Optimizing Stock Portfolios Through Mean-Variance Disentanglement Using Deep Reinforcement Learning

Ahmad Asadi, Reza Safabakhsh*

Deep Learning Lab, Computer Engineering Department

Amirkabir University of Technology, Tehran, Iran.

Abstract

The rise of high-frequency trading, particularly in volatile markets like cryptocurrencies, demands sophisticated models for optimizing short-term stock portfolios. This study presents a cutting-edge approach to portfolio optimization by leveraging mean-variance disentanglement with a Variational Autoencoder to extract features that predict the persistence of price trends. These features are then fed into an Actor-Critic model, which dynamically selects the most promising stocks based on their trend persistence probabilities. Our model consists of a feature extractor that isolates latent factors influencing price movements and an Actor-Critic component that optimizes stock selection. Experimental results highlight the effectiveness of our method, achieving a 125% total ROI over one year, significantly outperforming the best baseline strategy, which attained a 69% ROI. This innovative method offers a robust framework for enhancing returns and managing risks in high-frequency trading environments.

*Corresponding author

Email addresses: `ahmad.asadi@aut.ac.ir` (Ahmad Asadi), `safa@aut.ac.ir` (Reza Safabakhsh)

Keywords: deep learning portfolio optimization, disentangled representation learning, mean-variance disentanglement, variational auto-encoders, actor-critic models

1. Introduction

Stock portfolio management is a crucial aspect of investment strategy, involving the selection and allocation of assets to achieve specific financial goals. The importance of portfolio management lies in its ability to optimize risk and return trade-offs, ultimately leading to the maximization of wealth for investors. Investors typically mitigate risk and enhance returns by carefully selecting a diverse range of assets.

Portfolio diversification is a critical strategy in risk management within investment portfolios. This approach involves allocating investments across various asset classes and industries to mitigate the impact of individual asset price fluctuations on the overall portfolio performance. By diversifying, investors can reduce volatility and safeguard against substantial losses in any single investment, thus enhancing the stability and robustness of the portfolio's returns. Additionally, a well-diversified portfolio has the potential to enhance risk-adjusted returns by leveraging the advantages of diverse market conditions (Markovitz [1]).

High frequency trading techniques have emerged as a prominent strategy in financial markets, leveraging advanced algorithms and rapid execution speeds to capitalize on fleeting market opportunities (Li and Hoi [3]). The inherent volatility and speed of high frequency trading present unique challenges, particularly in avoiding small losses that can quickly accumulate.

In the realm of high frequency trading, to avoiding small losses often strategies rely on the aggregation of information from diverse sources (Thakkar and Chaudhari [4]). By incorporating a wide range of technical indicators (Chen et al. [5]), crowd-sourced data (Liu et al. [6]), quantitative data (Gomber and Haferkorn [7]), and multi-type data fusion models (Liu et al. [6]) high frequency traders can enhance their ability to identify and react to market trends with greater precision and speed. This multi-faceted approach not only enables traders to more effectively manage risk by mitigating small losses but also provides a comprehensive view of market dynamics that can inform more strategic trading decisions. By leveraging a sophisticated blend of quantitative analysis and real-time information sources, high frequency trading techniques can strive to optimize performance and navigate the complex interplay between risk and return in dynamic financial markets.

The majority of the researchers active in portfolio management believe that employing complex models for fusing the information from different sources enables models to find pivot points in stock price series([4]). However, we believe that for finding pivot points in stock price series before occurrence it is required to extract features of the price behavior by disentangling the impact of interconnected factors from the last time steps and measuring the uncertainty of the persistence of the current trend in price series of each stock. Due to the unpredictable nature of financial markets and the dynamic interactions between various factors, uncertainty is inherent in stock portfolio management. Then, measuring and managing uncertainty is essential for making investment decisions and designing robust portfolio strategies.

In this paper, a two-step framework is presented which is able to first extract the features indicating the probability of price trend change in each stock in the future, and second, to adjust the portfolio based on the extracted features in order to minimize the loss probability at each transaction. The contributions of this work can be concluded as follows:

1. A feature extraction module is presented that is able to extract features indicating the probability of the price trand change in the future for each stock.
2. A deep reinforcement learning model with a novel reward function is proposed which is able to switch between the stocks in the market based on the uncertainty of their current price trend in the future.
3. The proposed model outperforms the state-of-the-art models in stock portfolio management in crypto-currencies market.

The rest of this paper is organized as follows: In Section 2 we review the existing models for stock portfolio management and stock price trend predictors. Section 3 describes in detail the structures of the time-series uncertainty estimation model and the deep reinforcement learning model. The proposed model is then evaluated and the evaluation results are reported and discussed in Section 4. Finally, the conclusions of the paper are summarized in Section 5.

2. Related Work

2.1. Information Fusion in Stock Portfolio Management

The use of deep learning models in stock portfolio management is very well justified due to their ability to process vast amounts of data and extract complex patterns that traditional models may overlook. Traditional portfolio management strategies often rely on statistical models (Li et al. [9]) and optimization models (Pennanen [10]) to make investment decisions. However, with the advent of deep learning and deep reinforcement learning techniques, researchers and practitioners have started exploring more sophisticated and data-driven approaches mostly based on using technical indicators to portfolio management (Ayala et al. [11], Agrawal et al. [12], Taghian et al. [13], and Taghian et al. [14]).

Due to the capabilities of deep learning models in financial markets specially their abilities to process large amounts of complex data, the fusion of various types of information plays a crucial role in enhancing the performance of deep learning-based portfolio management systems. One common approach to information fusion in deep learning-based stock portfolio management is the use of multi-modal data inputs (Asadi and Safabakhsh [8]). This involves combining different types of data, such as historical price data, financial statements, news sentiment, and social media feeds, into a single model. By incorporating multiple modalities of information, these models can capture a more comprehensive view of market dynamics and make more informed investment decisions.

Another approach to information fusion is the use of ensemble methods, where multiple deep learning models are trained on different subsets of data or with different architectures (Carta et al. [15]). The outputs of these models are then combined to generate a consensus prediction, which can be more robust and accurate than any individual model. Lin et al. [16] proposed a deep reinforcement learning (DRL) based model with multiple agents and designed a long-term reward function to reduce the risk of investment with fusing the decisions of different agents. Hao et al. [17] proposed a three-dimensional fuzzy representation of stock price trend and employed an ensemble DRL model for stock portfolio management based on the fuzzy representation of the price trend.

Furthermore, attention mechanisms have been proposed as a way to selectively focus on relevant information within a dataset, specifically in gathering information from multiple markets (Zhao et al. [18]). Another important deep learning model which is used for fusing temporal information for stock portfolio management is the Transformers model (Gullotto [19]). Kisiel and Gorse [20] proposed a deep learning model based on a transformer for minimizing the small losses of stock trading with optimizing the Sharpe ratio Sharpe [21] directly. Liu et al. [22] combined a non-stationary transformer model with a DRL model for fusing macro-economic information with targeted news sentiments.

2.2. Uncertainty Measurements in Stock Trading Strategies

Uncertainty is a fundamental aspect of financial markets, and accurately measuring and managing uncertainty is crucial for developing effective stock trading strategies. In recent years, researchers have focused on exploring various methods and metrics to quantify uncertainty in stock trading, with the aim of improving decision-making processes and reducing risk exposure. This literature review provides an overview of the key studies and approaches related to uncertainty measurements in stock trading strategies (Abdar et al. [23]).

The first category of models proposed for portfolio management with considering the risk of investment, are those that try to minimize investment risk by diversification of the proposed portfolios. Du [24] proposed a mean-variance portfolio proposition based on the co-integrated stocks and their correlation.

Considering risk-related measures for optimizing deep learning model weights is another approach in which the trained model would consider the uncertainty of the stock prices in portfolio proposition. Syu et al. [25] proposed a DRL based model in which the Sharpe ratio of portfolio proposals is used to optimize the model weights.

A novel model recently introduced by Abdulsahib and Ghaderi [26] in this field utilizes disentangled representation learning to break down the input features from various markets and identify the common factors during the representation learning phase. By decomposing the features of distinct markets and identifying shared features between them, the model can better understand the interconnected dynamics between the markets and predict price movements in one market based on the behavior of shared features with another market. Furthermore, the concept of disentangled representation learning was previously employed by Abdulsahib et al. [27] to study the price dynamics of stocks in financial markets.

While most of related work has focused on integrating information from various sources in stock portfolio management, this study seeks to break down the fundamental factors that influence price behavior. The goal is to detect indications of trend changes and identify price pivot points before they occur. To achieve this, disentangled representation learning is utilized to create a latent space based on historical stock price data, where latent features indicate potential price changes. A feature extractor is developed to detect price pivot points, and an actor-critic model is proposed to pinpoint the optimal time to switch between stocks to mitigate losses.

3. Proposed Method

The model proposed in this paper consists of two modules. The first module is a feature extractor which is enabled to extract features indicating the strength of the current price trend in the near future. The second module is an actor-critic based model which selects a single coin to buy and hold until the next investment interval based on the uncertainty about the price trends of each coin extracted by the first module.

Figure 1 illustrates the architecture of the model presented in this study. The model is built on the concept of disentangled representation learning, where a feature extractor is trained to extract features from the historical price data of each coin. These extracted features are then combined and fed into an actor-critic model to select a single coin from the available options in the market. It is important to note that the portfolio proposals in this research consist of only one coin, and no combinations of coins are considered. This constraint is imposed to enhance the clarity of the model's investigation and facilitate comparisons with the existing models in the field.

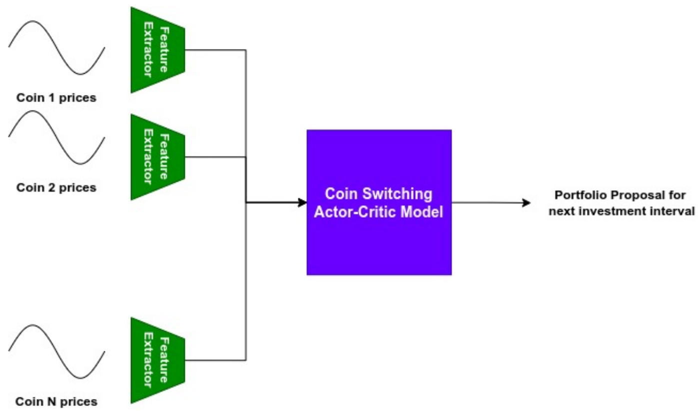

Figure 1: The structure of Model Switching idea for stock trading

3.1. Feature Extraction

Disentangled representation learning is a powerful technique in machine learning that aims to learn representations of data where different factors of variation are separated into distinct and interpretable components. This can be particularly useful in time series data, where multiple underlying processes may be present and need to be disentangled for better understanding and analysis.

Let's consider a time series dataset $X = \{x^1, x^2, ..., x^t\}$, where x^i represents the observation at time step i. The goal of disentangled representation learning on time series is to learn a set of latent variables $Z = \{z_1, z_2, ..., z_K\}$ that capture the underlying factors of variation in the data. These latent variables should be disentangled, meaning that each z_k represents a different aspect of the data that is independent of the others.

Given the assumption of independence among the generating factors, the task of disentangled representation learning in a dimension-wise manner aims to encode the information pertaining to these generating factors by means of a latent vector of lower dimensionality - as a compact representation of the high-dimensional observations. The relationship between the observations and the latent vectors can be formally characterized by their joint distribution

$$p_\theta(x, z) = p_\theta(x|z)p_\theta(z) \tag{1}$$

$$p_\theta(z) = N(z|0, \sigma^2 I) \tag{2}$$

where θ denotes the set of parameters of the feature extractor model and the prior distribution $p(z)$ is typically assumed to be a multidimensional Gaussian distribution with the same variance in all directions

One common approach to achieve disentangled representation learning on time series data is through Variational Autoencoders (VAEs) (Duan et al. [28], Li et al. [29], and Li et al. [30]). VAEs are generative models that learn a probabilistic mapping between the observed data and the latent variables. By introducing a prior distribution over the latent variables, VAEs can learn disentangled representations by encouraging the latent variables to capture independent factors of variation in the data.

In the first module of the proposed method, we utilize a VAE structure to capture the underlying distribution of the data and extract disentangled representations for uncertainty measurement in stock portfolio management. The VAE is designed to learn a latent space that separates different factors of variation in the data, enabling us to better understand and quantify the uncertainty associated with our model's predictions.

The VAE consists of an encoder network that maps input data (e.g., historical stock prices, market indicators) to a latent space representation, and a decoder network that reconstructs the input data from the latent space. By training the VAE to minimize the reconstruction error and maximize the mutual information between the input and latent representations, we aim to learn a compact and meaningful representation of the data that facilitates uncertainty estimation.

To encourage disentangled representation learning in the VAE, we incorporate regularization techniques, such as β-VAE (Burgess et al. [31]) or disentanglement loss functions, that promote the separation of different factors of variation (e.g., market trends, individual stock performance) in the latent space. This disentangled representation enables us to measure uncertainty more effectively and make informed decisions in stock portfolio management.

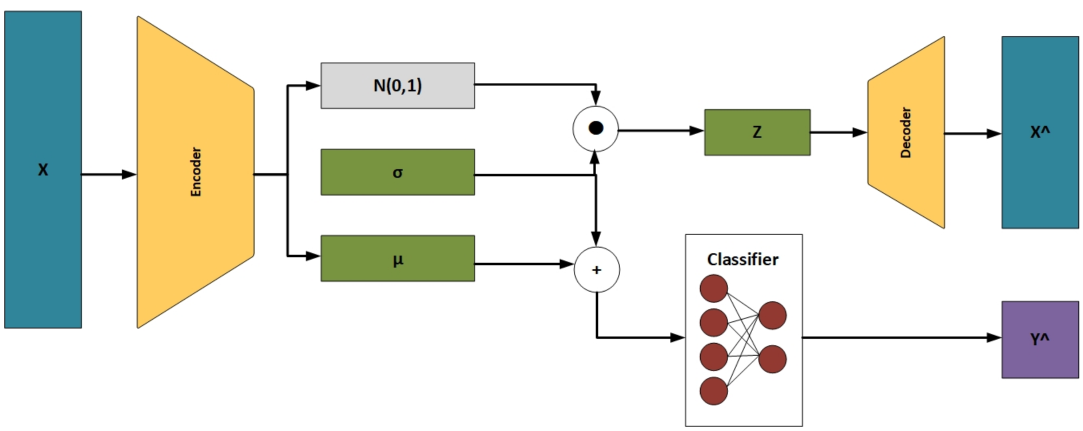

Figure 2: The structure of the proposed VAE model for uncertainty estimation.

The relation between the latent features and input data in the proposed VAE model is described in equation (3).

$$z_i = \hat{\mu}_i + \hat{\sigma}_i \epsilon \tag{3}$$

where z_i denotes the ith element of the latent vector, $\hat{\mu_i}$ and $\hat{\sigma_i}$ respectively represent the latent vectors learned by the encoder to simulate the mean and variance of the price series, and $\epsilon \propto N(0, 1)$ denotes a random white noise.

Since the latent vectors $\hat{\mu_i}$ and $\hat{\sigma_i}$ are the vectors that are passed to the portfolio manager module, they are assumed to encode information regarding price trend. In addition, these vectors should be informative enough to estimate our uncertainty about the persistence of the current price trend in the near future. Therefore, a classifier is embedded inside the VAE model and an extra term is added to VAE model's loss function to ensure that $\hat{\mu_i}$ and $\hat{\sigma_i}$ encode necessary information.

A binary classifier $\Phi(\hat{\mu}, \hat{\sigma})$ is augmented into the VAE model in order to enforce latent variables to encode information related to future price trend. The ground-truth labels for this classifier are generated via equation (4).

$$Y_i^t = \begin{cases} 1 & \text{if } \frac{x_i^{t+l}}{x_i^t} > 1 \\ \text{-1} & \text{otherwise} \end{cases} \tag{4}$$

where $\frac{x_i^{t+l}}{x_i^t}$ computes the total return of stock i in l steps ahead from time step t and Y_i^t denotes the label of the classifier for stock i in time step t. Based on equation (4), the proposed loss function for learning the set of parameters of the VAE model is presented in equation (5).

$$\mathcal{L} = \mathcal{E} + \mathcal{C} - \mathcal{K} \tag{5}$$

$$\mathcal{E} = \Sigma_{i=1}^{S} \Sigma_{t=w}^{T} (\hat{X_i^t} - X_i^T)^2 \tag{6}$$

$$\mathcal{C} = \Sigma_{i=1}^{S} \Sigma_{t=w}^{T} \Gamma(\Phi(\hat{\mu_i^t}, \hat{\sigma_i^t}), Y_i^t) \tag{7}$$

$$\mathcal{K} = \frac{1}{2} \Sigma_{i=1}^{S} \Sigma_{t=w}^{T} (1 + log(\hat{\sigma_i^t})^2 - \hat{\mu_i^t}^2 - (\hat{\sigma_i^t})^2 \tag{8}$$

where the loss function $\mathcal{L}$ for the VAE model is decomposed into three distinct components: $\mathcal{E}$ for reconstruction loss, $\mathcal{K}$ for Kullback-Leibler divergence, and $\mathcal{C}$ for the classification loss. Moreover, the variables w, S and T represent the window size, the number of stocks in the dataset, and the maximum time-step in the training set, respectively.

The VAE model undergoes training separately from the other components of the proposed model. The dataset necessary for training the VAE model comprises time-series data of stock historical prices divided into windows of size w, with the corresponding labels based on equation (4).

3.1.1. Stock Switching

In the second part of our method, we propose an Actor-Critic neural network architecture for generating optimal stock portfolios over a set of liquid coins in the cryptocurrencies market. The Actor network learns a policy that selects one of the available stocks based on the current state of the market and the uncertainty estimates on current price trend of each stock provided by the VAE. The Critic network evaluates the value of the chosen actions and provides feedback to update the policy.

The Actor-Critic architecture leverages reinforcement learning techniques to optimize the stock portfolio management strategy over time, taking into account both immediate rewards (e.g., profit/loss) and long-term objectives (e.g., investment risk and returns in long run). By incorporating uncertainty measurements from the VAE into the decision-making process, our model can adapt to changing market conditions and make more robust portfolio recommendations.

Overall, our method combines disentangled representation learning with deep reinforcement learning to enhance stock portfolio management by effectively measuring uncertainty and optimizing portfolio decisions in the volatile cryptocurrencies market. Through this integrated approach, we aim to improve the performance, stability, and interpretability of AI-driven investment strategies for financial applications.

The Actor network is designed to generate actions based on the input data. It consists of two fully connected layers with LeakyReLU activation functions to introduce non-linearity and facilitate learning complex patterns. The final layer of the Actor network is a Softmax layer, which normalizes the output values into a probability distribution over the available actions. This distribution determines the action to be taken at each time step.

The Critic network is responsible for evaluating the actions generated by the Actor network. It consists of two fully connected layers, similar to the Actor network. However, the last layer of the Critic network contains only one node, which outputs a scalar value representing the estimated return associated with the generated action. The Critic network utilizes a logarithmic estimate of the return values as its reward function, providing a measure of the quality of the actions taken by the Actor network.

The overall structure of the Actor-Critic model is illustrated in Figure 3 below. The Actor network generates actions based on the input data, while the Critic network evaluates these actions to provide feedback to the Actor network. This feedback loop enables the model to learn and improve its trading strategies over time.

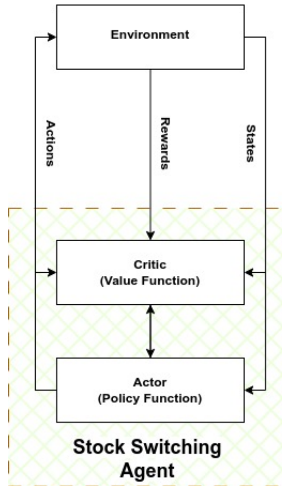

Figure 3: The structure of the proposed Actor-Critic model for portfolio management

In addition, a novel immediate reward function is proposed. It calculates the difference between the immediate return of the model's proposed action and the optimal return achievable based on future prices of each coin. This reward function is defined by Equation (9).

$$\mathcal{R}_t = log(\frac{1}{1 + [\Sigma_j a_j^t * R_j^t] - max_j(R_j^t)}) \tag{9}$$

where $\mathcal{R}_t$ denotes the immediate reward of the stock switching agent at time step t, $R^t = \{r_1^t, \cdots, r_n^t\}$ is the set of stock returns in which $r_i^t = \frac{X_i^{t+1}}{X_i^t}$ represents the next time-step return of stock i, and the jth element of the action vector at time step t is denoted by a_j^t. The best immediate stock return is calculated as $max_j(R_j^t)$ and the return of the agent's action is computed as $[\Sigma_j a_j^t * R_j^t]$. If the agent selects the best stock at time-step t, the difference between the action return and the best stock return is zero; otherwise, it is a negative value greater than -1. Equation (9) suggests that the maximum reward is attained when there is no difference between the agent's selection and the best stock.

4. Experimental Results

4.1. The Dataset

In our study, we analyze a dataset consisting of historical daily OHLC (Open, High, Low, Close) data for the ten most liquid crypto-currencies in the market over the recent years[1]. The coins chosen for our research are listed in Table 1 . It is important to note that in cases where a coin has fewer than 3838 days of price data, the price series is extended by repeating the initial data point to fill the time interval.

This dataset provides a comprehensive view of the price movements and trends in the crypto market, enabling us to evaluate the performance of the proposed method for stock portfolio management under real-world conditions. Figure 4 illustrates 100 random price series available in the dataset.

In the rest of this section we will investigate different aspects of the proposed model in portfolio management. Section 4.2 studies the features of the latent space learned by the VAE model. Section 4.3 provides an ablation study on the proposed model and a makes a comparison between the performance of different versions of the proposed model. Section 4.4 compares the results of the proposed model with the baseline models.

[1]The OHLC prices were collected using the "CoinGeckoAPI" in Python. For more information, visit `https://www.coingecko.com/en/api`

Table 1: List of selected coins in experimental data-set

Symbol	Start Date	End Date	Number of available daily records
bitcoin	2013-04-29	2023-11-01	3838
ethereum	2015-08-07	2023-11-01	3008
binancecoin	2017-09-17	2023-11-01	2236
binance-peg-xrp	2021-05-12	2023-11-01	903
solana	2020-04-10	2023-11-01	1300
litecoin	2013-04-29	2023-11-01	3838
binance-peg-polkadot	2021-05-12	2023-11-01	903
binance-peg-cardano	2021-05-12	2023-11-01	903
dogecoin	2013-12-17	2023-11-01	3606
matic-network	2019-04-26	2023-11-01	1650

4.2. Latent Space Investigation

The primary goal of the proposed model is to establish a suitable framework for quantifying the uncertainty surrounding the persistence of the prevailing stock price trend, with the aim of identifying the underlying rationale for adjusting portfolio assets based on forecasts of future price movements for individual stocks. The proposed VAE model outlined in Section 3.1 is tasked with capturing a latent space from the input price time-series data to assess the likelihood of a shift in trend direction in the future. Consequently, an examination of the characteristics of the acquired latent space within this model can offer insights into the efficacy of the proposed approach and its overarching objective.

To facilitate a clearer examination of the latent space and to shift the

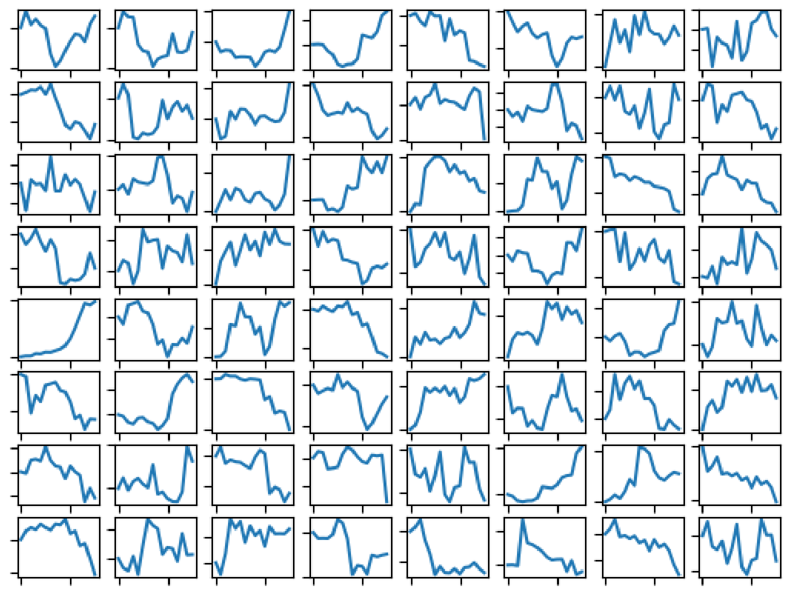

Figure 4: Sample coin price time-series from the available dataset.

emphasis towards analyzing the model's behavior within this learned space, we investigate the model's performance using a synthetic dataset comprising noisy sine waves. Given that the concept of *"mean reversion"*[2] is commonly utilized in the analysis of financial time series, we opt for a dataset featuring noisy sine waves due to their simplistic structure and resemblance to financial time series exhibiting mean reversion over extended periods.

The synthesized series in this tiny dataset are generated based on the

[2]Mean reversion is a statistical property observed in financial markets where stock prices tend to fluctuate around a long-term average, with extreme price movements typically followed by a corrective movement that brings the price closer to its historical mean. For more information about reports investigating mean reversion property in financial markets refer to Enow [32] and Corbet and Katsiampa [33].

equation (10).

$$\zeta_t^i = \epsilon_t + sin(\nu^i + \frac{t}{2\pi}) \tag{10}$$

where ζ_t^i represents the tth element in the ith time series within the generated dataset. White noise $\epsilon_t \propto N(0,1)$ is introduced at each time step to add noise to the generated sequences, and a random uniform offset $\nu^i \propto U(0, 2\pi)$ is applied to the ith sample in the dataset. Random windows based on equation (10) are generated and labeled using equation (4). The VAE model is trained using the synthetic time series dataset. Given that all input samples in the generated dataset consist of sequences of scalar values, the model assumes the cardinality of both latent features are set to 1.

Figure 5 demonstrates the VAE's generation of random samples with varying $\hat{\mu}$ and $\hat{\sigma}$. The illustration involves traversing the latent space while adjusting $\hat{\mu}$ and $\hat{\sigma}$ values from -1 to $+1$. The top left image corresponds to $\hat{\mu} = -1$ and $\hat{\sigma} = -1$, the top right image to $\hat{\mu} = +1$ and $\hat{\sigma} = -1$, the bottom left image to $\hat{\mu} = -1$ and $\hat{\sigma} = +1$, and the bottom right image to $\hat{\mu} = +1$ and $\hat{\sigma} = +1$.

Figure 5 highlights two key aspects. First, the VAE model effectively learns to separate data based on the series trend within the latent space. Samples near the center of the space, around $\hat{\mu} = 0$ and $\hat{\sigma} = 0$, exhibit a subtle change in price trend, whereas those in the corners display a clear and distinct trend. Second, the latent space is populated with quarters of sine series that exhibit transitions in trend at their beginnings or endings. This indicates that the proposed VAE model can differentiate factors associated with price changes by leveraging its classification loss to disentangle the latent space effectively.

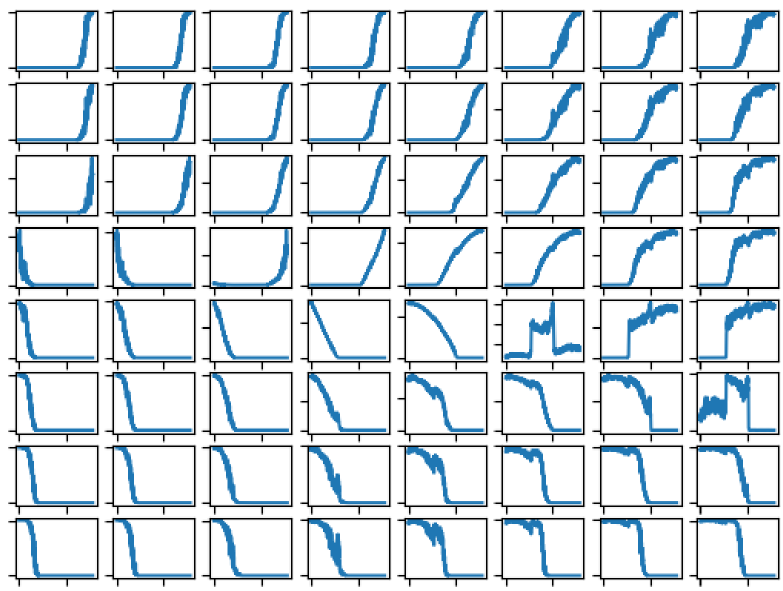

Figure 5: Illustration of the latent space learned by simple VAE over noisy sine series.

In the next experiment, the VAE model was trained using the closing price sequences from the crypto-currency dataset. Similarly, both latent features were constrained to a cardinality of 1 for this experiment. The latent space generated from this training is visualized in Figure 6. The series displayed in the upper section of the figure represent series with consistently positive price trend, whereas those in the lower section depict abrupt declines. This visualization effectively demonstrates the VAE model's ability to disentangle distinct trends within the data.

4.3. Ablation Study

To assess the effectiveness of different components of the proposed method, we conduct an ablation study where we systematically analyze the impact of key elements such as the VAE for uncertainty measurement and the Actor-

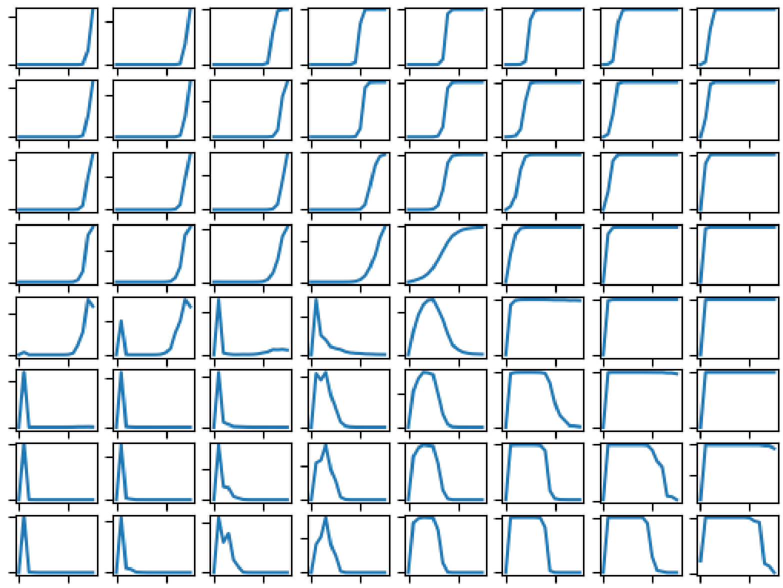

Figure 6: Illustration of the latent space learned by VAE with classifier over model performance series

Critic neural network for stock portfolio proposal. By comparing the performance of the complete model with variations that exclude specific components, we aim to understand the contribution of each part to the overall coin switching strategy.

4.3.1. Feature Extraction

In the initial experiment in ablation study, our objective is to examine the influence of the VAE model on classification of price trend movement direction. For this aim, we utilize a multi-layer perceptron (MLP) network to categorize the direction of price movements based on the definition outlined in equation (4). The outcomes of price trend prediction using various classifiers are compared in Table 2.

The baseline models considered in this study are as follows:

- **MLP**. This model is a basic MLP classifier trained directly on the input series without any feature extraction.
- **VAE-Z**. Here, the VAE is trained, and only the embedding vector Z is provided to the classifier as the extracted feature.
- **VAE-TR**. In this model, the VAE is trained, and the concatenated latent vectors $\hat{\mu}$ and $\hat{\sigma}$ are passed to the classifier as the extracted feature.
- **VAE-Y**. This model involves training the VAE, and only the VAE's embedded classifier output $\hat{Y} = \Phi(\hat{\mu}, \hat{\sigma})$ is used as the extracted feature.
- **VAE-FULL**. In this setup, the VAE is trained, and all encoder outputs containing Z, $\hat{\mu}$, and $\hat{\sigma}$ are concatenated and provided to the classifier as the extracted feature.

Table 2 presents a comparison of the baseline models' accuracy in the classification task. The results indicate that model **VAE-Y** performs the best as a feature extractor. This observation supports our hypothesis that accurately predicting the likelihood of continuation of the current stock price trend can assist portfolio management models in determining optimal points for asset switching within the portfolio.

4.3.2. Latent Space Dimensionality

The following experiment aims to explore how the size of latent vectors $\hat{\mu}$ and $\hat{\sigma}$ influences the quality of extracted features. In this study, we varied

Table 2: Comparison between baseline models accuracy in predicting the future price trend direction

Method	MLP	VAE-Z	VAE-TR	VAE-Y	VAE-FULL
Train	63.48	76.94	95.36	**95.45**	94.36
Test	61.80	73.59	93.47	**95.55**	94.72

the dimensions of the latent vectors and assessed the classifier's accuracy in predicting the next price movement direction.

The comparison of these models is presented in Table 3. The findings from this table suggest that enhancing the dimensionality of latent vectors leads to improved model performance. This improvement can be attributed to the larger latent vector size enabling the model to more effectively encapsulate the extracted information into a single vector. It is worth noting that as the dimensionality increases, more data is required for training, and therefore, performance gains may plateau after reaching a certain threshold with a fixed dataset size.

Table 3: The impact of the latent vectors size on the accuracy of the price trend prediction

Method	1	10	50	150	200
Train	95.45	96.02	98.51	**98.85**	95.46
Test	95.55	94.16	97.37	**98.23**	91.24

4.3.3. Reward Function

The reward function proposed for the coin-switching agent in equation 9 is designed based on the distance between the selected coin by the agent and the best coin at each time step. This approach aims to incentivize the agent to learn to switch to the best coin at every time step. Another commonly used reward function in this research domain is the total portfolio return at each time step. To evaluate the effectiveness of the proposed reward function, a comparison between these two models is presented in table 4. The table includes metrics such as total return on investment (**total-ROI**), maximum drawdown (**MDD**), and average return (**AR**) for each model. The results indicate that the model utilizing the proposed return function demonstrates superior performance compared to the traditional reward function commonly used in the field.

Table 4: The impact of proposed reward function

Method	total-ROI	MDD	AR
Proposed Reward	**125 %**	**-19 %**	**0.27** %
Common Reward	54 %	-20 %	0.15 %

4.3.4. Considering Risk-free Asset

One of the main underlying assumptions in our study was that mitigating minor losses would enhance portfolio management effectiveness. Consequently, we conducted an additional experiment to investigate how our model performs with and without the inclusion of a risk-free asset. Table 5 presents a comparison of the model's performance under these conditions. The results indicate that the model performs better when the risk-free asset is included, particularly in terms of metrics such as maximum drawdown. This suggests that timely switching to the risk-free asset can help prevent losses in the investment process.

Table 5: The impact of risk-free asset presence on model's performance

Method	total-ROI	MDD	AR
with tether	**125 %**	**-19 %**	**0.27** %
without tether	28 %	-34 %	0.11 %

4.4. Portfolio Proposals

The primary objective of the proposed model is to identify optimal time points for transitioning between different coins in the market. While the model's primary focus is distinct from portfolio management in which models are supposed to propose a combination of instruments at each time-step, we conducted a performance evaluation by comparing it against both single asset trading strategies and portfolio management models to assess its effectiveness in identifying optimal transition points between coins.

4.4.1. Baselines

The baseline models that have been compared with the proposed model are outlined below:

- **BaH(X)**. This denotes the buy and hold strategy, where coin X is purchased at the beginning of the experiment and held until the end.
- **UCRP**. Uniform Constant Rebalanced Portfolio strategy involves setting the portfolio proposal vector to a uniform vector $w^t = (\frac{1}{n}, \cdots, \frac{1}{n})$ at the start of each time step, where n represents the number of available coins and $|w^t| = n$.

- **FTW**. Following the winner strategy adjusts the portfolio proposal vector at each time step based on the historical performance of coins, with weights of winning coins revised according to their cumulative return $w^t = SoftMax(CR_t)$, where CR_t calculated as in equation (11).

$$\begin{aligned} CR_t = \{R(X_i^{1:t}) |\, X_i^{1:t} = \{X_i^1, \cdots, X_i^t\} \wedge \\ R(X_i^{1:t}) = \frac{X_i^t}{X_i^1} \wedge \\ i \in \{1, \cdots, n\}\} \end{aligned} \tag{11}$$

- **FTL**. Following the loser strategy refines the portfolio proposal vector at the beginning of each time step by adjusting the weights of losing coins based on the inverse of their historical performance $w^t = SoftMax(CR_t^{-1})$, using CR_t^{-1} computed as shown in equation (12).

$$\begin{aligned} CR_t^{-1} = \{R(X_i^{1:t})^{-1} | X_i^{1:t} = \{X_i^1, \cdots, X_i^t\} \wedge \\ R(X_i^{1:t})^{-1} = \frac{1}{1 + log(\frac{X_i^t}{X_i^1})} \wedge \\ i \in \{1, \cdots, n\}\} \end{aligned} \tag{12}$$

Table 6 presents the comparison of the performance of the proposed model against the baseline models in terms of total return on investment (RoI), maximum drawdown, and average return. In this table, the term $CoinSwitching$ denotes the proposed model with the common reward term, while $CoinSwitching^*$ represents the proposed model with the proposed reward function, which is considered the most effective version of the proposed model.

Furthermore, the diagram in Figure 7 illustrates the evolution of portfolio values for various strategies over time. It is evident from the graph that the

Table 6: Comparison of the performance of the proposed model versus portfolio management, and single asset trading strategies.

Strategy	Model	total-ROI	MDD	AR
Single-Asset	BaH(Bitcoin)	69 %	-20 %	0.17 %
	BaH(Ethereum)	14 %	-27 %	0.08 %
	BaH(Binancecoin)	-30 %	-41 %	0.06 %
	BaH(Solana)	19 %	-73 %	0.21 %
	BaH(Dogecoin)	-52 %	-59 %	0.13 %
Portfolio Management	UCRP	-14 %	-35 %	0.010 %
	FTW	-15 %	-33 %	0.017 %
	FTL	-15 %	-41 %	0.014 %
Proposed Model	$CoinSwitching$	54 %	-20 %	0.15 %
	$CoinSwitching^*$	**125 %**	**-19 %**	**0.27** %

suggested approach outperforms other strategies by mitigating minor losses. While different segments of the proposed strategy's portfolio value curve resemble those of other strategies, the key distinction lies in its ability to avert losses by switching between selected coins, thereby enhancing overall portfolio performance significantly.

Moreover, Figure 8 depicts the behaviors of the optimal coin-switching strategy over the investment period. As shown in the figure, the strategy predominantly involves switching between coins when there are noticeable declines in coin prices, while minor price fluctuations have minimal impact on the agent's decisions.

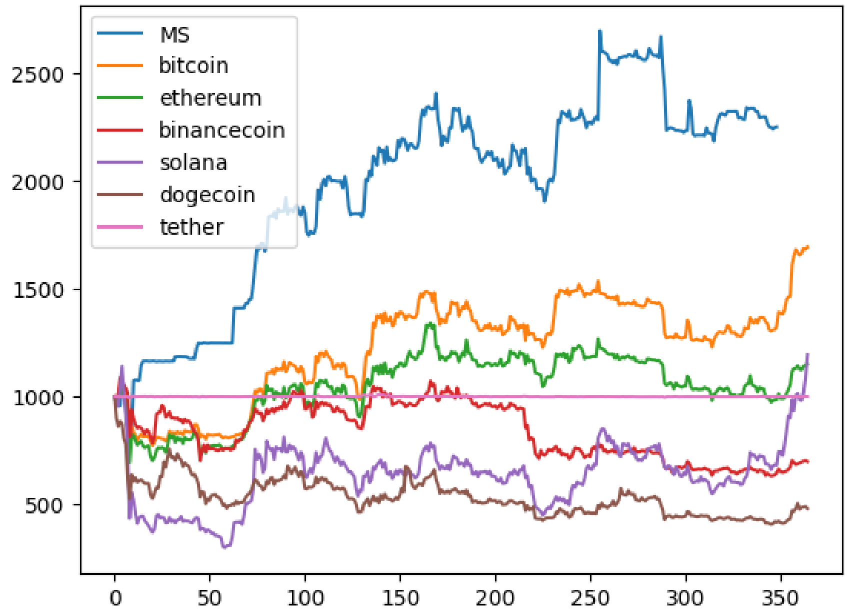

Figure 7: Comparison of the proposed model with single asset strategies.

5. Conclusion

In conclusion, the proposed model for stock portfolio management in the cryptocurrencies market demonstrates several key advantages and some limitations that are important to consider:

5.1. Advantages

1. Effective uncertainty estimation: The integration of a Variational Autoencoder (VAE) enables our model to accurately quantify uncertainty in stock price predictions, providing valuable insights for risk management and decision-making.
2. Dynamic portfolio optimization: The Actor-Critic neural network architecture allows for adaptive and dynamic portfolio rebalancing based

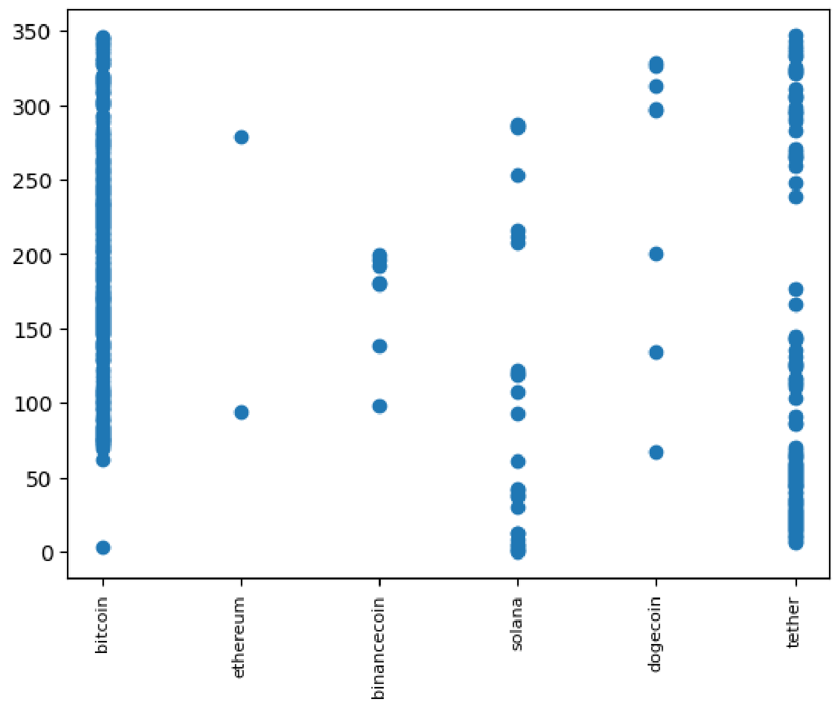

Figure 8: Model selection during investment timesteps

on changing market conditions, leading to improved performance and resilience against volatility.

3. Real-world applicability: Leveraging a comprehensive dataset of historical cryptocurrency prices, our model operates under realistic market conditions, enhancing its practical relevance and applicability for financial institutions and investors.

5.2. Weaknesses

1. Data dependency: The performance of our model heavily relies on the quality and availability of historical asset price data during the training phase, which may limit its effectiveness in scenarios where data is scarce

or unreliable.

Overall, the benefits of our proposed model outweigh its limitations, as it offers a data-driven approach to assess uncertainty of price trends in stock portfolio management in the volatile cryptocurrencies market. By addressing uncertainties, optimizing portfolios dynamically our method provides a valuable tool for investors seeking to navigate the complexities of cryptocurrency trading with enhanced confidence and efficiency. Further research and refinement of the model could help mitigate its limitations and unlock even greater potential for AI-driven financial strategies in the future.

References

[1] H. Markovitz, Portfolio selection: Efficient diversification of investments, NY: John Wiley (1959).

[2] H. Markowitz, Portfolio selection, the journal of finance. 7 (1), N 1 (1952) 71–91.

[3] B. Li, S. C. Hoi, Online portfolio selection: A survey, ACM Computing Surveys (CSUR) 46 (2014) 1–36.

[4] A. Thakkar, K. Chaudhari, Fusion in stock market prediction: a decade survey on the necessity, recent developments, and potential future directions, Information Fusion 65 (2021) 95–107.

[5] J.-C. Chen, Y. Zhou, X. Wang, Profitability of simple stationary technical trading rules with high-frequency data of chinese index futures,

Physica A: Statistical Mechanics and its Applications 492 (2018) 1664–1678.

[6] P. Liu, Y. Zhang, F. Bao, X. Yao, C. Zhang, Multi-type data fusion framework based on deep reinforcement learning for algorithmic trading, Applied Intelligence 53 (2023) 1683–1706.

[7] P. Gomber, M. Haferkorn, High frequency trading, in: Encyclopedia of Information Science and Technology, Third Edition, IGI Global, 2015, pp. 1–9.

[8] A. Asadi, R. Safabakhsh, Multi-level graph neural network for information fusion in learning stock market dynamics, Available at SSRN 4423354 (2023).

[9] B. Li, P. Zhao, S. C. Hoi, V. Gopalkrishnan, Pamr: Passive aggressive mean reversion strategy for portfolio selection, Machine learning 87 (2012) 221–258.

[10] T. Pennanen, Introduction to convex optimization in financial markets, Mathematical programming 134 (2012) 157–186.

[11] J. Ayala, M. García-Torres, J. L. V. Noguera, F. Gómez-Vela, F. Divina, Technical analysis strategy optimization using a machine learning approach in stock market indices, Knowledge-Based Systems 225 (2021) 107119.

[12] M. Agrawal, P. K. Shukla, R. Nair, A. Nayyar, M. Masud, Stock prediction based on technical indicators using deep learning model., Computers, Materials & Continua 70 (2022).

[13] M. Taghian, A. Asadi, R. Safabakhsh, A reinforcement learning based encoder-decoder framework for learning stock trading rules, arXiv preprint arXiv:2101.03867 (2021).

[14] M. Taghian, A. Asadi, R. Safabakhsh, Learning financial asset-specific trading rules via deep reinforcement learning, Expert Systems with Applications 195 (2022) 116523.

[15] S. Carta, A. Corriga, A. Ferreira, A. S. Podda, D. R. Recupero, A multi-layer and multi-ensemble stock trader using deep learning and deep reinforcement learning, Applied Intelligence 51 (2021) 889–905.

[16] Y.-C. Lin, C.-T. Chen, C.-Y. Sang, S.-H. Huang, Multiagent-based deep reinforcement learning for risk-shifting portfolio management, Applied Soft Computing 123 (2022) 108894.

[17] Z. Hao, H. Zhang, Y. Zhang, Stock portfolio management by using fuzzy ensemble deep reinforcement learning algorithm, Journal of Risk and Financial Management 16 (2023) 201.

[18] Y. Zhao, H. Du, Y. Liu, S. Wei, X. Chen, F. Zhuang, Q. Li, G. Kou, Stock movement prediction based on bi-typed hybrid-relational market knowledge graph via dual attention networks, IEEE Transactions on Knowledge and Data Engineering (2022).

[19] M. Gullotto, Portfolio management and Deep learning: Reinforcement learning and Transformer applied to stock market data, Ph.D. thesis, Politecnico di Torino, 2021.

[20] D. Kisiel, D. Gorse, Portfolio transformer for attention-based asset allocation, in: International Conference on Artificial Intelligence and Soft Computing, Springer, 2022, pp. 61–71.

[21] W. F. Sharpe, The sharpe ratio, Streetwise–the Best of the Journal of Portfolio Management 3 (1998) 169–185.

[22] Y. Liu, D. Mikriukov, O. C. Tjahyadi, G. Li, T. R. Payne, Y. Yue, K. Siddique, K. L. Man, Revolutionising financial portfolio management: The non-stationary transformer's fusion of macroeconomic indicators and sentiment analysis in a deep reinforcement learning framework, Applied Sciences 14 (2023) 274.

[23] M. Abdar, F. Pourpanah, S. Hussain, D. Rezazadegan, L. Liu, M. Ghavamzadeh, P. Fieguth, X. Cao, A. Khosravi, U. R. Acharya, et al., A review of uncertainty quantification in deep learning: Techniques, applications and challenges, Information fusion 76 (2021) 243–297.

[24] J. Du, Mean–variance portfolio optimization with deep learning based-forecasts for cointegrated stocks, Expert Systems with Applications 201 (2022) 117005.

[25] J.-H. Syu, M.-E. Wu, J.-M. Ho, Portfolio management system with reinforcement learning, in: 2020 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE, 2020, pp. 4146–4151.

[26] H. M. Abdulsahib, F. Ghaderi, Cross-domain disentanglement: A novel approach to financial market prediction, IEEE Access (2024).

[27] H. M. Abdulsahib, F. Ghaderi, et al., Glad: Global–local approach; disentanglement learning for financial market prediction, IET Signal Processing 2023 (2023).

[28] S. Duan, L. Matthey, A. Saraiva, N. Watters, C. P. Burgess, A. Lerchner, I. Higgins, Unsupervised model selection for variational disentangled representation learning, arXiv preprint arXiv:1905.12614 (2019).

[29] Y. Li, Z. Chen, D. Zha, M. Du, J. Ni, D. Zhang, H. Chen, X. Hu, Towards learning disentangled representations for time series, in: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, 2022, pp. 3270–3278.

[30] Y. Li, Z. Chen, D. Zha, M. Du, D. Zhang, H. Chen, X. Hu, Learning disentangled representations for time series, arXiv preprint arXiv:2105.08179 (2021).

[31] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, A. Lerchner, Understanding disentangling in β-vae, arXiv preprint arXiv:1804.03599 (2018).

[32] S. T. Enow, Investigating mean reversion in financial markets using hurst model, International Journal of Research in Business and Social Science (2147-4478) 12 (2023) 197–201.

[33] S. Corbet, P. Katsiampa, Asymmetric mean reversion of bitcoin price returns, International Review of Financial Analysis 71 (2020) 101267.