling without holding assets or consistently choosing to *hold*.

The distribution of actions during training shows that in the initial phase, the agent is still exploring, randomly trying various actions. As the epsilon value decreases, the agent begins to exploit actions deemed profitable more frequently. However, towards the end of the training process, the agent appears to be choosing the *hold action more frequently*. This could indicate that the agent considers the *hold action* to be the safest or most profitable option based on the Q value it has learned. This could be a sign that the agent's learning process is stalling, or that the *reward function* used is not strong enough to encourage the agent to actively buy and sell.

However, when evaluated on the test data, the agent's performance declined drastically. All actions taken were *hold*, without a single *buy* or *sell action*, resulting in total profit stagnating at 0 and total reward dropping to -114. This phenomenon indicates that the model has not been able to effectively generalize from the training data to the test data. One possible cause is the difference in price movement patterns in the test data, making the learned strategy less relevant when tested, or at least requiring adjustments to be

more adaptive to different market conditions. Furthermore, the designed reward function may not have adequately emphasized the importance of active actions such as buying and selling, resulting in the model learning that *holding* is the safest strategy to minimize penalties, especially when the Q values for all actions tend to be negative.

The DRQN model used relies on an LSTM architecture to understand time series patterns, but the assumption that the last five days (window size = 5) is sufficient to represent the overall market conditions may be too narrow. In the stock market, trends can form over a longer period, so agents lose broader context when making decisions. Furthermore, the Islamic stock index data used is relatively stable and less volatile than individual stocks, so price differences between days are not extreme enough to provide a strong signal for agents to act.

The initial hypothesis that the DRQN model could recognize patterns and generate profitable strategies proved to be limited to the training data. When faced with new market conditions (the test set), the agent tended to adopt a passive approach. This indicates that the model is overfitting to the training data and has not yet learned general trading principles that can be transferred to new conditions. Therefore, further approaches or adjustments to the architecture and reward function are needed to achieve better generalization.

V. CONCLUSION

This study aims to explore the application of the Deep Recurrent Q-Network (DRQN) algorithm in investment decision-making on the Indonesian Sharia-compliant stock index. The model is designed to learn market movement patterns from historical data and generate optimal action strategies (buy, sell, or hold) in a simulated trading context. Experimental results indicate that the DRQN model trained using the last five years of data achieved significant total profits during the training phase. This demonstrates that the LSTM-based reinforcement learning approach can recognize short-term price patterns and utilize them to execute profitable trades, at least under conditions of previously encountered data.

However, when the model was evaluated against test data (data it had never seen before), the agent's performance declined drastically. The agent failed to make active decisions and instead chose to *hold* during testing. As a result, the total profit stagnated at zero, and the overall reward was negative. These results indicate that the model struggled to generalize to new market conditions and potentially *overfitted* to the training data. Furthermore, the reward function used likely did not provide enough explicit incentives to encourage the agent to take bolder actions, such as buying or selling, especially in less volatile market conditions like those of the Islamic stock index.

This study has several limitations. First, the time window used was only five days, which may not adequately represent the overall market trend context. Second, the model only used basic technical features such as price and volume, without considering advanced technical indicators or market sentiment. Third, the evaluation was conducted as a static simulation without considering transaction costs, slippage, or liquidity factors found in real markets.

Nevertheless, this research provides an initial contribution to the use of reinforcement learning in the context of Islamic capital markets, a relatively underexplored area. The developed model can serve as a basis for developing automated decision-making systems that comply with Islamic principles. For future research, it is recommended that the model be trained using longer time windows, more complex and realistic reward functions, and integrated with technical indicators or broader macroeconomic data. Furthermore, other approaches such as Dueling DQN, Double DQN, or attention-based models are also worth exploring to improve agents' ability to understand market dynamics and generate more adaptive and intelligent investment decisions.

REFERENCES

[1] T. Faturohman and T. Nugraha, "Islamic Stock Portfolio Optimization Using Deep Reinforcement Learning," *Journal of Islamic Monetary Economics and Finance*, vol. 8, no. 2, pp. 181–200, May 2022, doi: 10.21098/jimf.v8i2.1430.

[2] T. Théate and D. Ernst, "An application of deep reinforcement learning to algorithmic trading," *Expert Syst Appl*, vol. 173, p. 114632, Jul. 2021, doi: 10.1016/j.eswa.2021.114632.

[3] C. Y. Huang, "Financial Trading as a Game: A Deep Reinforcement Learning Approach," Jul. 2018.

[4] B. Sohet, Y. Hayel, O. Beaude, and A. Jeandin, "Learning Pure Nash Equilibrium in Smart Charging Games," Nov. 2021, doi: 10.1109/CDC42340.2020.9304486.

[5] Z. Ding, Y. Huang, H. Yuan, and H. Dong, "Introduction to Reinforcement Learning," in *Deep Reinforcement Learning*, Singapore: Springer Singapore, 2020, pp. 47–123. doi: 10.1007/978-981-15-4095-0_2.

[6] J. L. Tan, B. A. Taha, N. A. Aziz, M. H. H. Mokhtar, M. Mukhlisin, and N. Arsad, "A Review of Reinforcement Learning Evolution: Taxonomy, Challenges and Emerging Solutions," *International Journal of Advanced Computer Science and Applications*, vol. 16, no. 1, 2025, doi: 10.14569/IJACSA.2025.0160149.

[7] Y. Bai, Y. Gao, R. Wan, S. Zhang, and R. Song, "A Review of Reinforcement Learning in Financial Applications," Nov. 2024.

[8] D. I. León Nieto, "Reinforcement learning for finance: A review," *ODEON*, no. 24, pp. 7–24, Nov. 2023, doi: 10.18601/17941113.n24.02.

[9] A. Mohammadshafie, A. Mirzaeinia, H. Jumakhan, and A. Mirzaeinia, "Deep Reinforcement Learning Strategies in Finance: Insights into Asset Holding, Trading Behavior, and Purchase Diversity," Jun. 2024.

[10] E. Mienye, N. Jere, G. Obaido, I. D. Mienye, and K. Aruleba, "Deep Learning in Finance: A Survey of Applications and Techniques," *AI*, vol. 5, no. 4, pp. 2066–2091, Oct. 2024, doi: 10.3390/ai5040101.

[11] P. Yu, J. S. Lee, I. Kulyatin, Z. Shi, and S. Dasgupta, "Model-based Deep Reinforcement Learning for Dynamic Portfolio Optimization," Jan. 2019.

[12] X. Jiang, "Comparison of Deep Reinforcement Learning Algorithms for Trading Strategy," 2024, pp. 4–14. doi: 10.2991/978-94-6463-370-2_2.

[13] D. Saepudin and K. Rauf, "Application of Deep Reinforcement Learning for Stock Trading on The Indonesia Stock Exchange," *Jurnal Nasional Pendidikan Teknik Informatika (JANAPATI)*, vol. 14, no. 1, pp. 144–157, Mar. 2025, doi: 10.23887/janapati.v14i1.83775.

[14] M. Kong and J. So, "Empirical Analysis of Automated Stock Trading Using Deep Reinforcement Learning," *Applied Sciences*, vol. 13, no. 1, p. 633, Jan. 2023, doi: 10.3390/app13010633.

[15] H. Yang, X. Y. Liu, S. Zhong, and A. Walid, "Deep reinforcement learning for automated stock trading: An ensemble strategy," in *ICAIF*

2020 - 1st ACM International Conference on AI in Finance, Association for Computing Machinery, Inc, Oct. 2020. doi: 10.1145/3383455.3422540.

[16] S. Ouf, M. El Hawary, A. Aboutabl, and S. Adel, "A Deep Learning-Based LSTM for Stock Price Prediction Using Twitter Sentiment Analysis," *International Journal of Advanced Computer Science and Applications*, vol. 15, no. 12, 2024, doi: 10.14569/IJACSA.2024.0151223.

[17] A. Sarkar and G. Vadivu, "An Advanced Ensemble Deep Learning Framework for Stock Price Prediction Using VAE, Transformer, and LSTM Model," Mar. 2025.

[18] X. Chen, Q. Wang, L. Yuxin, C. Hu, C. Wang, and Q. Yan, "Stock Price Forecast Based on Dueling Deep Recurrent Q-network," in *2023 IEEE 6th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*, IEEE, Aug. 2023, pp. 1091–1096. doi: 10.1109/PRAI59366.2023.10332127.

[19] F. Sarlakifar, M. M. Asl, S. R. Khaledi, and A. Salimi-Badr, "A Deep Reinforcement Learning Approach to Automated Stock Trading, using xLSTM Networks," Mar. 2025.

[20] M. S. Iqbal, F. A. M. S. B. Sukamto, S. N. B. Norizan, S. Mahmood, A. Fatima, and F. Hashmi, "AI in Islamic finance: Global trends, ethical implications, and bibliometric insights," *Review of Islamic Social Finance and Entrepreneurship*, pp. 70–85, Mar. 2025, doi: 10.20885/RISFE.vol4.iss1.art6.

[21] Rajesh Dey, "Applications of Machine Learning in Islamic Finance," *Journal of Information Systems Engineering and Management*, vol. 10, no. 26s, pp. 794–804, Mar. 2025, doi: 10.52783/jisem.v10i26s.4286.

[22] E. R. Kismawadi, M. Irfan, and I. Harahap, "Integrating Artificial Intelligence in Islamic Financial Management: Opportunities and Challenges in Maintaining Shariah Compliance," in *Indigenous Empowerment through Human-Machine Interactions*, Emerald Publishing Limited, 2025, pp. 273–288. doi: 10.1108/978-1-83608-068-820251016.

[23] M. M. Uula and S. F. M. Kassim, "Machine Learning in Islamic Finance," *Islamic Economics Methodology*, vol. 3, no. 2, Feb. 2025, doi: 10.58968/iem.v3i2.595.

[24] B. Charoenwong, "Shariah-Compliant Investing in the Machine Age: Equity Classifications with Machine Learning," *SSRN Electronic Journal*, 2023, doi: 10.2139/ssrn.4611253.

[25] I. B. Adeyemi and Ö. F. Tekdoğan, "Understanding the Screening Criteria for Shariah-Compliant Stocks," *Adam Akademi Sosyal Bilimler Dergisi*, vol. 14, no. 2, pp. 371–395, Dec. 2024, doi: 10.31679/adamakademi.1415744.

[26] H. Shalhoob, "The role of AI in enhancing shariah compliance: Efficiency and transparency in Islamic finance," *Journal of Infrastructure, Policy and Development*, vol. 9, no. 1, p. 11239, Jan. 2025, doi: 10.24294/jipd11239.

[27] H. Shalhoob, "The role of AI in enhancing shariah compliance: Efficiency and transparency in Islamic finance," *Journal of Infrastructure, Policy and Development*, vol. 9, no. 1, p. 11239, Jan. 2025, doi: 10.24294/jipd11239.

[28] H. Shalhoob and I. Babiker, "Exploration of AI in Ensuring Sharia Compliance in IF Institutions: Focus on Accounting Practices," *Open Journal of Business and Management*, vol. 13, no. 02, pp. 1435–1448, 2025, doi: 10.4236/ojbm.2025.132075.

[29] "Global Islamic Fintech Report," 2024.

[30] M. S. Iqbal, F. A. M. S. B. Sukamto, S. N. B. Norizan, S. Mahmood, A. Fatima, and F. Hashmi, "AI in Islamic finance: Global trends, ethical implications, and bibliometric insights," *Review of Islamic Social Finance and Entrepreneurship*, pp. 70–85, Mar. 2025, doi: 10.20885/RISFE.vol4.iss1.art6.

[31] Y. Bai, Y. Gao, R. Wan, S. Zhang, and R. Song, "A Review of Reinforcement Learning in Financial Applications," Nov. 2024.

[32] Y. Huang, C. Zhou, L. Zhang, and X. Lu, "A Self-Rewarding Mechanism in Deep Reinforcement Learning for Trading Strategy Optimization," *Mathematics*, vol. 12, no. 24, p. 4020, Dec. 2024, doi: 10.3390/math12244020.