Deep Reinforcement Learning for Goal-Based Investing Under Regime-Switching

Tessa Bauman*[1], Bruno Gašperov[1], Sven Goluža[1], and Zvonko Kostanjčar[1]

[1]Faculty of Electrical Engineering and Computing, University of Zagreb
{tessa.bauman, bruno.gasperov, sven.goluza, zvonko.kostanjcar}@fer.hr

Abstract

Goal-based investing focuses on helping investors achieve specific financial goals, shifting away from the volatility-based risk paradigm. While numerous methods exist for this type of problem, the majority of them struggle to properly capture the non-stationary dynamics of real-world financial markets. This paper introduces a novel deep reinforcement learning framework for goal-based investing that addresses market non-stationarity through prompt reactions to regime switches. It relies on the integration of regime probability estimates directly into the state space. The experimental results indicate that the proposed method significantly outperforms several benchmarks commonly used in goal-based investing.

1 Introduction

Goal-based investing (GBI) constitutes approaches to investing that focus on helping investors attain their well-defined short- and long-term financial goals through portfolio management [1]. For example, a long-term investor might desire to reach a target wealth level by the time she retires. The resulting objective can simply be expressed as a binary function indicating whether the investment goal has been achieved. Under such a paradigm, risk is defined as the probability of not attaining the desired goal(s). This stands in stark contrast to classical portfolio optimization approaches, typically based on mean-variance optimization [2], where risk is represented by price volatility, with upside and downside price movements treated equivalently. Given that GBI requires dynamically responding to time-varying features, such as the current wealth level and the remaining time, it can be framed as a problem of sequential decision-making under uncertainty. Consequently, it can be naturally tackled by deep reinforcement learning (DRL) techniques.

To ensure high performance under real-world conditions, GBI frameworks should also account for another dynamic: the non-stationarity of financial markets [3], including abrupt regime switches. Notably, in classical portfolio optimization, regime-based asset allocation has been shown to enhance portfolio risk and return, especially by mitigating potential drawdowns through swift reactions to market changes [4]. We hypothesize that such advantages could also be transferred to the GBI setting. Motivated by this, in this paper, we extend the previous work on DRL for GBI by introducing regime-switching considerations. Our main contribution is a novel DRL framework for GBI under regime-switching that directly incorporates regime probability estimates into the state space.

2 Literature Review

2.1 Goal-Based Investing

A range of different approaches to GBI [5–7] exist, all of which rely on some type of time-dependent risk mitigation.

A deterministic glide path is a simple and static GBI strategy given by: $\alpha_t = 1 - \frac{t}{T}$, where α_t represents the stock portfolio weight at time t, and T the target time. As such, it depends only on the fraction of the remaining time and does not take into account dynamic market conditions or even more sophisticated investment goals [8]. Nevertheless, it has secured popularity among investors due to its simplicity and intuitive nature, being commonly used in retirement asset allocation.

Merton introduced a seminal work on lifelong portfolio selection under uncertainty [5]. Several assumptions were first made: a) the return rate r of the riskless asset is set to be constant; b) the price of the risky asset, S_t, follows a log-normal process with the expected rate of return μ and volatility σ; and c) the investor maximizes the Constant Relative Risk Aversion (CRRA) utility $\mathcal{U}(x) = x^\gamma/\gamma,\ \gamma < 1$, where $1-\gamma$ is the coefficient of relative risk aversion. Under these settings, it is shown that maintaining constant portfolio weights for each asset is optimal, with the risky asset weight equal to $\alpha_t = \frac{\mu - r}{(1-\gamma)\sigma^2}$.

Bruder *et al.* [9] suggest limiting the cumulative variance of the portfolio instead of using a utility function to describe one's risk aversion. The variance budget V^2 represents the maximum amount of risk that the investor is willing to take on during the investment period: $\int_0^T \alpha_t^2 \sigma^2 X_t^2 dt \leq V^2$. The

*Corresponding Author.

Proceedings of the 5th Northern Lights Deep Learning Conference (NLDL), PMLR 233, 2024.

optimal risky asset weight is given by: $\alpha_t = \frac{V}{\sigma\sqrt{T}X_t}$.

Das *et al.* [10] propose a dynamic programming approach that generates a wealth-dependent trading strategy. In each timestep, the strategy performs portfolio rebalancing by selecting a single portfolio from the set of 15 predefined portfolios lying on the efficient frontier, with the goal of maximizing the probability of reaching the investment goal. The proposed approach is flexible, as it can handle cash infusions or withdrawals of arbitrary size, arbitrary time periods, and multiple investment goals, while also adjusting risk by regulating the selection of predefined portfolios.

2.2 RL for Goal-Based Investing

Dixon and Halperin [11] introduce an algorithm for GBI based on G-learning, a probabilistic extension of Q-learning. It offers several advantages, including its ability to tackle noisy data, lack of assumptions about the data-generating process, and the use of one-step rewards. Furthermore, they propose GIRL, an augmentation of the algorithm to be used with inverse RL. Das and Varma [7] approach GBI with Q-learning, achieving the same results as those obtained through dynamic programming. Moreover, the authors provide a taxonomy of RL approaches viable for this problem and emphasize its scalability to large state and action spaces. Bauman *et al.* [12] focus on robust GBI and propose a solution based on DRL, which is demonstrated to outperform several benchmarks. Zhang *et al.* [13] use modified hybrid proximal policy optimization to bypass the need for discretization when dealing with the GBI problem, with results also showing superiority over traditional methods.

2.3 HMMs for Regimes

The non-stationarity of financial markets can be captured using of a Hidden Markov Model (HMM), a probabilistic model comprising two distinct processes: a Markov chain with hidden states, typically denoted by $X = (X_t : t \geq 0)$, and an observable process $Y = (Y_t : t \geq 0)$ whose outcomes are directly affected by the outcomes of X. Since X is unobservable, the goal is to learn about it indirectly through Y. The process X satisfies the Markov assumption $\mathcal{P}(X_t \,|\, X_{t-1}) = \mathcal{P}(X_t \,|\, X_{t-1}, \ldots, X_0)$, i.e., given the current state, the future state is not conditional on the prior (past) states. Furthermore, the output independence property holds, meaning that $\mathcal{P}(Y_t \,|\, X_t) = \mathcal{P}(Y_t \,|\, X_t, \ldots, X_0)$. The hidden state X_t is said to "emit" the observable Y_t.

HMMs have diverse applications in regime-based asset allocation and beyond. Kritzman *et al.* [14] show that, in the context of regime forecasting, HMMs dominate over simple data partitions based on thresholds. Furthermore, the authors successfully leverage HMM-based modeling of regime shifts to improve asset allocation. Kim *et al.* [15] utilize an HMM for identifying the phases of individual assets and suggest an investment approach that exploits price trends effectively. Wang *et al.* [16] employ an HMM to detect different market regimes and propose an investment strategy that adjusts factor investing based on the currently identified regime.

3 Data Generation

Our study employs a common two-asset model for goal-based investing, aiming to balance capital preservation and growth. While this model limits portfolio diversity, it enables direct comparisons with other approaches. We use monthly bond and stock returns provided by R. Shiller [1]. The dataset is split into a training set (spanning 1870 to 1991) and a testing set (from 1992 to 2022). The former is used for training the HMM with Gaussian mixture observations which is then employed to generate numerous trajectories. Creating more trajectories is imperative because deep reinforcement learning requires a substantial amount of data, and relying solely on historical data is insufficient. The test set is reserved solely for testing purposes. Since the data extends back to 1870, we use inflation-adjusted returns, i.e., Consumer Price Index (CPI)-adjusted returns. The CPI monitors the price change of consumer goods and services purchased by households, making it a popular measure of inflation and deflation.

3.1 Gaussian HMM

An HMM can be denoted by a quintuplet $\langle H, O, T, \Psi, \Pi \rangle$. Elements H, T, and Π describe the behavior of the underlying Markov chain $X = (X_t : t \geq 0)$, while O and Ψ specify the observable process $Y = (Y_t : t \geq 0)$.

More specifically, H represents the set of hidden Markov chain states, while $T = (t_{ij})_{i,j \in H}$ denotes the transition matrix describing its transition probabilities, i.e., $t_{ij} = \mathcal{P}(X_{t+1} = j \,|\, X_t = i)$. Furthermore, $\Pi = (\pi_i)_{i \in H}$ gives the initial probabilities $\pi_i = \mathcal{P}(X_0 = i)$. Similarly, O represents the set of values of the observable process Y and $\Psi = (\psi_{ik})_{i \in H,\, k \in O}$ is the emission probabilities matrix. Precisely, $\psi_{ik} = \mathcal{P}(Y_t = k \,|\, X_t = i)$, giving the probability that the hidden state $i \in H$ will emit the observation $k \in O$.

In what follows, we assume the hidden Markov chain X to have two states, $H = \{0, 1\}$, corresponding to two market regimes. For the observable process, we set $Y = (B_t, S_t)_{t \geq 0}$, where B_t (S_t) denotes bond (stock) return at time t. The underlying bivariate data, consisting of bond and stock returns,

[1] http://www.econ.yale.edu/~shiller/data.htm

is assumed to come from a Gaussian mixture. Put differently, the returns are presumed to follow a Gaussian distribution conditional on the market regime, i.e., $Y_{|X=x} \sim \mathcal{N}(\mu_x, \Sigma_x)$, for $x \in \{0, 1\}$. The estimation of the HMM parameters T, Ψ, and Π from the historical data is performed using the Baum-Welch algorithm [17], a special instance of the Expectation-Maximization (EM) algorithm. After 10,000 iterations of the algorithm, the following emission parameters are obtained:

$$\boldsymbol{\mu}_0 = \begin{bmatrix} 0.0087 & 0.0015 \end{bmatrix}, \ \boldsymbol{\Sigma}_0 = \begin{bmatrix} 0.0009 & 0.0001 \\ 0.0001 & 0.0001 \end{bmatrix}$$

$$\boldsymbol{\mu}_1 = \begin{bmatrix} -0.0067 & 0.0057 \end{bmatrix}, \ \boldsymbol{\Sigma}_1 = \begin{bmatrix} 0.0060 & 0.0004 \\ 0.0004 & 0.0008 \end{bmatrix}.$$

Based on the attained parameters for the Gaussian distributions of the observation process, we draw the following conclusions:

- When the hidden process is in the first regime ($X = 0$), both the mean values of stock and bond returns are positive, with the former being larger. Also, the stock volatility is higher than the bond volatility.

- When the hidden process is in the second regime ($X = 1$), the mean value of stock returns is negative, whereas the mean value of bond returns is positive and greater than in the first regime. Furthermore, the volatilities of both the stock and bond returns are higher than in the first regime, along with their covariance.

Fig. 1 shows the stock and bond returns from the train set along with the estimated corresponding regimes. It is clear that the low volatility periods are appointed to the first regime, whereas the second regime includes periods of high volatility and financial recessions (e.g., the Great Depression and the early 1980s recession).

The process of generating data to create trajectories for training the DRL agent was carried out in two steps: first, by generating a sequence of hidden states, and then by simulating observable values from the corresponding distributions. This newly generated data is utilized as the input for deep reinforcement learning.

4 Deep Reinforcement Learning Setting

The underlying problem is represented with a discrete-time Markov decision process (MDP) and treated as an episodic deep reinforcement learning (DRL) task.

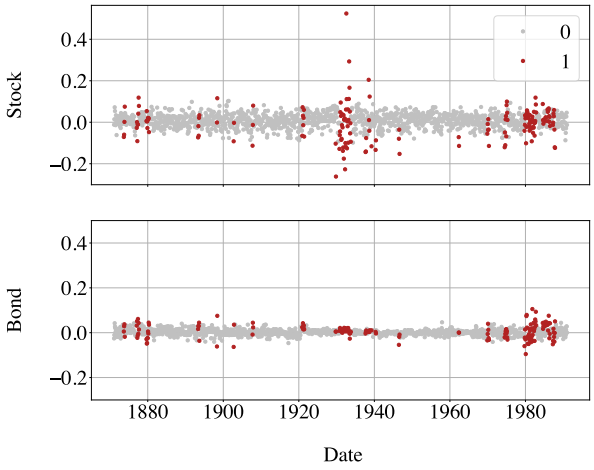

Figure 1. Stock and bond returns with corresponding regimes, i.e. hidden Markov states, from 1871 to 1991.

4.1 Markov Decision Process

An MDP is defined as a quintuplet $\langle S, A, P, R, \gamma \rangle$ consisting of a state space (S), action space (A), a transition probability function ($P : S \times A \times S \to [0, 1]$) dictating state transitions given actions, a reward function ($R : S \times A \to \mathbb{R}$) assigning values to actions, and a discount factor ($\gamma \in [0, 1]$) determining the preference for short-term rewards.

The state at time t is defined as:

$$s_t = \left(\frac{t}{T}, \frac{W_t}{W_G}, \hat{h}_{t-1} \right).$$

Here, T signifies the target date (also known as the investment horizon), W_t represents the current total wealth, and W_G is the desired wealth goal. Note that $W_t = w_{b,t} P_t^b + w_{s,t} P_t^s$, where P_t^b and P_t^s represent the prices of the riskless and risky assets at time t, respectively. The weights $w_{b,t}$ and $w_{s,t}$ denote the quantities of these assets held by the investor at t. We assume $w_b, w_s \geq 0$, i.e., short positions are not permitted. The third feature of the state space $\hat{h}_{t-1}$ accounts for the hidden regime - it represents the estimate of the hidden Markov state from the previous time step, and is given by:

$$\hat{h}_t = \mathcal{P}(X_t = 0 \,|\, Y_t = (B_t, S_t)).$$

The regime estimation involves determining the probability that the Markov model is in state 0 at time t, given the observation Y_t. (Considering there are only two regimes, $\mathcal{P}(X_t = 1) = 1 - \mathcal{P}(X_t = 0)$). This approach to regime detection takes into account not only the more probable regime but also the uncertainty of the actual market state.

The action at time t is defined as $a_t = w_{s,t}$, with $w_{s,t} \in [0, 1]$, representing the portion of funds allocated to the risky asset at time t. As $w_s + w_b = 1$, the allocation to the risk-free asset is uniquely defined.

Binary reward functions are a natural choice in the context of GBI, given that goals are either attained

(1) or not (0). Therefore, a positive reward is granted (at the end of the episode) only if the goal is attained ($r_T = \mathbb{1}_{W_T \geq W_G}$).

4.2 Training Procedure

The interaction between the DRL agent and the environment is given as follows. Each episode, generated by the HMM as described in 3.1, represents a span of 10 trading years, comprising 120 time steps aligning with trading months. The investor's goal is to achieve a 65% return on the initial investment, i.e. the investor begins with 100 and has to achieve the goal of 165. The previous month's returns S_{t-1} and B_{t-1} are then used to estimate the hidden state h_{t-1}. Based on the entire state s_t, the agent then decides on the fraction to invest in the risky (risk-free) asset and allocates the wealth accordingly. When the target date is reached, the episode ends and the agent receives the reward.

4.3 Algorithm and neural network architecture

The Proximal Policy Optimization (PPO) algorithm is used [18] for learning RL policies. The key idea behind PPO is to optimize a surrogate objective function, which is clipped to avoid large policy updates and thereby stay close to the old policy:

$$J(\boldsymbol{\theta}) = \mathbb{E}\left[\min\left(r_{\boldsymbol{\theta}} A(s,a), \text{clip}_{\epsilon}(r_{\boldsymbol{\theta}}) A(s,a)\right)\right],$$

with $\text{clip}_{\epsilon}(x) = \text{clip}(x, 1-\epsilon, 1+\epsilon)$ being a clip function with ϵ as a hyperparameter. $\mathbb{E}$ stands for the empirical expectation, $\boldsymbol{\theta}$ represents the policy parameters, and r_θ is the probability ratio under the new and previous policies. Finally, $A(s,a)$ signifies the advantage function. A feed-forward fully connected neural network comprised of 2 hidden layers, 6 neurons each, is used, employing the ReLU activation function. The value of the discount factor γ is set to 1, and the small learning rate of 0.0001 takes into account the stochastic nature of the environment. The implementation is based on the use of the Stable Baselines3 [19] and PyTorch frameworks.

5 Results

5.1 Benchmarks

To assess the DRL agent's performance, the entire set of benchmarks discussed in more detail in 2.1 were employed, encompassing both analytical and numerical methods. The risk parameters, which include Merton's risk aversion parameter, and the variance budgeting approach, were selected based on their performance on the train set, specifically the period from 1871 to 1991. The parameters that yielded the highest success rate in terms of achieving goals across all training episodes were chosen and used on the test set.

None of the benchmarks originally use regime switches. However, for a more rigorous evaluation of the DRL agent's performance, the estimates of the hidden state used in the model were also used to calibrate the benchmarks. This was done through the distribution parameters $(\boldsymbol{\mu}, \boldsymbol{\Sigma})$, whose calculation is required for all benchmarks except the deterministic glide path. More specifically, if the more probable estimated hidden regime at time t is 0, i.e. $\arg\max_x \mathcal{P}(X_t = x \,|\, Y_t = (B_t, S_t)) = 0$, the parameters used for the benchmarks are set to $(\boldsymbol{\mu}_0, \boldsymbol{\Sigma}_0)$ (outlined in 3.1), and vice versa when $\arg\max_x \mathcal{P}(X_t = x \,|\, Y_t = (B_t, S_t)) = 1$.

Table 1. Comparison of results

	Original	With regimes
DG	46.9%	46.9%
VB	58.2%	76.5%
MC	65.2%	78.9%
DP	68.4%	79.3%
DRL	88.7%	

5.2 Simulation results

Table 1 shows the percentage of episodes in which the goal (of 165 with an initial investment of 100) was achieved, comparing the benchmarks with DRL. The abbreviations are used as follows: DG – Deterministic glide path, MC – Merton's constant, VB – Variance budgeting, DP – Dynamic programming. The performance of the benchmarks was assessed under two scenarios: one with the inclusion of regime information and one without, highlighting the effect of regime detection on the overall results.

All of the benchmarks exhibit improved performance when using the estimated regime. These findings make it evident that regime detection plays a pivotal role in investing. When comparing the benchmark methods with the DRL agent, it is clear that DRL achieves a much higher success rate. In their totality, the findings underscore the significance of accounting for uncertainties when dealing with highly stochastic financial markets, which is naturally provided within the DRL framework.

To explore the performance of the DRL agent in more detail, we tested it on different levels of goal. Fig. 2 shows the achievement of the agent along with the benchmarks on multiple goal levels, presuming that the initial investment equals 100. The superior performance of the DRL agent compared to benchmark methods is evident, regardless of the target wealth.

Moreover, Fig. 3 presents a more in-depth analysis of the model's performance, extending beyond

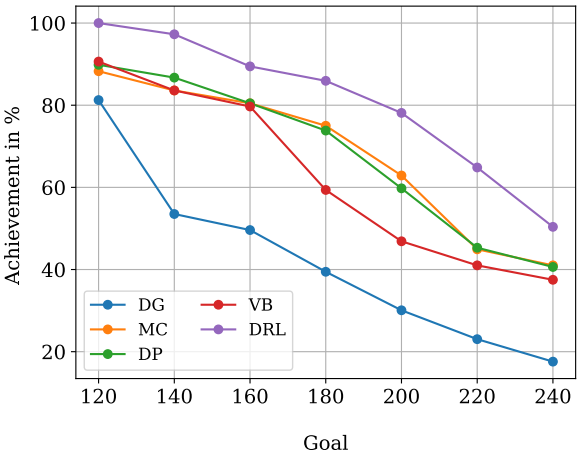

Figure 2. Performance of the DRL agent and benchmarks on the test set across various goal levels, assuming an initial investment of 100.

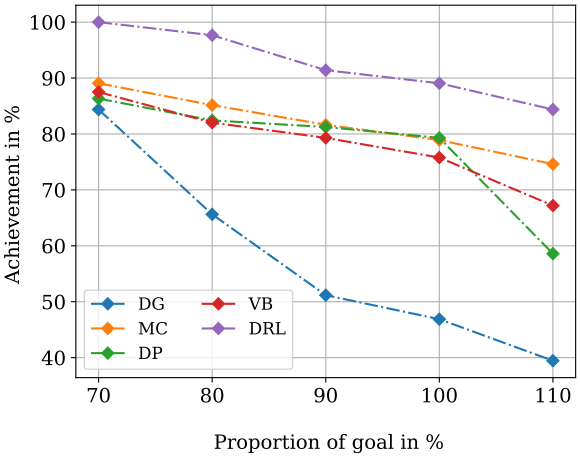

Figure 3. Performance of the DRL agent and benchmarks on the test set across various proportions of the primary goal.

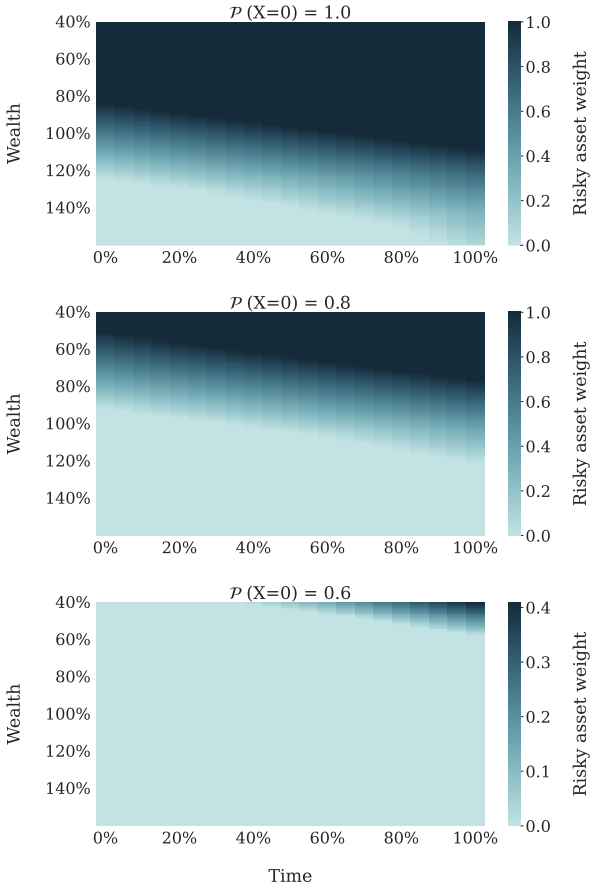

Figure 4. Policy learned by the DRL agent. Each subfigure shows the agent's actions across the state space assuming a fixed regime probability.

the exclusive consideration of achieving the target wealth. This is essential as the achievement metric alone does not capture potential catastrophic failures when the goal is not met. With the primary goal set to 165 and an initial investment of 100, the figure illustrates the methods' success in attaining various proportions of the goal. For instance, when trying to achieve a goal of 165, the DRL agent consistently achieves a minimum of 70% of the target, equivalent to 115.5 ($165 \cdot 70\% = 115.5$). This underscores that even when the primary goal is not met, the DRL agent still comes close.

The seemingly subtle addition of using the hidden state probability estimate results in a significant performance increase. To investigate this effect in greater detail, Fig. 4 is shown. It presents the dependence of the policy learned by the DRL agent on the regime probability estimate, i.e., the state feature $\hat{h}_t$. If the agent is certain that the market is in the first regime ($\mathcal{P}(X = 0) = 1$), it behaves as expected in GBI. At the beginning of the investment period, it takes on less risk due to the large amount of time left for accumulating the desired wealth. As the target date approaches, the agent becomes more inclined to accept more risk in order to reach the goal. As indicated by the parameter values ($\boldsymbol{\mu}_0$,$\boldsymbol{\Sigma}_0$), the first regime corresponds to a setting featuring one asset with high-reward and high-risk, and another with low-reward low-risk characteristics. The ensuing DRL policy therefore clearly follows a typical target-date investment paradigm of balancing between these two components. As the probability of being in the first state ($\mathcal{P}(X = 0)$) decreases, the boundary line between the risky and risk-free allocation regions moves until fully disappearing for $\mathcal{P}(X = 0) < 0.6$. Within this range of values, the agent only benefits from investing in the risk-free asset. This stems from the values of the estimated parameters for the second regime ($\boldsymbol{\mu}_1$,$\boldsymbol{\Sigma}_1$). Given that one asset has a negative mean return, the preferable action is to invest in the alternative asset, irrespective of the goal proximity.

6 Conclusion and Future Work

In this paper, we present a novel method for GBI under regime-switching based on DRL which involves enriching the state space with regime probability estimates. The experimental results point to the superiority of the method over several standard GBI benchmarks. The proposed approach could be expanded by exploring multiple regimes corresponding to a variety of different market conditions. Similarly, more regime-detecting models can be explored as well. Furthermore, the state space can be enriched with macroeconomic factors e.g. economic growth and inflation. These variables might prove beneficial by capturing even more of the market complexity. Future work could also involve considering multi-asset GBI frameworks, employing portfolios that are highly diversified across multiple asset classes, all with the aim of achieving both lower total risk and a higher rate of attaining financial objectives.

Acknowledgments

This work was supported in part by the Croatian Science Foundation under Project 5241, and in part by the European Regional Development Fund under Grant KK.01.1.1.01.0009 (DATACROSS).

References

[1] S. R. Das, D. N. Ostrov, A. Radhakrishnan, and D. Srivastav. "A new approach to goals-based wealth management". In: *Available at SSRN 3117765* (2018). DOI: 10.2139/ssrn.3117765.

[2] H. M. Markowitz and H. M. Markowitz. *Portfolio selection: efficient diversification of investments*. J. Wiley, 1967.

[3] A. Ang and A. Timmermann. "Regime changes and financial markets". In: *Annu. Rev. Financ. Econ.* 4.1 (2012), pp. 313–337.

[4] P. Nystrup, B. W. Hansen, H. O. Larsen, H. Madsen, and E. Lindström. "Dynamic allocation or diversification: A regime-based approach to multiple assets". In: *The Journal of Portfolio Management* 44.2 (2017), pp. 62–73.

[5] R. C. Merton. "Lifetime portfolio selection under uncertainty: The continuous-time case". In: *The review of Economics and Statistics* (1969), pp. 247–257.

[6] P. A. Forsyth and K. R. Vetzal. "Robust asset allocation for long-term target-based investing". In: *International Journal of Theoretical and Applied Finance* 20.03 (2017), p. 1750017.

[7] S. R. Das and S. Varma. "Dynamic goals-based wealth management using reinforcement learning". In: *Journal Of Investment Management* 18.2 (2020), pp. 1–20.

[8] P. A. Forsyth, Y. Li, and K. R. Vetzal. "Are target date funds dinosaurs? Failure to adapt can lead to extinction". In: (2017). eprint: 1705.00543v1.

[9] B. Bruder, L. Culerier, and T. Roncalli. "How to design target-date funds?" In: *Available at SSRN 2289099* (2012).

[10] S. R. Das, D. Ostrov, A. Radhakrishnan, and D. Srivastav. "Dynamic portfolio allocation in goals-based wealth management". In: *Computational Management Science* 17 (2020), pp. 613–640.

[11] M. Dixon and I. Halperin. "G-learner and girl: Goal based wealth management with reinforcement learning". In: *arXiv preprint arXiv:2002.10990* (2020).

[12] T. Bauman, B. Gašperov, S. Begušić, and Z. Kostanjčar. "Deep Reinforcement Learning for Robust Goal-Based Wealth Management". In: *IFIP International Conference on Artificial Intelligence Applications and Innovations*. Springer. 2023, pp. 69–80. DOI: 10.1007/978-3-031-34111-3_7.

[13] J. Zhang, C. Wan, M. Chen, and H. Liu. "An Efficient Reinforcement Learning Approach for Goal-Based Wealth Management". In: *Available at SSRN 4403929* ().

[14] M. Kritzman, S. Page, and D. Turkington. "Regime shifts: Implications for dynamic strategies". In: *Financial Analysts Journal* 68.3 (2012), pp. 22–39. ISSN: 0015198X. DOI: 10.2469/faj.v68.n3.3.

[15] E.-C. Kim, H.-W. Jeong, and N.-Y. Lee. "Global Asset Allocation Strategy Using a Hidden Markov Model". In: *Journal of Risk and Financial Management* 12.4 (2019), p. 168. DOI: 10.3390/jrfm12040168.

[16] M. Wang, Y.-H. Lin, and I. Mikhelson. "Regime-Switching Factor Investing with Hidden Markov Models". In: *Journal of Risk and Financial Management* 13.12 (2020), p. 311. ISSN: 19118074. DOI: 10.3390/jrfm13120311.

[17] J. A. Bilmes. *A Gentle Tutorial of the EM Algorithm and its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models*. "International Computer Science Insitute, UC Berkeley". "[Online http://www.leap.ee.iisc.ac.in/sriram/teaching/MLSP_18/refs/GMM_Bilmes.pdf; accessed22-August-2023]". 1998.

[18] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. K. Openai. *Proximal Policy Optimization Algorithms*. Tech. rep. arXiv: 1707.06347v2.

[19] A. Raffin, A. Hill, A. Gleave, A. Kanervisto, M. Ernestus, and N. Dormann. *Stable-Baselines3: Reliable Reinforcement Learning Implementations*. Journal of Machine Learning Research. [**22**(1), 12348–12355]. 2021.