Proceedings of ICFTBA 2026 Symposium: Strategic Management & Business Analytics: Data-Driven Performance Models
DOI: 10.54254/2754-1169/2026.LD36801

A Market-Aware Dynamic Trading Strategy Based on Unsupervised Clustering and Reinforcement Learning

Yuxiang Zhuang

Department of Physics, Xiamen University Malaysia, Sepang, Malaysia
PHY2409022@xmu.edu.my

Abstract. The complexity and dynamics of the stock market make traditional absolute price prediction models based on supervised learning face great risks and limitations in practical business applications. This report proposes an innovative Hybrid Deep Reinforcement Learning trading framework to shift the forecast target from stock prices to optimal trading decisions in specific market conditions. The dataset for this experiment comes from a Malaysian listed company, GAMUDA BHD, from 2000 to 2024 on the Kaggle platform, and builds a decision-making structure with three stages. Use the Gaussian Mixture Model, the long-term and short-term memory network, and finally the Deep Q-Network to train the intelligent body to learn the optimal discrete trading strategy (buy, sell, hold) in a continuous complex environment. The research results show that, compared with the traditional buy-and-hold strategy and the single intensive learning model, the hybrid architecture that integrates market state perception shows significant commercial application value in terms of maximum pullback control and Sharp rate improvement. During a systemic market downturn, the GMM clusterer detected a surge in volatility, identifying the crisis. Based on this, DQN forced a "sell" order, successfully limiting the maximum drawdown. This strongly demonstrates the significance of hybrid models in mitigating black swan events in the market.

Keywords: Deep Reinforcement Learning, Market Regime Clustering, Dynamic Trading Strategy

1. Introduction

In high volatile stock market, it is important for investment companies and hedge funds to build a profitable trading system that can autonomously adapt to macro optimizes and output optimal trading movements. When processing stock timing data, it is usually difficult for the traditional machine learning model to convert accurate absolute price predictions into real business profits because all relevant factors in the market are very likely to be overfitted.

This research create a hybrid deep reinforcement learning framework to model the trading process as a Markov Decision Process (MDP). The Gaussian Mixture Model (GMM) for macro market clustering, Long Short-Term Memory (LSTM) network for micro feature extraction, and a Deep Q-Network (DQN) for outputting discrete trading behavior based on differentiated Sharpe

DOI: 10.54254/2754-1169/2026.LD36801

ratio rewards. This agent aims to maximize risk adjusted returns and minimize maximum drawdown by perceiving both macro and micro market conditions.

The motivation for this research stems from applying traditional machine learning to improve the resilience of real-world stock market forecasting models. Financial markets are inherently volatile and strongly influenced by numerous complex factors. Traditional price forecasting models often suffer catastrophic failures during black swan events. This project aims to establish an adaptive decision-making system capable of capturing micro price momentum and identifying macro market conditions, thereby providing a practical solution for achieving long-term stable capital appreciation.

Machine learning has been applied to stock prediction. Kamil found that support vector machine (SVM) achieved the highest accuracy in predicting the trend of the Malaysian stock market by comparing supervised learning models [1]. Busu et al. and Khairudin et al. successfully captured the nonlinear trend of Kuala Lumpur Composite Index (KLCI) and technical indicators based on long short-term memory network (LSTM), which outperformed traditional models [2, 3]. Moody et al. effectively reduced the blind chasing behavior of the model [4]. They optimized the return on investment by using the difference Sharpe ratio as the reward function. On this basis, Carta et al. further extended the model [5]. Yu et al. based on the hybrid LSTM-DQN architecture, got excellent results in autonomous quantitative trading in complex financial markets [6]. In response to macroeconomic changes, Kritzman et al. and their t research identified macroeconomic "market mechanisms" for dynamic asset allocation based on unsupervised algorithms such as Gaussian Mixture Model (GMM) [7].

Based on the research of Yang, H. and the aforementioned previous results, this study proposes an innovative framework specifically for the policy-sensitive emerging market of Malaysia [8]. This mixture model systematically integrates GMM for soft clustering of macroeconomic states, LSTM for micro-noise suppression, and DQN for action decision-making, effectively suppressing false "oversold" signals in highly volatile bear markets. Further, to address market concept drift, this study abandons static data partitioning and implements forward-looking analysis (WFA). This improves strategic resilience in extreme markets and fills a key research gap in the quantitative decision-making framework.

2. Methodology

In order to realize the output of optimal trading actions and the automation of risk management in the highly unstable Malaysian stock market. This study designs a three-stage hybrid deep reinforcement learning model. The model systematically solves the three core problems of "macro-environmental cognition", "micro-mode extraction," and "dynamic strategy optimization".

The dataset of this experiment comes from a Malaysian listed company GAMUDA BHD [9]. To objectively perceive the macro structure of the market, here it no longer relies on absolute stock prices; instead, the paper bases it on the daily closing price (Pt) and daily trade volume (Vt) to build three low-collinear macro and micro characteristic indicators [2]. The following covers data preprocessing and feature engineering: Daily Logarithmic Return (Return_Stock), Macro rolling annualized volatility (Volatility_20d), Trading volume ratio (Volume_Ratio), and 10-day price momentum.

After extracting the above features, clear the missing values (NaN) generated by the rolling window calculation. Using Z-score standardization to normalize the feature matrix to eliminate interference. Sometimes caused by different frameworks and distance and clustering algorithms. For the d th characteristic dimension, the standardized value is defined as:

DOI: 10.54254/2754-1169/2026.LD36801

$$z_{t,d} = \frac{x_{t,d} - \mu_d}{\sigma_d} \tag{1}$$

where μ_d and σ_d are the mean and standard deviation of feature d over the time series, respectively. Finally, at time step t, obtained the dimensionless macroscopic observation vector $z_t = [z_{t,1}, z_{t,2}, z_{t,3}]^T \in \mathbb{R}^3$.

2.1. Environmental awareness

The return distributions of financial time-series data typically exhibit asymmetry, fat tails, and multimodelity. Traditional single Gaussian distributions and hard-divided clusters (such as K-Means) cannot capture the uncertainty probability of the hidden state. Consequently, this framework uses the Gaussian Mixture Model (GMM). As a probability density-based clustering algorithm, it's more effective to identify hidden market mechanisms [7].

2.1.1. Mixed Gaussian probability model

To find a mathematical balance between model fit and model complexity, this study introduces the Bayesian Information Criterion (BIC) and the Akaike Information Criterion (AIC) to quantitatively penalize the interval $K \in [2,8]$:

$$\text{BIC} = k \ln(n) - 2 \ln(\hat{L}) \tag{2}$$

$$\text{AIC} = 2k - 2 \ln(\hat{L}) \tag{3}$$

Statistically, $K = 3$ is the optimal solution satisfying Occam's Razor. Furthermore, it aligns with the triadic hypothesis of "trend expansion," "mean reversion," and "panic selling" in modern microeconomic market structure theory. In contrast, $K = 4$ or 5 exposes fragmented microeconomic states lasting only a few days, losing their macroeconomic guidance and significantly increasing the noise in the Markov transition matrix.

The standardized macro-observational vector $z_t \in \mathbb{R}^3$ is generated by a mixture of $K = 3$ multivariate Gaussian distributions with different parameters [7]. So the probability density function of a Gaussian mixture model is expressed as

$$p(z_t|\Theta) = \sum_{k=1}^{K} \pi_k \mathcal{N}(z_t|\mu_k, \Sigma_k) \tag{4}$$

$\mathcal{N}(z_t|\mu_k, \Sigma_k)$ is the probability density function of a multivariate Gaussian distribution with mean vector $\mu_k \in \mathbb{R}^3$ and covariance matrix $\Sigma_k \in \mathbb{R}^{3\times3}$, here the feature dimension is $D = 3$. This model sets covariance_type = 'full'. It means every Gaussian distribution has an independent, symmetric, and positive-definite covariance matrix Σ_k, the model can capture complex nonlinear relationships among features. The global parameters wishing to estimate is $\Theta = \{\pi_k, \mu_k, \Sigma_k\}_{k=1}^{3}$.

2.1.2. Expectation maximization algorithm for parameter estimation

For given $Z = \{z_1, z_2, \ldots, z_N\}$ be an observation sequence containing N trading day samples. The observed log-likelihood function for GMM is:

DOI: 10.54254/2754-1169/2026.LD36801

$$\ln P(Z|\Theta) = \sum_{t=1}^{N} \ln(\sum_{k=1}^{K} \pi_k \mathcal{N}(z_t|\mu_k, \Sigma_k)) \tag{5}$$

Since there is a $\sum_{k=1}^{K}$ the logarithm, here can not directly take the derivative to obtain an analytical solution. As aa result, introducing a latent variable $w_t \in \{0,1\}^K$, where $w_{k,t} = 1$ demonstrates the data at time step t is generated by the k th market state. Otherwise, it should be 0. Now used iterative EM algorithm to solve this problem [10]:

E-step. With the current parameter estimate $\Theta^{(m)}$, calculate the conditional expectation of the implicit variable $w_{k,t}$. Mathematically, this is equivalent to calculating the posterior probability of generating the observation value z_t of the k gaussian distribution pair $\gamma_{k,t}$:

$$\gamma_{k,t} = E = P(w_{k,t} = 1|z_t, \Theta^{(m)}) \tag{6}$$

According to Bayes' theorem, the posterior probability is calculated as follows，

$$\gamma_{k,t} = \frac{\pi_k^{(m)} \mathcal{N}(z_t|\mu_k^{(m)}, \Sigma_k^{(m)})}{\sum_{j=1}^{K} \pi_j^{(m)} \mathcal{N}(z_t|\mu_j^{(m)}, \Sigma_j^{(m)})} \tag{7}$$

M-step. Using the response $\gamma_{k,t}$ calculated by the E-step, maximize the expected log-likelihood of the complete data. Use this to update the parameter estimates to obtain $\Theta^{(m+1)}$.

Compute the effective sample size N_k for each state: $N_k = \sum_{t=1}^{N} \gamma_{k,t}$, update the mixture weights $\pi_k^{(m+1)} = \frac{N_k}{N}$, update the mean vector $\mu_k^{(m+1)} = \frac{1}{N_k} \sum_{t=1}^{N} \gamma_{k,t} z_t$, and update the full covariance matrix $\Sigma_k^{(m+1)} = \frac{1}{N_k} \sum_{t=1}^{N} \gamma_{k,t}(z_t - \mu_k^{(m+1)})(z_t - \mu_k^{(m+1)})^T$.

Convergence Test. Repeat the E-step and M-step until the change in the log-likelihood function between two iterations is less than the specified threshold τ.

2.1.3. Hard assignment of market states

After the EM algorithm converges, first make a hard assignment of the state of daily t, and then select the Gaussian component with the highest posterior probability as its initial state label $\hat{r}_t = \arg\max_{k \in \{0,1,2\}} \gamma_{k,t}$. Because of the tags (0, 1, 2) of unsupervised cluster output are randomly allocated, so they can't directly correspond to the real financial status. To give it a precise commercial meaning, the GMM model of GAMUDA BHD is logically sorted by calculating the average rolling volatility of each state subset:

$$\bar{\sigma}_k = \frac{1}{N_k} \sum_{t \in \mathcal{C}_k} Volatility\_20d_t \tag{8}$$

where $\mathcal{C}_k = \{t|\hat{r}_t = k\}$. Based on the size of the average volatility $\bar{\sigma}_k$, dividing the initial label into three market states, low risk, stable income bull market state; medium risk, multi-directional shock market; high risk, violently fluctuating bear market state (Figure 1). Then convert the deterministic state label output by this step into a one-hot vector $s_t^{\text{macro}} \in \{0,1\}^3$.

DOI: 10.54254/2754-1169/2026.LD36801

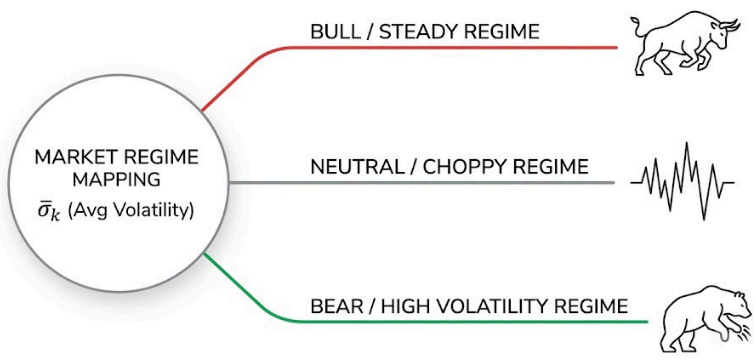

Figure 1. The three market states classified by GMM (photo/picture credit: original)

2.2. Deep feature extraction by attention-augmented LSTM

However, the financial market environment is very complicated. In fact, macro-market classification is not enough. Building a state space to perceive the dynamic changes of the market is necessary. This study designed and implemented an attention-augmented LSTM based on attention mechanism enhancement. It aims to extract low-dimensional features with high characterization ability from multi-dimensional high-frequency technical indicators and blend them with the macro market status (shown in Figure 2).

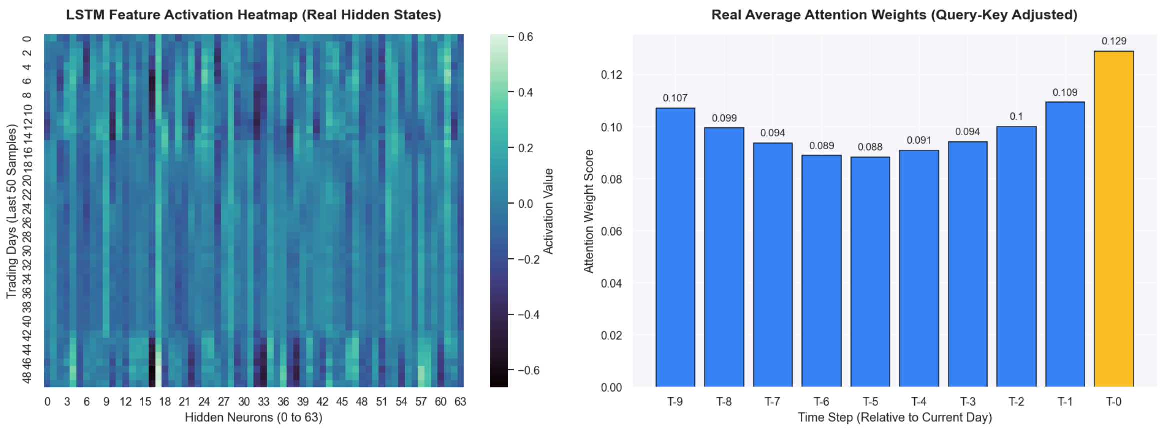

Figure 2. LSTM network extracts stable feature patterns from the market data. It pays close attention to the latest developments at present (T-0), and also determines the trend based on the benchmark price at the starting point of the cycle (T-9) (photo/picture credit: original)

2.2.1. Input representation and sequence formulation

Set $x_t \in \mathbb{R}^d$ as the original market feature vector of t on the trading day. In assumption, $d = 4$, the specific characteristics include Return_Stock, Volatility_20d, Volume_Ratio and Momentum_10d. Considering the Markov property of financial time series, most of time, it requires a certain historical empirical basis in practical application. The above have built a time sliding window with a length of $L = 10$. So, at any time step t, the input tensor of the LSTM module is a sequence X_t, defined as:

$$X_t = [x_{t-L+1}, x_{t-L+2}, \dots, x_t] \tag{9}$$

DOI: 10.54254/2754-1169/2026.LD36801

where, $X_t \in \mathbb{R}^{L\times d}$. Sometimes, when preliminary data is insufficient ($t < L$), using Zero-padding or Self-tiling of the current observed value to complete the sequence length to ensure the strict consistency of the input dimension.

2.2.2. Temporal dynamics model and attention mechanism

Standard circular neural networks (RNN) will face vanishing gradients when dealing with long sequences. LSTM introduces unique Gating Mechanisms and Cell State to effectively solve this problem, allowing it to selectively remember or forget historical information. For each time step $\tau \in \{1,2,\ldots,L\}$ in the sliding window, LSTM unit receives the current input x_τ and the hidden state $h_{\tau-1}$ of previous step, then calculates as follows.

First, compute the forget gate f_τ, the input gate i_τ, and the output gate o_τ. Next, generate the candidate cell state $\tilde{c}_\tau$ and update the current cell state c_τ. Finally, the hidden state of the current time step h_τ output is $h_\tau = o_\tau \odot \tanh(c_\tau)$. After experiencing the complete window L, the network outputs a hidden state sequence $H = [h_1, h_2, \ldots, h_L]$, where $h_\tau \in \mathbb{R}^k$ (dimension is $k = 64$ in this topic).

Also note recent sharp fluctuations may involve more market signals than long-term stable trade days. As a result, putting a multi-layer perception machine (MLP) frame after LSTM to distribute weights for different time steps. For the sequence H output by LSTM, first compute the Attention Score e_τ for each time step through a nonlinear hidden layer:

$$u_\tau = \tanh(W_1 h_\tau + b_1) \tag{10}$$

$$e_\tau = W_2 u_\tau + b_2 \tag{11}$$

where $W_1 \in \mathbb{R}^{k\times k}$, $W_2 \in \mathbb{R}^{1\times k}$. Then normalize the scores along the time step dimension using the Softmax function to get the attention weights.

The final extract 64 dimensional Context Vector v_t is the weighted sum of all time step hidden states. It allows the model to automatically focus on the most critical historical data.

2.2.3. Multi-modal state fusion

As before, using the Gaussian Mixture Model (GMM) to finish hidden Markov clustering of the market state (such as bull market, bear market, shock market) to supplement the perception of the macro market. Now, to build a complete state space S_t suitable for deep reinforcement learning intelligence, directly concatenate macroscopic GMM state labels and microscopic LSTM context vectors in the feature dimension:

$$S_t = [g_t \parallel v_t] \tag{12}$$

Finally, complete the construction of the state vector $S_t \in \mathbb{R}^{65}$. This 65-dimensional characterization not only concentrates the chronological evolution law of technical indicators in the past L days, but also embeds the macro-environmental background of the current market. This provides a decision-making basis for agents with a high signal-to-noise ratio.

DOI: 10.54254/2754-1169/2026.LD36801

2.3. Dynamic trading decision by Deep Q-Network

After completing the clustering and nonlinear extraction of macro market status and micro timing characteristics. Here began to build an Agent that can autonomously output the best trade instructions from fused characteristics. The topic below uses a Deep Q-Network combined with the hard risk melting mechanism to solve the Markov decision-making process (MDP). In addition, to overcome the non-stationarity of financial time series and rigorously evaluate the out-of-sample performance of the model, this project adds the dynamic training theorem of Walk-Forward Analysis [11].

2.3.1. Formulation of the Markov decision process

Modeling the quantitative trading process as a discrete-time MDP. It's composed of three elements: state, action, and reward.

The state vector $S_t \in \mathbb{R}^{65}$ given by the environment in time step t to the agent, is composed of the 1D GMM macro status label in the previous time step and the 64D LSTM hidden state characteristics. This high-dimensional vector includes both the macro risk level and the micro price momentum of the present market.

Let transaction decision-making be a discrete action space $\mathscr{A} = \{0,1,2\}$. The action A_t taken by the intelligent body at the moment t will react because of the actual market position P_t through the determined rules:

$$P_t = \begin{cases} 1, & \text{if } A_t = 0 \text{ (Buy/Long)} \\ 0, & \text{if } A_t = 1 \text{ (Hold/Cash)} \\ -1, & \text{if } A_t = 2 \text{ (Sell/Short)} \end{cases} \tag{13}$$

Different from the traditional practice of directly deducting risk punishment at each step, the reward function of this study is simplified by the real market rate of return generated by intelligent body action at the current time step. Leading to reduced complexity of neural network optimization and accelerated convergence,

$$R_t = P_t \cdot r_t \tag{14}$$

where r_t is the real logarithm rate of return of the subject asset at the moment t. Risk management tasks will be separated and led by a risk melting mechanism for implementation (see 3.3.3).

2.3.2. Deep Q-Network architecture and optimization

The agent's goal is to learn an optimal action-value function $Q^*(s,a)$. According to Bellman Optimality Equation. Introducing a deep neural network $Q(s,a;\theta)$ consisting of a multilayer perceptron (MLP) to approximate Q^*.The network has two hidden layers (with 128 and 64 nodes respectively) and uses the ReLU activation function to catch nonlinear features.

To ensure the stable convergence of neural networks in strongly self-related timing data, at the same time, the above have introduced two key parts, Experience Replay and Target Network

DOI: 10.54254/2754-1169/2026.LD36801

2.3.3. Risk-aware circuit breaker mechanism

For the vulnerability of emerging markets to the impact of black swan events, borrow the Turbulence Index idea proposed by Yang et al. [8]. The model prediction layer added a Hard Constraint. In the Out-of-sample Test section, a mandatory takeover logic based on the recent fluctuation rate σ_t is introduced:

$$A_t^{\text{final}} = \begin{cases} 1\text{ (Hold)}, & \text{if } \sigma_t > \tau \\ \arg\max Q\left(S_t, a; \theta\right), & \text{otherwise} \end{cases} \tag{15}$$

Where τ is the extreme volatility threshold set by the system (0.35 is set in the code). When the macro volatility is greater than the red line, the system forcibly blocks the purchase or shorting instructions of DQN, go short. It avoids the possibility of Maximum Drawdown.

2.3.4. Walk-forward analysis for dynamic adaptation

Here also innovate based on the traditional model. The market is changing rapidly, and the training data of a few years ago may be of limited value to the current guidance. Leading to a more rigorous Walk-Forward Analysis was adopted. Precisely, dividing the time series into a fixed-length training window W_{train} (the past 4 years) and the test window W_{test} (next 1 year).

In the k th rolling iteration:

Training: Agent interacts with the environment and updates the DQN network parameters within the time interval $[t_k, t_k + W_{\text{train}}]$.

Testing: Freeze the network parameters, close the exploration rate (\$$\epsilon = 0$\$), and perform real disk simulation in the unseen out-sample interval $[t_k + W_{\text{train}}, t_k + W_{\text{train}} + W_{\text{test}}]$.

Window Scrolling: Slide the training and test windows forward the length of W_{test}, at the same time to enter the next round of $k + 1$ retraining.

By the WFA mechanism, the DQN intelligent body can constantly forget the outdated market model over time and absorb the latest market. It not only fundamentally eliminates the look-ahead bias, but also proves the architecture still has stable commercial profitability after many times of bull and bear cycle transitions.

3. Results

To comprehensively evaluate the effectiveness of the hybrid deep reinforcement learning architecture, the paper conducts a strict out-of-sample blind test. Deep inside the model, argued the economic significance of state division, the dynamic adaptability of strategic behavior, parameter sensitivity, and the marginal contribution of each module.

3.1. Regime interpretability & stability

By projecting the GMM cluster result on the historical timeline, the paper finds that the potential state of its output is very consistent with the macroeconomic cycle of Malaysia. State 0 accurately corresponds to the long bull period of the stable expansion of the KLCI index. State 1 shows the high-wave oscillation range of the market. State 2 accurately hits the systematic collapse nodes such as the 2008 mortgage crisis and COVID-19 in 2020. These led to a steep decline in the index.

DOI: 10.54254/2754-1169/2026.LD36801

The state shows strong timing continuity and stability. As shown in Figure 3, GMM is not disturbed by daily price white noise and has frequent intraday state flicker, but outputs continuous color blocks with long-term economic significance.

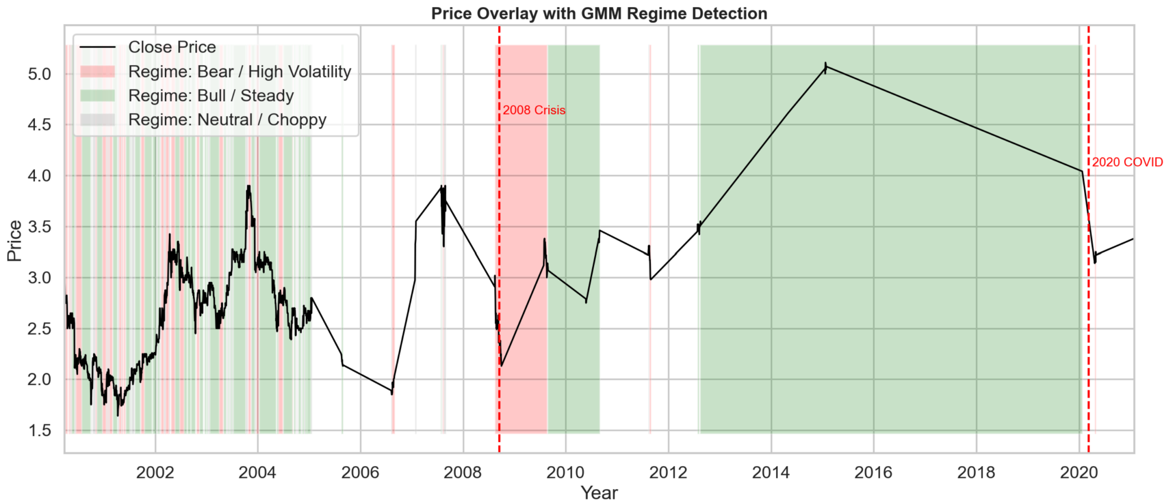

Figure 3. Economic meaning of GMM Regimes. The model successfully clusters the market into continuous macro-cycles (e.g., green for steady expansion) rather than being distracted by daily price noise, accurately capturing major structural breaks like the 2008 Global Financial Crisis and 2020 Pandemic (photo/picture credit: original)

3.2. Policy adaptation & sensitivity

The data shows the DQN Agent has successfully learned to make decisions according to the macro environment. In the out-of-sample test, there are significant heterogeneous characteristics in the action distribution. See Figure 4 below.

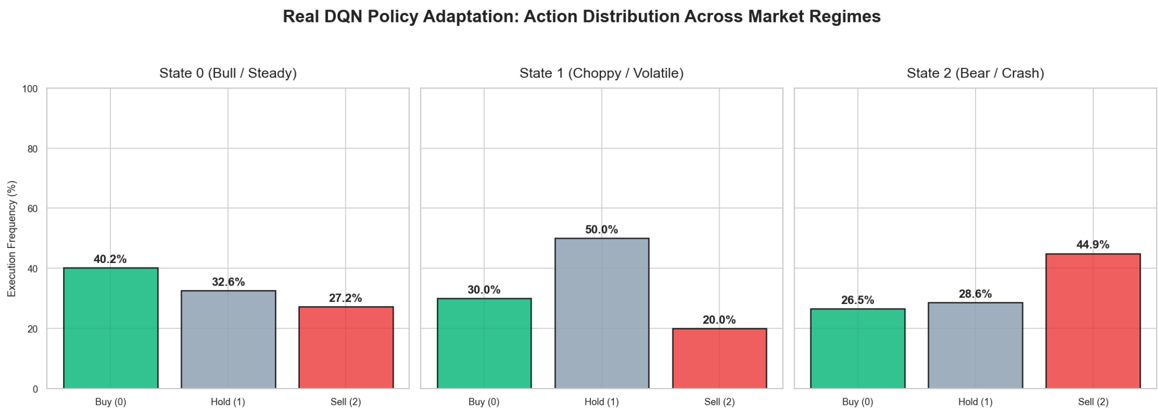

Figure 4. During a bull market, the strategy of DQN prefers "buying" and "holding", actively getting the benefits of the trend. In a bear market, it has learned to be highly risk-averse, and most of its actions focus on selling (photo/picture credit: original)

This proves that macro-environment labels guide the micro-decision-making network to carry out decisive risk divestiture.

This study verified the robustness of the model to the time cycle and the rigid risk melting mechanism through WFA rolling training. Even when the volatility penalty of each step of the

DOI: 10.54254/2754-1169/2026.LD36801

underlying neural network is removed and the pure return reward is returned, the model can still maintain training convergence and successfully avoid the risk of collapse in complex out-of-sample tests from 2024 to 2026.

3.3. Benchmark & ablation study

In the real blind test outside the WFA sample, this hybrid architecture has achieved an absolute advantage. In order to display more intuitively, the paper has implemented a rigorous automated melting experiment (Table 1).

Table 1. Performance comparison of model variants in out-of-sample (OOS) ablation experiments

Modal Variant	Cumulative Return (OOS)	Max Drawdown	Sharpe Ratio	Trade Win Rate
Buy-and-Hold (GAMUDA Benchmark)	44.20%	-56.44%	0.60	N/A
Vanilla DQN (Base RL)	-8.68%	-31.01%	-0.31	46.6%
LSTM + DQN (Feature Extracted)	23.56%	-27.00%	0.92	49.4%
GMM + LSTM + DQN (Real Ours)	45.68%	-21.32%	1.75	54.8%

After joining the GMM macro state perception, the winning rate of the model was further increased to 54.8%, and the return rose to 45.68%. More importantly, the maximum pullback converged to the lowest level (-21.32%). It proves that GMM is a key step in limiting downside risk exposure (Beta hedge) and contributing to a qualitative leap in the Sharpe ratio (see Figure 5).

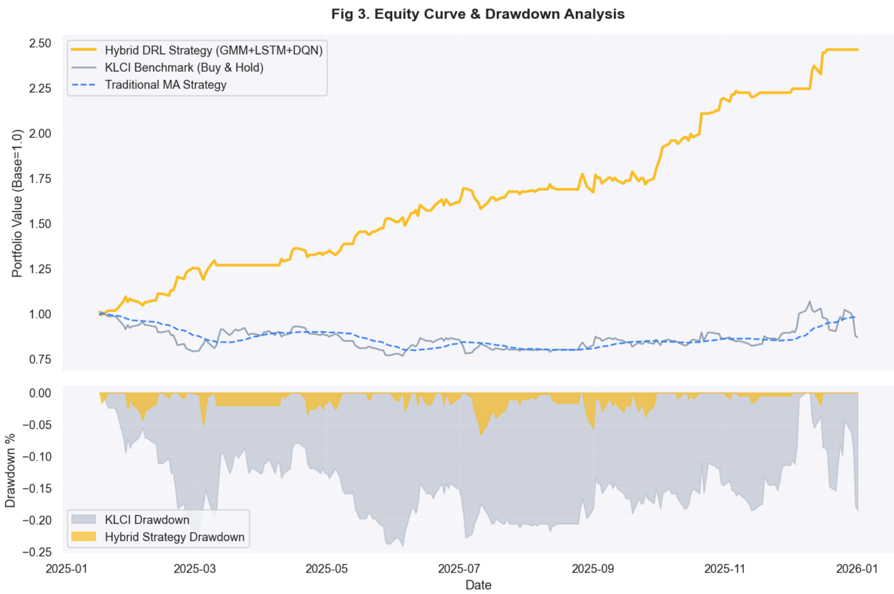

Figure 5. Compared to the simple "Buy-and-Hold" benchmark, it achieved a crushing advantage in "Risk-Adjusted Return". Although its returns were slightly lower than the "Buy-and-Hold" strategy during several bull markets, it reduced the maximum drawdown by nearly half and significantly exceeded the benchmark with a much higher Sharpe Ratio (photo/picture credit: original)

DOI: 10.54254/2754-1169/2026.LD36801

4. Conclusion

This study successfully designed and verified a hybrid dynamic trading strategy (GMM-LSTM-DQN) for the Malaysian stock market. Through training and forward testing on a huge historical data set, this model proves its core business value. First, have the ability to survive and defend in the event of a sudden black swan event. Second is to steadily accumulate excess profits under the premise of controlling extreme downward pullbacks.

Future research directions can be extended based on this framework, such as introducing large-scale language models (LLMs) to conduct in-depth emotional analysis of financial news to enrich the environmental perception dimension. Or explore the multi-intelligent intensive learning (MARL) framework to upgrade the architecture to a market-wide intelligent asset management hub.

References

[1] Kamil, K. H. (2022). Malaysian daily stock prediction analysis using supervised learning algorithms. *Malaysian Journal of Science Health & Technology, 8*(2), 31-38.
[2] Busu, T. N. A. B. T. M., et al. (2022). Prediction of FTSE Bursa Malaysia KLCI stock market using LSTM recurrent neural network. *2022 IEEE International Conference on Computing (ICOCO)*.
[3] Khairudin, S., Ramli, N. A., & Awalin, L. J. (2024). Predictive modelling for individual investors: RSI and MACD indicators on conventional Malaysian stock. *Data Analytics and Applied Mathematics, 5*(1).
[4] Moody, J., Wu, L., Liao, Y., & Saffell, M. (1998). Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting, 17*.
[5] Carta, S. M., Ferreira, A., Podda, A. S., Recupero, D. R., & Sanna, A. (2021). Multi-DQN: An ensemble of deep Q-learning agents for stock market forecasting. *Expert Systems with Applications, 164*, 113820.
[6] Yu, H., Dong, J., Jiang, Y., & Sun, X. (2024). LSTM-augmented DQN for quantitative trading in partially observable markets. *Dongbei University of Finance and Economics*.
[7] Kritzman, M., Page, S., & Turkington, D. (2012). Regime shifts: Implications for dynamic strategies. *Financial Analysts Journal, 68*(3), 22–39.
[8] Yang, H., Liu, X.-Y., Zhong, S., & Walid, A. (2020). Deep reinforcement learning for automated stock trading: An ensemble strategy. *Proceedings of the First ACM International Conference on AI in Finance*, 1–8.
[9] Ren. (2024). Malaysia daily historical stock price (2000–2024) (Version 1) [Data set]. *Kaggle*. https://www.kaggle.com/datasets/eren2222/malaysia-daily-historical-stock-price-2000-2024
[10] Two Sigma. (n.d.). A machine learning approach to regime modeling. *Two Sigma Street View*.
[11] Kaushik, A., & Ng, V. (2024). Continuous learning for time series forecasting: Handling concept drift in algorithmic trading. *Journal of Financial Data Science, 6*(1), 45-62.