University of Warsaw | Faculty of Economic Sciences

Working Papers
No. 27/2024 (463)

INFORMER IN ALGORITHMIC INVESTMENT STRATEGIES ON HIGH FREQUENCY BITCOIN DATA

Filip Stefaniuk
Robert Ślepaczuk

Warsaw 2024
ISSN 2957-0506

WORKING PAPERS 27/2024 (463)

UNIVERSITY OF WARSAW
FACULTY OF ECONOMIC SCIENCES
WORKING PAPERS

Informer in Algorithmic Investment Strategies on High Frequency Bitcoin Data

*Filip Stefaniuk[1], Robert Ślepaczuk[2]**

[1] *University of Warsaw, Faculty of Economic Sciences*

[2] *University of Warsaw, Faculty of Economic Sciences, Department of Quantitative Finance and Machine Learning, Quantitative Finance Research Group*

* *Corresponding author: rslepaczuk@wne.uw.edu.pl*

Abstract: The article investigates the usage of Informer architecture for building automated trading strategies for high frequency Bitcoin data. Three strategies using Informer model with different loss functions: Root Mean Squared Error (RMSE), Generalized Mean Absolute Directional Loss (GMADL) and Quantile loss, are proposed and evaluated against the Buy and Hold benchmark and two benchmark strategies based on technical indicators. The evaluation is conducted using data of various frequencies: 5 minute, 15 minute, and 30 minute intervals, over the 6 different periods. Although the Informer-based model with Quantile loss did not outperform the benchmark, two other models achieved better results. The performance of the model using RMSE loss worsens when used with higher frequency data while the model that uses novel GMADL loss function is benefiting from higher frequency data and when trained on 5 minute interval it beat all the other strategies on most of the testing periods. The primary contribution of this study is the application and assessment of the RMSE, GMADL and Quantile loss functions with the Informer model to forecast future returns, subsequently using these forecasts to develop automated trading strategies. The research provides evidence that employing an Informer model trained with the GMADL loss function can result in superior trading outcomes compared to the buy-and-hold approach.

Keywords: Machine Learning, Financial Series Forecasting, Automated Trading Strategy, Informer, Transformer, Bitcoin, High Frequency Trading, GMADL

JEL codes: C4, C14, C45, C53, C58, G13

Note: This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors

1 Introduction

Developing automated trading strategies is a challenging task that has captivated both institutional investors and individual researchers for a long time. With the rise of computing power and development of machine learning (ML), more and more automated algorithms are being developed and deployed into the market. It is estimated that currently 70% to 80% of all market transactions are carried out by automated trading software (Yadav 2015) and this number is expected to increase in the coming years.

Although numerous systems built in the past have handled daily market data, this is insufficient in the age of High-Frequency Trading and advanced computing power. This is especially true in the context of trading assets such as Bitcoin. Bitcoin (BTC) is a decentralized digital currency that allows peer-to-peer transactions over the Bitcoin network, which is a secure, public ledger using blockchain technology. It was first introduced in an anonymous 2009 white paper (Nakamoto 2009), by a person or group of people using the pseudonym Satoshi Nakamoto. Since then, Bitcoin has grown in popularity and many cryptocurrency exchanges that facilitate bitcoin sales and purchases have emerged. This asset is known for its high volatility, for example, in 2017 the Bitcoin price went up nearly 20 times, reaching \$19,497 and then dropping by 84%, in 2021 Bitcoin reached a new all-time high of \$63,314 but then the price fell to \$34,770, almost 55%[1]. Since then, Bitcoin has become somewhat more stable, especially after approval of Bitcoin Spot ETFs, although price movements are still greatly exaggerated compared to more classical investment instruments.

The study explores the idea of building an automated trading strategy for Bitcoin. Five strategies are proposed and evaluated on the historical Bitcoin data of high frequencies: 5 minute, 15 minute and 30 minute; from a period of 21.08.2019 to 24.07.2024. The first two strategies are treated as benchmarks and are based on classical technical indicators, namely the Moving Average Convergence Divergence (MACD) and Relative Strength Index (RSI). The other three employ the Informer (Zhou et al. 2021), a state-of-the-art attention-based neural network model designed to efficiently handle long time series, to predict the returns and subsequently choose positions according to the model's forecasts.

The work aims to answer the following research questions:

Q: *Is it possible to create an algorithmic strategy for trading Bitcoin, that is more efficient than Buy&Hold approach?*

Q: *Does signal from Informer model allow to create strategies that are more efficient on trading Bitcoin than strategies based on technical indicators?*

Q: *How does selection of the machine learning model loss function influence the strategy performance?*

Q: *Does usage of higher frequency data allow to create more efficient strategies?*

Answering those questions required designing and implementing a method to compare various trading strategies. Such comparison is a main contribution of this study. To the best of current knowledge, no other research has yet been performed where an Informer model is trained with the Quantile or GMADL (Michańków et al. 2024) loss function, followed by the utilization of its forecasts in buy/sell signals generation to develop automated trading strategies. The study involves an exhaustive analysis of the approach including the comparison to the benchmark strategies: buy-and-hold and two technical indicator based strategies, usage of data with different time intervals and multiple time periods. Finally, a sensitivity analysis is conducted to show how changing the parameters affect the perfomance of the tested strategies. An additional contribution of the research is an open sourced implementation of framework for efficient comparison of trading strategies, that is available on Gitlab and enables reproducing the results[2].

The structure of the thesis is as follows: Chapter 2 surveys the associated literature and gives a brief overview of prior similar studies. Chapter 3 explains the acquisition, preprocessing, and covers the analysis of the datasets used in the study. In Chapter 4, the notion of trading strategy is formally defined, the metrics for comparison are introduced, and the details of each examined strategy are outlined. Chapter 5 displays the experimental results, detailing the selection of strategy hyperparameters and presenting the outcomes of the strategy evaluation along with sensitivity analysis. Lastly, Chapter 7 draws the conclusions of the study.

[1]https://blog.obiex.finance/6-biggest-bitcoin-crashes-that-have-happened-in-crypto-history/
[2]https://gitlab.com/FilipStefaniuk/wne-msc-thesis

2 Literature Review

The study of financial market efficiency has a rich history. For years, researchers have been trying to develop algorithms capable of predicting the price movements of financial assets as well as creating automated trading strategies that rely on such signals. The concept of developing an automated system capable of regularly outperforming the market is so attractive that it leads to the publication of thousands of papers on that topic annually.

2.1 Efficient Market Hypothesis

These endeavors stand in contrast to the Efficient Market Hypothesis (EMH), also known as the efficient market theory, which argues that the stock prices already reflect all accessible information. EMH suggests that it is impossible to consistently achieve higher returns than average market returns, thus the best investing strategy is a passive portfolio. The hypothesis was formulated in three forms: *weak*, *semi-strong*, and *strong*. The *weak* form challenges the foundation of technical analysis by suggesting that stock prices move in a random manner. The *semi-strong* form argues that stock prices immediately incorporate all public information, arguing for the ineffectiveness of the fundamental analysis. The *strong* form extends this argument, implying that the price of the stock includes both public and private information. The EMH, even in its weakest form, implies that the unpredictability of short-term price fluctuations makes efforts to create automated trading systems ineffective (Malkiel 1973).

The EMH was acknowledged in the early review of theoretical and empirical market models by Fama (1970). The author put forth certain evidence indicating the presence of a statistically significant positive correlation in daily returns, suggesting that such patterns might be exploited to develop profitable trading strategies. However, they ultimately concluded that this evidence might not be robust enough to challenge the validity of the EMH. This research was continued in Fama (1991), which was published two decades later. In this publication, the author presented additional evidence for predicting daily and weekly returns based on historical returns, and the findings align with those of the earlier study.

The argument in favor of EMH was presented in Malkiel (2005). The publication demonstrated that over a long period, professional investment managers do not outperform their index benchmarks. The study showed that in a 20-year period 90% of managed index funds, was outperformed by the S&P 500 index, asserting that the only valid long-term strategy is the passive portfolio.

Conversely, the most persistent criticisms of the EMH relate to the preferences and actions of market participants. Psychologists and experimental economists recorded numerous specific behavioral biases that are prevalent in human decision making when faced with uncertainty. They argue that irrational behaviors such as overconfidence (Fischhoff and Slovic 1978; Barber and Odean 2001; Gervais and Odean 2001), overreaction (Bondt and Thaler 1985), loss aversion (Kahneman and Tversky 1979; Odean. 1996), herding (Huberman and Regev 2001), and psychological accounting (Tversky and Kahneman 1981) are predictable and can be utilized to develop profitable trading strategies that exceed passive benchmarks.

Despite extensive research, the validity of the EMH continues to be debated without a definitive conclusion. More modern approaches explore the idea of time-varying weak-form market efficiency (Lim and Brooks 2011) and the Adaptive Markets Hypothesis (Kim et al. 2011) arguing that the predictability of markets can change over time and depend on various conditions such as the composition of market participants.

2.2 Early Statistical Approach

Nevertheless, a number of statistical systems and techniques that attempt to analyze and forecast asset prices have been developed. The Box-Jenkins method applies autoregressive moving average (ARMA) or autoregressive integrated moving average (ARIMA) models to find the best fit of a time series model to past values of a time series (Box and Jenkins 1976). The model combines autoregression, differencing, and moving averages to predict stock prices, and despite its simplicity, it has been widely applied and successful in financial forecasting. Another method called the exponential smoothing model (ESM) employs the exponential window function to smooth the data in which exponential smoothing parameters are estimated (Billah et al. 2006). Furthermore, those approaches were often combined with several other classical Machine Learning non-linear models.

In the study by Zhang (2003), authors combined the ARIMA model with Artificial Neural Networks (ANNs) to predict the prices of pairs of currencies. Similar approach was further explored in Khashei and Bijari (2011), in which authors use ARIMA-ANN hybrid model to further improve predictions on the aforementioned currencies datasets. In Pai and Lin (2005), the authors conducted a comparison between the ARIMA model and Support Vector Machines (SVMs) across diverse stock prices, demonstrating that both methodologies successfully forecasted the prices. In Wang et al. (2012) the authors developed a hybrid approach that uses ESM, ARIMA, and ANN to predict daily prices of the Dow Jones Industrial Average and Shenzhen index. Adebiyi et al. (2014) utilized the ARIMA model in data sets from the New York and Nigeria Stock Exchanges, illustrating its effectiveness in forecasting short-term prices, and compared its results with ANN. Azari (2018) illustrated the effectiveness of the ARIMA model in forecasting Bitcoin's price by examining its predictions on a time series over a three-year period. However, they admitted that the approach was unable to predict abrupt price fluctuations, such as volatility observed in late 2017. Võ and Ślepaczuk (2022) utilized a combination of ARIMA and GARCH family models to predict S&P500 log returns, employing these predictions to develop a trading strategy. Their findings indicated that strategies incorporating hybrid models outperformed those relying only on the ARIMA model.

2.3 Early Machine Learning Systems

Advancements in computing power lead to wider adoption of ML models and development of more complex neural network architectures. Models such as Convolutional Neural Networks (CNNs) (LeCun et al. 2015) or Recurrent Neural Networks (RNNs) (and their variants Long-Short Term memory Networks (LSTMs) (Hochreiter and Schmidhuber 1997) and Gated Recurrent Units(GRUs) (Cho et al. 2014)) were developed. Initially, they were used mainly in the computer vision and natural language processing domains, but they got quickly adapted and successfully implemented for financial forecasting.

Chen et al. (2015) modeled and predicted China stock market daily returns with LSTM, showing it outperforms previous classic statistical approaches. Honchar and Di Persio (2016) compared how different RNN architectures are able to forecast price movements. They used Google stock price index data and applied multi-layer RNN, LSTM and GRU, to predict prices at different time horizons. The study showed that LSTM outperformed other architectures for long sequences. In the other publication Persio et al. (2017), the same authors compared Multi Layer Perceptron (MLP), CNN with LSTM architectures for predicting daily S&P500 returns and minute-by-minute FOREX EUR/USD returns. They reported that CNN can model financial time series better than other architectures. Fischer et al. (2017) utilized an LSTM model for selecting S&P 500 stocks in portfolio construction. The data came from 1992 until 2015, and the goal was formulated as a classification problem. The researches reported that the LSTM outperformed classical ML methods like random forest and logistic regression classifier. Moreover, they attempted to disentangle the black-box of LSTMs predictions by analyzing the selected stocks. They discovered that the selected stocks consist mainly of stocks with below-mean momentum, strong short-term reversal characteristics, and high volatility. Grudniewicz and Ślepaczuk (2023) applied several classical Machine Learning algorithms to technical analysis indicators for the WIG20, DAX, S&P 500, and a few selected CEE indices. The findings of the study demonstrate that quantitative methods outperform passive strategies when considering risk-adjusted returns, with the Bayesian Generalized Linear Model and Naive Bayes emerging as the leading models for the examined indices.

Selvin et al. (2017) predicted prices for 2000 NSE listed companies with LSTM, RNN, and CNN models. They identified CNN as the best model that turned out to be best in predicting the changes in trends. Nelson et al. (2017) used LSTM in combination with technical analysis indicators. They reported an average accuracy of 55.9% when predicting if the price of a particular stock is going to rise or not in the near future. Hossain et al. (2018) proposed model integrating LSTM and GRU architectures to predict S&P 500 prices. In this method, the input data is initially fed into the LSTM network to produce a preliminary prediction, after which the output from the LSTM layer is forwarded to the GRU layer for generating the final prediction. The research demonstrates that the suggested hybrid model surpasses the performance of standalone LSTM and GRUs. Siami-Namini et al. (2018) examined how deep learning-based time series forecasting algorithms such as LSTM compare to traditional-based algorithms such as the ARIMA model. The average reduction in error rates obtained by LSTM was found to be between 84 and 87 percent compared to ARIMA, indicating the superiority of LSTM over ARIMA. Similarly Kijewski et al. (2024) evaluated the effectiveness of investment strategies based on traditional models and an LSTM on the S&P 500 index time series, spanning 20 years of data from 2000 to 2020. The combination of signals from several methods doubled the returns on the same level of risk of the passive strategy benchmark. The study, however, concluded that the LSTM model exhibited a considerably lower robustness to parameter variations compared to traditional methods.

2.4 Modern Machine Learning Systems

Although RNNs became widely adopted and proved to be effective for sequential data, the architecture struggled with processing longer sequences due to exploding and vanishing gradient phenomena (Basodi et al. 2020). In order to overcome this limitation, the Attention mechanism (Bahdanau et al. 2014; Luong et al. 2015) was developed.

While initially it was used in combination with the RNNs, Vaswani et al. (2017) introduced a neural network architecture based solely on the Attention mechanism called Transformer. The newly introduced architecture exhibited a significant enhancement in the performance, leading to its widespread adoption across multiple domains. Nowadays, it lies at the heart of state-of-the-art large-scale, multimodal models such as GPT-4 (OpenAI et al. 2024) or Gemini (Team et al. 2024).

The time series forecasting domain also adopted the new architecture. The Benidis et al. (2023) collects general purpose state-of-the-art time-series forecasting architectures developed in recent years. Among the first of the modern deep forecasting models is DeepAR (Salinas et al. 2019) which still is primarily based on RNNs. The model outputs parameters of a previously chosen family of distributions. Samples from this distribution can be fed back into the model during prediction. However, most of the other modern models fully adapt the new Transformer architecture. Temporal Fusion Transformer (TFT) developed by Lim, Arik, et al. (2020), incorporates Transformer architecture with novel components for embedding static covariates, performing "variable selection", and gating components that skip over irrelevant parts of the context. The TFT is trained to predict forecast quantiles, and promotes forecast interpretability by modifying self-attention and learning input variable importance. Eisenach et al. (2020) proposed MQ-Transformer, a Transformer architecture that employs attention mechanisms in the encoder and decoder separately, and consider learning positional embeddings from event indicators. The authors discussed improvements not only on forecast accuracy but also on excess forecast volatility where their model improves over the state-of-the-art. Zhou et al. (2021) proposed the Informer, a computationally efficient Transformer architecture, that specifically targets applications with long forecast horizons. These robust novel architectures can be specifically utilized for forecasting financial time series, particularly when dealing with higher-frequency data.

Barez et al. (2023) explored the application of deep learning Transformers architectures for high-frequency Bitcoin-USDT log-return forecasting and compared them to the LSTM model. The results indicate that the model based on Transformer achieves a higher cumulative PnL than the LSTM when trading with multiple signals during backtesting. Hu and Hu (2021) applied TFT for stock price predictions comparing it with LSTM and again showing the superiority of Transformer model. Zhao et al. (2022) investigated the prediction capability of the Transformer model on Bitcoin and Ethereum price data, furthermore augmenting the model with sentiment data collected from Twitter. Similarly Hájek and Novotny (2024) integrated news sentiment and investor attention metrics with a Temporal Fusion Transformer framework. The effectiveness of the model was demonstrated by analyzing stock price data for 41 of the largest market capitalization companies over the period from 2010 to 2021.

Penmetsa and Vemula (2023) analyzed the effectiveness of deep learning to predict Bitcoin, Ethereum and Litecoin prices, by integrating momentum and volatility technical indicators into network's input. They conducted a study using LSTM and Transformer architectures, concluding that Transformers tend to outperform LSTM models in price prediction and trends in cryptocurrency data. Wang (2023) introduced hybrid approach combining Transformer architecture with bi-LSTM. The authors argued that Transformer is better at obtaining long distance information, while LSTM can better capture short distance signals. They presented the effectiveness of their method using 5 index stocks and 14 Shanghai and Shenzhen stocks.

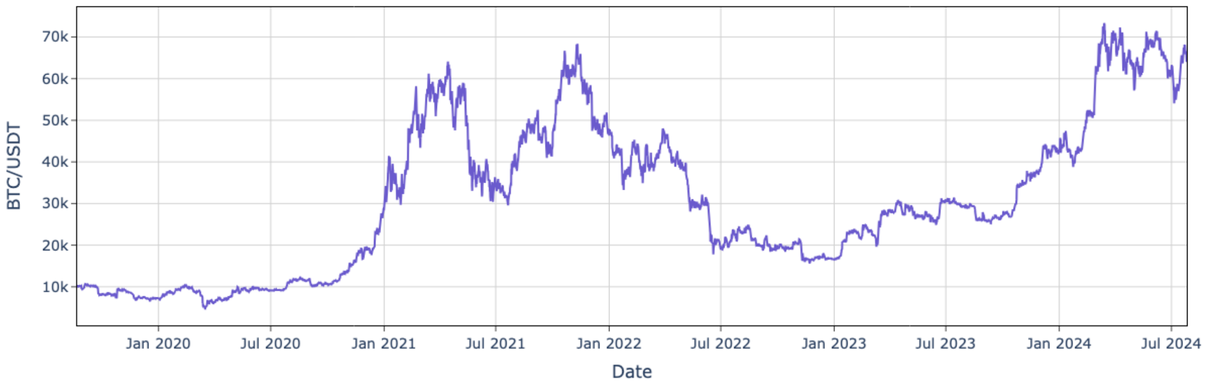
The Informer architecture was used by Ding et al. (2023) to predict intra-day Chinese stock price. They also did a comparison with LSTM model showing that Informer outperformed the RNN network. Another such comparison was done by Duan and Ke (2024) who systematically compared the effectiveness of Informer and LSTM model for price predictions concluding that while in general Informer outperforms LSTM, the fusion of the two combines the advantages of both models and enhances the prediction accuracy. Lastly Lu et al. (2023) designed three experiments to compare Informer with the commonly used networks of LSTM, Transformer and BERT (Devlin et al. 2018) on 1-minute and 5-minute frequencies for four different stocks and market indices. The prediction results are measured by three evaluation criteria: MAE, RMSE and MAPE. Informer obtained the best performance among all the networks on every dataset.

The chapter provided an overview of the historical development of research aimed at predicting financial asset movements. It emphasized several significant publications within this continually expanding area of study. Although financial forecasting has consistently held popularity, recent years have witnessed an intensified focus on predictions conducted at higher frequencies, aided by the utilization of increasingly sophisticated systems akin to the one presented in this study.

3 Data

The data used in the research consider the BTC/USDT cryptocurrency pair from a period of 21.08.2019 to 24.07.2024 (5 years). Bitcoin (BTC) is a decentralized digital currency that allows peer-to-peer transactions. Tether (USDT) is a type of stablecoin, with a value that closely mirrors that of the USD. It actively works to keep its valuation stable through market mechanisms: each Tether issued is backed by one US dollar worth of assets. The pair closely mirrors the behaviour of BTC/USD however it has the higher trading volume, as it is used by investors who want to hedge against the inherent volatility of their cryptocurrency investments while still keeping value within the crypto market.

Figure 1: Price of the BTC/USDT



Note: Price of BTC/USDT cryptocurrency pair in a period from 21.08.2019 to 24.07.2024. The data was obtained from https://www.binance.com/en-NG/landing/data.

The selection of this data for the research was driven by the following motivations:

- Availability of high quality historical data. Many cryptocurrency exchanges provide free access to historical data, with fine-grained intervals, even up to 1 second. The data is easily accessible through either a dedicated API or can be downloaded as csv files. This is a significant difference compared to availability of historical stock prices, which are not easily accessible even for hourly interval data, and usually the access requires a fee. Additionally, this concrete pair BTC/USDT was one of the earliest pairs available on the exchanges, so it has the longest historical data available.

- Cryptocurrency exchanges operate continuously, 24 hours a day, 7 days a week. Hence, there is no need for special handling of holidays and overnight price fluctuations during stock market closures. The historical price data forms a consistent time series with equally spaced intervals.

- Bitcoin stands as the most established cryptocurrency, with the BTC/USDT pair being the most widely traded and one of the highest in trading volume [3], which makes it more liquid and stable compared to the other cryptocurrency pairs.

The data was obtained through the Binance API[4]. Binance is one of the largest global cryptocurrency exchanges that provides a platform to trade various cryptocurrencies. It was established in 2017 and offers a comprehensive range of services, including trading, listing, fundraising, and delisting or withdrawal of cryptocurrencies. The API allows downloading the so-called k-line data, the candlestick data of various intervals, consisting of: `open time`, `close time`, `open price`, `high price`, `low price`, `close price` and `volume`. The research uses k-line intervals of `5min`, `15min` and `30min`.

After downloading, the data was checked for missing data points. Even though exchange operates continuously, there were maintenance windows and the exchange sometimes was down. 8 gaps in the data were detected, with missing data of over 101 hours (4.2 days) in total. This constitutes to 0.002 of all data points in the period, thus it shouldn't have significant impact into the experiment. For the

[3] https://coinranking.com/exchange/-zdvbieRdZ+binance/markets
[4] https://www.binance.com/en-NG/landing/data

simplicity sake and to make the time series consistent, the gaps were filled by copying the price and volume values from the last available data point. Then, returns from each interval are computed as:

$$\texttt{returns} = \frac{\texttt{close price} - \texttt{open price}}{\texttt{open price}} \tag{1}$$

Table 3 presents the descriptive statistics of the BTC/USDT returns for all three intervals. We can observe, that the standard deviation of the returns grows with the interval length, which is expected. The minimum and maximum returns for each interval are quite alike, suggesting that significant price fluctuations occur abruptly and swiftly. Interestingly, the maximum return was observed for the `15min` interval, indicating that the price must first have quickly increased and then dropped (in the span of 30 min). The distributions are leptokurtic, with kurtosis significantly higher than the normal distributions, that increases with smaller intervals. In all cases skewness is between -1 and 1 which indicates a relatively symmetrical distribution. Futhermore, in all three cases Kolmogorov-Smirnov test for normality has a test statistic equal to 0.49 and a p-value of 0.0. Therefore, a null hypothesis that the data is normally distributed is rejected. The given analysis underscores the significance of investigating and creating trading systems that function in shorter time intervals. This is because rapid and abrupt price movements, which are critical for effective trading strategies, would go unnoticed by algorithms that operate on longer intervals.

Table 1: Descriptive statistics of BTC/USDT

Statistic	**BTC/USDT 5min**	**BTC/USDT 15min**	**BTC/USDT 30min**
count	518400	172800	86400
mean	0.0000060	0.0000176	0.0000346
std	0.0021843	0.0036712	0.0050768
min	-0.1022537	-0.1191688	-0.1662924
25% percentile	-0.0007716	-0.0013018	-0.0017677
50% percentile	0	0.0000094	0.0000280
75% percentile	0.0007855	0.0013443	0.0018575
max	0.1842885	0.2262878	0.1460125
kurtosis	203.12	140.22	58.61
skewness	0.57	0.97	-0.29
KS test stat.	0.49	0.49	0.49
KS test p-value	0.00e+00	0.00e+00	0.00e+00

Note: Descriptive statistics of returns with intervals of `5min`, `15min` and `30min`. The statistics are not annualized. Null hypothesis of Kolmogorov-Smirnov (KS) test is that the distribution is normal.

3.1 Additional Data

The research utilizes data collected over several years, encompassing periods with varying market conditions and sentiments. To account for this, the data was enhanced with additional information that attempts to capture this information.

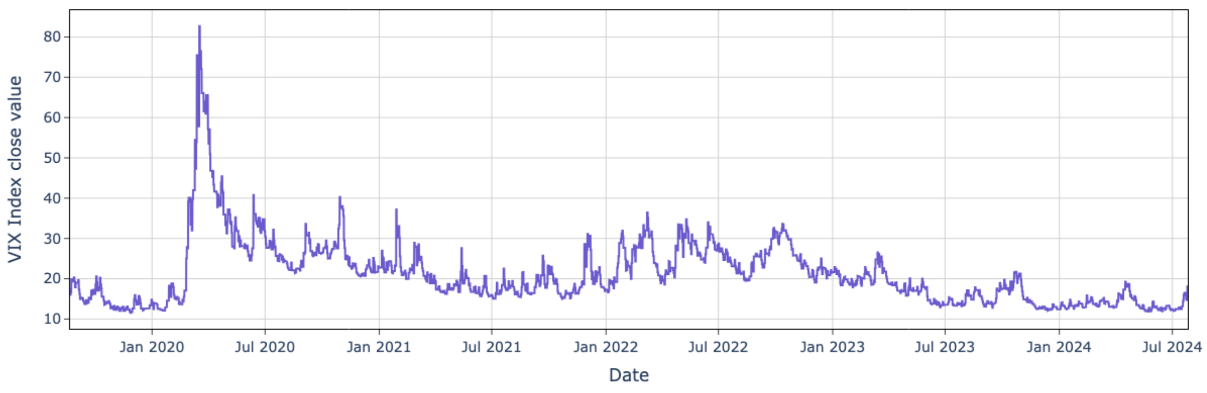
Additional data that was added composes of:

The Cboe Volatility Index (VIX Index)[5], a benchmark index that measures the market's expectation of future volatility. The VIX Index is based on options of the S&P 500 Index and is considered the leading indicator of the broad U.S. stock market. The VIX Index is recognized as the world's best gauge of U.S. equity market. The daily close value of the VIX index is used.

The Federal Funds effective rates[6], that is the interest rate at which depository institutions trade federal funds (balances held at Federal Reserve Banks) with each other overnight. The federal funds rate indirectly influences longer-term interest rates such as mortgages, loans, and savings. The data was available at the monthly frequency.

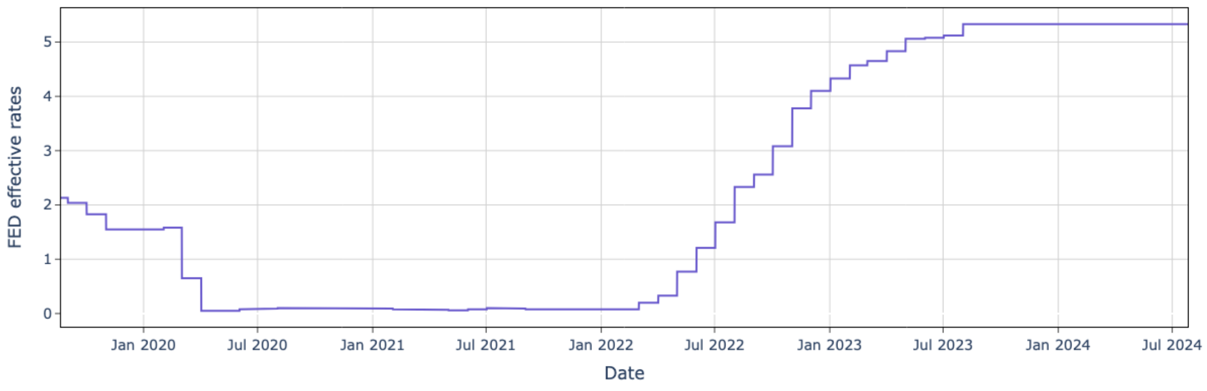
[5]https://www.cboe.com/tradable_products/vix/vix_historical_data/
[6]https://fred.stlouisfed.org/series/FEDFUNDS

Figure 2: Cboe Volatility Index (VIX)



Note: Cboe Volatility Index (VIX) value in a period from 21.08.2019 to 24.07.2024. Data was obtained from https://www.cboe.com/tradable_products/vix/vix_historical_data/

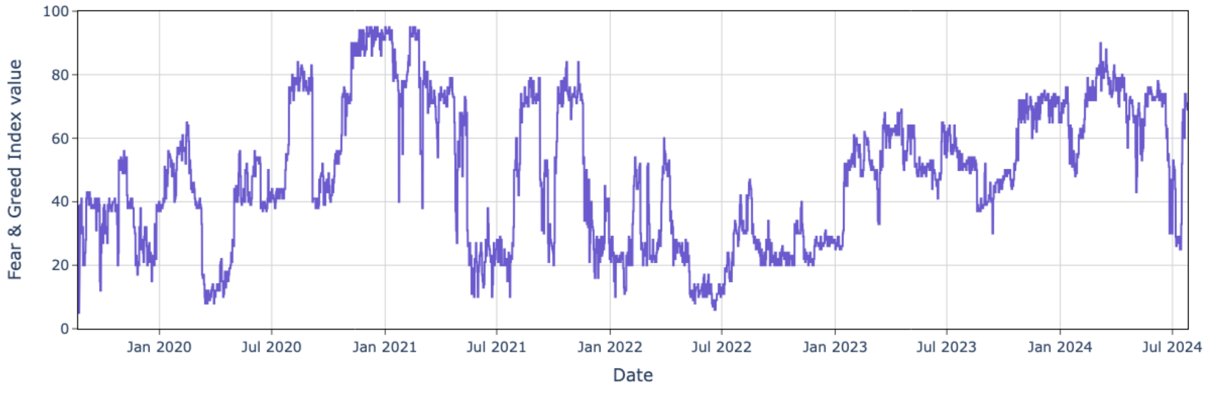
Figure 3: Note: The Federal Funds effective rates



Note: The federal funds effective rates in a period from 21.08.2019 to 24.07.2024. Data was obtained from https://fred.stlouisfed.org/series/FEDFUNDS

Crypto Fear/Greed index[7], which is founded on elements such as volatility, market momentum, reddit sentiment analysis, Google trends data and public surveys. The index is available at the daily frequency.

Figure 4: The Crypto Fear/Greed Index



Note: The Crypto Fear/Greed Index value in a period from 21.08.2019 to 24.07.2024. Data was obtained from https://alternative.me/crypto/fear-and-greed-index/

The aforementioned data was collected at a much lower frequency (daily/monthly) compared to the BTC/USDT data. Consequently, for every observation, a last known value had to be assigned, e.g.

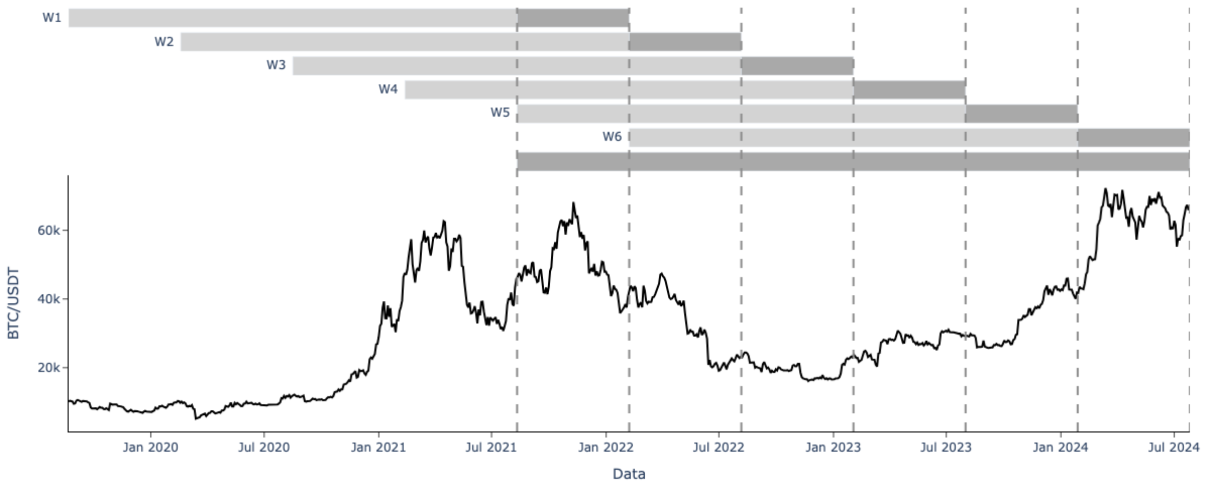
[7]https://alternative.me/crypto/fear-and-greed-index/

a `5min` observation with `open time` of 2022-08-17 04:05:00 was enriched with value of VIX Index and Crypto Fear/Greed Index from 2022-08-16 and effective rates value from 2022-07. Note that the values from the previous day/month are used as it is unknown at what hour/day the value was published. Using the information from the same day/month carries a potential risk of using the information that wasn't available at the time.

3.2 Data Windows

The research follows the approach of Michańków et al. 2022, where strategies are independently evaluated on a rolling window that passes through the testing period. A rolling window consists of an *in sample* part of the 24 months (2 years) and *out of sample* part of 6 months. Each next window is moved forward by the length of the *out of sample* part. In the end, the results of each subperiod are combined to provide testing results for the whole testing period. This method allows to better evaluate the robustness of the trading strategies, as it evaluates them independently on various periods, when the market conditions are different.

Figure 5: Rolling data windows



Note: The figure presents how the dataset was split into rolling data windows consisting of *in sample* and *out of sample* parts.

In total six windows with identical number of data points were created. The *out of sample* is used for testing, further referenced as the *test* part. The *in sample* part is split into *train* and *validation* parts, with the *validation* part being 20% of the *in sample* data.

Table 2: Number of data points

Part	**BTC/USDT 5min**	**BTC/USDT 15min**	**BTC/USDT 30min**
train	165888	55296	27648
validation	41472	13824	6912
test	51840	17280	8640

Note: Number of data points in each of the dataset parts for the `5min`, `15min` and `30min` rolling windows.

Tables 3, 3 and 5 further detail the descriptive statistics for each *out-of-sample* windows for `5min`, `15min` and `30min` interval data. The statistics are consistent with the ones computed for the whole period (Table 3). Standard deviation is higher for the first two windows in all frequencies, suggesting that the market was more volatile during those periods. Kurtosis is significantly higher than the normal distributions with the values largest for 1st and 5th window of `5min` data and 5th window for `15min` data, indicating many data points with returns close to zero in those windows. The Kolmogorov-Smirnov tests for normality indicate that none of the distributions is normal.

The differences between *out-of-sample* data windows are best visible in Figure 6. Heatmaps on the left side of Figure 6 show the distances between the returns distributions between each *out-of-sample* window, according to the Wasserstein metric (Vaserstein 1969). For all the frequencies, the most significant shift in the distributions can be observed between first two and the next three windows, and then with the last, sixth, window. The violin plots of the distributions on the right-hand side of the figure provide additional information about the shapes of the distributions.

Table 3: Descriptive statistics for 5 min out-of-sample data

Statistic	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**
count	51840	51840	51840	51840	51840	51840
mean	0.0000010	-0.0000089	0.0000016	0.0000050	0.0000080	0.0000099
std	0.0021663	0.0023011	0.0015967	0.0013956	0.0013791	0.0016259
min	-0.0989054	-0.0438821	-0.0469833	-0.0263642	-0.0866874	-0.0370693
25% percentile	-0.0010410	-0.0009298	-0.0005471	-0.0005338	-0.0004952	-0.0006930
50% percentile	-0.0000139	-0.0000029	0.0000070	0	0	0.0000064
75% percentile	0.0010127	0.0009137	0.0005638	0.0005350	0.0005185	0.0007262
max	0.0643117	0.0426144	0.0487321	0.0200162	0.0501479	0.0189663
kurtosis	124.60	27.33	72.17	27.74	419.46	18.82
skewness	-1.41	0.61	-0.27	-0.27	-3.84	-0.59
KS test stat.	0.49	0.49	0.50	0.50	0.50	0.50
KS test p-value	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00

Note: Descriptive statistics of returns in data windows of 5min interval data. The statistics are not annualized. Null hypothesis of Kolmogorov-Smirnov (KS) test is that the distribution is normal.

Table 4: Descriptive statistics for 15 min out-of-sample data

Statistic	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**
count	17280	17280	17280	17280	17280	17280
mean	0.0000024	-0.0000276	0.0000048	0.0000149	0.0000240	0.0000296
std	0.0035625	0.0038529	0.0027021	0.0023567	0.0023828	0.0027726
min	-0.0675796	-0.0470951	-0.0502659	-0.0276761	-0.0897967	-0.0346516
25% percentile	-0.0017774	-0.0015974	-0.0009356	-0.0008720	-0.0008726	-0.0012156
50% percentile	-0.0000367	0.0000399	0.0000138	-0.0000080	0.0000222	0.0000141
75% percentile	0.0017230	0.0015821	0.0009610	0.0009004	0.0009122	0.0012940
max	0.0366072	0.0581246	0.0435405	0.0331951	0.0554152	0.0281460
kurtosis	19.08	19.40	38.81	21.94	176.31	11.54
skewness	-0.30	0.40	-0.64	0.22	-2.28	-0.31
KS test stat.	0.49	0.49	0.49	0.49	0.49	0.49
KS test p-value	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00

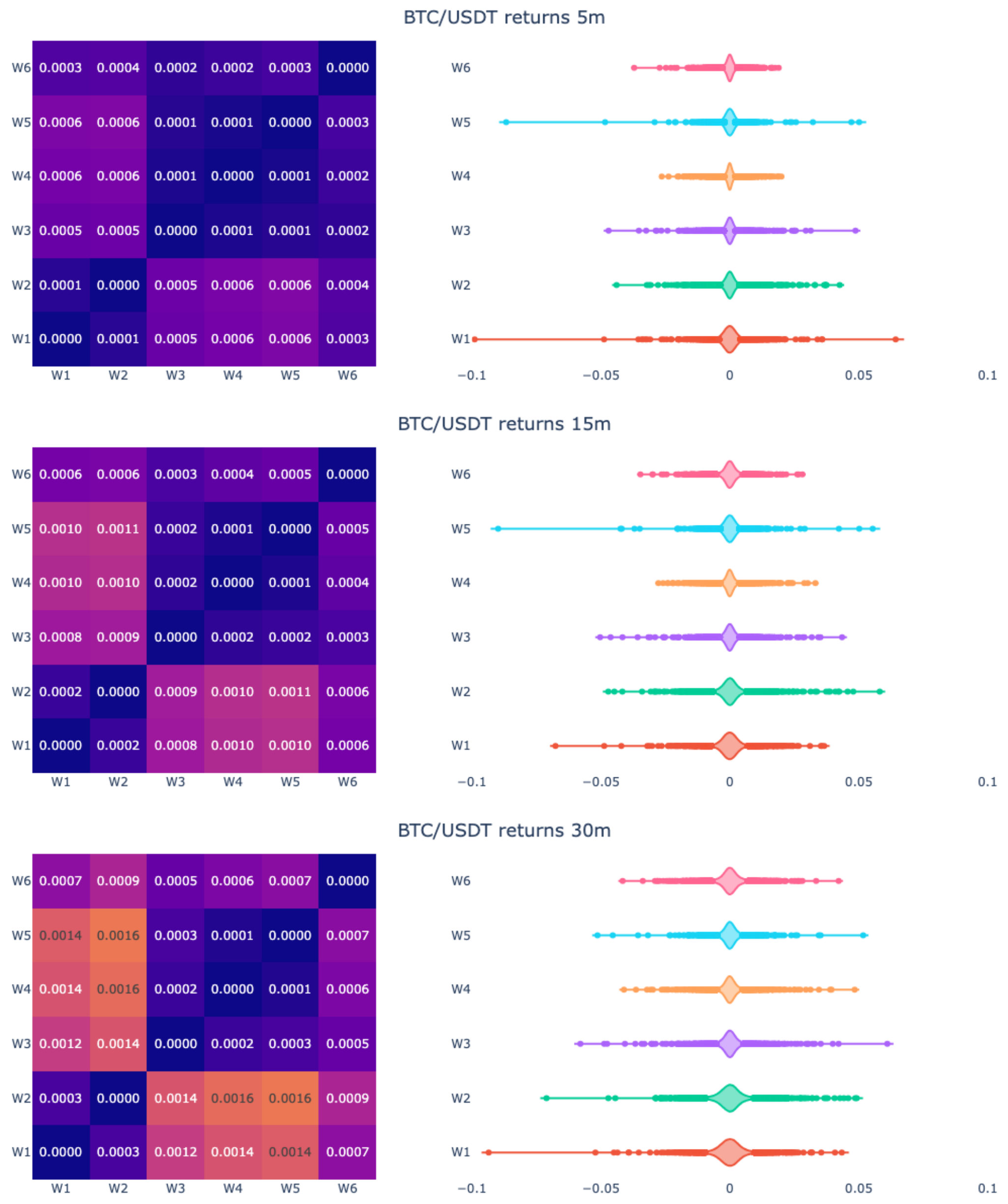
Note: Descriptive statistics of returns in data windows of 15min interval data. The statistics are not annualized. Null hypothesis of Kolmogorov-Smirnov (KS) test is that the distribution is normal.

Table 5: Descriptive statistics for 30 min out-of-sample data

Statistic	**W1**	**W2**	**W3**	**W4**	**W5**	**W6**
count	8640	8640	8640	8640	8640	8640
mean	0.0000043	-0.0000540	0.0000101	0.0000296	0.0000473	0.0000595
std	0.0049962	0.0054682	0.0038721	0.0034004	0.0031349	0.0039924
min	-0.0934631	-0.0710468	-0.0579109	-0.0410247	-0.0512786	-0.0415287
25% percentile	-0.0023584	-0.0023627	-0.0012400	-0.0012067	-0.0011561	-0.0016455
50% percentile	0.0000360	0.0000073	0.0000117	-0.0000114	0.0000311	0.0000521
75% percentile	0.0023458	0.0022461	0.0013298	0.0012539	0.0012576	0.0017785
max	0.0434281	0.0492234	0.0610820	0.0484100	0.0517262	0.0422052
kurtosis	22.73	12.11	34.48	25.83	34.78	10.52
skewness	-0.82	0.08	-0.33	0.41	-0.18	-0.10
KS test stat.	0.49	0.49	0.49	0.49	0.49	0.49
KS test p-value	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00	0.00e+00

Note: Descriptive statistics of returns in data windows of 30min interval data. The statistics are not annualized. Null hypothesis of Kolmogorov-Smirnov (KS) test is that the distribution is normal.

Figure 6: Distributions of returns for out-of-sample data



Note: Visualizations of the returns distributions for the out-of-sample data for different windows from testing period, for 5min, 15min and 30min intervals. On the left, heatmaps visualize distances between each pair of distributions according to Wasserstein distance. On the right distributions are visualized as violin plots.

4 Methodology

In this chapter, various trading strategies and the methodology for their evaluation are described. Fair comparison of strategies requires a formal definition of each strategy, clearly established comparison criteria, and metrics that compare different aspects of the strategies. Those are detailed in the initial section of this chapter (4.1). The latter section (4.2) describes the strategies that are compared in the study. For each strategy, a mechanism how the positions are selected is described as well as what are the hyperparameters that influence the strategy behavior.

4.1 Evaluation Framework

Strategies are evaluated over certain periods of time. Consider a time series consisting of T data points. The data point at each time step $t \in \{1..T\}$ represents a single time interval. During each time interval, one of the three possible positions can be taken:

- `long position` - The asset is bought. The value of the portfolio increases if the price of an asset increases.
- `short position` - The asset is short sold and will need to be bought back when the position is closed. The value of the portfolio increases if the price of an asset decreases.
- `no position` - No asset in the portfolio. The value of the portfolio does not change, regardless of the price of the asset.

Additionally, the following assumptions are made:

- When the strategy position changes, assets are always bought/sold at the close price of the previous interval. That is, if at the time $t-1$ the position was `no position` and at the time t the position is `long position`, the assets are bought at the `close price` of the interval at time $t-1$.
- Each such position change incurs an exchange fee e. In the experiments the value of e is assumed to be 0.1%.[8].
- Any fraction of the asset can be traded[9].
- No splitting of portfolio allocation, the asset is always bought/sold with the full portfolio value.
- At the end of the evaluation period all positions must be closed, that is, at time T the position must be `no position`.

Formally, a trading strategy is a function $s_\theta : \mathbb{R}^{\lambda \times d} \to \{-1, 0, 1\}$, where θ are strategy hyperparameters, λ is the size of the lookback window and d is the dimensionality of the vector $x_t \in \mathbb{R}^d$ that represents new, known and relevant information at each time step t, which can be used by the strategy. The codomain values $\{-1, 0, 1\}$ respectively, represent: `short position`, `no position`, and `long position`. Then, the recommended position p_t according to the strategy s at time t is calculated in the following way:

$$s_\theta([x_{t-\lambda}; ...; x_{t-1}]) = p_t \tag{2}$$

where $[\cdot;\cdot]$ is a concatenation operation. Note that the strategy has access only to information prior to time t. The positions are computed for each time index t over a given period of time. The portfolio value E_t at any time t can be calculated using the following formula:

$$E_0 = 1 \tag{3}$$

$$E_t = E_{t-1}(1 + r_t \cdot p_t)(1 - (|p_t - p_{t-1}|e)) \tag{4}$$

where r_t is a return in an interval t computed as in 1, p_t is a position recommended by the strategy and e is the exchange fee rate. The portfolio value at the end of the evaluation period T is equal to E_T.

Furthermore, following the methodology of Michańków et al. 2022, the following metrics are computed:

Annualized Return Compounded (ARC) Metric that shows the annualized rate of return for the evaluation period, computed as

$$ARC = (\frac{E_T}{E_0})^{\frac{Y}{T}} \tag{5}$$

where E_0 is the portfolio value at the beginning of the evaluation period, E_T is the portfolio value at the end of the evaluation period, T is number of the intervals in the evaluation period and Y is the number of the intervals in a year[10].

Annualized Standard Deviation (ASD) A metric that represents the yearly standard deviation of returns, assessing the strategy's volatility. The metric is determined as

$$ASD = \sqrt{\frac{Y}{T}\sum_{t=0}^{T}(r_t - \bar{r})^2} \tag{6}$$

where r_t is the return at time t, $\bar{r}$ is the mean return in the evaluation period, T is the number of intervals in the evaluation period and Y is the number of intervals in a year the same as in 5.

[8]The value is equal to standard spot fee rate on Binance https://www.binance.com/en/fee/trading

[9]In practice the smallest fraction of the asset that can be bought on Binance is 0.00001

[10]It is assumed that there are 365 days in a year, so for the datasets used in the research, this value is equal to $Y_{5m} = 105120$, $Y_{15m} = 35040$ and $Y_{30m} = 17520$.

Information Ratio (IR*) The Sharpie ratio (Sharpe 1998) that is calculated under the assumption of a risk-free rate of zero. The metric becomes the annualized return divided by its annualized standard deviation:

$$IR^* = \frac{ARC}{ASD} \tag{7}$$

Maximum Drawdown (MD) The portfolio's maximum drawdown represents the greatest decline measurable between the portfolio's highest value and its consecutive lowest point. This metric acts as an indicator of possible percentage loss within a specified period

$$MD = \max_{(t,\tau)\in\{[0..T]^2 : t<\tau\}} \left(\frac{E_t - E_\tau}{E_t}\right) \tag{8}$$

where T represents the total number of intervals during the evaluation timeframe, with E_t and E_τ denoting the portfolio values at times t and τ, respectively.

Modified Information Ratio (IR)** is the Information Ratio (7) adjusted by the Maximum Drawdawn (8), a metric initially introduced by Kosc et al. 2019:

$$IR^{**} = IR^* \times \frac{|ARC|}{MD} \tag{9}$$

Number of trades (N) The number of strategy position changes during the evaluation period. Calculated as

$$N = \sum_{t=1}^{T}(|p_t - p_{t-1}|) \tag{10}$$

where p_t is the position at the time t. Note that while it is possible to change the position from `long position` to `short position` in a single time frame, it is counted as two trades: closing the long position and opening the short position.

Percentage of Long Position (LONG) / Short Position (SHORT) Metric that measures the percentage of how many time intervals during the entire evaluation period the strategy took the long / short position.

$$LONG = \frac{1}{T}\sum_{t=0}^{T}\left(\frac{(p_t+1)p_t^2}{2}\right) \times 100\% \tag{11}$$

$$SHORT = \frac{1}{T}\sum_{t=0}^{T}\left(\frac{(1-p_t)p_t^2}{2}\right) \times 100\% \tag{12}$$

4.2 Strategies

This study involves the evaluation and comparison of five strategies. Each strategy considered in the study consists of the complete mechanism that converts the signal, whether originating directly from the data, a technical indicator or a machine learning model, into the actual position. This is something that is often neglected, especially when evaluating the effectiveness of machine learning models. Although the precision of the model's predictions is crucial, a fair assessment of the strategy is feasible only after these predictions are converted into portfolio positions.

4.2.1 Buy and Hold strategy

The first strategy, the only viable one according to the Efficient Markets Hypothesis, is the "Buy and Hold" strategy. It is regarded as a benchmark that other strategies aim to surpass. Framed as a formal function as described in (2), strategy has no hyperparameters and at every time interval takes a `long position`:

$$s^{BuyAndHold}(\cdot) = 1 \tag{13}$$

4.2.2 Moving Average Convergence Divergence strategy

The strategy is based on the technical indicator of the Moving Average Convergence Divergence (MACD). The indicator was proposed by Gerald Appel almost 50 years ago and is still widely used by technical analysts today (Appel 2005)[11]. The standard MACD oscillator is a collection of three time series calculated from daily historical price data, most often the closing price. Time series are calculated using the Exponential Moving Average (EMA)[12] with different window sizes. These are subsequently converted into positions as follows: when the MACD line moves above the signal line, it signifies an uptrend and acts as a buy signal. Conversely, when the MACD line falls below the signal line, it signals a downtrend, and one should exit the position:

$$MACD_t(\cdot) = EMA_t^{fast}(\cdot) - EMA_t^{slow}(\cdot) \tag{14}$$

$$SIGNAL_t(\cdot) = EMA_t^{signal}(MACD_t(\cdot)) \tag{15}$$

where the *fast*, *slow* and *signal* are window sizes for Exponential Moving Average. Typically, these values are 12, 26, and 9 respectively, but in the context of this study, they are considered as strategy hyperparameters:

$$s^{MACD}_{(fast,slow,signal,short)}(\cdot) = \begin{cases} 1 \quad if\ MACD_{t-1} \geq SIGNAL_{t-1} \\ 0 \quad if\ MACD_{t-1} < SIGNAL_{t-1} \text{ and } short = 0 \\ -1 \quad if\ MACD_{t-1} < SIGNAL_{t-1} \text{ and } short = 1 \end{cases} \tag{16}$$

where $short \in \{0, 1\}$ is an additional strategy hyperparameter that controls whether a strategy uses `exit position` or `short position` on sell signal.

4.2.3 Relative Strength Index strategy

The second strategy based on the technical indicator employs the Relative Strength Index (RSI) indicator. The RSI was developed by J.Welles Wilder Jr. and serves as a momentum oscillator designed to assess whether changes in asset prices suggest an overbought or oversold condition (Wilder 1978). The RSI is calculated as:

$$RSI_t(\cdot) = \left(\frac{100}{1 + RS_t(\cdot)}\right) \tag{17}$$

$$RS_t = \frac{SMMA^{window}(U(\cdot))}{SMMA^{window}(D(\cdot))} \tag{18}$$

$$U_t = \max(0, \texttt{close price}_t - \texttt{close price}_{t-1}) \tag{19}$$

$$D_t = \max(0, \texttt{close price}_{t-1} - \texttt{close price}_t) \tag{20}$$

where RS is a relative strength factor computed using Smoothed Moving Average (SMMA)[13] calculated over the "Up period" (U) and "Down period" (D) time series, using *window* time intervals prior to t-th interval. Although usually the window is assumed to be 14, the study treats *window* as a hyperparameter of the strategy.

To translate oscillator values into the positions, usually two threshold overbought/oversold for buy/sell signal are used. If the RSI crosses the over-sold threshold, it indicates a bullish sign, and when it crosses below the over-bought threshold, it is a bearish sign. Instead, in the study, four thresholds are used to indicate when: enter long position (`enter long`), exit long position (`exit long`), enter short position (`enter short`) and exit short position (`exit short`).

$$s^{RSI}_{\theta_{RSI}}(\cdot) = \begin{cases} 1 \quad if\ RSI_{t-1}(\cdot) > enter\ long \\ 0 \quad if\ RSI_{t-1}(\cdot) < exit\ long \text{ and } p_{t-1} = 1 \\ -1 \quad if\ RSI_{t-1}(\cdot) < enter\ short \\ 0 \quad if\ RSI_{t-1}(\cdot) > exit\ short \text{ and } p_{t-1} = -1 \\ p_{t-1} \quad else \end{cases} \tag{21}$$

where $\theta_{RSI} = (window, enter\ long, exit\ long, enter\ short, exit\ short)$ are strategy hyperparamethers and p_{t-1} is a position taken on the previous time interval.

[11] Although the indicator was initially shared in the 1970s within the Systems & Forecasts newsletter, the book detailing this indicator wasn't released until 2005.

[12] https://en.wikipedia.org/wiki/Exponential_smoothing

[13] https://en.wikipedia.org/wiki/Moving_average#Modified_moving_average

4.2.4 Informer based strategies

The last two strategies use the predictions of the state-of-the-art machine learning model for sequence modeling: Informer (Zhou et al. 2021). The informer uses the modified Transformer architecture (Vaswani et al. 2017) to better handle long sequences. Several studies have shown that it surpasses other deep learning architectures in predicting financial time series (Ding et al. 2023, Lu et al. 2023).

Informer Architecture follows the encoder-decoder structure (Chollet 2017, Bahdanau et al. 2014) where the encoder maps an input sequence of vectors $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_n)$ to a sequence of continuous representations $\mathbf{Z} = (\mathbf{z}_1, ..., \mathbf{z}_n)$. Given $\mathbf{Z}$, the decoder then generates an output sequence $\mathbf{Y} = (\mathbf{y}_1, ..., \mathbf{y}_m)$ one element at a time. Both the encoder and decoder are built with a number of identical layers. Let $\mathbf{X}_0 = \mathbf{X}$ be an input to the first encoder layer, each i-th encoder layer transforms the sequence of $\mathbf{X}_{i-1}$ input sequence to $\mathbf{X}_i$ sequence, where $i \in \{1, .., L^{enc}\}$ and L^{enc} is the number of encoder layers. The encoder output is equal to $\mathbf{X}_{L^{enc}} = \mathbf{Z}$. Similarly, each decoder layer transforms the sequence of previous predicted targets $\mathbf{Y}_0$ and the encoder output $\mathbf{Z}$ into the sequence $\mathbf{Y}_j$, which is then consumed by the next decoder layer. The final output of the decoder layer is the output of the decoder $\mathbf{Y} = \mathbf{Y}_{L^{dec}}$, where $j \in \{1..L^{dec}\}$ and L^{dec} is the number of decoder layers. *Number of the encoder layers* and *the number of decoder layers* are model hyperparameters. The main building block of both those layers is *multi-head probe-sparse self-attention.*

Multi-head probe-sparse self-attention: The canonical self attention of Transformer is defined based on the tuple inputs, i.e, query (Q), key (K) and value (V), which performs the scaled dot-product as:

$$\mathcal{A}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathit{softmax}\left(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \tag{22}$$

where $\mathbf{Q} \in \mathbb{R}^{L_Q \times d}$, $\mathbf{K} \in \mathbb{R}^{L_K \times d}$, $\mathbf{V} \in \mathbb{R}^{V_Q \times d}$ and L_Q, L_K, L_V are lengths of the query, key and value sequences respectively and the d is the dimensionality of the model. Softmax[14] is a function that converts a vector of real numbers, so it can be interpreted as a probability distribution. In this formulation, computing attention requires the quadratic time dot-product computation. The authors of Informer proposed a measurement that transforms $\mathbf{Q}$ into a sparse $\bar{\mathbf{Q}}$ matrix, that contains only the top $u = c \cdot ln(L_Q)$ queries under the sparsity measurement, where c is a constant sampling factor. The change makes the computation much more efficient and the architecture more applicable to the longer sequences.

$$\mathcal{A}'(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathit{softmax}\left(\frac{\bar{\mathbf{Q}}\mathbf{K}^T}{\sqrt{d}}\right)\mathbf{V} \tag{23}$$

In each layer, attention is performed h times with different linear projections of inputs. This mechanism is called multi-head attention and h is the model hyperparameter that controls the number of so called attention heads.

$$\mathcal{H}_i(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \mathcal{A}'(\mathbf{Q}\mathbf{W}_i^Q, \mathbf{K}\mathbf{W}_i^K, \mathbf{V}\mathbf{W}_i^V) \tag{24}$$

$$\mathcal{MH}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = [\mathcal{H}_1(\cdot); ...; \mathcal{H}_h(\cdot)] \tag{25}$$

where the projections $\mathbf{W}_i^Q \in \mathbb{R}^{d \times L_Q}$, $\mathbf{W}_i^K \in \mathbb{R}^{d \times L_K}$, $\mathbf{W}_i^V \in \mathbb{R}^{d \times V_Q}$ are learned parameters.

The self-attention means that the layer "attends" to itself, that is, all queries, keys, and values consider the same input sequence i.e. $\mathbf{Q} = \mathbf{K} = \mathbf{V} = \mathbf{X}$, where $\mathbf{X}$ is an input to the *multi-head probe-sparse self-attention layer* ($\mathcal{MHSA}$).

$$\mathcal{MHSA}(\mathbf{X}) = \mathcal{MH}(\mathbf{X}, \mathbf{X}, \mathbf{X}) \tag{26}$$

Encoder: The encoder layers of Informer consist of multi-head probe-sparse self-attention layer with distilling operation that trims the input's time dimension. Given $\mathbf{X}_{i-1}$ input to the i-th Informer layer, the output of the i-th layer is calculated as:

$$\mathbf{X}_i = \mathit{MaxPool}(\mathit{ELU}(\mathit{Conv1d}(\mathcal{MHSA}(\mathbf{X}_{i-1})))) \tag{27}$$

where *Conv1d* performs a one dimensional convolutional filter over the time dimension (Kiranyaz et al. 2019), with *ELU* activation function (Clevert et al. 2016) and max-pooling layer (Nagi et al. 2011) that down-samples the layer output.

[14]https://en.wikipedia.org/wiki/Softmax_function

Decoder: Informer uses standard Transformer decoder, which consists of two $\mathcal{MHSA}$ layers in which the first takes an input of past target outputs and the second output of the encoder. Those are followed with a position wise fully connected layer ($\mathcal{FFN}$). It consists of two linear transformations with a ReLU (Agarap 2019) activation in between, and is applied to vectors at each position separately:

$$\mathcal{FFN}(\mathbf{x}) = max(0, \mathbf{x}\mathbf{W}_1 + \mathbf{b}_1)\mathbf{W}_2 + \mathbf{b}_2 \tag{28}$$

where $\mathbf{W}_1 \in \mathbb{R}^{d \times f}, \mathbf{W}_2 \in \mathbb{R}^{f \times d}, \mathbf{b}_1 \in \mathbb{R}^{f}, \mathbf{b}_2 \in \mathbb{R}^{d}$ are trained parameters and d is the model dimensionality and f is the size of the fully connected layer, both of these values are model hyperparameters. Around of each of those sub-layers there is a residual connection (Kolen and Kremer 2001), dropout regularization (*Dropout*) and a layer normalization (*LayerNorm*) (Ba et al. 2016).

$$\mathbf{Y}'_j = LayerNorm(Dropout(\mathcal{MHSA}(\mathbf{Y}_{j-1})) + \mathbf{Y}_{j-1}) \tag{29}$$

$$\mathbf{Y}''_j = LayerNorm(Dropout(\mathcal{MHSA}(\mathbf{Z}) + \mathbf{Z}) \tag{30}$$

$$\mathbf{Y}'''_j = \mathbf{Y}'_j + \mathbf{Y}''_j = [\mathbf{y}'''_{1j}; ...; \mathbf{y}'''_{mj}] \tag{31}$$

$$\mathbf{Y}_j = LayerNorm(Dropout([\mathcal{FFN}(\mathbf{y}'''_{ij}); ...; \mathcal{FFN}(\mathbf{y}'''mj)]) + \mathbf{Y}''') \tag{32}$$

Model input: The model input $\mathbf{X}_0$ consists of sequence of n d-dimensional input vectors. These vectors are derived by adding together three input components:

- **Real input variables:** All real-value input variables are concatenated and linearly transformed into d-dimensional vector.
- **Categorical variables:** All categorical variables are mapped into the embeddings - real-value vectors that are model parameters. Then they are concatenated and linearly transformed into d-dimensional vector.
- **Positional embeddings:** Since Informer model contains no recurrence and convolution, the ordering of input vectors needs to be modeled explicitly. To achieve this, a method known as positional encodings is employed. For each position a vector of sine and cosine functions of different frequencies is computed, that is, each dimension of the positional encoding corresponds to a sinusoid.

Loss function: As the first loss function used to train Informer model the study uses Root Mean Squared Error (RMSE) loss function which is commonly used for financial data forecasting[15]. RMSE loss function is defined as:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2} \tag{33}$$

where y is observed value, $\hat{y}$ is model prediction, N is number of observations.

However, optimizing model with such these loss function does not align with the objective of finding a strategy aimed at maximizing returns. Consider an asset, which price does not move for 99 out of 100 intervals and then suddenly drops. A model that predicts the same price every time would have a 99% accuracy, however a strategy based on such model predictions would have very poor results. This example illustrates that while the model using RMSE loss functions may be trained to have high accuracy of predictions, those may not necessarily be *meaningful* predictions in terms of building the trading strategy. Moreover, with point-wise predictions, such as the ones coming from RMSE loss, there is no information of the certainty of the prediction.

To address the aforementioned issues the study introduces two other loss functions that may be better suited for training models that are later used in trading strategy. The first loss is a Quantile Loss, that is, instead of predicting the point, the model predicts a distribution, more precisely quantiles of the returns distribution. With such signal, a strategy can be constructed that considers confidence in the prediction of the model. The Quantile Loss is defined as:

$$QuantileLoss = \frac{1}{N}\sum_{i=1}^{N}\sum_{q} max(q \cdot (y - \hat{y}), (1 - q) \cdot (\hat{y} - y)) \tag{34}$$

[15]Another common loss function is Mean Squared Error (MSE), but the difference between the two is just that one is scaled version of the other.

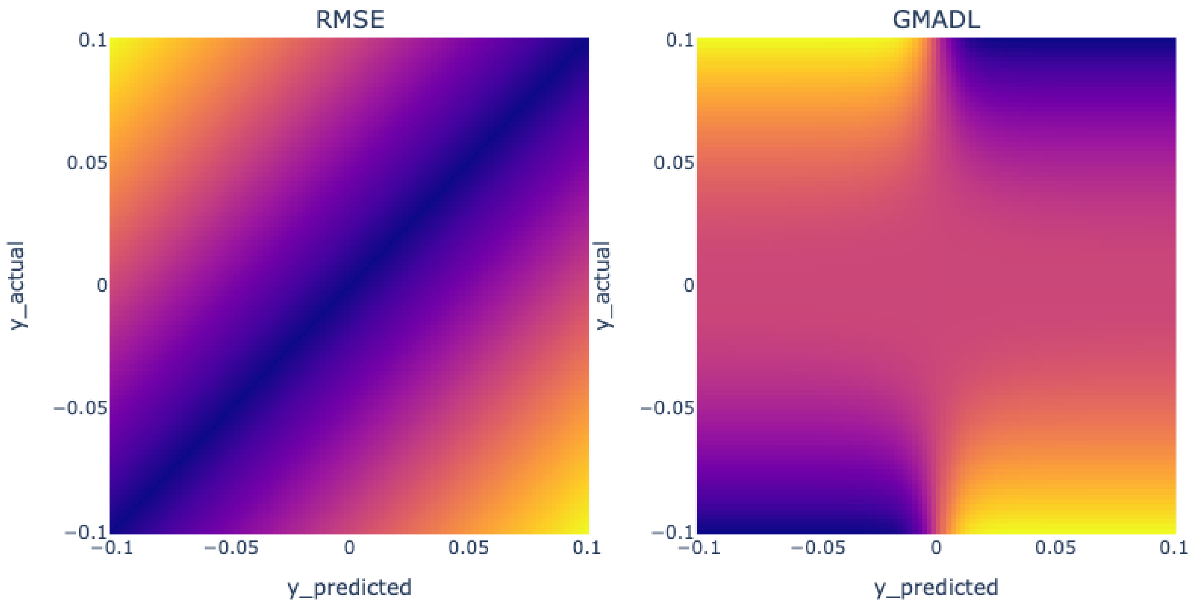
where y is observed value, $\hat{y}$ is model prediction, N is number of observations. $q \in [0, 1]$ is a quantile, in the study the following quantiles were used $q \in \{0.01, 0.02, 0.03, 0.05, 0.1, 0.25, 0.5, 0.75, 0.9, 0.95, 0.97, 0.98, 0.99\}$[16].

The other loss function considered is a Generalized Mean Absolute Directional Loss (GMADL) (Michańków et al. 2024). It puts more emphasis on the direction of the returns, i.e. whether they were positive or negative rather than precision and rewards the model for correctly predicting larger return values. The loss function is defined as

$$GMADL = \frac{1}{N}\sum_{i=1}^{N}(-1)\cdot(\frac{1}{1+e^{-a\cdot y\cdot\hat{y}}} - \frac{1}{2})\cdot(|y|)^{b} \tag{35}$$

where y is observed value, $\hat{y}$ is model prediction, N is number of observations. a and b are loss function parameters that control the stepness of the function slope. In the study they are considered to be equal to $a = 100$ and $b = 2$.

Figure 7: RMSE vs GMADL loss functions.



Note: Comparison of the two loss functions: Mean Absolute Error (MAE) on the left and Generalized Mean Absolute Directional Loss (GMADL) on the right.

Strategies Leveraging the predictions generated by the Informer model, one can formulate a trading strategy. Given that the models were trained using three distinct approaches, the RMSE loss, Quantile Loss and GMADL, each method provides the opportunity to develop a unique strategy that optimally utilizes the model's predictions.

RMSE Informer strategy: Strategy that uses predictions of the Informer trained with RMSE is defined as the following: four threshold values are defined when to *enter long*, *exit long*, *enter short* and *exit short*. Those thresholds are compared directly with the return predicted by the model.

$$s^{RMSE}_{\theta_{RMSE}}(\cdot) = \begin{cases} 1 \quad if\ \hat{y}_t > enter\ long \\ 0 \quad if\ \hat{y}_t < exit\ long \text{ and } p_{t-1} = 1 \\ -1 \quad if\ \hat{y}_t < enter\ short \\ 0 \quad if\ \hat{y}_t > exit\ short \text{ and } p_{t-1} = -1 \\ p_{t-1} \quad else \end{cases} \tag{36}$$

[16] A decision to train the model with many different quantiles was motivated by the fact, that then a strategy that bases on this model predictions can be fine-tuned with different quantile values without need for retraining the model.

where $\hat{y}_t$ is the model prediction for time step t and p_{t-1} is position on the previous timestep. The strategy hyperparameters are $\theta_{RMSE} = (\Theta_{RMSE}$, *enter long*, *exit long*, *enter short*, *exit short*) and Θ_{RMSE} are parameters of the Informer model trained with RMSE loss.

Quantile Informer strategy: A strategy based on the predictions of the model trained with Quantile Loss is constructed in the following manner: probabilities are defined to *enter long*, *exit long*, *enter short* and *exit short* positions, as well as the *threshold* value. If the return predicted by the model is above the *threshold* with certain probability, i.e. $1 -$ *enter long*[17] quantile is above the threshold, a strategy enters `long position`. Conversly, if the *enter short* quantile is below $-$*threshold* a `short position` is opened. The interpretation is that the model prediction has confidence of *enter long*/*enter short* that the returns are above/below the *threshold*.

$$s^{Quantile}_{\theta_{Quantile}}(\cdot) = \begin{cases} 1 \quad if\ \hat{y}_{t,1-enter\ long} > threshold \\ 0 \quad if\ \hat{y}_{t,exit\ long} < -threshold \text{ and } p_{t-1} = 1 \\ -1 \quad if\ \hat{y}_{t,enter\ short} < -threshold \\ 0 \quad if\ \hat{y}_{t,1-exit\ short} > threshold \text{ and } p_{t-1} = -1 \\ p_{t-1} \quad else \end{cases} \tag{37}$$

where $\hat{y}_{t,q}$ is a model prediction at time t for the q quantile and p_{t-1} is a position on the previous timestep. The strategy hyperparameters are $\theta_{Quantile} = (\Theta_{Quantile}$, *enter long*, *exit long*, *enter short*, *exit short*, *threshold*). and $\Theta_{Quantile}$ are parameters of the Informer model trained with Quantile loss.

GMADL Informer strategy: Strategy that uses predictions of the Informer trained with GMADL is defined analogously to the one using RMSE: four threshold values are defined when to *enter long*, *exit long*, *enter short* and *exit short* and the thresholds are compared directly with the return predicted by the model. While in this case one does not know explicitly the confidence of those predictions, because of how GMADL is constructed, for the larger returns they should be more reliable than the predictions of the model trianed with RMSE loss.

$$s^{GMADL}_{\theta_{GMADL}}(\cdot) = \begin{cases} 1 \quad if\ \hat{y}_t > enter\ long \\ 0 \quad if\ \hat{y}_t < exit\ long \text{ and } p_{t-1} = 1 \\ -1 \quad if\ \hat{y}_t < enter\ short \\ 0 \quad if\ \hat{y}_t > exit\ short \text{ and } p_{t-1} = -1 \\ p_{t-1} \quad else \end{cases} \tag{38}$$

where $\hat{y}_t$ is the model prediction for time step t and p_{t-1} is position on the previous timestep. The strategy hyperparameters are $\theta_{GMADL} = (\Theta_{GMADL}$, *enter long*, *exit long*, *enter short*, *exit short*) and Θ_{GMADL} are parameters of the Informer model trained with GMADL loss.

5 Experiments

The evaluation framework and strategies described in Chapter 4 were implemented in Python[18] and are publicly available in the Gitlab repository[19]. Technical indicators were computed using TA-lib library[20], Informer model was trained using PytorchForecasting[21] and the Informer implementation was taken from the following Github repository[22]. Section 5.1 describes the hyperparameter selection process for each strategy and presents the values that were selected. The evaluation of the test data is then presented and analyzed in Section 5.2. The last Section 6 presents the sensitivity analysis of how the selection of strategy hyperparameters influences the strategy performance.

[17]Model prediction $\hat{y}_{t,q}$ for a quantile q means that the value is below $\hat{y}_{t,q}$ with q probability, and above $\hat{y}_{t,q}$ with $1-q$ probability.
[18]https://www.python.org/
[19]https://gitlab.com/FilipStefaniuk/wne-msc-thesis
[20]https://ta-lib.org/
[21]https://pytorch-forecasting.readthedocs.io/en/stable/
[22]https://github.com/martinwhl/Informer-PyTorch-Lightning

5.1 Training and hyperparameter selection

Each strategy was independently instanciated for each data window. Strategies that were based on machine learning model were trained using the *training* part of the data window and the hyperparameters were selected using *validation* part. Strategies that do not require training i.e. MACD based strategy and RSI based strategy had their hyperparameters selected using only the *validation* part of the dataset. An alternative would be to use all in-sample (*train* and *validation* parts) to select hyperparameters for non-ml strategies, however a decision to use only *validation* part was motivated by the fact that to make comparison fair, the same data should be used for similar strategy hyperparameter selection (thresholds) for ml and non-ml based strategies. The best hyperparameter combinations were selected by comparing the IR** (4.1) metric. The reminder of this section details the hyperparameter search space for each strategy and lists which hyperparameter values were selected for the strategies for each data window.

5.1.1 MACD Strategy hyperparameters

As described in Section 4.2.2, the MACD strategy has four hyperparameters, three of which (*fast*, *slow*, *signal*) control the size of the window for moving averages and the *short* whether a sell signal should be interpreted to close `long position` or to additionally open `short position`. Table 6 presents the values tested for the hyperparameters. A *short* hyperparameter takes binary values. Window sizes take values of the 16 values of the Fibonacci sequence, the sequence has been chosen as it grows exponentially, thus allows to search efficiently the space of possible windows - using the equally distributed values would be less efficient as large window sizes are less sensitive to small window size changes. Note that even though combinations were searched exhaustively, an additional constraint that prevents the *fast* window to be longer than *slow* window has been applied.

Table 6: Hyperparameter search space for MACD strategy.

Parameter	**Values**
fast	[2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584]
slow	[2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584]
signal	[2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584]
short	[0, 1]

Note: Considered values of hyperparameters for MACD strategy, all 8192 combinations were searched exhaustively.

Hyperparameters selected for each window for every data interval are presented in Table 7. A bias towards larger window sizes can be observed, suggesting that MACD indicator might not be appropriate for high frequencies such as 5, 15 or 30 minutes. Additionally for all three frequencies, the strategies for the last two windows do not use short positions. Setting the short position on the down trend should always be beneficial, so this can suggest that the data in the last two windows may be much less predictable.

5.1.2 RSI strategy hyperparameters

As detailed in Section 4.2.3 the RSI strategy has five hyperparameters. The *window* parameter sets the window size for computing the moving averages. The other four set thresholds when `long position` and `short position` should be opened or closed. For the window size, similarly as with MACD strategy values of the Fibonacci sequence were used. Combinations of window sizes and various thresholds listed in Table 8 were searched exhaustively, "-" symbolize that the position will never be opened/closed regardless of the RSI oscillator value. Note that *exit long* and *exit short* hyperparameter values make sense to be different from "-" only if corresponding *enter long* and *enter short* hyperparameters are different than "-". However, the reverse situation can occur as the long/short position is automatically closed as the next short/long position is entered, that is e.g. a combination where *enter long* is not equal to "-" and *exit long* is equal to "-" can occur is *enter short* is present as the `long position` will be closed if *enter short* threshold is reached.

The table lists selected hyperparameters. Contrary to the MACD strategy, the RSI strategy seems to benefit from higher frequencies, and the selected values for the *window* hyperparameter were rather small. Only in few cases *exit long* and *exit short* hyperparameters were set, meaning that in most cases when the long/short position was closed, the short/long position was immediately opened.

Table 7: Selected hyperparameters for MACD strategy.

Window	*fast*	*slow*	*signal*	*short*
W1-5min	8	2584	987	1
W2-5min	377	610	610	1
W3-5min	987	2584	987	1
W4-5min	610	2584	987	1
W5-5min	987	1597	987	0
W6-5min	1597	2584	377	0
W1-15min	8	377	1597	1
W2-15min	144	987	2584	1
W3-15min	377	2584	2584	1
W4-15min	377	2584	610	1
W5-15min	233	610	233	0
W6-15min	144	377	2584	0
W1-30min	1597	2584	2584	1
W2-30min	233	2584	2584	1
W3-30min	13	2584	2584	1
W4-30min	233	987	233	1
W5-30min	144	233	233	0
W6-30min	1597	2584	1597	0

Note: Table presents hyperparameter values selected for the MACD strategy for each window of 5min, 15min and 30min interval data.

Table 8: Hyperparameter space for RSI strategy.

Parameter	**Values**
window	[2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584]
enter long	[-, 70, 75, 80, 85, 90, 95]
exit long	[-, 5, 10, 15, 20, 25, 30]
enter short	[-, 5, 10, 15, 20, 25, 30]
exit short	[-, 70, 75, 80, 85, 90, 95]

Note: Considered values of hyperparameters for RSI strategy, all 38416 combinations were searched exhaustively.

Table 9: Selected hyperparameters for the RSI strategy.

Window	*window*	*enter long*	*exit long*	*enter short*	*exit short*
W1-5min	21	80	-	25	-
W2-5min	34	75	-	25	-
W3-5min	5	95	-	15	-
W4-5min	8	90	5	-	-
W5-5min	13	90	20	-	-
W6-5min	21	85	15	-	-
W1-15min	5	80	-	15	-
W2-15min	34	75	-	30	-
W3-15min	5	95	-	10	-
W4-15min	34	75	-	30	-
W5-15min	13	70	-	5	-
W6-15min	21	85	15	-	-
W1-30min	5	95	-	5	-
W2-30min	21	70	-	30	-
W3-30min	5	95	-	15	85
W4-30min	8	95	5	-	-
W5-30min	13	70	-	5	-
W6-30min	34	80	25	-	-

Note: Table presents hyperparameter values selected for the RSI strategy for each window of 5min, 15min and 30min interval data.

5.1.3 Informer Strategies training and hyperparameters

The process of selecting the hyperparameters for Informer-based strategies was divided into two steps. First, a model hyperparameters were selected by training the Informer with different combinations of model hyperparameters, then the strategy hyperparameters were selected for the best model. In other words, first, the model that can produce the best predictions was selected and then, based on the predictions of the models, the best strategy was constructed. The reason for this split was motivated mostly by the heaviness of the training process and the fact that if both model and strategy parameters are changed at the same time, it is difficult to assess which has the biggest impact on the final strategy result. This divide carries the risk that the best model, in terms of the loss function, does not necessarily mean the best strategy, in terms of IR**. However, this is addressed, to some extent, by selecting a loss function that aligns more closely with the objective of maximizing returns (4.2.4).

Training and model hyperparameters A separate instance of Informer was trained for each data window and for each RMSE, Quantile and GMADL loss functions. The target variable in each case was `returns`. Each training run consisted of 40 epochs with an early stopping[23]. The input to the model consisted of a sequence whose length was controlled by the *past window* hyperparameter. The sequence consisted of two types of variables: `real` and `categorical`. The `real` variables were normalized before they were passed to the model. The `categorical` variables were mapped into real-value embeddings.

Table 10: Input variables to Informer model.

Variable type	**Variable**
real	open price
	high price
	low price
	close price
	volume
	returns
	vix
	fed rates
	fear and greed
	open to close price
	high to close price,
	low to close price
	vol 1h
	vol 1d
	vol 7d
	sma 1h to close price
	sma 1d to close price
	sma 7d to close price
	ema 1h to close price
	ema 1d to close price
	macd
	macd signal
	rsi
	low bband to close price
	up bband to close price
	mid bband to close price
categorical	hour
	weekday

Table 10 presents the variables that were passed to the model as an input. The `open price`, `high price`, `low price`, `close price` and `volume` come directly from the dataset described in Chapter 3. The `returns` were computed according to (1) and the `vix`, `fed rates`, `fear and greed` are additional data that the dataset was augmented with (Section 3.1). The other variables were derived from the variables in the data set: `open to close price`, `high to close price`, `low to close price`, `high to low price` are the ratios of the corresponding price values. `vol 1h`, `vol 1d`, `vol 7d` variables are volatility computed on the 1 hour, 1 day and 7 days past data, respectively. `sma 1h to close price`, `sma 1d to close price` and `sma 7d to close price` are ratios of Simple Moving Average computed on 1 hour, 1 day, and 7 day windows to `close price`. `ema 1h to close price` and `ema 1d to close`

[23]The number of observations in each dataset (5min, 15min 30min) varied and was rather large for a 5 min dataset, so instead of running validation at the end of each epoch, it was run after every 300 batches. The *patience* parameter that controls early stopping was set to 15, meaning that if in the next 15 consecutive validations the validation loss does not improve, the training is stopped.

`price` are ratios of Exponential Moving Average computed on data from past 1 hour and 1 day to `close price`. The `macd` and `macd` comes from the MACD technical indicator computed using the `close price` value and default values for window sizes[24]. The `rsi` is a RSI technical indicator, computed using the 14 past data points. Finally `low bband to close price`, `up bband to close price` and `mid bband to close price` are ratios of Bollinger Bands (Bollinger 2002) to `close price`.

The `categorical` variables have values of 0 to 23 for `hour` and 0 to 6 for `weekday` and represent an hour-of-the-day and a day-of-the-week of the interval (in respect to interval's `close time`)[25].

The hyperparameters of the model were determined by randomly sampling the hyperparameter space. The random search was carried out due to a large number of possible combinations and the training time, which took approximately 1 hour per training run[26]. A sample of 30 combinations (out of 1166400) was tested and the combination of hyperparameters with the lowest validation loss value was selected. Note that since two values of the two different loss functions cannot be directly compared, this procedure had to be carried out separately for each loss function and for every frequency, totaling in $\textit{num frequencies} \times \textit{num loss functions} \times \textit{sample size} = 3 \times 3 \times 30 = 270$ training runs. Furthermore, as this process is extremely heavy, the hyperparameter search was done only on the first data window, and the exact selected combination of training and model hyperparameters were later used to train the model on other data windows for the given time interval. The space of Informer model hyperparameters is presented in Table 11. where *past window* is a length of the input sequence, *batch size* and *learning rate* are the standard machine learning training parameters. Other hyperparameters are specific to Informer architecture and are described in Section 4.2.4.

Table 11: Informer model hyperparameter space.

Parameter	**Values**
past window	[20, 21, ..., 119, 120]
batch size	[64, 128, 256]
learning rate	[0.001, 0.0005, 0.0001]
model dimensionality (d)	[256, 512, 1024]
fully connected dimensionality (f)	[256, 512, 1024]
attention heads (h)	[1, 2, 4, 6]
dropout	[0.05, 0.1, 0.2, 0.3]
number of encoder layers	[1, 2, 3]
number of decoder layers	[1, 2, 3]

Note: Table presents all hyperparameter values that were considered. The dimension of the grid is 1166400, so the search was done by randomly sampling 30 combinations.

The Table 12 lists all the hyperparameter combinations selected for the Informer model for different data frequency and loss functions. Interestingly, it seems that the better models were the smaller ones, having shorter input sequence (20-30) and smaller dimensionality. This might have been caused by still relatively small size of the training dataset (Fig 2). The different values of *batch size* and *learning rate* from the tested range did not seem to have significant impact on the model training.

Table 12: Selected hyperparameters for Informer model.

	Quantile RMSE			**Quantile Informer**			**GMADL Informer**		
Parameter	**5min**	**15min**	**30min**	**5min**	**15min**	**30min**	**5min**	**15min**	**30min**
past window	102	110	115	22	28	72	28	30	26
batch size	64	64	64	64	64	128	256	128	128
model dimensionality (d)	512	512	256	256	256	1024	256	256	256
fully connected layer dim	512	256	512	512	256	1024	256	256	256
attention heads h	4	4	2	2	6	2	2	2	4
dropout	0.3	0.1	0.2	0.05	0.1	0.05	0.01	0.01	0.01
number of encoder layers	2	3	2	1	1	1	1	1	1
number of decoder layers	3	2	2	1	3	3	3	1	3
learning rate	0.0005	0.0005	0.0005	0.0001	0.001	0.0001	0.0001	0.001	0.0001

Note: Table presents hyperparameter values selected for Informer model trained on 5min, 15min and 30min interval data with RMSE loss, Quantile Loss and GMADL loss functions.

[24]The window sizes used for computing MACD were: 12 for fast window, 26 for slow window and 9 for signal window.

[25]E.g. an interval with close time at 2023-10-18 01:29:59 will have values of 1 (second hour of the day) for the `hour` variable and 2 (third day of the week - Wednesday) for the `weekday` variable

[26]The training was done using a single Nvidia GTX 1080TI card.

RMSE Informer strategy hyperparameters For each data window, once the model was trained, its predictions were used to create a trading strategy. To select the optimal hyperparameters, *validation* part of the data window was used. The hyperparameters of this strategy are *enter long*, *exit long*, *enter short* and *exit short*. The tested values range from 0.0001 to 0.007. "-" indicates that the position will not be taken regardless of the model prediction. The possible values of the hyperparameters are listed in Table 13.

Table 13: Hyperparameter space for RMSE Informer Strategy

Parameter	**Values**
enter long	[-, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007]
exit long	[-, -0.001, -0.002, -0.003, -0.004, -0.005, -0.006, -0.007]
enter short	[-, -0.001, -0.002, -0.003, -0.004, -0.005, -0.006, -0.007]
exit short	[-, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007]

Note: Considered values of hyperparameters for RMSE Informer strategy, all 4096 combinations were searched exhaustively.

The selected hyperparameter values are presented in Table 14. It can be observed that the meaningful values could be selected only for the 30 minute interval. For the 15 and 5 minute interval the predictions from the model were close to 0 (smaller than the transaction fee), so to do any trades the strategy had to pick the smallest value possible for the threshold which was 0.001.

Table 14: Selected hyperparameter values for RMSE Informer Strategy

Window	***enter long***	***exit Long***	***enter Short***	***exit Short***
W1-5min	0.002	-	-0.001	0.001
W2-5min	-	-	-0.001	0.001
W3-5min	-	-	-0.001	0.001
W4-5min	-	-	-0.001	0.001
W5-5min	-	-	-0.001	0.001
W6-5min	-	-	-0.001	0.001
W1-15min	0.002	-	-0.001	-
W2-15min	-	-	-0.002	0.002
W3-15min	-	-	-0.001	0.002
W4-15min	0.001	-	-0.002	-
W5-15min	-	-	-0.001	0.001
W6-15min	-	-	-0.002	0.001
W1-30min	0.007	-	-0.001	-
W2-30min	0.003	-	-0.007	-
W3-30min	-	-	-0.001	0.005
W4-30min	0.002	-	-0.006	-
W5-30min	0.005	-0.001	-	-
W6-30min	0.003	-	-0.004	-

Note: Table presents hyperparameter values selected for the RMSE Informer strategy for each window of 5min, 15min and 30min interval data.

Quantile Informer strategy hyperparameters For model with Quantile loss, the parameters consisted of *enter long*, *exit long*, *enter short*, *exit short* and *threshold* hyperparameters described in Section 4.2.4. Here the possible hyperparameter values correspond to the quantiles of the loss function, with "-" meaning that the position won't be opened/closed regardless of the model prediction. Three values for the *threshold* were tested with the smallest one equal to the *exchange fee*. All the possible values of the hyperparameters are listed in Table 15.

Selected hyperparameters are presented in Table 16. It can be observed that mostly the parameters that require high model confidence to open/close the position were selected. This may be caused by the fact, that each position incurs a fee and each mistake is extremely costly. With transaction fee equal to 0.1% even if the asset price stays the same, already with $log_{0.999}(0.5) \approx 692$ incorrect position changes the portfolio value is halved - which is not that many considering the minute frequency data.

Table 15: Hyperparameter space for Quantile Informer Strategy

Parameter	**Values**
enter long	[-, 0.75, 0.9, 0.95, 0.97, 0.98, 0.99]
exit long	[-, 0.75, 0.9, 0.95, 0.97, 0.98, 0.99]
enter short	[-, 0.75, 0.9, 0.95, 0.97, 0.98, 0.99]
exit short	[-, 0.75, 0.9, 0.95, 0.97, 0.98, 0.99]
threshold	[0.001, 0.002, 0.003]

Note: Considered values of hyperparameters for Quantile Informer strategy, all 12288 combinations were searched exhaustively.

Table 16: Selected hyperparameter values for Quantile Informer strategy.

Window	***enter long***	***exit long***	***enter short***	***exit short***	***threshold***
W1-5min	-	-	0.980	0.970	0.003
W2-5min	0.970	-	0.950	-	0.003
W3-5min	0.990	-	0.950	-	0.002
W4-5min	0.970	0.990	-	-	0.003
W5-5min	0.900	0.900	-	-	0.002
W6-5min	0.950	0.950	-	-	0.003
W1-15min	0.980	-	0.990	-	0.003
W2-15min	0.990	0.950	-	-	0.003
W3-15min	-	-	0.990	0.990	0.003
W4-15min	0.990	0.970	-	-	0.003
W5-15min	0.950	0.990	-	-	0.001
W6-15min	0.990	0.950	-	-	0.002
W1-30min	0.980	-	0.950	0.970	0.003
W2-30min	-	-	0.980	0.990	0.001
W3-30min	-	-	0.990	0.900	0.003
W4-30min	0.950	0.990	-	-	0.001
W5-30min	0.900	0.990	-	-	0.001
W6-30min	0.970	0.980	-	-	0.003

Note: Table presents hyperparameter values selected for the Quantile Informer strategy for each window of 5min, 15min and 30min interval data.

GMADL Informer strategy hyperparameters The last strategy was created using the Informer trained with GMADL loss funtion. The hyperparameters of this strategy are *enter long*, *exit long*, *enter short* and *exit short*. They are described in Section 4.2.4. The tested values range from 0.0001 to 0.007 - the same as with RMSE Informer strategy. ”-” indicates that the position will not be taken regardless of the model prediction. The possible values of the hyperparameters are listed in Table 17.

Table 17: Hyperparameter space for GMADL Informer Strategy

Parameter	**Values**
enter long	[-, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007]
exit long	[-, -0.001, -0.002, -0.003, -0.004, -0.005, -0.006, -0.007]
enter short	[-, -0.001, -0.002, -0.003, -0.004, -0.005, -0.006, -0.007]
exit short	[-, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007]

Note: Considered values of hyperparameters for GMADL Informer strategy, all 4096 combinations were searched exhaustively.

The selected hyperparameter values are presented in Table 14. Similarly as with the RSI strategy, the values for *exit long* and *exit short* were rarely selected, meaning that throughout the period, the `long position` or `short position` was always open. There seems to be no bias towards larger return threshold in the lower frequency data, indicating that the sudden price changes happen and can be leveraged even in such frequent intervals as 5 min. In case of this strategy, the larger threshold could be selected even for smaller intervals.

Table 18: Selected hyperparameter values for GMADL Informer Strategy

Window	***enter long***	***exit Long***	***enter Short***	***exit Short***
W1-5min	0.004	-	-0.005	-
W2-5min	0.002	-	-0.001	-
W3-5min	-	-	-0.006	0.003
W4-5min	0.002	-	-0.005	-
W5-5min	0.002	-	-0.003	-
W6-5min	0.001	-	-0.007	-
W1-15min	0.005	-	-0.002	-
W2-15min	0.006	-	-0.002	-
W3-15min	-	-	-0.001	0.005
W4-15min	0.007	-	-0.005	-
W5-15min	0.001	-	-0.004	-
W6-15min	0.002	-0.002	-	-
W1-30min	-	-	-0.007	0.005
W2-30min	-	-	-0.007	0.004
W3-30min	-	-	-0.004	0.007
W4-30min	0.003	-0.007	-	-
W5-30min	0.006	-	-0.004	-
W6-30min	0.001	-	-0.005	-

Note: Table presents hyperparameter values selected for the GMADL Informer strategy for each window of 5min, 15min and 30min interval data.

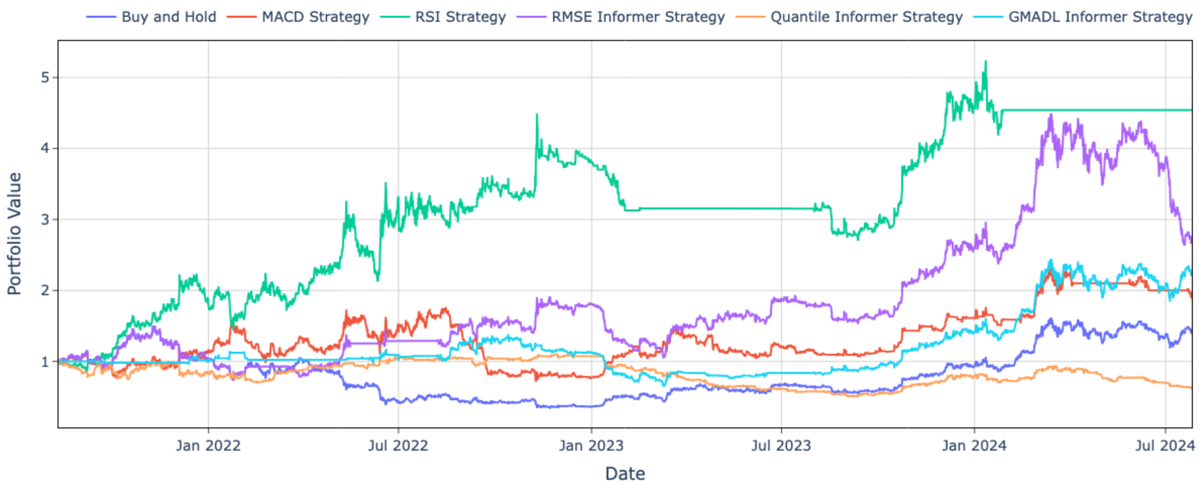
5.2 Evaluation Results

After the hyperparameters were selected, all strategies were evaluated on the corresponding test window parts. The metrics were computed for each window and the whole period by concatenating the positions selected by each strategy for every period. In this section, first, the evaluation results are analyzed separately for each frequency, then the strategies that achieved the best overall results are compared and discussed.

5.2.1 Evaluation on 30 min data

With the largest 30 minute interval dataset, the best strategy turned out to be the one based on RSI indicator. The results presented in Figure 8, show that all strategies except the one based on Quantile Informer signal achieved better results than buy-and-hold strategy. While RSI strategy overall achieved the best results, there were long periods during which this strategy took no position. The RMSE Informer, GMADL Informer and MACD strategy achieved similar results, though it is worth noting that GMADL strategy made much more trades 811 while RMSE Informer strategy changed its position only 34 times.

Figure 8: Evaluation results on 30 min data

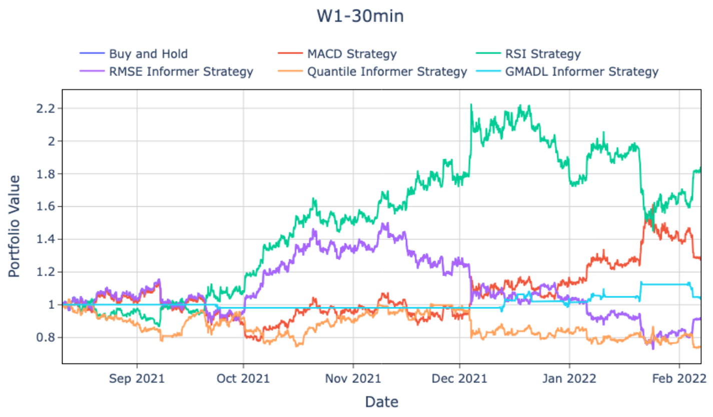


Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.440	13.12%	55.95%	0.235	77.20%	0.040	2	100.00%	0.00%
MACD Strategy	1.952	25.37%	52.36%	0.485	59.24%	0.207	327	52.30%	28.30%
RSI Strategy	4.542	66.77%	46.25%	1.444	39.91%	2.415	377	30.79%	28.03%
RMSE Informer	2.727	40.37%	50.47%	0.800	51.75%	0.624	34	64.40%	24.67%
Quantile Informer	0.629	-14.51%	36.91%	-0.393	55.09%	-0.104	1783	30.24%	15.27%
GMADL Informer	2.263	31.79%	36.70%	0.866	53.35%	0.516	811	35.51%	19.59%

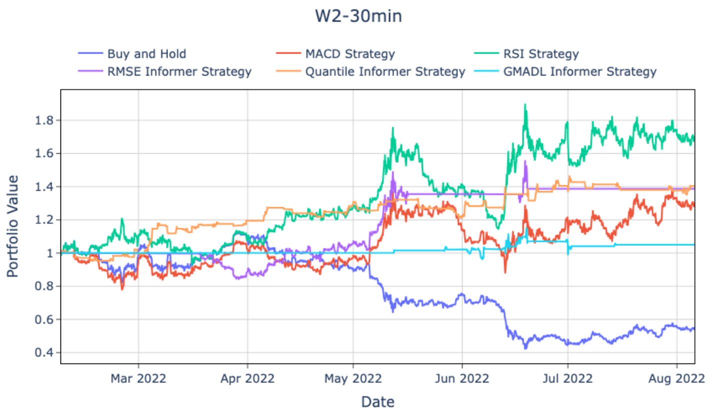
Note: Evaluation results on the whole testing period of 30min interval BTC/USDT data. The presented metrics are: portfolio value at the end of the evaluation period (VAL), Annualized Return Compound (ARD), Annualized Standard Deviation (ASD), Information Ratio (IR*), Maximum Drawdown (MD), Modified Information Ratio (IR**), number of trades (N) and percent of the long/short positions (LONG/SHORT).

Looking at the results in the individual windows (Figure 9) it can be observed that the periods when RSI strategy did not take any posisions happened in 4th and 6th window. In fact, during the first of these periods, the RSI strategy took a position for only 0.02% of the time in the fourth window, and no position in the second. This may suggest that the distribution of validation data on which the hyperparameters were selected differed from the testing data or that there was high uncertainty during those periods. MACD strategy achieved better results than buy-and-hold benchmark in majority of periods, though overall performance was hindered by the poor result during the 3rd period. Informer based strategies with GMADL and RMSE achieved very similar results, with GMADL Informer beased strategy achieving better result when the Bitcoin was in upward trend (5th window) and RMSE Informer based strategy achieving better result when it was in downward trend (2nd window). Quantile Informer strategy had poor performance in 4th and 6th window wich greatly influenced the overall performance. The fact that these were the windows in which the best strategy RSI took no position and that the second best strategy in both of these periods was buy-and-hold approach may suggest that they were the most difficult periods.

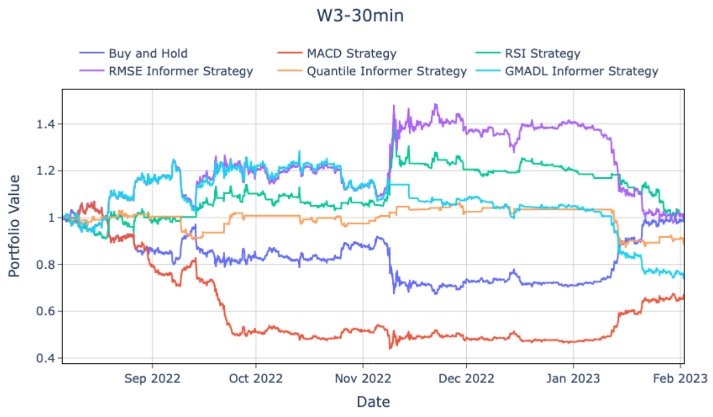
Figure 9: Evaluation results for individual windows on 30 min data



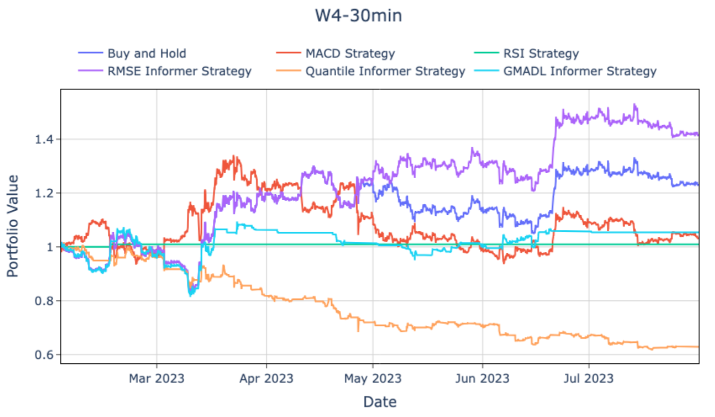
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	0.924	-14.84%	66.12%	-0.225	51.75%	-0.064	2	100.00%	0.00%
MACD Strategy	1.269	62.19%	65.90%	0.944	31.81%	1.845	15	50.36%	47.71%
RSI Strategy	1.843	245.59%	66.51%	3.693	35.02%	25.897	26	42.40%	57.60%
RMSE Informer	0.924	-14.84%	66.12%	-0.225	51.75%	-0.064	2	100.00%	0.00%
Quantile Informer	0.743	-45.20%	60.57%	-0.746	29.11%	-1.159	443	30.43%	53.98%
GMADL Informer	1.030	6.23%	19.98%	0.312	10.01%	0.194	39	0.00%	8.72%



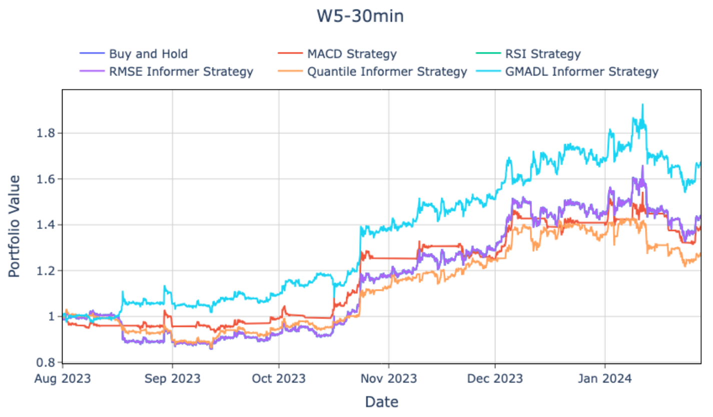
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	0.548	-70.51%	72.38%	-0.974	63.18%	-1.087	2	100.00%	0.00%
MACD Strategy	1.303	70.93%	71.82%	0.988	36.06%	1.943	19	57.32%	40.76%
RSI Strategy	1.703	194.24%	72.36%	2.684	34.89%	14.945	106	39.96%	60.04%
RMSE Informer	1.387	94.09%	41.55%	2.264	16.34%	13.037	8	0.00%	34.41%
Quantile Informer	1.403	98.74%	27.62%	3.575	9.44%	37.393	255	0.00%	18.79%
GMADL Informer	1.050	10.44%	23.40%	0.446	14.45%	0.322	90	0.00%	5.98%



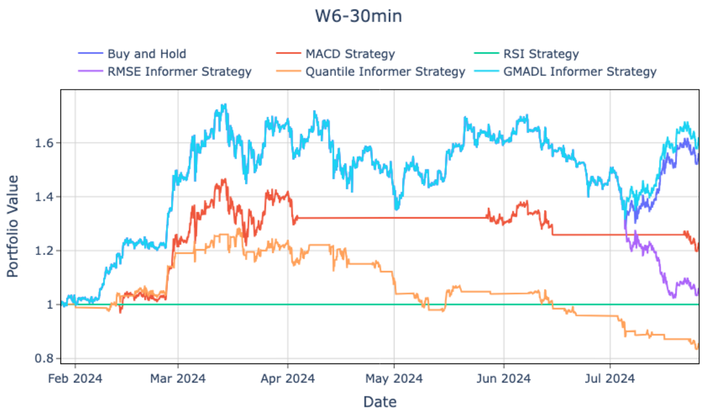
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.018	3.75%	51.25%	0.073	37.47%	0.007	2	100.00%	0.00%
MACD Strategy	0.673	-55.16%	51.01%	-1.081	59.24%	-1.007	95	81.60%	16.48%
RSI Strategy	0.988	-2.50%	38.04%	-0.066	30.42%	-0.005	238	2.34%	50.54%
RMSE Informer	0.980	-4.00%	51.36%	-0.078	34.51%	-0.009	2	0.00%	100.00%
Quantile Informer	0.884	-22.19%	23.41%	-0.948	18.39%	-1.144	171	0.00%	18.83%
GMADL Informer	0.737	-46.11%	42.58%	-1.083	42.71%	-1.169	302	0.00%	81.21%



Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.229	51.90%	45.01%	1.153	21.74%	2.753	2	100.00%	0.00%
MACD Strategy	1.031	6.45%	45.08%	0.143	30.15%	0.031	94	35.18%	64.82%
RSI Strategy	1.010	2.03%	1.87%	1.084	0.30%	7.410	4	0.02%	0.00%
RMSE Informer	1.414	101.80%	45.03%	2.260	21.74%	10.584	10	92.95%	7.05%
Quantile Informer	0.629	-60.99%	28.76%	-2.121	39.23%	-3.297	448	36.32%	0.00%
GMADL Informer	1.054	11.31%	28.25%	0.400	24.18%	0.187	345	34.69%	0.00%



Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.439	109.30%	41.49%	2.634	20.18%	14.265	2	100.00%	0.00%
MACD Strategy	1.393	95.71%	31.47%	3.041	15.00%	19.407	90	48.66%	0.00%
RSI Strategy	1.439	109.30%	41.49%	2.634	20.18%	14.265	2	100.00%	0.00%
RMSE Informer	1.439	109.30%	41.49%	2.634	20.18%	14.265	2	100.00%	0.00%
Quantile Informer	1.277	64.08%	34.28%	1.869	16.61%	7.212	311	76.90%	0.00%
GMADL Informer	1.672	183.71%	41.56%	4.420	20.18%	40.230	22	78.48%	21.52%



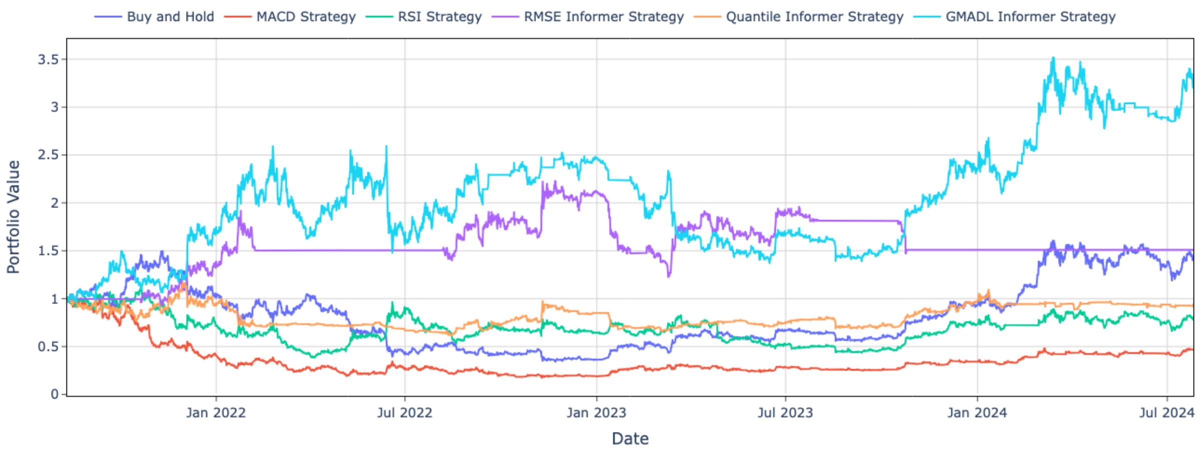
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.559	145.95%	52.85%	2.762	26.69%	15.103	2	100.00%	0.00%
MACD Strategy	1.227	51.44%	36.38%	1.414	18.55%	3.921	11	40.69%	0.00%
RSI Strategy	1	0.00%	0.00%	0	0.00%	0	0	0.00%	0.00%
RMSE Informer	1.059	12.32%	52.86%	0.233	41.44%	0.069	10	93.44%	6.56%
Quantile Informer	0.855	-27.14%	34.50%	-0.787	35.23%	-0.606	150	37.79%	0.00%
GMADL Informer	1.617	165.10%	52.85%	3.124	24.64%	20.929	10	99.88%	0.12%

Note: Evaluation results for each of the out-of-sample data window of 30min interval BTC/USDT data. The presented metrics are: portfolio value at the end of the evaluation period (VAL), Annualized Return Compound (ARD), Annualized Standard Deviation (ASD), Information Ratio (IR*), Maximum Drawdown (MD), Modified Information Ratio (IR**), number of trades (N) and percent of the long/short positions (LONG/SHORT).

5.2.2 Evaluation on 15 min data

The results for the 15 min data are presented in Figure 10. Here, the GMADL Informer strategy yielded the best results. The modified information ratio was was the best one from the strategies evaluated on 15 min interval. The next best strategy was RMSE Informer and the 3rd best buy-and-hold benchmark. The strategies based on technical indicators as well as Quantile Informer performed poorly, achieving results worse than buy-and-hold. Quantile Informer Strategy had a position only for the 40% of time suggesting the selection of the hyperparameters that too aggressively limit the event when the position is entered/closed. MACD and RSI Strategies had more position changes, which might have negatively contributed to their performance due to high exchange fees.

Figure 10: Evaluation results on 15 min data

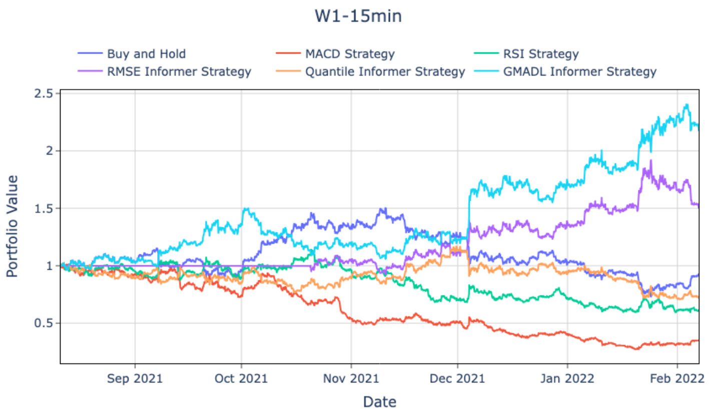


Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.440	13.10%	56.03%	0.234	77.23%	0.040	2	100.00%	0.00%
MACD Strategy	0.468	-22.64%	52.43%	-0.432	83.18%	-0.118	1311	51.80%	31.33%
RSI Strategy	0.800	-7.28%	55.66%	-0.131	66.67%	-0.014	1206	54.02%	43.08%
RMSE Informer	1.509	14.93%	34.90%	0.428	45.54%	0.140	16	15.24%	27.60%
Quantile Informer	0.945	-1.91%	37.77%	-0.051	48.30%	-0.002	824	28.48%	16.77%
GMADL Informer	3.296	49.65%	52.70%	0.942	47.39%	0.987	362	49.37%	37.72%

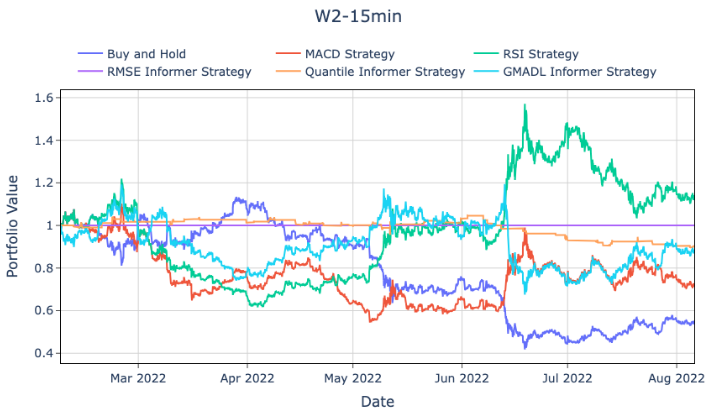
Note: Evaluation results on the whole testing period of 15min interval BTC/USDT data. The presented metrics are: portfolio value at the end of the evaluation period (VAL), Annualized Return Compound (ARD), Annualized Standard Deviation (ASD), Information Ratio (IR*), Maximum Drawdown (MD), Modified Information Ratio (IR**), number of trades (N) and percent of the long/short positions (LONG/SHORT).

The results of the 15-minute individual windows presented in Figure 11, show that the GMADL Informer strategy was the best only in two windows (first and third) and the majority of the value was accumulated during the first period. What is more it was the worst performing strategy in the fourth period. The RMSE Informer strategy also accumulated most of the value during the first period. Also it did not perform almost any trades during 2nd, 5th and 6th periods. This may suggest that the signal from Informer model was too weak - all predictions of returns were too close to 0 and the larger price movements that would incur position change were not detected correctly. Although the Quantile Informer strategy again poor overall results, it was the best performing strategy in the fourth window. However, during the other periods the positions either were changed too frequently or were missing on the high price swings. We can again observe that the last period proved to be the most difficult, and no strategy beat the buy-and-hold benchmark during that period. Despite poor overall results, MACD Strategy wasn't significantly worse than the other strategies except on the first window, on which a significant loss of portfolio value (value at the end of this period was 0.35) influenced the result on the whole testing period.

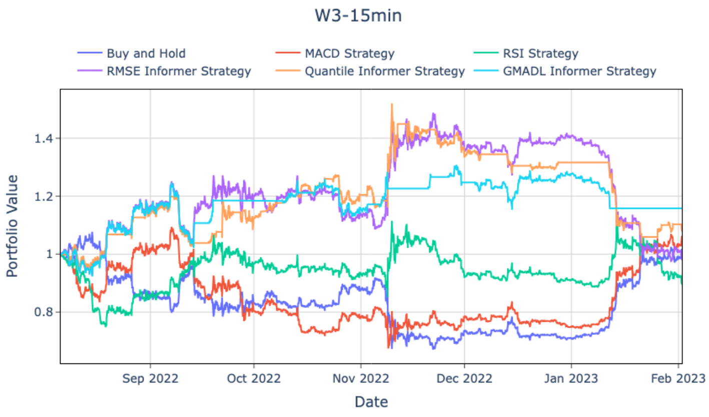
Figure 11: Evaluation results for individual windows on 15 min data



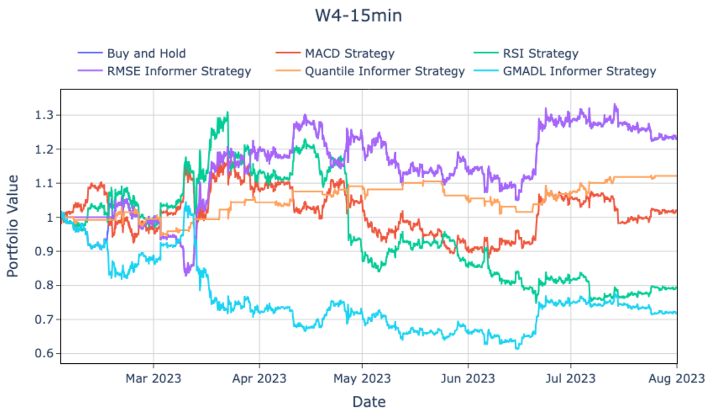
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	0.933	-13.15%	66.69%	-0.197	51.81%	-0.050	2	100.00%	0.00%
MACD Strategy	0.357	-87.59%	66.96%	-1.308	73.46%	-1.559	852	52.49%	47.51%
RSI Strategy	0.621	-61.93%	66.94%	-0.925	48.44%	-1.183	882	58.61%	41.39%
RMSE Informer	1.498	127.03%	52.56%	2.417	22.20%	13.827	3	0.00%	61.05%
Quantile Informer	0.715	-49.34%	66.74%	-0.739	40.17%	-0.908	182	52.75%	47.25%
GMADL Informer	2.173	382.27%	66.83%	5.720	29.76%	73.474	146	36.88%	63.12%



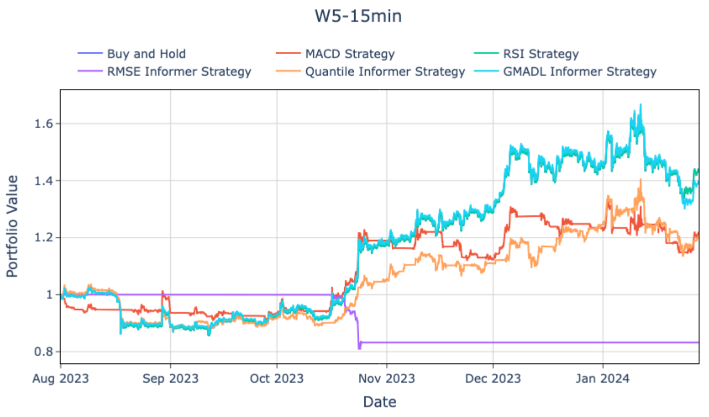
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	0.548	-70.48%	72.12%	-0.977	63.18%	-1.090	2	100.00%	0.00%
MACD Strategy	0.714	-49.48%	72.07%	-0.687	50.67%	-0.670	118	57.61%	42.39%
RSI Strategy	1.141	30.61%	72.05%	0.425	49.51%	0.263	58	19.64%	80.36%
RMSE Informer	1	0.00%	0.00%	0	0.00%	0	0	0.00%	0.00%
Quantile Informer	0.903	-18.73%	15.66%	-1.196	14.91%	-1.503	202	4.74%	0.00%
GMADL Informer	0.885	-21.88%	72.10%	-0.303	43.48%	-0.153	54	35.85%	64.15%



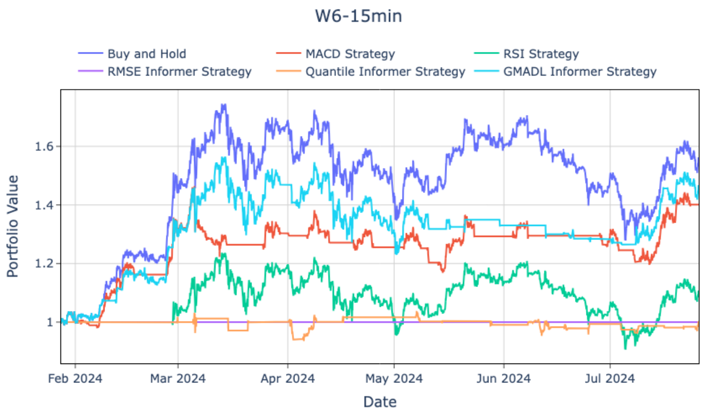
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.016	3.26%	50.58%	0.065	37.76%	0.006	2	100.00%	0.00%
MACD Strategy	1.060	12.62%	50.57%	0.250	38.35%	0.082	55	66.38%	32.66%
RSI Strategy	0.898	-19.62%	50.65%	-0.387	25.20%	-0.302	174	23.30%	76.70%
RMSE Informer	0.982	-3.55%	50.69%	-0.070	34.60%	-0.007	2	0.00%	100.00%
Quantile Informer	1.081	17.17%	40.53%	0.424	31.82%	0.229	103	0.00%	53.36%
GMADL Informer	1.158	34.60%	31.80%	1.088	18.06%	2.085	32	0.00%	59.07%



Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.231	52.34%	44.11%	1.186	22.01%	2.822	2	100.00%	0.00%
MACD Strategy	1.020	4.03%	44.11%	0.091	25.36%	0.015	74	34.59%	65.41%
RSI Strategy	0.794	-37.41%	44.15%	-0.848	43.03%	-0.737	86	39.98%	60.02%
RMSE Informer	1.232	52.63%	42.30%	1.244	22.01%	2.975	3	91.43%	0.00%
Quantile Informer	1.122	26.21%	21.18%	1.238	9.32%	3.481	106	24.72%	0.00%
GMADL Informer	0.718	-48.86%	44.08%	-1.108	41.58%	-1.302	10	60.34%	39.66%



Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.440	109.38%	44.60%	2.452	20.31%	13.204	2	100.00%	0.00%
MACD Strategy	1.218	49.24%	33.63%	1.464	13.88%	5.195	118	50.76%	0.00%
RSI Strategy	1.440	109.38%	44.60%	2.452	20.31%	13.204	2	100.00%	0.00%
RMSE Informer	0.832	-31.06%	13.66%	-2.274	19.29%	-3.662	4	0.00%	4.56%
Quantile Informer	1.206	46.21%	37.95%	1.218	19.49%	2.887	147	78.97%	0.00%
GMADL Informer	1.398	97.31%	44.61%	2.181	22.22%	9.554	26	99.64%	0.36%



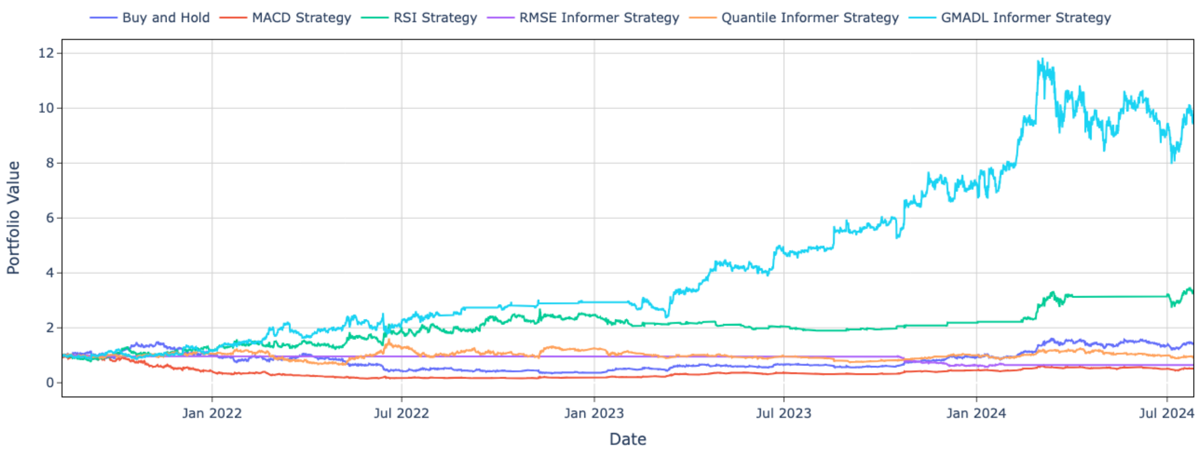
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.558	145.73%	51.90%	2.808	26.76%	15.290	2	100.00%	0.00%
MACD Strategy	1.402	98.29%	34.32%	2.864	20.22%	13.925	93	48.96%	0.00%
RSI Strategy	1.104	22.18%	49.12%	0.452	26.76%	0.374	3	82.57%	0.00%
RMSE Informer	1	0.00%	0.00%	0	0.00%	0	0	0.00%	0.00%
Quantile Informer	1.000	0.04%	18.06%	0.002	9.32%	0.000	81	9.73%	0.00%
GMADL Informer	1.463	116.41%	45.31%	2.569	21.45%	13.942	94	63.52%	0.00%

Note: Evaluation results for each of the out-of-sample data window of 15min interval BTC/USDT data. The presented metrics are: portfolio value at the end of the evaluation period (VAL), Annualized Return Compound (ARD), Annualized Standard Deviation (ASD), Information Ratio (IR*), Maximum Drawdown (MD), Modified Information Ratio (IR**), number of trades (N) and percent of the long/short positions (LONG/SHORT).

5.2.3 Evaluation on 5 min data

Finally evaluation results for the 5 min interval throughout the testing period are presented in Figure 12, a GMADL Informer strategy significantly outperformed the other strategies. Throughout the testing period, it achieved annualized returns of 115%, while maintaining annualized standard division at a level similar to the buy-and-hold benchmark. What is more, the maximum drawdown of this strategy was only 32.7% which is significantly lower than the 77.3% of Buy and Hold. The strategy kept the long position and *short position* for approximately a similar period of time. The next best strategy was an RSI based strategy, which however for almost 40% of time did not hold a position. These are the only two strategies that beat the benchmark. It is worth noting that they had much fewer positions changes than the other strategies that performed worse, suggesting that they were better at detecting the trend changes and filtering out the insignificant signal. Both Quantile Informer and RMSE Informer achieved poor results, suggesting that these loss functions are not fit for higher frequency data. Though each for a different reason. RMSE Informer based strategy took only 16 trades during the whole period further showing the effect initially observed on 15 min interval. That with the many smaller returns model trained with this loss function has difficulty correctly predicting large returns (since when predicted incorrectly they are heavily penalized) and thus most of the model predictions are close to zero, falling below the exchange fee making it impossible to build strategy upon. On the other hand, Quantile informer performed the most trades from all the strategy suggesting the difficulty with filtering out insignificant signals, and in process loosing lots of value due to transaction fees.

Figure 12: Evaluation results on 5 min data



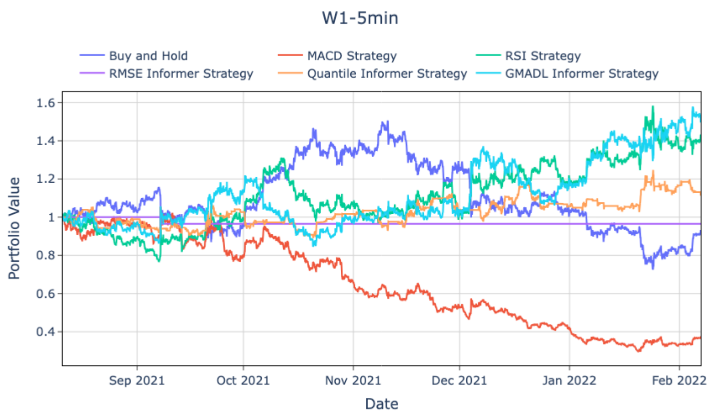
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.441	13.14%	57.74%	0.228	77.31%	0.039	2	100.00%	0.00%
MACD Strategy	0.516	-20.04%	54.14%	-0.370	85.77%	-0.087	2535	50.39%	32.38%
RSI Strategy	3.341	50.34%	50.41%	0.999	29.99%	1.676	846	28.29%	33.47%
RMSE Informer	0.643	-13.88%	15.13%	-0.917	44.61%	-0.285	16	0.00%	9.58%
Quantile Informer	0.956	-1.52%	47.91%	-0.032	53.96%	-0.001	3395	40.24%	27.84%
GMADL Informer	9.747	115.88%	54.44%	2.129	32.66%	7.552	846	44.80%	41.51%

Note: Evaluation results on the whole testing period of 5min interval BTC/USDT data. The presented metrics are: portfolio value at the end of the evaluation period (VAL), Annualized Return Compound (ARD), Annualized Standard Deviation (ASD), Information Ratio (IR*), Maximum Drawdown (MD), Modified Information Ratio (IR**), number of trades (N) and percent of the long/short positions (LONG/SHORT).

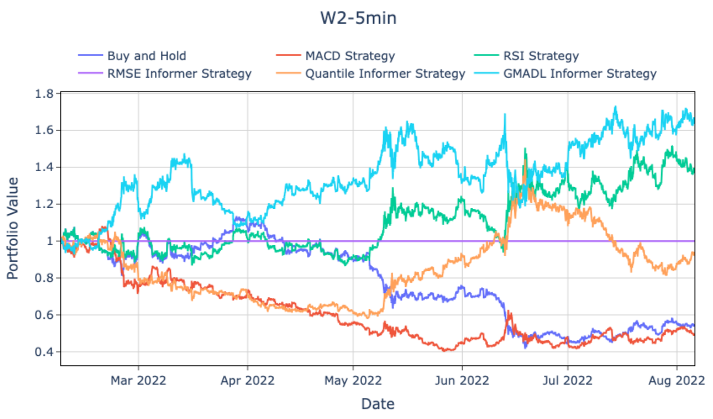
Looking at the evaluation results for each of the data windows separately, presented in Figure 13, confirms the superiority of the GMADL informer strategy. It was the top performing strategy on all the evaluation periods but the last one, on which the best performing strategy was Buy and Hold, suggesting this was the most difficult period to trade. Most of the GMADL Strategy trades were done during the second window period, while during the other periods the positions were changed sparingly. In the third period the number of transactions was the smallest, only 16 and the strategy did not hold any position for 80% of the time; however, it still achieved the best results compared to the other strategies, as it correctly predicted and leveraged the sudden price swings.

The RSI strategy performed quite well on the first three periods; however, on the next three the quality suddenly dropped, again suggesting the market being more unpredictable during past two windows.

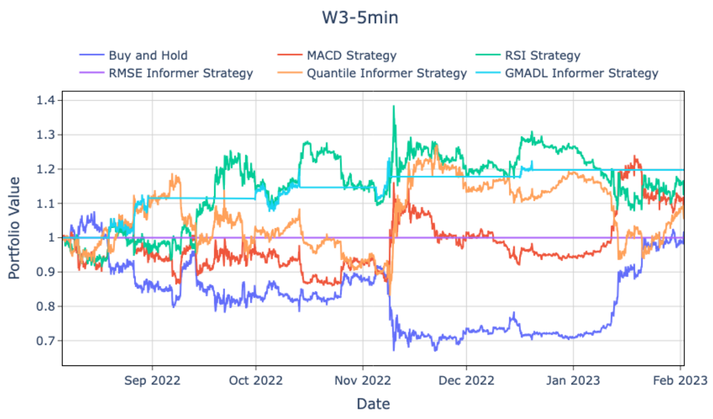
Figure 13: Evaluation results for individual windows on 5 min data



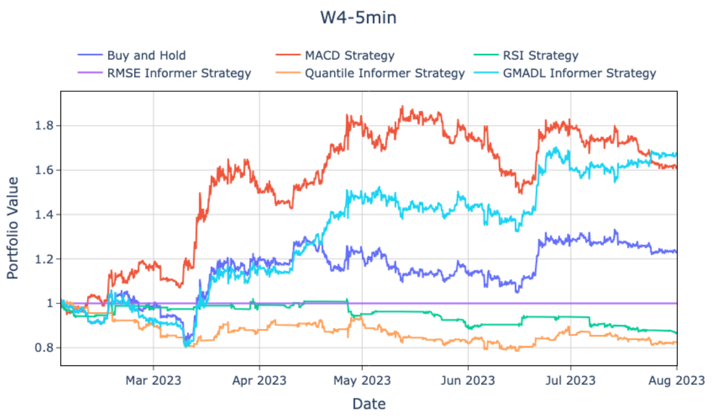
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	0.930	-13.76%	70.24%	-0.196	51.87%	-0.052	2	100.00%	0.00%
MACD Strategy	0.375	-86.35%	70.71%	-1.221	71.90%	-1.467	1542	52.05%	47.95%
RSI Strategy	1.428	106.05%	70.55%	1.503	26.24%	6.074	114	26.31%	73.69%
RMSE Informer	0.965	-6.93%	3.11%	-2.227	3.54%	-4.361	12	0.02%	0.00%
Quantile Informer	1.117	25.15%	43.69%	0.576	16.27%	0.889	365	0.00%	45.14%
GMADL Informer	1.502	128.10%	70.54%	1.816	30.70%	7.576	82	41.54%	58.46%



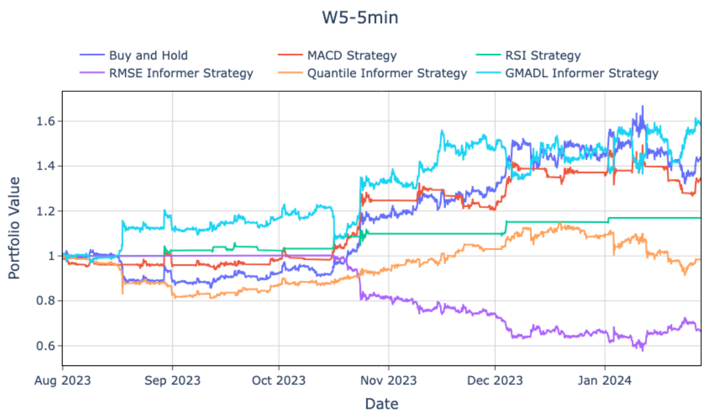
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	0.549	-70.37%	74.61%	-0.943	63.35%	-1.048	2	100.00%	0.00%
MACD Strategy	0.498	-75.63%	74.63%	-1.013	62.97%	-1.217	446	50.72%	49.28%
RSI Strategy	1.368	88.76%	74.58%	1.190	27.34%	3.864	78	61.40%	38.60%
RMSE Informer	1	0.00%	0.00%	0	0.00%	0	0	0.00%	0.00%
Quantile Informer	0.941	-11.54%	74.69%	-0.154	44.84%	-0.040	990	50.50%	49.50%
GMADL Informer	1.635	170.94%	74.67%	2.289	30.47%	12.844	522	20.20%	79.80%



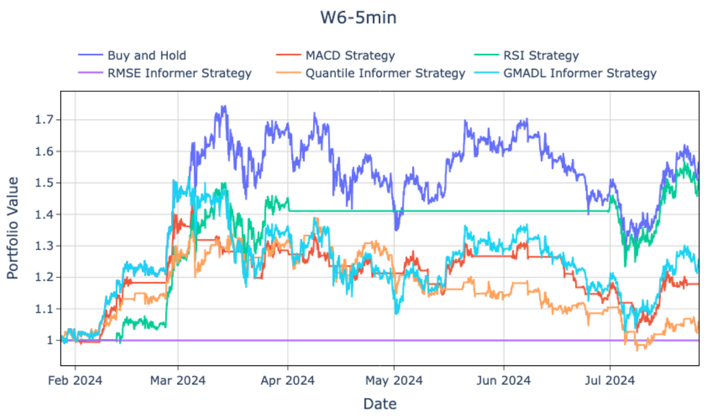
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.016	3.23%	51.77%	0.062	37.98%	0.005	2	100.00%	0.00%
MACD Strategy	1.081	17.10%	51.79%	0.330	20.87%	0.270	154	58.24%	41.76%
RSI Strategy	1.127	27.41%	51.92%	0.528	22.30%	0.649	398	11.48%	88.52%
RMSE Informer	1	0.00%	0.00%	0	0.00%	0	0	0.00%	0.00%
Quantile Informer	1.057	11.93%	51.90%	0.230	30.10%	0.091	894	27.62%	72.38%
GMADL Informer	1.198	44.15%	19.18%	2.302	7.68%	13.227	16	0.00%	17.86%



Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.231	52.35%	45.25%	1.157	22.29%	2.718	2	100.00%	0.00%
MACD Strategy	1.612	163.18%	45.34%	3.599	21.23%	27.660	190	44.69%	55.31%
RSI Strategy	0.866	-25.30%	19.03%	-1.330	15.65%	-2.150	205	23.63%	0.00%
RMSE Informer	1	0.00%	0.00%	0	0.00%	0	0	0.00%	0.00%
Quantile Informer	0.825	-32.24%	27.88%	-1.156	22.72%	-1.641	290	44.45%	0.00%
GMADL Informer	1.673	183.80%	45.26%	4.061	24.78%	30.125	62	65.14%	34.86%



Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.439	109.23%	44.72%	2.443	21.07%	12.664	2	100.00%	0.00%
MACD Strategy	1.346	82.57%	32.85%	2.514	15.44%	13.447	94	49.06%	0.00%
RSI Strategy	1.169	37.33%	13.86%	2.692	4.73%	21.237	42	5.67%	0.00%
RMSE Informer	0.667	-56.06%	36.94%	-1.518	42.77%	-1.989	3	0.00%	57.45%
Quantile Informer	0.984	-3.12%	33.97%	-0.092	20.63%	-0.014	578	59.26%	0.00%
GMADL Informer	1.582	153.60%	45.11%	3.405	14.37%	36.394	130	44.70%	55.30%



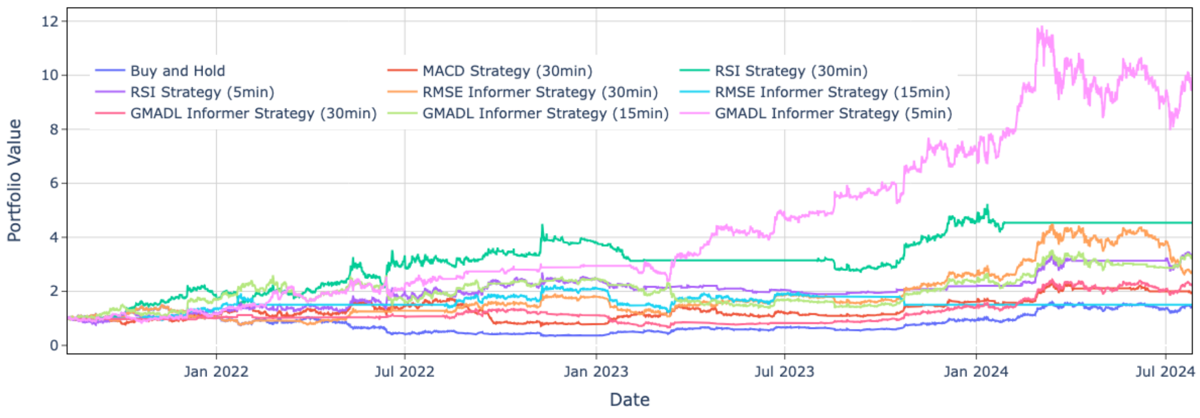
Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.560	146.27%	52.72%	2.775	26.96%	15.054	2	100.00%	0.00%
MACD Strategy	1.179	39.63%	34.65%	1.144	28.87%	1.570	111	47.58%	0.00%
RSI Strategy	1.507	129.58%	38.15%	3.396	17.97%	24.493	7	41.29%	0.00%
RMSE Informer	1	0.00%	0.00%	0	0.00%	0	0	0.00%	0.00%
Quantile Informer	1.056	11.76%	40.73%	0.289	30.60%	0.111	280	59.60%	0.00%
GMADL Informer	1.253	57.92%	52.71%	1.099	32.66%	1.949	34	97.21%	2.79%

Note: Evaluation results for each of the out-of-sample data window of 5min interval BTC/USDT data. The presented metrics are: portfolio value at the end of the evaluation period (VAL), Annualized Return Compound (ARD), Annualized Standard Deviation (ASD), Information Ratio (IR*), Maximum Drawdown (MD), Modified Information Ratio (IR**), number of trades (N) and percent of the long/short positions (LONG/SHORT).

5.2.4 Best strategies

The strategies that performed better than the buy-and-hold benchmark are again presented in Figure 14.

Figure 14: Best strategies



Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.441	0.131	0.577	0.228	0.773	0.039	2	100.00%	0.00%
MACD Strategy (30min)	1.952	0.254	0.524	0.485	0.592	0.207	327	52.30%	28.30%
RSI Strategy (30min)	4.542	0.668	0.462	1.444	0.399	2.415	377	30.79%	28.03%
RSI Strategy (5min)	3.341	0.503	0.504	0.999	0.300	1.676	846	28.29%	33.47%
RMSE Informer (30min)	2.727	0.404	0.505	0.800	0.518	0.624	34	64.40%	24.67%
RMSE Informer (15min)	1.509	0.149	0.349	0.428	0.455	0.140	16	15.24%	27.60%
GMADL Informer (30min)	2.263	0.318	0.367	0.866	0.533	0.516	811	35.51%	19.59%
GMADL Informer (15min)	3.296	0.496	0.527	0.942	0.474	0.987	362	49.37%	37.72%
GMADL Informer (5min)	9.747	1.159	0.544	2.129	0.327	7.552	846	44.80%	41.51%

Note: Strategies that achieved better performance than the buy-and-hold benchmark evaluated on testing period with BTC/USDT data. The presented metrics are: portfolio value at the end of the evaluation period (VAL), Annualized Return Compound (ARD), Annualized Standard Deviation (ASD), Information Ratio (IR*), Maximum Drawdown (MD), Modified Information Ratio (IR**), number of trades (N) and percent of the long/short positions (LONG/SHORT).

GMADL Informer strategy seems to be benefiting from the higher frequency data, with the IR** value monotonically increasing when the data frequency is increased. An opposite relation can be observed with the RMSE Informer, where the performance of the strategy decreases as the data interval becomes smaller. These relations seem consistent with assumptions that were made when designing the GMADL loss function, that it is better at pushing the model towards correctly predicting the direction of the return (positive vs negative), especially with sudden large changes, and that this signal can be better utilized for constructing automated trading strategies.

The RSI-based strategy turned out to give quite good results, both for 5min and 30min data. This indicates that despite the long-standing presence of technical indicators and the availability of more advanced and intricate models for generating trading signals, these indicators remain valuable and can be employed to develop effective trading strategies. MACD strategy was performing poorly for the higher frequency data, however, for the 30 min interval data the strategy was in the top five best. This might indicate that the MACD oscillator might not be applicable for the higher frequencies but is still useful for the lower frequencies. This hypothesis seems to be confirmed by the hyperparameters selected for the strategies that had a bias towards longer sequences.

The Quantile Informer strategy did not beat the Buy and Hold benchmark. In fact, it was one of the worst performing strategies. This may suggests that Quantile loss function is not the best choice for training a model for trading strategy or the proposed strategy do not use the model predictions effectively. However, more in depth analysis should be carried out, testing the training with various quantile combinations to confirm this hypothesis. Specifically, in this study a decision has been made to use a Quantile Loss with many different quantiles and then selecting the quantile value for entering/exiting the position as a part of strategy hyperparameter search. This method provided the advantage of eliminating the requirement to retrain the model every time a new quantile pair is selected. However, training the model once with many quantiles could have impact on the predictions quality. It may also suggest that the strategy that uses signal from the Quantile Informer model was build improperly and should be revisited. A shape of the distribution e.g. by including information about the variance, could be utilised in the strategy, possibly making it better in selecting the relevant signals.

To verify statistical significance of the results, a probabilistic t-test that compares the Information Ratios of the best strategies against the buy-and-hold benchmark was conducted. The test uses a simplified formula for the standard error of Sharpie ratio, expressed as $\frac{\sigma}{\sqrt{N}}$, where N is the sample size of the returns and σ is the standard deviation of the excess of the strategy return over the benchmark return. The full test statistic is then as follows:

$$t = \frac{IR^*_{\text{strategy}} - IR^*_{\text{benchmark}}}{\frac{\sigma}{\sqrt{N}}}$$

The test assumes the differences in returns follow a normal distribution. The results are presented in Table 19. The null hypothesis that the Information Ratio of the top performing strategies is not greater than the buy-and-hold Modified Information Ratio was rejected for all the tested strategies.

Table 19: Statistical t-test for comparing the performance of strategies over buy-and-hold.

Strategy	**N**	σ	**t-statistic**	**p-value**
MACD (30 min)	51840	0.326504	174.34	0.000000***
RSI (30 min)	51840	1.065079	258.49	0.000000***
RSI (5 min)	311040	0.648741	662.77	0.000000***
RMSE (30 min)	51840	0.796647	161.58	0.000000***
RMSE (15 min)	103680	0.541611	115.29	0.000000***
GMADL (30 min)	51840	0.320689	448.49	0.000000***
GMADL (15 min)	103680	0.558743	408.13	0.000000***
GMADL (5min)	311040	2.820834	375.84	0.000000***

Note: T-test H0: The information Ratio of the strategy is not greater than the buy-and-hold information ratio. The values marked with *** indicate the p-value lower than the critical value $\alpha = 0.01$, rejecting the null hypothesis.

6 Sensitivity Analysis

The sensitivity analysis of the parameters of the Informer model requires retraining of the Informer model for each modified parameter and is outside the scope of this study due to the high computational cost. However, this section presents a brief analysis of how factors that impact the selection of strategy hyperparameters influence the strategy performance.

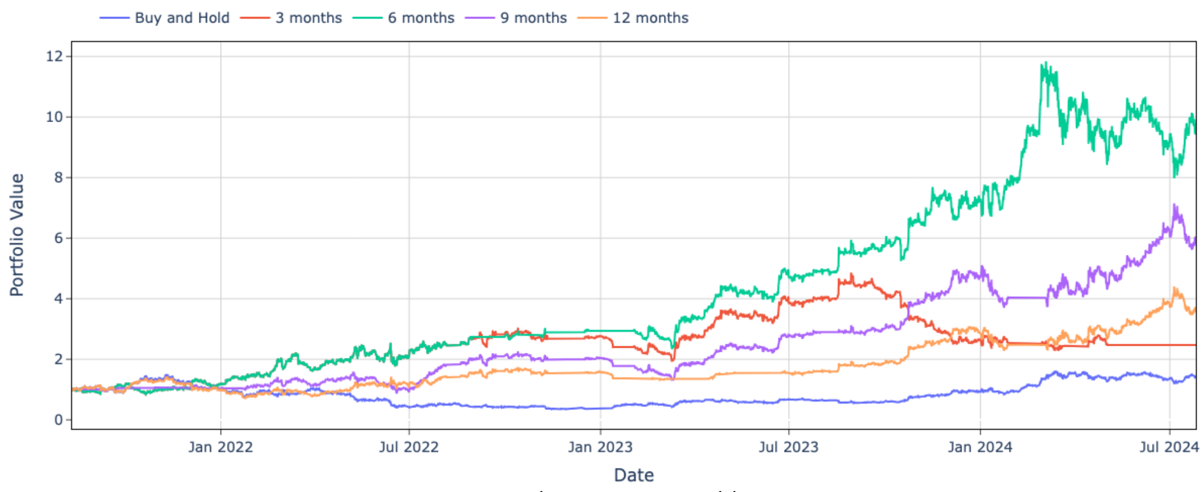
6.1 Validation part size

The hyperparameter selection for the strategies (described in Section 5.1) was carried out using the validation part of the data windows. The size of this part was arbitrarily selected to be 0.2 of the out-of-sample data size, which, in terms of the length of the period, is roughly equal to 6 months. The length of this period influences what strategy hyperparamers are selected. If the period is too short, the data sample would not be representative enough to select meaningful values for the hyperparameters. On the other hand, if the period is too long, the strategy might not be relevant to the current market conditions. Thus, it is important to perform an analysis of how this parameter influence the strategy performance.

The analysis has been carried out only for the best strategies, that is GMADL Informer with 5min data.

The results for evaluating GMADL Informer on 5 minute interval dataset when selecting hyperparameters using various validation part sizes are presented on Figure 15. For this strategy, selecting a longer or shorter validation period than 6 months resulted in decreased strategy performance. However, regardless of the length of the validation part, the strategy outperformed the benchmark, and Modified Information Ratio stayed on relatively high level. It can be seen that the number of trades decreases monotonically when the validation window is elongated. This suggests that the longer validation periods require the strategy to adapt to varying market conditions and more aggressively reduce the signals to change positions. The worst performing variant was with the validation windows length of three months; in this case the strategy recorded significant losses during the last two windows of the testing period.

Figure 15: GMADL Informer strategies with parameters selected using different lengths of validation windows



Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.441	13.1%	57.7%	0.228	77.3%	0.039	2	100.0%	0.0%
3 months	2.473	35.8%	52.5%	0.682	53.3%	0.458	912	27.0%	51.6%
6 months	9.747	115.9%	54.4%	2.129	32.7%	7.552	846	44.8%	41.5%
9 months	5.847	81.6%	48.5%	1.683	45.1%	3.048	300	38.7%	35.5%
12 months	3.602	54.2%	52.2%	1.038	51.7%	1.088	214	38.1%	39.7%

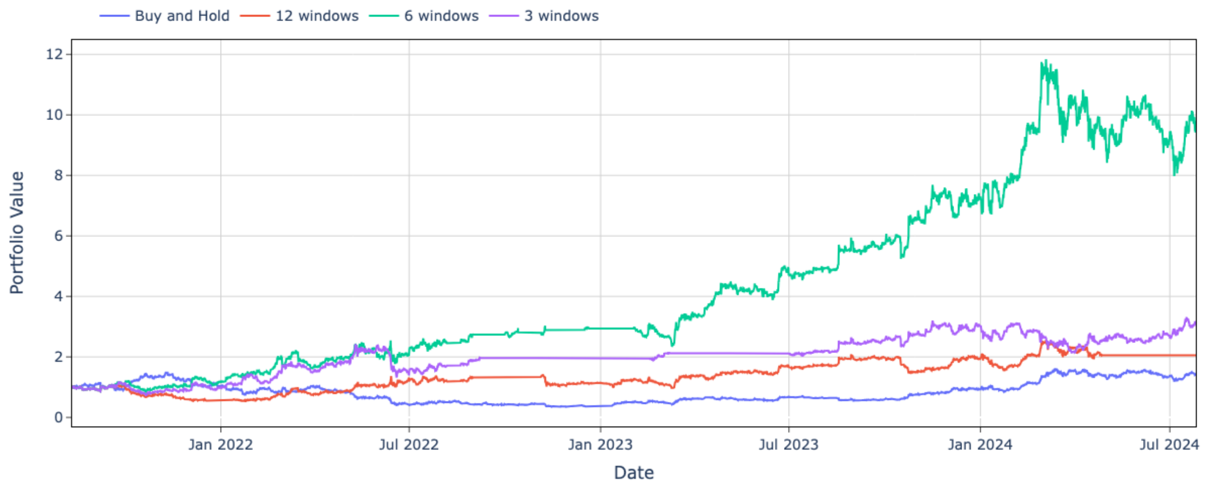
Note: Best GMADL Informer strategy with hyperparameters selected on various lengths of validation windows. The presented metrics are: portfolio value at the end of the evaluation period (VAL), Annualized Return Compound (ARD), Annualized Standard Deviation (ASD), Information Ratio (IR*), Maximum Drawdown (MD), Modified Information Ratio (IR**), number of trades (N) and percent of the long/short positions (LONG/SHORT).

6.2 Number of data windows

Another implicit parameter that may affect the evaluation results is the number of data windows in a period in which the evaluation is carried out. In the original experiment, six data windows, with 6-month out-of-sample testing periods were considered. This section explores how the result of the best strategy - GMADL Informer with 5 min data - is affected, if the testing period is split differently: into three or twelve windows. Note that while selecting a different number of windows affects the length of the testing period of each window, the total testing period does not change and the in-sample (training and validation parts) length was kept constant.

Evaluation of the GMADL Informer strategy with a different number of windows required retraining of the underlying Informer model with the new data. For each of the cases considered: three and twelve windows, a new model was trained for each window, keeping the values of all the other hyperparameters the same as in the base scenario. The evaluation results are presented in Figure 16. It can be seen that the performance of the strategy was worse both when the number of windows increased and decreased. Although the Modified Information Ratio was still higher than the benchmark, it was significantly lower than the one in the six-window evaluation. This may suggest that such an excellent performance of the GMADL Informer strategy might have been an isolated event caused by the selection of a concrete number of evaluation windows. However, this conclusion should not belittle the fact that the GMADL Informer strategy, evaluated on different numbers of windows, still achieves impressive results compared to the benchmark.

Figure 16: GMADL Informer strategies evaluated on different number of windows



Strategy	VAL	ARC	ASD	IR*	MD	IR**	N	LONG	SHORT
Buy and Hold	1.44	13.1%	57.7%	0.23	77.3%	0.04	2	100.0%	0.0%
12 windows	2.06	27.6%	53.1%	0.52	53.3%	0.27	568	43.0%	37.9%
6 windows	9.75	115.9%	54.4%	2.13	32.7%	7.55	846	44.8%	41.5%
3 windows	3.08	46.2%	51.6%	0.90	45.9%	0.90	426	27.1%	45.9%

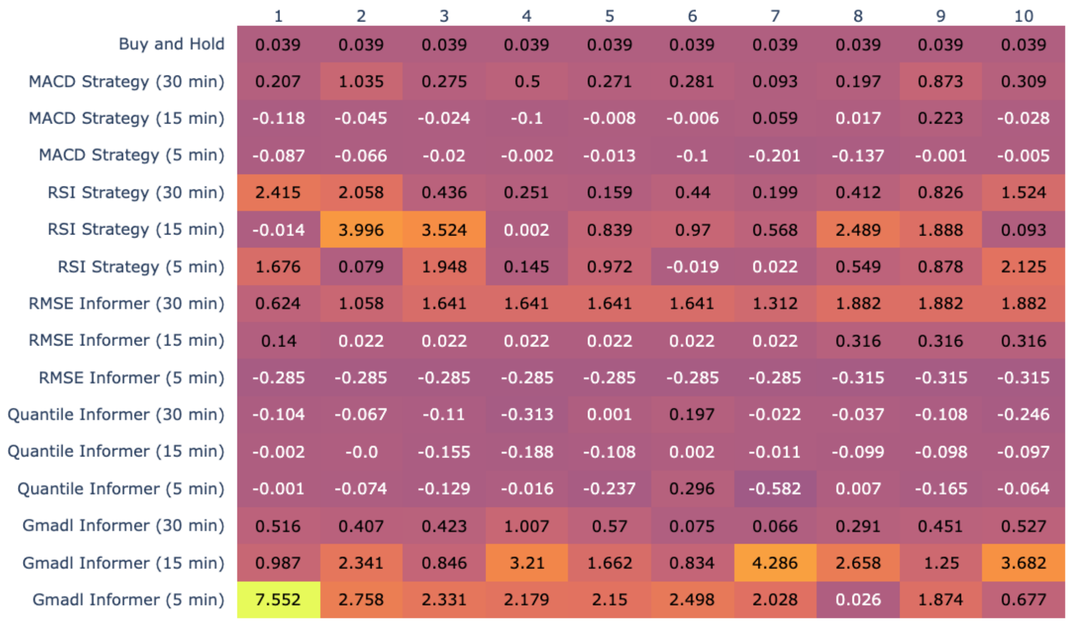
Note: Best GMADL Informer strategy evaluated on different number of windows. The presented metrics are: portfolio value at the end of the evaluation period (VAL), Annualized Return Compound (ARD), Annualized Standard Deviation (ASD), Information Ratio (IR*), Maximum Drawdown (MD), Modified Information Ratio (IR**), number of trades (N) and percent of the long/short positions (LONG/SHORT).

6.3 Top n-th strategy

During the selection of hyperparameters, it was implicitly assumed that the best hyperparameters for a strategy come from the strategy that performs the best in the validation period. This might not be true, as the best-performing strategy might be overfitted to said period.

Figure 17 present the modified information ratio after evaluating strategies with the top 10 hyperparameter sets from validation windows. In case of GMADL Strategy with 5 min data, the best overall performing strategy is indeed the one with the first set of hyperparameters, and the consecutive sets gradually decrease the results of the strategy. However, in case of 15min and 30min data, the second best would turn out to be better. Similar situation can be observed with the best RSI strategy, the overall best RSI strategy with 30min windows was the first one from the validation, but in case of 15min data the second best was significantly better. For RMSE Informer with 30min data the first two sets of hyperparameters seemed overfitted, with the performance greatly increasing past second set. In case of strategies that had overall poor performance, selection of top n-th set of the hyperparameters wouldn't have significant impact.

Figure 17: Strategies with top 10 hyperparameter combinations for 5min data



	1	2	3	4	5	6	7	8	9	10
Buy and Hold	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039	0.039
MACD Strategy (30 min)	0.207	1.035	0.275	0.5	0.271	0.281	0.093	0.197	0.873	0.309
MACD Strategy (15 min)	-0.118	-0.045	-0.024	-0.1	-0.008	-0.006	0.059	0.017	0.223	-0.028
MACD Strategy (5 min)	-0.087	-0.066	-0.02	-0.002	-0.013	-0.1	-0.201	-0.137	-0.001	-0.005
RSI Strategy (30 min)	2.415	2.058	0.436	0.251	0.159	0.44	0.199	0.412	0.826	1.524
RSI Strategy (15 min)	-0.014	3.996	3.524	0.002	0.839	0.97	0.568	2.489	1.888	0.093
RSI Strategy (5 min)	1.676	0.079	1.948	0.145	0.972	-0.019	0.022	0.549	0.878	2.125
RMSE Informer (30 min)	0.624	1.058	1.641	1.641	1.641	1.641	1.312	1.882	1.882	1.882
RMSE Informer (15 min)	0.14	0.022	0.022	0.022	0.022	0.022	0.022	0.316	0.316	0.316
RMSE Informer (5 min)	-0.285	-0.285	-0.285	-0.285	-0.285	-0.285	-0.285	-0.315	-0.315	-0.315
Quantile Informer (30 min)	-0.104	-0.067	-0.11	-0.313	0.001	0.197	-0.022	-0.037	-0.108	-0.246
Quantile Informer (15 min)	-0.002	-0.0	-0.155	-0.188	-0.108	0.002	-0.011	-0.099	-0.098	-0.097
Quantile Informer (5 min)	-0.001	-0.074	-0.129	-0.016	-0.237	0.296	-0.582	0.007	-0.165	-0.064
Gmadl Informer (30 min)	0.516	0.407	0.423	1.007	0.57	0.075	0.066	0.291	0.451	0.527
Gmadl Informer (15 min)	0.987	2.341	0.846	3.21	1.662	0.834	4.286	2.658	1.25	3.682
Gmadl Informer (5 min)	7.552	2.758	2.331	2.179	2.15	2.498	2.028	0.026	1.874	0.677

Note: Modified Information Ratio(IR**) of the strategies evaluated on the whole testing period, with the strategies using the top 10 hyperparameter sets according to IR** from the evaluation on the validation set.

7 Conclusion

The research aimed to explore different methods to create automated investment strategies for trading Bitcoin. Five strategies were proposed, two using technical indicators signal and three using predictions of the Informer machine learning model. The strategies were evaluated independently over six periods on BTC/USDT data of various frequencies: 5 minutes, 15 minutes, and 30 minutes. For each data window and every frequency, the strategies that use a machine learning model were first trained using the training part of the data window. The hyperparameters were then selected using the validation part of the data window. Finally, a strategy with the best set of hyperparameters according to the Modified Information Ratio was evaluated on the test part of the data window. Eight strategies beat the Buy and Hold benchmark, when compared throughout the testing period, i.e., combined results from all data windows. The best performing strategy was the one based on predictions of the Informer model trained on 5-minute data with the GMADL loss function. In addition, it significantly outperformed all the other strategies, proving to be best in almost all testing periods. A sensitivity analysis was performed to measure the impact of the selection of hyperparameters on strategy performance. It showed that the size of the validation window proved to have an impact on the selection of hyperparameters and overall strategy performance. This was also the case when modifying the number of testing periods, changing that parameter affected results and decreased the performance of the strategy, although the GMADL Informer strategy in all cases still managed to beat the benchmark.

Based on the results of the research an attempt to answer the research questions can be made:

Q: *Is it possible to create an algorithmic strategy for trading Bitcoin, that is more efficient than Buy&Hold approach?*

The research showed that it is possible to create an algorithmic strategy that outperforms the Buy&Hold benchmark on the Bitcoin data. Eight strategies achieved better results during the testing period than the benchmark. The strategies were evaluated in six different test windows, through an overall testing period of almost three years.

Q: *Does signal from Informer model allow to create strategies that are more efficient on trading Bitcoin than strategies based on technical indicators?*

While the strategies based on Informer trained with RMSE and GMADL loss functions turned out to be more efficient than the strategy based on MACD indicator, the strategy based on RSI indicator achieved comparable results. Moreover, strategy based on RSI indicator on 30 min windows was the second best overall performing strategy. This shows that even though the technical indicators have been around for a long time and are well established in the literature, they still provide a valid alternative to expensive, machine learning based approaches.

Q: *How does selection of the machine learning model loss function influence the strategy performance?*

Different behaviours of the strategies based on the model trained with different loss functions could be observed. Especially when analyzed together with looking at the different data frequencies. Predictions of the Informer trained with RMSE loss function were worse for the sudden, large returns, making it difficult to cunstruct the strategy for higher frequency data. In contrast Informer trained with GMADL loss function benefited from higher frequency. Approach utilizing a model with the Quantile loss function underperformed relative to a benchmark, showing the difficulty for constructing the strategy based on the predictions of model trained with such loss. From the three tested loss function, GMADL loss function seem to be best fitted for creating models that are to be used in automated trading strategies. What is more important, the selection of the loss function seem to be one of the deciding factors when training the model that generates the signal for algorithmic trading systems.

Q: *Does usage of higher frequency data allow to create more efficient strategies ?*

The strategies with GMADL and RMSE loss functions achieved comparable results when evaluated on 30 min data, though usage of high frequency data improved the performance of the GMADL Informer strategy, while deteriorated the performance of the strategy based on the Informer with RMSE loss function. Strategies based on technical indicators did not seem to benefit much from the higher frequencies.

The main contribution of the study was the analysis of the novel application of loss functions: Quantile and GMADL loss, to train Informer model to predict returns of Bitcoin and compare results with the better established approach of training the machine learning model with RMSE loss. Predictions of all three model types, were used to construct automated trading strategies which were compared against buy-and-hold benchmark and the two benchmark strategies based on MACD and RSI technical indicators. The exhaustive analysis included usage of different testing periods, different data intervals and sensitivity analysis. It showed that the GMADL loss function allows to effectively train the machine learning model, which is able to provide meaningful signal for the trading strategy. Such a strategy can be deployed to provide higher yields than the buy-and-hold approach.

There are various directions in which the research could be extended. A more extensive sensitivity analysis could be carried to better understand the impact of various hyperparameters on the strategies, especially in case of Informer based strategies and loss functions used for training the model. Another direction would be to test even higher data frequencies and explore whether the performance of the Informer-based model with GMADL loss function would be further improved. Finally, datasets of other financial instruments could be used, to see whether the same strategies perform well on other data than Bitcoin.

References

Adebiyi, Ayodele, Aderemi Adewumi, and Charles Ayo (Mar. 2014). "Stock price prediction using the ARIMA model". In: DOI: `10.1109/UKSim.2014.67`.

Agarap, Abien Fred (2019). *Deep Learning using Rectified Linear Units (ReLU)*. arXiv: `1803.08375 [cs.NE]`. URL: `https://arxiv.org/abs/1803.08375`.

Appel, Gerald (2005). *Technical analysis: power tools for active investors*. First. FT Press. ISBN: 0131479024.

Azari, Amin (Oct. 2018). *Bitcoin Price Prediction: An ARIMA Approach*. DOI: `10.48550/arXiv.1904.05315`.

Ba, Jimmy Lei, Jamie Ryan Kiros, and Geoffrey E. Hinton (2016). *Layer Normalization*. arXiv: `1607.06450 [stat.ML]`. URL: `https://arxiv.org/abs/1607.06450`.

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). *Neural Machine Translation by Jointly Learning to Align and Translate*. cite arxiv:1409.0473Comment: Accepted at ICLR 2015 as oral presentation. URL: `http://arxiv.org/abs/1409.0473`.

Barber, Brad and Terrance Odean (2001). "Boys will be Boys: Gender, Overconfidence, and Common Stock Investment". In: *The Quarterly Journal of Economics* 116.1, pp. 261–292. URL: `https://EconPapers.repec.org/RePEc:oup:qjecon:v:116:y:2001:i:1:p:261-292.`.

Barez, Fazl et al. (2023). "Exploring the Advantages of Transformers for High-Frequency Trading". In: *Social Science Research Network*. DOI: 10.2139/ssrn.4364833.

Basodi, Sunitha et al. (2020). "Gradient amplification: An efficient way to train deep neural networks". In: *Big Data Mining and Analytics* 3.3, pp. 196–207. DOI: 10.26599/BDMA.2020.9020004.

Benidis, Konstantinos et al. (July 2023). "Deep Learning for Time Series Forecasting: Tutorial and Literature Survey". en. In: *ACM Computing Surveys* 55.6, pp. 1–36. ISSN: 0360-0300, 1557-7341. DOI: 10.1145/3533382. URL: https://dl.acm.org/doi/10.1145/3533382 (visited on 03/29/2023).

Billah, Baki et al. (Feb. 2006). "Exponential smoothing model selection for forecasting". In: *International Journal of Forecasting* 22, pp. 239–247. DOI: 10.1016/j.ijforecast.2005.08.002.

Bollinger, J. (2002). *Bollinger on Bollinger Bands*. McGraw-Hill Education. ISBN: 9780071373685. URL: https://books.google.pl/books?id=FLlxAz85iysC.

Bondt, Werner F. M. De and Richard Thaler (1985). "Does the Stock Market Overreact?" In: *The Journal of Finance* 40.3, pp. 793–805. ISSN: 00221082, 15406261. URL: http://www.jstor.org/stable/2327804 (visited on 10/05/2024).

Box, George.E.P. and Gwilym M. Jenkins (1976). *Time Series Analysis: Forecasting and Control*. Holden-Day.

Chen, Kai et al. (2015). "A LSTM-based method for stock returns prediction: A case study of China stock market". In: *null*. DOI: 10.1109/bigdata.2015.7364089.

Cho, Kyunghyun et al. (2014). "Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation". In: *CoRR* abs/1406.1078. arXiv: 1406.1078. URL: http://arxiv.org/abs/1406.1078.

Chollet, Francois (July 2017). "Xception: Deep Learning with Depthwise Separable Convolutions". In: pp. 1800–1807. DOI: 10.1109/CVPR.2017.195.

Clevert, Djork-Arné, Thomas Unterthiner, and Sepp Hochreiter (2016). *Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)*. arXiv: 1511.07289 [cs.LG]. URL: https://arxiv.org/abs/1511.07289.

Devlin, Jacob et al. (2018). "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding". In: *CoRR* abs/1810.04805. arXiv: 1810.04805. URL: http://arxiv.org/abs/1810.04805.

Ding, Ying, Changsheng Zhang, and Chen Zhang (2023). "An Informer Based Method for Stock Intraday Price Prediction". In: *2023 19th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*, pp. 1–6. DOI: 10.1109/ICNC-FSKD59587.2023.10280785.

Duan, Chuyang and Wenjun Ke (2024). "Advanced Stock Price Prediction Using LSTM and Informer Models". In: *Journal of Artificial Intelligence General science (JAIGS) ISSN:3006-4023*. DOI: 10.60087/jaigs.v5i1.183.

Eisenach, Carson, Yagna Patel, and Dhruv Madeka (2020). "MQTransformer: Multi-Horizon Forecasts with Context Dependent and Feedback-Aware Attention". In: *CoRR* abs/2009.14799. arXiv: 2009.14799. URL: https://arxiv.org/abs/2009.14799.

Fama, Eugene (1970). "Efficient Capital Markets: A Review of Theory and Empirical Work". In: *Journal of Finance* 25, pp. 383–417.

— (Dec. 1991). "Efficient Capital Markets: II". In: *Journal of Finance* 46.5, pp. 1575–1617. URL: https://ideas.repec.org/a/bla/jfinan/v46y1991i5p1575-617.html.

Fischer, Thomas et al. (2017). "Deep learning with long short-term memory networks for financial market predictions". In: *European Journal of Operational Research*. DOI: 10.1016/j.ejor.2017.11.054.

Fischhoff, Baruch and Paul Slovic (June 1978). "A Little Learning...: Confidence in Multicue Judgment Tasks". In: p. 58.

Gervais, Simon and Terrance Odean (Feb. 2001). "Learning To Be Overconfident". In: *Review of Financial Studies* 14, pp. 1–27. DOI: 10.2139/ssrn.36313.

Grudniewicz, Jan and Robert Ślepaczuk (2023). "Application of machine learning in algorithmic investment strategies on global stock markets". In: *Research in International Business and Finance* 66, p. 102052. ISSN: 0275-5319. DOI: https://doi.org/10.1016/j.ribaf.2023.102052. URL: https://www.sciencedirect.com/science/article/pii/S0275531923001782.

Hájek, Petr and Josef Novotny (2024). "Beyond Sentiment in Stock Price Prediction: Integrating News Sentiment and Investor Attention with Temporal Fusion Transformer". In: *Artificial Intelligence Applications and Innovations*. DOI: 10.1007/978-3-031-63219-8_3.

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1997). "Long Short-Term Memory". In: *Neural Comput.* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

Honchar, Oleksandr and Luca Di Persio (Jan. 2016). "Artificial neural networks approach to the forecast of stock market price movements". In: Vol.1, pp. 158–162.

Hossain, Mohammad et al. (Nov. 2018). "Hybrid Deep Learning Model for Stock Price Prediction". In: pp. 1837–1844. DOI: 10.1109/SSCI.2018.8628641.

Hu, Xiaokang and Xiaokang Hu (2021). "Stock Price Prediction Based on Temporal Fusion Transformer". In: *2021 3rd International Conference on Machine Learning, Big Data and Business Intelligence (MLBDBI)*. DOI: 10.1109/mlbdbi54094.2021.00019.

Huberman, Gur and Tomer Regev (2001). "Contagious Speculation and a Cure for Cancer: A Nonevent that Made Stock Prices Soar". In: *Journal of Finance* 56.1, pp. 387–396. URL: https://EconPapers.repec.org/RePEc:bla:jfinan:v:56:y:2001:i:1:p:387-396.

Kahneman, Daniel and Amos Tversky (Mar. 1979). "Prospect Theory: An Analysis of Decision under Risk". In: *Econometrica* 47.2, pp. 263–291. URL: https://ideas.repec.org/a/ecm/emetrp/v47y1979i2p263-91.html.

Khashei, Mehdi and Mehdi Bijari (2011). "A novel hybridization of artificial neural networks and ARIMA models for time series forecasting". In: *Applied Soft Computing* 11.2. The Impact of Soft Computing for the Progress of Artificial Intelligence, pp. 2664–2675. ISSN: 1568-4946. DOI: https://doi.org/10.1016/j.asoc.2010.10.015. URL: https://www.sciencedirect.com/science/article/pii/S1568494610002759.

Kijewski, Mateusz, Robert Slepaczuk, and Maciej Wysocki (Sept. 2024). "Predicting Prices Of S&P 500 Index Using Classical Methods and Recurrent Neural Networks". In: *Proceedings of the 32nd International Conference on Information Systems Development (ISD 2024)*. DOI: 10.62036/ISD.2024.89.

Kim, Jae H., Abul Shamsuddin, and Kian-Ping Lim (2011). "Stock return predictability and the adaptive markets hypothesis: Evidence from century-long U.S. data". In: *Journal of Empirical Finance* 18.5, pp. 868–879. DOI: 10.1016/j.jempfin.2011.08. URL: https://ideas.repec.org/a/eee/empfin/v18y2011i5p868-879.html.

Kiranyaz, Serkan et al. (2019). *1D Convolutional Neural Networks and Applications: A Survey.* arXiv: 1905.03554 [eess.SP]. URL: https://arxiv.org/abs/1905.03554.

Kolen, John F. and Stefan C. Kremer (2001). "Gradient Flow in Recurrent Nets: The Difficulty of Learning LongTerm Dependencies". In: *A Field Guide to Dynamical Recurrent Networks*, pp. 237–243. DOI: 10.1109/9780470544037.ch14.

Kosc, Krzysztof, Paweł Sakowski, and Robert Ślepaczuk (2019). "Momentum and contrarian effects on the cryptocurrency market". In: *Physica A: Statistical Mechanics and its Applications* 523, pp. 691–701. ISSN: 0378-4371. DOI: https://doi.org/10.1016/j.physa.2019.02.057. URL: https://www.sciencedirect.com/science/article/pii/S037843711930216X.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (May 2015). "Deep learning". en. In: *Nature* 521.7553, pp. 436–444. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14539. URL: https://www.nature.com/articles/nature14539 (visited on 10/06/2024).

Lim, Bryan, Sercan O. Arik, et al. (2020). *Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting.* arXiv: 1912.09363 [stat.ML]. URL: https://arxiv.org/abs/1912.09363.

Lim, Kian-Ping and Robert Brooks (Feb. 2011). "The Evolution Of Stock Market Efficiency Over Time: A Survey Of The Empirical Literature". In: *Journal of Economic Surveys* 25.1, pp. 69–108. URL: https://ideas.repec.org/a/bla/jecsur/v25y2011i1p69-108.html.

Lu, Yuze, Hailong Zhang, and Qiwen Guo (2023). *Stock and market index prediction using Informer network.* arXiv: 2305.14382 [q-fin.ST]. URL: https://arxiv.org/abs/2305.14382.

Luong, Thang, Hieu Pham, and Christopher D. Manning (Sept. 2015). "Effective Approaches to Attention-based Neural Machine Translation". In: *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing.* Ed. by Lluís Màrquez, Chris Callison-Burch, and Jian Su. Lisbon, Portugal: Association for Computational Linguistics, pp. 1412–1421. DOI: 10.18653/v1/D15-1166. URL: https://aclanthology.org/D15-1166.

Malkiel, Burton (Feb. 2005). "Reflections on the Efficient Market Hypothesis: 30 Years Later". In: *The Financial Review* 40, pp. 1–9. DOI: 10.1111/j.0732-8516.2005.00090.x.

Malkiel, Burton. G. (1973). *A Random Walk Down Wall Street.* Norton, New York.

Michańków, Jakub, Paweł Sakowski, and Robert Ślepaczuk (2022). "LSTM in Algorithmic Investment Strategies on BTC and S&P500 Index". In: *Sensors* 22.3. ISSN: 1424-8220. DOI: 10.3390/s22030917. URL: https://www.mdpi.com/1424-8220/22/3/917.

— (2024). "Mean Absolute Directional Loss as a new loss function for machine learning problems in algorithmic investment strategies". In: *Journal of Computational Science* 81, p. 102375. ISSN: 1877-

7503. DOI: https://doi.org/10.1016/j.jocs.2024.102375. URL: https://www.sciencedirect.com/science/article/pii/S1877750324001686.

Nagi, Jawad et al. (2011). "Max-pooling convolutional neural networks for vision-based hand gesture recognition". In: *2011 IEEE International Conference on Signal and Image Processing Applications (ICSIPA)*, pp. 342–347. DOI: 10.1109/ICSIPA.2011.6144164.

Nakamoto, Satoshi (Mar. 2009). "Bitcoin: A Peer-to-Peer Electronic Cash System". In: *Cryptography Mailing list at https://metzdowd.com*.

Nelson, David M. et al. (2017). "Stock market's price movement prediction with LSTM neural networks". In: DOI: 10.1109/ijcnn.2017.7966019.

Odean., Terrance (Nov. 1996). *Are Investors Reluctant to Realize Their Losses?* Research Program in Finance Working Papers RPF-269. University of California at Berkeley. URL: https://ideas.repec.org/p/ucb/calbrf/rpf-269.html.

OpenAI et al. (2024). *GPT-4 Technical Report*. arXiv: 2303.08774 [cs.CL]. URL: https://arxiv.org/abs/2303.08774.

Pai, Ping-Feng and Chih-Sheng Lin (2005). "A hybrid ARIMA and support vector machines model in stock price forecasting". In: *Omega* 33.6, pp. 497–505. ISSN: 0305-0483. DOI: https://doi.org/10.1016/j.omega.2004.07.024. URL: https://www.sciencedirect.com/science/article/pii/S0305048304001082.

Penmetsa, Siddharth and Maruthi Vemula (2023). "Cryptocurrency Price Prediction with LSTM and Transformer Models Leveraging Momentum and Volatility Technical Indicators". In: *2023 IEEE 3rd International Conference on Data Science and Computer Application (ICDSCA)*. DOI: 10.1109/icdsca59871.2023.10393319.

Persio, Luca Di et al. (2017). "Recurrent Neural Networks Approach to the Financial Forecast of Google Assets". In: *null*. DOI: null.

Salinas, David, Valentin Flunkert, and Jan Gasthaus (2019). *DeepAR: Probabilistic Forecasting with Autoregressive Recurrent Networks*. arXiv: 1704.04110 [cs.AI]. URL: https://arxiv.org/abs/1704.04110.

Selvin, Sreelekshmy et al. (2017). "Stock price prediction using LSTM, RNN and CNN-sliding window model". In: *null*. DOI: 10.1109/icacci.2017.8126078.

Sharpe, William F. (1998). "The Sharpe Ratio (Fall 1994)". In: *The Best of The Journal of Portfolio Management*. Ed. by Peter L. Bernstein and Frank J. Fabozzi. Princeton: Princeton University Press, pp. 169–178. ISBN: 9781400829408. DOI: doi:10.1515/9781400829408-022. URL: https://doi.org/10.1515/9781400829408-022.

Siami-Namini, Sima et al. (2018). "Forecasting Economics and Financial Time Series: ARIMA vs. LSTM." In: *arXiv: Learning*. DOI: null.

Team, Gemini et al. (2024). *Gemini: A Family of Highly Capable Multimodal Models*. arXiv: 2312.11805 [cs.CL]. URL: https://arxiv.org/abs/2312.11805.

Tversky, Amos and Daniel Kahneman (1981). "The Framing of Decisions and the Psychology of Choice". In: *Science* 211.4481, pp. 453–458. DOI: 10.1126/science.7455683. eprint: https://www.science.org/doi/pdf/10.1126/science.7455683. URL: https://www.science.org/doi/abs/10.1126/science.7455683.

Vaserstein, L.N. (1969). "Markov processes over denumerable products of spaces, describing large systems of automata". In: *Probl. Peredachi Inf.* 5.

Vaswani, Ashish et al. (2017). "Attention Is All You Need". In: *CoRR* abs/1706.03762. arXiv: 1706.03762. URL: http://arxiv.org/abs/1706.03762.

Võ, Thi Ái Nguyên and Robert Ślepaczuk (Jan. 2022). "Applying Hybrid ARIMA-SGARCH in Algorithmic Investment Strategies on S&P500 Index". In: *Entropy* 24, p. 158. DOI: 10.3390/e24020158.

Wang, Ju-Jie et al. (2012). "Stock index forecasting based on a hybrid model". In: *Omega* 40.6. Special Issue on Forecasting in Management Science, pp. 758–766. ISSN: 0305-0483. DOI: https://doi.org/10.1016/j.omega.2011.07.008. URL: https://www.sciencedirect.com/science/article/pii/S0305048311001435.

Wang, S. (2023). "A Stock Price Prediction Method Based on BiLSTM and Improved Transformer". In: *IEEE Access*. DOI: 10.1109/access.2023.3296308.

Wilder, J.W. (1978). *New Concepts in Technical Trading Systems*. Trend Research. ISBN: 9780894590276. URL: https://books.google.pl/books?id=WesJAQAAMAAJ.

Yadav, Y. (Nov. 2015). "How algorithmic trading undermines efficiency in capital markets". In: 68, pp. 1607–1671.

Zhang, G.Peter (2003). “Time series forecasting using a hybrid ARIMA and neural network model”. In: *Neurocomputing* 50, pp. 159–175. ISSN: 0925-2312. DOI: https://doi.org/10.1016/S0925-2312(01)00702-0. URL: https://www.sciencedirect.com/science/article/pii/S0925231201007020.

Zhao, Huali, Martin Crane, and Marija Bezbradica (2022). “Attention! Transformer with Sentiment on Cryptocurrencies Price Prediction”. In: *International Conference on Complex Information Systems*. DOI: 10.5220/0011103400003197.

Zhou, Haoyi et al. (2021). *Informer: Beyond Efficient Transformer for Long Sequence Time-Series Forecasting*. arXiv: 2012.07436 [cs.LG]. URL: https://arxiv.org/abs/2012.07436.

University of Warsaw
Faculty of Economic Sciences
44/50 Długa St.
00-241 Warsaw
www.wne.uw.edu.pl
ISSN 2957-0506