KeAi
CHINESE ROOTS
GLOBAL IMPACT

Available online at www.sciencedirect.com

ScienceDirect

The Journal of Finance and Data Science 8 (2022) 331–352
http://www.keaipublishing.com/en/journals/jfds/

Machine learning for cryptocurrency market prediction and trading

Patrick Jaquart*, Sven Köpke, Christof Weinhardt

Institute of Information Systems and Marketing, Karlsruhe Institute of Technology, Germany

Received 29 August 2022; revised 4 December 2022; accepted 4 December 2022
Available online 14 December 2022

Abstract

We employ and analyze various machine learning models for daily cryptocurrency market prediction and trading. We train the models to predict binary relative daily market movements of the 100 largest cryptocurrencies. Our results show that all employed models make statistically viable predictions, whereby the average accuracy values calculated on all cryptocurrencies range from 52.9% to 54.1%. These accuracy values increase to a range from 57.5% to 59.5% when calculated on the subset of predictions with the 10% highest model confidences per class and day. We find that a long-short portfolio strategy based on the predictions of the employed LSTM and GRU ensemble models yields an annualized out-of-sample Sharpe ratio after transaction costs of 3.23 and 3.12, respectively. In comparison, the buy-and-hold benchmark market portfolio strategy only yields a Sharpe ratio of 1.33. These results indicate a challenge to weak form cryptocurrency market efficiency, albeit the influence of certain limits to arbitrage cannot be entirely ruled out.

Keywords: Financial market prediction; Market efficiency; Statistical arbitrage; Machine learning; GRU; LSTM; Neural network; Random forest; Gradient boosting; Temporal convolutional neural network

1. Introduction

In 2008, Nakamoto[1] has introduced the electronic peer-to-peer cash system Bitcoin to the world. Since then, Bitcoin has inspired numerous other cryptocurrencies with varying technical properties and use cases. Over the last decade, the cryptocurrency market has grown tremendously, whereby individual cryptocurrency prices have exhibited large volatility.[2] There exists mixed evidence with regards to the market efficiency of Bitcoin and other cryptocurrencies.[3,4,5] These studies usually apply specific statistical tests that are in some form based on auto-regressive approaches, whereby potential non-linear interactions, if considered, are modeled explicitly. Machine learning methods can flexibly learn the functional form between features and targets[6] and have been applied successfully to the domain of Bitcoin and cryptocurrency market prediction in the past.[7,8,9] Therefore, these methods may uncover and utilize high-dimensional non-linear feature interactions beyond the interactions modeled in specific market efficiency tests. In this study, we shed

* Corresponding author.
E-mail addresses: patrick.jaquart@kit.edu (P. Jaquart), sven.koepke@student.kit.edu (S. Köpke), christof.weinhardt@kit.edu (C. Weinhardt).
Peer review under responsibility of KeAi.

https://doi.org/10.1016/j.jfds.2022.12.001

light on the potential of different machine learning models regarding market prediction and trading. Hence, the overarching research question of this study is:

Research Question: *What is the performance of machine learning models for generating statistical arbitrage in the cryptocurrency market?*

To answer this research question, we employ six machine learning classifiers to predict the relative daily performance of the 100 largest cryptocurrencies by market capitalization. Furthermore, we employ a long-short trading strategy based on the out-of-sample predictions of each model and evaluate the resulting trading outcomes. We analyze five heterogeneous study periods, with each spanning 800 days. This study has two main contributions:

First, we highlight the potential of machine learning for cryptocurrency market prediction, as all employed models make statistically viable predictions. In doing that, we find that recurrent neural networks and tree-based ensembles are particularly effective in classifying the relative daily performance of cryptocurrencies. Second, we demonstrate the potential for statistical arbitrage in the cryptocurrency market, as the employed long-short portfolio strategy outperforms the market benchmark on a risk-adjusted basis after transaction costs.

The remainder of this paper is structured as follows: Chapter 2 presents related work, Chapter 3 describes our methodological approaches, and Chapter 4 presents the results of our analyses. 5 discusses the implications of these results and Chapter 6 concludes this study.

2. Related work

Fischer et al[8] examine the potential of machine learning predictions to generate statistical arbitrage in the cryptocurrency market utilizing a dataset from June to September 2018. They train a random forest classifier and a logistic regression model to predict the relative performance of the 40 largest cryptocurrencies over the next 120 min based on the temporal distribution of past returns over the last day. As an out-of-sample long-short trading strategy based on these model predictions yields daily returns of 7.1 bps per day, their findings indicate an impairment of cryptocurrency market efficiency. Fil and Kristoufek[10] apply pairs trading to the cryptocurrency market, assuming a long-term stable state between different cryptocurrency pairs. They use data from January 2018 to September 2019 and compare 5-min, hourly, and daily trading frequencies. Fil and Kristoufek[10] find that pairs-trading may perform well for shorter frequencies in the cryptocurrency market, whereby these results are highly dependent on the selected market parameters (e.g., the magnitude of transaction cost).

Betancourt and Chen[11] examine the potential of deep reinforcement learning for cryptocurrency trading based on a dataset ranging from August 2017 until November 2020. In the presented approach, agents repeatedly analyze the 20-day history of price, volume, and market capitalization of a specific cryptocurrency to make one-day trading decisions. Betancourt and Chen[11] find that their approach is promising for cryptocurrency trading. McNally et al[12] compare an Elman recurrent neural network, a long short-term neural network, and an autoregressive integrated moving average approach to predict binary daily Bitcoin market movements. They utilize data from August 2013 to July 2016 and find that the long short-term neural network exhibits the highest predictive performance with a model accuracy of 52.78%. Dutta et al[13] compare different neural network approaches to predict daily Bitcoin prices based on a feature set consisting of various technical, blockchain-based, asset-based, and interest-based features. They find that a gated recurrent unit implementation with recurrent dropout yields the highest performance on their dataset, which ranges from January 2010 until June 2019.

Chen et al[14] employ and compare various linear statistical methods and machine learning approaches for 5-min and daily Bitcoin market prediction on data from February 2017 to February 2019. They document a higher predictive performance of the employed statistical methods for the daily prediction horizon, while the machine learning methods exhibit a higher performance on the 5-min horizon. Alessandretti et al[15] design different models based on gradient boosting classifiers and long short-term neural network approaches to predict the daily returns of 1681 cryptocurrencies. They utilize data from November 2015 until April 2018 and show that portfolio strategies based on these predictions outperform a baseline approach. Lahmiri and Bekiros[16] compare a long short-term memory neural network and a generalized regression neural network approach to predict the prices of Bitcoin, Digital Cash, and Ripple. They utilize data sets with different temporal lower bounds that end in October 2018 and find that the employed long short-term memory neural network yields better forecasts than the generalized regression neural network.

3. Methodology

The study comprises four main steps and builds on the methodological approaches Fischer et al.[8], Fischer Krauss[17] In the first step, we obtain the relevant data from various sources. We then generate features and targets from the raw price data, which we use to model coin returns. The next step is to split the complete data set into overlapping study periods with varying market constituents and non-overlapping test folds for backtesting. The final step is to train and tune the models used, individually for all study periods, and simulate trading based on the model predictions.

3.1. Data

For this study, we use daily closing price and market capitalization data denoted in U.S. Dollars (USD) over the period from February 8, 2018, to May 15, 2022, obtained through CoinGecko's (CG) API.[18]

3.1.1. Coin market capitalization data

To avoid survivorship bias, the constituents of the investment universe are determined as the top 100 crypto assets by market capitalization on the first trading day of the training set. This ensures that no look-ahead bias is introduced through the construction of the coin universe while providing a sufficient number of training instances for each coin. Stablecoins pegged to the USD or any other fiat currency are excluded from the list of eligible candidates, as their prices quoted in USD are static by design or entirely dependent on currency exchange rates. We exclude a list of ten other coins due to data issues such as missing data and erroneous values in the data sources (full exclusion list in Appendix B.1).

Daily market capitalization data for the largest 1750 crypto assets (as of June 8, 2022) is obtained through the CG API, which provides market capitalization data denoted in USD calculated as the product of known available supply and the asset's price. For each study period, we rank all coins based on their market capitalization on the first trading day of the training set and the top 100 cryptocurrencies are used as the asset universe for creating the crypto asset portfolios.

3.1.2. Coin price data

We use aggregated market price data from the CoinGecko for the return calculations. CG provides aggregate prices based on the pairings (cryptocurrency vs. fiat currency or cryptocurrency vs. cryptocurrency) available on all monitored exchanges by applying a global volume-based weighting. Despite these prices being non-traded aggregate prices,[19] shows that such artificially compounded prices are a fair representation of the overall crypto-currency market. The author finds that aggregating different exchange platforms to compute a singular price does not affect market efficiency for liquid cryptocurrencies. Since cryptocurrency exchanges are open around the clock, we create artificial closing prices from the market price at midnight (UTC). The CG API provides daily price data with a 00:00:00 UTC timestamp associated with the following day. Therefore, the retrieved time series of daily quotes are shifted by one day to calculate the previous day's returns. Thus, using p_t^c to denote the aggregate market price for coin c at the end of day t (measured at 00:00:00 UTC on day t+1), the m-period returns $r_t^{m,c}$ are calculated as follows:

$$r_t^{m,c} = \frac{p_t^c}{p_{t-m}^c} - 1, \tag{1}$$

where

$r_t^{m,c}$ Return for coin c on day t over the last m days

p_t^c Aggregate closing price for coin c on day t.

For $m = 1$, we thus obtain the asset's daily returns, while for $m > 1$, $r_t^{m,c}$ represents the cumulative returns over the last m days.

3.1.3. Risk-free rate of return

The U.S. treasury's three-month Treasury Bill (T-bill) secondary market rate[20] is used as the risk-free rate to calculate excess returns. The T-bill is a short-term debt obligation backed by the Treasury Department of the United

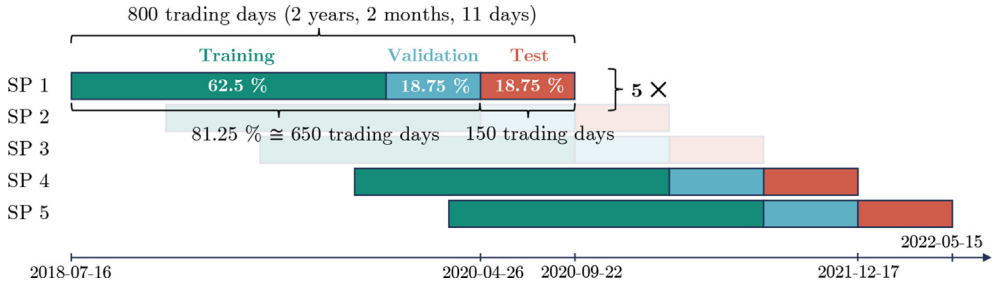

Fig. 1. Study period composition and train-validation-test split.

States government with a maturity of three months. The annual interest rate is converted to daily returns by simple deannualization to calculate daily excess returns for calculating risk-adjusted return metrics such as the Sharpe ratio and the Sortino ratio. For most of the period under consideration, the risk-free rate, as measured by the T-bill rate cited above, is close to zero, ranging from 2.4×10^{-7} to 2.8×10^{-5} per day, with a mean value of 3.9×10^{-6}.

3.2. Software and hardware

We use Python 3.9 for data acquisition, processing, and analysis throughout the study. The numpy[21] and pandas[22] software packages are used for data processing and feature creation. Deep learning models are built using Keras[23] with the *TensorFlow* backend[24] and all other machine learning models are built and trained using the *scikit-learn*[25] library. All models are trained on a CPU (Intel Core i5-8400, 2.8 GHz).

3.3. Data split

The full study time frame is divided into five overlapping study periods (SPs), each comprising 800 trading days, i.e., prediction targets. The range of dates used for each SP includes the 90 days prior to the first trading day, as each prediction uses data from a 3-month look-back period as model inputs. Each study period consists of training (500 days), validation (150 days; for hyperparameter-tuning), and out-of-sample test sets (150 days) in chronological order, as illustrated in Fig. 1. Table 1 shows the exact split of each study period into the three respective data folds.

The training and validation splits together constitute the formation period, during which the models are trained and the best hyperparameters for each model are selected based on the validation performance. The test split of each study period is used for out-of-sample testing and simulated trading. Study periods are shifted by the length of the test period to allow for five non-overlapping test sets for successive evaluation. Using multiple study periods allows for periodic retraining of models and thus captures the concept drift that occurs due to changing market phases.

3.4. Features

All models are trained on the binary classification problem of predicting whether a single coin will outperform the cross-sectional median of returns on the subsequent day, based on solely on price information of the previous 90 days. Thus, we derive the features for all models from the individual coin returns three months prior to trading.

Table 1
Study periods and the respective date ranges for the training, validation, and test sets.

SP No	Training Set	Validation Set	Test Set
1	2018-07-16 - 2019-11-27	2019-11-28 - 2020-04-25	2020-04-26 - 2020-09-22
2	2018-12-13 - 2020-04-25	2020-04-26 - 2020-09-22	2020-09-23 - 2021-02-19
3	2019-05-12 - 2020-09-22	2020-09-23 - 2021-02-19	2021-02-20 - 2021-07-19
4	2019-10-09 - 2021-02-19	2021-02-20 - 2021-07-19	2021-07-20 - 2021-12-16
5	2020-03-07 - 2021-07-19	2021-07-20 - 2021-12-16	2021-12-17 - 2022-05-15

Feature generation is performed separately for the two main types of classifiers used in this study, namely models with a memory function and models without memory. For the three deep-learning models with internal memory, the LSTM, the GRU, and the TCN, standardized daily return sequences of length 90 are created. The daily returns are standardized by subtracting the mean and dividing by the standard deviation of the respective training set. The tree-based classifiers and the LR use lagged returns as model inputs due to their lack of memory. Tuples of input sequences and target labels are created successively by generating overlapping sequences of length 90 that are iteratively shifted forward by one day. The procedure for creating input sequences with corresponding target labels for the deep learning methods is exemplified in Fig. 2.

Since the memory-free models (i.e., GBC, RF, and LR) are not inherently capable of using temporal input data, we create time-lagged features by aggregating returns over different intervals of increasing length. Based on Takeuchi & Lee[26] and, Krauss et al,[27] we use multi-period returns with lags $m \in \{\{1, 2, \ldots, 20\} \cup \{30, 40, \ldots, 90\}\}$, increasing the resolution to 10 days after using daily increments for the first 20 days, resulting in a total of 27 features per sample. The multi-period returns are calculated using Equation (1). The creation of the return features and corresponding target labels for the tree-based methods and LR is illustrated in Fig. 3. Per coin and study period (for all coins, respectively), both feature generation methods yield 500 (50,000) training samples, 150 (15,000) validation samples, and 150 (15,000) test samples.

3.5. Targets

The binary prediction problem is to forecast whether an individual coin will outperform the cross-sectional median on the day after portfolio formation. Hence, for each trading day, the daily returns for all coins are sorted in descending order, and the class label 1 is assigned to all coins above or equal to the cross-sectional median of returns and 0 otherwise. The target class for coin c at time t, y_t^c is thus given by

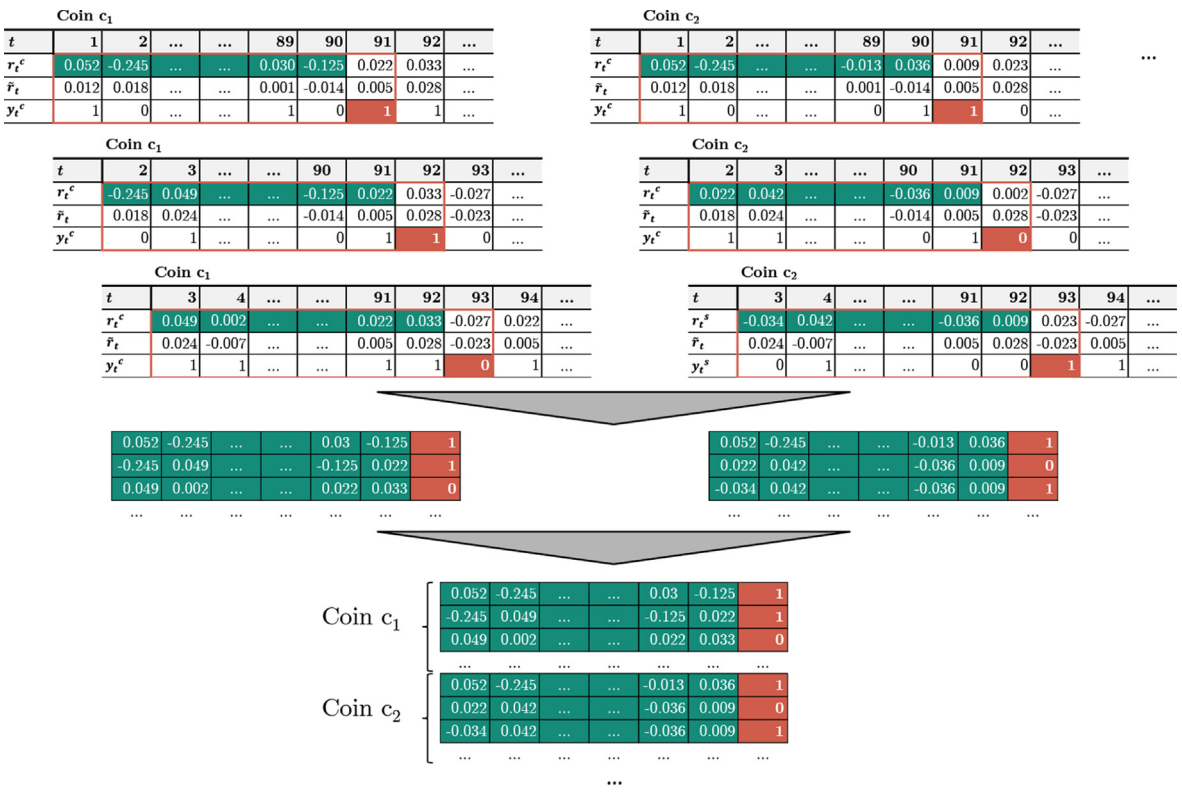

Fig. 2. Creation of feature sequences and corresponding target labels for models with memory.

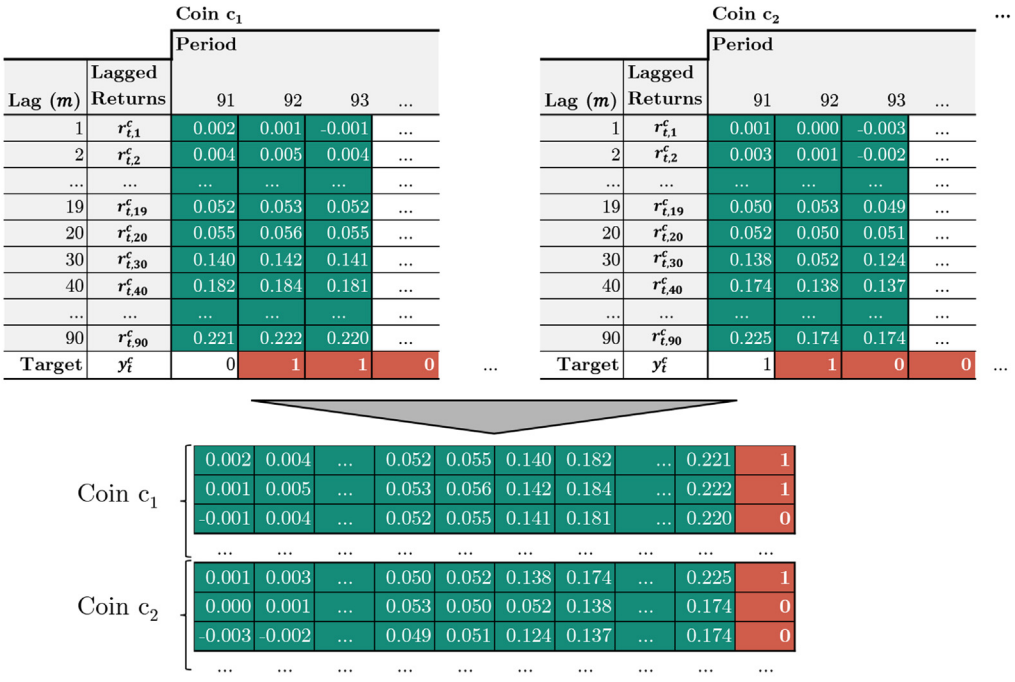

Fig. 3. Creation of tree-based and logistic regression feature sets and corresponding target labels.

$$y_t^c = \begin{cases} 1, & \text{if } r_t^c \geq Median(r_t^c),\ \forall c \in C(t) \\ 0, & \text{otherwise,} \end{cases} \tag{2}$$

where

r_t^c Daily return for coin c on day t

$C(t)$ Set of coins that are part of the coin universe on day t.

3.6. Models

We test and compare different types of predictive models, including recurrent neural networks, convolutional neural networks, tree-based ensemble methods, and the logistic regression (LR) model as a simple and efficiently computed benchmark. Due to the stochastic nature of their training process, we train all models except the logistic regression with ten different random seeds and create ensemble models by averaging the cross-sectional ranks resulting from the predicted probabilities. The hyperparameters are optimized separately for each study period using the classification

Table 2
Parameter tuning space and selected configuration per study period based on validation accuracy.

Model	Parameter Grid	SP 1	SP 2	SP 3	SP 4	SP 5
GRU	Number of memory cells: {5, 10, 15, 20}	10	10	15	20	5
LSTM	Number of memory cells: {5, 10, 15, 20}	15	20	20	10	10
TCN	Number of filters: {2, 4, 6}	6	6	6	4	4
GBC	Max. tree depth: {1, 2, 3, 5, 10}	2	1	2	2	2
RF	Max. tree depth: {1, 2, 3, 5, 10, 20, None} x max. number of features per split: {1, 3, 5, 7, 10, None}	(5, 3)	(5, 3)	(5, 1)	(3, 5)	(2, 3)
LR	–	–	–	–	–	–

accuracy of the respective validation fold for model selection. For the logistic regression, we use the default parameters without optimization. Table 2 shows the hyperparameter grid used for optimizing each model and the selected configuration per study period.

3.6.1. Deep neural networks

We compare three different deep neural network architectures from two families: recurrent neural networks (RNN) and convolutional neural networks (CNN). RNNs maintain an internal state, or memory, and can therefore process input sequences of variable length. Temporal convolutional networks (TCN) use a number of variations of standard CNN architectures that allow them to also retain a long-term memory.

All deep neural network models are trained for a maximum of 25 epochs with a batch size of 1024 and a learning rate of 0.002 using the Adam[28] optimizer for optimizing the binary cross-entropy loss. Early stopping with a patience of 4 epochs at a threshold of 1×10^{-4} with respect to the validation loss is used to mitigate overfitting and thus improve generalization.

3.6.2. Long short-term memory

Long short-term memory (LSTM) neural networks are a prominent member of the class of recurrent neural networks commonly used for time series forecasting in various domains. They were first introduced by Hochreiter et al[29] for learning long-term dependencies in long series of data and remedy the exploding vanishing and exploding gradient problems that vanilla RNNs suffer from. This is achieved by making use of several gating mechanisms (input gate, output gate, and forget gate). We use a simple architecture with a single LSTM layer containing a varying number of memory blocks of $nblocks \in \{5, 10, 15, 20\}$ and a dropout ratio of 0.1, followed by two dense layers. The first dense layer contains five neurons and uses the rectified linear unit (ReLU) as the activation function, while the final layer consists of a single neuron with a sigmoid activation function to produce a probability output in the range between zero and one.

3.6.3. Gated recurrent unit

The gated recurrent unit (GRU) is an RNN closely related to the LSTM architecture, but has fewer parameters as it lacks a dedicated output gate. This is achieved by using one gate (the update gate) that controls both the forgetting and the output simultaneously. We use the same architecture and learning parametrization as for LSTM and only replace the single LSTM layer with a single GRU layer.

3.6.4. Temporal convolutional network

Temporal convolutional networks are a fairly recent type of convolutional neural network with design characteristics that enable them to work well with long time series. The TCN architecture is characterized by (1) the use of one-dimensional causal convolutions (which ensure that the ordering of temporal data is preserved), (2) the ability to transform a sequence of arbitrary length into an output sequence of the same length, and (3) the use of dilated convolutions to enable effective long-term memory.[30]

We use a TCN architecture consisting of a single TCN layer with a kernel size of three and exponentially increasing dilation rates ($d = [1, 2, 4, 8, 16]$) for the dilated causal convolutions. The number of filters is used as a tuning parameter ($k_f \in \{2, 4, 6\}$). After the TCN layer, we use dropout of 0.25 followed by two dense layers. The first dense layers consists of five neurons with a ReLU activation and the output layer consists of a single unit and a sigmoid activation function.

3.6.5. Memory-free prediction models

We use two different memory-free, tree-based ensemble methods for benchmarking the deep learning models used: *random forest classifiers* (RF) and *gradient boosting classifiers* (GBC). Both ensemble models use a collection of decision trees as base learners to mitigate overfitting, a common problem when using single decision trees. Logistic regression is used as a simple benchmark.

3.6.6. Random forest

Random forests are based on randomized decision trees and work by creating and combining a diverse set of weak learners, i.e., decision trees, whose individual predictions are combined to an ensemble prediction by averaging their

outputs. The used RF implementation of the *scikit-learn* library combines the individual classifiers by averaging their probability predictions. We implement the random forest with the default setting of 100 base learners and tune both the maximum tree depth and the maximum number of features used to split the decision trees.

3.6.7. Gradient boosting classifier

Gradient boosting classifiers rely on the sequential construction of shallow decision-trees based on the errors made in the previous iteration to reduce the bias of the combined estimator. They are a generalization of the AdaBoost algorithm by Freund et al[31] proposed in the seminal work of Friedman.[32] We use the scikit-learn[25] implementation with the default value of 100 estimators and use the maximum depth of each trees as a tuning parameter.

3.6.8. Logistic regression

The logistic regression model serves as a simple and efficiently trainable benchmark model against which the more complex models are compared. LR is the equivalent of simple linear regression for binary response variables, as is the case with classification problems, and models the probability of a binary event occurring based on a linear combination of a set of predictors. Even though there is no closed-form solution as in standard linear regression, the global optimum can be found efficiently by numerical methods due to the convexity of the loss function. Note that the LR model is the only model used for inference without creating an ensemble of individual models trained with different seeds, since it has a unique solution and is not subject to a stochastic optimization process.

Alternatively, the LR model can be represented by a simple single-layer neural network with one neuron and a sigmoid activation function when using binary cross-entropy as the loss function. We use the *scikit-learn* implementation of LR with Newton-CG as the solver for the optimization problem and a maximum number of iterations of 1000. For all other hyperparameters, we use the default values and do not perform any tuning.

3.7. Prediction and portfolio formation

At the end of each day t in the trading period and for each model type, we predict the probability of outperforming the market on day $t+1$, $\widehat{\mathcal{P}}^{c}_{t+1|t}$, independently for each coin $c \in C$ $(t+1)$, using only information available on day t. For all model types except the LR, we make the probability predictions for all 10 individual models (ensemble constituents) trained on different random seeds. To obtain the final class predictions per model type, coin, and day, these probability predictions are sorted in descending order for each model and day and ranks are assigned accordingly. The ten individual ranks of the constituent models per model type are then averaged to obtain a final ranking for all coins per model type and day.

We opt to use the individual ensemble model constituents' rank predictions instead of the predicted probabilities for obtaining the ensemble predictions. In doing so, we account for the fact that the probability predictions are only meaningful relative to the predicted probabilities of all other assets for the same day. By averaging the ranks instead of probabilities, we retain the information about predicted relative performance that is included in the prediction of each constituent model. Based on the averaged ranks for each day, the bottom half is assigned the label 0 and the top half is assigned the label 1. In other words, we use the averaged predicted cross-sectional ranks of coins to obtain a balanced set of predictions for the balanced set of true class labels.

The averaged prediction ranks per model type and trading day are used for portfolio formation by assigning the top k coins to the long leg and the bottom k coins to the short leg for the following trading day. As a result, the created long-short portfolios for each model contain $2k$ different coins from the available asset universe of 100 coins representing the market. The final class predictions and rankings are then used to calculate the predictive accuracy of the models for different portfolio sizes by restricting the calculation to the cryptoassets selected for each portfolio.

3.8. Backtesting

We base our long-short trading strategy on the portfolio selection rule described above, using the average ranks of the model predictions to form a balanced and dollar-neutral long-short portfolio of size $2k$ for different values of k. The

portfolio positions are opened at the end of day t at the market closing prices and closed at the end of day $t+1$ after a holding period of one full day. Each time the portfolio positions are opened and closed, they incur the assumed transaction costs of 15 bps of the transaction volume. After closing of each position at the end of the holding period, the resulting cash position is used to fund the next day's trades. For each model, we thus hold a long-short portfolio containing $2k$ coins at any given time, changing its composition once per day while incurring transaction costs for every trade. The financial performance of this daily trading strategy is calculated using the net asset value of investments, which includes all coin returns and incurred trading costs.

Transaction costs. When in comes to arbitrage on cryptocurrency markets, the transaction costs for trading on various cryptocurrency exchanges must be taken into account. Transactions costs consist of commission fees, market impact and, in the case of short-selling, short-selling costs. For the calculation of daily returns, half-turn transaction costs of 15 bps are assumed, following similar work by Fischer et al.[8] Additional short-selling costs are not taken into account as short-selling of cryptoassets is not possible for all considered coins as of the time of writing and an estimation of short-selling related costs not feasible.

4. Results

In this section, we present the results of the compared prediction methods in terms of predictive accuracy and financial performance achieved with the derived long-short trading strategy. An equally-weighted buy-and-hold market portfolio (MKT) is used as the benchmark for evaluating the portfolio performance. The choice to use an equally-weighted market portfolio as the benchmark is motivated by the fact that the trading strategy evaluated is based on daily equally-weighted long-short portfolios.

First, we analyze the overall results across all five study periods for different portfolio sizes ($k \in \{1, 2, 5, 10, 20, 50\}$), where each portfolio consists of $2k$ stocks. For predictive accuracy, we then take a more granular look at the results for each study period to see how the performance varies over time. We then focus our analysis on the $k = 5$ portfolio, which contains a sufficient number of assets to diversify risk, but not so many as to negate the effect of selecting coins with a relatively high degree of certainty.

4.1. Model accuracy

The employed models' ability to accurately predict whether a coin will outperform the cross-sectional median is the basis for forming a profitable long-short portfolio. Thus, we first evaluate and compare the models in terms of their predictive accuracy. Following the approach of Fischer & Krauss,[17] we evaluate the models' accuracy by calculating the probability of a random classifier achieving the same accuracy or higher. To do this, we model the number of correctly classified coins in the chosen portfolio of size k, X_k, as a binomial distribution under the assumption of a true classification accuracy of 50%, $X_k \sim B(n = 15{,}000 \cdot \frac{2k}{100},\ p = 0.5)$, and calculate for each model the p-values for achieving the number of correctly classified coins or better by chance alone.

4.1.1. Full time period

Using the method described above, the prediction accuracy for the entire time period examined (i.e., all test sets of the five study periods combined) is significantly higher than 50% for all surveyed prediction methods and all portfolio sizes (see Fig. 4 and Table 3). For $k = 50$ (i.e., taking all 100 daily predictions into account), the RF performs best with an accuracy of 54.2%, closely followed by the LSTM with 54.1%. The probabilities of a random classifier scoring at least as good is 3.419×10^{-119} and 3.203×10^{-110}, respectively, indicating a clear advantage of the two recurrent neural networks over a random classifier. The complete summary of p-values is given in B.2. Following behind at a comparable accuracy level are the TCN and GBC (both 53.6%), and the GRU (53.5%). All machine learning models thus considerably outperform the LR benchmark (52.9%).

Note that the prediction accuracy for $k = 50$ corresponds to the models' actual (i.e., unrestricted) test set performances. For all $k < 50$, the accuracy scores are those achieved on the corresponding subset of predictions. For different portfolio sizes, the accuracy is monotonically increasing for smaller values of k for all models (see Fig. 4).

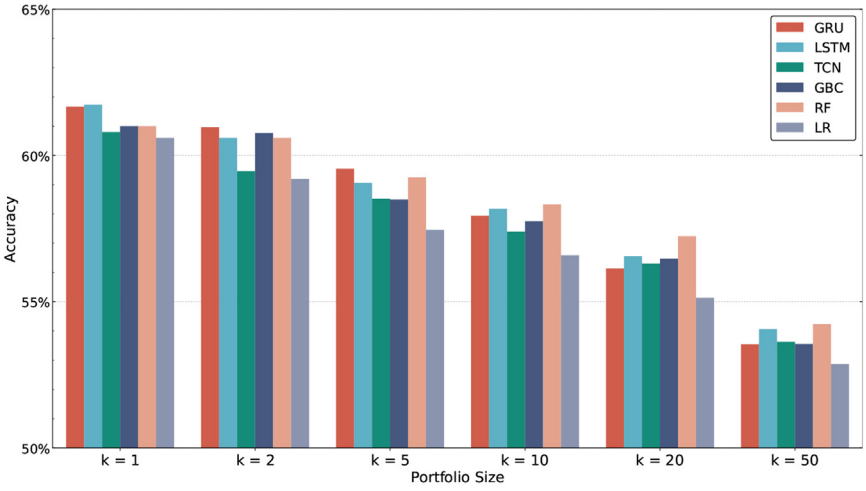

Fig. 4. Prediction accuracy for long-short portfolio and different portfolio sizes.

Table 3
Prediction accuracy for the long-short portfolio per model for different portfolio sizes.

Portfolio Size	**k = 1**	**k = 2**	**k = 5**	**k = 10**	**k = 20**	**k = 50**
GRU	0.617	0.610	0.595	0.579	0.561	0.535
LSTM	0.617	0.606	0.591	0.582	0.566	0.541
TCN	0.608	0.595	0.585	0.574	0.563	0.536
GBC	0.610	0.608	0.585	0.578	0.565	0.536
RF	0.610	0.606	0.593	0.583	0.572	0.542
LR	0.606	0.592	0.575	0.566	0.551	0.529

For the restricted out-of-sample performances the pattern of model rankings remains similar, with all models improving in accuracy as k decreases. The LR model performs worst in terms of prediction accuracy regardless of the portfolio size, while the RF maintains its lead over all other models for $k = 20$ (57.2%) and $k = 10$ (58.3%). For $k = 5$, the GRU outscores the RF with a result of 59.5% compared to 59.3% of the RF and also outperforms the LSTM model (59.1%). For the $k = 2$ portfolio, the GRU (61.0%) still performs best, closely followed by the GBC (60.8%), and remains slightly better than the RF and LSTM models (both 60.6%). The TCN is more than one percentage point behind the GRU with only 59.5%, while the LR model is even further behind with 59.2%. When only one coin is selected for the long side and one for the short side of the portfolio (i.e., $k = 1$), the two recurrent neural networks have an advantage over all other models. The LSTM narrowly outperforms the GRU with an accuracy of 61.73% compared to 61.67% of

Table 4
Prediction accuracy for long and short legs of the long-short portfolio per model for different portfolio sizes.

Portfolio Size	**k = 1**		**k = 2**		**k = 5**		**k = 10**		**k = 20**		**k = 50**	
Portfolio Type	**Long**	**Short**	**Long**	**Short**	**Long**	**Short**	**Long**	**Short**	**Long**	**Short**	**Long**	**Short**
GRU	0.601	0.632	0.583	0.636	0.569	0.622	0.555	0.604	0.550	0.573	0.535	0.535
LSTM	0.604	0.631	0.585	0.627	0.569	0.612	0.566	0.598	0.560	0.571	0.541	0.541
TCN	0.579	0.637	0.572	0.617	0.561	0.609	0.556	0.592	0.557	0.570	0.536	0.536
GBC	0.585	0.635	0.583	0.633	0.567	0.603	0.564	0.591	0.556	0.573	0.535	0.536
RF	0.597	0.623	0.578	0.634	0.570	0.615	0.569	0.598	0.564	0.581	0.542	0.543
LR	0.591	0.621	0.561	0.623	0.549	0.601	0.544	0.588	0.540	0.563	0.529	0.529

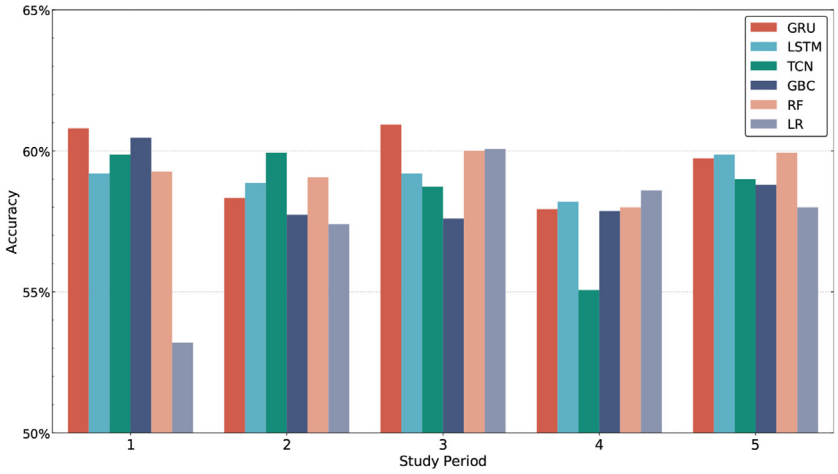

Fig. 5. Accuracy of $k = 5$ long-short portfolio prediction by study period.

the GRU. The tree-based ensemble models (GBC and RF) follow with 61.0% and the TCN achieves 60.8%, performing only slightly better than the LR (60.6%).

When considering the individual class predictions, a divergence between the performance of the long and short legs is evident for all $k < 50$. The accuracy within the predicted classes corresponds to the respective class precision (positive and negative predictive value). Note that for $k = 50$, the long and short leg performances are equal by construction. A slight deviation in the calculated accuracies is due to numerical limitations. For all models, the respective precision for the short positions is considerably higher than for the long positions (see Table 4). At the same time, the monotonicity of higher class precision for a lower value of k persists with the notable exception of the TCN model, which achieves a slightly lower long-only precision for $k = 10$ at 55.59% than for $k = 20$ at 55.65%. On the short side, the monotonicity holds up to the $k = 2$ and $k = 1$ portfolios, where precision decreases for the GRU, RF, and LR models.

4.1.2. Individual study periods

In this section, we analyze the models' accuracy for the $k = 5$ portfolio over time. Fig. 5 depicts how the achieved accuracies vary across study periods. The ranking between the models is not constant and does not follow a clear pattern, but all accuracies for each period are significantly larger than 50% based on the binomial test (see Appendix B.3 for an overview of all p-values).

The GRU performs best in the first study period with an accuracy of 60.8% (primarily due to its high short leg precision of 62.8%, which exceeds the TCN's precision of 61.5%), followed by the GBC with 60.5%. Compared to the overall accuracies across all periods, the LR model's individual period performance exhibits the largest variability. The TCN, which performs worst among the three deep learning models (GRU, LSTM, and TCN) overall for $k = 5$, achieves the highest long-short accuracy in the second study period with 59.9%, compared to the 58.9% of the LSTM and the GRU's 58.3%. This is mainly due to its superior long leg precision of 58.3% in that period (see Table 5).

Table 5
Prediction accuracy for long and short legs of $k = 5$ portfolio for individual study periods.

Study Period	**1**		**2**		**3**		**4**		**5**	
Portfolio Type	**Long**	**Short**	**Long**	**Short**	**Long**	**Short**	**Long**	**Short**	**Long**	**Short**
GRU	0.588	0.628	0.557	0.609	0.597	0.621	0.551	0.608	0.551	0.644
LSTM	0.575	0.609	0.553	0.624	0.591	0.593	0.567	0.597	0.561	0.636
TCN	0.583	0.615	0.583	0.616	0.581	0.593	0.485	0.616	0.575	0.605
GBC	0.605	0.604	0.576	0.579	0.545	0.607	0.553	0.604	0.555	0.621
RF	0.576	0.609	0.577	0.604	0.584	0.616	0.564	0.596	0.551	0.648
LR	0.483	0.581	0.565	0.583	0.597	0.604	0.575	0.597	0.523	0.637

Fig. 6. Cumulative market returns and Bitcoin (BTC) returns for the full test set date range with starting value 1 at the beginning of the test set of study period 1 (2020/04/26).

In the third study period, the GRU again achieves the highest long-short accuracy again with 60.9%, as its short-only precision of 62.1% is considerably higher than that of all other models in this period (see 5). Interestingly, the LR model (60.1%) has the second best performance of all models, with a slight edge over the RF (60.0%). The TCN (58.7%) performs worse than the LSTM (59.2%), but manages to outperform the GBC, which performs worst with 57.6%. The fourth study period sees the TCN's long-short accuracy drop to only 55.1%, as its long leg performance declines to a mere 48.5%. At the same time, it exhibits the highest short leg accuracy of all models and catches up again with the other models in study period 5, where all models score within a relatively narrow range between 58.0% (LR) and 59.9% (both LSTM and RF). while showing the highest short leg accuracy among all models, before it catches up again in study period 5 by achieving 59%, where all models score within a relatively narrow margin between 58.0% (LR) and 59.9% (both LSTM and RF).

In summary, the LR's relative disadvantage with regard to overall accuracy is due to its sub-par performance in period 1, while the overall best GRU model performs competitively across all study periods and outperforms all other models in periods 1 and 3. The same is true for the second and third best overall models, the RF and the LSTM, which also score competitively in all study periods and narrowly outperform all other models in study period 5.

4.2. Trading results

This section presents the financial performance results of the employed long-short trading strategy as well as the long side of the strategy. The financial performance is analyzed along the dimensions of return, risk, and risk-return metrics for the $k = 5$ portfolio, a diversified portfolio with five long and five short positions. We restrict the financial analysis to this portfolio, which includes 10% of the market constituents, similar to Fischer et al,[8] who include 15% of the respective market in their portfolio.

The equally-weighted market portfolio (consisting of the 100 eligible coins that constitute the asset universe for each study period) serves as the natural benchmark for the tested trading strategies. Fig. 6 shows the performance of the market index over all study periods as well as the performance of Bitcoin over the same time period and illustrates their high correlation. Furthermore, Appendices A.1 to A.5 show the performance indices for the $k = 5$ portfolios detailed over the different algorithms and study periods. Appendix B.5 gives the number of occurrences of individual coins in the long and short legs of the resulting portfolios.

Table 6
Daily return, risk, and annualized risk-return metrics for all models and the market (MKT) for the $k = 5$ long-short portfolio.

	GRU	LSTM	TCN	GBC	RF	LR	MKT
Mean Return	0.01348	0.01436	0.00308	0.00991	0.01098	0.01176	0.00330
Return Stdv.	0.08254	0.08504	0.07830	0.07615	0.07683	0.08212	0.04729
Downside Risk	0.83468	0.78989	0.81382	0.77421	0.76958	0.76237	0.54851
VaR 1%	−0.23565	−0.20450	−0.20177	−0.22286	−0.19656	−0.19230	−0.14187
VaR 5%	−0.10640	−0.10520	−0.11342	−0.09984	−0.09976	−0.10364	−0.07389
CVaR 1%	−0.32224	−0.30404	−0.28801	−0.28829	−0.28309	−0.28156	−0.19054
CVaR 5%	−0.18979	−0.17214	−0.17799	−0.17402	−0.16961	−0.16893	−0.11744
Ann. Volatility	1.57702	1.62474	1.49593	1.45486	1.46775	1.56897	0.90338
Sharpe Ratio	3.12061	3.22663	0.75165	2.48733	2.73153	2.73590	1.33105
Sortino Ratio	4.89560	5.51085	1.14722	3.88098	4.32569	4.67511	1.82023
Excess Sharpe	0.10596	0.11276	−0.00213	0.07226	0.08315	0.08729	–

4.2.1. Daily returns

For portfolio size $k = 5$, the two recurrent neural network architectures (LSTM and GRU) achieve the highest daily returns (with 1.44% and 1.35%, respectively). Table 6 shows the daily return metrics for all models. Notably, the simplest model, the logistic regression, outperforms the tree-based models (RF and GBC) with a daily return of 1.18% (compared to 1.10% and 0.99%, respectively), and performs only slightly worse than the LSTM and GRU models. The TCN performs worst among all models, earning daily returns of only 0.31%, making it the only model to underperform the general market (0.33%).

In terms of risk as measured by the returns' standard deviation, all models exhibit a considerably higher volatility than the overall market (0.0473). The LSTM's strategy returns exhibit the largest standard deviation (0.0850), closely followed by the GRU (0.0825) and LR (0.0821). The TCN model (0.0783) as well as the RF (0.0768) and GBC (0.0762) have a slightly lower volatility. When only taking into account the downside risk of returns, measured by the downside standard deviation, the GRU exhibits the highest risk at 0.8347, followed by the TCN (0.8138) and the LSTM (0.7899). As is the case with the standard deviation, all three deep learning models have slightly higher downside deviations than the tree-based ensemble models (GBC: 0.7742, RF: 0.7696) and considerably higher values than the overall market (0.5485).

Quantifying the financial risk in terms of the value-at-risk at 1%, the GRU (−23.56%) and the GBC (−22.29%) perform slightly worse than the rest of the tested models, which range between −20.45% (LSTM) and −19.23% (LR).

Considering only the long leg of the portfolio results in similarly profitable daily returns for all models with the TCN model also lagging behind the other models (see Table 7). Compared to the full long-short portfolio, the long leg returns are slightly lower for the GRU and LSTM models, while the TCN and GBC have higher mean returns. For the RF, the long leg of the portfolio generates the same mean returns as when the short leg is included in the portfolio. This implies a positive contribution of the short leg of the long-short portfolio for the GRU and LSTM and a negative

Table 7
Daily return, risk, and annualized risk-return metrics for all models and the market (MKT) for the long leg of the $k = 5$ portfolio.

	GRU	LSTM	TCN	GBC	RF	LR	MKT
Mean Return	0.01131	0.01371	0.00529	0.01108	0.01097	0.01163	0.00330
Return Stdv.	0.07259	0.07445	0.05621	0.06962	0.06851	0.07398	0.04729
Downside Risk	0.62922	0.57016	0.44772	0.59317	0.59060	0.64008	0.54851
VaR 1%	−0.15797	−0.13844	−0.12434	−0.15598	−0.14938	−0.16315	−0.14187
VaR 5%	−0.08319	−0.08026	−0.05743	−0.07369	−0.07086	−0.08074	−0.07389
CVaR 1%	−0.23227	−0.19367	−0.17349	−0.22397	−0.22932	−0.23833	−0.19054
CVaR 5%	−0.13600	−0.12351	−0.09442	−0.12579	−0.12429	−0.13877	−0.11744
Ann. Volatility	1.38691	1.42229	1.07397	1.33012	1.30886	1.41336	0.90338
Sharpe Ratio	2.97663	3.51991	1.79749	3.04240	3.05875	3.00373	1.33105
Sortino Ratio	5.44768	7.29051	3.58004	5.66458	5.62834	5.50700	1.82023
Excess Sharpe	0.14801	0.18077	0.04223	0.15336	0.14865	0.14177	–

mean short-only returns for the TCN and GBC. Both the RF and the LR have negligible positive contributions on the short side.

With regard to the long leg's return distribution, we observe a lower standard deviation than for the long-short portfolio for all models, indicating a detrimental contribution of the short leg in terms of added risk. The same is true when only downside risk is considered. Here, the relative advantage of the long leg of the portfolio is even more pronounced. The lower downside risk for the long leg is most notable for the TCN, where the downside risk decreases from 0.8138 to 0.4477 when the short leg is not included.

4.2.2. Risk-return characteristics

Considering the realized excess returns in relation to the incurred risk, the ranking is the same as for mean daily returns for the $k = 5$ long-short portfolio strategy, with the LSTM and GRU performing best with annualized Sharpe ratios of 3.23 and 3.12, respectively. Only the TCN (0.75) performs worse than the general market (1.33), while all other models perform considerably better. The substantial advantage of the RF over the LR in terms of accuracy (59.3% vs. 57.5%) notably does not translate into a more favorable Sharpe ratio, as the LR has a slight edge over with a ratio of 2.736 compared to the RF's 2.732.

The overall ranking in terms of risk-return performance remains unchanged when using the Sortino ratio, which only considers downside deviation for quantifying investment risk. The LSTM leads with a ratio of 5.51 ahead of the GRU with 4.90, while the TCN falls behind all other models and the general market (1.82) with a Sortino ratio of 1.15.

On the long-only side, all models except the GRU perform better in terms of the Sharpe ratio compared with the long-short portfolio performance, whereas the GRU drops from 3.12 to 2.98. The LSTM's long side of the portfolio strategy performs considerably better than all other models with a Sharpe ratio of 3.52. All other models, except the TCN, exhibit Sharpe ratios close to 3. The RF performs slightly better than GBC (3.04) at 3.06, while the GRU (2.98) underperforms the LR model (3.00). The TCN lags far behind with a ratio of only 1.80, but still manages to outperform the market (1.33).

As is the case for the full long-short portfolio, the LSTM achieves the highest long-only Sortino ratio with a value of 7.29, well ahead of the next best models, the GBC (5.66), RF (5.63), LR (5.51), and the GRU (5.45). However, all models have a higher Sortino ratio for the long side of the portfolio compared to the portfolio as a whole. Compared to the other models, the TCN model performs considerably worse with only 3.58, but has the largest relative difference compared with the long-short portfolio of more than 100%.

5. Discussion

This study demonstrates the potential of machine learning for cryptocurrency market prediction, as all utilized models significantly outperform a random classifier. Our analysis indicates that the employed recurrent neural networks, the temporal convolutional network, and tree-based ensembles are particularly effective in correctly classifying the relative daily performance of cryptocurrencies. Comparing the long and short leg predictions indicates that short legs are more predictable, as we generally document a slightly higher accuracy for short leg predictions. A potential explanation for this finding might be that investors cognitively process financial gains and losses differently.[33] This asymmetry may lead to investor behavior being more predictable during a market downturn due to an increased herding behavior induced by loss aversion. Behavioral biases might be more pronounced for the cryptocurrency market, as cryptocurrencies do not exhibit a fundamental value in the traditional sense.

In evaluating the economic implications of these predictions, we examine the performance of long-short portfolios that trade 10% of all market constituents. The higher overall portfolio risk of the resulting long-short portfolios may be driven by a lower degree of portfolio diversification compared to the market portfolio. However, as the portfolio returns of five of the six employed machine learning models yield positive returns of at least three times the market portfolio return, the long-short portfolios based on these models outperform the buy-and-hold market portfolio on a risk-adjusted basis. The temporal convolutional neural network is a noteworthy exception, as its high relative accuracy does not directly translate into a high prediction performance. This finding indicates that the temporal convolutional neural network is more confident in classifying observations with lower corresponding absolute returns than the other models.

The GRU and LSTM models appear especially well-suited for the employed trading strategy, as the long-short portfolios based on these models' predictions yield the highest risk-adjusted performance. Overall, these results indicate a challenge to weak form cryptocurrency market efficiency,[34] albeit the influence of certain limits to arbitrage cannot be entirely ruled out.

Furthermore, the risk-adjusted outperformance of the employed long-short strategies may have been reduced by an inflation of the buy-and-hold market benchmark. The potential inflation stems from the overall market upturn in the out-of-sample periods, as the buy-and-hold market strategy is exposed to the long-side of the cryptocurrency market. At the same time, the long-short portfolios exhibit zero net exposure. Also, due to the overall market upturn, the positive performance of the long-short portfolios is primarily driven by the long-side of the long-short portfolios, despite the slightly higher short leg classification accuracy. An exception is study period five, characterized by an overall market downturn. In that specific period, the short leg of the portfolio is responsible for the overall outperformance of the best-performing long-short portfolios.

This study contributes to research in cryptocurrency market prediction by providing a comparative analysis of different machine learning ensembles for predicting market movements and presenting a concrete trading approach. To the best of our knowledge, temporal convolutional neural networks have yet to be applied to the domain of cryptocurrency pricing before. Furthermore, we closely examine and highlight the potential of the different machine learning-based portfolio strategies in heterogeneous market environments.

The presented results are subject to several limitations and assumptions. First, we assume to be able to, on average, buy and sell cryptocurrencies at mid-price. Second, we assume to be able to short-sell the considered cryptocurrencies. Short-selling generally induces additional costs and is not consistently possible for all included cryptocurrencies. In this study, the risk-adjusted outperformance of the employed portfolio strategy compared to the market benchmark is predominantly driven by the long portfolio legs. Therefore, this limitation would be more pronounced in more neutral market environments. Third, as with every empirical study, this study is limited by its finite sample size. Fourth, the applied long-short strategy may be exposed to cryptocurrency risk factors[35] that may not have been reflected in the applied risk metrics. Finally, the external validity of the results may be limited by the use of cryptocurrency price data aggregated over multiple exchanges. Regarding the latter potential limitation, Vidal-Tomás[19] finds that the use of aggregated cryptocurrency exhibits the same processes as data from individual exchanges. He infers that aggregated cryptocurrency data is appropriate to utilize in research.

Based on the results of this study, there are several pathways for future research. First, it remains an open question how cryptocurrency market predictability and market mechanisms change over time. Second, future research may examine the predictive power of different predictive feature groups in the cryptocurrency market. In combination with using more feature groups, researchers may also apply future insights from the field of explainable artificial intelligence to cryptocurrency market prediction to enhance model transparency. Third, future research may examine the source of the predictive power of technical features in behavioral experiments to shed further light on cryptocurrency investing.

6. Conclusion

In this study, we employ several machine learning models to predict the relative daily market movements of the 100 largest cryptocurrencies by market capitalization. We show that all employed models make statistically viable predictions, whereby the average accuracy values calculated on all cryptocurrencies range from 52.9% to 54.1% for the different models. Accuracy values range from 57.5% to 59.5% when calculated on the subset of predictions with the 10% highest model confidences per class per day. A long-short portfolio strategy based on the predictions of the employed LSTM and GRU ensemble models yields an annualized out-of-sample Sharpe ratio after transaction costs of 3.23 and 3.12, respectively. In comparison, the buy-and-hold benchmark market portfolio strategy only yields a Sharpe ratio of 1.33. These results indicate a challenge to weak form cryptocurrency market efficiency, albeit the influence of certain limits to arbitrage cannot be entirely ruled out.

Acknowledgements

We acknowledge support by the KIT-Publication Fund of the Karlsruhe Institute of Technology.

Declaration of competing interest

None.

Appendix

Appendix A. Supplemental Figures

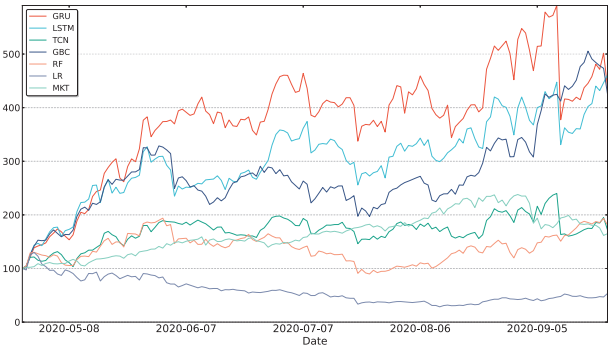

Figure A.1. Performance Index for Study Period 1 for the k = 5 portfolio.

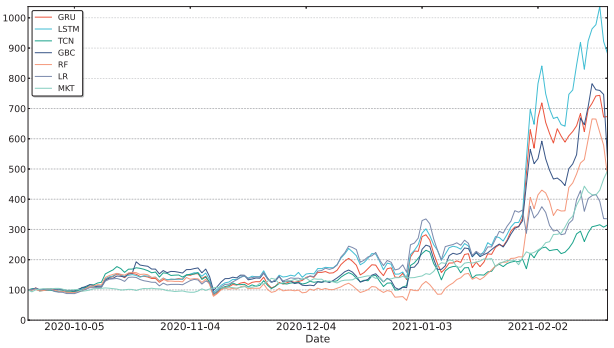

Figure A.2. Performance Index for Study Period 2 for the k = 5 portfolio.

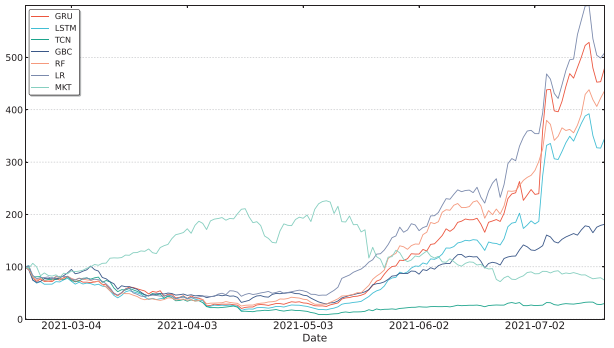

Figure A.3. Performance Index for Study Period 3 for the k = 5 portfolio.

Figure A.4. Performance Index for Study Period 4 for the k = 5 portfolio.

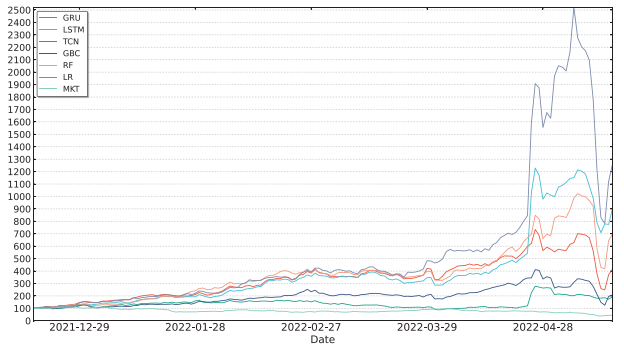

Figure A.5. Performance Index for Study Period 5 for the k = 5 portfolio.

Appendix B. Supplemental Tables

Table B.1
Overview of coins excluded from the study and the reason for their exclusion.

No	Coin Symbol	Exclusion Reason
1	BLUNA	Data issues
2	KNCL	Data issues
3	CDAI	Data issues
4	LN	Data issues
5	SOLVE	Data issues
6	VERI	Data issues
7	VEE	Data issues
8	JASMY	Data issues
9	MSOL	Data issues
10	MAID	Data issues
11	BUSD	Stablecoin
12	HUSD	Stablecoin
13	SAI	Stablecoin
14	DAI	Stablecoin
15	TUSD	Stablecoin
16	USDC	Stablecoin
17	USDT	Stablecoin
18	UST	Stablecoin
19	FRAX	Stablecoin
20	MIM	Stablecoin
21	MUSD	Stablecoin
22	SUSD	Stablecoin
23	USDN	Stablecoin

Table B.2
Complete p-values for the binomial test of achieving the respective model's accuracies for the combined test sets of all study periods with the null hypothesis of each model having atrue predictive performance of 0.5 or below.

Portfolio Size	k = 1	k = 2	k = 5	k = 10	k = 20	k = 50
GRU	7.012528e-20	1.048364e-33	7.864622e-62	7.823276e-85	1.137218e-100	2.136124e-84
LSTM	4.334874e-20	1.334199e-31	4.308048e-56	8.736139e-90	9.042803e-115	3.202961e-110
TCN	2.829318e-17	1.483387e-25	9.661208e-50	8.459317e-74	3.218419e-106	3.765699e-88
GBC	7.385857e-18	1.505160e-32	1.925738e-49	5.630044e-81	1.079329e-111	1.047139e-84
RF	7.385857e-18	1.334199e-31	2.365775e-58	5.677780e-93	1.566451e-139	3.419382e-119
LR	1.057233e-16	3.115607e-24	1.693509e-38	5.755924e-59	4.059272e-71	6.528285e-56

Table B.3
Complete p-values per study period for the binomial test of realizing the respective prediction accuracy with the null hypothesis of each model having a true predictive performance of 0.5 or below.

	Portfolio Size	1	2	5	10	20	50
Model	**Study Period**						
GRU	1	1.966113e-09	8.143005e-10	2.829318e-17	8.593063e-21	4.129621e-27	5.916977e-23
	2	2.053107e-04	5.951555e-06	5.829624e-11	6.540142e-14	9.855201e-25	1.468607e-25
	3	5.147597e-05	9.767993e-08	1.813239e-17	1.306882e-18	1.149283e-16	1.406433e-11
	4	5.147597e-05	1.355567e-09	4.344051e-10	1.987192e-16	2.808584e-21	9.363543e-19
	5	2.294356e-03	1.518496e-07	2.403512e-14	1.998324e-23	1.325015e-19	3.519168e-15
LSTM	1	1.273132e-06	2.496326e-08	5.388406e-13	3.441800e-18	3.811074e-26	8.770927e-26
	2	2.053107e-04	1.793673e-05	3.445781e-12	2.301733e-17	3.811074e-26	1.414983e-36
	3	5.147597e-05	9.767993e-08	5.388406e-13	9.309605e-20	2.131927e-18	2.449832e-13
	4	5.147597e-05	8.143005e-10	1.610661e-10	1.466401e-16	2.353418e-27	2.909156e-25
	5	1.914557e-05	2.344697e-07	1.074886e-14	4.629964e-26	3.368743e-25	1.966950e-20
TCN	1	7.103290e-07	3.679503e-09	1.074886e-14	6.873145e-25	1.140469e-34	1.860338e-27
	2	3.160513e-05	2.759758e-06	7.158810e-15	6.864123e-15	4.259964e-29	1.040745e-28
	3	3.908212e-07	1.518496e-07	7.102294e-12	3.353233e-13	4.073274e-15	3.161564e-10
	4	2.156425e-02	1.233186e-02	4.772219e-05	9.130275e-10	3.453358e-12	1.125556e-11
	5	3.280082e-03	3.960555e-08	1.653800e-12	6.080850e-21	7.581627e-28	2.945825e-22
GBC	1	7.103290e-07	9.016950e-14	2.511002e-16	9.616070e-27	3.234463e-37	1.994906e-29
	2	3.171100e-04	7.123364e-05	1.143767e-09	1.955044e-13	1.286722e-24	1.546419e-26
	3	2.294356e-03	1.246133e-06	2.940061e-09	1.215689e-14	1.074106e-18	2.375169e-14
	4	2.250362e-06	2.241060e-09	6.014537e-10	1.987192e-16	5.379156e-19	3.092007e-15
	5	7.274396e-04	2.759758e-06	4.953697e-12	6.806688e-19	2.980659e-22	1.369435e-10
RF	1	5.147597e-05	8.290170e-07	3.687780e-13	6.579187e-24	4.078689e-39	4.881953e-36
	2	1.311896e-04	9.716225e-09	1.141086e-12	8.954819e-18	3.219094e-28	1.468607e-25
	3	1.584634e-03	2.759758e-06	4.754797e-15	4.162588e-23	7.330139e-30	9.167393e-21
	4	1.144208e-05	1.355567e-09	4.344051e-10	3.144490e-17	1.664621e-26	8.046203e-25
	5	1.914557e-05	3.596098e-07	7.158810e-15	2.532673e-19	5.769668e-25	2.281399e-21
LR	1	7.274396e-04	1.233186e-02	7.072222e-03	3.693249e-05	9.511710e-05	3.971647e-06
	2	1.080569e-03	1.246133e-06	7.378645e-09	1.618262e-11	1.714117e-21	9.816374e-24
	3	7.103290e-07	2.496326e-08	3.149451e-15	1.816706e-19	2.852275e-24	3.783289e-11
	4	6.745841e-06	3.679503e-09	1.448182e-11	5.845011e-17	8.535440e-19	1.837793e-15
	5	3.280082e-03	1.249970e-05	4.344051e-10	3.634647e-16	7.494437e-15	2.085115e-10

Table B.4
Daily return and risk metrics and annualized risk-return metrics for all models and the market (MKT) for the short leg of the k = 5 portfolio.

	GRU	LSTM	TCN	GBC	RF	LR	MKT
Mean Return	0.00217	0.00065	−0.00221	−0.00117	0.00002	0.00013	0.00330
Return Standard Deviation	0.07646	0.07612	0.07676	0.07285	0.07236	0.07025	0.04729
Downside Risk	0.90256	0.92223	0.93714	0.88138	0.85539	0.81862	0.54851
VaR 1%	−0.22028	−0.22320	−0.22297	−0.21930	−0.20235	−0.19868	−0.14187
VaR 5%	−0.13972	−0.13221	−0.13441	−0.12916	−0.12400	−0.11725	−0.07389
CVaR 1%	−0.28448	−0.30545	−0.31089	−0.27844	−0.26553	−0.24951	−0.19054
CVaR 5%	−0.19742	−0.19863	−0.19963	−0.18967	−0.18092	−0.17207	−0.11744
Annual. Volatility	1.46079	1.45425	1.46656	1.39176	1.38247	1.34212	0.90338
Sharpe Ratio	0.54182	0.16135	−0.55060	−0.30860	0.00309	0.03408	1.33105
Sortino Ratio	0.72814	0.21126	−0.71546	−0.40462	0.00414	0.04639	1.82023
Excess Sharpe	−0.00993	−0.02357	−0.04824	−0.04013	−0.02977	−0.02910	–

Table B.5
Number of daily long and short portfolio positions for each respective coin aggregated over all study periods for the k = 5 portfolio.

Symbol	# Long Positions						# Short Positions					
	GRU	LSTM	TCN	GBC	RF	LR	GRU	LSTM	TCN	GBC	RF	LR
ABT	7	4	3	5	5	3	12	12	15	3	12	14
ADA	30	32	20	32	30	29	14	16	25	12	2	7
AE	38	32	41	29	31	36	68	64	18	74	67	70
AGIX	10	11	12	12	7	14	16	14	4	15	5	13
AION	23	16	24	12	9	20	31	27	13	41	27	30
ALGO	15	13	12	18	12	11	13	16	4	8	13	11
ANT	20	24	31	26	23	22	24	16	0	23	24	28
ARDR	27	25	32	29	21	19	22	22	17	22	18	15
ARK	30	24	25	21	23	17	13	20	10	19	10	18
ASD	7	6	2	7	6	10	36	48	125	29	44	27
ATOM	33	35	31	33	37	31	27	28	3	14	15	24
BAT	27	25	25	16	16	15	7	5	19	22	12	8
BCD	34	30	39	37	39	28	41	39	24	52	59	45
BCH	10	9	10	10	8	6	10	8	47	26	30	12
BCN	45	41	32	27	20	21	101	118	118	64	85	94
BEAM	13	12	13	16	11	14	18	22	4	13	14	16
BNB	21	21	12	26	19	26	4	2	35	32	8	2
BNT	45	49	45	46	44	57	37	31	30	21	19	33
BSV	8	8	6	4	5	5	14	14	59	27	45	32
BTC	8	8	2	5	6	6	5	8	146	41	66	6
BTG	22	22	24	24	34	24	11	13	29	13	12	8
BTM	18	17	16	17	19	13	42	46	20	38	48	41
BTS	18	15	22	17	15	15	25	33	23	50	35	32
CEL	8	9	6	9	10	9	22	18	15	12	23	15
CHZ	8	4	6	4	5	7	1	1	0	5	1	1
CKB	1	0	2	3	2	2	7	11	1	5	8	3
CRO	24	22	16	34	29	27	10	6	17	5	10	7
CRPT	17	19	15	17	18	14	24	26	1	17	18	22
CTXC	6	4	4	2	2	3	6	5	5	19	6	8
CVC	29	34	47	33	41	39	14	11	4	17	4	24
DASH	19	14	15	14	12	11	8	11	10	17	19	14
DCR	27	26	21	38	27	37	39	46	47	23	41	30
DENT	22	27	19	47	30	62	12	8	9	37	23	5
DGB	33	34	39	39	33	37	36	39	17	21	19	29
DOGE	57	63	59	71	97	124	19	11	36	53	19	15
DRGN	27	29	31	30	34	26	43	47	21	25	21	43
ELA	41	37	48	26	29	29	56	59	40	48	57	52

(*continued on next page*)

Table B.5 (*continued*)

Symbol	# Long Positions						# Short Positions					
	GRU	LSTM	TCN	GBC	RF	LR	GRU	LSTM	TCN	GBC	RF	LR
ELF	36	37	34	26	36	40	33	25	18	29	11	32
ENJ	44	44	41	48	45	56	11	9	4	15	11	10
EOS	16	13	13	12	12	8	6	6	23	24	26	11
ETC	28	29	28	43	63	61	10	15	29	24	5	12
ETH	4	8	5	7	7	4	5	5	36	10	7	2
ETN	53	51	52	43	51	44	79	82	35	38	41	63
FIRO	26	23	25	22	21	13	39	38	14	34	33	32
FSN	13	17	10	20	18	22	19	16	6	3	0	12
FTM	36	40	32	48	48	50	16	12	0	7	5	19
FTT	13	14	3	5	14	8	8	5	6	5	3	4
FUN	30	33	28	36	30	45	38	37	39	33	37	39
GAS	24	22	22	31	29	17	13	10	13	23	14	13
GBYTE	17	12	16	9	13	16	38	41	26	36	28	43
GLM	36	33	29	21	33	35	17	23	19	31	20	21
GRIN	11	10	9	14	10	12	43	47	27	20	22	39
GT	5	6	0	0	1	9	9	6	44	17	14	2
GTO	14	18	11	10	8	13	19	16	8	15	15	19
GUSD	7	9	0	2	5	7	26	24	131	53	69	38
GXC	23	23	35	19	20	15	20	22	30	40	57	30
HBAR	5	4	3	2	4	3	0	2	1	3	0	2
HC	15	12	13	12	11	12	40	43	69	43	64	50
HNS	12	6	9	11	9	12	22	28	8	5	12	14
HOT	43	49	43	54	51	74	18	16	14	37	17	18
HT	20	18	6	12	8	17	39	48	132	70	114	34
ICX	28	25	18	30	21	17	23	32	12	27	23	17
IOST	31	33	31	33	33	40	13	15	15	16	8	15
IOTX	21	26	43	55	40	33	6	3	0	7	1	20
KAN	4	4	5	2	3	7	5	5	9	25	28	11
KCS	29	30	35	30	38	33	25	29	62	46	61	15
KIN	42	44	48	55	56	61	52	53	26	31	23	62
KMD	19	15	25	19	14	13	24	25	34	25	29	31
LEO	29	35	11	13	26	29	36	34	83	65	49	30
LINK	39	37	29	36	35	27	20	27	19	15	17	32
LOOMOLD	26	27	22	14	20	19	23	24	10	22	13	24
LRC	71	82	88	109	103	84	55	48	21	22	30	55
LSK	27	21	36	22	23	20	16	18	10	31	35	32
LTC	17	13	13	14	11	13	9	9	23	22	23	11
LUNA	54	70	57	57	65	79	37	20	1	26	17	32
MANA	58	70	59	86	81	81	12	14	11	20	9	17
MATIC	11	17	13	12	14	8	5	4	1	10	4	7
MIOTA	17	16	15	10	18	9	16	14	9	14	8	15
MITH	26	24	27	25	27	20	25	15	8	14	10	23
MKR	37	37	26	31	43	29	39	34	42	24	29	22
NAS	13	5	11	10	7	9	12	17	17	27	20	16
NEC	29	35	27	33	27	36	24	24	13	15	7	32
NEO	15	12	11	13	16	15	6	5	12	20	3	3
NEX	40	37	36	27	37	27	88	90	30	62	86	87
NEXO	36	34	24	35	27	48	41	36	37	21	16	16
NMR	10	10	8	6	11	9	2	2	21	8	4	7
NPXS	124	121	207	113	99	131	104	78	18	114	117	130
NRG	34	26	29	16	20	24	71	72	24	37	55	64
NULS	29	32	28	17	21	18	30	31	18	17	20	21
NXS	3	2	2	1	1	2	1	1	8	9	2	1
NXT	12	13	10	21	11	9	44	49	51	31	53	31
NXT	12	13	10	21	11	9	44	49	51	31	53	31
OKB	43	40	29	33	31	32	41	46	55	42	64	40
OMG	42	38	37	63	59	53	29	27	12	29	18	35

Table B.5 (*continued*)

Symbol	# Long Positions						# Short Positions					
	GRU	LSTM	TCN	GBC	RF	LR	GRU	LSTM	TCN	GBC	RF	LR
ONT	12	8	8	10	5	10	9	8	17	22	9	15
PART	25	24	29	20	17	18	25	26	5	30	18	36
PIVX	13	11	12	7	10	5	23	22	14	14	16	13
POLY	25	27	29	30	23	20	25	29	19	12	6	20
POWR	27	33	28	18	20	23	15	9	18	14	11	20
PPT	39	38	43	36	25	44	38	40	15	25	28	37
QASH	15	12	8	7	9	10	31	31	63	44	74	29
QKC	14	16	13	9	19	7	11	12	14	8	6	8
QNT	26	24	25	35	27	37	26	23	5	11	10	25
QTUM	34	27	29	24	19	20	10	8	11	23	6	10
RDD	87	83	110	74	57	68	130	118	22	91	87	132
REN	23	21	30	18	23	18	23	27	0	21	24	22
REP	19	19	23	19	18	22	16	10	28	16	17	34
RIF	1	1	1	2	1	7	6	10	60	9	6	3
RLC	43	41	52	48	40	38	42	41	10	20	13	29
RVN	19	25	20	40	36	44	10	10	14	42	24	11
SAN	10	7	9	4	1	10	12	10	14	15	7	13
SC	25	30	21	33	29	21	18	18	14	19	12	19
SNT	26	23	34	23	22	13	24	25	21	25	22	25
SNX	17	18	20	17	25	17	23	18	0	24	23	29
STEEM	27	31	33	16	28	23	15	15	20	26	27	21
STMX	9	11	9	13	15	15	12	15	3	8	9	14
STORJ	39	39	40	50	55	39	25	22	9	15	15	34
STRAX	39	37	31	24	24	31	39	34	33	27	23	42
STX	5	5	9	8	5	5	1	5	0	4	8	4
SXP	8	7	7	9	7	12	11	18	1	13	15	10
SYS	20	22	19	25	22	22	28	26	14	17	12	24
THETA	60	67	52	81	78	62	41	38	7	22	26	30
TOMO	7	8	7	10	8	6	19	19	4	11	18	11
TRX	24	30	26	22	26	18	9	12	61	27	11	10
UBT	20	17	29	22	21	20	38	38	0	29	41	44
USDK	4	3	0	0	0	0	21	26	143	32	59	16
VET	17	17	12	22	16	11	13	15	7	5	5	9
VSYS	8	8	6	7	7	11	17	17	30	16	29	11
WAN	20	18	24	16	12	16	22	25	13	14	8	16
WAVES	67	76	69	92	101	76	61	44	35	41	58	58
WAXP	39	41	35	44	48	39	38	40	31	21	37	29
WICC	5	5	6	2	3	5	8	5	5	13	8	6
WIN	10	11	8	13	10	10	4	5	0	4	4	8
WRX	4	3	8	6	2	4	16	17	9	5	12	10
WTC	17	16	24	9	5	5	19	27	13	23	24	28
XEM	24	27	22	38	43	34	22	21	17	37	37	38
XLM	24	20	20	28	29	17	9	8	25	16	7	12
XMR	25	26	9	5	11	24	19	19	50	32	36	20
XNO	31	28	39	27	38	26	22	22	16	18	9	28
XPX	39	36	39	25	16	27	60	58	16	49	41	67
XRP	39	41	35	32	56	26	21	20	36	43	35	23
XTZ	23	22	21	21	22	20	14	19	23	9	7	13
XVG	34	29	34	35	21	27	39	38	14	15	17	30
XYO	19	19	27	10	8	11	32	32	9	34	34	40
ZB	11	14	6	7	8	11	28	41	220	71	155	35
ZEC	32	31	24	18	29	24	30	24	22	31	15	27
ZEN	39	40	44	42	45	38	48	57	19	37	28	38
ZIL	58	65	66	89	77	61	24	18	14	11	8	32
ZRX	28	29	32	24	34	25	18	17	12	17	7	15

References

1. Nakamoto S. Bitcoin: a peer-to-peer electronic cash system. Working Paper URL: https://bitcoin.org/bitcoin.pdf; 2008.
2. Coinmarketcap. *Coinmarketcap*; 2022. URL: https://coinmarketcap.com/. Accessed July 21, 2022. https://coinmarketcap.com/.
3. Kristoufek L, Vosvrda M. Cryptocurrencies market efficiency ranking: not so straightforward. *Phys Stat Mech Appl*. 2019;531, 120853.
4. Le Tran V, Leirvik T. Efficiency in the markets of crypto-currencies. *Finance Res Lett*. 2020;35, 101382.
5. Kakinaka S, Umeno K. Cryptocurrency market efficiency in short-and long-term horizons during covid-19: an asymmetric multifractal analysis approach. *Finance Res Lett*. 2022;46, 102319.
6. Gu S, Kelly B, Xiu D. Empirical asset pricing via machine learning. *Rev Financ Stud*. 2020;33:2223–2273. https://doi.org/10.1093/rfs/hhaa009.
7. Huang JZ, Huang W, Ni J. Predicting bitcoin returns using high-dimensional technical indicators. *J Finance Data Sci*. 2019;5:140–155.
8. Fischer T, Krauss C, Deinert A. Statistical arbitrage in cryptocurrency markets. *J Risk Financ Manag*. 2019;12:31. https://doi.org/10.3390/jrfm12010031.
9. Jaquart P, Dann D, Weinhardt C. Short-term bitcoin market prediction via machine learning. *J Finance Data Sci*. 2021;7:45–66.
10. Fil M, Kristoufek L. Pairs trading in cryptocurrency markets. *IEEE Access*. 2020;8:172644–172651. https://doi.org/10.1109/ACCESS.2020.3024619.
11. Betancourt C, Chen WH. Reinforcement learning with self-attention networks for cryptocurrency trading. *Appl Sci*. 2021;11:7377. https://doi.org/10.3390/app11167377.
12. McNally S, Roche J, Caton S. Predicting the price of bitcoin using machine learning. In: *Proceedings of 2018 Euromicro International Conference on Parallel, Distributed, and Network-Based Processing*. 2018:339–343. https://doi.org/10.1109/PDP2018.2018.00060.
13. Dutta A, Kumar S, Basu M. A gated recurrent unit approach to bitcoin price prediction. *J Risk Financ Manag*. 2020;13. https://doi.org/10.3390/jrfm13020023. Article 23.
14. Chen Z, Li C, Sun W. Bitcoin price prediction using machine learning: an approach to sample dimension engineering. *J Comput Appl Math*. 2020;365. https://doi.org/10.1016/j.cam.2019.112395. Article 112395.
15. Alessandretti L, ElBahrawy A, Aiello LM, Baronchelli A. *Anticipating Cryptocurrency Prices Using Machine Learning Complexity*; 2018. https://doi.org/10.1155/2018/8983590.
16. Lahmiri S, Bekiros S. Cryptocurrency forecasting with deep learning chaotic neural networks. *Chaos, Solit Fractals*. 2019;118:35–40.
17. Fischer T, Krauss C. Deep learning with long short-term memory networks for financial market predictions. *Eur J Oper Res*. 2018;270:654–669. https://doi.org/10.1016/j.ejor.2017.11.054.
18. CoinGecko. *Methodology*; 2022. https://www.coingecko.com/en/methodology. Accessed July 21, 2022.
19. Vidal-Tomás D. Which cryptocurrency data sources should scholars use? *Int Rev Financ Anal*. 2022;81, 102061.
20. Board of Governors of the Federal Reserve System. 3-Month treasury bill secondary market rate. Discount Basis [DTB3]. URL: https://fred.stlouisfed.org/series/DTB3; 2022.
21. Harris CR, Millman KJ, van der Walt SJ, et al. Array programming with numpy. *Nature*. 2020;585:357–362. https://doi.org/10.1038/s41586-020-2649-2.
22. McKinney W. Data structures for statistical computing in python. In: *Proceedings of the 9th Python in Science Conference*. 2010:56–61. https://doi.org/10.25080/Majora-92bf1922-00a. SciPy.
23. Chollet F. keras https://keras.io/; 2015. Accessed July 21, 2022.
24. Abadi Martín, Agarwal Ashish, Paul Barham, et al. *Tensorflow: Large-Scale Machine Learning on Heterogeneous Systems*; 2015. https://www.tensorflow.org/. Accessed July 21, 2022. accessed.
25. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. *J Mach Learn Res*. 2011;12:2825–2830.
26. Takeuchi L, Lee YYA. *Applying deep learning to enhance momentum trading strategies in stocks*. 2013. Working Paper.
27. Krauss C, Do XA, Huck N. Deep neural networks, gradient-boosted trees, random forests: statistical arbitrage on the S&P 500. *Eur J Oper Res*. 2017;259:689–702.
28. Kingma DP, Ba J. *Proceedings of 3rd International Conference on Learning Representations URL*. In: *Adam: a method for stochastic optimization*; 2015. arXiv:1412.6980 http://arxiv.org/abs/1412.6980.
29. Hochreiter S, Schmidhuber J. Long short-term memory. *Neural Comput*. 1997;9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.
30. Bai S, Kolter JZ, Koltun V. *An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling*; 2018. https://arxiv.org/pdf/1803.01271. Accessed July 21, 2022.
31. Freund Y, Schapire RE. A decision-theoretic generalization of on-line learning and an application to boosting. *J Comput Syst Sci*. 1997;55:119–139. https://doi.org/10.1006/jcss.1997.1504.
32. Friedman JH. Greedy function approximation: a gradient boosting machine. *Ann Stat*. 2001;29:1189–1232. https://doi.org/10.1214/aos/1013203451.
33. Kahneman D, Tversky A. *Prospect Theory: An Analysis of Decision under*. vol. 47. 1979:263–292. Source: Econometrica http://www.jstor.org/stable/1914185.
34. Fama EF. Efficient capital markets : a review of theory and empirical work. *J Finance*. 1970;25:383–417. https://doi.org/10.2307/2325486.
35. Liu Y, Tsyvinski A, Wu X. Common risk factors in cryptocurrency. *J Finance*. 2022;77:1133–1177.