UNIVERSITY OF WARSAW

Faculty of Economic Sciences

Working Papers

No. 21/2022 (397)

A COMPARISON OF LSTM AND GRU ARCHITECTURES WITH NOVEL WALK-FORWARD APPROACH TO ALGORITHMIC INVESTMENT STRATEGY

Illia Baranochnikov

Robert Ślepaczuk

Warsaw 2022

UNIVERSITY OF WARSAW
FACULTY OF ECONOMIC SCIENCES
WORKING PAPERS

A comparison of LSTM and GRU architectures with novel walk-forward approach to algorithmic investment strategy

Illia Baranochnikov[a], Robert Ślepaczuk[b]

[a] *University of Warsaw, Faculty of Economic Sciences, Quantitative Finance Research Group;*
[b] *University of Warsaw, Faculty of Economic Sciences, Quantitative Finance Research Group, Department of Quantitative Finance*

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

Abstract: The aim of this work is to build a profitable algorithmic investment strategy on various types of assets. The algorithm is built using recurrent neural networks (LSTM and GRU) as the primary source of signals to buy/sell financial instruments. LSTM and GRU architectures are compared in terms of obtaining the best results and beating the market. The algorithm is tested for four financial instruments (Bitcoin, Tesla, Brent Oil and Gold) on daily and hourly data frequencies. The out-of-sample period is from 1 January 2021 to 1 April 2022. A walk-forward process is responsible for training models and selecting the best model to forecast asset prices in the future. Ten model architectures with various hyperparameters are trained during each step of the walk-forward process. The model architecture with the highest Information Ratio (IR*) in the validation period is used for forecasting in the out-of-sample period. For each strategy, the performance metrics are calculated based on which the profitability of the algorithm is evaluated. At the end, a detailed sensitivity analysis with regards to the main hyperparameters is conducted. The results reveal that LSTM outperforms GRU in most of the cases and that investment strategy built based on LSTM/GRU architecture is able to beat the market only on 50% of tested cases.

Keywords: deep learning, recurrent neural networks, algorithm, trading strategy, LSTM, GRU, walk-forward process

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

This research aims to build a profitable algorithmic investment strategy (AIS) and explain what type of Recurrent Neural Network (RNN) is more efficient in generating buy/sell signals for this purpose. The issue is important because accurate modelling of the financial markets can offer many opportunities to increase the efficiency of asset management. Efficient asset management allows, first of all, to correctly evaluate the risk arising from possessing specific assets. It can help investors choose financial instruments in line with their risk expectations and avoid the emergence of other financial crises during which people lose their savings.

Two types of RNN are considered in this study: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) models. The first hypothesis (RH1) states that *the LSTM model outperforms the GRU model in most cases (in more than 50% of cases)*. The second hypothesis (RH2) states that *it is possible to build an investment algorithm that will obtain a higher risk-adjusted rate of return than the benchmark for every tested asset, which contradicts the weak form of the efficient-market hypothesis in the information sense described by Fama (1970)*. The "Buy & Hold" strategy on the selected type of asset is perceived as a benchmark. Additionally, the following research questions are formulated:

RQ1. Is the investment strategy robust to changes in the financial instrument it predicts?
RQ2. Is the investment strategy robust to changes in the data frequency?
RQ3. Is the investment strategy robust to changes in model hyperparameters?
RQ4. Is the ensemble AIS able to obtain a higher risk-adjusted rate of return than the benchmark?

In order to verify the main hypotheses and answer the research questions posed, an empirical study is conducted in which an AIS is created. The source of buy/sell signals is in indications of LSTM or GRU models. The AIS is designated to select from a range of 10 different models with different hyperparameters. The hyperparameters of these models are selected based on the literature discussed in Section 2. The core element of the algorithm is the walk-forward process, which is responsible for training the models and selecting the best model based on the calculated Information Ratio performance metrics. The walk forward process trains these models on different time-series datasets (multiple training, validation and testing periods).

The algorithmic investment strategies are tested for four different assets: Bitcoin coin, Tesla stock, Brent Oil Futures and Gold Futures. The main idea behind the selection of these financial instruments is to check the behaviour of the investment strategy on assets with different volatility. The investment algorithm uses time series with a daily and hourly frequency. The dataset covers data in the time range from 27 November 2019 to 1 April 2022. The out-of-sample period starts on 1 January 2021 and ends on 1 April 2022.

LSTM is expected to be more profitable than GRU due to its more complicated architecture. Also, the algorithm probably will not be able to beat the market because the financial instruments selected for this study recorded a significant increase in value during this period. In addition, it is not expected that the investment strategy will be robust to the changes discussed in research questions RQ1-RQ3.

The paper has the following structure: Section 2 presents a description of the literature and the papers on which the work is based. The following section presents all the necessary information on the data and financial instruments used for this study. Section 4 demonstrates how the AIS is implemented. Additionally, it explains what performance metrics are considered during the testing process. In Section 5, the empirical results are presented for individual financial instruments and the strategy's profitability is compared to the benchmark - the "Buy & Hold" strategy. In Section 6, a sensitivity analysis is conducted, inspecting how strong the impact of changing individual parameters of the investment algorithm is. Section 7 demonstrates an ensemble AIS, comparing its profitability to the benchmark one. The last section concludes the study.

2. Literature review

For a long time, researchers worldwide have been looking for a way to model financial markets to build algorithmic investment strategies. In almost every case, behind the creation of a trading algorithm is the desire to make profits through the use of less or more complex financial theories and models. Each such efficient strategy contradicts the efficient-market hypothesis. In an efficient market, the prices of financial instruments already reflect all available information and the expected profit of the speculative investment strategy above the market would be equal to zero. Fama (1970), in his study, explore the efficient-market hypothesis from the empirical point of view, dividing it into three forms: weak, semi-strong and strong. In the weak form of the efficient-market hypothesis, it is assumed that the prices of assets fully reflect all the information connected with historical asset prices. The semi-strong form includes all publicly available information, such as news and financial reports. Finally, the strong form of the efficient-market hypothesis holds that asset prices take into account all information, including information that is not available to the public. Most of the papers discussed below build investment algorithms using only historical stock prices, thus contradicting the weak form of the efficient-market hypothesis.

One of the main tools for a long time for modelling financial time series has been models such as ARIMA, which Box and Jenkins introduced in 1976. Nevertheless, technology has improved with time, and it has become possible to use methods requiring a lot of computing power. Such methods include, among others, neural networks, training of which can take up to several days. Adebiyi et al. (2014) present a study where they compare the effectiveness of the ARIMA and Artificial Neural Network (ANN) models in forecasting the daily closing prices of NYSE stocks. The performance of the ANN model turns out to be higher than the corresponding statistic for the ARIMA model.

At some point in time-series analysis, Recurrent Neural Networks (RNN), previously mainly used for Natural Language Processing, have begun to be used more often. Rumelhart et al. (1986) are the first to introduce the concept of the RNN model. A fundamental problem with RNN is that the model cannot capture long-term relationships due to the vanishing (or exploding) gradient problem discussed by Hochreiter (1991). Hochreiter and Schmidhuber (1997) in their paper written six years later present Long Short-Term Memory (LSTM) model. LSTM neural network is an improved version of the Recurrent Neural Network. The main advantage over traditional RNNs is the lack of a vanishing gradient problem. Roondiwala et al. (2015) use the LSTM model to forecast the Indian Stock Market NIFTY 50 Index level. They get the lowest Root Mean Square Error (RMSE) for the LSTM model, which has 500 epochs and takes into account four features: high, low, open and close prices. Siami-Namini et al. (2018) employ LSTM neural networks to forecast monthly closing prices for 11 stock market indices. Comparing the efficiency with the ARIMA model, they conclude that LSTM reduces the RMSE by 85% more than the ARIMA model. Lim and Lundgren (2019) choose ten stocks from the S&P500 index. Next, they make an investment strategy for the time series with a frequency of 5 minutes. The authors note that the LSTM performs the worst on the Mean Square Error statistics. However, the critical fact is that the LSTM turns out to be better in terms of the Sharpe Ratio metrics than the benchmark "Buy & Hold" strategy and the VARMAX model. Kijewski and Ślepaczuk (2020) conduct research comparing the profitability of investment strategies based on the ARIMA, LSTM and other classical methods. LSTM has higher risk-adjusted returns than the ARIMA model and the "Buy & Hold" benchmark. The authors stress an essential warning that these models are not robust to the selected hyperparameters.

Another type of recurrent neural network is called the Gated Recurrent Unit (GRU), a modified version of the LSTM model. The architecture of the GRU model was firstly presented in the work of Cho et al. (2014). In the study, the authors use the GRU model to build a Neural Machine Translation system, not even mentioning the possibility of using GRU to model financial time series. The main advantage of GRU is the simplified architecture which allows GRU to reduce the time needed to train the model compared to the LSTM. Site et al. (2019) compare the accuracy of all the three types of recurrent neural networks (RNN, GRU and LSTM) to other regression methods: Support Vector Regression, Linear Regression and Ridge Regression. The study is conducted based on the historical prices of the shares of Google and Amazon. The GRU and LSTM models achieve the lowest values for the MSE statistics. Sethia and Raut (2019) build investment algorithms that invest in the S&P500 index based on signals from the following models: Artificial Neural Network, Support Vector Machines, Gated Recurrent Unit and Long Short-Term Memory. The recurrent neural networks (GRU and LSTM) achieve the best results.

Another crucial aspect in building algorithmic investment strategies is the feature selection process, i.e. selecting appropriate time series based on which algorithm makes decisions. Krauss et al. (2017) and Fischer and Krauss (2018) make investment algorithms based on stock returns as the only feature. They use random forests and the LSTM model, obtaining the highest rate of return for the latter one. Then the exact configuration of models is repeated by Ghosh et al. (2021), using three different features, which increase the average rate of return of the algorithm strategy from 0.41% daily return to 0.64% for the LSTM model. Du et al. (2019) implement two LSTM models to forecast Apple stock prices. The first model uses closing prices as the only feature, while the second one uses a more extensive range of features: the lowest price, the highest price, the opening price and other stock data. The results show that increasing features can reduce the Mean Absolute Error (MAE) statistics four times. Bahadur Shahi et al. (2020) use the LSTM and GRU networks to forecast stock prices. They make two versions: the first contains only stock attributes, and the second additionally includes the news sentiment score. Model taking into account news sentiment achieves lower MAE and higher Directional Accuracy metrics.

While recurrent neural networks perform well on their own compared to classical methods, researchers are constantly looking for a way to improve results. Girsang et al. (2020) propose a hybrid model that combines the LSTM model with an algorithm that optimizes the model training process. The hybrid model achieves better results in RMSE, MAE, Mean Absolute Percentage Error (MAPE) and R^2 statistics than the regular LSTM or the ARIMA model. Zou and Qu (2020) implement the LSTM model that incorporates the Attention Mechanism. The authors test the investment algorithm on ten different stocks, and for each stock, the hybrid algorithm achieves a lower MSE statistic than in the case of the simple LSTM model.

3. Data

In our study, the investment algorithm is tested on various types of financial instruments that differ, among others, in terms of volatility and liquidity. The following categories of financial instruments were chosen: cryptocurrencies, stocks, energy commodities and metals commodities. For each type, one example of such an instrument was chosen. So, the algorithm was tested for the subsequent assets: Bitcoin coin, Tesla stock, Brent Oil Futures and Gold Futures. Figure 1 presents the time series for individual financial instruments.

Bitcoin's closing prices were taken from the Binance cryptocurrency exchange. Data for Tesla stock was downloaded from the alphavantage.co webpage. Historical prices of the

Brent Oil Futures and the Gold Futures were taken from dukascopy.com website.

Time series with historical prices of the given assets were collected at a 15-minute frequency. Then, daily and hourly time series are made based on these time series. The gaps for daily data were filled in by downloading daily data from the Yahoo Finance platform. Then the time series with closing prices of assets are converted into time series of simple rates of return, which are used to build and test the investment algorithm. The formula based on which the rates of return are calculated is presented below:

$$R_t = \frac{S_t}{S_{t-1}} - 1 \tag{1}$$

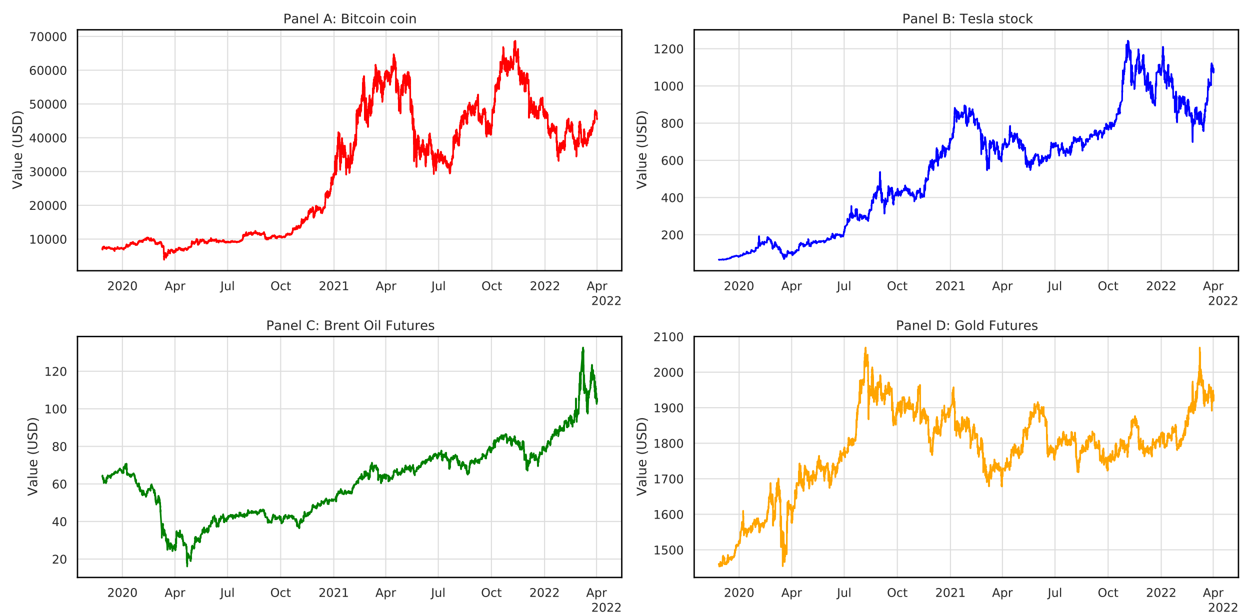
where:

R_t - rate of return in period t

S_t, S_{t-1} - financial instrument prices in periods t and t-1, respectively

In this paper, to forecast the price of a financial instrument for the period t + 1, the historical values of the returns of this instrument are used as the only feature. Also, any variable normalization is not to applied to this feature.

Figure 1: Bitcoin coin, Tesla stock, Brent Oil Futures, Gold Futures prices



Note: The chart shows the prices of the financial instruments over the time frame from 27 November 2019 to 1 April 2022.

Table 1: The descriptive statistics of the daily and hourly returns for every financial instrument

Financial instrument	Frequency	Mean	SD	Min	1st quartile	3st quartile	Max	Jarque-Bera	p-value(JB)
Bitcoin coin	daily	0.30%	3.96%	-24.20%	-1.57%	2.21%	19.22%	788.1	0.0
	hourly	0.01%	0.87%	-15.94%	-0.31%	0.34%	30.03%	7499110.0	0.0
Tesla stock	daily	0.59%	4.83%	-18.81%	-1.63%	2.68%	24.25%	309.7	0.0
	hourly	0.04%	1.18%	-11.33%	-0.35%	0.40%	15.85%	155791.0	0.0
Brent Oil Futures	daily	0.14%	3.32%	-29.60%	-1.23%	1.44%	19.67%	9613.4	0.0
	hourly	0.01%	0.76%	-22.07%	-0.23%	0.26%	13.20%	4380220.6	0.0
Gold Futures	daily	0.05%	1.03%	-4.77%	-0.41%	0.59%	4.58%	335.2	0.0
	hourly	0.00%	0.22%	-2.32%	-0.08%	0.09%	2.70%	121193.7	0.0

Note: Descriptive statistics calculated for all the financial instruments on simple returns for the period from 27 November 2019 to 1 April 2022.

Table 1 shows the descriptive statistics calculated for all financial instruments for both frequencies. It is worth noting that for each time series except one (Gold Futures with hourly frequency), the mean of the rates of return is positive, which demonstrates that these assets had an upward trend over the selected period. Additionally, the Jarque-Bera test is performed. For each time series, the Jarque-Bera statistic obtains large values and a zero p-value, i.e. these time series do not have a normal distribution.

4. Methodology

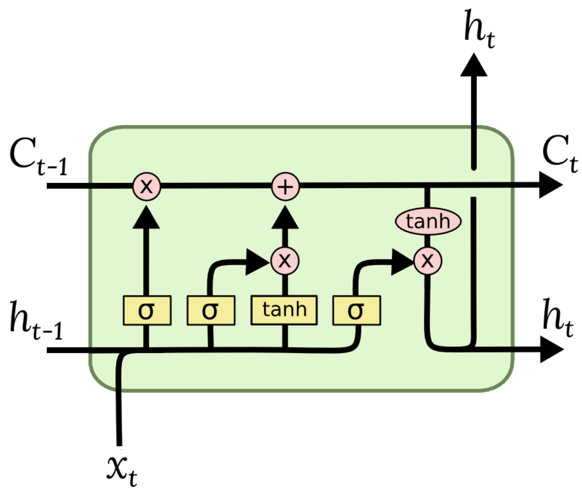
4.1. Recurrent Neural Networks

Our investment strategy uses Recurrent Neural Networks (RNN) as the primary source of buy/sell signals. RNNs are more effective in time series modelling than traditional feed-forward neural networks. Time series often have dependencies between successive inputs that the RNN can capture due to its structure. Two types of neural networks are used: Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU). Both model architectures are based on the original RNN, but they do not have the vanishing gradient problem.

4.1.1. LSTM

Long Short-Term Memory (LSTM) is based on the idea of gates. The main element of LSTM is the cell state (C_t) that can "remember" long-term dependencies, and the gates decide how the information flow. Figure 2 presents the architecture of LSTM, where the structure of the gates can be examined. The model includes three gates: input, output, and forget.

Figure 2: The Long Short-Term Memory architecture



Note: The architecture of LSTM, source: https://colah.github.io/posts/2015-08-Understanding-LSTMs

The forget gate decides which part of the cell state from the previous period (C_{t-1}) will be forwarded. This gate receives a combined vector consisting of the hidden state from the previous period (h_{t-1}) and the input from the current period (x_t). Forget gate returns a value from 0 to 1, where 0 means complete "forgetting" and 1 means all the information from the previous cell state is transferred.

The input gate decides how much newly received information (combined vector of h_{t-1} and x_t) is added to the current cell state. Additionally, this gate applies appropriate transformations to the newly received information.

Finally, the output flow is controlled by the output gate. It first decides what parts of the cell state are output and then transforms them accordingly using the tanh function.

$$f_t = \sigma(W_f * [h_{t-1}, x_t] + b_f) \tag{2}$$

$$i_t = \sigma(W_i * [h_{t-1}, x_t] + b_i) \tag{3}$$

$$\widetilde{C}_t = tanh(W_C * [h_{t-1}, x_t] + b_C) \tag{4}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{5}$$

$$o_t = \sigma(W_o * [h_{t-1}, x_t] + b_o) \tag{6}$$

$$h_t = o_t * tanh(C_t) \tag{7}$$

where:

$f_t, i_t, o_t, \widetilde{C}_t$ - activation vectors

W_f, W_i, W_C, W_o - weight matrices

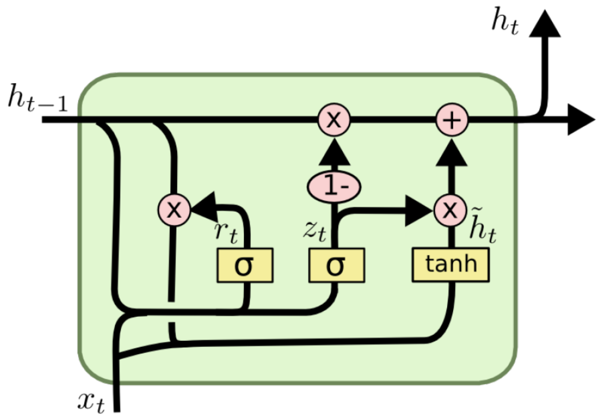
b_f, b_i, b_C, b_o - biases

Formulas 2-7 describe the architecture of the LSTM model, while formulas 8-11 describe the GRU model. The information is taken from the https://colah.github.io/posts/2015-08-Understanding-LSTMs website.

4.1.2. GRU

The main difference between LSTM and GRU is that GRU does not contain a cell state and uses a hidden state to transfer information instead. In addition, the structure of gates in GRU is changed and has only two gates: a reset gate and an update gate. This change in structure allows the GRU to use less computing power than LSTM and is faster to train. Figure 3 shows the structure of the GRU.

Figure 3: The Gated Recurrent Unit architecture



Note: The architecture of GRU, source: https://colah.github.io/posts/2015-08-Understanding-LSTMs

The reset gate determines how much information from the previous hidden state is forwarded. This gate takes the hidden state from the previous time step and the input from the current time step. It returns a value between 0 and 1, where 1 is the complete historical information, and 0 means historical information is entirely forgotten.

The update gate decides the proportion of the old and new information transferred. This gate returns a value between 0 and 1, where 0 means that only the information from the previous hidden state is forwarded. In contrast, 1 means that only the new information is forwarded.

$$r_t = \sigma(W_r * [h_{t-1}, x_t] + b_r) \tag{8}$$

$$z_t = \sigma(W_z * [h_{t-1}, x_t] + b_z) \tag{9}$$

$$\widetilde{h_t} = tanh(W_h * [r_t * h_{t-1}, x_t] + b_h) \tag{10}$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \widetilde{h_t} \tag{11}$$

where:

$r_t, z_t, \widetilde{h_t}$ - activation vectors

W_r, W_z, W_h - weight matrices

b_r, b_z, b_h - biases

4.2. Model architectures

Based on the literature presented in Section 2, 10 neural network architectures are identified. They are used as the primary source of the investment strategy signals. Table 2 shows the hyperparameters for each architecture. An important note is that these models are not exactly like these from the literature. A few parameters are chosen that are taken from the literature, and the rest is set in the process of this research. If the selected paper does not specify the value of the given hyperparameter, it is also defined by ourselves. Based on the literature, the following hyperparameters are chosen:

- the number of RNN layers
- the number of neurons
- the dropout rate
- the batch size
- the amount of epochs
- the learning rate

The maximum number of epochs is limited to 100. For each model, the sequence parameter is at the level of 20, which means that the model will forecast the rate of return in the period T+1 using the previous 20 historical rates of return. Hyperparameters of the neural networks are the same for LSTM and GRU. Models are trained using the Mean Square Error loss function and Adam optimizer with the AMSGrad extension. All other hyperparameters not listed here take default values specified in the Python Keras library.

Table 2: The model architectures used in the algorithmic investment strategy

	Model #1	Model #2	Model #3	Model #4
1st layer neurons	64	30	4	25
2nd layer neurons	128	0	10	0
Dropout rate	0.3	0.2	0.1	0.1
Batch size	[64]	[64]	[64]	[64]
Epochs	[100]	100	[100]	30
Learning rate	[0.001]	0.01	0.0001	0.001
Source	Sethia and Raut (2019)	Kijewski and Ślepaczuk (2020)	Benjamin Lim and Lundgren (2019)	Ghosh et al.(2021)

	Model #5	Model #6	Model #7	Model #8
1st layer neurons	[32]	12	128	32
2nd layer neurons	0	12	64	16
Dropout rate	0.2	[0.2]	0	0.2
Batch size	30	30	[64]	[64]
Epochs	100	[100]	[100]	[100]
Learning rate	[0.001]	0.03	[0.001]	0.002
Source	Zou and Qu (2020)	Du et al. (2019)	Roondiwala et al. (2015)	Site et al. (2019)

	Model #9	Model #10
1st layer neurons	120	50
2nd layer neurons	0	0
Dropout rate	0.2	0.25
Batch size	30	32
Epochs	100	100
Learning rate	[0.001]	[0.001]
Source	Shahi et al. (2020)	Girsang et al. (2020)

Note: If the model has 0 neurons in the second layer, the above model only contains one recurrent layer. Square brackets indicate hyperparameters that are changed or set in the process of the research. Every model is trained using the Mean Square Error loss function and Adam optimizer with the AMSGrad extension. Also, every model takes the sequence equal to 20 previous historical rates of return as an input.

4.3. Performance metrics

In order to assess the profitability and effectiveness of our investment strategy, several metrics are calculated for every strategy and asset. Then, based on the calculated statistics, it is possible to decide whether it is worth employing a specific investment algorithm. Performance metrics presented in the study of Ryś and Ślepaczuk (2018) are used.

4.3.1. ARC - Annualized Return Compounded.

ARC is a return metric that calculates the compounded interest rate of return of the AIS per annum, considering the annual number of observations for a particular financial instrument.

$$ARC = (\prod_{t=1}^{N}(1+R_t))^{\frac{observations.year}{N}} - 1 \tag{12}$$

where:

observations.year - the number of observations during the year for a given financial instrument (365 for Bitcoin and 252 for other instruments under investigation)

N - the number of observations over the entire period under study

R_t - the simple rate of return in period t

4.3.2. ASD - Annualized Standard Deviation

ASD is a risk metric that calculates the standard deviation of the AIS per annum, considering the annual number of observations for a particular financial instrument.

$$ASD = \sqrt{observations.year} * \sqrt{\frac{\sum_{t=1}^{N}(R_t - \bar{R})^2}{N-1}} \tag{13}$$

where:

$\bar{R}$ - the average rate of return over the entire period under study

4.3.3. IR - Information Ratio*

IR* is a risk-adjusted return metric that reflects the annualized risk-adjusted rate of return of the AIS by dividing ARC by ASD.

This performance metric differs from the traditional Information Ratio. Here it is assumed that the benchmark rate of return is 0, so the formula is as follows:

$$IR^* = \frac{ARC}{ASD} \tag{14}$$

4.3.4. MD - Maximum Drawdown

MD is a risk metric that measures the highest percentage loss of the AIS relative to the highest historical capital level.

$$MD = \max_{a<b} \frac{Equity.Line(a) - Equity.Line(b)}{Equity.Line(a)} \tag{15}$$

where:

$Equity.Line(a), Equity.Line(b)$ - the capital level in the period a and b, respectively

*4.3.5. IR** - Adjusted Information Ratio*

IR** is a modified IR* metric, which is the annualized risk-adjusted rate of return for AIS that considers not only ASD but also MD.

$$IR^{**} = \frac{ARC^2 * sign.ARC}{ASD * MD} \tag{16}$$

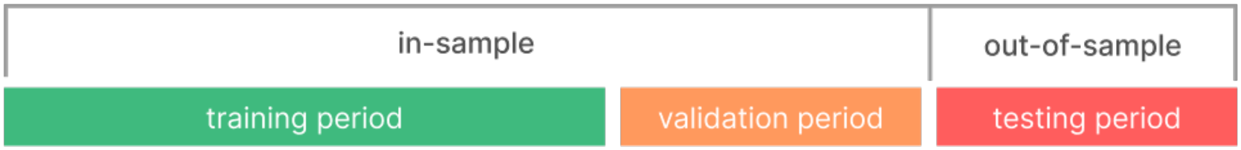
where:

$sign.ARC$ - equals +1 if $ARC \geq 0$; -1 if $ARC < 0$

4.4. Walk-forward process for AIS testing

The investment algorithm uses a walk-forward process, which allows us to test the investment strategy on a more extended out-of-sample period. A primary element in this process is a walk-forward process unit, presented in Figure 4. It has three different periods. The first is the training period; observations from this period are used to train recurrent neural networks. During this period, all the models presented in Section 4.2 are trained. The next period is the validation period. All trained models are tested for profitability during this period and calculate an IR* statistic for each model. The model with the highest statistic in the validation period is selected and used in the testing period. The last period is the testing period. The best algorithm from the validation period is used to generate buy/sell signals on tested financial instruments in the testing period.

Figure 4: A walk-forward process unit



Note: The presentation of three periods of one walk-forward process unit. These periods may vary in duration; the diagram is for visualization purposes only. The exact values for the duration of these periods are presented in Section 5.

The training and validation periods are in-sample periods, which means that AIS (in T_0) has access to these observations and can use them for analytical purposes. On the other hand, the testing period is the out-of-sample period, which means AIS does not know these observations yet because they are from the future.

Figure 5 presents the idea of the walk-forward process. After the end of the testing period, a new walk-forward process unit is created, which starts from the previous position plus the duration of the testing period. Thanks to this shift, AIS can choose models with more updated data and get a very long out-of-sample period, starting in the T_0 period.

Figure 5: The logic behind the rolling walk-forward process

Note: The diagram shows how a long out-of-sample period can be achieved while using up-to-date information to train the models. T_0 means the start of the long out-of-sample period.

4.5. All the tested approaches

There were other approaches tested before deciding which one would be a base case methodological approach selected for testing purposes.

Approach 1: Initially, an approach without a walk-forward process was tested. The LSTM and the GRU models were trained during one training period and tested during another testing period. The parameters in the models were selected based on the process of hyperparameters tuning that was done with the use of GridSearch.

Approach 2: In the next step, a walk-forward process was introduced, but only to extend the out-of-sample period. The walk-forward process did not include an optimization algorithm based on the IR* statistic in the validation period. So, the walk-forward process always had the same model with the same hyperparameters as in Approach 1.

Approach 3.1: Then, ten papers were selected based on which the hyperparameters for the models were chosen. This approach was used in this study and is already described in detail above.

Approach 3.2: This approach was based on approach 3.1, but here buy/sell signals were generated by the three models that achieved the highest Information Ratio (IR *) statistic during the validation period.

Approach 3.3: In this approach, one best model was selected based on a ranking that took into account the results from the previous five validation periods.

Approach 3.4: Here not only the rates of return were taken into account, but also volume, high and low prices.

4.6. Research Description

This study consisted of several steps, which are presented below:

1. Selecting financial instruments and data frequency, data downloading
2. Code base preparing, data cleaning, data preparation and AIS engineering. At this step, the code supporting the entire study was written. Additionally, the base case scenario was chosen. All the tested approaches are described in the section above.

3. Running the tests for the selected AIS and improving the testing methodology
4. Conducting a sensitivity analysis
5. Building an ensemble AIS based on the signals generated by the base case scenario model

5. Empirical results

Our investment algorithm is tested on daily and hourly data frequencies. The out-of-sample period starts on 1 January 2021 and ends on 1 April 2022. The transaction cost of 0.1% is charged every time when our algorithm generates a different signal than in the previous time step. The primary statistic based on which the performance of our investment algorithm is evaluated is the Adjusted Information Ratio, which is described by formula 16. The investment strategy results are compared with the "Buy&Hold" strategy, which means buying an asset at the beginning of the out-of-sample period from Figure 5, and holding the position until the end of the test.

Durations of the walk-forward process unit periods are different for the daily and hourly data. In the case of daily data, the duration of the training period is 720, the validation period lasts 90 observations, and the testing period also lasts 90. These periods are extended for data with hourly frequency: the training period contains 1800 observations, and the validation and testing periods contain 900 observations each.

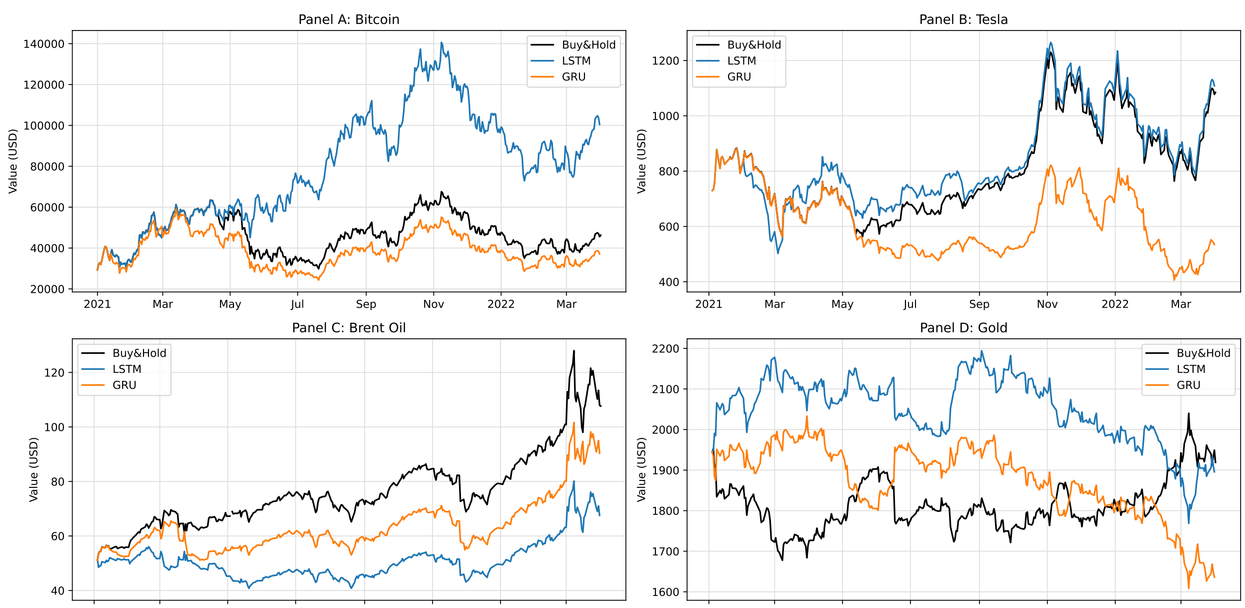
Table 3 shows the aggregated results of our trading algorithm for the daily data for all financial instruments. The table shows all the necessary performance metrics, including the Adjusted Information Ratio. Additionally, Figure 7 presents the same results in the form of equity lines. Our investment algorithm can outperform the Benchmark in terms of IR** statistics only twice. It beats the "Buy & Hold" strategy for Bitcoin and Tesla with the LSTM model. Additionally, when comparing the model architectures, the LSTM achieves a higher IR** statistic than the GRU for three out of the four financial instruments, with the Brent Oil exception.

Table 3: Performance metrics of the investment algorithm for daily data

		ARC(%)	ASD(%)	IR*	MD(%)	IR**	nTrades
Panel A	Bitcoin "Buy&Hold"	44.86%	88.24%	0.51	54.53%	0.42	1
	Bitcoin LSTM	**168.23%**	**77.66%**	**2.17**	**48.15%**	**7.57**	**39**
	Bitcoin GRU	20.74%	78.02%	0.27	58.83%	0.09	9
Panel B	Tesla "Buy&Hold"	39.30%	58.34%	0.67	43.60%	0.61	1
	Tesla LSTM	**39.89%**	**58.05%**	**0.69**	**42.92%**	**0.64**	**15**
	Tesla GRU	-22.04%	58.17%	-0.38	53.92%	-0.15	63
Panel C	Brent Oil "Buy&Hold"	**77.09%**	**38.40%**	**2.01**	**26.26%**	**5.89**	**1**
	Brent Oil LSTM	25.14%	38.98%	0.65	27.32%	0.59	18
	Brent Oil GRU	58.04%	38.85%	1.49	22.87%	3.79	31
Panel D	Gold "Buy&Hold"	**0.56%**	**14.56%**	**0.04**	**14.18%**	**0.00**	**1**
	Gold LSTM	-2.05%	15.12%	-0.13	19.40%	-0.01	42
	Gold GRU	-12.91%	15.24%	-0.85	20.89%	-0.52	43

Note: Performance metrics for the algorithm tested from 1 January 2021 to 1 April 2022. The algorithm uses data with a daily frequency. Panel A shows the results for Bitcoin, and Panel B shows the results for Tesla, Panel C - Brent Oil, and Panel D - Gold. LSTM/GRU stands for investment algorithms using these architectures of recurrent neural networks. The training period of the walk-forward process has 720 observations, and the validation and testing periods contain 90 observations each. Each Panel has one strategy in bold, which means that the given strategy has the highest Adjusted Information Ratio.

Figure 7: Equity lines for daily data



Note: Performance metrics for the algorithm tested from 1 January 2021 to 1 April 2022. The algorithm uses data with a daily frequency. Panel A presents the equity lines for Bitcoin, Panel B presents the equity lines for Tesla, Panel C - Brent Oil, and Panel D - Gold. LSTM/GRU stands for investment algorithms using these architectures of recurrent neural networks. The training period of the walk-forward process has 720 observations, and the validation and testing periods contain 90 observations each.

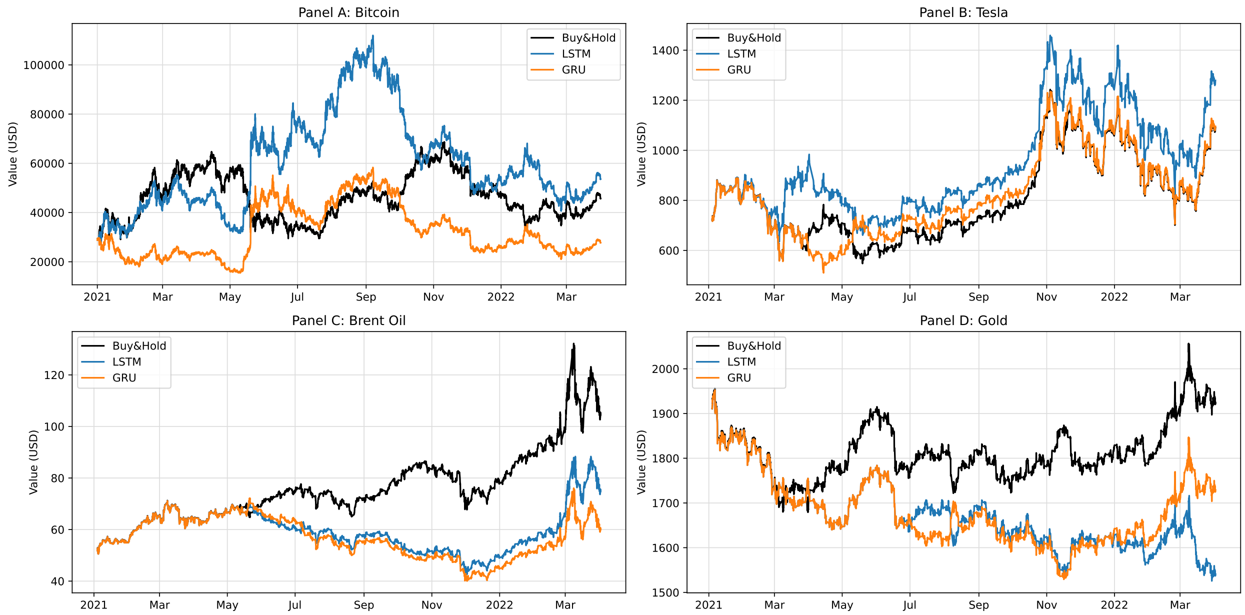
Figure 8 and Table 4 present the results for the trading strategy tested on hourly data. The investment strategy with the LSTM architecture achieves better results than the "Buy&Hold" strategy for two out of the four assets: Bitcoin and Tesla. The algorithm with GRU architecture can beat the market only for Tesla. Comparing these architectures shows that the LSTM model has higher IR** statistics than the GRU model for every asset except Gold.

Table 4: Performance metrics of the investment algorithm for hourly data

		ARC(%)	ASD(%)	IR*	MD(%)	IR**	nTrades
Panel A	Bitcoin "Buy&Hold"	44.86%	88.24%	0.51	54.53%	0.42	1
	Bitcoin LSTM	**64.50%**	**88.25%**	**0.73**	**63.71%**	**0.74**	**317**
	Bitcoin GRU	-3.01%	88.25%	-0.03	63.94%	0.00	479
Panel B	Tesla "Buy&Hold"	39.30%	58.34%	0.67	43.60%	0.61	1
	Tesla LSTM	**58.22%**	**58.31%**	**1**	**43.60%**	**1.33**	**139**
	Tesla GRU	39.79%	58.34%	0.68	42.91%	0.63	63
Panel C	Brent Oil "Buy&Hold"	**77.09%**	**38.40%**	**2.01**	**26.26%**	**5.89**	**1**
	Brent Oil LSTM	34.85%	38.40%	0.91	40.05%	0.79	17
	Brent Oil GRU	12.63%	38.44%	0.33	44.52%	0.09	27
Panel D	Gold "Buy&Hold"	**0.56%**	**14.56%**	**0.04**	**14.18%**	**0.00**	**1**
	Gold LSTM	-16.11%	14.64%	-1.1	21.87%	-0.81	71
	Gold GRU	-7.84%	14.56%	-0.54	21.69%	-0.19	7

Note: Performance metrics for the algorithm tested from 1 January 2021 to 1 April 2022. The algorithm uses data with an hourly frequency. Panel A shows the results for Bitcoin, and Panel B shows the results for Tesla, Panel C - Brent Oil, and Panel D - Gold. LSTM/GRU stands for investment algorithms using these architectures of recurrent neural networks. The training period of the walk-forward process has 1800 observations, and the validation and testing periods contain 900 observations each. Each Panel has one strategy in bold, which means that the given strategy has the highest Adjusted Information Ratio.

Figure 8: Equity lines for hourly data



Note: Performance metrics for the algorithm tested from 1 January 2021 to 1 April 2022. The algorithm uses data with an hourly frequency. Panel A presents the equity lines for Bitcoin, Panel B presents the equity lines for Tesla, Panel C - Brent Oil, and Panel D - Gold. The training period of the walk-forward process has 1800 observations, and the validation and testing periods contain 900 observations each. LSTM/GRU stands for investment algorithms using these architectures of recurrent neural networks.

Based on the results for both data frequencies, it can be said that there are no grounds to reject the first hypothesis (RH1) because the LSTM model, in most cases (6 out of 8), outperforms the GRU model. On the other hand, in the case of the second hypothesis (RH2), there are already grounds to reject it because no selected model architecture can beat the market for more than two out of four assets. In addition, it can be said that the investment algorithm is not robust to changes in the financial instrument because its profitability differs for individual assets (RQ1). Also, the investment strategy is not robust to data frequency changes (RQ2) what can be seen based on the comparison of Tables 3 and 4. Change of the frequency of the data from daily to hourly leads to entirely different results. In order to answer on the third research question (RH3), a sensitivity analysis is conducted in Section 6, while Section 7 is devoted to the answer on the fourth research question (RQ4).

6. Sensitivity analysis

To answer the third research question (RQ3) and ensure that the results obtained from the investment algorithm are stable, a sensitivity analysis is conducted. During this analysis, the robustness of our algorithm is checked by changing the following parameters:

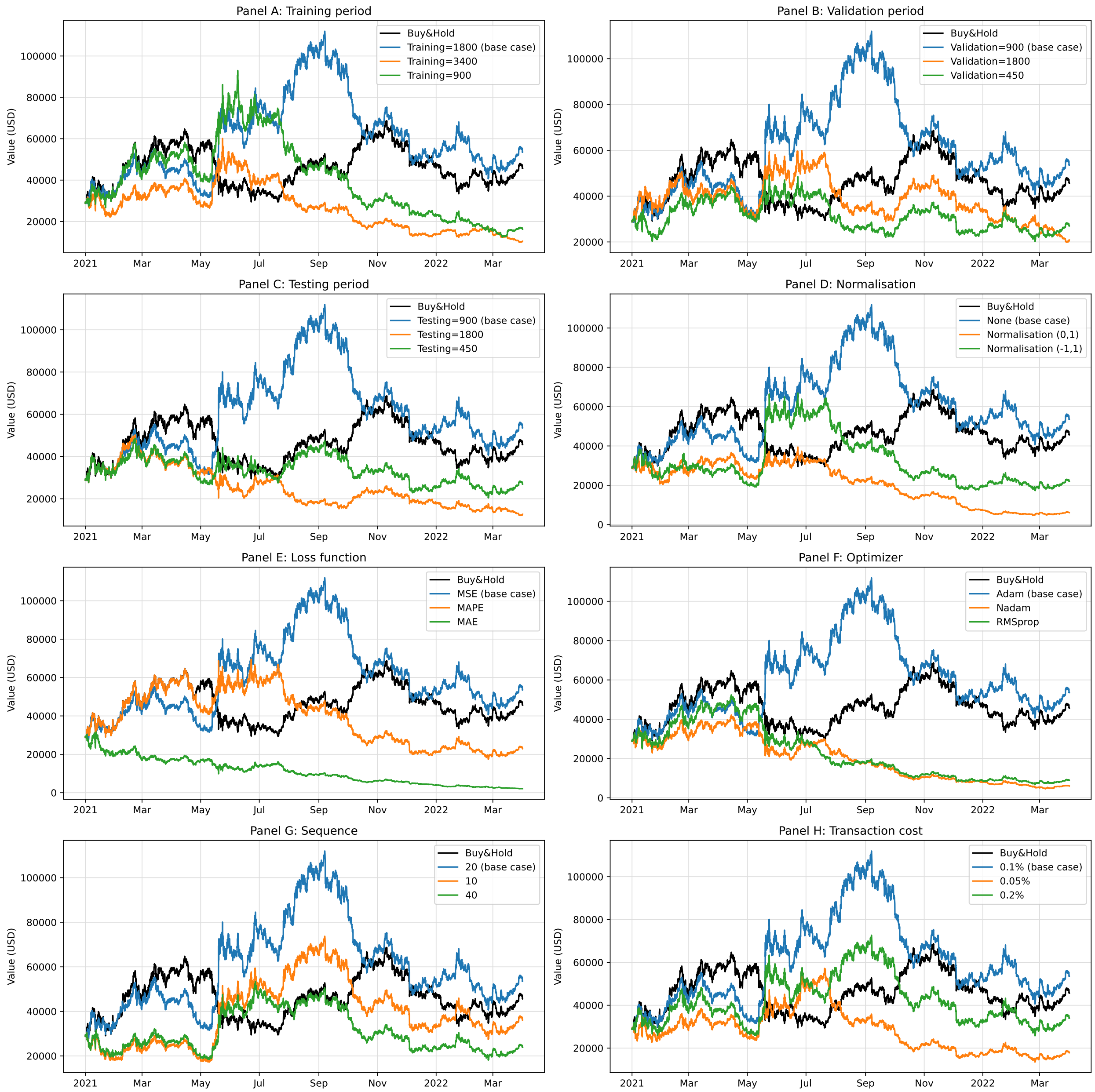
- the duration of the training period
- the duration of the validation period
- the duration of the testing period
- the type of input variable normalization
- the type of loss function
- the type of optimizer
- the sequence length
- the transaction cost

Each parameter is changed one at a time, using the ceteris paribus assumption for all the other parameters. After conducting these tests, the results are compared with the base case scenario and the "Buy&Hold" benchmark. In order to obtain a concise analysis report, sensitivity analysis for the algorithm is performed using only the LSTM architecture and only for hourly data.

The sensitivity of the walk-forward process unit periods duration is checked by reducing and increasing durations twice. The exception is the sensitivity analysis performed for the training period; the upper sensitivity testing limit is 3400 instead of 3600 due to insufficient hourly data. Similar idea is used for transaction cost and sequence length testing. Two types of variable normalization are checked: normalization to the range from 0 to 1 and normalization to the range from -1 to 1. Two additional optimizers are tested: Nadam and RMSprop. Also, MAPE and MAE loss functions are checked.

Table 5 and Figure 9 present the sensitivity analysis results for Bitcoin. The results show that the best IR** statistics are obtained for the Base Case scenario for every tested parameter. In most cases changing the parameters leads to negative returns. It is important to note that even reducing the transaction cost does not deliver a higher Adjusted Information Ratio (IR**). The reason for that is that the AIS has a walk-forward process that continuously selects the model with the highest IR*. Reducing the transaction cost causes the selection of totally different models that make transactions more frequently.

Figure 9: Sensitivity Analysis for Bitcoin



Note: The sensitivity analysis for Bitcoin is conducted in the period from 1 January 2021 to 1 April 2022. Each panel presents the equity lines for different parameters we perform sensitivity analysis for. In addition, we include the equity line of the Buy&Hold strategy to be able to compare the results.

Table 5: Sensitivity Analysis for Bitcoin

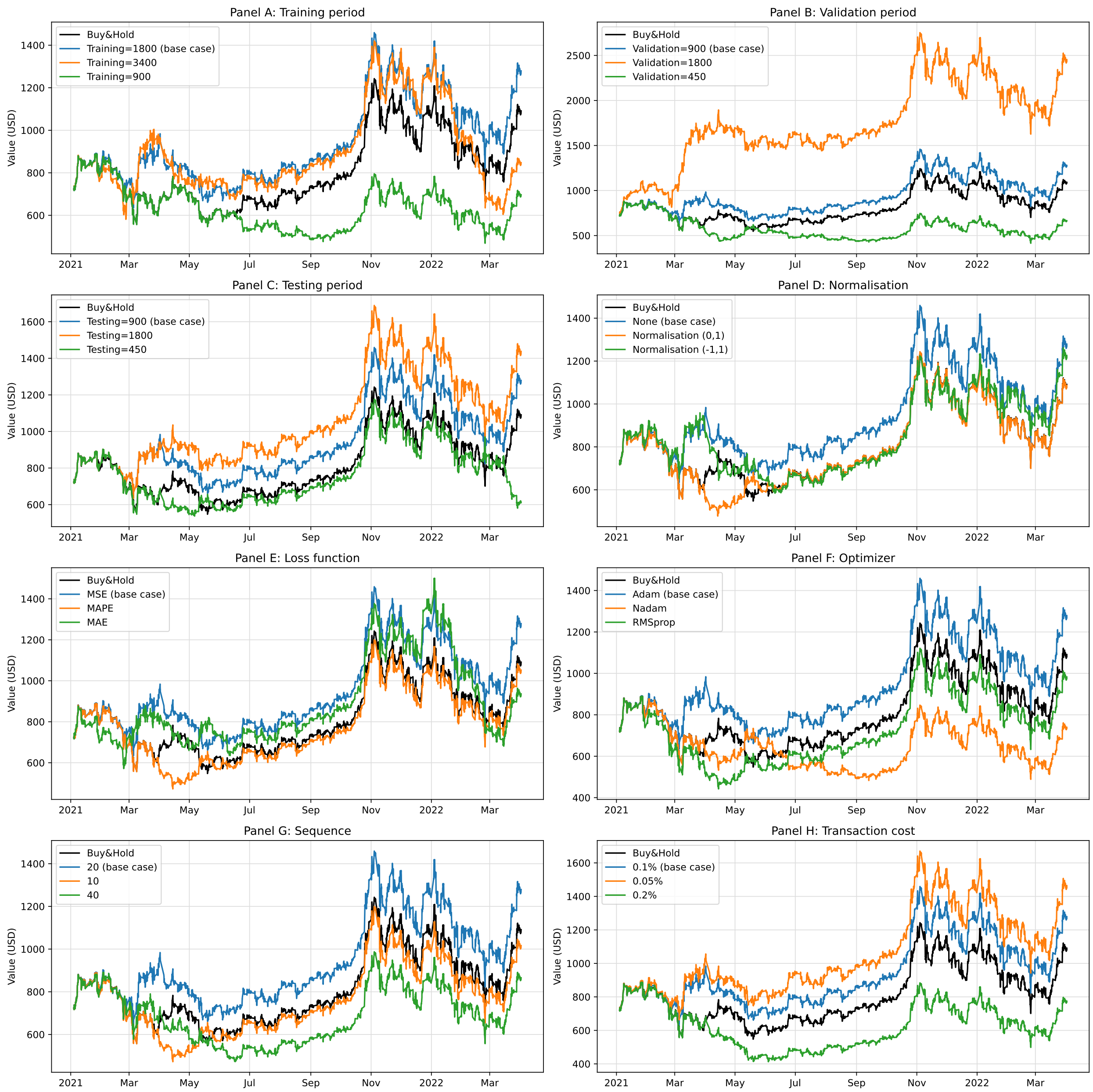
		ARC(%)	ASD(%)	IR*	MD(%)	IR**	nTrades
	Benchmark "Buy&Hold"	44.86%	88.24%	0.51	54.53%	0.42	1
Panel A: training period	Training period = 900	-36.60%	88.29%	-0.41	86.62%	-0.18	619
	Base case scenario (1800)	**64.50%**	**88.25%**	**0.73**	**63.71%**	**0.74**	**317**
	Training period = 3400	-55.77%	88.35%	-0.63	83.42%	-0.42	762
Panel B: validation period	Validation period = 450	-5.06%	88.28%	-0.06	59.11%	0.00	539
	Base case scenario (900)	**64.50%**	**88.25%**	**0.73**	**63.71%**	**0.74**	**317**
	Validation period = 1800	-22.80%	88.28%	-0.26	66.78%	-0.09	330
Panel C: testing period	Testing period = 450	-5.06%	88.24%	-0.06	57.94%	0.00	327
	Base case scenario (900)	**64.50%**	**88.25%**	**0.73**	**63.71%**	**0.74**	**317**
	Testing period = 1800	-48.18%	88.33%	-0.55	75.92%	-0.35	404
Panel D: normalisation	Normalisation = (0,1)	-71.33%	88.29%	-0.81	88.25%	-0.65	337
	Base case scenario (None)	**64.50%**	**88.25%**	**0.73**	**63.71%**	**0.74**	**317**
	Normalisation = (-1,1)	-19.50%	88.17%	-0.22	74.13%	-0.06	261
Panel E: loss function	Loss function = MAPE	-16.59%	88.24%	-0.19	74.93%	-0.04	37
	Base case scenario (MSE)	**64.50%**	**88.25%**	**0.73**	**63.71%**	**0.74**	**317**
	Loss function = MAE	-87.73%	88.35%	-0.99	93.45%	-0.93	728
Panel F: optimizer	Optimizer = Nadam	-71.49%	88.32%	-0.81	88.97%	-0.65	467
	Base case scenario (Adam)	**64.50%**	**88.25%**	**0.73**	**63.71%**	**0.74**	**317**
	Loss function = RMSprop	-61.03%	88.30%	-0.69	87.11%	-0.48	551
Panel G: sequence	Sequence = 10	19.96%	88.25%	0.23	62.81%	0.07	449
	Base case scenario (20)	**64.50%**	**88.25%**	**0.73**	**63.71%**	**0.74**	**317**
	Sequence = 40	-13.78%	88.33%	-0.16	66.03%	-0.03	489
Panel H: transaction cost	Transaction cost = 0.05%	-31.95%	88.24%	-0.36	76.36%	-0.15	645
	Base case scenario (0.1%)	**64.50%**	**88.25%**	**0.73**	**63.71%**	**0.74**	**317**
	Transaction cost = 0.2%	14.14%	88.39%	0.16	64.54%	0.04	277

Note: The sensitivity analysis for Bitcoin is conducted in the period from 1 January 2021 to 1 April 2022. Each panel presents the performance statistics for different parameters we perform sensitivity analysis for. In addition, we include the performance metrics of the Buy&Hold strategy to be able to compare the results. Each panel has one investment strategy in bold, which means the given strategy has the highest Adjusted Information Ratio.

Table 6 and Figure 10 present the sensitivity analysis for Tesla. Panel A, covering the training period, shows that the base case scenario has the highest IR** metric. Panel B, representing the sensitivity analysis testing the duration of the validation period, shows that the algorithm achieves the best results for a longer validation period of 1800 observations.

There is the same conclusion in the case with sensitivity analysis regarding the duration of the testing period. Panel H shows that the model achieves the highest IR** statistic for a reduced transaction cost. For the remaining parameters, the base case scenario is the best choice.

Figure 10: Sensitivity Analysis for Tesla



Note: The sensitivity analysis for Tesla is conducted in the period from 1 January 2021 to 1 April 2022. Each panel presents the equity lines for different parameters we perform sensitivity analysis for. In addition, we include the equity line of the Buy&Hold strategy to be able to compare the results.

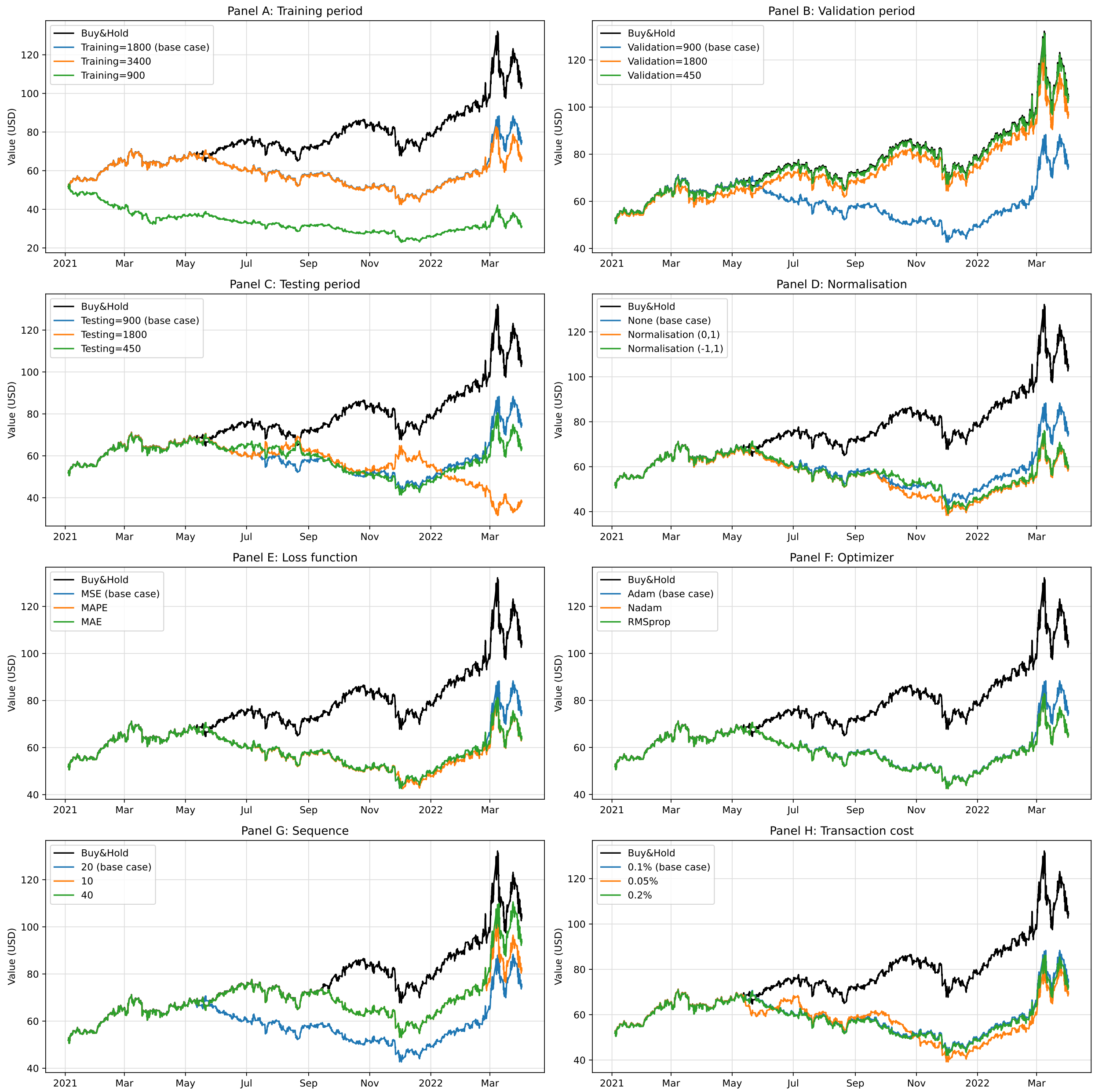
Table 6: Sensitivity Analysis for Tesla

		ARC(%)	ASD(%)	IR*	MD(%)	IR**	nTrades
	Benchmark "Buy&Hold"	39.30%	58.34%	0.67	43.60%	0.61	1
Panel A: training period	Training period = 900	-2.67%	58.33%	-0.05	47.34%	0.00	13
	Base case scenario (1800)	**58.22%**	**58.31%**	**1**	**43.60%**	**1.33**	**139**
	Training period = 3400	13.85%	58.41%	0.24	57.44%	0.06	303
Panel B: validation period	Validation period = 450	-6.29%	58.34%	-0.11	53.44%	-0.01	69
	Base case scenario (900)	58.22%	58.31%	1	43.60%	1.33	139
	Validation period = 1800	**166.49%**	**58.30%**	**2.86**	**40.94%**	**11.61**	**245**
Panel C: testing period	Testing period = 450	-12.43%	58.35%	-0.21	50.68%	-0.05	98
	Base case scenario (900)	58.22%	58.31%	1	43.60%	1.33	139
	Testing period = 1800	**73.73%**	**58.31%**	**1.26**	**43.60%**	**2.14**	**107**
Panel D: normalisation	Normalisation = (0,1)	38.89%	58.34%	0.67	46.32%	0.56	17
	Base case scenario (None)	**58.22%**	**58.31%**	**1**	**43.60%**	**1.33**	**139**
	Normalisation = (-1,1)	52.96%	58.31%	0.91	39.09%	1.23	267
Panel E: loss function	Loss function = MAPE	35.32%	58.33%	0.61	46.92%	0.46	3
	Base case scenario (MSE)	**58.22%**	**58.31%**	**1**	**43.60%**	**1.33**	**139**
	Loss function = MAE	23.24%	58.38%	0.4	54.62%	0.17	385
Panel F: optimizer	Optimizer = Nadam	2.06%	58.33%	0.04	45.73%	0.00	49
	Base case scenario (Adam)	**58.22%**	**58.31%**	**1**	**43.60%**	**1.33**	**139**
	Loss function = RMSprop	28.15%	58.34%	0.48	50.30%	0.27	23
Panel G: sequence	Sequence = 10	31.64%	58.34%	0.54	46.92%	0.37	5
	Base case scenario (20)	**58.22%**	**58.31%**	**1**	**43.60%**	**1.33**	**139**
	Sequence = 40	15.76%	58.36%	0.27	46.96%	0.09	131
Panel H: transaction cost	Transaction cost = 0.05%	**76.32%**	**58.31%**	**1.31**	**43.60%**	**2.29**	**181**
	Base case scenario (0.1%)	58.22%	58.31%	1	43.60%	1.33	139
	Transaction cost = 0.2%	5.92%	58.40%	0.1	53.50%	0.01	77

Note: The sensitivity analysis for Tesla is conducted in the period from 1 January 2021 to 1 April 2022. Each panel presents the performance statistics for different parameters we perform sensitivity analysis for. In addition, we include the performance metrics of the Buy&Hold strategy to be able to compare the results. Each panel has one investment strategy in bold, which means the given strategy has the highest Adjusted Information Ratio.

Table 7 and Figure 11 describe the results of the sensitivity analysis conducted for Brent Oil. The table shows that the base case scenario is the best choice for the majority of the parameters. The highest statistic is achieved for a shorter validation period. Additionally, an increase of sequence length to 40 leads to better results in terms of IR** statistic.

Figure 11: Sensitivity Analysis for Brent Oil



Note: The sensitivity analysis for Brent Oil is conducted in the period from 1 January 2021 to 1 April 2022. Each panel presents the equity lines for different parameters we perform sensitivity analysis for. In addition, we include the equity line of the Buy&Hold strategy to be able to compare the results.

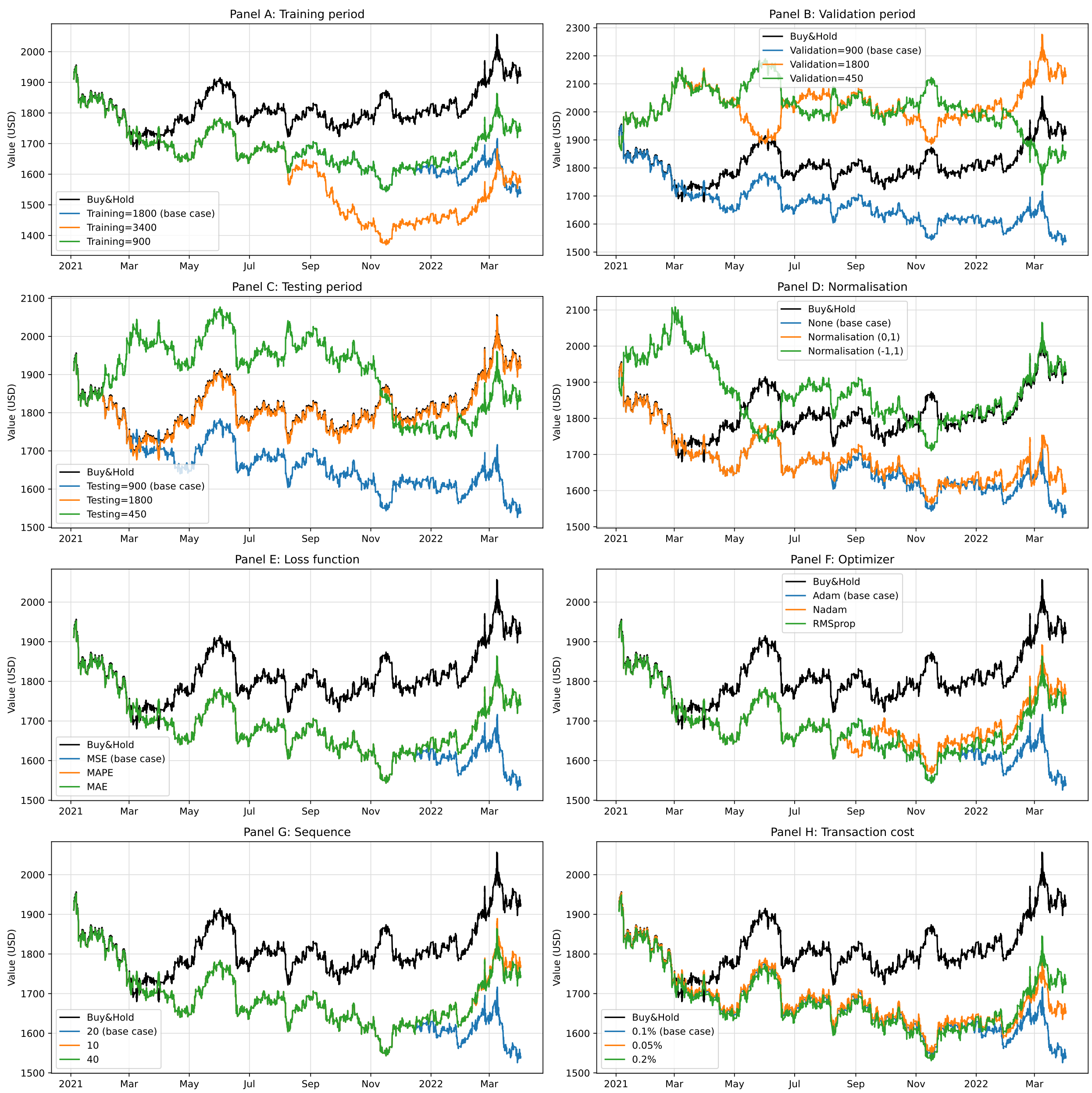
Table 7: Sensitivity Analysis for Brent Oil

		ARC(%)	ASD(%)	IR*	MD(%)	IR**	nTrades
	Benchmark "Buy&Hold"	77.09%	38.40%	2.01	26.26%	5.89	1
Panel A: training period	Training period = 900	-34.31%	38.43%	-0.89	56.71%	-0.54	174
	Base case scenario (1800)	**34.85%**	**38.40%**	**0.91**	**40.05%**	**0.79**	**17**
	Training period = 3400	22.35%	38.42%	0.58	40.34%	0.32	55
Panel B: validation period	Validation period = 450	**75.88%**	**38.40%**	**1.98**	**26.26%**	**5.71**	**3**
	Base case scenario (900)	34.85%	38.40%	0.91	40.05%	0.79	17
	Validation period = 1800	66.39%	38.41%	1.73	26.26%	4.37	15
Panel C: testing period	Testing period = 450	17.96%	38.41%	0.47	41.63%	0.20	9
	Base case scenario (900)	**34.85%**	**38.40%**	**0.91**	**40.05%**	**0.79**	**17**
	Testing period = 1800	-22.14%	38.41%	-0.58	55.66%	-0.23	4
Panel D: normalisation	Normalisation = (0,1)	11.04%	38.42%	0.29	46.18%	0.07	61
	Base case scenario (None)	**34.85%**	**38.40%**	**0.91**	**40.05%**	**0.79**	**17**
	Normalisation = (-1,1)	12.63%	38.43%	0.33	45.23%	0.09	97
Panel E: loss function	Loss function = MAPE	18.47%	38.39%	0.48	40.24%	0.22	21
	Base case scenario (MSE)	**34.85%**	**38.40%**	**0.91**	**40.05%**	**0.79**	**17**
	Loss function = MAE	18.86%	38.41%	0.49	40.05%	0.23	9
Panel F: optimizer	Optimizer = Nadam	20.79%	38.40%	0.54	40.34%	0.28	5
	Base case scenario (Adam)	**34.85%**	**38.40%**	**0.91**	**40.05%**	**0.79**	**17**
	Loss function = RMSprop	20.79%	38.40%	0.54	40.34%	0.28	5
Panel G: sequence	Sequence = 10	44.90%	38.40%	1.17	31.60%	1.66	3
	Base case scenario (20)	34.85%	38.40%	0.91	40.05%	0.79	17
	Sequence = 40	**61.96%**	**38.41%**	**1.61**	**31.60%**	**3.16**	**13**
Panel H: transaction cost	Transaction cost = 0.05%	26.77%	38.43%	0.7	45.03%	0.41	363
	Base case scenario (0.1%)	**34.85%**	**38.40%**	**0.91**	**40.05%**	**0.79**	**17**
	Transaction cost = 0.2%	31.15%	38.42%	0.81	40.77%	0.62	17

Note: The sensitivity analysis for Brent Oil is conducted in the period from 1 January 2021 to 1 April 2022. Each panel presents the performance statistics for different parameters we perform sensitivity analysis for. In addition, we include the performance metrics of the Buy&Hold strategy to be able to compare the results. Each panel has one investment strategy in bold, which means the given strategy has the highest Adjusted Information Ratio.

Table 8 and Figure 12 show results for Gold. The base case scenario does not obtain the highest IR** statistic for any of the parameters. The highest statistics are achieved for the shorter training period and more extended validation and testing periods. The highest IR** statistic is achieved for the normalization of the variable to the range from -1 to 1. The best results are obtained for both MAPE and MAE loss functions. Also, the model using the Nadam optimizer is the best. Reducing the sequence length leads to higher IR** statistic. Additionally, the most surprising finding is that an increase in transaction cost gives higher returns.

Figure 12: Sensitivity Analysis for Gold



Note: The sensitivity analysis for Gold is conducted in the period from 1 January 2021 to 1 April 2022. Each panel presents the equity lines for different parameters we perform sensitivity analysis for. In addition, we include the equity line of the Buy&Hold strategy to be able to compare the results.

Table 8: Sensitivity Analysis for Gold

		ARC(%)	ASD(%)	IR*	MD(%)	IR**	nTrades
	Benchmark "Buy&Hold"	0.56%	14.56%	0.04	14.18%	0.00	1
Panel A: training period	Training period = 900	**-7.16%**	**14.56%**	**-0.49**	**20.98%**	**-0.17**	**5**
	Base case scenario (1800)	-16.11%	14.64%	-1.1	21.87%	-0.81	71
	Training period = 3400	-14.51%	14.60%	-0.99	29.88%	-0.48	53
Panel B: validation period	Validation period = 450	-2.43%	14.56%	-0.17	20.56%	-0.02	11
	Base case scenario (900)	-16.11%	14.64%	-1.1	21.87%	-0.81	71
	Validation period = 1800	**9.30%**	**14.57%**	**0.64**	**12.66%**	**0.47**	**4**
Panel C: testing period	Testing period = 450	-3.24%	14.57%	-0.22	16.67%	-0.04	19
	Base case scenario (900)	-16.11%	14.64%	-1.1	21.87%	-0.81	71
	Testing period = 1800	**0.56%**	**14.56%**	**0.04**	**14.18%**	**0.00**	**1**
Panel D: normalisation	Normalisation = (0,1)	-13.50%	14.61%	-0.92	19.99%	-0.62	29
	Base case scenario (None)	-16.11%	14.64%	-1.1	21.87%	-0.81	71
	Normalisation = (-1,1)	**0.96%**	**14.60%**	**0.07**	**18.88%**	**0.00**	**60**
Panel E: loss function	Loss function = MAPE	**-7.16%**	**14.56%**	**-0.49**	**20.98%**	**-0.17**	**5**
	Base case scenario (MSE)	-16.11%	14.64%	-1.1	21.87%	-0.81	71
	Loss function = MAE	**-7.16%**	**14.56%**	**-0.49**	**20.98%**	**-0.17**	**5**
Panel F: optimizer	Optimizer = Nadam	**-6.01%**	**14.56%**	**-0.41**	**19.77%**	**-0.13**	**5**
	Base case scenario (Adam)	-16.11%	14.64%	-1.1	21.87%	-0.81	71
	Loss function = RMSprop	-7.16%	14.56%	-0.49	20.98%	-0.17	5
Panel G: sequence	Sequence = 10	**-6.06%**	**14.57%**	**-0.42**	**20.98%**	**-0.12**	**13**
	Base case scenario (20)	-16.11%	14.64%	-1.1	21.87%	-0.81	71
	Sequence = 40	-7.16%	14.56%	-0.49	20.98%	-0.17	5
Panel H: transaction cost	Transaction cost = 0.05%	-11.10%	14.58%	-0.76	20.74%	-0.41	71
	Base case scenario (0.1%)	-16.11%	14.64%	-1.1	21.87%	-0.81	71
	Transaction cost = 0.2%	**-7.92%**	**14.58%**	**-0.54**	**21.45%**	**-0.20**	**5**

Note: The sensitivity analysis for Gold is conducted in the period from 1 January 2021 to 1 April 2022. Each panel presents the performance statistics for different parameters we perform sensitivity analysis for. In addition, we include the performance metrics of the Buy&Hold strategy to be able to compare the results. Each panel has one investment strategy in bold, which means the given strategy has the highest Adjusted Information Ratio.

Summarizing the conducted sensitivity analysis, it can be said that our strategy is not robust to changes in the walk-forward process unit periods duration. Changing the duration of the periods may lead to an improvement or a deterioration in the results. The same conclusion also applies to the rest of the tested model parameters (RQ3). Table 9 shows how often a particular parameter value is selected due to the highest statistic of Adjusted Information Ratio.

Table 9: Summary of the conducted sensitivity analysis

Panel A: training period		***Panel E: loss function***	
Training period = 900	1	Loss function = MAPE	0.5
Base case scenario (1800)	2	Base case scenario (MSE)	3
Training period = 3400	1	Loss function = MAE	0.5

Panel B: validation period		***Panel F: optimizer***	
Validation period = 450	1	Optimizer = Nadam	1
Base case scenario (900)	1	Base case scenario (Adam)	3
Validation period = 1800	2	Loss function = RMSprop	0

Panel C: testing period		***Panel G: sequence***	
Testing period = 450	0	Sequence = 10	1
Base case scenario (900)	2	Base case scenario (20)	2
Testing period = 1800	2	Sequence = 40	1

Panel D: normalisation		***Panel H: transaction cost***	
Normalisation = (0,1)	0	Transaction cost = 0.05%	1
Base case scenario (None)	3	Base case scenario (0.1%)	2
Normalisation = (-1,1)	1	Transaction cost = 0.2%	1

Note: The table shows for how many financial instruments each hyperparameter value obtained the highest IR** statistics during the sensitivity analysis.

7. Ensemble AIS

In order to diversify the results of the investment algorithm among all financial instruments, ensemble AIS is created. The idea behind ensemble AIS is that 25,000 dollars are invested in each financial instrument, assuming that these instruments are perfectly divisible. As the source of buy/sell signals, the algorithm described in this paper is used. The ensemble AIS has LSTM architecture and base case scenario parameters described in Section 5. The testing period is the same as for the previous test. It starts on 1 January 2021 and ends on 1 April 2022.

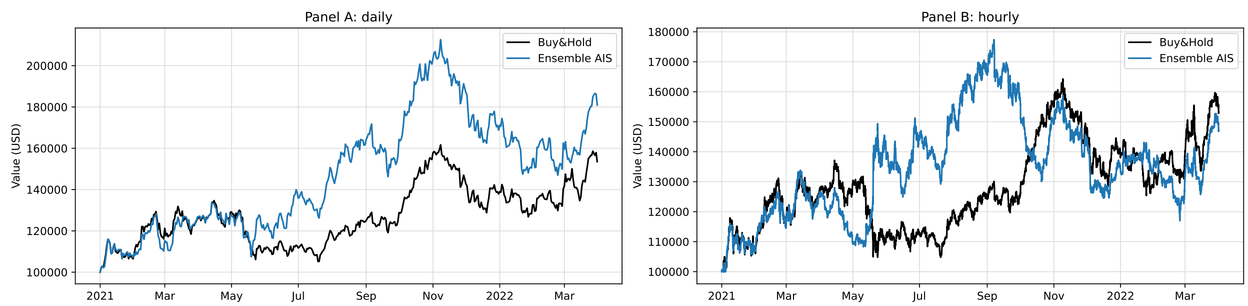
Table 10 and Figure 13 show the Ensemble AIS results tested at daily and hourly frequencies. The ensemble AIS is able to obtain a higher risk-adjusted return than the benchmark only for daily frequency (RQ4). It is worth noting that the investment algorithm tested on hourly data makes five times more transactions than the one performed on daily data.

Table 10: Performance metrics for the ensemble AIS

		ARC(%)	ASD(%)	IR*	MD(%)	IR**	nTrades
Panel A: hourly	"Buy&Hold"	**40.76%**	**32.89%**	**1.24**	**23.66%**	**2.14**	**4**
	Ensemble AIS	36.23%	37.01%	0.98	34.03%	1.04	544
Panel B: daily	"Buy&Hold"	41.16%	32.94%	1.25	21.87%	2.35	4
	Ensemble AIS	**61.09%**	**38.86%**	**1.57**	**30.89%**	**3.11**	**114**

Note: Performance metrics for the ensemble AIS tested from 1 January 2021 to 1 April 2022. Panel A shows the results for daily data, and Panel B shows the results for hourly data. Each Panel has one strategy in bold, which means that the given strategy has the highest Adjusted Information Ratio.

Figure 13: Equity lines for the ensemble AIS



Note: Equity lines for the ensemble AIS tested from 1 January 2021 to 1 April 2022. Panel A shows the results for daily data, and Panel B shows the results for hourly data.

8. Conclusions

Our study aimed to build a profitable trading strategy and explain what architectures of recurrent neural networks (LSTM or GRU) better predict the prices of selected financial instruments. For this purpose, 10 different models were selected based on the literature. Then a walk-forward process was made that was responsible for training all the models and selecting the model with the highest Information Ratio (IR*) statistics in the validation period. Next, the selected model generated buy/sell signals in the walk-forward testing period. The investment algorithm was tested for four different asset classes: cryptocurrencies (Bitcoin), shares (Tesla), energy commodities (Brent Oil), and metals commodities (Gold). The AIS was tested from 1 January 2021 to 1 April 2022 on the data, which has daily and hourly frequencies. Our algorithm made decisions solely based on historical rates of returns on the selected asset.

In this work, two hypotheses and four research questions were posed:

RH1: *LSTM model outperforms the GRU model in most cases (in more than 50% of cases).* In Section 5, the results of our tests for all the financial instruments were presented. The LSTM performed better on three out of the four instruments for both frequencies, so there are no grounds to reject this hypothesis.

RH2: *The algorithm is able to obtain a higher risk-adjusted rate of return than the "Buy&Hold" strategy for every tested asset.* The results presented in Section 5 show that our algorithm for the selected LSTM / GRU architecture cannot beat the market for more than two out of the four instruments, so this hypothesis is rejected.

RQ1: *Is the investment strategy robust to changes in the financial instrument it predicts?* The results differ significantly for each of the financial instruments, so it can be said that the investment strategy is not robust to changes in assets it predicts.

RQ2: *Is the investment strategy robust to changes in the data frequency?* Comparing the results in Tables 3 and 4, it can be noticed that the results differ significantly for different data frequencies, so the investment strategy is not robust.

RQ3: *Is the investment strategy robust to changes in model parameters?* The sensitivity analysis performed in Section 6 showed that the investment strategy is not robust to changes in model parameters. Changes in different parameters led to an improvement or a deterioration in the results.

RQ4: *Is the ensemble AIS able to obtain a higher risk-adjusted rate of return than the benchmark?* Section 7 presented the results for the Ensemble AIS that beat the benchmark "Buy&Hold" for daily data. So, the answer to this question is yes, but it depends on the data frequency.

There are several areas in which our study can be extended. First of all, it is worth checking whether the results depend on the volatility of financial instruments by increasing the number of assets in each class of assets. Additionally, it is good to extend the number of features in input layer - use not only rates of return but also other information. Also, the way how the best model is selected during the validation period can be changed. Furthermore, ensembling several models (models different from RNN) can be considered.

References

[1] Adebiyi A. A., Adewumi A. O., Ayo C. K., 2014, *Comparison of ARIMA and Artificial Neural Networks Models for Stock Price Prediction*, Journal of Applied Mathematics, vol. 2014, Article ID 614342. https://doi.org/10.1155/2014/614342

[2] Box G. E. P., Jenkins G.M., 1976, *Time series analysis: „Forecasting and control"*, Holden-Day, San Francisco.

[3] Cho K., van Merrienboer B., Bahdanau D., Bengio Y., 2014, *On the Properties of Neural Machine Translation: Encoder-Decoder Approaches*, Cornell University, Computation and Language, https://doi.org/10.48550/arXiv.1409.1259

[4] Du J., Liu Q., Chen K., Wang J., 2019, *Forecasting stock prices in two ways based on LSTM neural network*, 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019, pp. 1083-1086, doi: 10.1109/ITNEC.2019.8729026

[5] Fama E. F., 1970, *Efficient capital markets: a review of theory and empirical work*, The Journal of Finance, Volume 25, Issue 2, pp. 383-417

[6] Fischer T., Krauss C., 2018, *Deep learning with long short-term memory networks for financial market predictions*, European Journal of Operational Research,Volume 270, Issue 2, Pages 654-669

[7] Ghosh P., Neufeld A., Sahoo J. K., 2021, *Forecasting directional movements of stock prices for intraday trading using LSTM and random forests*, Finance Research Letters, Elsevier, vol. 46(PA)

[8] Girsang A. S., Lioexander F., Tanjung D., 2020, *Stock Price Prediction Using LSTM and Search Economics Optimization*, International Journal of Computer Science, Volume 47, Issue 4.

[9] Hochreiter, S., 1991. *Untersuchungen zu dynamischen neuronalen Netzen*, Diploma thesis, Institut für Informatik, Technische Universität München

[10] Hochreiter S., Schmidhuber J., 1997, *Long Short-Term Memory*, Neural Computation, Volume 9, Issue 8

[11] Kijewski M., Ślepaczuk R., 2020, *Predicting prices of S&P500 index using classical methods and recurrent neural networks*, Working Papers of Faculty of Economic Sciences, University of Warsaw, WP 27/2020 (333)

[12] Krauss C., Do X. A., Huck N., 2017, *Deep neural networks, gradient-boosted trees, random forests: Statistical arbitrage on the S&P 500*, European Journal of Operational Research, Volume 259, Issue 2, Pages 689-702

[13] Lim D. B., Lundgren J., 2019, *Algorithmic Trading using LSTM-Models for Intraday Stock Predictions*

[14] Roondiwala M., Patel H., Varma S., 2017, *Predicting Stock Prices Using LSTM*, International Journal of Science and Research, Volume 6 Issue 4

[15] Rumelhart, D., Hinton, G., Williams, R., 1986, *Learning representations by back-propagating errors.*, Nature 323, 533–536. https://doi.org/10.1038/323533a0

[16] Ryś P., Ślepaczuk R., 2018, *Machine Learning Methods in Algorithmic Trading Strategy Optimization – Design and Time Efficiency*, Central European Economic Journal, vol.5, no.52, 2018, pp.206-229. https://doi.org/10.1515/ceej-2018-0021

[17] Sethia A., Raut P., 2019, *Application of LSTM, GRU and ICA for Stock Price Prediction*, Information and Communication Technology for Intelligent Systems. Smart Innovation, Systems and Technologies, vol 107. Springer, Singapore.

[18] Shahi T. B., Shrestha A., Neupane A., Guo W., 2020, *Stock Price Forecasting with Deep Learning: A Comparative Study*, Mathematics, 8, 1441. https://doi.org/10.3390/math8091441

[19] Siami-Namini S., Tavakoli N., Namin A. S., 2018, *A Comparison of ARIMA and LSTM in Forecasting Time Series*, 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 1394-1401, doi: 10.1109/ICMLA.2018.00227.

[20] Site A., Birant D., Işık Z., 2019, *Stock Market Forecasting Using Machine Learning Models*, 2019 Innovations in Intelligent Systems and Applications Conference (ASYU), pp. 1-6, doi: 10.1109/ASYU48272.2019.8946372.

[21] Zou Z., Qu Z., 2020, *Using LSTM in Stock prediction and Quantitative Trading*, CS230: Deep Learning

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