Digital Finance (2025) 7:107–131
https://doi.org/10.1007/s42521-024-00123-2

RESEARCH

Regime switching forecasting for cryptocurrencies

Ilyas Agakishiev[1] · Wolfgang Karl Härdle[1] · Denis Becker[2] · Xiaorui Zuo[3]

Received: 10 October 2024 / Accepted: 29 December 2024 / Published online: 29 January 2025

Abstract
There are many ways to model complex time series. The simplest approach is to increase the complexity, and thus, the flexibility of the model, for the entire time series. As an example, one could use a neural network. Another solution would be to change the parameters of a model dependent on the "state" or "regime" of the time series. A typical example here would be the Hidden Markov model (HMM). This paper combines the two concepts to create a Reinforcement Learning (RL) model that adds variables that depend on the state of the time series. To test the concept, the RL model is used with cryptocurrency data to determine the share to invest into the cryptocurrency index CRIX in order to maximize wealth. The results have shown that cryptocurrency metadata is useful as supplementary data for analysis of the respective prices. The Reinforcement learning model with regimes shows potential for investment management, but comes with some caveats.

Keywords Regime switching · Machine learning · Crypto currencies · Reinforcement learning · FinTech

JEL Classification C14 · C15 · C87 · C63

1 Introduction

Asset returns behavior tends to change ever so often, making returns difficult to predict and portfolio management less consistent. This is especially true, if the asset class in question is based on cryptocurrencies. More complex models seem to have more success in the prediction process than the straightforward linear models, but

This research was supported by the Deutsche Forschungsgemeinschaft's IRTG 1792 grant, "IDA Institute of Digital Assets", contract number CF166/15.11.2022, and the project "AI4EFin AI for Energy Finance", contract number CF162/15.11.2022, financed under the Romania's National Recovery and Resilience Plan, Apel nr. PNRR-III-C9-2022-I8., COST Actions "Fintech and Artificial Intelligence in Finance" (FinAI-CA19130) and "Text, functional and other high-dimensional data in econometrics: New models, methods, applications" (HiTEc-CA21163), and the Yushan Scholar Program of Taiwan.

Extended author information available on the last page of the article

they are difficult to interpret, and thus may not be replicable in a different dataset or environment.

In portfolio management, it is common to adjust the allocation weights based on the market volatility. For example, for equity, growth stocks are likely more popular in low volatility environments, while value stocks are preferential in the case of high volatility.

In this paper, a regime-switching reinforcement learning model, is applied to portfolio management of cryptocurrencies. The three regimes are defined based on volatility and return quantiles. While in many asset classes, volatility and return have a high negative correlation, this is, at this point, not the case with cryptocurrencies Karim et al. (2023). Reinforcement learning models are, in theory, capable of adapting to such a changing environment, however, splitting up the task into two stages (one to determine the regime and the other to set the portfolio weights) can prove to be more effective.

The proposed model is comprised of two steps. The first one determines the regime for the next time period. This can be done using a simple ARMA-GARCH model, or a more complex LSTM model. One advantage of the LSTM is that the importance (weight) of the correct prediction of a regime switch can be easily regulated with a hyperparameter. The output is a probability vector for each of the regimes. The data used for this step are the Bitcoin (BTC) metadata provided by the Blockchain Research Center (BRC).

The second step trains three Reinforcement Learning models to determine the portfolio weights. One of them will have regime probabilities (the raw output of the previous step), the second one will only include the predicted regime (the regime with the highest probability), as well as a reference one with no regimes. This step uses the hourly data for the CRIX cryptocurrency index, also provided by the BRC.

The remainder of the paper will be structured as follows. Section 2 gives a literature overview. Then, Sect. 3 describes the methodology for both steps of the model. Section 4 shows the model configuration in practice for BTC analysis and CRIX trading. For the first step, it is evaluated, how the weighting changes the regime prediction, and how well both regimes in general as well as regime changes are predicted. Then, LIME is used to determine, which of the variables had the strongest influence on the decision making. For the second step, the trading of CRIX with the three different methods is evaluated and compared. Afterwards, the results are discussed in more detail and a conclusion is drawn. All the code used is available on the Quantlet[1] website and a Courselet for this paper can be found on Quantinar.[2]

[1] https://quantlet.com.

[2] https://quantinar.com.

2 Literature overview

Traditional statistical methods, like ARIMA (Autoregressive Integrated Moving Average) (Box et al., 2015), have long dominated time series forecasting. However, they often struggle with non-linear relationships and complex dependencies within the data. Neural networks, on the other hand, excel at capturing these non-linearities and, in principle, can learn and fit intricate patterns from large data volumes.

The most common neural network architectures for time series are the Fully Recurrent Neural Network (FRNN), Gated Recurrent Unit (GRU) (Cho et al., 2014) and Long Short Term Memory (LSTM) (Hochreiter & Schmidhuber, 1997). These networks effectively handle sequential data by incorporating connections that allow them to remember past information and utilize it for future predictions.

Another architecture that is often used with time series is the Convolutional Neural Network (CNN) (Lecun et al., 1998). While primarily used for image data, CNNs have been adapted for time series by treating the data as a one-dimensional "image". This allows them to capture local patterns and temporal dependencies within the data. Other techniques include Time Delay Neural Networks (Waibel et al., 1989) and Attention Mechanisms (Vaswani et al., 2017).

Reinforcement learning is a framework designed to learn the optimal actions over a period with the goal to maximize the reward over that period. It is often used in tandem with neural networks, as fully modeling Q-tables is often not an option. Q-learning is the simplest method, and it has many variations, such as Deep Q-Learning (DQN), Double DQN (Hasselt et al., 2016) and Dueling DQN (Wang et al., 2016). Common actor-critic algorithms are Deep Deterministic Policy Gradient (DDPG) (Lillicrap et al., 2016), Advantage Actor Critic (A2C) (Mnih et al., 2016) and Proximal Policy Optimization (PPO) (Schulman et al., 2017).

Reinforcement Learning has found many applications in economics and finance (Charpentier et al., 2021; Mosavi et al., 2020). Several studies have used reinforcement leaning for learning optimal trading decisions with cryptocurrencies like Bitcoin, Ethereum and Litecoin (Lucarelli & Borrotti, 2020; Li et al., 2019; Sattarov et al., 2020; Liu et al., 2021). The decision space in these studies was restricted to discrete decisions of the form "buy", "sell" and "hold". Different algorithms and policies have been applied like PPO with MLP and LSTM policies (Liu et al., 2021), Double and Dueling Double Deep Q-Learning Networks, and actor-critic algorithms (Li et al., 2019). In addition to average or cumulative profits some studies suggested a Sharpe-ratio-based performance criterion.

All of the mentioned studies have shown that RL-based trading recommendations outperform various benchmark strategies based on different moving-average crossovers (golden cross, death cross or double cross), momentum, or simple buy and hold.

Regime switching is a concept similar to the Hidden Markov Model (Rabiner & Juang, 1986). Is has been applied to reinforcement learning (Maringer & Ramtohul, 2012), albeit with a simple AR-model.

Table 1 Coin features extracted from Blockchain Research Center

Feature	Format
active_addresses_per_day	Percentage change
avg_block_size_per_day	Percentage change
avg_block_time_per_day	Value
avg_fee_to_reward	Value
avg_hashrate_per_day	Percentage change
avg_mining_difficulty_per_day	Percentage change
avg_transaction_fee_per_day	Value
avg_transaction_value_per_day	Value
market_capitalization	Percentage change
mining_profitability	Value
num_transactions_per_day	Value
num_unique_addresses_per_day	Value
sent_coins	Value
30-day volatility (derived from returns)	Value
Maximum drawdown (derived from returns)	Value

3 Methodology

3.1 Data overview and preprocessing

The CRIX was initially developed by Härdle and Trimborn (2015) and further studied by Trimborn and Härdle (2018), Kim et al. (2021) and Hou et al. (2019). At the time of writing, the CRIX is provided by Royalton Partners.

The dominant cryptocurrency of the CRIX is Bitcoin (BTC), and the other coins in the index are highly positively correlated with BTC (Candila, 2021), so, to simplify the calculations, only BTC metadata is used. The dataset ranges from 07.09.2015 to 31.05.2022 and is recorded daily. The data are extracted from the Blockchain Research Center[3] (BRC) and is listed in Table 1.

The descriptive statistics for the raw data can be found in the Appendix A.

At the preprocessing stage, for all variables (except volatility and drawdown), a 30-day moving average is taken to reduce noise. In addition, missing values are filled using the value of the previous time step.

3.2 Regime definition

Next, the regimes are defined. There are three regimes: Low volatility, high volatility and distress. The regimes thresholds seem to be arbitrary, but clearly distinguish between common scenarios in the market. During the "Low volatility" regime, the price movements are relatively slow and predictable. The "High volatility" regime poses higher risk for investors, but also offers the highest opportunities for gains, while

[3] https://blockchain-research-center.com.

Table 2 Regime criteria

Regime	Criteria
Low volatility	$p_\sigma < 0.3$
Buffer zone 1	$0.3 < p_\sigma \leq 0.4$
High volatility	$0.4 < p_\sigma \leq 0.6$
Buffer zone 2	$0.6 < p_\sigma \leq 0.7$ or $p_d \leq 0.85$
Distress	$p_d > 0.85$

the "Distress" regime reflects the sudden downwards movements. In order to avoid constant switching between the regimes, two "buffer zones" are established. If the volatility level is in a "buffer zone", the regime will not change, until the next regime is properly reached.

The distress regime includes return quantiles p_r as a additional criterion. It was necessary to introduce it, as, unlike most assets, cryptocurrencies do not have a significantly negative skewness. In fact, the skewness of BTC in during the combined training and test sets is almost zero (−0.0362). Thus, very high volatility in cryptocurrencies have a similar probability to result in positive returns as negative returns (Table 2).

Overall, distress is determined using a distress score p_d, which is a combination of the volatility (p_σ) and return quantiles (p_r):

$$p_d = \frac{p_\sigma + (1 - p_r)}{2} \tag{1}$$

Figure 1 shows the results.

3.3 Weighting system

The prediction of the regime poses two problems which need to be tackled:

- The resulting regime data is imbalanced
- The importance of regime switches should be increased, as it would otherwise be optimal to always predict the previous regime for the next one

With time series data, techniques such as oversampling or undersampling are not feasible, so instead, weights for each data point are introduced to solve both problems.

Denote $g \in G, G \overset{\text{def}}{=} \{1, 2, 3\}$ as a regime in the regime space, and g_t is the regime at time t. Then, in order to account for the imbalance, the weight w_t is introduced when calculating the loss function:

$$w_t = (1 - \lambda)\frac{T}{\sum_{s=1}^{T} I\{g_t = g_s\}} + \lambda \frac{\sum_{s=1}^{T} I\{g_s \neq g_{s-1}\}}{\sum_{s=1}^{T} I\{g_t = g_s\} I\{g_s \neq g_{s-1}\}} \tag{2}$$

where T is the total length of the time series in the training set and is a hyperparameter. Here, the first term corrects for the imbalance in the data and the second adds

Fig. 1 BTC prices. Regime breakdown: low volatility, buffer zone 1, high volatility, buffer

more weight to data points, where the regime switches. $\lambda \in [0, 1]$ determines, how important regime-switching data points relative to other data points are.

3.4 Regime forecasting

The regimes are approximated using a neural network with a LSTM layer, followed by two dense layers, as shown in the following figure (Fig. 2):

The LSTM allows to preserve information from past data points while effectively forgetting the irrelevant information.

The parameters are shown in the following table (Table 3):

The loss function is categorical cross-entropy with the weights defined in Eq. (2):

$$l = -\sum_{t=1}^{T} w_t y_t \log \hat{y}_t \tag{3}$$

In order to evaluate variable importance, the LIME algorithm is applied (Ribeiro et al. 2016). The objective of the algorithm is to ensure both interpretability and local fidelity. This is achieved by solving the minimization problem:

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{4}$$

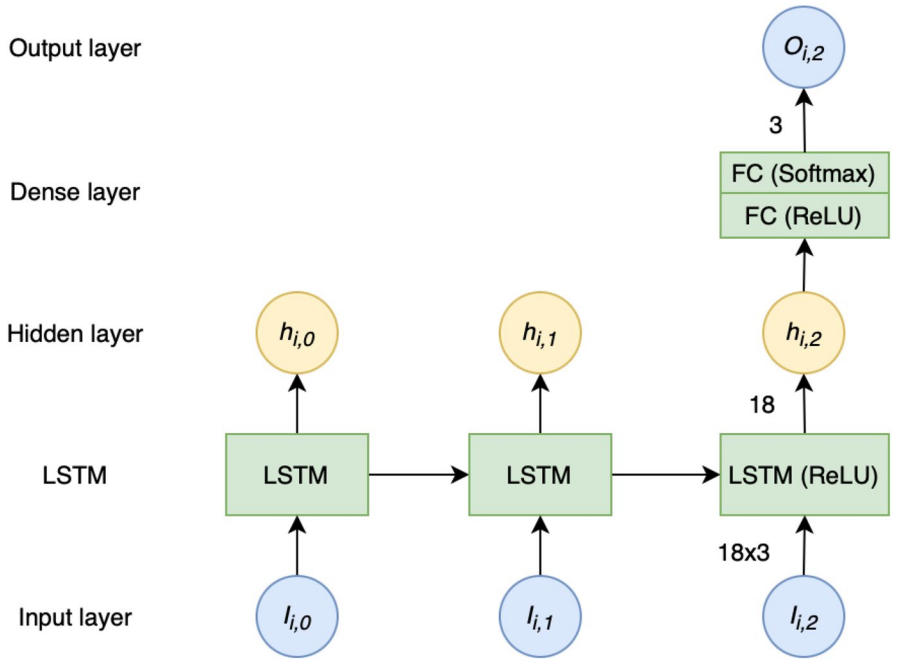


Fig. 2 LSTM architecture for regime forecasting

Table 3 LSTM parameters

Parameter	Value
Batch size	32
Epochs	100
LSTM input layer size	3
Learning rate	0.0003
Dropout rate	0.4
Hidden layer activation function	ReLU
Output layer activation function	Softmax

where L measures how close the explanation is to the prediction of the original model f, $\Omega(g)$ is the penalty for model complexity, and π_x is the magnitude of neighbourhood around instance x.

3.5 Reinforcement learning model

Reinforcement learning allows to have a single objective throughout an entire time period. This is a useful trait for investment management, as decisions could be made every day, with the objective of maximizing returns in the entire period (or alternatively a return-risk measure, like the Sharpe Ratio).

Some of the most common variations of Reinforcement Learning algorithm are Deep Deterministic Policy Gradient (DDPG), Advantage Actor Critic (A2C) and Proximal Policy Optimization (PPO).

Proximal policy optimization is a policy gradient reinforcement learning algorithm, which includes a policy update clipping feature. This lowers the chances of the policy to be updated into a less desirable state and overall stabilizes the learning process. It is achieved by introducing a cap and a floor to the loss function value. The loss function and the constraint are defined in Eqs. (5) and (6) respectively:

$$L(s,a,\theta_k,\theta) = \min\left(\frac{\pi_\theta(a|s)}{\pi_{\theta_k}(a|s)}A_{\pi_{\theta_k}}(s,a), g(\epsilon, A_{\pi_{\theta_k}})\right) \tag{5}$$

$$g(\epsilon, A) = \begin{cases} (1+\epsilon)A, A \geq 0 \\ (1-\epsilon)A, A < 0 \end{cases} \tag{6}$$

Here, a—action, s—state, π_θ—current policy, π_{θ_k}—previous policy, A—advantage, $A_{\pi_{\theta_k}}$—advantage of the previous policy, ϵ—hyperparameter used to regulate clipping range.

After testing all three variations on our dataset, we came to the conclusion that the improved stability of the PPO was of great importance, as it was the only algorithm that would learn in a consistent manner. Thus, only PPO will be considered from now on.

The state space s_t is updated every hour and consists of the following variables:

- Return R_t
- Traded volume V_t
- Hour of the day
- Technical indicators:
  - Moving average convergence divergence (MACD) (Appel, 2005)

$$MACD = EMA_{12} - EMA_{26} \tag{7}$$

    where EMA_t—exponential moving average for t days
  - Relative strength index (RSI) (Wilder, 1978)

$$RSI = 100 - \left[100/\left(1 + \frac{n_{up}}{n_{down}}\right)\right] \tag{8}$$

    where n_{up}—average gain during upside periods and n_{down}—average gain during downside periods
  - Commodity channel index (CCI) (Schlossberg, 2006)

$$CCI = \frac{1}{0.015}\frac{p_t - MA(p_t)}{MD(p_t)} \tag{9}$$

where $p_t = \frac{p_{high}+p_{low}+p_{close}}{3}$—typical price, p_{high}—highest price of the day, p_{low}—lowest price of the day, p_{close}—end of day price, MA—moving average and MD—mean absolute deviation

  - Average directional movement index (ADX) (Wilder, 1978)

- Current regime prediction

Generally, the regimes $w_{i,t}$ are represented as three variables with the following constraints:

$$w_{i,t} \in [0, 1] \tag{10}$$

$$\sum_{i=1}^{N} w_{i,t} = 1 \tag{11}$$

There are two ways to get w_t as a result:

- Winner-takes-all—the regime g_t with the highest probability will have a value of 1. For example, if regime "Low" is predicted, the input vector is (0, 0, 1).
- Weighted average—the values are probabilities of a regime being in place. This is essentially the raw output of the regime prediction stage.

Both of these options will be examined as separate models. In addition, a third version without any regimes is tested as a reference.

After the actions are received, the reward is calculated:

$$C_t = a_t \times B_{t-1}/P_{t-1} \tag{12}$$

$$B_t = C_t \times P_t + (1 - a_t) \times B_{t-1} \tag{13}$$

$$r_t = B_t - B_{t-1} \tag{14}$$

$$R = \sum_{t=1}^{T} r_t \tag{15}$$

B_t—wealth at time t, C_t—amount of BTC bought, P_t—price of BTC.

Table 4 F1-Score for different methods and λ

	$\lambda = 0.5$	$\lambda = 0.75$	$\lambda = 1$
LSTM with weights	1.000	0.787	0.749
LSTM w/o weights	0.240	0.436	0.624
ARMA-GARCH	0.595	0.696	0.800

4 Experimental results

4.1 Regime forecasting

4.1.1 Performance

At this stage, the data are split a follows:

- Training set: 08.08.2015–10.12.2019
- Test set: 11.12.2019–27.02.2022

The results for BTC are presented in the following tables for various metrics. As a reference for the main LSTM with weights, results for LSTM without weights and the ARMA-GARCH model are presented (Table 4):

The results demonstrate, that LSTM with weights performs better, the higher the weight for a regime switch is. Regulating λ will therefore have significant influence over the performance of the RL model, though an optimal λ cannot be determined at this stage. Figure 3 shows the regimes as defined in Sect. 3.2 as well as prediction under different λ.

A low λ would encourage predicting the previous regime, as regime changes are not very common. A high λ would immediately try to guess the next regime instead. In order to have a behaviour, where regime changes are predicted with good timing, λ is set to 0.5.

4.1.2 Interpretation

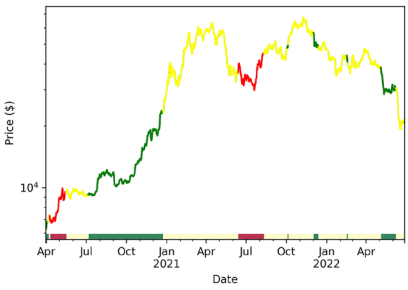
The model will be interpreted using LIME (Local Interpretable Model-Agnostic Explanations) (Ribeiro et al., 2016). LIME is a local surrogate model, i.e. it approximates the predictions of the black box model with an interpretable one. To determine the impact for each variable $\xi(x)$, values are perturbed, forming a new dataset, then the interpretable model is fit, such that the loss function L is minimized:

$$\xi(x) = \arg\min_{g \in G} L(f, g, \pi_x) + \Omega(g) \tag{16}$$

Here, f is the original model and g is the interpretable model (out of all possible models G). π_x is the proximity measure that defines the neighbors around data point x used for explanation. While the model complexity $\Omega(g)$ is often stated as a penalty for model complexity, most implementations only minimize the loss function. This is also the case in the python implementation of package "lime", which is used in the practical implementation. The interpretable model in the package is the Ridge regression.

The full procedure is as follows. To get an explanation for an instance, first, get the perturbations and run the original model. The perturbations are generated by

Fig. 3 Bitcoin price plot with regimes. Regime breakdown: zone 2, distress. RRL_Regime-Prediction



(a) Defined regimes

(b) Regime prediction for $\lambda = 0.5$

(c) Regime prediction for $\lambda = 0.75$

(d) Regime prediction for $\lambda = 1$

drawing from a normal distribution, the mean and standard deviation being that of the feature. Afterwards, the samples are weighted by their proximity using an

Table 5 Reinforcement learning parameters

Parameter	Value
Epochs	50
Total training steps	29,26,750
Learning rate	0.000005
Gamma	1

exponential smoothing kernel (see Eq. 17), which are then used to train the interpretable model.

$$K(x) = \sqrt{e^{-x^2} / \frac{n\sqrt{3}}{2}} \tag{17}$$

To determine the most influential variables, the LIME-values were calculated for every data point in the test set. A sample for a single data point can be found in Appendix B.

For an approximation of the overall influence, the variables are ranked based on the absolute values (with rank 1 being the most influential one) for every data point in the test set. The mean of the values is then calculated for the final ranking, which can be observed in Appendix B.

It it fairly obvious that previous regimes, price, volatility and drawdown all have a considerable influence over the current regime. However, metadata variables, such as sent coins, average transaction fee per day, average transaction value per day, number of unique addresses per day and average fee to reward. These variables mostly relate to trading volume. Thus, there is potential that some metadata can be used to substitute or even complement trading volume data.

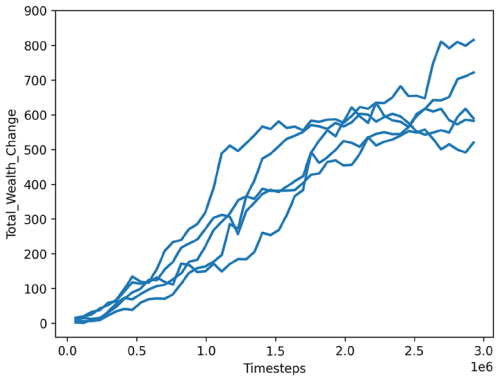
4.2 Reinforcement learning model

We will use the "stable-baselines3" python package for the reinforcement learning training. First, the training of the model is performed. The hyperparameters that are used for that are stated in Table 5.

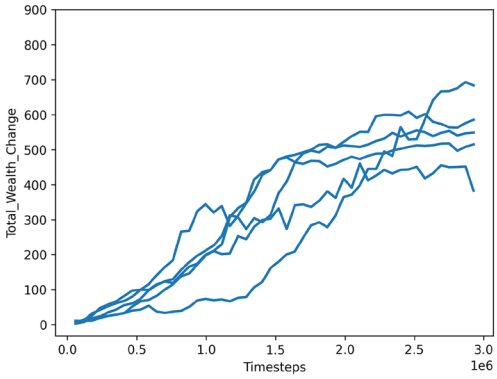
The learning rate, while very small, was necessary to ensure stable performance improvement throughout the learning period. Thus, the slow learning was compensated with a large number of training steps. For this task, reward decay would serve no purpose, hence gamma is set to 1.

The dataset is composed like this:

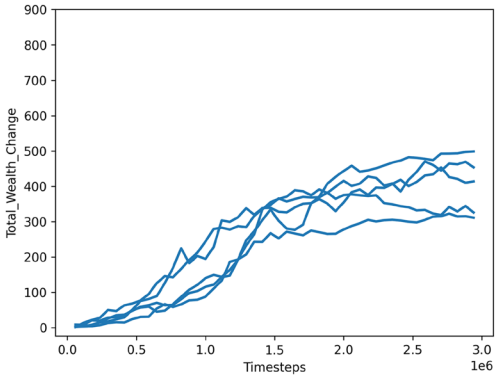
- Training set: 06.03.2020–08.07.2021
- Validation set: 09.07.2021–19.12.2021
- Test set: 20.12.2021–31.05.2022



(a) Regime prediction probabilities



(b) Regime predictions



(c) No regimes

Fig. 4 Training set reward progression during the training process. Five runs were made for each dataset to confirm the consistency of the results. RRL_Reinforcement-Learning

Then, for all three variants of the model, the rewards are tracked as they are trained. They can be observed in Fig. 4. As the training process is stochastic, each variant is run five times to make sure the performance is consistent.

Table 6 Annualized performance metrics (return, volatility and Sharpe ratio)

Regimes	Training set			Validation set			Test set		
	Return	Vol	SR	Return	Vol	SR	Return	Vol	SR
Weighted	3.8919	2.2193	1.7536	0.1307	0.3798	0.3442	−0.3779	0.3340	−1.1316
Predicted	2.8668	1.6422	1.7457	0.5325	0.3108	1.7134	−0.0876	0.2991	−0.2929
None	2.3176	2.1320	1.0870	0.4676	0.4924	0.9497	−0.5680	0.3525	−1.6112

At least in the training set, the additional regime data does make a difference. Both the predicted regime probabilities and regime values significantly increase the reward compared to the "no-regimes" variant for the same training period, so it can be stated that the regimes do contain meaningful information for investment purposes.

As for the performance in the validation and test sets, the difference there is rather insignificant. The performance metrics for the training, validation and test sets can be observed in Table 6, while the reward development over time is shown in Appendix C.

Unfortunately, it seems like the non-stationary nature complicates any kind of prediction task. Of course, it could just be, that regime predictions happened to be beneficial for the task in one time period, but not the other. It could therefore be a case of insufficient data, as regime changes are uncommon during the given time period.

5 Discussion and limitations

While using a secondary model for regime determination has its benefits in certain use cases, it comes with several limitations. First, there should be meaningful information to determine the regime. In this case, the obtained metadata was helpful for this task. Second, the data distribution should stay relatively the same between training and test set. This obviously complicates any kind of prediction task. Here, the regime definitions may introduce noise to a future time series. Thus, a sufficiently large time series for training is required. Finally, it cases where the two-step approach is applicable, a one-step approach should also be considered, with data used for the regime determination being instead part of the Reinforcement learning dataset.

In the future, this approach could be tested in different environments, such as the stock market, where larger amount of data is available. Additionally, other Reinforcement learning architectures could be used. One example would be to split the network in a way that each network would train for a separate regime (see Fig. 5). To make the current version of the model more comparable to a buy-and-hold strategy and expand on its ideas, trading costs, leverage and short-selling could be introduced.

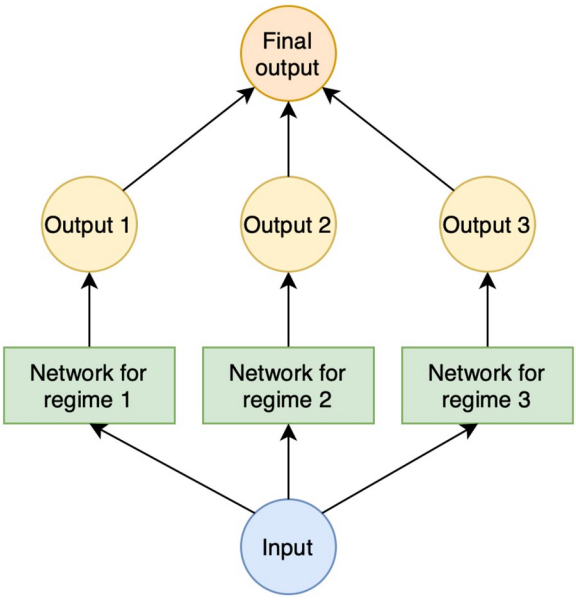


Fig. 5 Sample architecture of the policy learning network

6 Conclusion

A new approach to analyze the cryptocurrency market was introduced and applied to Bitcoin (BTC). Regimes were introduced that could be useful for further analysis. Using a weight system to counteract imbalance, a method was shown that could effectively predict future regimes with a high emphasis on when the regime switches. The LIME method was used to gain insight to what variables from the metadata had the most impact on the outcome. Those variables ended up being related to trading volume.

Afterwards, the regime prediction was used as part of a dataset in a reinforcement learning setting and compared to a version without this data. Both the predictions themselves and the probabilities were attempted. While the results were promising during the training process, with both the regimes and the probabilities achieving higher reward values with the same architecture and hyperparameters. However, for out-of-sample data, an improvement could not be detected. Therefore, more investigation on the potential application opportunities for regime prediction needs to be done.

Appendix

A. Descriptive statistics

See Table 7.

Table 7 Descriptive statistics of the Bitcoin metadata

	Mean	STD	Min	25%	50%	75%	Max
avg_transaction_value_per_day	0.0089	0.1509	−0.8521	−0.0181	−0.0005	0.0199	4.3792
avg_hashrate_per_day	0.0006	0.0131	−0.0485	−0.0054	−0.0006	0.0050	0.1393
num_transactions_per_day	466.81	401.47	39.27	120.75	293.20	752.42	2070.63
mining_profitability	1.2479	1.3671	0.0710	0.1868	0.5480	1.9148	6.2230
sent_coins	0.0110	0.1782	−0.8608	−0.0186	−0.0004	0.0186	3.9734
avg_block_time_per_day	0.000003	0.001981	−0.023922	−0.000192	0.000000	0.000192	0.024738
avg_price_per_day	0.000424	0.015763	−0.060496	−0.008477	−0.001770	0.006489	0.142100
avg_transaction_fee_per_day	0.000877	0.001441	0.000028	0.000105	0.000317	0.000948	0.007897
market_capitalization	7797052	13217950	361375	1789636	3052346	7686239	80923482
avg_fee_to_reward	0.019344	0.698054	−0.900255	−0.008290	0.000096	0.008693	33.565310
active_addresses_per_day	688.15	554.54	68.40	192.65	463.73	1083.37	2264.90
num_unique_addresses_per_day	496.81	439.68	18.67	114.70	333.63	792.18	2110.80
avg_block_size_per_day	419.14	206.56	133.20	285.73	349.36	493.97	1760.39
avg_mining_difficulty_per_day	46.52	33.22	2.41	11.52	52.67	71.92	157.38

B. LIME analysis

See Figs. 6, 7, 8 and Table 8.

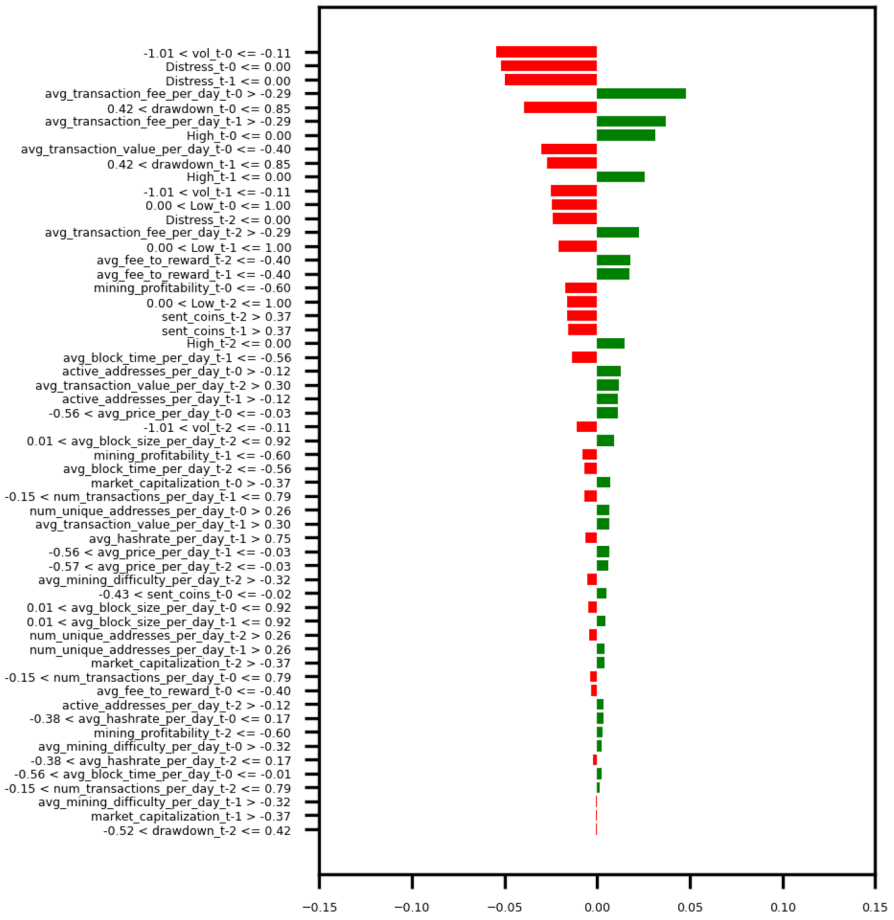


Fig. 6 Local explanation values for class "Distress" for March 21, 2020. Predicted regime is "Low". RRL_Regime-Prediction

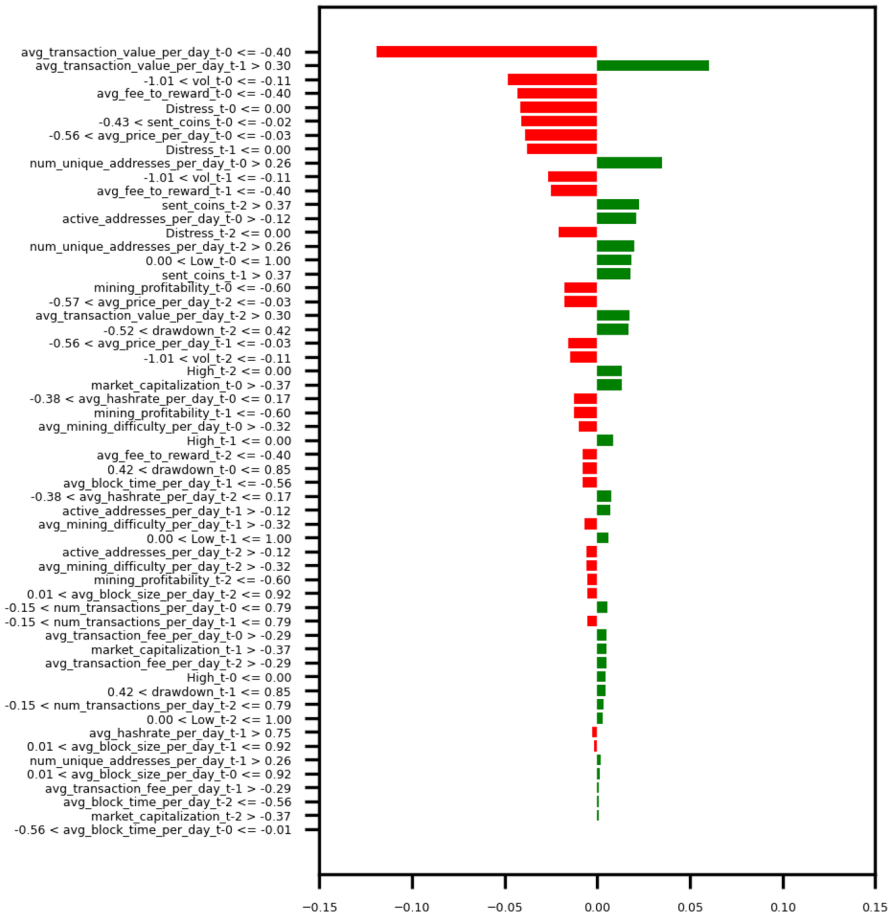


Fig. 7 Local explanation values for class "High" for March 21, 2020. RRL_Regime-Prediction

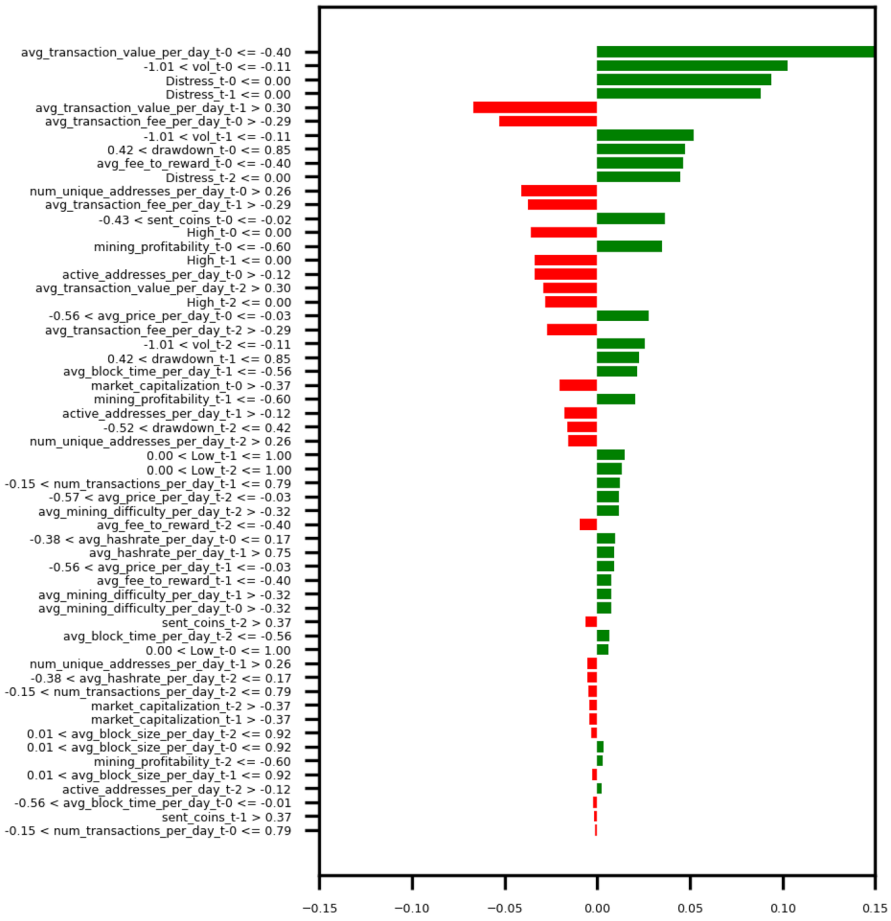


Fig. 8 Local explanation values for class "Low" for March 21, 2020. RRL_Regime-Prediction

Table 8 Average importance rank for the test set

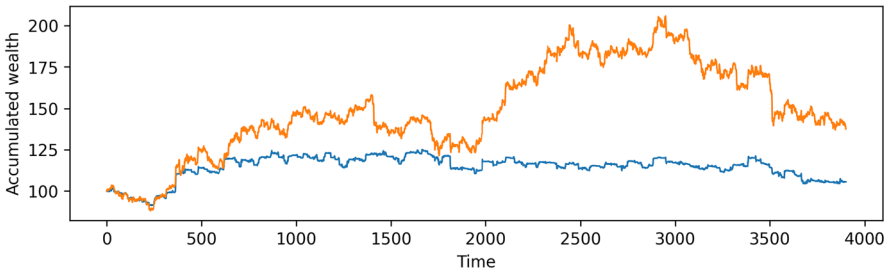
	Distress	High	Low
avg_transaction_value_per_day_t-2	38.82	24.01	27.52
avg_transaction_value_per_day_t-1	32.79	20.35	22.31
avg_transaction_value_per_day_t-0	23.04	14.17	15.21
avg_hashrate_per_day_t-2	42.75	36.15	41.71
avg_hashrate_per_day_t-1	42.28	34.47	37.90
avg_hashrate_per_day_t-0	37.32	39.23	42.81
num_transactions_per_day_t-2	40.10	39.21	43.94
num_transactions_per_day_t-1	34.27	40.36	39.14
num_transactions_per_day_t-0	32.23	31.29	44.05
mining_profitability_t-2	40.06	31.13	34.18
mining_profitability_t-1	35.57	27.96	29.86
mining_profitability_t-0	25.74	22.28	23.19
sent_coins_t-2	33.86	32.22	43.08
sent_coins_t-1	30.11	26.81	42.45
sent_coins_t-0	30.65	15.10	21.05
avg_block_time_per_day_t-2	41.27	39.49	41.10
avg_block_time_per_day_t-1	42.96	40.04	43.88
avg_block_time_per_day_t-0	40.51	35.59	37.02
avg_price_per_day_t-2	22.63	19.70	32.37
avg_price_per_day_t-1	18.90	16.75	27.62
avg_price_per_day_t-0	13.46	14.22	13.23
avg_transaction_fee_per_day_t-2	29.31	37.16	30.81
avg_transaction_fee_per_day_t-1	22.57	34.50	24.21
avg_transaction_fee_per_day_t-0	14.91	33.50	18.12
market_capitalization_t-2	41.15	38.28	39.63
market_capitalization_t-1	40.74	37.22	37.41
market_capitalization_t-0	37.32	36.48	34.80
avg_fee_to_reward_t-2	33.36	30.01	42.56
avg_fee_to_reward_t-1	32.53	26.70	39.38
avg_fee_to_reward_t-0	40.53	20.18	25.38
active_addresses_per_day_t-2	36.60	40.13	40.01
active_addresses_per_day_t-1	35.82	39.95	37.71
active_addresses_per_day_t-0	32.82	26.11	26.07
num_unique_addresses_per_day_t-2	42.14	34.10	37.62
num_unique_addresses_per_day_t-1	41.89	28.80	32.79
num_unique_addresses_per_day_t-0	43.31	20.95	26.77
avg_block_size_per_day_t-2	42.35	34.03	38.25
avg_block_size_per_day_t-1	38.27	33.28	32.84
avg_block_size_per_day_t-0	28.23	37.77	30.86
avg_mining_difficulty_per_day_t-2	42.85	39.53	43.67
avg_mining_difficulty_per_day_t-1	42.19	38.65	43.81
avg_mining_difficulty_per_day_t-0	42.71	29.96	37.56
vol_t-2	11.19	9.15	7.41

Table 8 (continued)

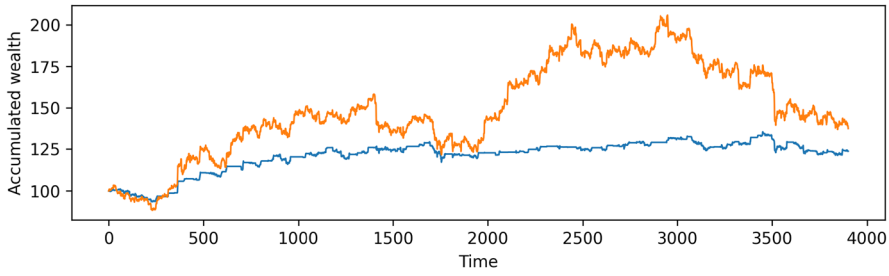
	Distress	High	Low
vol_t-1	7.09	5.82	4.46
vol_t-0	4.03	1.99	1.43
drawdown_t-2	24.27	35.91	35.67
drawdown_t-1	17.89	39.40	24.66
drawdown_t-0	9.15	28.53	14.43
Distress_t-2	16.63	18.04	12.38
Distress_t-1	9.06	10.18	6.83
Distress_t-0	3.48	7.30	3.88
High_t-2	16.12	29.62	16.55
High_t-1	11.71	23.14	12.48
High_t-0	8.74	24.61	10.90
Low_t-2	21.02	41.28	30.81
Low_t-1	16.70	40.92	27.13
Low_t-0	15.03	39.29	20.07

C. Trading results

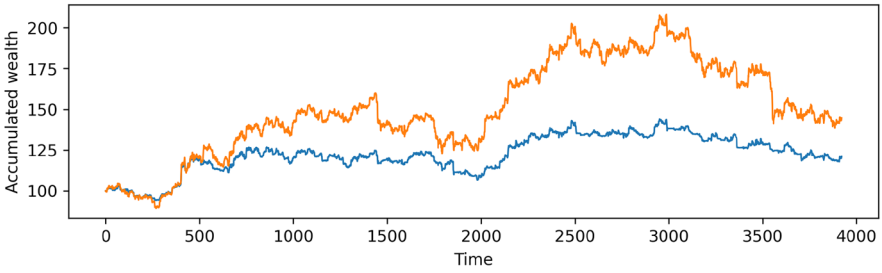
See Figs. 9 and 10.



(a) Regime prediction probabilities

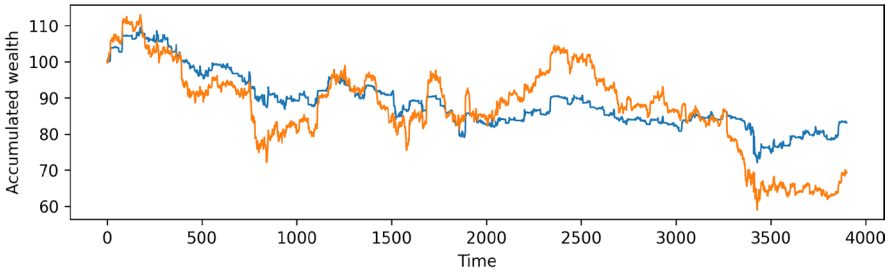


(b) Regime predictions

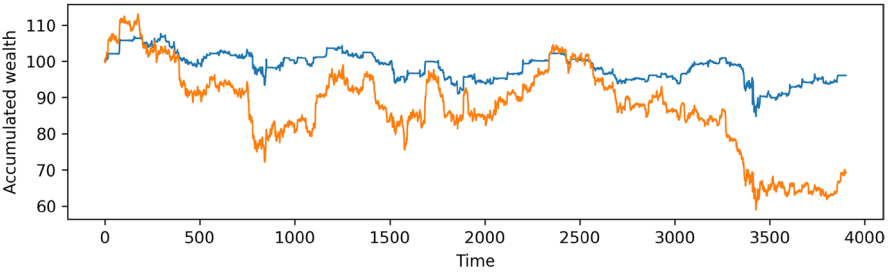


(c) No regimes

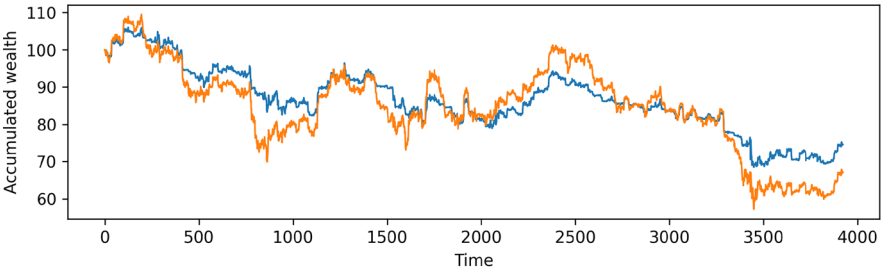
Fig. 9 Validation set



(a) Regime prediction probabilities



(b) Regime predictions



(c) No regimes

Fig. 10 Test set. RRL_Reinforcement-Learning

Author Contributions Ilyas Agakishiev has done: data analysis, coding, Wolfgang Karl Härdle proposed the regime switching framework, Denis Becker implemented the data analysis and Xiaorui ZUO has put the research elements together and assisted in coding the reinforcement training

Funding Open Access funding enabled and organized by Projekt DEAL. Deutsche Forchungsgemeinschaft IRTG1792.

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no conflict of interest.

References

Appel, G. (2005). *Technical analysis: Power tools for active investors*. FT Press.

Box, G. E., Jenkins, G. M., Reinsel, G. C., & Ljung, G. M. (2015). *Time series analysis: Forecasting and control*. John Wiley & Sons.

Candila, V. (2021). Multivariate analysis of cryptocurrencies. *Econometrics*. https://doi.org/10.3390/econometrics9030028

Charpentier, A., Élie, R., & Remlinger, C. (2021). Reinforcement learning in economics and finance. *Computational Economics, 62*, 425–462. https://doi.org/10.1007/s10614-021-10119-4

Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning phrase representations using RNN encoder–decoder for statistical machine translation. In A. Moschitti, B. Pang, & W. Daelemans (Eds.), *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1724–1734). Association for Computational Linguistics.

Härdle, W.K., & Trimborn, S. (2015). Crix or evaluating blockchain based currencies. Technical report. SFB 649 discussion paper.

Hasselt, Hv., Guez, A., & Silver, D. (2016). Deep reinforcement learning with double q-learning. *Proceedings of the thirtieth AAAI conference on artificial intelligence* (pp. 2094–2100). AAAI Press.

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation, 9*, 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735

Hou, A.J., Wang, W., Chen, C.Y.H., & Härdle, W.K., (2019). Pricing cryptocurrency options: The case of bitcoin and crix. Available at SSRN 3159130 .

Karim, M. M., Ali, M. H., Yarovaya, L., Uddin, M. H., & Hammoudeh, S. (2023). Return-volatility relationships in cryptocurrency markets: Evidence from asymmetric quantiles and non-linear ardl approach. *International Review of Financial Analysis, 90*, 102894. https://doi.org/10.1016/j.irfa.2023.102894

Kim, A., Trimborn, S., & Härdle, W. K. (2021). Vcrix—a volatility index for crypto-currencies. *International Review of Financial Analysis, 78*, 101915.

Lecun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*, 2278–2324. https://doi.org/10.1109/5.726791

Li, Y., Zheng, W., & Zheng, Z. (2019). Deep robust reinforcement learning for practical algorithmic trading. *IEEE Access, 7*, 108014–108022. https://doi.org/10.1109/ACCESS.2019.2932789

Lillicrap, T.P., Hunt, J.J., Pritzel, A., Heess, N., Erez, T., Tassa, Y., Silver, D., & Wierstra, D., (2016). Continuous control with deep reinforcement learning. In Y. Bengio & Y. LeCun (Eds.), *4th International conference on learning representations*. ICLR 2016, San Juan, Puerto Rico, May 2–4, 2016, conference track proceedings. arXiv:1509.02971.

Liu, F., Li, Y., Li, B., Li, J., & Xie, H. (2021). Bitcoin transaction strategy construction based on deep reinforcement learning. *Applied Soft Computing, 113*, 107952. https://doi.org/10.1016/j.asoc.2021.107952

Lucarelli, G., & Borrotti, M. (2020). A deep q-learning portfolio management framework for the cryptocurrency market. *Neural Computing and Applications*. https://doi.org/10.1007/s00521-020-05359-8

Maringer, D., & Ramtohul, T. (2012). Regime-switching recurrent reinforcement learning for investment decision making. *Computational Management Science, 9*, 89–107. https://doi.org/10.1007/s10287-011-0131-1

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T., Harley, T., Silver, D., & Kavukcuoglu, K. (2016). Asynchronous methods for deep reinforcement learning. In M. F. Balcan & K. Q.

Weinberger (Eds.), *Proceedings of the 33rd international conference on machine learning* (pp. 1928–1937). New York, New York, USA: PMLR.

Mosavi, A., Faghan, Y., Ghamisi, P., Duan, P., Ardabili, S. F., Salwana, E., & Band, S. S. (2020). Comprehensive review of deep reinforcement learning methods and applications in economics. *Mathematics*. https://doi.org/10.3390/math8101640

Rabiner, L., & Juang, B. (1986). An introduction to hidden Markov models. *IEEE ASSP Magazine, 3*, 4–16. https://doi.org/10.1109/MASSP.1986.1165342

Ribeiro, M., Singh, S., & Guestrin, C. (2016). "Why should I trust you?'': Explaining the predictions of any classifier. In J. DeNero, M. Finlayson, & S. Reddy (Eds.), *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: demonstrations* (pp. 97–101). San Diego, California: Association for Computational Linguistics.

Sattarov, O., Muminov, A., Lee, C. W., Kang, H. K., Oh, R., Ahn, J., Oh, H. J., & Jeon, H. S. (2020). Recommending cryptocurrency trading points with deep reinforcement learning approach. *Applied Sciences*. https://doi.org/10.3390/app10041506

Schlossberg, B. (2006). *Technical analysis of the currency market*. Wiley Trading.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. CoRR abs/1707.06347. http://dblp.uni-trier.de/db/journals/corr/corr1707.html#SchulmanWDRK17.

Trimborn, S., & Härdle, W. K. (2018). Crix an index for cryptocurrencies. *Journal of Empirical Finance, 49*, 107–122.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L.u., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in neural information processing systems.* Curran Associates, Inc.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., & Lang, K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing, 37*, 328–339. https://doi.org/10.1109/29.21701

Wang, Z., Schaul, T., Hessel, M., Van Hasselt, H., Lanctot, M., & De Freitas, N. (2016). Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33rd international conference on international conference on machine learning* (Vol. 48, pp. 1995–2003).

Wilder, J. (1978). New concepts in technical trading systems. Trend Research. https://books.google.de/books?id=WesJAQAAMAAJ.

Authors and Affiliations

Ilyas Agakishiev[1] · Wolfgang Karl Härdle[1] · Denis Becker[2] · Xiaorui Zuo[3]

✉ Wolfgang Karl Härdle
haerdle@hu-berlin.de

Ilyas Agakishiev
ilyas.agakishiev@hu-berlin.de

Denis Becker
denis.becker@ntnu.no

Xiaorui Zuo
e1349341@u.nus.edu

[1] Humboldt Universität zu Berlin, Dorotheenstraße 1, Berlin, Germany

[2] NTNU Business School, Postboks 8900, Torgarden, 7491 Trondheim, Norway

[3] NUS Risk Management Institute, Shaw Foundation, Alumni House, 11 Kent Ridge Dr, #03-04, 119244 Singapore, Singapore