

Cryptocurrency Return Predictability: A Machine-Learning Analysis

Abstract

We investigate the out-of-sample predictability of daily cryptocurrency returns using modern machine-learning methods. We consider a large number of cryptocurrencies (41) and a rich set of predictors relating to a cryptocurrency's network value and activity, time-series momentum, technical signals, and investor attention and sentiment. Our results indicate that return predictability is an important feature of the cryptocurrency market: machine-learning methods significantly improve the statistical accuracy of cryptocurrency return forecasts and provide substantial economic value to an investor. We find that a diverse set of predictors contribute to cryptocurrency return predictability and that nonlinearities play a prominent role.

JEL classifications: C52, C53, G11, G12, G17

Keywords: Out-of-sample return prediction, Random forest, XGBoost, Deep neural network, Network value, Network activity, Time-series momentum, Investor attention, Shapley values

1. Introduction

Cryptocurrencies are an asset class with a relatively short and checkered history. Cryptocurrencies have experienced large swings in value, casting doubt on their ability to serve as new mediums of exchange, and their comparatively weak regulatory oversight raises concerns about market manipulation and fraud (e.g., Griffin and Shams 2020; Flitter and Yaffe-Bellany 2023). Nevertheless, cryptocurrencies and their underlying blockchain technologies have the potential to profoundly affect the financial system and central banking (Prasad 2021), and they are now an important asset class for investors (Harvey et al. forthcoming). Concomitant with the advent and growth of cryptocurrencies, a burgeoning literature investigates asset pricing in the cryptocurrency market along various dimensions, such as arbitrage across exchanges (e.g., Makarov and Schoar 2020), market equilibrium (e.g., Cong, Li, and Wang 2021; Biais et al. forthcoming), and empirical risk factors (e.g., Bhambhwani, Delikouras, and Korniotis 2021; Liu, Tsyvinski, and Wu 2022).

In this paper, we investigate the out-of-sample predictability of a large number of daily cryptocurrency returns using modern machine-learning methods. Return predictability is a leading topic in asset pricing, as it relates to, among other things, information processing in financial markets and the determinants of time-varying expected returns, and it has important implications for investors. A number of studies analyze out-of-sample cryptocurrency return prediction, including with machine-learning tools. For example, Huang, Huang, and Ni (2019), Detzel et al. (2021), and Gradojevic et al. (2023) use technical indicators to predict daily and weekly Bitcoin returns, while Cheah et al. (2020) and Chen et al. (2021) predict daily Bitcoin returns using a variety of predictors. We contribute to the literature by undertaking a comprehensive analysis of cryptocurrency return predictability. Our analysis simultaneously considers a large number of cryptocurrencies, a broad array of predictors, and a diverse set of machine-learning techniques. We examine both the statistical and economic significance of cryptocurrency return predictability and investigate the sources of return predictability.

We start with daily data for 48 cryptocurrencies from [CoinMetrics](#) for which prices and data for an extensive set of characteristics relating to network value and activity are available. After dropping “stable coins” and “wrapped tokens” that behave very differently from most cryptocurrencies, we are left with 41 cryptocurrencies. In addition to predictors based on network value and activity

(i.e., cryptocurrency “fundamentals”), we construct market-based predictors relating to time-series momentum (Moskowitz, Ooi, and Pedersen 2012) and price-based technical signals. To consider measures of investor attention and sentiment, we also download cryptocurrency search-volume data from [Google Trends](#) as well as [Reddit](#) comments and [Factiva](#) articles relating to cryptocurrencies. This provides us with a rich set of predictors from manifold categories.

We compute daily cryptocurrency return forecasts using various machine-learning methods. The first forecast is based on elastic net (ENet, Zou and Hastie 2005) estimation of a linear model that includes all of the predictors (linear-ENet forecast). Conventional ordinary least squares (OLS) estimation of a large-scale linear model with correlated predictors is highly susceptible to in-sample overfitting. To help guard against overfitting, the ENet employs penalized regression to shrink the coefficient estimates to zero. The other forecasts allow for general nonlinearities in the prediction model, thereby increasing the scope of the forecasts to capture potentially important nonlinearities in the data. We construct two forecasts based on decision trees: random forest (Breiman 2001) and [XGBoost](#) (Chen and Guestrin 2016). In addition, we use a deep neural network to forecast cryptocurrency returns. Random forests, XGBoost, and deep neural networks are powerful machine-learning techniques that perform well in a variety of domains. We investigate their ability to forecast a large number of daily cryptocurrency returns based on a wide array of predictors.¹ Because asset returns inherently contain a large unpredictable component, we are dealing with noisy data, which makes it especially important to guard against overfitting when training the machine-learning models. To help prevent overfitting, we tune multiple hyperparameters for each machine-learning model; we also retune the hyperparameters and retrain the models on a regular basis over time. We only use information available at the time of forecast formation when tuning the hyperparameters and training the models, so the forecasts do not entail “look-ahead” bias. In addition to forecasts based on the individual machine-learning methods, we consider ensemble forecasts that combine the individual machine-learning forecasts.

We find that daily cryptocurrency return predictability is empirically important, both statistically and economically. In terms of out-of-sample mean squared error (MSE), the machine-learning

¹Machine learning is growing in popularity in finance; for example, a spate of recent studies use machine-learning methods to forecast stock returns, including Chincio, Clark-Joseph, and Ye (2019), Freyberger, Neuhierl, and Weber (2020), Gu, Kelly, and Xiu (2020), Kozak, Nagel, and Santosh (2020), Dong et al. (2022), Avramov, Cheng, and Metzker (forthcoming), and Chen, Pelger, and Zhu (forthcoming).

forecasts outperform the prevailing mean benchmark forecast in the vast majority of cases for individual cryptocurrencies. The prevailing mean forecast assumes that returns are not predictable; given the intrinsically large unpredictable component in asset returns, it constitutes a stringent benchmark (e.g., Goyal and Welch 2008). The machine-learning forecasts also significantly outperform the prevailing mean benchmark in terms of MSE for all of the cryptocurrencies taken together. Furthermore, the machine-learning forecasts that allow for nonlinearities—random forest, XGBoost, and deep neural network—are significantly more accurate than the linear-ENet forecast, so accommodating nonlinearities in fitted prediction models improves out-of-sample cryptocurrency return prediction. Ensemble forecasts also perform well.

In addition to improving statistical accuracy, the machine-learning forecasts are economically valuable. We consider a mean-variance investor who allocates between an individual cryptocurrency and risk-free Treasury bills. The investor realizes substantial utility gains by relying on machine-learning forecasts to guide asset allocation compared to ignoring return predictability by relying on the prevailing mean benchmark forecast. We also construct long-short portfolios that invest in multiple cryptocurrencies guided by the machine-learning forecasts of daily cryptocurrency returns. Specifically, we sort the available cryptocurrencies based on their return forecasts; the portfolio goes long (short) the cryptocurrencies with the highest (lowest) return forecasts. The long-short portfolios exhibit impressive performance in terms of Sharpe, Calmar, and Sortino ratios, and they typically perform substantially better than passively holding a value-weighted market portfolio of cryptocurrencies (as well as a value-weighted equity market portfolio). The long-short portfolios based on the machine-learning forecasts also generate statistically significant and economically sizable alphas in the context of the Liu, Tsyvinski, and Wu (2022) cryptocurrency three-factor model. With regard to economic value, we again find that nonlinearities are important, as portfolios based on the nonlinear forecasts usually generate larger economic gains than those based on the linear-ENet forecast.

To glean insight into the sources of cryptocurrency return predictability, we use Shapley (1953) values to interpret the fitted machine-learning models that generate the return forecasts. Shapley-based variable-importance measures indicate that predictors from a variety of categories are relevant for forecasting daily cryptocurrency returns. Among the most important predictors are time-series momentum, network value, the address-to-network value ratio, the number of transactions, and

Google Trends search volume. Investor attention to cryptocurrencies, as captured by the number of Reddit comments and Factiva articles, appears more important than sentiment and uncertainty in the comments and articles. Overall, we find that cryptocurrency return predictability emanates from a diversity of sources. Plots of Shapley values reveal important nonlinearities in marginal predictive relationships for the fitted models, further highlighting the relevance of nonlinearities for forecasting cryptocurrency returns.

The remainder of the paper is organized as follows. Section 2 presents the data. Section 3 describes the machine-learning methods that we use to forecast cryptocurrency returns. Section 4 reports the out-of-sample results, focusing on statistic accuracy. Section 5 analyzes the economic value of the cryptocurrency return forecasts. Section 6 uses Shapley values to investigate the sources of cryptocurrency return predictability. Section 7 concludes.

2. Data

We use daily cryptocurrency prices in US dollars from CoinMetrics, a leading provider of cryptocurrency data. We begin with 48 cryptocurrencies for which prices and data for a large number of characteristics are available. Five of the cryptocurrencies are “stable coins” that are at least putatively pegged to the US dollar. Two more cryptocurrencies are “wrapped tokens” that are designed to trade 1:1 with Bitcoin. Because they behave quite differently from other cryptocurrencies, we drop the five stablecoins. We also drop the two wrapped tokens, as they follow Bitcoin. This leaves 41 cryptocurrencies.

CoinMetrics provides US dollar closing prices (“reference rates”) at midnight coordinated universal time for every day of the year. Each price is a volume-weighted average of prices across exchanges that qualify as “constituent markets” according to criteria established by CoinMetrics.² We compute the day- t return as the proportional change in the closing price from day $t - 1$ to t and compute the excess return using the daily risk-free return from CRSP.³ The cryptocurrencies come into existence at different dates, so data for different cryptocurrencies start at different dates. Our data extend through the end of 2021.

²The list of constituent markets varies across cryptocurrencies.

³CRSP does not provide risk-free return observations for weekends and holidays. We fill in each missing value with the previous day’s value. Our out-of-sample forecasting results are similar if we use the simple return instead of the excess return.

Table 1: Summary Statistics

The table reports summary statistics for daily excess returns for the cryptocurrency in the first column. Cryptocurrency returns are computed using US dollar daily closing prices from [CoinMetrics](#). The market capitalization in the fourth column is for the end of the sample (2021-12-31). The statistics in the fifth through tenth columns are computed after omitting the first 28 days of data.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Cryptocurrency	Sample start	Obs.	Market cap. (\$B)	Ann. mean (%)	Ann. vol. (%)	Min. (%)	Max. (%)	Ann. Sharpe ratio	Auto-corr.
linch	2020-12-27	370	3.59	140.18	154.59	-35.75	42.66	0.91	-0.01
Aave	2020-10-11	447	4.10	251.38	141.91	-31.05	30.81	1.77	0.01
Algorand	2019-06-23	923	16.69	125.62	135.42	-46.72	48.00	0.93	-0.06
Balancer	2020-06-26	554	0.82	137.34	149.89	-29.21	58.25	0.92	-0.03
Basic Attention Token	2017-10-07	1,547	1.82	137.00	133.09	-42.14	46.61	1.03	-0.09
Bitcoin	2011-01-01	4,018	876.86	152.45	96.30	-48.57	48.57	1.58	0.02
Bitcoin Cash	2017-08-02	1,613	8.17	77.74	131.19	-42.93	51.05	0.59	0.02
Bitcoin SV	2018-11-16	1,142	2.30	113.11	153.88	-43.81	144.08	0.74	-0.08
Cardano	2017-12-02	1,491	43.17	91.85	118.99	-38.89	34.00	0.77	-0.04
Chainlink	2017-09-30	1,554	19.62	207.95	143.81	-47.06	60.79	1.45	-0.05
Compound	2020-06-19	561	2.00	103.24	131.11	-25.72	23.05	0.79	-0.02
Crypto.com Coin	2019-03-21	1,017	55.95	133.21	113.52	-37.84	58.59	1.17	-0.09
Curve DAO Token	2020-08-16	503	8.97	245.86	188.96	-41.04	54.24	1.30	-0.03
Dash	2014-02-09	2,883	1.41	150.58	136.90	-37.78	107.00	1.10	-0.01
Decentraland	2017-08-26	1,589	7.19	334.39	245.93	-67.52	299.28	1.36	0.03
Decred	2016-05-18	2,054	0.96	155.39	135.59	-38.90	56.94	1.15	-0.11
Dogecoin	2014-01-24	2,899	22.62	177.98	181.45	-40.17	308.39	0.98	0.06
Ethereum	2015-08-09	2,337	433.79	193.48	117.34	-43.20	35.07	1.65	-0.04
Ethereum Classic	2016-07-26	1,985	4.52	136.98	129.93	-39.80	74.66	1.05	-0.01
FTX Token	2019-08-21	864	12.85	190.45	92.76	-24.99	32.21	2.05	-0.03
Gnosis	2017-05-03	1,704	5.31	94.52	126.45	-35.77	64.44	0.75	-0.03

Table 1 reports summary statistics for the 41 cryptocurrency excess returns. Bitcoin is the oldest cryptocurrency, with a sample starting date of January 1, 2011, resulting in 4,018 daily observations. Litecoin is the next oldest, with a sample starting date of April 2, 2013 (3,196 observations). The youngest cryptocurrency in the sample is Internet Computer, which has a sample starting date of May 12, 2021 (234 days). The fourth column reports market capitalizations at the end of 2021. Two cryptocurrencies stand out in terms of market capitalization: Bitcoin (\$877B) and Ethereum (\$434B). The third largest market capitalization is for XRP (\$83B), which

Table 1 (continued)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)	(9)	(10)
Cryptocurrency	Sample start	Obs.	Market cap. (\$B)	Ann. mean (%)	Ann. vol. (%)	Min. (%)	Max. (%)	Ann. Sharpe ratio	Auto-corr.
HedgeTrade	2019-11-03	790	0.71	291.80	291.28	-40.24	258.78	1.00	-0.05
Huobi Token	2019-03-07	1,031	4.70	99.16	104.71	-38.70	60.96	0.95	-0.06
Internet Computer	2021-05-12	234	11.67	-102.69	151.67	-22.37	43.53	-0.68	0.05
Litecoin	2013-04-02	3,196	10.15	118.39	131.31	-42.64	110.30	0.90	0.01
Livepeer	2018-12-21	1,107	0.98	256.31	198.43	-53.37	139.58	1.29	-0.01
Maker	2017-12-27	1,466	2.31	88.92	124.99	-56.84	51.77	0.71	-0.13
Neo	2017-07-16	1,630	2.57	72.32	127.85	-38.77	45.75	0.57	-0.05
OMGNetwork	2017-07-16	1,630	0.83	94.08	139.88	-42.61	69.18	0.67	-0.05
Perpetual Protocol	2021-02-05	330	1.31	195.64	170.70	-25.37	39.46	1.15	-0.06
Polkadot	2020-08-21	498	30.21	218.37	137.93	-35.06	39.07	1.58	-0.03
Quant	2019-03-17	1,021	4.37	257.47	147.47	-42.70	53.63	1.75	-0.07
Stellar	2015-10-01	2,284	28.19	186.93	157.16	-33.96	101.07	1.19	0.10
SushiSwap	2020-09-02	486	2.26	305.43	171.79	-34.96	37.19	1.78	-0.09
Synthetix	2020-04-10	631	1.32	211.81	142.76	-34.18	25.62	1.48	-0.05
Tezos	2018-07-01	1,280	3.80	97.37	123.47	-44.25	32.88	0.79	-0.06
UMA	2020-09-09	479	0.97	144.20	161.04	-33.29	70.66	0.90	0.03
Uniswap	2020-09-19	469	17.09	244.94	146.43	-29.87	45.94	1.67	-0.08
XRP	2014-08-16	2,695	83.09	166.83	149.10	-47.09	174.22	1.12	-0.02
yearn.finance	2020-07-26	524	1.22	181.94	157.92	-34.51	46.65	1.15	0.05
Zcash	2016-10-30	1,889	1.76	87.99	123.96	-38.42	64.73	0.71	-0.01

is around ten and five times smaller than those for Bitcoin and Ethereum, respectively. The vast majority of the cryptocurrencies in the sample have market capitalizations above \$1B at the end of 2021; the exceptions are Balancer (\$0.82B), Decred (\$0.96B), HedgeTrade (\$0.71B), Livepeer (\$0.98B), OMGNetwork (\$0.83B), and UMA (\$0.97B).

The magnitudes of the annualized means and volatilities in the fifth and sixth columns, respectively, of Table 1 are much higher than those typically seen for assets such as equities, bonds, and currencies. The youngest cryptocurrency in the sample, Internet Computer, has an annualized average excess return of -102.69%. The average excess returns are positive over the sample for the remaining cryptocurrencies, ranging from 72.32% (Neo) to 334.39% (Decentraland). The average excess return is above 100% (200%) for 31 (11) of the cryptocurrencies. All but two of the annualized volatilities in the sixth column are above 100% and reach as high as 291.28% (HedgeTrade).

Despite the high volatilities, the high average excess returns typically translate into sizable annualized Sharpe ratios in the ninth column of Table 1. The Sharpe ratios are at least 0.57 for all of the cryptocurrencies, with the exception of Internet Computer (-0.68), and over half (23) have Sharpe ratios of one or more. The Sharpe ratios for the two oldest cryptocurrencies, Bitcoin and Ethereum, are 1.58 and 1.65, respectively. The minimum and maximum values in the seventh and eighth columns, respectively, clearly show that daily cryptocurrency returns are characterized by quite extreme values, while the autocorrelations in the last column are small in magnitude.

Figure 1 depicts log cumulative excess returns for the cryptocurrencies, with each cryptocurrency identified by its symbol in the panel heading.⁴ In line with the volatilities and minimum and maximum values in Table 1, the cryptocurrencies display large swings in value in Figure 1. There is a tendency for the swings to correlate across cryptocurrencies; for example, numerous cryptocurrencies evince large appreciations followed by prolonged depreciations in late 2017 and throughout 2018, while many experience sizable appreciations with the advent of COVID-19 in 2020.

We consider a broad array of predictors for forecasting cryptocurrency returns. The first set of predictors is based on data from CoinMetrics.⁵

Network value Supply of the cryptocurrency in US dollars at the end of that day (also known as “market capitalization”). It is computed using the daily closing price on that day.

Network value-to-transaction ratio Network value divided by the adjusted transfer value for that day. The adjusted transfer value is computed using the native units network value and adjusted transaction volume.

Active address-to-network value ratio Number of unique addresses that are active in the network (either as a destination or source of a ledger change) on that day divided by network value.

Address-to-network value ratio Number of unique addresses holding any amount of native units at the end of that day divided by network value.

⁴The symbols for the 41 cryptocurrencies are as follows: linch (1INCH), Aave (AAVE), Algorand (ALGO), Balancer (BAL), Basic Attention Token (BAT), Bitcoin (BTC), Bitcoin Cash (BCH), Bitcoin SV (BSV), Cardano (ADA), Chainlink (LINK), Compound (COMP), Crypto.com Coin (CRO), Curve DAO Token (CRV), Dash (DASH), Decentraland (MANA), Decred (DCR), Dogecoin (DOGE), Ethereum (ETH), Ethereum Classic (ETC), FTX Token (FTT), Gnosis (GNO), HedgeTrade (HEDG), Huobi Token (HT), Internet Computer (ICP), Litecoin (LTC), Livepeer (LPT), Maker (MKR), Neo (NEO), OMGNetwork (OMG), Perpetual Protocol (PERP), Polkadot (DOT), Quant (QNT), Stellar (XLM), SushiSwap (SUSHI), Synthetix (SNX), Tezos (XTZ), UMA (UMA), Uniswap (UNI), XRP (XRP), yearn.finance (YFI), Zcash (ZEC).

⁵See the documentation on the [CoinMetrics](#) website for more information on the predictors.

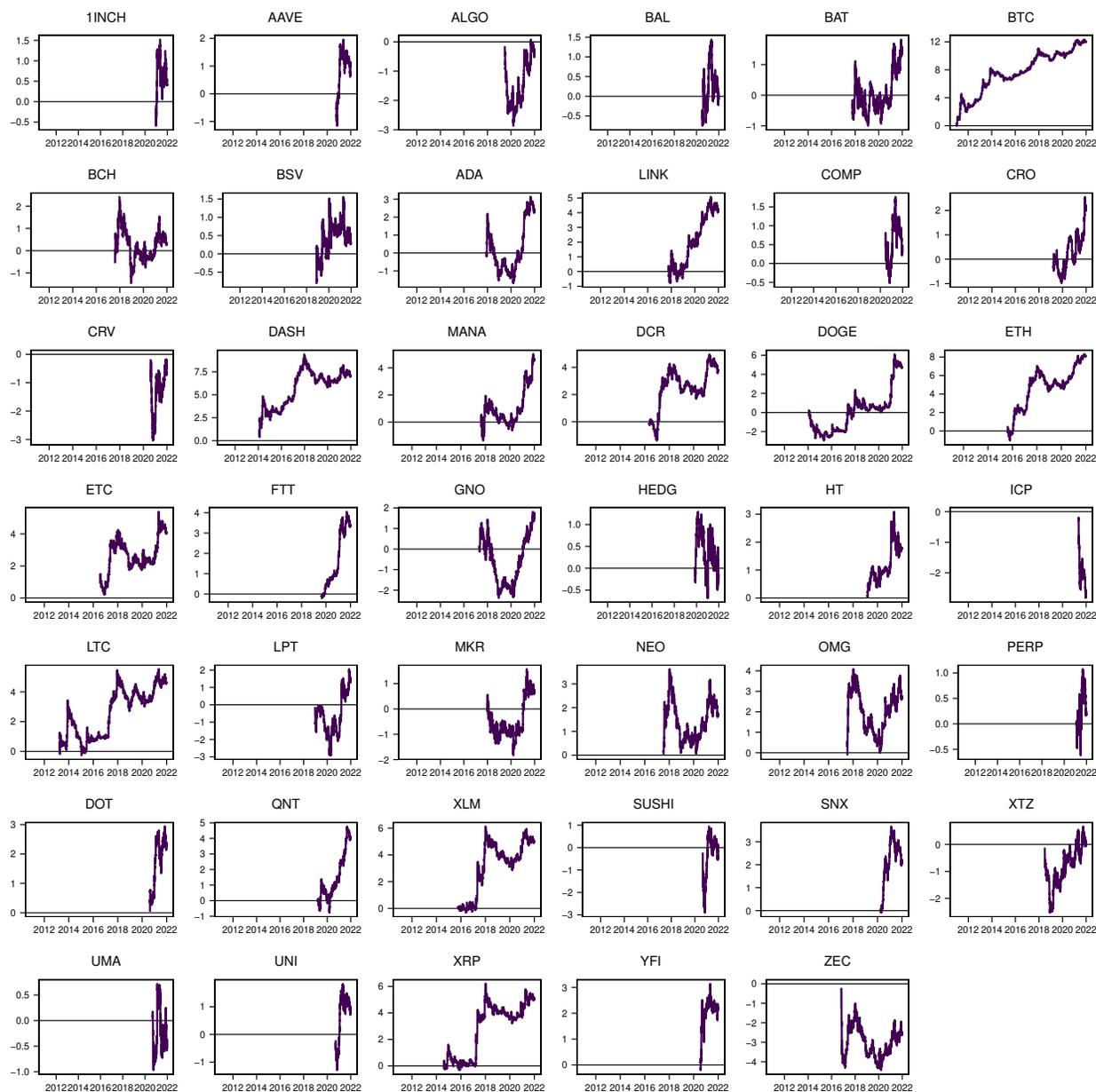


Figure 1: Log Cumulative Excess Returns

Each panel depicts the log cumulative excess return for the cryptocurrency (denoted by its symbol) in the panel heading.

Velocity Value transferred over the last year divided by the current supply on that day; a measure of turnover.

Number of transactions Number of transactions on that day. Transactions are counted whether or not they involve the transfer of native units from one ledger entity to another.

Number of transfers Number of transfers on that day. Transfers constitute movements of native units from one ledger entity to another.

Mean dollar size of transfers US dollar value of native units transferred divided by the number of transfers on that day.

Median dollar size of transfers Median US dollar value transferred per transfer on that day.

Active supply Number of unique native units that transacted at least once in the last 30 days.

Realized network value US dollar value of the supply of the cryptocurrency at the end of that day, based on the closing price on the day that a native unit last moved.

Current-to-realized ratio Network value divided by realized network value.

Supply equality ratio Supply of the cryptocurrency held by all addresses with less than one-ten-millionth of the current supply of native units divided by the supply held by the top one percent of addresses on that day.

Volatility Standard deviation of the log return over the last 30 days.

We use cumulative returns to measure time-series momentum (Moskowitz, Ooi, and Pedersen 2012) and prices to compute technical signals based on moving averages, which are popular with trend-following traders.

7-day time-series momentum 7-day cumulative excess return.

28-day time-series momentum 28-day cumulative excess return.

Price signal Price-based moving-average signal.

We also construct predictors designed to measure attention and sentiment via online activity and publications, beginning with Google Trends search volume.

Google Trends search volume Search-volume index for Google Trends searches involving the cryptocurrency's name or symbol for that day.

For a given query, the Google Trends search-volume index is computed based on the number of searches involving a term divided by the total number of searchers in a geographical region for a specific time period. The search-volume index provided by Google Trends is an integer between zero and 100, where the scale is determined using the smallest and largest values of the search ratios for the query’s time period. Because Google Trends places limits on the amount of data that can be downloaded for a given query, we need to download the search-volume index data in batches. Due to the scaling of the search-volume index by Google Trends, figures for the index are not comparable over time for different batches. To address this issue, we download daily data in batches from January to July and July to January and use data from the overlapping months to compute adjustment factors for splicing together series that are comparable over time.

In addition, we download Reddit comments that appear in the “Cryptocurrency” subreddit and include the cryptocurrency’s name or symbol. We combine the text for all of the comments for that day.

Number of Reddit comments Number of Reddit comments for that day.

Reddit sentiment score Number of positive words minus the number of negative words, all divided by the total number of words, for Reddit comments for that day. The positive and negative words are from the lists in Loughran and McDonald (2011).

Reddit uncertainty score Number of uncertainty words divided by the total number of words for Reddit comments for that day. The uncertainty words are from the list in Loughran and McDonald (2011).

Following standard practice, before calculating the sentiment and uncertainty scores, we eliminate stop words (e.g., “a,” “I,” “and,” “or”). We also tokenize the text by applying the Porter (1980) stemmer, which eliminates suffixes (e.g., “-tion”) from words to ensure that relevant words match those in the Loughran and McDonald (2011) lists.

Finally, we compute metrics based on articles from Factiva.⁶ Specifically, we download articles from the “Cryptocurrency” category that contain the cryptocurrency’s name or ticker. We combine the text for all of the articles for that day.

Number of Factiva articles Number of Factiva articles for that day.

Factiva article sentiment score Number of positive words minus the number of negative words, all divided by the total number of words, for words appearing in the bodies of Factiva articles for that day.

Factiva article uncertainty score Number of uncertainty words divided by the total number of words for words appearing in the bodies of Factiva articles for that day.

Factiva headline sentiment score Number of positive words minus the number of negative words, all divided by the total number of words, for words appearing in headlines of Factiva articles for that day.

Factiva headline uncertainty score Number of uncertainty words divided by the total number of words for words appearing in the headlines of Factiva articles for that day.

Table 2 lists the predictors, along with their abbreviations and transformations. We transform each predictor by computing log deviations (or, in some cases, simple deviations) from 7- and 28-day moving averages. This “feature engineering” helps to render the predictors stationary and provides sharper signals for forecasting cryptocurrency returns.⁷ We also include the 7- and 28-day time-series momentum in their levels (i.e., without transformation). After all of the transformations, we have a total of 54 predictors.

⁶Articles from Factiva are drawn from the following 47 publications from around the world: *The Cointelegraph*, *CoinDesk.com*, *Blockonomi*, *Dow Jones Newswires*, *express.co.uk* (UK), *PR Newswire*, *CE NoticiasFinancieras* (Latin America), *Investing.com*, *Financial Times*, *Reuters*, *iCrowdNewswire*, *The Wall Street Journal*, *M2 Presswire*, *The Independent*, *Blockchain.News*, *The Times* (UK), *Investor’s Business Daily* (US), *The Telegraph* (UK), *MarketWatch*, *Brave New Coin*, *Sputnik News Service* (Russia), *Benzinga.com*, *Mondaq Business Briefing*, *Business Insider*, *CNN*, *Forbes*, *Business Wire*, *City AM* (London), *South China Morning Post*, *GlobeNewswire* (US), *Investment Weekly News*, *The Economic Times*, *ACCESSWIRE*, *Postmedia Breaking News* (Canada), *Hedge Week*, *Daily Mail*, *The Australian*, *Financial News* (Europe), *Exchange News Direct*, *Korea Times* (South Korea), *The Globe and Mail*, *Agence France Presse*, *Institutional Asset Manager*, *The Canadian Press*, *Barron’s*, *Times of India*, *The New York Times*.

⁷The results are robust to the lengths of the moving averages; for example, log (or simple) deviations from 7- and 70-day moving averages produce similar results.

Table 2: Predictors

The table lists the predictors used to forecast daily cryptocurrency excess returns. The predictors are defined in the text. The third column gives the transformations applied to the predictors.

(1) Predictor	(2) Abbreviation	(3) Transformations
Panel A: Network		
Network value	NV	Log deviations from 7- and 28-day moving averages
Network value-to-transaction ratio	NVT	Log deviations from 7- and 28-day moving averages
Active address-to-network value ratio	AdrActNV	Log deviations from 7- and 28-day moving averages
Address-to-network value ratio	AdrNV	Log deviations from 7- and 28-day moving averages
Panel B: Transactions		
Velocity	Vel1y	Log deviations from 7- and 28-day moving averages
Number of transactions	NumTrx	Log deviations from 7- and 28-day moving averages
Number of transfers	NumTrf	Log deviations from 7- and 28-day moving averages
Mean dollar size of transfers	TrfSizeMean	Log deviations from 7- and 28-day moving averages
Median dollar size of transfers	TrfSizeMed	Log deviations from 7- and 28-day moving averages
Active supply	Sup30d	Log deviations from 7- and 28-day moving averages
Realized network value	RealNV	Log deviations from 7- and 28-day moving averages
Current-to-realized ratio	CRNV	Log deviations from 7- and 28-day moving averages
Supply equality ratio	SER	Log deviations from 7- and 28-day moving averages
Panel C: Market		
Volatility	Vol30d	Deviations from 7- and 28-day moving averages
7-day time-series momentum	TSM7	Levels, deviations from 7- and 28-day moving averages
28-day time-series momentum	TSM28	Levels, deviations from 7- and 28-day moving averages
Price signal	PrcMA	Log deviations from 7- and 28-day moving averages
Panel D: Online Activity		
Google Trends search volume	GTSrch	Log deviations from 7- and 28-day moving averages
Number of Reddit comments	RedNumCom	Log deviations from 7- and 28-day moving averages
Reddit sentiment score	RedSent	Log deviations from 7- and 28-day moving averages
Reddit uncertainty score	RedUnc	Log deviations from 7- and 28-day moving averages
Number of Factiva articles	FacNumArt	Log deviations from 7- and 28-day moving averages
Factiva article sentiment score	FacSentArt	Deviations from 7- and 28-day moving averages
Factiva article uncertainty score	FacUncArt	Deviations from 7- and 28-day moving averages
Factiva headline sentiment score	FacSentHdl	Deviations from 7- and 28-day moving averages
Factiva headline uncertainty score	FacUncHdl	Deviations from 7- and 28-day moving averages

To aid in the interpretation of fitted models in Section 6, Table 2 divides the predictors into four groups. The first group, “Network,” includes network value and valuation ratios relating to network value. These ratios can be viewed as cryptocurrency counterparts to popular equity valuation ratios (Liu, Tsyvinski, and Wu 2021), such as price-to-dividend and book-to-market ratios. The second group is “Transactions” and is comprised of a variety of variables relating to activity on a cryptocurrency’s network. The first two groups together can be viewed as constituting “fundamentals” for forecasting cryptocurrency returns. The next group, “Market,” is made up of return volatility, time-series momentum, and price-based technical signals in the form of moving-average rules. Liu and Tsyvinski (2021) report in-sample evidence of time-series momentum in cryptocurrency returns, while Huang, Huang, and Ni (2019), Detzel et al. (2021), and Gradojevic et al. (2023) find that technical indicators are useful for forecasting Bitcoin returns. Finally, the “Online Activity” group includes Google Trends search volume and Reddit- and Factiva-based metrics. The variables in the last group reflect investor attention and sentiment surrounding a cryptocurrency.

3. Forecasting Methods

We begin with the following general model for predicting the one-day-ahead cryptocurrency excess return:

$$r_{i,t+1} = f(\mathbf{x}_{i,t}; \boldsymbol{\eta}) + \varepsilon_{i,t+1}, \quad (3.1)$$

where $r_{i,t}$ is the day- t excess return for cryptocurrency i (i.e., the target), $\mathbf{x}_{i,t} = [x_{i,1,t} \ \cdots \ x_{i,k,t}]'$ is a k -vector of predictors, $f(\mathbf{x}_{i,t}; \boldsymbol{\eta})$ is the conditional expectation (or prediction) function that depends on a vector of parameters $\boldsymbol{\eta}$, and $\varepsilon_{i,t+1}$ is a zero-mean disturbance term. Observe that Equation (3.1) is a panel model, as it assumes that the prediction function and its parameters are the same across cryptocurrencies. The parameter homogeneity assumption substantially reduces the number of parameters that we need to estimate (since we do not have to estimate a separate set of parameters for each cryptocurrency), which helps to improve out-of-sample performance in light of the bias-variance trade-off. In the context of asset return prediction using machine learning,

Freyberger, Neuhierl, and Weber (2020) and Gu, Kelly, and Xiu (2020), among others, use a panel approach to forecast individual stock returns, while Filippou et al. (2022) employ such an approach to forecast exchange rate changes.

Based on Equation (3.1), the excess return forecast is given by

$$\hat{r}_{i,t+1} = \hat{f}(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}), \quad (3.2)$$

where $\hat{f}(\cdot; \hat{\boldsymbol{\eta}})$ is the fitted prediction function based on data through day t . By fitting the prediction function using data through day t , we ensure that there is no look-ahead bias in the forecast. We consider different machine-learning methods for estimating the prediction function.⁸

3.1. Linear Model Estimated via the Elastic Net

We first consider a linear specification for the prediction function, where we fit the linear model via the ENet (Zou and Hastie 2005). The linear-ENet excess return forecast (abstracting from the intercept term) can be expressed as

$$\hat{r}_{i,t+1}^{\text{LinENet}} = \mathbf{x}'_{i,t} \hat{\boldsymbol{\eta}}, \quad (3.3)$$

where $\hat{\boldsymbol{\eta}}$ is the vector of coefficients for the linear model estimated via the ENet based on data through t . By construction, conventional OLS estimation of a linear model maximizes the fit of the model over the training sample. However, especially in the presence of a large number of correlated predictors and a low signal-to-noise ratio, OLS estimation is highly susceptible to overfitting the model to the training data, which harms out-of-sample performance. The ENet is an extension of the well-known least absolute shrinkage and selection operator (LASSO, Tibshirani 1996). Like the LASSO, the ENet uses penalized (or regularized) regression to shrink the parameter estimates toward zero, thereby helping to guard against overfitting.

The ENet objective function for the linear model is given by

$$\arg \min_{\boldsymbol{\eta}} \frac{1}{2(t-1)n} \left[\sum_{i=1}^n \sum_{s=1}^{t-1} (r_{i,s+1} - \mathbf{x}'_{i,s} \boldsymbol{\eta})^2 \right] + \lambda P_{\delta}(\boldsymbol{\eta}), \quad (3.4)$$

⁸We fit the prediction models and generate the forecasts in Python using the `scikit-learn`, `XGBoost`, and `TensorFlow` packages.

where

$$P_\delta(\boldsymbol{\eta}) = 0.5(1 - \delta)\|\boldsymbol{\eta}\|_2^2 + \delta\|\boldsymbol{\eta}\|_1; \quad (3.5)$$

$\lambda \geq 0$ is a hyperparameter that governs the degree of shrinkage; $\|\cdot\|_1$ and $\|\cdot\|_2$ are the ℓ_1 and ℓ_2 norms, respectively; $0 \leq \delta \leq 1$ is a hyperparameter for blending the ℓ_1 and ℓ_2 components in the penalty term; and n is the number of cryptocurrencies.⁹ The ENet objective function in Equation (3.4) reduces to that for OLS when $\lambda = 0$. If $\delta = 1$, then Equation (3.4) corresponds to the LASSO objective function, while $\delta = 0$ corresponds to the ridge (Hoerl and Kennard 1970) objective function. Because Equation (3.5) includes an ℓ_1 component (as in the LASSO), it permits shrinkage to exactly zero, so the ENet also performs variable selection. To implement ENet estimation, we need to tune the hyperparameters λ and δ . We describe the walk-forward cross-validation strategy that we use to tune hyperparameters in Section 3.5.

3.2. Random Forest

The forecast in Equation (3.3) is based on a linear approximation to the prediction function for the general model in Equation (3.1). Next, we consider three machine-learning techniques that provide flexible nonlinear approximations to the prediction function, beginning with the random forest (Breiman 2001). A random forest is based on decision trees, which allow for multiway interactions and higher-order effects of predictors. In essence, a decision tree partitions the predictor space into non-overlapping regions and assigns a prediction (or score) for the target in each region. The partitions are generated by a sequence of splitting rules, typically based on the classification and regression tree (CART) algorithm (Breiman et al. 1984). The split at the top of a tree is the “root node,” while the final set of subgroups defining the predictive regions at the bottom of a tree is comprised of “terminal” or “leaf nodes”; the splits in between are “internal nodes.” For a regression problem, the prediction is the average value of the target observations in a given leaf node. The

⁹For notational simplicity, Equation (3.4) assumes a balanced panel. For our application, the panel is unbalanced. It is straightforward to modify the notation for an unbalanced panel.

forecast corresponding to a regression tree with U leaf nodes can be expressed as

$$\hat{r}_{i,t+1}^{\text{RegTree}} = \sum_{u=1}^U \bar{r}_u \mathbf{1}_u(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}_u), \quad (3.6)$$

where the indicator function $\mathbf{1}_u(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}_u) = 1$ if $\mathbf{x}_{i,t} \in R_u(\hat{\boldsymbol{\eta}}_u)$ for the u th region denoted by R_u (corresponding to the parameter vector $\hat{\boldsymbol{\eta}}_u$ that is based on the splits defining the tree) and zero otherwise, and \bar{r}_u is the average value of the target observations in R_u for the training sample.

A regression tree with a large number of leaves (i.e., a “deep” tree) can capture complex non-linear predictive relationships in the data.¹⁰ Although a deep tree substantially reduces the bias of the fitted tree, its high variance makes it susceptible to overfitting. To reduce the variance and thereby improve out-of-sample performance in light of the bias-variance trade-off, a random forest employs bagging (Breiman 1996) by averaging forecasts over many deep trees, where each tree is constructed based on a bootstrap sample of the original training data. To further reduce the variance by decorrelating the trees, each split is based on a randomly selected subset of the predictors. Indexing the bootstrap samples by b , the random forest forecast is given by

$$\hat{r}_{i,t+1}^{\text{RanFor}} = \frac{1}{B} \sum_{b=1}^B \left[\sum_{u=1}^U \bar{r}_u^{(b)} \mathbf{1}_u^{(b)}(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}_u) \right], \quad (3.7)$$

where B is the number of bootstrap samples, and $\bar{r}_u^{(b)}$ and $\mathbf{1}_u^{(b)}(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}_u)$ are the analogs to \bar{r}_u and $\mathbf{1}_u(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}_u)$, respectively, in Equation (3.6) for the b th bootstrap sample. Using walk-forward cross-validation, we tune the following hyperparameters for the random forest: maximum depth of each tree, maximum number of predictors to consider for a split, minimum number of target observations in a leaf node, minimum number of predictor observations needed to split an internal node, impurity threshold for splitting a node, minimum number of predictor observations needed for a leaf node, number of trees in the forest (B).¹¹

¹⁰In the extreme, if the tree is grown so that there is only one target observation in each leaf node, then the regression tree is “fully grown.” A fully grown tree perfectly fits (or interpolates) the training data.

¹¹The `sklearn.ensemble.RandomForestRegressor` documentation provides details for the hyperparameters for the random forest.

3.3. XGBoost

A boosted tree is another approach for forecasting with a regression tree. It is based on gradient boosting (Breiman 1997; Friedman 2001), a sequential ensemble method for improving out-of-sample prediction. The idea is to fit the prediction function additively:

$$\hat{f}(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}) = \sum_{m=1}^M \hat{f}_m(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}_m). \quad (3.8)$$

Each function $\hat{f}_m(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}_m)$ on the right-hand-side of Equation (3.8) is “weak” learner (i.e., a relatively simple model). Although relatively simple models help to guard against overfitting, they are more likely to suffer from biases and thus evince poor fit. Boosting improves the fit by adding an $\hat{f}_m(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}_m)$ element to model the residuals from the previous function in the sequence. In this way, boosting refines a sequence of simple models to reduce the bias. In the context of decision trees, boosting entails constructing a sequence of relatively “shallow” trees, which are then combined into an ensemble. Random forests and boosted trees follow different tacks to improve out-of-sample performance: a random forest begins with a deep tree with low bias and uses bagging across a large number of trees to reduce the variance; a boosted tree begins with a shallow tree with low variance and refines the tree to reduce the bias.

To make boosting more robust, Friedman (2002) proposes stochastic gradient boosting. In the spirit of bagging, instead of basing each $\hat{f}_m(\mathbf{x}_{i,t}; \hat{\boldsymbol{\eta}}_m)$ in the sequence on all of the training data, each element is based on a randomly drawn (without replacement) subsample of the data. We fit boosted trees via stochastic gradient boosting using the XGBoost algorithm (Chen and Guestrin 2016). We tune the following hyperparameters for the XGBoost algorithm using walk-forward cross-validation: number of trees in the sequence (M), minimum number of target observations in a child after splitting, maximum depth of a tree, step-size shrinkage for gradient boosting, subsample ratio for the training data for each tree, subsample ratio of predictors for each tree, minimum loss reduction needed to make a split, ℓ_1 and ℓ_2 regularization hyperparameters.¹²

¹²The **XGBoost** documentation provides details for the XGBoost algorithm, including its hyperparameters.

3.4. Deep Neural Network

A feedforward neural network is another machine-learning technique that provides a flexible nonlinear approximation to the prediction function. A neural network is comprised of multiple layers. The set of predictors makes up the first (or “input”) layer. Next, there are $L \geq 1$ “hidden” layers, where each hidden layer l contains P_l neurons. Each neuron takes signals from the neurons in the previous layer to generate a new signal:

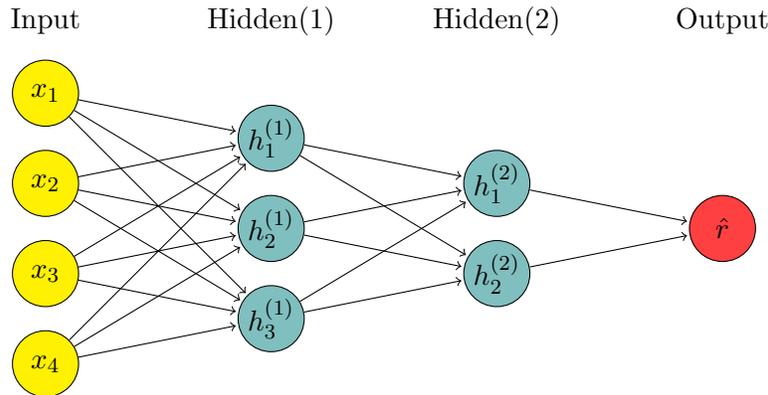
$$h_m^{(l)} = g \left(\omega_{m,0}^{(l)} + \sum_{j=1}^{P_{l-1}} \omega_{m,j}^{(l)} h_j^{(l-1)} \right) \quad \text{for } m = 1, \dots, P_l; l = 1, \dots, L, \quad (3.9)$$

where $h_m^{(l)}$ is the signal corresponding to the m th neuron in the l th hidden layer¹³; $\omega_{m,0}^{(l)}, \omega_{m,1}^{(l)}, \dots, \omega_{m,P_{l-1}}^{(l)}$ are weights; and $g(\cdot)$ is a nonlinear activation function. The final layer is the “output” layer that takes the signals from the last hidden layer and transforms them into a prediction:

$$\hat{r}_{i,t+1}^{\text{Net}} = \omega_0^{(L+1)} + \sum_{j=1}^{P_L} \omega_j^{(L+1)} h_j^{(L)}. \quad (3.10)$$

For the activation function, we use the popular rectified linear unit (ReLU) or “leaky” ReLU (Maas, Hannun, and Ng 2013).¹⁴

The following diagram illustrates the basic structure of a simple feedforward neural network with four inputs and two hidden layers with three and two neurons, respectively:



¹³For the first hidden layer, $h_j^{(0)} = x_{i,j,t}$ for $j = 1, \dots, k$.

¹⁴The ReLU function is $g(x) = \max\{x, 0\}$; for the leaky ReLU, $g(x) = x$ if $x > 0$ and $0.01x$ if $x \leq 0$. The leaky ReLU adjusts the conventional ReLU to help prevent the neural network from “dying,” meaning that no neurons are activated in the network.

The network interactions and activation function allow for complex nonlinearities as the inputs feed through to the hidden layers and finally to the output layer. Although a single hidden layer with enough neurons is theoretically sufficient for approximating any smooth function (e.g., Cybenko 1989; Funahashi 1989; Hornik, Stinchcombe, and White 1989; Hornik 1991), neural networks with multiple hidden layers are commonly used, as there are performance advantages to including multiple hidden layers (e.g., Goodfellow, Bengio, and Courville 2016; Rolnick and Tegmark 2018). Accordingly, we consider a “deep” neural network with three hidden layers.¹⁵

Fitting a neural network entails estimating the weights. This is usually done via a stochastic gradient descent algorithm. We fit the deep neural network by minimizing the training sample MSE using the Adam stochastic gradient descent algorithm (Kingma and Ba 2015). We use walk-forward cross-validation to tune the following: activation function (ReLU or leaky ReLU); dropout rate (Srivastava et al. 2014); learning rate for stochastic gradient descent; number of neurons in the first, second, and third hidden layers (up to 100, 50, and 30, respectively).¹⁶

3.5. Hyperparameter Tuning

We tune the hyperparameters via walk-forward five-fold cross-validation. The “walk-forward” aspect of the procedure respects the time-series dimension of our panel data. We divide the available data at the time of forecast formation into initial training and validation samples, where the latter is comprised of panel data observations for the last 300 days. The validation sample is further divided into five folds comprised of the panel data observations for the first through fifth 60-day periods of the validation sample.

For a given vector of hyperparameter values, we fit the model using the initial training sample, generate forecasts for the first fold of the validation sample for the available cryptocurrencies, and store the MSE. Next, we fit the model using panel data observations for the initial training sample and the first fold of the validation sample, generate forecasts for the second fold of the validation sample for the available cryptocurrencies and store the MSE. We proceed in this manner through

¹⁵A neural network with one or two (three or more) hidden layers is typically referred to as a “shallow” (“deep”) network.

¹⁶For the Adam algorithm, we allow up to 100 epochs (with early stopping based on a validation loss) and use a batch size of 128. To reduce the influence of weight initialization in the algorithm, we fit the model five times and take an average of the forecasts generated by the five fitted models.

the remaining folds of the validation sample and compute the average MSE over the five folds. Finally, we select the vector of hyperparameters that minimizes the average MSE.¹⁷

We typically consider a lengthy grid of values for each of the hyperparameters. For the random forest, XGBoost, and deep neural network, it becomes computationally infeasible to compute the cross-validation MSE for all possible combinations of hyperparameter values implied by the grids. Instead, we use the *Optuna* (Akiba et al. 2019) framework, which employs a Bayesian approach to smartly select a subset of vectors from among all possible combinations of hyperparameter values so that we are more likely to select a vector of hyperparameter values that is nearly optimal. After tuning the hyperparameters, we train the model using all of the available data at the time of forecast formation to generate the forecasts. We tune the hyperparameters and train the models every 30 days using data for the available cryptocurrencies at the time of forecast formation.

3.6. Ensembles

We also consider a pair of ensemble (or combination) forecasts:

Ensemble-nonlinear Average of the random forest, XGBoost, and neural network forecasts.

Ensemble-all Average of the linear-ENet, random forecast, XGBoost, and neural network forecasts.

Ensemble forecasts often perform nearly as well as or better than the best individual forecast, and they are frequently used in machine-learning applications. Intuitively, averaging across multiple forecasts can reduce the risk of relying on a single forecast—similarly to diversifying across assets to reduce portfolio risk—which can improve out-of-sample performance (Timmermann 2006; Rapach, Strauss, and Zhou 2010). Ensemble forecasts are also useful in a practical sense in that it is difficult to know a priori the best individual forecast.

4. Out-of-Sample Results

We compare cryptocurrency excess return forecasts based on the machine-learning methods described in Section 3 to the prevailing mean benchmark forecast. The prevailing mean forecast for

¹⁷The results are robust to other fold patterns (e.g., ten folds comprised of panel data observations for the first through tenth 30-day periods of the validation sample).

cryptocurrency i is the average of the excess return observations available at the time of forecast formation:

$$\hat{r}_{i,t+1}^{\text{Bench}} = \left(\frac{1}{t}\right) \sum_{s=1}^t r_s. \quad (4.1)$$

Equation (4.1) corresponds to the constant expected excess return model:

$$r_{i,t+1} = \mu_i + \varepsilon_{i,t+1}. \quad (4.2)$$

Equation (4.2) assumes that the excess return is unpredictable (apart from its unconditional mean). Because asset returns contain an inherently large unpredictable component, the prevailing mean benchmark is a relevant and stringent benchmark (e.g., Goyal and Welch 2008).¹⁸

We can conveniently compare the out-of-sample MSE for the prevailing mean benchmark to that of a competing forecast via the out-of-sample R^2 statistic (Fama and French 1989; Campbell and Thompson 2008):

$$R_{i,\text{OS}}^2 = 1 - \frac{\sum_{s=1}^{T-t_{\text{in}}} \hat{r}_{i,t_{\text{in}}+s}^{\text{Compete}}}{\sum_{s=1}^{T-t_{\text{in}}} \hat{r}_{i,t_{\text{in}}+s}^{\text{Bench}}}, \quad (4.3)$$

where $\hat{r}_{i,t}^{\text{Compete}}$ generically denotes a competing forecast and t_{in} (T) is the end of the initial in-sample period (total sample). Equation (4.3) measures the proportional reduction in out-of-sample MSE for the competing forecast vis-à-vis the prevailing mean benchmark. We use the Diebold and Mariano (1995) and West (1996) (DMW) statistic to test whether the competing forecast delivers a statistically significant reduction in MSE relative to the prevailing mean benchmark. The DMW statistic can be computed via the t -statistic corresponding to the intercept a_i in the following time-series regression:

$$\underbrace{\left(r_{i,t} - \hat{r}_{i,t}^{\text{Bench}}\right)^2 - \left(r_{i,t} - \hat{r}_{i,t}^{\text{Compete}}\right)^2}_{d_{i,t}} = a_i + \varepsilon_{i,t} \quad \text{for } t = t_{\text{in}} + 1, \dots, T, \quad (4.4)$$

¹⁸We use the first 28 days of available excess return observations as an initial in-sample estimation period for computing the prevailing mean.

where $d_{i,t}$ is the day- t loss differential (i.e., the difference between the squared errors for the benchmark and competing forecasts). We test the null hypothesis $H_0: a_i \leq 0$ against the (one-sided, upper-tail) alternative $H_A: a_i > 0$, which is tantamount to testing $H_0: R_{i,\text{OS}}^2 \leq 0$ against $H_A: R_{i,\text{OS}}^2 > 0$.

We also compute a pooled version of Equation (4.3) for the entire set of cryptocurrencies taken together:

$$R_{\text{All,OS}}^2 = 1 - \frac{\sum_{i=1}^n \sum_{s=1}^{T-t_{\text{in}}} \hat{r}_{i,t_{\text{in}}+s}^{\text{Compete}}}{\sum_{i=1}^n \sum_{s=1}^{T-t_{\text{in}}} \hat{r}_{i,t_{\text{in}}+s}^{\text{Bench}}}, \quad (4.5)$$

Equation (4.5) is the proportional reduction in out-of-sample MSE for the competing forecast vis-à-vis the benchmark forecast across all of the cryptocurrencies. In the context of Equation (4.5), we compute the DMW statistic via the t -statistic corresponding to the intercept a in the following pooled regression:

$$d_{i,t} = a + \varepsilon_{i,t} \quad \text{for } i = 1, \dots, n; t = t_{\text{in}} + 1, \dots, T. \quad (4.6)$$

We test $H_0: a \leq 0$ ($R_{\text{All,OS}}^2 \leq 0$) against $H_A: a > 0$ ($R_{\text{All,OS}}^2 > 0$). When computing the DMW statistic using Equation (4.6), we account for cross-sectional dependency by clustering the standard error by cryptocurrencies.

Table 3 reports R_{OS}^2 statistics for daily excess return forecasts for each cryptocurrency and the different forecasting methods. Nearly all of the R_{OS}^2 statistics are positive, so the different forecasting methods consistently outperform the prevailing mean benchmark in terms of out-of-sample MSE.¹⁹ For the linear-ENet forecast, 39 of the 41 R_{OS}^2 statistics are positive, and 18 are significant at conventional levels according to the DMW statistic. Turning to the decision trees, the random forest and XGBoost forecasts both have a lower MSE than the prevailing mean benchmark for 37 cryptocurrencies, and 18 and 26, respectively, of these improvements in MSE are significant. For the neural network forecast, all 41 of the R_{OS}^2 statistics are positive, and 27 are significant. Taking all of the cryptocurrencies together (see the ‘‘All’’ row in Table 3), the R_{OS}^2 statistics are

¹⁹In contrast, as anticipated (see Section 3.1), linear forecasts based on conventional OLS estimation substantially underperform the prevailing mean benchmark for all of the cryptocurrencies.

Table 3: R_{OS}^2 Statistics

The table reports out-of-sample R^2 (R_{OS}^2) statistics in percent for daily excess return forecasts for the cryptocurrency in the first column. The start of the out-of-sample period for forecast evaluation is given in the second column; the end of the out-of-sample period is 2021-12-31 for all of the cryptocurrencies. The “All” row reports R_{OS}^2 statistics for all 41 of the cryptocurrencies taken together. The R_{OS}^2 statistic measures the proportional reduction in out-of-sample mean squared error (MSE) for the competing forecast in the column heading vis-à-vis the prevailing mean benchmark forecast; based on the Diebold and Mariano (1995) and West (1996) test, *, **, and *** indicate that the reduction in out-of-sample MSE is significant at the 10%, 5%, and 1% levels, respectively. The penultimate (last) row reports the number of positive (significant, at the 10% level) R_{OS}^2 statistics for the 41 cryptocurrencies.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Cryptocurrency	Out-of-sample start	Linear-ENet	Random forest	XGBoost	Neural network	Ensemble-nonlinear	Ensemble-all
linch	2021-01-24	1.31	1.53	1.72	1.75	1.98	1.95
Aave	2020-11-08	1.55**	3.01	3.71*	1.75**	3.04*	2.73**
Algorand	2019-07-21	1.27	1.53*	1.56	1.35	1.59*	1.53*
Balancer	2020-07-24	0.94**	2.26**	2.21**	1.35**	2.10**	1.88**
Basic Attention Token	2017-11-04	0.48*	1.73**	2.11**	1.07***	1.93***	1.64***
Bitcoin	2014-01-01	-0.05	0.22	1.98***	1.06**	1.58***	1.31***
Bitcoin Cash	2017-08-30	0.48	-0.78	-0.35	0.54	0.21	0.43
Bitcoin SV	2018-12-14	0.29*	0.50	0.78**	0.78***	0.81***	0.73***
Cardano	2017-12-30	2.47**	2.80**	4.45***	2.87**	3.64***	3.40**
Chainlink	2017-10-28	0.33	0.66	0.73	0.45	0.87*	0.81**
Compound	2020-07-17	0.40	0.95	1.40	0.75	1.25	1.14
Crypto.com Coin	2019-04-18	0.41	0.64*	1.52*	0.63**	1.06**	0.92**
Curve DAO Token	2020-09-13	1.40	0.12	0.39	1.66	0.97	1.19
Dash	2014-03-09	4.72***	4.77***	6.16***	6.32**	6.12***	5.88***
Decentraland	2017-09-23	1.12*	1.36*	1.55**	1.64**	1.61**	1.51**
Decred	2016-06-15	0.18	0.07	1.29**	0.81*	0.94***	0.79***
Dogecoin	2014-02-21	0.19	0.36	1.33**	0.52**	0.87***	0.74**
Ethereum	2015-09-06	0.14	1.72**	2.40**	0.38	2.09***	1.72***
Ethereum Classic	2016-08-23	3.29***	4.02***	4.74***	3.52***	4.33***	4.12***
FTX Token	2019-09-18	0.37	2.32	1.77*	0.46*	1.71*	1.43*
Gnosis	2017-05-31	0.50*	0.68	1.04*	0.94***	1.07**	0.97**

1.11%, 1.45%, 2.08%, and 1.54% for the linear-ENet, random forest, XGBoost, and neural network forecasts, respectively, all of which are significant at the 1% level.

Table 3 (continued)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Cryptocurrency	Out-of-sample start	Linear-ENet	Random forest	XGBoost	Neural network	Ensemble-nonlinear	Ensemble-all
HedgeTrade	2019-12-01	-0.01	0.14	0.76	0.17	0.51*	0.42*
Huobi Token	2019-04-04	0.45	-0.51	-0.19	0.97	0.54	0.71
Internet Computer	2021-06-09	4.12	3.42	3.95	3.66	3.86	4.00
Litecoin	2014-01-01	0.68**	0.77	2.11***	0.85*	1.55***	1.43***
Livepeer	2019-01-18	0.98**	1.60*	2.03**	1.21**	1.69**	1.55**
Maker	2018-01-24	0.13	0.41	1.41**	0.40*	0.94***	0.78**
Neo	2017-08-13	2.54***	3.35***	3.64***	3.31***	3.67***	3.48***
OMGNetwork	2017-08-13	5.15***	6.40***	6.22***	5.58***	6.31***	6.11***
Perpetual Protocol	2021-03-05	0.96***	0.16	-0.45	1.50***	1.03	1.19
Polkadot	2020-09-18	0.77	0.53	1.98	1.38	1.58	1.54
Quant	2019-04-14	0.32	1.25	1.80**	0.64*	1.34**	1.11**
Stellar	2015-10-29	0.17	-0.02	0.86***	0.05	0.45***	0.41***
SushiSwap	2020-09-30	2.43**	3.81**	4.04**	3.11**	3.82**	3.52**
Synthetix	2020-05-08	0.30	0.60	1.20	0.47	0.87	0.75
Tezos	2018-07-29	0.35	-0.30	-0.08	0.18	0.15	0.31
UMA	2020-10-07	1.54*	2.49*	1.38	1.60*	1.95*	1.91*
Uniswap	2020-10-17	1.82**	5.15*	4.21	2.81**	4.28*	3.76**
XRP	2014-09-13	0.29	0.64*	1.71***	0.85**	1.31***	1.11***
yearn.finance	2020-08-23	3.52	5.55*	6.27**	4.26*	5.49*	5.06*
Zcash	2016-11-27	1.00*	1.87**	2.11**	1.95***	2.23**	2.01**
All	2014-01-01	1.11***	1.45***	2.08***	1.54***	1.91***	1.78***
Number > 0		39	37	37	41	41	41
Number sig.		18	18	26	27	31	31

According to the last two columns of Table 3, the ensemble forecasts provide effective strategies for forecasting daily cryptocurrency excess returns. The ensemble-nonlinear forecast—an average of the random forest, XGBoost, and neural network forecasts—generates positive R_{OS}^2 statistics for all of the cryptocurrencies, 31 of which are significant. Taking the cryptocurrencies together, the R_{OS}^2 for the ensemble-nonlinear forecast is 1.91% (significant at the 1% level). Like the ensemble-nonlinear forecast, the ensemble-all forecast outperforms the prevailing mean benchmark for all of the cryptocurrencies, and again 31 of the R_{OS}^2 statistics are significant. When we take all of the cryptocurrencies together, the ensemble-all produces an R_{OS}^2 statistic of 1.78% (significant at the 1% level).

Because daily asset returns inherently contain a large unpredictable component, the R_{OS}^2 statistics will naturally be limited in magnitude. Nevertheless, they are often relatively sizable in many cases; furthermore, we show in Section 5 that the degree of out-of-sample excess return predictability indicated by the R_{OS}^2 statistics translates into substantial economic value. Many of the R_{OS}^2 statistics in Table 3 are above 1%, and they range from approximately 2.5% to well above 6% for cryptocurrencies such as Cardano, Dash, Ethereum Classic, Internet Computer, Neo, OMGNetwork, SushiSwap, and yearn.finance. These cryptocurrencies became available in 2017, 2014, 2016, 2021, 2017, 2017, 2020, and 2020, respectively, so it is not the case that only relatively young or old cryptocurrencies evince the strongest return predictability.

Overall, the results in Table 3 demonstrate that machine-learning techniques provide an effective means of extracting information from a wide range of predictors to improve daily cryptocurrency return prediction. All of the forecasts in Table 3 perform well, so the out-of-sample gains are robust across the different machine-learning methods as well as the ensemble forecasts.

While both the linear and nonlinear forecasts perform well in Table 3, the R_{OS}^2 statistics in the “All” row of Table 3 suggest that nonlinearities are important for improving out-of-sample performance. Specifically, the R_{OS}^2 statistic for the linear-ENet forecast (1.11%) is lower than that for the nonlinear random forest, XGBoost, and neural network forecasts (1.45%, 2.08%, and 1.54%, respectively) as well as the ensemble-nonlinear forecast (1.91%) that is an average of the three nonlinear forecasts. Thus, allowing for nonlinearities improves the accuracy of daily cryptocurrency return forecasts in terms of out-of-sample MSE. We test for a significant difference between MSEs for the linear-ENet and a nonlinear forecast via a pooled DMW statistic, which corresponds to the t -statistic for the intercept a in a modified version of Equation (4.6):

$$\left(r_{i,t} - \hat{r}_{i,t}^{\text{LinENet}}\right)^2 - \left(r_{i,t} - \hat{r}_{i,t}^{\text{Nonlin}}\right)^2 = a + \varepsilon_{i,t} \quad \text{for } i = 1, \dots, n; t = t_{\text{in}} + 1, \dots, T, \quad (4.7)$$

where $\hat{r}_{i,t}^{\text{Nonlin}}$ generically denotes a nonlinear forecast. The left-hand side of Equation (4.7) is the loss differential between the linear-ENet and nonlinear forecasts. We continue to account for cross-sectional dependency by clustering the standard error by cryptocurrencies.

The t -statistics corresponding to a in Equation (4.7) are reported in Table 4. We reject the null hypothesis in the upper tail at the 1% level for the random forest, XGBoost, neural network,

Table 4: Linear-ENet Versus Nonlinear Forecasts

The table reports the t -statistic corresponding to a for the following regression:

$$(r_{i,t} - \hat{r}_{i,t}^{\text{LinENet}})^2 - (r_{i,t} - \hat{r}_{i,t}^{\text{Nonlin}})^2 = a + \varepsilon_{i,t} \quad \text{for } i = 1, \dots, n; t = t_{\text{in}} + 1, \dots, T,$$

where $r_{i,t}$ is the day- t cryptocurrency excess return, $\hat{r}_{i,t}^{\text{LinENet}}$ is the linear-ENet forecast, and $\hat{r}_{i,t}^{\text{Nonlin}}$ is the nonlinear forecast in the column heading. The unbalanced panel sample includes 41 cryptocurrencies, and the forecast evaluation period is 2014-01-01 to 2021-12-31. The t -statistic is a pooled Diebold and Mariano (1995) and West (1996) statistic for comparing the predictive accuracy of two forecasts; *, **, and *** indicate significance at the 10%, 5%, and 1% levels, respectively.

(1)	(2)	(3)	(4)
Random forest	XGBoost	Neural network	Ensemble-nonlinear
3.69***	8.51***	5.29***	9.83***

and ensemble-nonlinear forecasts, so the nonlinear forecasts significantly reduce out-of-sample MSE compared to the linear forecast. In sum, there is significant evidence that allowing for nonlinearities in the prediction model improves the accuracy of daily cryptocurrency excess return forecasts. We explore the role of nonlinearities further in Section 6.

Table 3 shows that daily cryptocurrency return forecasts based on our rich set of predictors outperform the prevailing mean benchmark over the full forecast evaluation period available for each cryptocurrency. To get a sense of the consistency of the out-of-sample gains over time, Figure 2 depicts the cumulative difference in squared forecast errors for the prevailing mean benchmark vis-à-vis the ensemble-all forecast. To conserve space, we focus on the ensemble-all forecast in Figure 2; as shown in Figures A1 to A5 in the Online Appendix, the plots are qualitatively similar for the other forecasts. The cumulative difference in squared forecast errors is an informative graphical device suggested by Goyal and Welch (2003, 2008) for comparing the predictive accuracy of a competing forecast to that of a benchmark over time. We can determine if the competing forecast outperforms the benchmark in terms of out-of-sample MSE for any subsample by comparing the height of the curve at the beginning and end of the interval corresponding to the subsample. If the curve is higher (lower) at the end of the interval relative to the beginning, then the competing forecast outperforms (underperforms) the benchmark during the subsample. A uniformly positively sloped curve indicates that the competing forecast always outperforms the benchmark. While such an outcome is infeasible in practice with regard to daily asset return forecasting, more realistically,

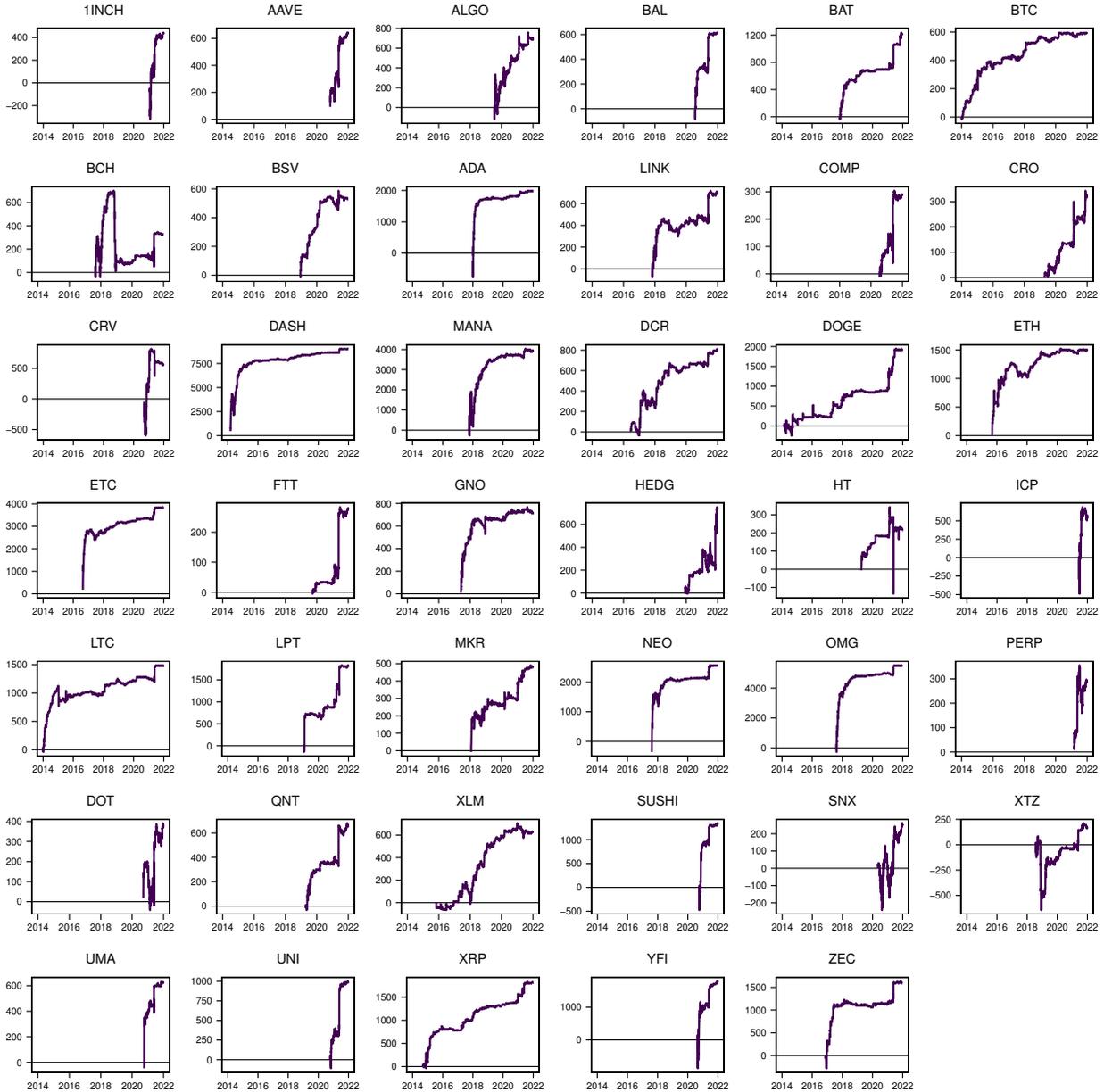


Figure 2: Cumulative Differences in Squared Forecast Errors

Each panel depicts the cumulative difference in squared forecast errors for the prevailing mean vis-à-vis the ensemble-all excess return forecast for the cryptocurrency (denoted by its symbol) in the panel heading.

we seek a curve with a predominantly positive slope that avoids long segments with steeply negative slopes.

Figure 2 indicates that the ensemble-all forecast outperforms the prevailing mean benchmark consistently over time for the different cryptocurrencies. The slopes are positive the vast majority

of the time, and there are many segments with steeply positive slopes, so the information in the predictors substantially improves out-of-sample accuracy. The curve initially has a quite steeply positive slope for some cryptocurrencies, such as Ethereum Classic (ETC), Neo (NEO), OMGNetwork (OMG), and UMA. However, it is not the case that return predictability is always strongest at the outset of the out-of-sample period near the introduction of a cryptocurrency. For example, numerous cryptocurrencies that were introduced relatively early exhibit strong predictability from late 2017 through 2018. Interestingly, this period coincides with the substantial appreciation and subsequent depreciation evident for many cryptocurrencies during this same period in Figure 1. Numerous cryptocurrencies in Figure 2 also have steeply positive slopes in early 2020, corresponding to the substantive appreciations for many cryptocurrencies seen in Figure 1 during the advent of COVID-19. Thus, it appears that the information in the predictors becomes especially useful for predicting returns when cryptocurrencies experience large swings in value. Furthermore, the out-of-sample gains are very consistent for many cryptocurrencies in Figure 2, as the negatively sloped segments of the curve are short-lived and not very steep. This is the case for some of the oldest and best-known cryptocurrencies, including Bitcoin (BTC), Ethereum (ETH), Litecoin (LTC), and XRP.

We further examine how cryptocurrency return prediction changes over time based on financial market conditions. We measure financial market conditions using the VIX and TED spread, which are two of the most popular variables for tracking uncertainty in financial markets. To investigate how cryptocurrency return predictability varies with financial market uncertainty, we estimate a panel version of Equation (4.4) that includes fixed effects and the VIX or TED spread as an explanatory variable:

$$d_{i,t} = a_i + bz_t + \varepsilon_{i,t+1} \quad \text{for } i = 1, \dots, n; t = t_{\text{in}} + 1, \dots, T, \quad (4.8)$$

where $d_{i,t}$ is the loss differential in Equation (4.4), and z_t is the VIX or TED spread. We transform the VIX and TED spread using the deviation from a 7- or 28-day moving average, which reduces the substantial persistence in the daily VIX and TED spread. If $b > (<) 0$ in Equation (4.8), then the competing forecast becomes more (less) accurate vis-à-vis the benchmark in terms of out-of-sample

MSE as financial uncertainty increases. We continue to account for cross-sectional dependency by clustering the standard error for b by cryptocurrencies.

Table 5: Return Predictability and Financial Market Uncertainty

The table reports the estimate of b for the following fixed-effects panel regression:

$$d_{i,t} = a_i + bz_t + \varepsilon_{i,t+1} \quad \text{for } i = 1, \dots, n; t = t_{\text{in}} + 1, \dots, T,$$

where $d_{i,t}$ is the day- t loss differential for the prevailing mean benchmark forecast vis-à-vis the competing forecast in the column heading, and z_t is the VIX or TED spread. The VIX and TED spread are transformed by computing the deviation from a 7- or 28-day moving average. The unbalanced panel sample includes 41 cryptocurrencies, and the forecast evaluation period is 2014-01-01 to 2021-12-31. If $b > (<) 0$, then the competing forecast becomes more (less) accurate vis-à-vis the benchmark in terms of out-of-sample mean squared error as z_t increases. Standard errors are in parentheses; *, **, and *** indicate significance at the 10%, 5%, and 1% levels, respectively.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Transformation	Linear-ENet	Random forest	XGBoost	Neural network	Ensemble-nonlinear	Ensemble-all
Panel A: VIX						
Deviation from 7-day moving average	0.04 (0.05)	-0.05 (0.05)	0.13* (0.06)	0.09* (0.05)	0.06 (0.05)	0.05 (0.05)
Deviation from 28-day moving average	0.01 (0.02)	0.01 (0.02)	0.11*** (0.02)	0.04** (0.02)	0.05*** (0.02)	0.04** (0.02)
Panel B: TED spread						
Deviation from 7-day moving average	1.15*** (0.40)	-2.07** (0.80)	6.04*** (0.92)	0.09* (0.05)	1.64** (0.62)	1.50*** (0.52)
Deviation from 28-day moving average	0.32 (0.27)	-0.48 (0.35)	1.61*** (0.43)	0.07 (0.28)	0.49 (0.33)	0.46 (0.31)

Table 5 reports estimates of b in Equation (4.8) for the two uncertainty measures and their two transformations. Of the 24 estimates of b , 21 are positive. The estimate of b is only significantly negative (at the 5% level) for the random forest forecast and the TED spread when it is transformed using the deviation from a 7-day moving average. The estimate of b is significantly positive at the 10%, 5%, and 1% levels in twelve, nine, and six cases, respectively. In general, Table 5 indicates that an increase in financial uncertainty—as captured by an increase in the VIX or TED spread relative to its recent average—is associated with an increase in the degree of cryptocurrency return predictability.

5. Economic Value

In this section, we examine the economic value of machine-learning forecasts of cryptocurrency excess returns in an investment context. We first consider a mean-variance investor who allocates between an individual cryptocurrency and risk-free Treasury bills. At the end of day t , the investor determines their allocations based on the following objective function:

$$\arg \max_{w_{i,t+1}} w_{i,t+1} \hat{r}_{i,t+1} - 0.5\gamma w_{i,t+1}^2 \hat{\sigma}_{i,t+1}^2, \quad (5.1)$$

where γ is the coefficient of relative risk aversion; $w_{i,t+1}$ and $1 - w_{i,t+1}$ are the day- $(t+1)$ allocations to cryptocurrency i and risk-free bills, respectively, which are determined at the end of day t ; and $\hat{r}_{i,t+1}$ ($\hat{\sigma}_{i,t+1}^2$) generically denotes the investor's forecast of the cryptocurrency excess return (variance), which are based on data available through day t . The well-known solution to equation (5.1) is given by

$$w_{i,t+1}^* = \left(\frac{1}{\gamma}\right) \left(\frac{\hat{r}_{i,t+1}}{\hat{\sigma}_{i,t+1}^2}\right). \quad (5.2)$$

We assume that the investor uses an exponentially weighted moving-average estimator for $\hat{\sigma}_{i,t+1}^2$, which is a popular variance estimator among practitioners.²⁰ To prevent the allocations from becoming implausible, we impose the restriction that $-1 \leq w_{i,t+1} \leq 2$. We assume that $\gamma = 3$; the results are similar for reasonable alternative values for γ .

We consider two cases. In the first, the investor uses the prevailing mean benchmark to forecast the cryptocurrency excess return in Equation (5.2); in the second, the investor instead uses a machine-learning forecast of the cryptocurrency excess return.²¹ The average utility (or certainty equivalent return) realized by the investor is given by

$$\bar{U}_i^j = \bar{r}_i^j - 0.5\gamma \hat{\sigma}_i^{2,j} \quad \text{for } j = \text{Bench, Compete}, \quad (5.3)$$

²⁰We use a value of 0.94 for the decay parameter.

²¹The investor always uses the exponentially weighted moving-average estimator to forecast the variance.

where \bar{r}_i^{Bench} ($\bar{r}_i^{\text{Compete}}$) and $\hat{\sigma}_i^{2,\text{Bench}}$ ($\hat{\sigma}_i^{2,\text{Compete}}$) are the mean and variance, respectively, for the portfolio excess return over the out-of-sample period when the investor uses the prevailing mean benchmark (competing machine-learning) forecast of the cryptocurrency excess return. The average utility gain for the investor when they use the machine-learning forecast in lieu of the prevailing mean benchmark is then given by

$$\Delta_i = \bar{U}_i^{\text{Compete}} - \bar{U}_i^{\text{Bench}}. \quad (5.4)$$

After multiplying the average utility gain in Equation (5.4) by 365, it can be interpreted as the annualized portfolio management fee that the investor would be willing to pay to have access to the information in the competing machine-learning forecast vis-à-vis the prevailing mean benchmark. In this way, Equation (5.4) measures the economic value of return predictability to the investor.

Table 6 reports annualized average utility gains (in percent) for each of the cryptocurrencies and the different forecasting methods. Of the 246 average utility gains, 228 (93%) are positive. Thus, in the vast majority of cases, the investor benefits from relying on a machine-learning forecast in lieu of the prevailing mean benchmark to guide asset allocation. Furthermore, the positive gains are extremely large in general. They are typically above 20%, and a number are well above 100%.²² Focusing on the two largest and best-known cryptocurrencies, Bitcoin and Ethereum, the average utility gains range from 3.86% (linear-ENet) to 111.31% (XGBoost) for the former and 30.62% (linear-ENet) to 78.47% (XGBoost) for the latter. Thus, return predictability has substantive economic value for the most familiar cryptocurrencies.

Figure 3 plots log cumulative excess returns for the portfolios based on the ensemble-all and prevailing mean benchmark forecasts. We focus on the ensemble-all forecast in Figure 3 to conserve space; as shown in Figures A6 to A10 in the Online Appendix, the pictures are qualitatively similar for the other machine-learning forecasts. For the vast majority of cryptocurrencies, the cumulative excess return profile for the portfolio based on the ensemble-all forecast appears clearly superior to that based on the prevailing mean benchmark forecast, consistent with the results in Table 6. Figure 3 provides further evidence of the value of machine-learning forecasts for cryptocurrency return prediction.

²²Such large gains easily survive even sizable transaction costs. However, transaction costs are usually relatively low in cryptocurrencies.

Table 6: Average Utility Gains

The table reports annualized average utility gains in percent for a mean-variance investor with a coefficient of relative risk aversion of three who allocates daily between the cryptocurrency in the first column and risk-free Treasury bills. The utility gain corresponds to the case where the investor allocates their portfolio based on the forecast of the daily cryptocurrency excess return in the column heading instead of the prevailing mean benchmark forecast. The investor always uses an exponentially weighted moving average estimator (with a decay parameter of 0.94) to forecast the variance. The start of the out-of-sample period for forecast evaluation is given in the second column; the end of the out-of-sample period is 2021-12-31 for all of the cryptocurrencies.

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Cryptocurrency	Out-of-sample start	Linear-ENet	Random forest	XGBoost	Neural network	Ensemble-nonlinear	Ensemble-all
1inch	2021-01-24	83.85	48.94	38.50	95.43	72.26	83.10
Aave	2020-11-08	78.83	55.62	108.45	82.93	93.12	97.04
Algorand	2019-07-21	103.87	100.76	101.31	114.88	110.50	110.00
Balancer	2020-07-24	32.23	41.34	66.47	53.68	61.35	60.05
Basic Attention Token	2017-11-04	19.51	32.67	27.24	35.36	46.34	45.88
Bitcoin	2014-01-01	3.86	34.98	111.31	46.16	81.25	61.69
Bitcoin Cash	2017-08-30	10.17	-45.46	-25.09	6.17	-6.00	-0.09
Bitcoin SV	2018-12-14	32.52	12.57	47.81	66.25	55.40	52.83
Cardano	2017-12-30	22.85	33.94	94.08	44.05	71.36	62.51
Chainlink	2017-10-28	17.16	17.71	47.07	17.89	38.14	36.57
Compound	2020-07-17	17.99	16.69	15.04	23.18	19.76	24.44
Crypto.com Coin	2019-04-18	18.12	17.52	50.04	26.92	44.66	39.41
Curve DAO Token	2020-09-13	81.64	25.39	4.20	92.28	48.60	62.93
Dash	2014-03-09	119.27	117.74	149.60	58.13	118.89	135.66
Decentraland	2017-09-23	196.36	185.61	166.24	210.41	202.18	205.58
Decred	2016-06-15	14.24	-0.69	30.99	35.95	32.99	30.84
Dogecoin	2014-02-21	-19.27	-34.90	-18.80	1.15	4.70	6.12
Ethereum	2015-09-06	30.62	35.05	78.47	40.83	77.38	70.87
Ethereum Classic	2016-08-23	90.47	98.83	93.12	99.78	117.06	114.84
FTX Token	2019-09-18	32.54	28.38	23.75	36.71	36.83	36.82
Gnosis	2017-05-31	-17.92	-6.68	26.14	23.85	24.57	14.92

Next, we construct a zero-investment long-short portfolio that invests in multiple cryptocurrencies. We form the portfolio by sorting the available cryptocurrencies in our sample using their excess return forecasts based on one of the machine-learning methods. Specifically, at the end of day t , we compute excess return forecasts for all of the available cryptocurrencies for day $t + 1$

Table 6 (continued)

(1)	(2)	(3)	(4)	(5)	(6)	(7)	(8)
Cryptocurrency	Out-of-sample start	Linear-ENet	Random forest	XGBoost	Neural network	Ensemble-nonlinear	Ensemble-all
HedgeTrade	2019-12-01	72.55	84.64	-33.34	84.23	77.85	82.45
Huobi Token	2019-04-04	37.82	14.34	34.79	54.72	37.11	40.93
Internet Computer	2021-06-09	109.12	88.22	132.00	77.68	108.81	110.78
Litecoin	2014-01-01	44.75	41.50	79.59	57.62	74.68	70.80
Livepeer	2019-01-18	39.85	-6.86	-34.82	60.13	13.18	31.00
Maker	2018-01-24	-10.27	-5.56	33.99	7.32	25.13	19.02
Neo	2017-08-13	66.28	69.38	90.95	86.65	93.09	89.70
OMGNetwork	2017-08-13	136.43	132.95	183.50	142.58	164.10	164.07
Perpetual Protocol	2021-03-05	103.76	52.38	32.18	115.38	85.14	99.69
Polkadot	2020-09-18	84.13	38.04	38.82	67.25	52.91	70.23
Quant	2019-04-14	23.44	24.51	18.67	27.31	29.36	31.62
Stellar	2015-10-29	44.66	33.47	47.41	27.76	52.18	53.39
SushiSwap	2020-09-30	110.36	100.57	101.35	130.96	112.36	117.77
Synthetix	2020-05-08	44.09	26.19	66.33	42.10	51.81	51.34
Tezos	2018-07-29	12.68	2.23	-8.45	-4.68	7.86	6.78
UMA	2020-10-07	142.24	150.07	101.15	143.89	138.21	143.91
Uniswap	2020-10-17	128.54	98.48	67.42	156.65	104.25	108.82
XRP	2014-09-13	44.05	49.83	103.22	74.23	98.66	91.64
yearn.finance	2020-08-23	150.12	105.39	132.36	178.47	152.23	162.58
Zcash	2016-11-27	13.94	26.60	-3.35	57.43	43.95	43.12

using a machine-learning method and sort the cryptocurrencies based on the return forecasts. The portfolio goes long (short) the 30% of cryptocurrencies with the highest (lowest) return forecasts.²³ When forming the long and short legs, we consider both value and equal weighting.

Table 7 reports performance metrics for long-short portfolios constructed using the sorted machine-learning cryptocurrency excess return forecasts. In addition, the table reports metrics for a cryptocurrency market portfolio based on the available cryptocurrencies in our dataset. Specifically, we compute the excess return for a value-weighted portfolio that invests in all of the available cryptocurrencies on a given day. We also report metrics for the CRSP value-weighted aggregate equity market portfolio based on daily excess return data from Kenneth French’s [Data Library](#) for

²³For some days, a tree-based forecast gives the same excess return forecast for all of the available cryptocurrencies, so we cannot sort the cryptocurrencies by the forecasts. For such days, we assume that the portfolio does not invest in cryptocurrencies, so the excess return for the long-short portfolio is zero.

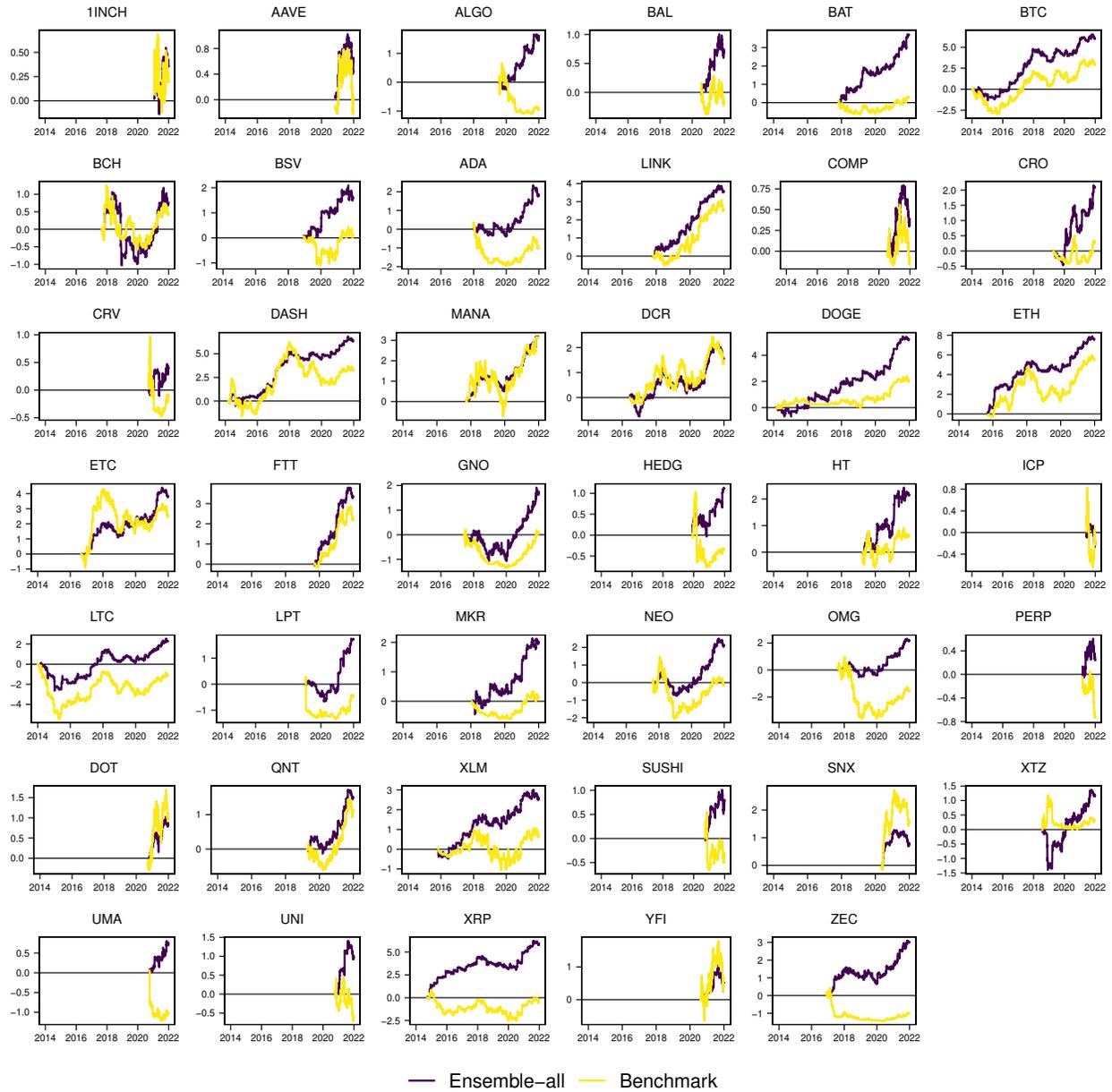


Figure 3: Portfolio Log Cumulative Excess Returns

Each panel depicts the portfolio log cumulative excess return for a mean-variance investor with a coefficient of relative risk aversion of three who allocates daily between the cryptocurrency (denoted by its symbol) in the panel heading and risk-free Treasury bills. The investor allocates their portfolio based on the ensemble-all or prevailing mean benchmark forecast.

January 1, 2014 to December 31, 2021 (matching the forecast evaluation period). To facilitate comparisons across portfolios, we scale the long-short portfolios based on the machine-learning forecasts

Table 7: Performance Metrics

The table reports performance metrics for zero-investment long-short portfolios that invest in multiple cryptocurrencies. The long-short portfolio is constructed by sorting cryptocurrencies according to their excess return forecasts for the available cryptocurrencies in our dataset on a given day based on the machine-learning method in the first column; the portfolio goes long (short) the 30% of cryptocurrencies with the highest (lowest) excess return forecasts. The long and short legs use value (equal) weighting in Panel A (B). Panel C reports performance metrics based on the excess return for a cryptocurrency market portfolio; it is a value-weighted portfolio that invests in all of the available cryptocurrencies on a given day. Panel D reports performance metrics based on the excess return for the CRSP value-weighted aggregate equity market portfolio. The portfolios are scaled to have the same annualized volatility as the equity market portfolio (17.58%). The forecast evaluation period is 2014-01-01 to 2021-12-31.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Forecast	Ann. mean (%)	Ann. Sharpe ratio	Maximum drawdown (%)	Ann. Calmar ratio	Ann. downside risk (%)	Ann. Sortino ratio
Panel A: Value weighting						
Linear-ENet	21.34	1.21	17.81	1.20	11.43	1.87
Random forest	22.54	1.28	18.58	1.21	17.23	1.31
XGBoost	39.00	2.22	12.39	3.15	12.83	3.04
Neural network	18.64	1.06	32.26	0.58	15.84	1.18
Ensemble-nonlinear	37.00	2.10	23.97	1.54	12.66	2.92
Ensemble-all	39.84	2.27	13.40	2.97	10.62	3.75
Panel B: Equal weighting						
Linear-ENet	18.05	1.03	27.83	0.65	13.67	1.32
Random forest	25.23	1.44	14.34	1.76	16.90	1.49
XGBoost	47.20	2.68	9.57	4.93	11.30	4.18
Neural network	20.00	1.14	35.59	0.56	15.84	1.26
Ensemble-nonlinear	45.17	2.57	15.91	2.84	11.58	3.90
Ensemble-all	43.33	2.46	11.95	3.63	10.74	4.04
Panel C: Cryptocurrency market portfolio						
Excess return	19.08	1.09	33.37	0.57	13.67	1.40
Panel D: Equity market portfolio						
Excess return	14.41	0.82	34.35	0.44	15.28	0.94

and the cryptocurrency market portfolio to have an annualized volatility of 17.58%, corresponding to that for the equity market portfolio.²⁴

²⁴Equity market excess return observations are not available for weekends and holidays. The performance metrics for the equity market portfolio in Table 7 are computed using the available observations.

The annualized average excess return for the cryptocurrency market portfolio in Panel C is 19.08%, which is over 450 basis points higher than that for the equity market portfolio in Panel D. With the exception of the neural network (linear-ENet) forecast in Panel A (B), the machine-learning forecasts generate higher average excess returns than the cryptocurrency market portfolio. Indeed, the average excess returns are approximately twice as large or larger for the XGBoost, ensemble-nonlinear, and ensemble-all forecasts for both value and equal weighting. The annualized Sharpe ratio is 1.09 (0.82) for the cryptocurrency (equity) market portfolio. Nearly all of the long-short portfolios based on the machine-learning forecasts generate larger Sharpe ratios than the cryptocurrency market portfolio, and numerous forecasts produce Sharpe ratios that are well above two.

The maximum drawdown for the cryptocurrency market portfolio is 33.37%, which is just below that (34.35%) for the equity market portfolio. With one exception (the neural network forecast in Panel B), the maximum drawdowns for the long-short portfolios are below that for the cryptocurrency market portfolio; in a number of cases, they are less than half as large. The average excess returns and maximum drawdowns typically translate into large Calmar ratios in the fifth column of Table 7. The annualized Calmar ratio for the cryptocurrency (equity) market portfolio is 0.57 (0.44), while the Calmar ratios are much higher for the XGBoost and ensemble-all (XGBoost, ensemble-nonlinear, and ensemble-all) forecasts in Panel A (B), with values of 3.15 and 2.97 (4.93, 2.84, and 3.63), respectively. In many cases, the annualized downside risks for the long-short portfolios in Panels A and B are smaller than those for the cryptocurrency and equity market portfolios (13.67% and 15.28%, respectively). The average excess returns and downside risks lead to annualized Sortino ratios in the last column of Table 7 that are often considerably larger in Panels A and B than those for the cryptocurrency and equity market portfolios (1.40 and 0.94, respectively), especially for the XGBoost, ensemble-nonlinear, and ensemble-all forecasts. In general, Table 7 indicates that the machine-learning forecasts generate substantive improvements in portfolio performance relative to passively holding a value-weighted cryptocurrency or equity market portfolio.

We also test whether the long-short portfolios based on sorted machine-learning forecasts of cryptocurrency returns generate significant alpha in the context of the Liu, Tsyvinski, and Wu (2022) cryptocurrency three-factor model. From a list of leading equity market characteristics,

Liu, Tsyvinski, and Wu (2022) compute cryptocurrency analogs and find that a three-factor model comprised of market, size, and momentum factors can account for cross-sectional cryptocurrency excess returns. We construct cryptocurrency market, size, and momentum factors for the available cryptocurrencies in our dataset along the lines of Liu, Tsyvinski, and Wu (2022). The market factor (CMKT) is the value-weighted excess return for the cryptocurrency market portfolio previously described. The size factor (CSMB) is formed by sorting the available cryptocurrencies according to their market capitalization at the end of the previous day; the size factor is the return on a portfolio that goes long (short) the 30% of cryptocurrencies with the smallest (largest) market capitalization, where the long and short legs are based on value weighting. Finally, for the momentum factor (CMOM), we sort cryptocurrencies according to their cumulative returns over the previous 21 days and compute the return on a portfolio that goes long (short) the 30% of cryptocurrencies with the highest (lowest) cumulative returns, where again use value weighting in the long and short legs.

Multifactor model estimation results are presented in Table 8.²⁵ For value weighting in Panel A, all of the long-short portfolios based on the machine-learning forecasts evince statistically significant exposures to the market factor, but the estimated betas are fairly moderate, ranging from 0.06 (random forecast) to 0.24 (XGBoost). All of the long-short portfolios have significantly negative exposures to the size factor, so the portfolios have a large-cap tilt. In addition, the exposures to the momentum factor are all significantly positive, so the portfolios also have a momentum tilt. The significant factor exposures, however, are typically unable to explain the average excess returns for the long-short portfolios, as the portfolios deliver significant risk-adjusted excess returns. In particular, the portfolio alphas are significant at the 5% level for the neural network forecast and the 1% level for the XGBoost, ensemble-nonlinear, and ensemble-all forecasts; for the latter three forecasts, the annualized alphas are 208%, 196%, and 186%, respectively.

The results for long-short portfolios based on equal weighting in Panel B are similar to those in Panel A, in that the portfolios exhibit significantly positive (negative) exposures to the market and momentum (size) factors and typically generate significant alpha. With the exception of the linear-ENet forecast, the alphas in Panel B are larger than the corresponding values in Panel A, with the XGBoost, ensemble-nonlinear, and ensemble-all forecasts delivering alphas of 248%, 240%, and 204%, respectively, in Panel B.

²⁵The long-short portfolio volatilities are not scaled in Table 8.

Table 8: Multifactor Model Estimation Results

The table reports multifactor model estimation results for zero-investment long-short portfolios that invest in multiple cryptocurrencies. The long-short portfolio is constructed by sorting cryptocurrencies according to their excess return forecasts for the available cryptocurrencies in our dataset on a given day based on the machine-learning method in the first column; the portfolio goes long (short) the 30% of cryptocurrencies with the highest (lowest) excess return forecasts. The long and short legs use value (equal) weighting in Panel A (B). The multifactor model includes the cryptocurrency market (CMKT), size (CSMB), and momentum (CMOM) factors. The forecast evaluation period is 2014-01-01 to 2021-12-31. Standard errors are in parentheses; *, **, and *** indicate significance at the 10%, 5%, and 1% levels, respectively.

(1)	(2)	(3)	(4)	(5)	(6)	(7)
Coefficient	Linear-ENet	Random forest	XGBoost	Neural network	Ensemble-nonlinear	Ensemble-all
Panel A: Value weighting						
$\hat{\alpha}$ (annualized %)	37.65 (43.47)	46.97 (29.07)	207.60*** (48.25)	91.50** (44.31)	196.00*** (48.79)	185.63*** (44.07)
$\hat{\beta}_{\text{CMKT}}$	0.16*** (0.03)	0.06*** (0.02)	0.24*** (0.03)	0.15*** (0.03)	0.22*** (0.03)	0.23*** (0.03)
$\hat{\beta}_{\text{CSMB}}$	-0.12*** (0.02)	-0.15*** (0.02)	-0.48*** (0.03)	-0.15*** (0.02)	-0.44*** (0.03)	-0.43*** (0.02)
$\hat{\beta}_{\text{CMOM}}$	0.65*** (0.02)	0.27*** (0.01)	0.31*** (0.02)	0.09*** (0.02)	0.32*** (0.02)	0.39*** (0.02)
R^2 (%)	32.83	16.50	16.33	2.58	15.09	21.24
Panel B: Equal weighting						
$\hat{\alpha}$ (annualized %)	29.46 (43.35)	55.17** (26.84)	247.55*** (42.57)	99.57** (41.83)	240.12*** (43.35)	203.98*** (41.16)
$\hat{\beta}_{\text{CMKT}}$	0.06** (0.03)	0.04** (0.02)	0.21*** (0.03)	0.09*** (0.03)	0.17*** (0.03)	0.18*** (0.03)
$\hat{\beta}_{\text{CSMB}}$	-0.16*** (0.02)	-0.20*** (0.02)	-0.59*** (0.02)	-0.20*** (0.02)	-0.56*** (0.02)	-0.54*** (0.02)
$\hat{\beta}_{\text{CMOM}}$	0.51*** (0.02)	0.23*** (0.01)	0.23*** (0.02)	0.06*** (0.02)	0.23*** (0.02)	0.30*** (0.02)
R^2 (%)	22.80	16.68	20.92	2.88	18.59	22.16

Complementing the evidence in Tables 3 and 4, Table 8 suggests that nonlinearities are important for forecasting cryptocurrency excess returns. Long-short portfolios based on the linear-ENet forecast fail to produce significant alpha. In contrast, with the exception of the random forest in Panel A, the other forecasts—all of which incorporate nonlinearities in the prediction models—generate significant alpha. Indeed, the alphas are usually much larger in magnitude for forecasts

that incorporate nonlinearities. We investigate nonlinearities in marginal predictive relationships in fitted models in Section 6.

6. Interpretation

Sections 4 and 5 show that machine-learning methods based on a rich set of predictors lead to significant excess return predictability—both statistically and economically—for a large number of cryptocurrencies. In this section, we analyze the role of the predictors in producing out-of-sample forecasts, which allows us to glean insight into the economic sources of cryptocurrency return predictability. We focus on the XGBoost forecast, as it is the most accurate for all of the cryptocurrencies taken together in Table 3; as shown in Tables A1 to A5 and Figures A11 to A15 in the Online Appendix, the conclusions are similar for the other forecasts that incorporate nonlinearities. We use model-interpretation tools based on Shapley (1953) values. Utilizing the analogy between players in a cooperative game earning a payoff and predictors in a model generating a forecast, Štrumbelj and Kononenko (2010, 2014) and Lundberg and Lee (2017) develop Shapley-based measures for interpreting fitted machine-learning models. Because Shapley values have a set of appealing properties, they are generally viewed as providing the best basis for interpreting fitted models (Molnar 2022). We use Shapley-based measures to assess variable (or predictor) importance as well as the strength of nonlinearities in marginal predictive relationships in a fitted model. We compute Shapley-based measures for the fitted XGBoost model that generates the final set of daily cryptocurrency excess return forecasts (for December 31, 2021), as this is the fitted XGBoost model based on the largest training sample of panel data in our application.

First, we compute the variable importance in the fitted XGBoost model for each of the predictors in Table 2. The results are reported in Table 9. To facilitate the interpretation of the model, we combine variable-importance measures across the two transformations (log or simple deviations from 7- and 28-day moving averages), as indicated by “(7,28)” appended to the predictor abbreviations in the first and fourth columns. We also combine the variable-importance measures for the two time-series momentum predictors in levels (TSM7 and TSM28). We scale the variable-importance measures to sum to 100.

Table 9: Variable Importance

The table reports Shapley-based variable-importance (VI) measures for the predictors in the final fitted XGBoost prediction model for cryptocurrency excess returns. The predictors are denoted by their abbreviations in Table 2. We combine the VI measures for TSM7 and TSM28 in levels; we combine the VI measures for all of the predictors when they are transformed using log or simple deviations from 7- and 28-day moving averages, as indicated by “(7,28).” The VI measures are scaled to sum to 100. The second and fifth columns give the category for the predictor in Table 2.

(1) Predictor	(2) Category	(3) VI	(4) Predictor	(5) Category	(6) VI
TSM7, TSM28	Market	17.19	RealNV(7,28)	Transactions	1.44
NV(7,28)	Network	15.69	SER(7,28)	Transactions	1.29
AdrNV(7,28)	Network	10.12	TrfSizeMean(7,28)	Network	1.15
TSM28(7,28)	Market	8.59	NVT(7,28)	Transactions	1.07
TSM7(7,28)	Market	7.82	Sup30d(7,28)	Transactions	1.02
GTSrch(7,28)	Online Activity	5.22	TrfSizeMed(7,28)	Online Activity	0.94
CRNV(7,28)	Transactions	5.14	FacSentArt(7,28)	Online Activity	0.93
NumTrx(7,28)	Transactions	4.34	FacSentHdl(7,28)	Transactions	0.90
FacNumArt(7,28)	Online Activity	3.66	NumTfr(7,28)	Network	0.81
RedNumCom(7,28)	Online Activity	3.63	RedUnc(7,28)	Online Activity	0.72
Vol30d(7,28)	Market	3.10	AdrActNV(7,28)	Online Activity	0.63
Velly(7,28)	Transactions	2.86	FacUncArt(7,28)	Online Activity	0.18
RedSent(7,28)	Market	2.10	FacUncHdl(7,28)	Online Activity	0.11
PrcMA(7,28)	Transactions	1.45			

Time-series momentum in levels is the most important predictor for the fitted XGBoost model. This aligns with Liu and Tsyvinski (2021), who find significant evidence of time-series momentum in cryptocurrency returns on an in-sample basis. Network value (NV) is the next most important predictor, followed by the address-to-network value ratio (AdrNV). These network-related variables can be regarded as capturing aspects of a cryptocurrency’s fundamental value. The importance of NVT is consistent with Liu, Tsyvinski, and Wu (2021), who find evidence of a “value effect” in cross-sectional cryptocurrency returns. TSM7 and TSM28, after transforming them to deviations from 7- and 28-day moving averages, are the fourth and fifth most important predictors, respectively, reinforcing the relevance of time-series momentum. The sixth most important predictor is Google Trends search volume (GTSrch), an attention-based measure that points to the pertinence of online searches for future cryptocurrency returns. The current-to-realized ratio (CRNV) and the number

of transactions (NumTrx) are next. These two predictors reflect activity on a cryptocurrency’s network. Rounding out the top ten are the number of Factiva articles (FacNumArt) and the number of Reddit comments (RedNumCom), further highlighting the relevance of investor attention for cryptocurrency return prediction. Note that the attention-based measures for Reddit and Factiva relating to the number of comments and articles are more important than the sentiment or uncertainty of the comments and articles.

Interestingly, the top ten predictors in Table 9 include multiple predictors from each of the four categories in Table 2. There are three predictors each from the Market and Online Activity categories and two each from the Network and Transactions categories. Thus, cryptocurrency return predictability appears to emanate from a diversity of influences. This is perhaps not surprising. Cryptocurrencies are a new asset class and inherently difficult to value relative to traditional asset classes such as equities and bonds (e.g., Detzel et al. 2021). This creates scope for a variety of predictors to anticipate cryptocurrency returns, as prices take time to move to “fair” values that are difficult to pin down relative to more established asset classes. Along this line, cryptocurrencies are subject to large swings in value and bubble-like behavior relating to investor fads. Indeed, the results in Section 4 indicate that cryptocurrency return predictability tends to be relatively strong around large swings in value.

Finally, we compute Shapley values for the predictors to provide perspective on the importance of nonlinearities in the fitted XGBoost model. The Shapley values are plotted in Figure 4. For each observation in the training sample, the Shapley value measures the contribution of a given predictor to the predicted target value for that observation.²⁶ The Shapley values give a sense of how a fitted model’s prediction changes as the predictor changes. As such, they are informative for helping to identify nonlinearities in a fitted model’s prediction function.

The Shapley values in Figure 4 reveal important nonlinearities in the fitted XGBoost model. In particular, there are a number of strong threshold effects that often characterize fitted models based on decision trees. For example, there is a sharp jump in the predicted return for the smallest values of TSM7, indicating a reversal in the future return when a cryptocurrency experiences a relatively large cumulative loss over the last seven days. As another example, the predicted return increases

²⁶Because there are a very large number of observations (54,535) in the final training sample that is used to fit the XGBoost model, we smooth the plots in Figure 4 using 200 bins.

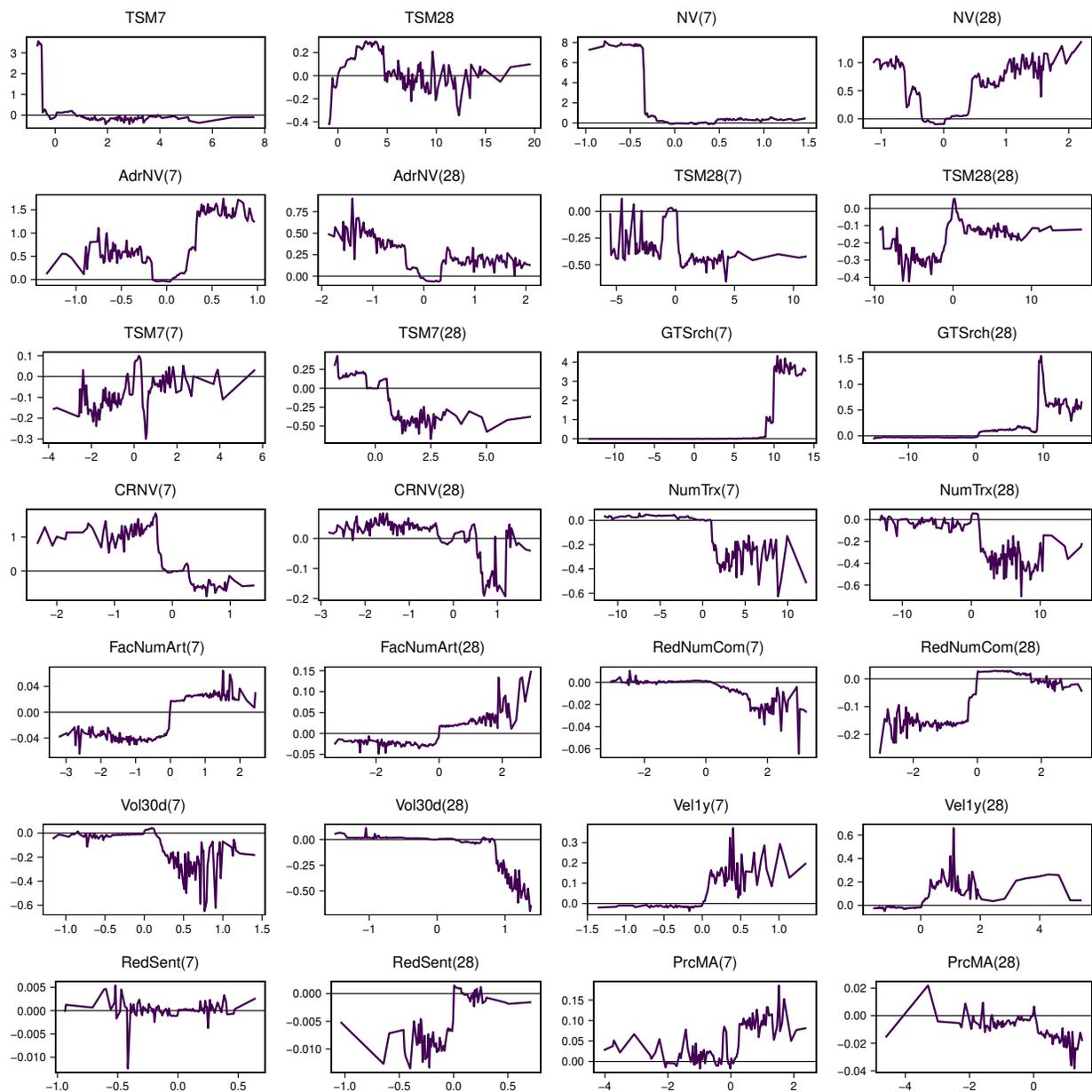


Figure 4: Shapley Values

Each panel depicts Shapley values for the predictor (denoted by its abbreviation in Table 2) in the panel heading. Parentheses indicate that the predictor is transformed using the log or simple deviation from a 7- or 28-day moving average. The Shapley values in the plots are smoothed using 200 bins.

markedly when AdrNV(7) moves above zero to around 0.3, pointing to a strong value effect in this interval. For an example relating to investor attention, the Shapley plot for GTSrch(28) displays a sharp increase near ten, so there is a substantive increase in the predicted return when Google search volume (in log deviation from a 28-day moving average) crosses a threshold of approximately

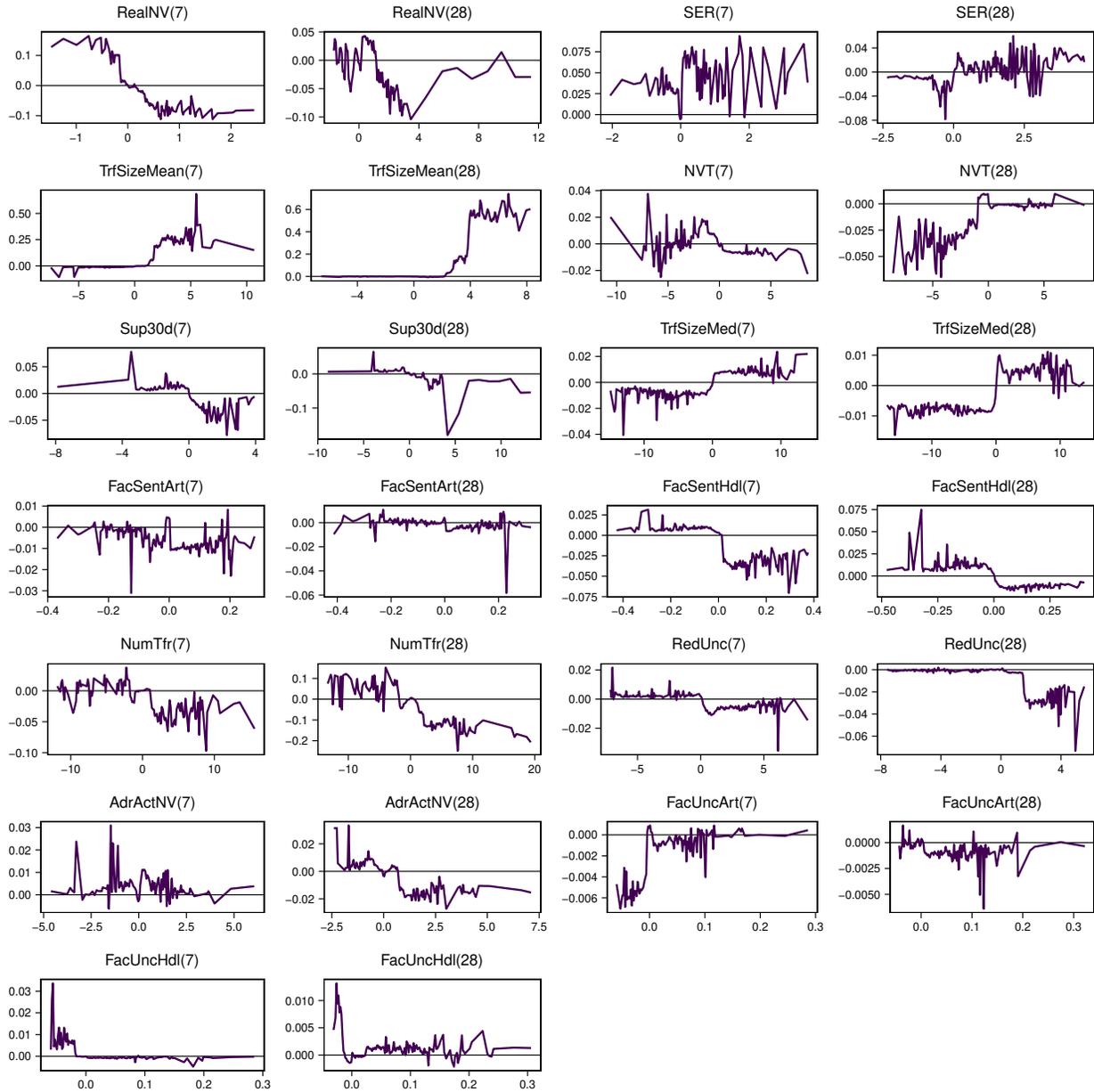


Figure 4 (continued)

ten. In sum, Figure 4 highlights strong nonlinearities, often in the form of threshold effects, in the fitted XGBoost model. Results in Sections 4 and 5 indicate that such nonlinearities improve cryptocurrency return forecasts relative to the linear-ENet forecast.

7. Conclusion

This paper provides an in-depth investigation of the out-of-sample predictability of a large number of daily cryptocurrency excess returns. Our analysis uses a large set of predictors from diverse categories and employs modern machine-learning methods to generate cryptocurrency return forecasts. Incorporating the information in the predictors via machine learning substantially improves out-of-sample forecasts of daily cryptocurrency excess returns. Machine-learning forecasts are significantly more accurate in terms of out-of-sample MSE, and relying on machine-learning forecasts to guide asset allocation provides substantive economic value. Machine-learning methods that allow for nonlinearities in fitted prediction models are especially useful for generating out-of-sample gains. Based on the interpretation of fitted models, a variety of predictors appear relevant for forecasting daily cryptocurrency returns, including predictors relating to time-series momentum, network valuation and activity, and online attention paid to cryptocurrencies. Our results indicate that cryptocurrency return predictability based on a variety of sources is an important stylized fact, at least to this point. The cryptocurrency market is a relatively young market that has experienced fast growth but also large swings in value as well as concerns about its reliability and trustworthiness. As the market matures—assuming that it survives—and investors become more familiar with this new asset class, it will be interesting to monitor the future evolution of cryptocurrency return predictability.

References

- Akiba, T., S. Sano, T. Yanase, T. Ohta, and M. Koyama (2019). Optuna: A Next-Generation Hyperparameter Optimization Framework. In *Proceedings of the 25rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Avramov, D., S. Cheng, and L. Metzker (forthcoming). Machine Learning Versus Economic Restrictions: Evidence from Stock Return Predictability. *Management Science*.
- Bhambhwani, S., S. Delikouras, and G. M. Korniotis (2021). Blockchain Characteristics and the Cross-Section of Cryptocurrency Returns. Working Paper (available at <https://ssrn.com/abstract=3342842>).
- Biais, B., C. Bisiere, M. Bouvard, C. Casamatta, and A. J. Menkveld (forthcoming). Equilibrium Bitcoin Pricing. *Journal of Finance*.
- Breiman, L. (1996). Bagging Predictors. *Machine Learning* 24:1, 123–140.
- Breiman, L. (1997). Arcing the Edge. Technical Report 486, Statistics Department, University of California, Berkeley.
- Breiman, L. (2001). Random Forests. *Machine Learning* 45:1, 5–32.
- Breiman, L., J. Friedman, C. J. Stone, and R. A. Olshen (1984). *Classification and Regression Trees*. New York: CRC Press.
- Campbell, J. Y. and S. B. Thompson (2008). Predicting Excess Stock Returns Out of Sample: Can Anything Beat the Historical Average? *Review of Financial Studies* 21:4, 1509–1531.
- Cheah, J. E.-T., D. Luo, Z. Zhang, and M.-C. Sung (2020). Predictability of Bitcoin Returns. *European Journal of Finance* 28:1, 1–20.
- Chen, L., M. Pelger, and J. Zhu (forthcoming). Deep Learning in Asset Pricing. *Management Science*.
- Chen, T. and C. Guestrin (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.
- Chen, W., H. Xu, L. Jia, and Y. Gao (2021). Machine Learning Model for Bitcoin Exchange Rate Prediction Using Economic and Technology Determinants. *International Journal of Forecasting* 37:1, 28–43.

- Chinco, A., A. D. Clark-Joseph, and M. Ye (2019). Sparse Signals in the Cross-Section of Returns. *Journal of Finance* 74:1, 449–492.
- Cong, L. W., Y. Li, and N. Wang (2021). Tokenomics: Dynamic Adoption and Valuation. *Review of Financial Studies* 34:3, 1105–1155.
- Cybenko, G. (1989). Approximation by Superpositions of a Sigmoidal Function. *Mathematics of Control, Signals, and Systems* 2:4, 303–314.
- Detzel, A. L., H. Liu, J. Strauss, G. Zhou, and Y. Zhu (2021). Learning and Predictability via Technical Analysis: Evidence from Bitcoin and Stocks with Hard-to-Value Fundamentals. *Financial Management* 50:1, 107–137.
- Diebold, F. X. and R. S. Mariano (1995). Comparing Predictive Accuracy. *Journal of Business and Economic Statistics* 13:3, 253–263.
- Dong, X., Y. Li, D. E. Rapach, and G. Zhou (2022). Anomalies and the Expected Market Return. *Journal of Finance* 77:1, 639–681.
- Fama, E. F. and K. R. French (1989). Business Conditions and the Expected Returns on Stocks and Bonds. *Journal of Financial Economics* 25:1, 23–49.
- Filippou, I., D. E. Rapach, M. P. Taylor, and G. Zhou (2022). Out-of-Sample Exchange Rate Prediction: A Machine Learning Perspective. Working Paper (available at <https://ssrn.com/abstract=3455713>).
- Flitter, E. and D. Yaffe-Bellany (2023). FTX Founder Gamed Markets, Crypto Rivals Say. *New York Times*. January 18, 2023.
- Freyberger, J., A. Neuhierl, and M. Weber (2020). Dissecting Characteristics Nonparametrically. *Review of Financial Studies* 33:5, 2326–2377.
- Friedman, J. H. (2001). Greedy Function Approximation: A Gradient Boosting Machine. *Annals of Statistics* 29:5, 1189–1232.
- Friedman, J. H. (2002). Stochastic Gradient Boosting. *Computational Statistics & Data Analysis* 38:4, 367–378.
- Funahashi, K.-I. (1989). On the Approximate Realization of Continuous Mappings by Neural Networks. *Neural Networks* 2:3, 183–192.
- Goodfellow, I., Y. Bengio, and A. Courville (2016). *Deep Learning*. Boston: MIT Press.

- Goyal, A. and I. Welch (2003). Predicting the Equity Premium with Dividend Ratios. *Management Science* 49:5, 639–654.
- Goyal, A. and I. Welch (2008). A Comprehensive Look at the Empirical Performance of Equity Premium Prediction. *Review of Financial Studies* 21:4, 1455–1508.
- Gradojevic, N., D. Kukolj, R. Adcock, and V. Djakovic (2023). Forecasting Bitcoin with Technical Analysis: A Not-So-Random Forest? *International Journal of Forecasting* 39:1, 1–17.
- Griffin, J. M. and A. Shams (2020). Is Bitcoin Really Untethered? *Journal of Finance* 75:4, 1913–1964.
- Gu, S., B. Kelly, and D. Xiu (2020). Empirical Asset Pricing via Machine Learning. *Review of Financial Studies* 33:5, 2223–2273.
- Harvey, C. R., T. A. Zeid, T. Draaisma, M. Luk, H. Neville, A. Rzym, and O. V. Hemert (forthcoming). An Investor’s Guide to Crypto. *Journal of Portfolio Management*.
- Hoerl, A. E. and R. W. Kennard (1970). Ridge Regression: Applications to Nonorthogonal Problems. *Technometrics* 12:1, 69–82.
- Hornik, K. (1991). Approximation Capabilities of Multilayer Feedforward Networks. *Neural Networks* 4:2, 251–257.
- Hornik, K., M. Stinchcombe, and H. White (1989). Multilayer Feedforward Networks Are Universal Approximators. *Neural Networks* 2:5, 359–366.
- Huang, J.-Z., W. Huang, and J. Ni (2019). Predicting Bitcoin Returns Using High-Dimensional Technical Indicators. *Journal of Finance and Data Science* 5:3, 140–155.
- Kingma, D. P. and J. Ba (2015). Adam: A Method for Stochastic Optimization. In *Third Annual International Conference on Learning Representations*.
- Kozak, S., S. Nagel, and S. Santosh (2020). Shrinking the Cross-Section. *Journal of Financial Economics* 135:2, 271–292.
- Liu, Y. and A. Tsyvinski (2021). Risks and Returns of Cryptocurrency. *Review of Financial Studies* 34:6, 2689–2727.
- Liu, Y., A. Tsyvinski, and X. Wu (2021). Accounting for Cryptocurrency Value. Working Paper (available at <https://ssrn.com/abstract=3951514>).
- Liu, Y., A. Tsyvinski, and X. Wu (2022). Common Risk Factors in Cryptocurrency. *Journal of Finance* 77:2, 1133–1177.

- Loughran, T. and B. McDonald (2011). When Is a Liability Not a Liability? Textual Analysis, Dictionaries, and 10-Ks. *Journal of Finance* 66:1, 35–65.
- Lundberg, S. M. and S.-I. Lee (2017). A Unified Approach to Interpreting Model Predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*.
- Maas, A. L., A. Y. Hannun, and A. Y. Ng (2013). Rectifier Nonlinearities Improve Neural Network Acoustic Models. In *Proceedings of the 30th International Conference on Machine Learning*.
- Makarov, I. and A. Schoar (2020). Trading and Arbitrage in Cryptocurrency Markets. *Journal of Financial Economics* 135:2, 293–319.
- Molnar, C. (2022). *Interpretable Machine Learning: A Guide for Making Black Box Models Explainable*. Creative Common License (available at <https://christophm.github.io/interpretable-ml-book/>).
- Moskowitz, T. J., Y. H. Ooi, and L. H. Pedersen (2012). Time Series Momentum. *Journal of Financial Economics* 104:2, 228–250.
- Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program* 14:3, 130–137.
- Prasad, E. S. (2021). *The Future of Money*. Cambridge, MA: Harvard University Press.
- Rapach, D. E., J. K. Strauss, and G. Zhou (2010). Out-of-Sample Equity Premium Prediction: Combination Forecasts and Links to the Real Economy. *Review of Financial Studies* 23:2, 821–862.
- Rolnick, D. and M. Tegmark (2018). The Power of Deeper Networks for Expressing Natural Functions. In *Proceedings of the Sixth Annual International Conference on Learning Representations*.
- Shapley, L. S. (1953). A Value for n -Person Games. *Contributions to the Theory of Games* 2:28, 307–317.
- Srivastava, N., G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov (2014). Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research* 15:56, 1929–1958.
- Štrumbelj, E. and I. Kononenko (2010). An Efficient Explanation of Individual Classifications Using Game Theory. *Journal of Machine Learning Research* 11:1, 1–18.
- Štrumbelj, E. and I. Kononenko (2014). Explaining Prediction Models and Individual Predictions with Feature Contributions. *Knowledge and Information Systems* 41:1, 647–665.

- Tibshirani, R. (1996). Regression Shrinkage and Selection via the LASSO. *Journal of the Royal Statistical Society. Series B (Methodological)* 58:1, 267–288.
- Timmermann, A. (2006). Forecast Combinations. In *Handbook of Economic Forecasting*. Ed. by G. Elliott, C. W. J. Granger, and A. Timmermann. Vol. 1. Amsterdam: Elsevier, pp. 135–196.
- West, K. D. (1996). Asymptotic Inference About Predictive Ability. *Econometrica* 64:5, 1067–1084.
- Zou, H. and T. Hastie (2005). Regularization and Variable Selection via the Elastic Net. *Journal of the Royal Statistical Society. Series B (Statistical Methodology)* 67:2, 301–320.