International Journal of Finance & Economics

WILEY

RESEARCH ARTICLE OPEN ACCESS

# Forecasting Digital Asset Return: An Application of Machine Learning Model

Vito Ciciretti[1] | Alberto Pallotta[2] | Suman Lodh[3] | P. K. Senyo[4] | Monomita Nandy[5]

[1]Independent Researcher, Berlin, Germany | [2]Middlesex University Business School, Middlesex University, London, UK | [3]Kingston Business School, Kingston Hill Campus, Kingston Upon Thames, Kingston University London, Surrey, UK | [4]Department of Decision and Risk, University of Southampton Business School, University of Southampton, Southampton, UK | [5]Brunel Business School, Brunel University of London, Uxbridge, UK

**Correspondence:** Monomita Nandy (monomita.nandy@brunel.ac.uk)

**Received:** 21 March 2023 | **Revised:** 21 September 2024 | **Accepted:** 12 October 2024

**Keywords:** bitcoin | digital asset | double deep Q-learning | forecasting price | machine learning | reinforcement learning | time-series

## ABSTRACT

In this study, we aim to identify the machine learning model that can overcome the limitations of traditional statistical modelling techniques in forecasting Bitcoin prices. Also, we outline the necessary conditions that make the model suitable. We draw on a multivariate large data set of Bitcoin prices and its market microstructure variables and apply three machine learning models, namely double deep Q-learning, XGBoost and ARFIMA-GARCH. The findings show that the double deep Q-learning model outperforms the others in terms of returns and Sortino ratio and is capable of one-step-ahead sign forecast of the returns even on synthetic data. These critical insights in forecasting literature will support practitioners and regulators to identify an economically viable cryptocurrency forecasting return model.

## 1 | Introduction

In recent years, there has been growing interest in Bitcoin investment as the cryptocurrency gains global popularity and acceptance in some countries (Xie, Chen, and Hu 2020; Rehman, Asghar, and Kang 2020). There are more than 81 million crypto wallets user across the world as of November 2022 (Statista 2021). The rapid evolution of Bitcoin trading over the past years has often raised concerns among investors in terms of overvaluation, overreaction, and irrational behaviour of the cryptocurrency prices (Amini et al. 2013; Borgards and Czudaj 2020; Corbet and Katsiampa 2020; Mattke et al. 2021). Investors, market practitioners, and regulators have shown vigorous interest in understanding and explaining the movements of cryptocurrency prices in detail (Raimundo Junior et al. 2020; Signature Bank failure, March 2023). Nevertheless, understanding the drivers of changes in cryptocurrency prices remains an open question as the application of econometric and statistical modelling has largely failed to adequately provide actionable insights in forecasting Bitcoin prices (Chen et al. 2021; Wang, Andreeva, and Martin-Barragan 2023).

Given that Bitcoin transactions generate large data sets that can provide critical insights, it is therefore important to explore if big data analytical tools such as machine learning could be useful solution to overcome the limitation in forecasting Bitcoin prices (Tofangchi et al. 2021). In addition, cryptocurrencies like Bitcoin are less efficient when compared to the traditional financial assets (Al-Yahyaee, Mensi, and Yoon 2018), in the context of volatility. Even though, we observe a decrease in this volatility over the time, but the historical volatility of Bitcoin remains almost 10 times higher than gold and several conventional currencies (Bianchetti, Ricci, and Scaringi 2018).

Moreover, Bitcoin, possess a combination of properties of other traditional financial and speculative asset and has a low correlation with other financial instruments traded in the financial market (Klein, Thu, and Walther 2018). Thus, following the literature, we use the highly liquid cryptocurrency, Bitcoin in this study (Amiram, Jørgensen, and Rabetti 2022). We summarise the main socio-economic impact of Bitcoin. Over the years, we mainly observe that researchers either focus on one type of value creation mechanism (Kitchens et al. 2018) or adopt the additive approach (Grover et al. 2018). Such approaches are difficult to apply in a complex set-up, such as explaining complex financial relationships (Gradojevic et al. 2021; Newell and Marabelli 2015). However, in recent years, there have been several recommendations on the importance of applying big data analytics to generate valuable insights about business operations (Grover et al. 2018). As suggested by Hendershott et al. (2021), the adoption of machine learning models can be a game changer in the context of investment in cryptocurrency trading (Müller et al. 2016). However, the application of the machine learning model invites the challenge of identifying the algorithm that possesses the capability to forecast the cryptocurrency return with real time data. Thus, in this research we ask the following question: Does double deep Q-learning model outperforms the other popular models (XGBoost and ARFIMA-GARCH) in forecasting cryptocurrency return? Because of wider discussion about prediction efficiency of XGBoost and ARFIMA-GARCH model in forecasting, we decided to compare their performance with the double deep Q-learning model.

Extant literature documents the challenges in smoothing and forecasting Bitcoin prices (Miller et al. 2019; Jana, Ghosh, and Das 2021; Kraaijeveld and De Smedt 2020), especially due to its high volatility (Li et al. 2021; Yaya et al. 2021; Gradojevic and Tsiakas 2021). Moreover, while advanced machine learning algorithms are capable to deliver exceptional in-sample performances, the ability to generalise out-of-sample remains inherent to a more limited reach (Keilbar and Zhang 2021; Anyfantaki, Arvanitis, and Topaloglou 2021). Out-of-sample performance is the most important performance indicator to find whether a financial model will deliver the expected performance in the real world (Catania and Grassi 2022; Liang et al. 2020). The lack of logical understanding of outputs generated by complex algorithms is often regarded as black boxes and because of such complication, the application of machine learning models remains limited. So far, prior literature has largely focused on complex models such as reinforcement learning (Tofangchi et al. 2021). Nevertheless, practitioners such as regulators, and some individual investors are unlikely to favour the implementation of complex, black box-deemed algorithms over simpler ones where the relation between cryptocurrency returns and explanatory variables can be easily explained and interpreted. Thus, to contribute to the ongoing discussion on forecasting cryptocurrency returns, we examine the research question by applying machine learning algorithms and displaying various levels of their complexity. In addition, to provide critical insights about the model interpretability, we conduct out-of-sample performance test.

The sample time-series data set consists of daily, open, spot prices of Bitcoin for the period February 2012–December 2023 sourced from Quandl (3290 daily observations). We randomly select the above period to set up a synthetic data set for the simulation purpose. In the simulation process, we aim to invest according to the algorithms and calculate the investment performance. We selected Bitcoin in this study because of its popularity, maturity, market position as the leading cryptocurrency (Gradojevic et al. 2021) and for its long-term social impact.[1] Our unique finding shows that the double deep Q-learning model outperforms the others in terms of returns and Sortino ratio and is capable of one-step-ahead sign forecast of the returns even on synthetic data. According to these results, the success of machine learning models in the prediction of cryptocurrency returns is re-established and double deep Q-learning model adds an extra layer of confidence about its predictability in the forecasting literature.

Our contributions are threefold—first, previous research has conflicting views on the suitability of these models. To our best knowledge, this is the first study to resolve debates with empirical evidence on the effectiveness of machine learning models in predicting cryptocurrency returns. Second, the study uses the Sortino ratio instead of the Sharpe ratio, focusing on downside volatility to provide a more accurate risk assessment. Thus, it highlights the difference in risk considerations between proprietary traders and investment funds/banks, emphasising the latter's focus on controlled risk scenarios. Finally, to address the unreliability of historical data (Pintelas et al. 2020) we propose using a Variational Autoencoder to create synthetic data sets for out-of-sample performance evaluation.

The rest of the paper is organised as follows. Section 2 further offers a general overview of the relevant literature and an in-depth discussion of reinforcement learning models. Section 3 outlines the research methodology by discussing the data and the training architectures. Section 4 introduces the out-of-sample backtesting methodology based on synthetic data extracted from the Variational Autoencoder model. Section 5 compares the out-of-sample performances in terms of investment strategies and classification statistics. Section 6 concludes and addresses potential future work.

## 2 | Literature Review

Introduced by Sutton and Barto (1998), the literature on reinforcement learning relishes several extensions that enrich its original scope and application opportunities to various industries (Van Moffaert, and Now'e 2014). The application of reinforcement learning includes self-driving cars, mastering board games such as the AlphaZero chess engine (Silver et al. 2017), and so on. Among the extensions of reinforcement learning, the basic Q-learning algorithm (Watkins 1989) was revised as double Q-learning and can address an overestimation bias of the basic Q-learning model (Van Hasselt, Guez, and Silver 2016). Moreover, recent reinforcement learning literature focuses on prioritised experience replay to improve data efficiency (Schaul et al. 2015), the duelling network architecture (Wang et al. 2016) and noisy double Q-learning (Fortunato et al. 2017) for stochastic network layers to improve exploration. These several contributions are blended into a rainbow model showing that most of the extensions are complementary and capable of producing outperforming performances (Hessel et al. 2018).

On the other hand, financial literature has adopted reinforcement learning models in the recent years. In our survey of literature, we find an excellent application of reinforcement learning in financial markets (Fischer 2018). Lee et al. (2007)

apply multiple Q-learning agents to a stock-trading framework focused on Korean stock market. Jiang, Xu, and Liang (2017) use a 30-min cryptocurrency trading strategy and apply an ensemble of identical independent reinforcement learning evaluators based on a convolutional neural network, a recurrent neural network, and a long-short term memory model. Sadighian (2020) applies deep reinforcement learning to create an intelligent market-making strategy testing seven reward functions, extending the previous reinforcement learning market-making models based on time-based event environments. Xiong et al. (2018) show how a Deep Deterministic Policy Gradient can build an optimal portfolio that outperforms the traditional mean-variance asset allocation and a buy-and-hold strategy on the Dow Jones Industrial Average. Wu et al. (2020) apply the Gated Recurrent Unit model to extract informative financial features that are eventually used to extract intrinsic characteristics of the US stock market. Besides reinforcement learning, several neural network applications have been deployed to the problems of financial forecasting, portfolio optimization, investment strategies and risk management. In the work of Chen, Leung, and Daouk (2003), we observe one of the first applications of neural networks in finance, where they predicted the return direction of the Taiwanese Stock Exchange index by means of a probabilistic neural network and showing its capability to outperform non-neural network-based strategies. In the field of time series forecasting, recurrent neural networks have proven to be particularly useful, thanks to their stateful architecture which allows modelling of serial autocorrelation. Edet (2017), predicts the movements of the S&P 500 index using a recurrent neural network and its variations, namely the long-short-term memory and the gated recurrent unit. They applied the networks to 14 economic variables and 4 levels of hidden layers.

Following Baillie, Chung, and Tieslau (1996) and Gianfreda and Grossi (2012) we use the ARFIMA-GARCH regression model and Chen and Guestrin (2016) for the XGBoost model. After critically examining the relevant literature, we cannot find any evidence of studies focusing on the out-of-sample performance via synthetic data sets produced with a Variational Autoencoder of a Bitcoin investment strategy based on reinforcement learning, XGBoost and the ARFIMA-GARCH regression model.

In the extant literature, some reinforcement learning approaches, frameworks and models have been proposed, see Table A1 in Appendix A. Despite the contributions of these studies, some limitations exist. First, we observe that models in existing studies (see Wu et al. 2020) have largely relied on an out-of-sample performance evaluation on a single set of historical data, making it difficult to generalise the results. In literature, we mainly observe out-of-sample performance evaluation on historical data, with limited focus on synthetic data (Catania and Grassi 2022). When the training data are highly imbalanced (especially relevant for cryptocurrencies given the highly volatile and leptokurtic distributions), then models using synthetic data could generate more accurate results when applied on real data. One of the most efficient ways to generate a synthetic data set is by means of a Variational Autoencoder (VAE). In this technique, the encoder compresses the original data set into a more compact structure, which is, in turn, transmitted to the decoder to generate an output which represents the original data set with some noise. The lack of attention by scholars on synthetic data sets motivates us to focus on the out-of-sample performance. Second, we observe that models in existing studies (see Li, Zheng, and Zheng 2019) target the maximisation of total returns or cumulated profits and do not use explanatory variables, rather only focus on time-series dependencies. Algorithms targeting total returns or cumulated profits result in extreme portfolios, with large exposures in a single asset that widely vary over time. On the other hand, targeting a risk-adjusted measure—such as the Sortino ratio—results in more stable, less extreme investment strategies. However, cryptocurrencies are characterised by complex distributions that cannot be explained by their respective univariate time series, rather the usage of explanatory variables is deemed necessary.

## 3 | Methodology

### 3.1 | Reinforcement Learning—Model Specification

Reinforcement learning is a reward-driven process where an agent learns to interact with a complex environment via trial-and-error to achieve rewarding outcomes (Sutton and Barto 1998). The agent learns to maximise the reward by choosing the best action in each state of the environment. At the heart of reinforcement learning lies the explore-exploit dilemma. In practice, the agent faces the dilemma of either exploiting what has been learned thus far or exploring to gain additional knowledge at the risk of recording lower payoffs.

Consider an agent within the environment $\Omega$ in discrete time with single step $t = 1, 2, \ldots, n$ coupled with the triplet action, state, and reward $(a_t, s_t, r_t)$. At each time $t$, the agent is in state $s_t$ and selects an action $a_t$. The interaction with the environment $\Omega$ returns the next reward $r_t + 1$ and the next state $s_t + 1$. The entire set of states and environment rules for transitioning from one state to another may be represented as a Markov decision process. In fact, the current state $s_t$ encompasses all the information needed by the environment for processing state transitions and assigning rewards. Therefore, an agent tries to choose an action $r_t \in A$ that maximises the expected conditional future reward. This approach is named Q-learning (Watkins 1989), a form of temporal difference learning (Sutton and Barto 1998).

Deep reinforcement learning involves the usage of deep neural network architectures to serve as function approximators. A deep-Q-network is a multi-layered neural network $f(x): \mathbb{R}^n \rightarrow \mathbb{R}^m$ that outputs $Q(a_t, s_t)$, where $a_t \in A$, $s_t \in S$ and $r_t \in R$. As a result, the objective of the reinforcement learning becomes learning the optimal set of neural network weights $w_t \in W$ that minimises a loss function. The latter, however, is an unobservable process which depends on the future combinations of $(a_t, s_t)$. As such, one needs to solve a dynamic programming algorithm in the form of a Bellman equation.

This optimization mechanism, however, would lead to quickly forgetting rare outcomes as well as it is prone to strongly correlated updates that violate the *i.i.d.* assumption of stochastic gradient descent algorithms. Experience replay (Lin 1992) addresses these issues as experience is stored in a replay memory from where the network can draw input values, thus potentially including long-term learning and rare outcomes. At the same time, this allows mixing more with less recent experiences for the updates, leading to an update distribution closer to being *i.i.d.* (Mnih et al. 2015) introduce experience replay to the deep-Q-network architecture. Moreover, it would be more efficient to sample more frequently replay batches where there is more to learn. To do so, (Schaul et al. 2015) introduced prioritised experience replay.

We define the following reinforcement learning environment composed of:

- State $S = [p, h]$: a set including the univariate time-series of prices $p \in \mathbb{R}_+$ and the number of contracts held $h \in \mathbb{R}_+$;
- Action $S = [1, -1]$: a set of actions where 1 represents a buying order and $-1$ a selling one. The action leads to changes in the holding balance $h \in \mathbb{R}_+$;
- Reward $r(s_t, a_t, s_{t+1})$: the change of the cumulated return of the investment strategy when action at is taken in state $s_t$ and eventually leading to the new state $s_{t+1}$ and where $r_t = \ln\left(\frac{p_t}{p_{t-1}}\right)$;
- No contract accumulation is possible; hence a single contract can be traded each time.

The goal is to design an investment strategy that maximises the Sortino ratio $SR_t$ for the investors in the Bitcoin exchange:

$$SR_t = \frac{r_t - rf_t}{\sigma_{t,semi}} \tag{1}$$

where $\sigma_{t,semi}$ is the semi-standard deviation of the returns generated by the investment strategy, and $rf_t$ is the risk-free rate which we set equal to the 3 months LIBOR/SOFR rate. We choose to target the Sortino ratio to limit the downside volatility on the strategy since the Bitcoin market is characterised by frequent and pronounced volatility spikes.

### 3.2 | XGBoost—Model Specification

XGBoost is an implementation of gradient-boosting machines belonging to the broader collection of tools under the umbrella of the Distributed Machine Learning Community. Its widespread adoption followed winning the Higgs Machine Learning Challenge. The XGBoost library provides two wrapper classes that allow the random forest implementation provided by the library to be used with the scikit-learn machine learning library. One of the most important differences between XGBoost and Random Forest is that the XGBoost always gives more importance to functional space when reducing the cost of a model while Random Forest tries to give more preferences to hyperparameters to optimise the model. As such, while the XGBoost model often achieves higher accuracy than decision trees, it sacrifices the interpretability of the explanatory variables. Unlike gradient boosting that works as gradient descent in function space, a second order Taylor approximation is used in the loss function to make the connection to the Newton-Raphson method. For an overview of XGBoost models, see Chen and Guestrin (2016).

### 3.3 | GARCH Model—Model Specification

Generalised Autoregressive Conditional Heteroskedasticity (GARCH) is a statistical model used for analysing time-series data where the variance error is serially autocorrelated. GARCH models assume that the variance of the error term follows an autoregressive moving average process. GARCH models are commonly employed in modelling financial time series that exhibit time-varying volatility and volatility clustering. For an overview of GARCH models, see Bollerslev (1987).

## 4 | Data

The data set used is a time-series of daily, open, spot prices of Bitcoin futures (BTC) for the period February 2012–December 2023 sourced from Quandl (3290 daily observations). The choice of the data set has been driven by data completeness and the availability of many explanatory variables. Due to the persistent correlation and Granger causality between Bitcoin prices and other cryptocurrencies (Ghorbel and Jeribi 2021), investigation of the former allows for establishing a gauge over the entire market. In Table 1, we define every explanatory variable used in this study.

We fractionally differentiate each time series to achieve stationarity. To estimate the fractional parameter, we used the algorithm of Geweke and Porter-Hudak (2008), whose estimator is based on the regression equation using the periodogram function as an estimate of the spectral density.

We trained each model on a training sample composed of 60% of the observations and cross-validated the in-sample estimations by means of k-fold cross-validation, with $k = 5$. Since a Portmanteau test rejected the null hypothesis of identically and independently distributed data at any confidence interval, the cross-validation was purged to take into consideration the serial correlation of the data. As such, in Figure 1 we formed the

**TABLE 1** | The variables composing the data set used in this study.

| Name | Description |
|---|---|
| BTC | Bitcoin price in USD |
| AVBLS | Average block size |
| HRATE | Hash rate (diff.) |
| ETRAV | Estimated transaction volume (diff.) |
| NTRBL | Transactions per block (diff.) |
| NADDU | Transactions excluding popular addresses (diff.) |
| NTREP | Number of transactions (diff.)—1.6 |

*Note:* In this table we define the variables used in this study. The abbreviations are defined in the second column and in other tables and in the text we used the abbreviations.

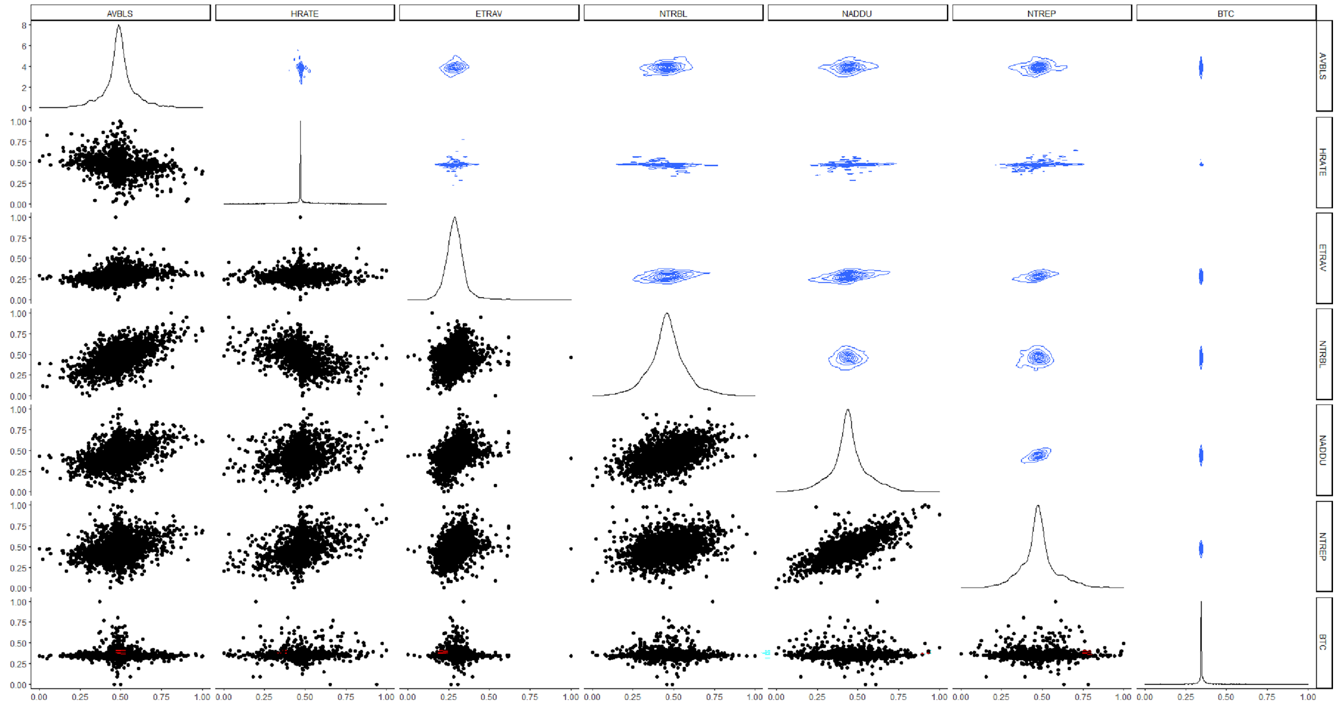


**FIGURE 1** | Scatter plots and densities for each pair of variables as well as univariate distributions. [Colour figure can be viewed at wileyonlinelibrary.com]

validation folds with adjacent observations, rather than with randomly picked ones.

The model specifications are fine-tuned via grid search. For the multilinear perceptron within the reinforcement learning model, we use two hidden layers with leaky RELU activation function, the Glorot kernel initializer, a Ridge regularizer of 0.01 and a Lasso regularizer of 0.01. The training is done via Stochastic Gradient Descent with a Nesterov momentum of 0.6. The loss function is binary cross-entropy. We chose the multilinear perceptron model as more complex models, such as convolutional neural networks or recurrent neural networks, improved the in-sample fitting but worsened the model's capability to generalise out-of-sample. In other words, the reduced bias due to enhanced model complexity is more than offset by the increased variance. For the XGBoost, we use a Ridge regularizer of 0.05 and a Lasso regularizer of 0.02, a maximum tree depth of 5 and accuracy as the evaluation metric. In both cases, a validation sample of 20% of the training observation was used to apply the purged fivefold cross-validation algorithm. For the univariate ARFIMA-GARCH model, we use an eGARCH(1,1) specification with skewed *t*-student distribution and an ARIMA(2,0,2) for the mean model.

## 5 | Out-of-Sample Validation Methodology

To address the research question, we follow the literature and apply the double deep Q-learning, as it avoids the overestimation problem associated with Q-learning. Application of machine learning is challenging in the case of stock market forecasting because of the noisy nature of the historical data. Competitive machine learning approaches mostly act in a supervised manner, ignoring several macro factors affecting the financial market, which leads to over-fitting. As reinforcement learning approaches can learn the process to maximise a return function during the training stage, we can minimise the overfitting problem. Thus, we use a Q-learning agent, which can be trained several times using the same training data and can be important in the real-world stock markets (Carta et al. 2021). However, double deep Q-learning might underestimate the action values at times. Since neural networks and machine learning models are prone to overfitting in the training sample, the focus should be on the capability to generalise out-of-sample. For this reason, we evaluate the models on a strict out-of-sample framework based on 10 synthetic data sets generated by means of a variational autoencoder (VAE), which was introduced by Kingma and Welling (2013). The VAE reduces the reconstruction error between the input and output of the network when applied on real data. Thus, VAE improves the generated data quality by minimising the distribution distances between the real posterior and the estimated one (Tables C1 and C2) in Appendix C.

The next of this paragraph introduces the VAE model used to produce the synthetic data sets.

Consider a data set $X = \left\{X^{(i)}\right\}_{i=1}^{N}$ composed of N *i.i.d.* samples coming from a random variable $x$. Let's assume that the data is generated by a random process involving an unobserved continuous random variable $z$. The process consists of first generating a value $z$ from some prior distribution $p_\theta(z)$ to then generating a value $x^i$ from the conditional distribution $p_\theta(x|z)$. Let's assume that the PDFs of $p_\theta(z)$ and $p_\theta(x|z)$ are differentiable almost everywhere with respect to $z, \theta$. However, the true parameters $\theta$ and the values of the latent variable $z$ are unknown. The objective is to find an efficient neural network approximation for the latent variable $z$ as this would allow to mimic the hidden random process and generate a synthetic data set that resembles the real data. To do so, we employ a Variational Auto-Encoder. Assume the prior over the latent variables to be a centred isotropic multivariate Gaussian $p_\theta(z) = N(z, 0, I)$. Let $p_\theta(x|z)$ be a multivariate Gaussian with $\theta$ estimated via a fully connected neural network with a single hidden layer. The true posterior is intractable but assuming that is approximated by a Gaussian distribution with an approximately diagonal covariance, then the variational approximate posterior is a multivariate Gaussian with a diagonal covariance structure:

$$\log q_\theta\left(z|x^i\right)\ )\log N\left(z, \mu^i, \sigma^i I\right) \tag{2}$$

where $q_\theta\left(z|x^i\right)$ is based on an alternative technique for sampling $z$ such as Monte Carlo and $\left(\mu^i, \sigma^i\right)$ are the mean and standard deviation of the approximate posterior which are outputted by the neural network as nonlinear functions of $x^i$ and the variational parameters $\phi$.

Afterwards, one simply need to sample from the posterior $z^{i,l} \sim q_\theta\left(z|x^i\right)$ with $z^{i,l} = g_\theta\left(x^i, \epsilon^l\right) = \mu^i + \sigma^i \epsilon^l$, where $\epsilon^l \sim N(0, I)$. It can be proven that the Kullback–Leibler divergence can be computed without estimation and the resulting estimator for the data point $x^i$ is given by:

$$\begin{aligned}\mathcal{L}\left(\theta, \phi, x^i\right) \cong \frac{1}{2}\sum_{j=1}^{J}\left(1 + log\left(\left(\sigma_j^{(i)}\right)^2\right) - \left(\mu_j^{(i)}\right)^2 - \left(\sigma_j^{(i)}\right)^2\right) \\ + \frac{1}{L}\sum_{l=1}^{L}\log p_\theta\left(x^i|z^{i,l}\right)\end{aligned} \tag{3}$$

where $\log p_\theta\left(x^i|z^{i,l}\right)$ is a Gaussian fully connected neural network decoding term.

The robustness of the VAE used in our study is coherent with other studies (Camuto et al. 2021). Given this framework, we produce 10 multivariate synthetic data sets composed of 1316 observations (40% of out-of-sample observations). Once again, focusing on out-of-sample performance is essential in financial applications to avoid in-sample overfitting. On the other hand, the models have been trained and validated on the 60% of in-sample observations. The model specification is fine-tuned via grid search. For the VAE, we use two hidden layers with five hidden units activated by means of the leaky RELU function, initialised with the Glorot kernel initialiser, with a Ridge regularizer of 0.02 and a Lasso regularizer of 0.01. The training is done via Stochastic Gradient Descent with a Nesterov momentum of 0.6. The loss function targets the representation error via the mean squared error.

Figures B1–B7 in Appendix B compare each variable's density plot in the original data with those in the 10 synthetic data sets. The grey-shaded area is the distribution of the synthetic variable in each data set, while the red-shaded one is the distribution of the same variable in the original data set. As is visible, the synthetic distributions closely match the original data set with few discrepancies, which are mostly limited to higher standard deviation around the mean and rare differences in the tails.

## 6 | Results

In this section, we discuss our findings. We follow the extant literature (e.g., Ding, Cui, and Zhang 2022; Wang, Andreeva, and Martin-Barragan 2023; Kumar et al. 2024) to evaluate the performance of our double deep Q-learning model and compare its performance with two existing models, namely XGBoost and ARFIMA-GARCH. In line with our aim of offering an approach to forecast Bitcoin prices, we focused our evaluation on 10 synthetic data sets in terms of an investment strategy based on their one-step-ahead sign forecasts and in terms of their performances as classifiers. According to Huang (2021) the significance of model-predicted signs is crucial for investment strategies and indicates while ordinary least squares (OLS) estimators generally yield better Sharpe ratios, sign regression can outperform for certain assets. Similarly using sign prediction, Sebastião and Godinho (2021) examine the predictability of digital currency using linear models, random forests, and support vector machines. They show how the combination of multiple models can achieve annualised Sharpe ratios of 80.17% for Ethereum and 91.35% for Litecoin (despite changes in trading costs and market volatility). So, we first illustrate how the sign predictions are translated into investment strategies. Starting from the double deep Q-learning model, the output of the learning process is the triplet state, action, and reward $(a_t, s_t, r_t)$. As such, given a state $s_t$, the chosen actions $a_t \in [1, -1]$ are directly translated into rewards $r_t$. Hence, the profit and loss of the investment strategy are the direct output of the learning algorithm.

The XGBoost model, instead, forecasts the one step-ahead probability $p_t$ of the next fractional return $r_{t+1}$ being positive (Chen et al. 2021). Hence, we convert the probability into a trading action by using a static threshold of 0.5 with a buying mechanism triggered when $p_t > 0.5$, and vice versa. This is a direct yield of having a binary classification as the evaluation objective. Finally, the ARFIMA-GARCH is a univariate regression model that forecasts the one-step-ahead fractional return. Hence, the strategy is to 'buy' when the one-step-ahead predicted return, $r_{t+1}$, is positive and vice versa. The fractional differentiation algorithm does not alter the sign of the price change. As such, increase in price reflects positive fractional returns, and vice versa. This property is important since both the XGBoost and the ARFIMA-GARCH models use fractional returns of Bitcoin prices as these are not stationary (Almaafi, Bajaba, and Alnori 2023).

Following Dos Santos and Aguilar (2024), at the next stage, we compare the investment performance achieved by the three models on the 10 synthetic data sets (also see Arian, Norouzi, and Seco 2024). To showcase the applicability of these models, we add to the comparison a naive buy-and-hold strategy on the BTC. Tables C3 and C4 in Appendix C report the descriptive statistics of the returns as well as the Sharpe and Sortino ratios achieved by each strategy in each of the 10 out-of-sample synthetic data sets. To improve comparability, we also include the average across the 10 data sets. Double deep Q-learning achieves the highest average annualised mean return of 15.8%. This suggests that the model is capable of generating substantial returns over time. However, high returns often come with higher risk. Investors should evaluate their risk tolerance and ensure they are comfortable with the potential volatility associated with this strategy. While XGBoost outperforms in terms of average annual median return (10.5%) which indicates a more consistent and stable return profile. This could be appealing to investors who prioritise stability and predictability in their investments. In addition, XGBoost may be less prone to extreme outcomes, making it a potentially safer choice for risk-averse investors. Both double deep Q-learning and XGBoost outperform the buy-and-hold strategy (7.9% annual mean and 4.8% annual median), ARFIMA-GARCH is the worst performer in terms of both metrics. At the same time, ARFIMA-GARCH strategies achieve unstable performances across the 10 data sets compared to the double deep Q-learning and XGBoost whose statistics are stable across the synthetic data sets.

Nevertheless, the double deep Q-learning records the highest annualised standard deviation across the 10 data sets, averaging 33.2% compared to 26.9% of XGBoost and 24% of ARFIMA-GARCH. Investors might use double deep Q-learning as part of a diversified portfolio to balance risk and reward, leveraging its potential for high returns while mitigating overall portfolio risk. All the models have a larger standard deviation compared to the buy-and-hold strategy whose annualised standard deviation is 20.7% (also see Gort et al. 2022). At the same time, double deep Q-learning records a large average kurtosis of 9.7x, while XGBoost and ARFIMA-GARCH achieve a kurtosis below three, improving on the 6.7x of the buy-and-hold strategy. However, the larger standard deviation and kurtosis of double deep Q-learning are offset by the largest, positive average skewness of 2.4 and the lowest downside volatility of 3.1%. In fact, the objective function of double deep Q-learning is to maximise the Sortino ratio, which embeds minimising the downside volatility while maximising the returns. This implies that its higher average standard deviation is the result of higher upside volatility. Figure 2 plots the average standard deviations while Figure 3 plots the average downside volatilities both calculated on a rolling window of 100 days. ARFIMA-GARCH returns, instead, are mostly symmetrical (positive and negative returns are roughly equal in magnitude and frequency) while the downside volatility is the highest across all the synthetic data sets. This means that while the returns are symmetrical, the negative returns (losses) can be quite large and frequent, leading to higher risk during downturns. So, the model's performance could be more erratic during the financial crisis, requiring investors to be vigilant and possibly adjust their strategies accordingly. Investors may need to implement robust risk management strategies to mitigate this downside risk. From these figures, it is evident that both double deep Q-learning and XGBoost can effectively curtail the standard deviations and the downside volatility of the investment strategies. From this follows an expectation of superior risk-adjusted performances, such as Sharpe and Sortino ratios, wherein an improved performance is linked to lower volatility, all else equal.

In terms of Sharpe ratio, double deep Q-learning and XGBoost achieve a similar performance of 0.63x, while ARFIMA-GARCH underperforms due to lower returns not sufficiently offset by lower standard deviation. The findings are consistent with the existing studies (Wang et al. 2020). For the ARFIMA-GARCH strategy, the same applies in terms of the Sortino ratio. Double deep Q-learning, instead, is the best-performing model in terms of Sortino ratio (best risk-adjusted returns by focusing on downside risk) on the back of lower downside volatility (meaning it is less likely to experience significant losses, which is crucial for risk-averse investors) and higher returns, followed by the XGBoost strategy which also outperforms the buy-and-hold strategy. This makes the machine learning model an attractive option for investors seeking high returns with controlled risk.

Moving to the performance of the classification models, Table 2 reports statistics of the confusion matrices generated by each classifier. All the confusion matrices are based on the model's ability to correctly classify the sign of the one-step-ahead realised return on the average of the 10 out-of-sample synthetic data sets. The sign is extracted in the same way as presented in the previous section. The realised classes in the average

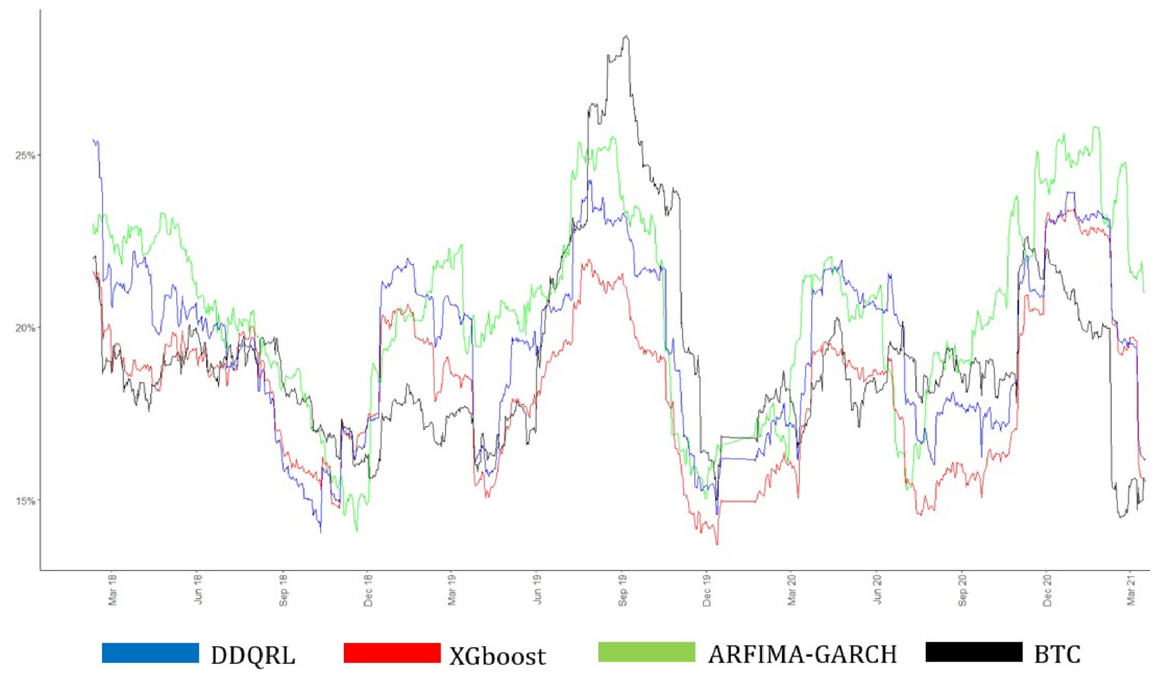


**FIGURE 2** | Rolling standard deviations obtained by each strategy. [Colour figure can be viewed at wileyonlinelibrary.com]

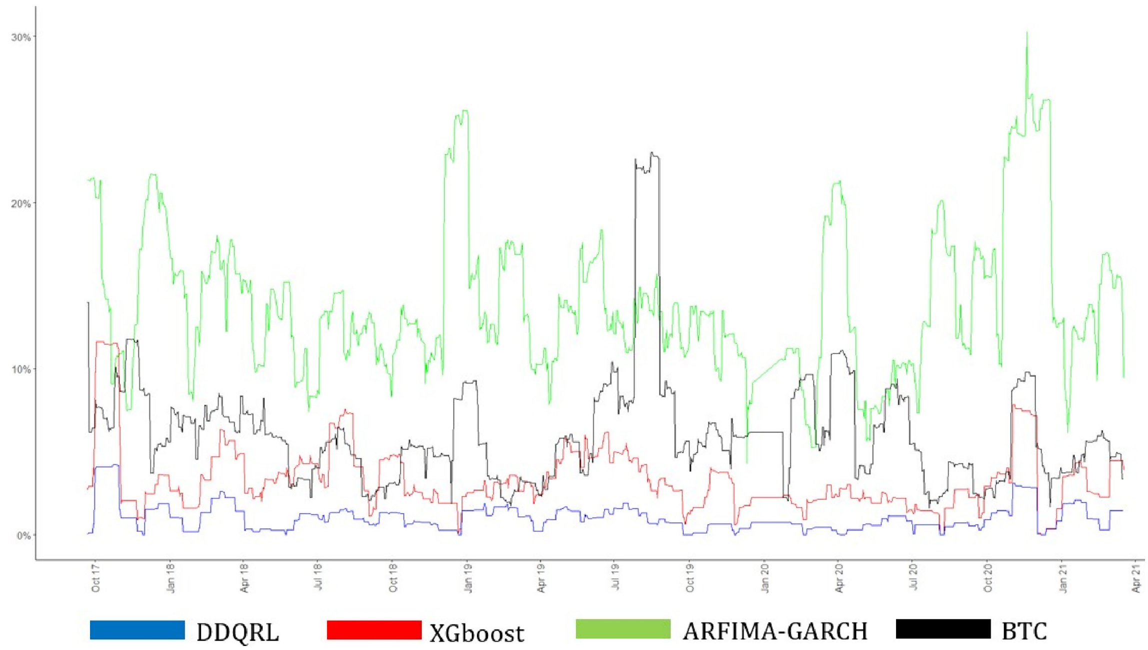


**FIGURE 3** | Rolling downside volatilities obtained by each strategy. [Colour figure can be viewed at wileyonlinelibrary.com]

**TABLE 2** | Statistics of confusion matrices.

| | Double deep Q-learning | XGBoost | ARFIMA-GARCH |
|---|---|---|---|
| Accuracy (%) | 75.27 | 76.35 | 51.93 |
| 95% CI lower (%) | 77.6 | 78.34 | 54.68 |
| 95% CI upper (%) | 72.8 | 73.94 | 49.17 |
| Sensitivity (%) | 74.11 | 74.57 | 52.4 |
| Specificity (%) | 76.42 | 78.1 | 51.45 |
| Balanced accuracy (%) | 75.26 | 76.34 | 51.93 |

*Note:* The statistics of the confusion matrices generated by each classifier. The classifiers are double deep Q-learning; XGBoost and ARFIMA-GARCH. The statistics are reported in percentages.

out-of-sample data sets are well balanced, with 50.3% of the observations belonging to the buying class and the rest to the selling one. Double deep Q-learning and XGBoost outperform ARFIMA-GARCH in terms of all the metrics proposed. The latter, in fact, can barely improve on the performance of a random classifier as it records a 51.9% accuracy and a 95% confidence interval lower bound below the 50% threshold. Double deep Q-learning and XGBoost, instead, achieve fairly similar results. The latter records the highest out-of-sample classification accuracy of 76.35%. Moreover, both double deep Q-learning and XGBoost have larger specificity compared with sensitivity. In other words, both models are better suited to identifying days when selling is the best strategy compared to buying (Filos 2019). ARFIMA-GARCH, on the other hand, records higher sensitivity compared to specificity, yet not far enough from the performance of a random classifier.

## 7 | Conclusions

In this study, we investigate whether a machine learning-inspired model can successfully forecast cryptocurrency returns. We need machine learning models to find easy explanation for investors and policy makers and also to address the limitations of statistical models explained in the existing recent studies (Chen et al. 2021; Wang, Andreeva, and Martin-Barragan 2023). To address the above question, we evaluated the performance of three models, namely double deep Q-learning, XGBoost and ARFIMA-GARCH in forecasting Bitcoin prices as well as to a buy-and-hold strategy. Our results show that the double deep Q-learning model outperforms the other models in terms of returns and Sortino ratio while the ARFIMA-GARCH model represented the worst-performing model across all tests. In layman's terms, by using the above models we identified one of the best models that can consider the risk factor appropriately in forecasting and can be suitable for any investors with any level of risk-taking behaviour. The findings suggest that in practice, an investment strategy will only be penalised for volatility in a down-moving market, which is a great assurance for investors. It is also important for policymakers to know that our findings suggest less damage to economic value of the market during a time of extreme volatility. Based on these results, the study offers three main contributions.

First, our study contributes to the literature on cryptocurrency returns by settling the debates on the suitability of machine learning models in forecasting cryptocurrency returns. In prior finance literature, there are debates about the suitability of machine learning models in forecasting (Sun Yin et al. 2019). While some studies (Chen et al. 2021; Gradojevic et al. 2021) argue that machine learning models can forecast cryptocurrency returns, others (Christodoulou et al. 2019) disagree, leading to inconclusive debates. Moreover, existing research (Xie, Chen, and Hu 2020) tends to use regression approaches or primarily focus on out-of-sample performance evaluation on a single subset of historical data opening the results to more criticisms as to the suitability of machine learning models. However, in this study, we move beyond the existing research by evaluating the suitability of three machine learning models as well as used the entire historical data of Bitcoin to overcome the single historical snapshot criticism. By doing so, the study offers critical insights that address the criticism of existing research as well as attempts to settle the ongoing debate in the literature.

Second, whereas existing research mainly uses the total cumulated profit as target function, only focusing on the time series of returns and out-of-sample backtesting on historical data, our study contributes new insights by targeting a risk-adjusted measure such as the Sortino ratio, which penalises an investment strategy only for volatility in a down-moving market, as opposed to Sharpe ratio which penalises for volatility in any market movement. For instance, prior studies (Li, Zheng, and Zheng 2019) use cumulated profit, which is criticised for not accurately reflecting risk considerations. However, our approach of risk adjusting using the Sortino ratio offers a more holistic representation of risk targeting cumulated profit results in an investment strategy with extreme allocations, without considering the riskiness of the position. The difference becomes particularly relevant when different types of investors are considered. While, on one hand, proprietary traders are focused on maximising the profit and loss function of their investment strategies, investment funds and banks are focused on delivering higher returns amid controlled risk scenarios. For these reasons, these institutions are mostly evaluated against risk-adjusted measures, such as the Sortino ratio. A widespread adoption of cryptocurrencies to foster broad societal consequences also passes through the inclusion of these instruments among the traded instruments of such large institutions.

Lastly, our study enriches the literature on financial asset forecasting by offering an alternative perspective in forecasting Bitcoin returns. By conducting this study on Bitcoin price forecasting, we enrich the investment literature (Mattke et al. 2021; Mai and Hranac 2013; Gefen 2002). The extant literature (Ibrahim, Kashef, and Corrigan 2021) predominantly uses time-series data on cryptocurrency prices, which often do not take other critical peculiarities—such as market microstructure—into consideration. By relying solely on time series data of cryptocurrency price, the results of these studies are sometimes criticised for robustness. Our study overcame this challenge by using a set of explanatory variables (average block size, has rate, transaction volume, transaction per block, transactions excluding popular addresses and number of transactions), in addition to time-series data of cryptocurrency prices. Thus, this approach allows us to take into consideration the peculiar market microstructure of cryptocurrencies. By using explanatory variables to augment limitations in solely relying on time-series data, this study contributes a novel process that advances cryptocurrency returns forecasting research. In addition, Bitcoin time-series exhibit high volatility and leptokurtosis, which, coupled with the short trading history, makes out-of-sample evaluations based on historical data highly unreliable (Pintelas et al. 2020). For this reason, we contribute to the finance literature by proposing the usage of a Variational Autoencoder to simulate the original distribution of the underlying data in 10 synthetic data sets and evaluate the out-of-sample performances on these.

Practically, the study also offers some critical insights. First, the results demonstrate that it is possible to use machine learning models to successfully predict cryptocurrency returns. This means practitioners using and those thinking of using machine learning models can be more confident in applying machine models. It is difficult for the practitioners to use the findings of the prior studies because most of them cannot accurately reflect risk consideration by using cumulated profit (Li, Zheng, and Zheng 2019). However, we use a more holistic approach where the Sortino ratio is relevant for different types of investors. Second, the results of this study offer practitioners a benchmark and reference point for their application of machine learning models since existing research has only provided anecdotal evidence. Lastly, the study offers some strategies to cope with the cryptocurrency volatility. For instance, we define an environment where an agent learns to choose the best suited between two actions, buy or sell a BTC future contract, during each trading day to maximise the Sortino ratio of the investment strategy. We choose to target the Sortino ratio to limit the downside volatility on the strategy since the Bitcoin market is characterised by frequent and pronounced volatility spikes. Creating a consistently profit-making investment algorithm based on online learning would attract more long-term investors and potentially win the regulatory consensus (Sun Yin et al. 2019) for creating regulated spot trading venues. Therefore, this would result in improved long-term market liquidity and lower market volatility. The virtuous cycle would complete with the more widespread adoption of digital coins, fuelling positive societal impact hidden beneath the merely speculative aims. As such, the scope of this manuscript is to propose an online machine learning model, the double deep Q-learning, and analyze its performance both in terms of investment strategy and as a classifier. Like all research, this study has some limitations, which presents an avenue for future studies. First, this research only focused on Bitcoin, therefore future studies can use other cryptocurrencies such as Ethereum, Litecoin, Doge coin and so on; validate our findings towards wide generalisation. Second, our study used three popular machine learning models, namely double deep Q-learning, XGBoost and ARFIMA-GARCH in forecasting Bitcoin prices. Second, future studies can explore further development and refinement of cross-validation methods tailored to financial data, particularly focusing on mitigating overfitting and improving model robustness. It is also important to examine the application of the combinatorial purged cross-validation (CPCV) method to real-world financial markets to validate its effectiveness and practicality. Finally, financial institutions can explore the validation techniques for regulatory compliance.

## Acknowledgements

The authors would like to acknowledge the anonymous reviewers for their constructive feedback to improve the quality of the paper. In addition, authors would also like to thank the Associate Editor and the Editor of the journal for guiding them in revising the paper.


## Conflicts of Interest

The authors declare no conflicts of interest.

## Data Availability Statement

Data sharing is not applicable to this article as no new data were created or analyzed in this study.

## Endnotes

[1] Socio-economic impact of Bitcoin is summarised.

## References


Adhami, S., and D. Guegan. 2020. "Crypto Assets: The Role of Ico Tokens Within a Well-Diversified Portfolio." *Journal of Industrial and Business Economics* 47, no. 2: 219–241.

Almaafi, A., S. Bajaba, and F. Alnori. 2023. "Stock Price Prediction Using ARIMA Versus XGBoost Models: The Case of the Largest Telecommunication Company in the Middle East." *International Journal of Information Technology* 15, no. 4: 1813–1818.

Alonso-Monsalve, S., A. L. Suárez-Cetrulo, A. Cervantes, and D. Quintana. 2020. "Convolution on Neural Networks for High-Frequency Trend Prediction of Cryptocurrency Exchange Rates Using Technical Indicators." *Expert Systems With Applications* 149: 113250.

Al-Yahyaee, K. H., W. Mensi, and S. M. Yoon. 2018. "Efficiency, Multifractality, and the Long-Memory Property of the Bitcoin Market: A Comparative Analysis With Stock, Currency, and Gold Markets." *Finance Research Letters* 27: 228–234.

Amini, S., B. Gebka, R. Hudson, and K. Keasey. 2013. "A Review of the International Literature on the Short-Term Predictability of Stock Prices Conditional on Large Prior Price Changes: Microstructure, Behavioral and Risk Related Explanations." *International Review of Financial Analysis* 26: 1–17.

Amiram, D., B. N. Jørgensen, and D. Rabetti. 2022. "Coins for Bombs: The Predictive Ability of on-Chain Transfers for Terrorist Attacks." *Journal of Accounting Research* 38, no. 1: 483–517.

Anyfantaki, S., S. Arvanitis, and N. Topaloglou. 2021. "Diversification Benefits in the Cryptocurrency Market Under Mild Explosivity." *European Journal of Operational Research* 295: 378–393.

Arian, H. R., M. D. Norouzi, and L. A. Seco. 2024. "Backtest Overfitting in the Machine Learning Era: A Comparison of Out-of-Sample Testing Methods in a Synthetic Controlled Environment." *Available at SSRN*.

Baillie, R. T., C.-F. Chung, and M. A. Tieslau. 1996. "Analysing Inflation by the Fractionally Integrated ARFIMA–GARCH Model." *Journal of Applied Econometrics* 11, no. 1: 23–40.

Bianchetti, M., C. Ricci, and M. Scaringi. 2018. "Are Cryptocurrencies Real Financial Bubbles? Evidence From Quantitative Analyses." *SSRN Electronic Journal* 26.

Bollerslev, T. 1987. "A Conditionally Heteroskedastic Time Series Model for Speculative Prices and Rates of Return." *Review of Economics and Statistics* 69: 542–547.

Borgards, O., and R. L. Czudaj. 2020. "The Prevalence of Price Overreactions in the Cryptocurrency Market." *Journal of International Financial Markets Institutions and Money* 65: 101194.

Borrageiro, G., N. Firoozye, and P. Barucca. 2022. "The Recurrent Reinforcement Learning Crypto Agent." *IEEE Access* 10: 38590–38599. https://arxiv.org/abs/2201.04699.

Camuto, A., M. Willetts, S. Roberts, C. Holmes, and T. Rainforth. 2021. "Towards a Theoretical Understanding of the Robustness of Variational Autoencoders." In *International Conference on Artificial Intelligence and Statistics*, 3565–3573. PMLR.

Carta, S., A. Ferreira, A. S. Podda, D. R. Recupero, and A. Sanna. 2021. "Multi-DQN: An Ensemble of Deep Q-Learning Agents for Stock Market Forecasting." *Expert Systems With Applications* 164: 113820.

Catania, L., and S. Grassi. 2022. "Forecasting Cryptocurrency Volatility." *International Journal of Forecasting* 38, no. 3: 878–894.

Chen, A.-S., M. T. Leung, and H. Daouk. 2003. "Application of Neural Networks to an Emerging Financial Market: Forecasting and Trading the Taiwan Stock Index." *Computers & Operations Research* 30, no. 6: 901–923.

Chen, T., and C. Guestrin. 2016. "XGBoost: A Scalable Tree Boosting System." In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. New York, NY: Association for Computing Machinery.

Chen, W., H. Xu, L. Jia, and Y. Gao. 2021. "Machine Learning Model for Bitcoin Exchange Rate Prediction Using Economic and Technology Determinants." *International Journal of Forecasting* 37, no. 1: 28–43.

Christodoulou, E., J. Ma, G. S. Collins, E. W. Steyerberg, J. Y. Verbakel, and B. Van Calster. 2019. "A Systematic Review Shows no Performance Benefit of Machine Learning Over Logistic Regression for Clinical Prediction Models." *Journal of Clinical Epidemiology* 110: 12–22.

Corbet, S., and P. Katsiampa. 2020. "Asymmetric Mean Reversion of Bitcoin Price Returns." *International Review of Financial Analysis* 71: 101267.

Dempster, M. A., and V. Leemans. 2006. "An Automated FX Trading System Using Adaptive Reinforcement Learning." *Expert Systems With Applications* 30, no. 3: 543–552.

Deng, Y., F. Bao, Y. Kong, Z. Ren, and Q. Dai. 2016. "Deep Direct Reinforcement Learning for Financial Signal Representation and Trading." *IEEE Transactions on Neural Networks and Learning Systems* 28, no. 3: 653–664.

Ding, S., T. Cui, and Y. Zhang. 2022. "Futures Volatility Forecasting Based on Big Data Analytics With Incorporating an Order Imbalance Effect." *International Review of Financial Analysis* 83: 102255.

Dos Santos, R., and J. Aguilar. 2024. "A Synthetic Data Generation System Based on the Variational-Autoencoder Technique and the Linked Data Paradigm." *Progress in Artificial Intelligence* 13: 1–15.

Edet, S. 2017. "Recurrent Neural Networks in Forecasting S&P 500 Index." Available at SSRN 3001046.

Filos, A. 2019. "Reinforcement Learning for Portfolio Management." arXiv preprint arXiv: 1909.09571.

Fischer, T. G. 2018. "Reinforcement Learning in Financial Markets-a Survey, Technical Report." FAU Discussion Papers in Economics.

Fortunato, M., M. G. Azar, B. Piot, et al. 2017. "Noisy Networks for Exploration." arXiv Preprint arXiv: 1706.10295.

Gefen, D. 2002. "Customer Loyalty in e-Commerce." *Journal of the Association for Information Systems* 3, no. 1: 2.

Geweke, J., and S. Porter-Hudak. 2008. "The Estimation and Application of Long Memory Time Series Model." *Journal of Time Series Analysis* 4: 221–238.

Ghorbel, A., and A. Jeribi. 2021. "Investigating the Relationship Between Volatilities of Cryptocurrencies and Other Financial Assets." *Decisions in Economics and Finance* 44: 817–843.

Gianfreda, A., and L. Grossi. 2012. "Forecasting Italian Electricity Zonal Prices With Exogenous Variables." *Energy Economics* 34, no. 6: 2228–2239.

Gort, B. J. D., X. Y. Liu, X. Sun, J. Gao, S. Chen, and C. D. Wang. 2022. "Deep Reinforcement Learning for Cryptocurrency Trading: Practical Approach to Address Backtest Overfitting." arXiv Preprint arXiv: 2209.05559.

Gradojevic, N., D. Kukolj, R. Adcock, and V. Djakovic. 2021. "Forecasting Bitcoin With Technical Analysis: A Not-So-Random Forest?" *International Journal of Forecasting* 39: 1–17.

Gradojevic, N., and I. Tsiakas. 2021. "Volatility Cascades in Cryptocurrency Trading." *Journal of Empirical Finance* 62: 252–265.

Grover, V., R. H. Chiang, T.-P. Liang, and D. Zhang. 2018. "Creating Strategic Business Value From Big Data Analytics: A Research Framework." *Journal of Management Information Systems* 35, no. 2: 388–423.

Hendershott, T., X. Zhang, J. L. Zhao, and Z. Zheng. 2021. "Fintech as a Game Changer: Overview of Research Frontiers." *Information Systems Research* 32, no. 1: 1–17.

Hessel, M., J. Modayil, H. Van Hasselt, et al. 2018. "Rainbow: Combining Improvements in Deep Reinforcement Learning." In *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32. New Orleans, Louisiana: AAAI Press.

Huang, W. 2021. "Sign Prediction and Sign Regression." *Journal of Investment Strategies* 9, no. 3.

Ibrahim, A., R. Kashef, and L. Corrigan. 2021. "Predicting Market Movement Direction for Bitcoin: A Comparison of Time Series Modeling Methods." *Computers and Electrical Engineering* 89: 106905.

Jana, R., I. Ghosh, and D. Das. 2021. "A Differential Evolution-Based Regression Framework for Fore-Casting Bitcoin Price." *Annals of Operations Research* 306, no. 1: 295–320.

Jiang, Z., D. Xu, and J. Liang. 2017. "A Deep Reinforcement Learning Framework for the Financial Portfolio Management Problem." arXiv Preprint arXiv: 1706.10059.

Keilbar, G., and Y. Zhang. 2021. "On Cointegration and Cryptocurrency Dynamics." *Digital Finance* 3: 1–23.

Kingma, D. P., and M. Welling. 2013. "Auto-Encoding Variational Bayes." arXiv Preprint arXiv: 1312.6114.

Kitchens, B., D. Dobolyi, J. Li, and A. Abbasi. 2018. "Advanced Customer Analytics: Strategic Value Through Integration of Relationship-Oriented Big Data." *Journal of Management Information Systems* 35, no. 2: 540–574.

Klein, T., H. P. Thu, and T. Walther. 2018. "Bitcoin Is Not the New Gold–A Comparison of Volatility, Correlation, and Portfolio Performance." *International Review of Financial Analysis* 59: 105–116.

Kraaijeveld, O., and J. De Smedt. 2020. "The Predictive Power of Public Twitter Sentiment for Forecasting Cryptocurrency Prices." *Journal of International Financial Markets Institutions and Money* 65: 101188.

Kumar, S., M. H. Alsamhi, S. Kumar, A. V. Shvetsov, and S. H. Alsamhi. 2024. "Early MTS Forecasting for Dynamic Stock Prediction: A Double Q-Learning Ensemble Approach." *IEEE Access* 12: 69796–69811.

Lee, J. W., J. Park, O. Jangmin, J. Lee, and E. Hong. 2007. "A Multiagent Approach to q-Learning for Daily Stock Trading." *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 37, no. 6: 864–877.

Li, Y., W. Zheng, and Z. Zheng. 2019. "Deep Robust Reinforcement Learning for Practical Algorithmic Trading." *IEEE Access* 7: 108014–108022.

Li, Z., H. Dong, C. Floros, A. Charemis, and P. Failler. 2021. "Re-Examining Bitcoin Volatility: A Caviar-Based Approach." *Emerging Markets Finance and Trade* 58: 1–19.

Liang, C., Y. Zhang, X. Li, and F. Ma. 2020. "Which Predictor Is More Predictive for Bitcoin Volatility? And Why?" *International Journal of Finance and Economics* 27, no. 2: 1947–1961.

Lin, L.-J. 1992. "Self-Improving Reactive Agents Based on Reinforcement Learning, Planning and Teaching." *Machine Learning* 8, no. 3–4: 293–321.

Mai, E., and R. Hranac. 2013. "Twitter Interactions as a Data Source for Transportation Incidents, Technical Report."

Mattke, J., C. Maier, L. Reis, and T. Weitzel. 2021. "Bitcoin Investment: A Mixed Methods Study of Investment Motivations." *European Journal of Information Systems* 30, no. 3: 261–285.

Miller, N., Y. Yang, B. Sun, and G. Zhang. 2019. "Identification of Technical Analysis Patterns With Smoothing Splines for Bitcoin Prices." *Journal of Applied Statistics* 46: 2289–2297.

Mnih, V., K. Kavukcuoglu, D. Silver, et al. 2015. "Human-Level Control Through Deep Reinforcement Learning." *Nature* 518, no. 7540: 529–533.

Moody, J., L. Wu, Y. Liao, and M. Saffell. 1998. "Performance Functions and Reinforcement Learning for Trading Systems and Portfolios." *Journal of Forecasting* 17, no. 5–6: 441–470.

Müller, O., I. Junglas, J. V. Brocke, and S. Debortoli. 2016. "Utilizing Big Data Analytics for Informa- Tion Systems Research: Challenges, Promises and Guidelines." *European Journal of Information Systems* 25, no. 4: 289–302.

Newell, S., and M. Marabelli. 2015. "Strategic Opportunities (And Challenges) of Algorithmic Decision- Making: A Call for Action on the Long-Term Societal Effects of 'Datification." *Journal of Strategic Information Systems* 24, no. 1: 3–14.

Pintelas, E., I. E. Livieris, S. Stavroyiannis, T. Kotsilieris, and P. Pintelas. 2020. "Investigating the Problem of Cryptocurrency Price Prediction: A Deep Learning Approach." In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 99–110. Neos Marmaras, Greece: Springer.

Raimundo Junior, G. D. S., R. B. Palazzi, R. D. S. Tavares, and M. C. Kotze. 2020. "Market Stress and Herding: A New Approach to the Cryptocurrency Market." *Journal of Behavioral Finance* 23, no. 1: 43–57.

Rehman, M. U., N. Asghar, and S. H. Kang. 2020. "Do Islamic Indices Provide Diversification to Bitcoin? A Time-varying Copulas and Value at Risk Application." *Pacific-Basin Finance Journal* 61: 101326.

Sadighian, J. 2020. "Extending Deep Reinforcement Learning Frameworks in Cryptocurrency Market Making." arXiv Preprint arXiv: 2004.06985.

Schaul, T., J. Quan, I. Antonoglou, and D. Silver. 2015. "Prioritized Experience Replay." arXiv Preprint arXiv: 1511.05952.

Schnaubelt, M. 2022. "Deep Reinforcement Learning for the Optimal Placement of Cryptocurrency Limit Orders." *European Journal of Operational Research* 296, no. 3: 993–1006.

Sebastião, H., and P. Godinho. 2021. "Forecasting and Trading Cryptocurrencies With Machine Learning Under Changing Market Conditions." *Financial Innovation* 7: 1–30.

Silver, D., T. Hubert, J. Schrittwieser, et al. 2017. "Mastering Chess and Shogi by Self-Play With a General Reinforcement Learning Algorithm." arXiv Preprint arXiv: 1712.01815.

Statista. 2021. "Number of Blockchain Wallet Users Worldwide From November 2011 to September 13, 2021." https://www.statista.com/statistics/647374/worldwide-blockchain-wallet-users/.

Sun Yin, H. H., K. Langenheldt, M. Harlev, R. R. Mukkamala, and R. Vatrapu. 2019. "Regulating Cryptocurrencies: A Supervised Machine

Learning Approach to de-Anonymizing the Bitcoin Blockchain." *Journal of Management Information Systems* 36, no. 1: 37–73.

Sutton, R. S., and A. G. Barto. 1998. *Reinforcement Learning: An Introduction*, 22447. Cambridge, MA: MIT Press.

Tofangchi, S., A. Hanelt, D. Marz, and L. M. Kolbe. 2021. "Handling the Efficiency–Personalization Trade-Off in Service Robotics: A Machine-Learning Approach." *Journal of Management Information Systems* 38, no. 1: 246–276.

Van Hasselt, H., A. Guez, and D. Silver. 2016. "Deep Reinforcement Learning With Double Q-Learning." *Proceedings of the AAAI Conference on Artificial Intelligence* 30: 10295.

Van Moffaert, K., and A. Now'e. 2014. "Multi-Objective Reinforcement Learning Using Sets of Pareto Dominating Policies." *Journal of Machine Learning Research* 15, no. 1: 3483–3512.

Wang, L., F. Ma, J. Liu, and L. Yang. 2020. "Forecasting Stock Price Volatility: New Evidence From the GARCH-MIDAS Model." *International Journal of Forecasting* 36: 684–694.

Wang, Y., G. Andreeva, and B. Martin-Barragan. 2023. "Machine Learning Approaches to Forecasting Cryptocurrency Volatility: Considering Internal and External Determinants." *International Review of Financial Analysis* 90: 102914.

Wang, Z., T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, and N. Freitas. 2016. "Dueling Network Architectures for Deep Reinforcement Learning, in 'International Conference on Machine Learning'." *Proceedings of Machine Learning Research*: 1995–2003.

Watkins, C. J. C. H. 1989. "Learning From Delayed Rewards."

Wu, X., H. Chen, J. Wang, L. Troiano, V. Loia, and H. Fujita. 2020. "Adaptive Stock Trading Strategies With Deep Reinforcement Learning Methods." *Information Sciences* 538: 142–158.

Xie, P., H. Chen, and Y. J. Hu. 2020. "Signal or Noise in Social Media Discussions: The Role of Network Cohesion in Predicting the Bitcoin Market." *Journal of Management Information Systems* 37, no. 4: 933–956.

Xiong, Z., X.-Y. Liu, S. Zhong, H. Yang, and A. Walid. 2018. "Practical Deep Reinforcement Learning Approach for Stock Trading." arXiv Preprint arXiv: 1811.07522.

Yang, H., X.-Y. Liu, S. Zhong, and A. Walid. 2020. "Deep reinforcement learning for automated stock trading: An ensemble strategy." In *Proceedings of the First ACM International Conference on AI in Finance*, 1–8. New York, NY: Association for Computing Machinery.

Yaya, O. S., A. E. Ogbonna, R. Mudida, and N. Abu. 2021. "Market Efficiency and Volatility Persistence of Cryptocurrency During Pre- and Post-Crash Periods of Bitcoin: Evidence Based on Fractional Integration." *International Journal of Finance and Economics* 26, no. 1: 1318–1335.

Zhang, J., and D. Maringer. 2016. "Using a Genetic Algorithm to Improve Recurrent Reinforcement Learning for Equity Trading." *Computational Economics* 47, no. 4: 551–567.

Zhang, Z., S. Zohren, and S. Roberts. 2020. "Deep Reinforcement Learning for Trading." *Journal of Financial Data Science* 2, no. 2: 25–40.

## Appendix A

**TABLE A1** | Comparative table for existing literature.

| Articles | Reinforcement learning model | Application domain | Limitations |
|---|---|---|---|
| Wu et al. (2020) | Gated deep Q-learning | Equity single stock | Out-of-sample performance evaluated on a single set of historical data. |
| Borrageiro, Firoozye, and Barucca (2022) | Recurrent reinforcement learning | Bitcoin versus US Dollars; trading perpetual swap derivatives contract | Monte Carlo simulation of 250 trials, obtained reasonable variability of returns. No out-of-sample data analysis. |
| Zhang and Maringer (2016) | Genetic algorithm-recurrent reinforcement learning (GA-RRL) | Daily prices, trading volume, price-earning, price-cash flow, debt-market value of S&P 500 US firms | GA-RRL trading system did not outperform the buy-and-hold strategy by producing a greater number of positive Sharpe ratio. |
| Li, Zheng, and Zheng (2019) | Deep reinforcement learning | Equity single stock | Out-of-sample performance evaluated on a single set of historical data. Low volatility time series removed. Maximises cumulated profit. Cumulates large positions (n contracts). No explanatory variables (Time-dependency) |
| Zhang, Zohren, and Roberts (2020) | Deep Q reinforcement learning | Multi assets future contracts | Out-of-sample performance evaluated on a single set of historical data. Maximises cumulated profit. Cumulates large positions (n contracts). No explanatory variables (Time-dependency) |
| Deng et al. (2016) | Fuzzy deep direct reinforcement | Equity indexes and commodity futures | Out-of-sample performance evaluated on a single set of historical data. Maximises cumulated profit. Cumulates large positions (n contracts). No explanatory variables (Time-dependency) |
| Moody et al. (1998) | Recurrent reinforcement learning | Equity index | Out-of-sample performance evaluated on a single set of historical data. |
| Yang et al. (2020) | Deep Q-reinforcement learning | Equity Index | Out-of-sample performance evaluated on a single set of historical data. Maximises cumulated profit. |
| Lee et al. (2007) | Inverse reinforcement learning | Bitcoin | Out-of-sample performance evaluated ed. on multiple sets of historical data. Maximises cumulated profit. |
| Adhami and Guegan (2020) | DCC, ADCC | Multi assets, cryptocurrencies (bitcoin, tokens) | The evolution of the economic impact of ICOs on the real economy and financial stability is still to be tested. No use of machine learning. |
| Schnaubelt (2022) | Backward-induction Q-learning, deep double Q-networks | Multiple assets | Used out-of-sample performance of reinforcement learning algorithms and benchmark strategies. No implementation of feature representations from the data using CNN and routing orders to multiple exchanges |
| Alonso-Monsalve et al. (2020) | Compares the performance of four different network architectures | Multiple cryptocurrency forecasting | For dash and ripple because of noise and temporal behaviour. The data generation parameters are not sufficient. Short-term trend prediction has its own limitations with network architecture. |
| Dempster and Leemans (2006) | Adaptive reinforcement learning. The parameters are dynamically optimised to maximise a trader's utility. Adaptive reinforce-utility | Historical data on foreign exchange markets | Risk management layer can be extended to control several automated FX trading systems that trade different currencies. Out-of-sample cumulative profit measured. |

*Note:* Summary of the most relevant literature related to our study.

## Appendix B

Synthetic distributions (obtained via VAE) versus original ones (Figures B1–B7).

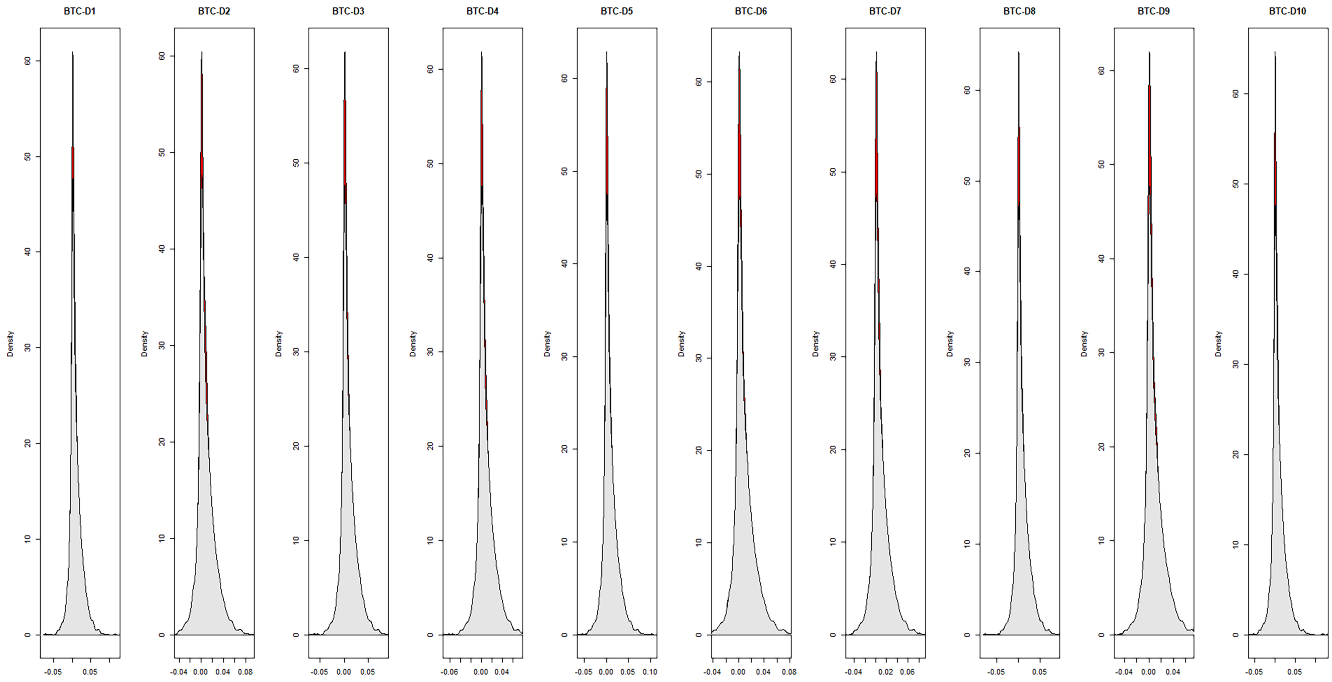


**FIGURE B1** | BTC—Bitcoin price. [Colour figure can be viewed at wileyonlinelibrary.com]

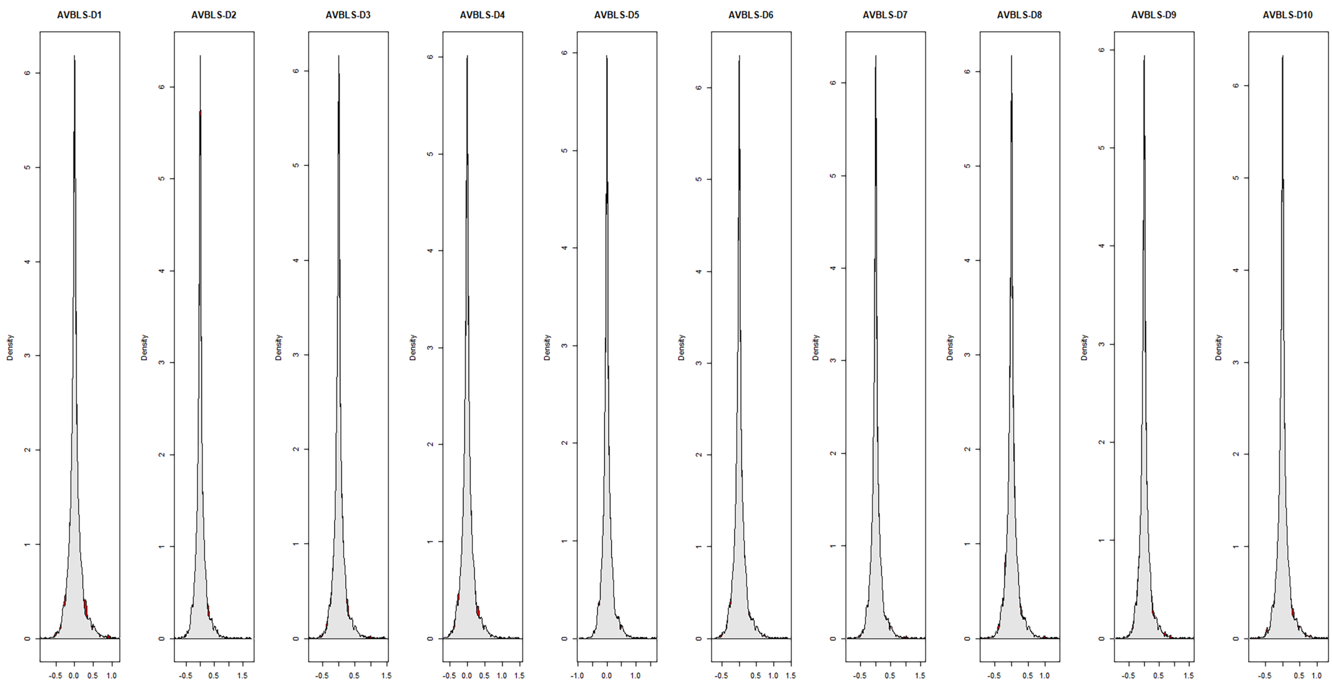


**FIGURE B2** | AVBLS—Average block size. [Colour figure can be viewed at wileyonlinelibrary.com]

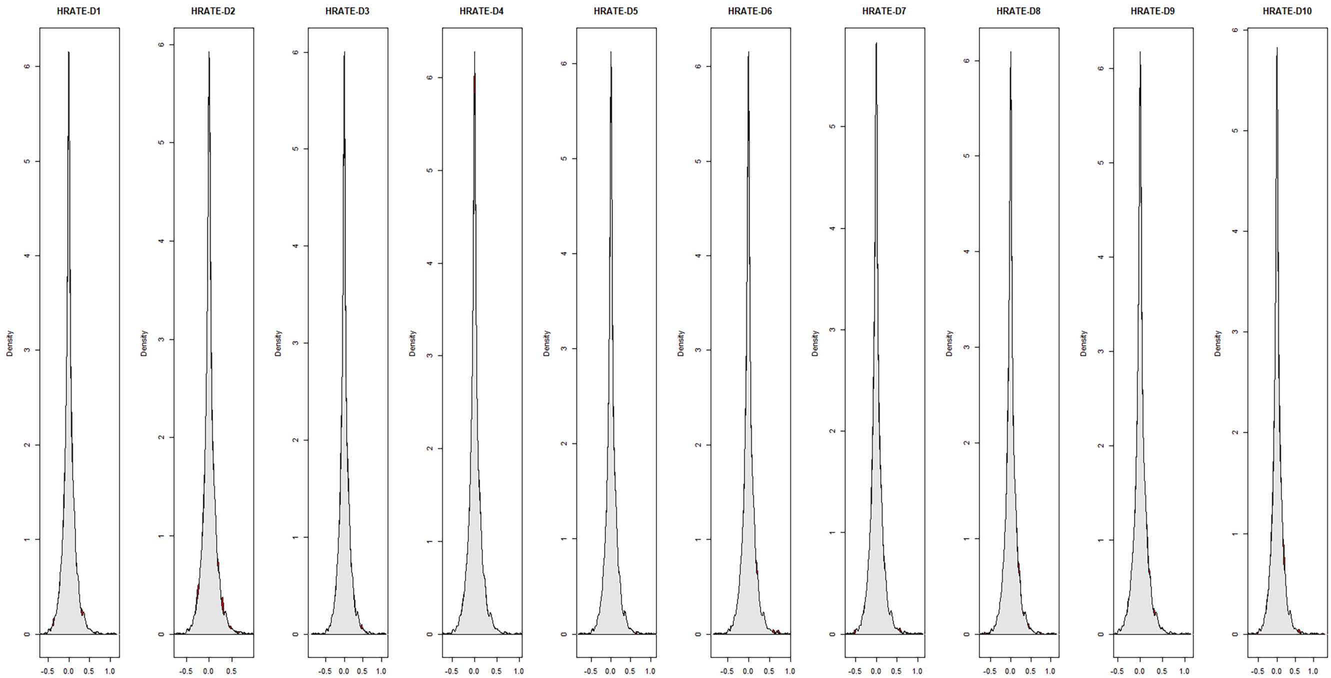


**FIGURE B3** | HRATE—Hash rate. [Colour figure can be viewed at wileyonlinelibrary.com]

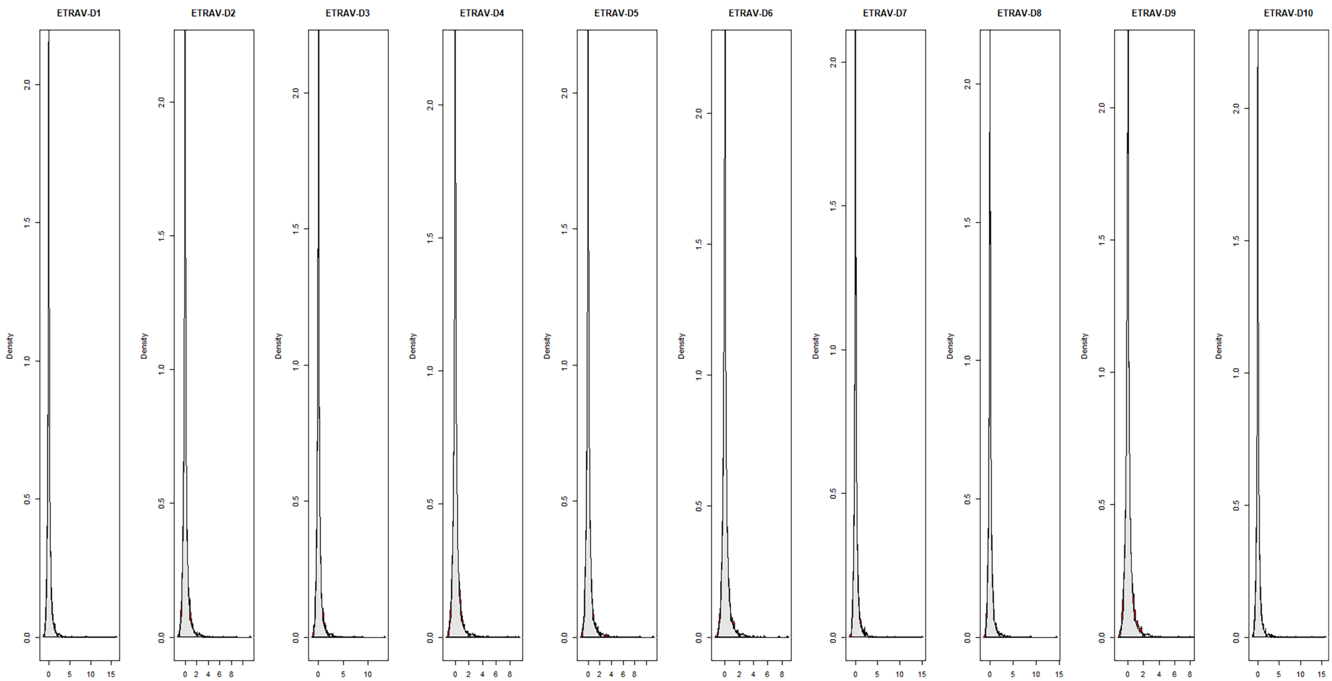


**FIGURE B4** | ETRAV—Estimated transaction volume. [Colour figure can be viewed at wileyonlinelibrary.com]

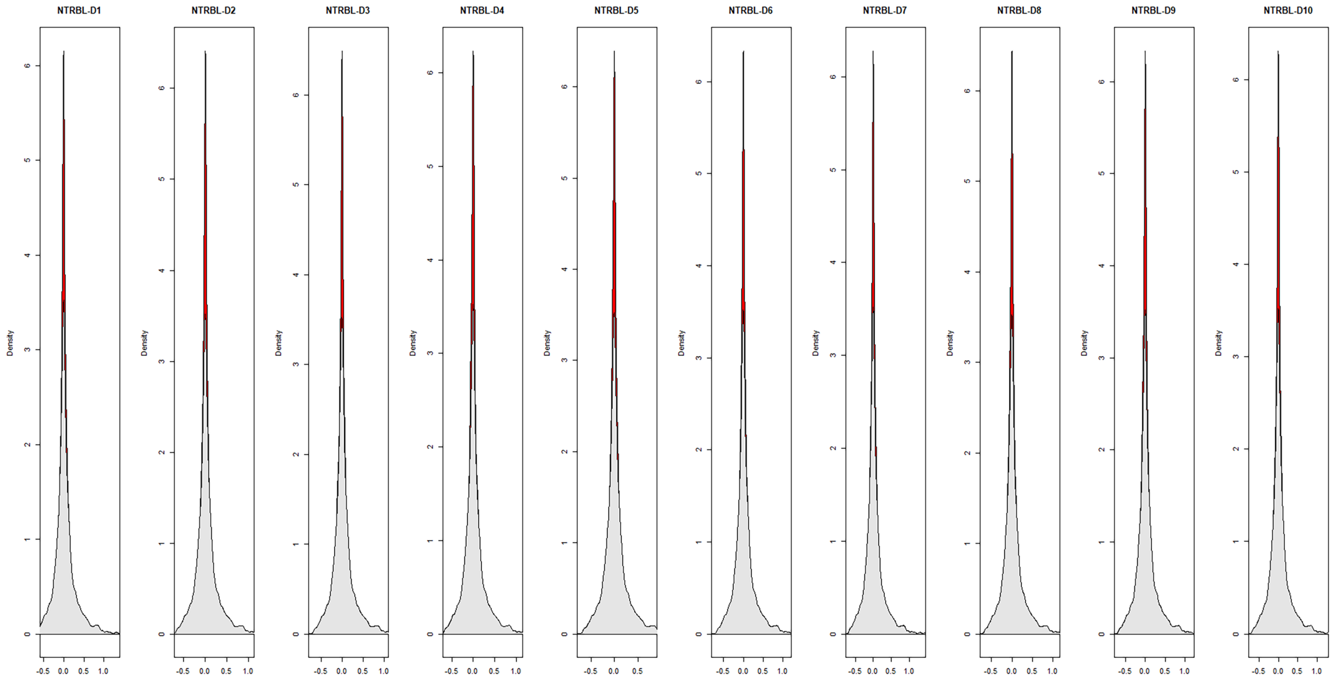


**FIGURE B5** | NTRBL—Transaction per block. [Colour figure can be viewed at wileyonlinelibrary.com]

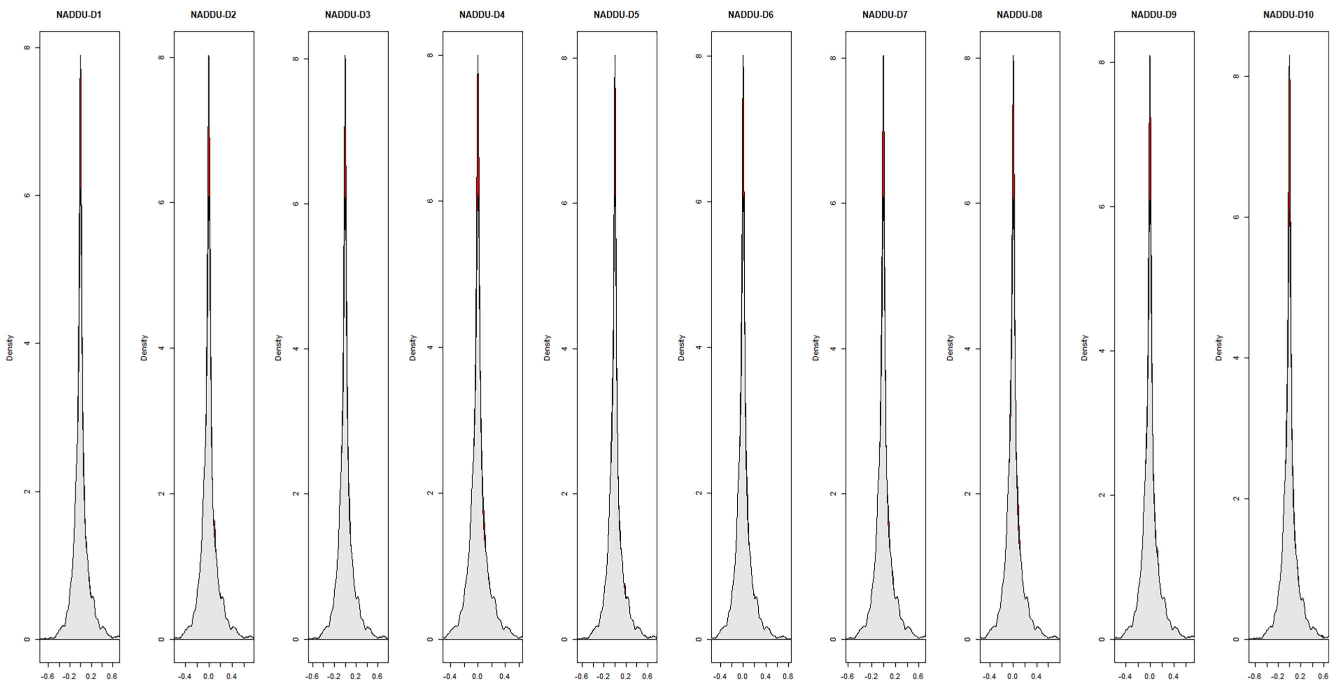


**FIGURE B6** | NADDU—Transactions excluding popular addresses. [Colour figure can be viewed at wileyonlinelibrary.com]

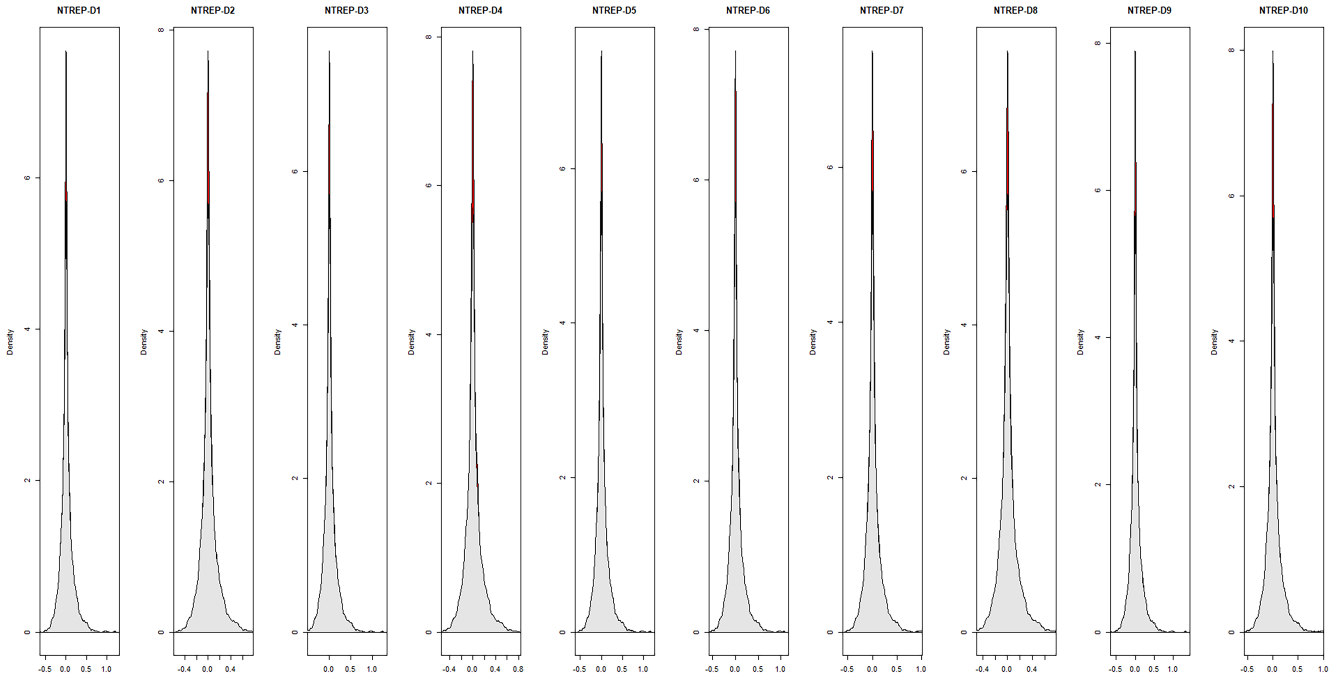


**FIGURE B7** | NTREP number of transactions. [Colour figure can be viewed at wileyonlinelibrary.com]

## Appendix C

Annualised descriptive statistics for investment strategies.

**TABLE C1** | Annualised descriptive statistics for double deep Q-learning investment strategy.

| Sample | Mean | Median | SD | Skewness | Kurtosis | Volatility skew | Downside volatility | Sharpe ratio | Sortino ratio |
|---|---|---|---|---|---|---|---|---|---|
| V1 | 15.629% | 8.214% | 33.005% | 1.990 | 6.023 | 60.054 | 3.011% | 0.592 | 4.190 |
| V2 | 15.385% | 7.950% | 33.718% | 2.194 | 6.150 | 61.466 | 3.258% | 0.589 | 4.333 |
| V3 | 15.811% | 7.746% | 31.953% | 1.844 | 5.455 | 64.304 | 2.735% | 0.649 | 4.258 |
| V4 | 14.885% | 7.646% | 32.691% | 2.493 | 6.517 | 68.208 | 3.085% | 0.655 | 4.412 |
| V5 | 15.762% | 6.941% | 34.191% | 2.310 | 5.753 | 71.336 | 3.213% | 0.636 | 4.422 |
| V6 | 15.666% | 8.247% | 33.215% | 1.931 | 5.866 | 66.233 | 2.698% | 0.627 | 4.097 |
| V7 | 15.915% | 8.146% | 33.869% | 2.583 | 6.273 | 66.259 | 3.104% | 0.603 | 4.386 |
| V8 | 15.865% | 7.123% | 33.040% | 2.321 | 6.483 | 67.120 | 2.948% | 0.611 | 4.380 |
| V9 | 15.756% | 7.016% | 33.817% | 1.762 | 5.636 | 71.153 | 2.739% | 0.618 | 4.180 |
| V10 | 15.396% | 7.246% | 33.684% | 2.358 | 6.912 | 69.235 | 3.232% | 0.598 | 4.137 |
| Average | 15.607% | 7.627% | 33.318% | 2.179 | 6.107 | 66.537 | 3.002% | 0.618 | 4.279 |

*Note:* In this table, we report the annualised descriptive statistics for double deep Q-learning investment strategy in each of the 10 out-of-sample synthetic data sets. V1–V10 denote each of the out-of-sample synthetic data sets. In the last row of the table, we included the average of the 10 data sets.

**TABLE C2** | Annualised descriptive statistics for XGBoost investment strategy.

| Sample | Mean | Median | SD | Skewness | Kurtosis | Volatility skew | Downside volatility | Sharpe ratio | Sortino ratio |
|---|---|---|---|---|---|---|---|---|---|
| V1 | 16.169% | 7.398% | 35.145% | 2.167 | 6.239 | 72.821 | 2.938% | 0.598 | 4.657 |
| V2 | 16.086% | 7.523% | 32.517% | 1.761 | 5.691 | 64.059 | 3.240% | 0.591 | 4.475 |
| V3 | 16.419% | 6.651% | 33.157% | 2.129 | 6.938 | 67.732 | 2.740% | 0.675 | 4.630 |
| V4 | 16.439% | 7.066% | 32.237% | 2.275 | 7.721 | 69.766 | 3.341% | 0.621 | 4.436 |
| V5 | 15.751% | 7.879% | 34.236% | 2.055 | 5.461 | 64.796 | 3.121% | 0.604 | 4.301 |
| V6 | 15.750% | 7.197% | 34.442% | 2.738 | 6.406 | 71.401 | 3.196% | 0.620 | 4.606 |
| V7 | 15.946% | 6.335% | 33.251% | 2.448 | 5.630 | 65.679 | 2.771% | 0.618 | 4.355 |
| V8 | 15.624% | 7.871% | 31.230% | 2.180 | 5.977 | 69.754 | 3.199% | 0.645 | 5.277 |
| V9 | 15.551% | 6.578% | 31.988% | 2.286 | 6.716 | 56.727 | 3.325% | 0.660 | 4.693 |
| V10 | 16.288% | 7.903% | 31.758% | 2.391 | 7.046 | 68.861 | 2.790% | 0.651 | 4.507 |
| Average | 16.002% | 7.240% | 32.996% | 2.243 | 6.382 | 67.160 | 3.066% | 0.628 | 4.594 |

*Note:* In this table, we report the annualised descriptive statistics for XGBoost investment strategy in each of the 10 out-of-sample synthetic data sets. V1–V10 denote each of the out-of-sample synthetic data sets. In the last row of the table, we included the average of the 10 data sets.

**TABLE C3** | Annualised descriptive statistics for ARFIMA-GARCH investment strategy.

| | Mean | Median | SD | Skewness | Kurtosis | Volatility skew | Downside volatility | Sharpe ratio | Sortino ratio |
|---|---|---|---|---|---|---|---|---|---|
| V1 | 3.49% | 1.85% | 17.38% | 0.27 | 3.08 | 1.16 | 7.44% | 0.03 | 0.04 |
| V2 | 5.59% | 4.02% | 26.34% | −0.04 | 2.13 | 1.07 | 11.54% | 0.03 | 0.04 |
| V3 | 4.92% | 3.07% | 23.21% | −0.08 | 1.93 | 1.09 | 10.12% | 0.03 | 0.04 |
| V4 | 6.27% | 11.28% | 24.31% | −0.18 | 1.79 | 1.03 | 10.74% | 0.03 | 0.05 |
| V5 | 4.83% | 0.21% | 21.49% | 0.22 | 1.81 | 1.21 | 9.11% | 0.03 | 0.04 |
| V6 | 15.06% | 14.01% | 30.66% | −0.02 | 1.32 | 1.22 | 13% | 0.06 | 0.1 |
| V7 | 2.82% | 6.75% | 25.13% | −0.13 | 1.89 | 0.98 | 11.23% | 0.01 | 0.02 |
| V8 | 1.79% | 3.59% | 19.89% | −0.14 | 2.08 | 0.98 | 8.89% | 0.01 | 0.02 |
| V9 | 5.56% | 1.86% | 29.51% | 0.05 | 1.11 | 1.12 | 12.77% | 0.02 | 0.04 |
| V10 | 4.83% | 9.29% | 22.60% | −0.22 | 3.92 | 0.98 | 10.11% | 0.03 | 0.04 |
| Average | 5.52% | 5.59% | 24.05% | −0.03 | 2.11 | 1.08 | 10.50% | 0.03 | 0.04 |

*Note:* The descriptive statistics of the returns as well as the Sharpe and Sortino ratios achieved by ARFIMA-GARCH investment strategy in each of the 10 out-of-sample synthetic data sets. To improve comparability, we also include the average across the 10 data sets. V1–V10 denotes the out-of-sample synthetic data sets.

**TABLE C4** | Annualised descriptive statistics for Bitcoin buy-and-hold investment strategy.

| | Mean | Median | SD | Skewness | Kurtosis | Volatility skew | Downside volatility | Sharpe ratio | Sortino ratio |
|---|---|---|---|---|---|---|---|---|---|
| BTC | 7.96% | 4.79% | 20.70% | 1.21 | 6.73 | 8.71 | 4.69% | 20.2 | 1.41 |

*Note:* The descriptive statistics of the returns as well as the Sharpe and Sortino ratios achieved by Bitcoin buy and hold investment strategy.