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

Illia Baranochnikov and Robert Ślepaczuk

QFRG&DSlab monthly meeting

Quantitative Finance Research Group, Department of Quantitative Finance,
Faculty of Economic Sciences, University of Warsaw

November 7, 2022

Motivation

- to build a profitable algorithmic investment strategy (AIS)
- to explain what type of Recurrent Neural Network (RNN) is more efficient
- to increase the efficiency of asset management
- to correctly evaluate the risk

Hypotheses and research questions - I

First Hypothesis:

The LSTM model outperforms the GRU model in most cases (in more than 50% of cases)

Second Hypothesis:

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)

Hypotheses and research questions - II

First Research Question:

Is the AIS robust to changes in the financial instrument it predicts?

Second Research Question:

Is the AIS robust to changes in the data frequency?

Third Research Question:

Is the AIS robust to changes in model hyperparameters?

Fourth Research Question:

Is the ensemble AIS able to obtain a higher risk-adjusted rate of return than the benchmark?

Literature review - I

1. **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.
2. **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.
3. **Rumelhart et al. (1986)** are the first to introduce the concept of the RNN model.
4. 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)**.
5. **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.
6. **Roondiwala et al. (2015)** use the LSTM model to forecast the Indian Stock Market NIFTY 50 Index level.
7. **Siami-Namini et al. (2018)** employ LSTM neural networks to forecast monthly closing prices for 11 stock market indices.
8. **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.

Literature review - II

9. **Kijewski and Ślepaczuk (2020)** conduct research comparing the profitability of investment strategies based on the ARIMA, LSTM and other classical methods.
10. The architecture of the GRU model was firstly presented in the work of **Cho et al. (2014)**.
11. **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.
12. **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.
13. **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.
14. 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.
15. **Du et al. (2019)** implement two LSTM models to forecast Apple stock prices.

Literature review - III

16. **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.
17. **Girsang et al. (2020)** propose a hybrid model that combines the LSTM model with an algorithm that optimizes the model training process.
18. **Zou and Qu (2020)** implement the LSTM model that incorporates the Attention Mechanism.

Data - I

4 different categories of assets

- cryptocurrencies (Bitcoin)
- stocks (Tesla)
- energy commodities (Brent Oil)
- metals commodities (Gold)

Features

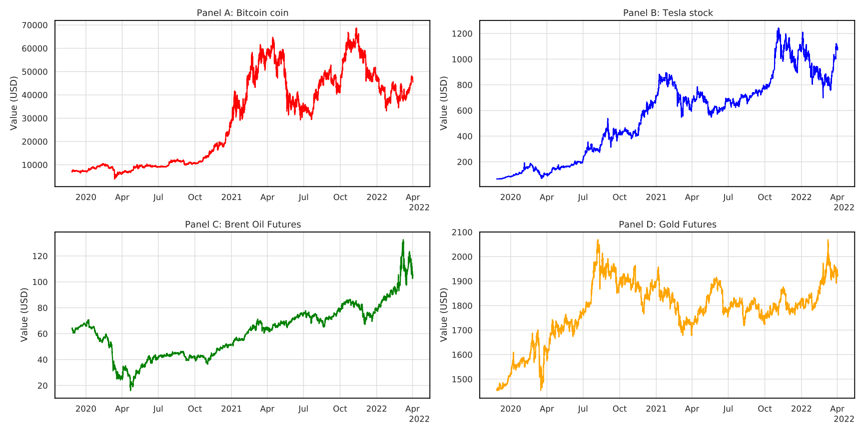
- daily and hourly frequency
- closing prices of assets => simple rates of return
- used as the only feature, with no normalization

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

where:
R_t - rate of return in period t
S_t, S_{t-1} - financial instrument prices in periods t and t-1, respectively

Data - II

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 November 27, 2019 to April 1, 2022.

Data - III

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 financial instruments on simple returns for the period from November 27, 2019 to April 1, 2022.

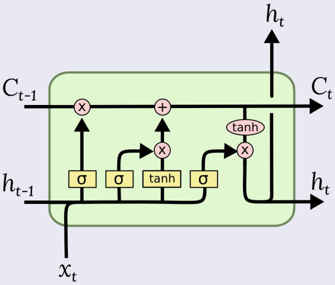
Methodology. Recurrent Neural Networks

- recurrent neural networks (RNNs) - primary source of buy/sell signals
- more effective in time series modelling than traditional feed-forward neural networks
- two types of RNNs: Gated Recurrent Unit (GRU) and Long Short-Term Memory (LSTM)
- no vanishing gradient problem

Methodology. Recurrent Neural Networks. LSTM - I

- LSTM is based on the idea of gates that decide how the information flow
- the cell state (C_t) that can "remember" long-term dependencies
- 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

Methodology. Recurrent Neural Networks. LSTM - II

- The forget gate

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

- The input gate

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

- The cell state

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

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

- The output gate

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

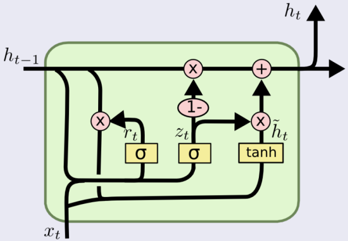
$$h_t = o_t * tanh(C_t) \tag{7}$$

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

Methodology. Recurrent Neural Networks. GRU - I

- hidden state instead of a cell state
- only two gates: a reset gate and an update gate
- use less computing power

Figure 3: The Gated Recurrent Unit architecture



Note: The architecture of GRU, source:

https://colah.github.io/posts/2015-08-Understanding-LSTMs

Methodology. Recurrent Neural Networks. GRU - II

- The reset gate

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

- The update gate

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

- The hidden state

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

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

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

Methodology. Model architectures - I

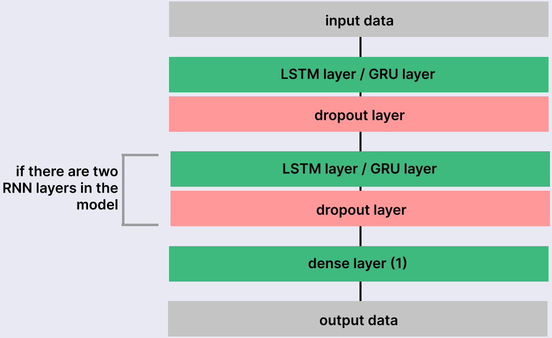
- 10 neural network architectures based on the literature
- the same architectures for LSTM and GRU
- these model architectures are not exactly like these from the literature

Common parameters of the models

- the maximum number of epochs is limited to 100
- the sequence length is 20 observations
- Mean Squared Error loss function and Adam optimizer with the AMSGrad extension

Methodology. Model architectures - II

Figure 4: The structure of every model



Note: The figure shows the overall structure of every model used in the study.

Methodology. Model architectures - III

Table 2: The model architectures used in the AIS

	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: Square brackets indicate hyperparameters that are changed or set in the process of the research.

Methodology. Performance metrics - I

1. ARC - Annualized Return Compounded.

The compounded interest rate of return of the AIS per annum, considering the annual number of observations for a particular financial instrument.

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

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 for daily data, this parameter was adequately adjusted for hourly data)
N - the number of observations over the entire period under study
R_t - the simple rate of return in period t

Methodology. Performance metrics - II

2. ASD - Annualized Standard Deviation

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

Methodology. Performance metrics - III

3. IR* - Information Ratio

The annualized risk-adjusted rate of return of the AIS.

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

4. MD - Maximum Drawdown

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

Methodology. Performance metrics - IV

5. IR** - Adjusted Information Ratio

The annualized risk-adjusted rate of return for AIS that considers not only ASD but also MD.

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

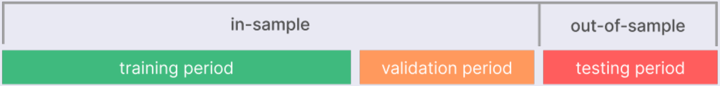
where:

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

Methodology. Walk-forward process for AIS testing - I

- Training period: observations from this period are used to train recurrent neural networks; during this period, all the models presented are trained
- 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
- Testing period: the best algorithm from the validation period is used to generate buy/sell signals on tested financial instruments in this period

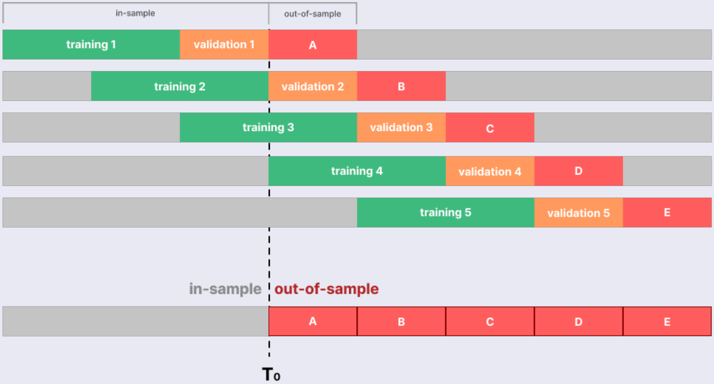
Figure 5: A walk-forward process unit



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.

Methodology. Walk-forward process for AIS testing - II

Figure 6: The logic behind the rolling walk-forward process



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

Methodology. All the tested approaches - I

1. The LSTM and the GRU models were trained during one training period and tested during another testing period. Hyperparameters tuning with the use of GridSearch.
2. A walk-forward process was introduced. The walk-forward process did not include an optimization algorithm based on the IR* statistic in the validation period.
3. Ten papers were selected based on which the hyperparameters for the models were chosen. This approach was used in this study.

Methodology. All the tested approaches - II

4. This approach was based on approach 3, but here buy/sell signals were generated by the three models that achieved the highest Information Ratio (IR *) statistic during the validation period.
5. One best model was selected based on the ranking that took into account the results from the previous five validation periods.
6. Here not only the rates of return were taken into account in the input layer, but also volume, high and low prices.

Methodology. Research Description

1. Selecting financial instruments and data frequency
2. Data downloading and cleaning.
3. AIS creating and 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.
4. Running the tests for the selected AIS and improving the testing methodology
5. Conducting a sensitivity analysis
6. Building an ensemble AIS based on the signals generated by the base case scenario model

Empirical results

- The out-of-sample period starts on January 1, 2021 and ends on April 1, 2022.
- The transaction cost of 0.1% is charged for every trade.
- Adjusted Information Ratio (IR**) as the primary perfomance metric.
- "Buy&Hold" strategy as the main benchmark. why?
- daily data walk-forward length:
  - the training period = 720 days
  - the validation period = 90 days
  - the testing period = 90 days
- hourly data walk-forward length:
  - the training period = 1800 observations
  - the validation period = 900 observations
  - the testing period = 900 observations

Empirical results - daily data

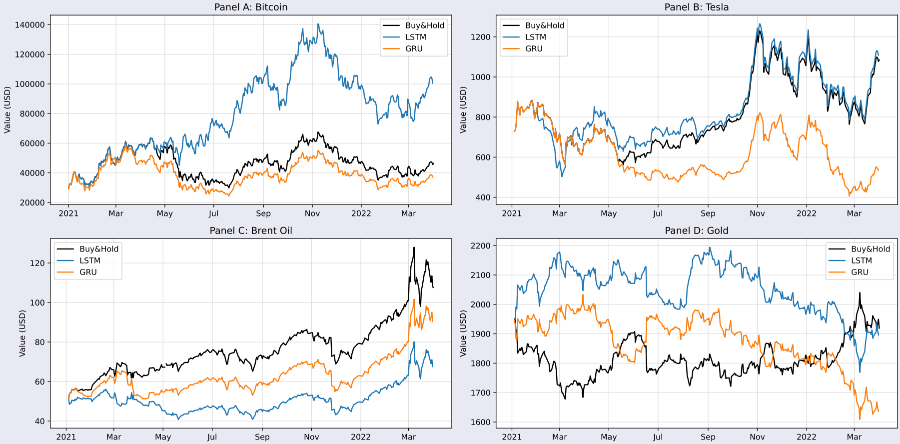
Table 3: Performance metrics of the investment algorithm for daily data

		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 January 1, 2021 to April 1, 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 (IR**).

Empirical results - daily data

Figure 7: Equity lines for daily data



Note: Performance metrics for the algorithm tested from January 1, 2021 to April 1, 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.

Empirical results - hourly data

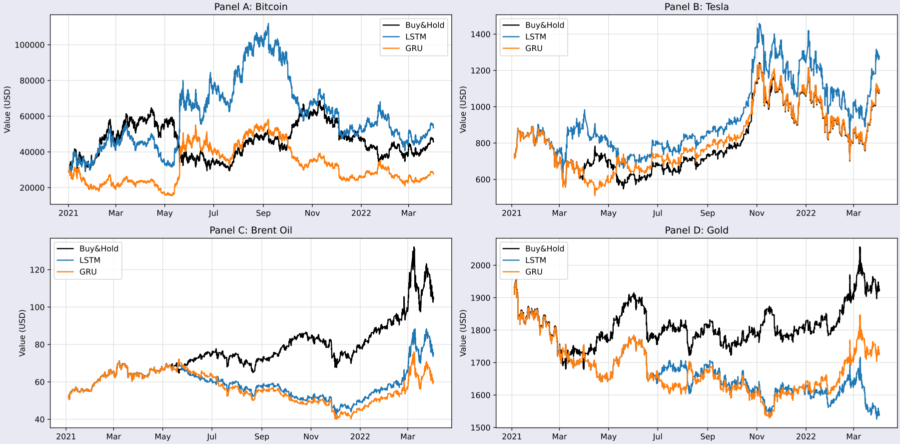
Table 4: Performance metrics of the investment algorithm for hourly data

		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 January 1, 2021 to April 1, 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 (IR**).

Empirical results - hourly data

Figure 8: Equity lines for hourly data



Note: Performance metrics for the algorithm tested from January 1, 2021 to April 1, 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.

Empirical results - daily and houry data

Empirical results - daily data

- Our investment algorithm outperforms the Benchmark in terms of IR** statistics only twice. It beats the "Buy & Hold" strategy for Bitcoin and Tesla with the LSTM model.
- the LSTM achieves a higher IR** statistic than the GRU for three out of the four financial instruments, with the Brent Oil exception.

Empirical results - 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.

Sensitivity analysis - for LSTM on hourly data

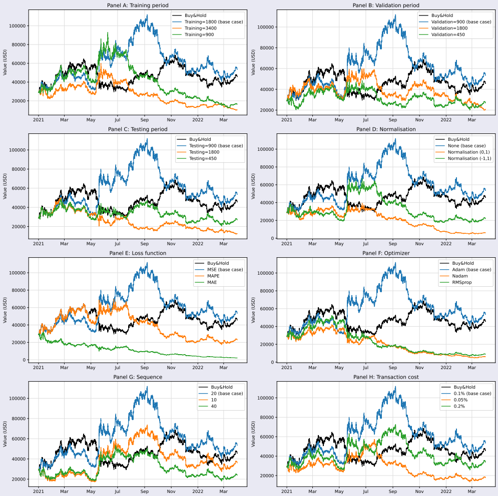
To answer the third research question (RQ3) and ensure that the results obtained from the investment algorithm are stable, a sensitivity analysis is conducted. During this analysis, the robustness of our algorithm is checked by changing the following parameters:

- the duration of the training period: {900, 1800, 3400}
- the duration of the validation period: {450, 900, 1800}
- the duration of the testing period: {450, 900, 1800}
- the type of input variable normalization: {(0, 1), None, (-1, 1)}
- the type of loss function: {MAPE, MSE, MAE}
- the type of optimizer: {Nadam, Adam, RMSprop}
- the sequence length: {10, 20, 40}
- the transaction cost: {0.05%, 0.1%, 0.2%}

Each parameter is changed using the *ceteris paribus* assumption for all the other parameters.

Sensitivity analysis - Bitcoin

Figure 9: Sensitivity Analysis for Bitcoin



Note: The sensitivity analysis for Bitcoin is conducted in the period from January 1, 2021 to April 1, 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.

Sensitivity analysis - Bitcoin

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

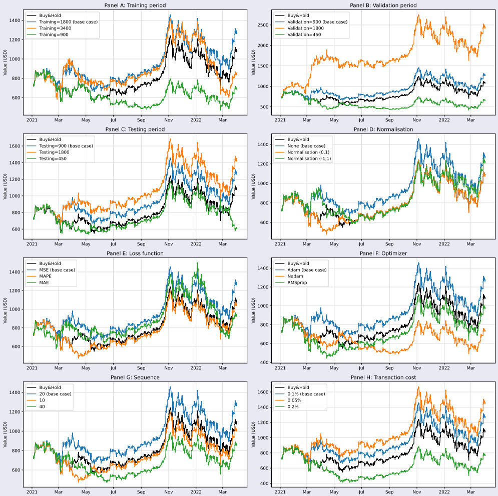
Note: The sensitivity analysis for Bitcoin is conducted in the period from January 1, 2021 to April 1, 2022. Each panel presents the performance statistics for different parameters we perform sensitivity analysis for. In addition, we include the performance metrics of the Buy&Hold strategy to be able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

Sensitivity analysis - Bitcoin

- the best IR** statistics are obtained for the Base Case scenario for every tested parameter.
- 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 Information Ratio. Reducing the transaction cost causes the selection of totally different models that make transactions more frequently

Sensitivity analysis - Tesla

Figure 10: Sensitivity Analysis for Tesla



Note: The sensitivity analysis for Tesla is conducted in the period from January 1, 2021 to April 1, 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.

Sensitivity analysis - Tesla

Table 6: Sensitivity Analysis for Tesla

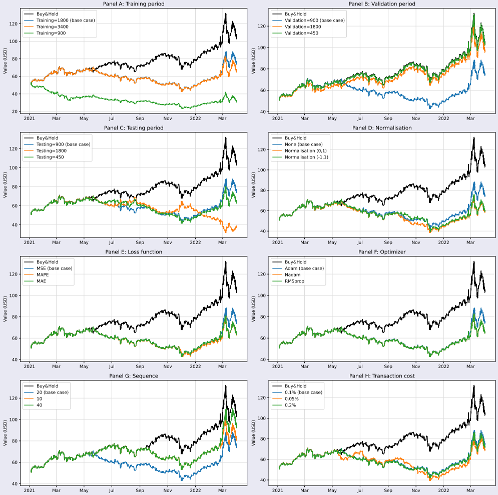
		ARC(%)	ASD(%)	IR*	MD(%)	IR**	nTrades
	Benchmark "Buy&Hold"	39.30%	58.34%	0.67	43.60%	0.61	1
Panel A: training period	Training period = 900	-2.67%	58.33%	-0.05	47.34%	0.00	13
	Base case scenario (1800)	**58.22%**	**58.31%**	**1**	**43.60%**	**1.33**	**139**
	Training period = 3400	13.85%	58.41%	0.24	57.44%	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 January 1, 2021 to April 1, 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.

Sensitivity analysis - Tesla

- the base case scenario has the highest IR** metric for:
    - trainng period
    - normalisation
    - loss function
    - optimizer
    - sequence
- but the better value was obtained for:
    - longer validation period (1800 observations)
    - longer testing period (1800 observations)
    - lower transactions cost

Figure 11: Sensitivity Analysis for Brent Oil



Note: The sensitivity analysis for Brent Oil is conducted in the period from January 1, 2021 to April 1, 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.

Sensitivity analysis - Brent Oil

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

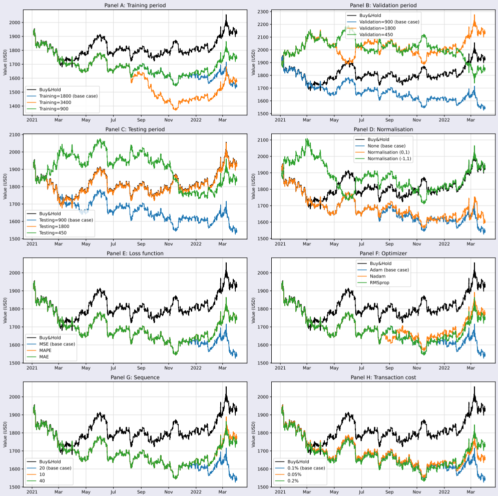
Note: The sensitivity analysis for Brent Oil is conducted in the period from January 1, 2021 to April 1, 2022. Each panel presents the performance statistics for different parameters we perform sensitivity analysis for. In addition, we include the performance metrics of the Buy&Hold strategy to be able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

Sensitivity analysis - Brent Oil

- the base case scenario has the highest IR** metric for:
  - trainng period
  - testing period
  - normalisation
  - loss function
  - optimizer
  - sequence
  - transactions cost
- but the better value was obtained for:
  - shorter validation period (450 observations)
  - longer sequence

Sensitivity analysis - Gold

Figure 12: Sensitivity Analysis for Gold



Note: The sensitivity analysis for Gold is conducted in the period from January 1, 2021 to April 1, 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.

Sensitivity analysis - Gold

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 January 1, 2021 to April 1, 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.

Sensitivity analysis - Gold

- The base case scenario does not obtain the highest IR** statistic for any of the parameters.
- the better value of IR** was obtained for:
  - longer testing period (1800 observations)
  - different normalisation (-1, 1)
- inconclusive (negative) value of IR** was obtained for:
  - training period
  - validation period
  - loss function
  - optimizer
  - sequence
  - transactions cost

Sensitivity analysis - Summary

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.

Sensitivity analysis - Summary

- 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, in sequence and in transaction costs.
- 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).
- the most stable results were obtained with regard to:
    - loss function
    - optimizer
    - normalisation
- the least stable results were for validation period

Ensemble AIS

- The idea behind ensemble AIS is that 1/4 of total equity is invested in each financial instrument, assuming that these instruments are perfectly divisible.
- As the source of buy/sell signals, the RNN 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 January 1, 2021 and ends on April 1, 2022.
- 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.

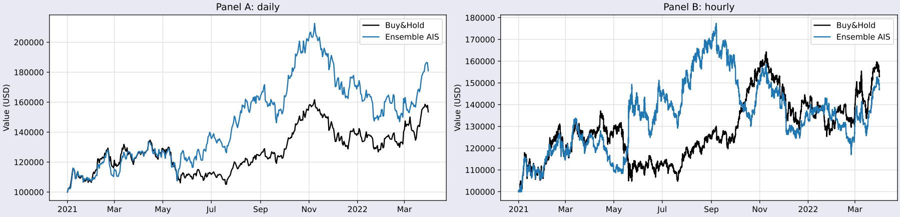
Ensemble AIS

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 January 1, 2021 to April 1, 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 January 1, 2021 to April 1, 2022. Panel A shows the results for daily data, and Panel B shows the results for hourly data.

Conclusions

RH1: LSTM model outperforms the GRU model in most cases (in more than 50% of cases).

The LSTM performed better for 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 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.

Potential extensions

- to check whether the results depend on the volatility of financial instruments by increasing the number of assets in each class of assets
- to extend the number of features in input layer - use not only rates of return but also other information
- to improve the way how the best model is selected during the validation period - various performance metrics
- to increase the number of correctly parametrized models in the selection phase
- to improve the process of ensembling by adding models different than RNN
- to include higher transaction costs in the training and validation phase in order increase the probability of selection of models with lower number of trades

References

- 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
- 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
- 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)
- 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.
- Hochreiter, S., 1991. Untersuchungen zu dynamischen neuronalen Netzen, Diploma thesis, Institut für Informatik, Technische Universität München
- Hochreiter S., Schmidhuber J., 1997, Long Short-Term Memory, Neural Computation, Volume 9, Issue 8
- 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)
- Roondiwala M., Patel H., Varma S., 2017, Predicting Stock Prices Using LSTM, International Journal of Science and Research, Volume 6 Issue 4
- 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.
- 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.
- Zou Z., Qu Z., 2020, Using LSTM in Stock prediction and Quantitative Trading, CS230: Deep Learning

Thank you for your attention!

Illia Baranochnikov
i.baranochniko@student.uw.edu.pl
University of Warsaw, Faculty of Economic Sciences
Quantitative Finance Research Group

Robert Ślepaczuk
rslepaczuk@wne.uw.edu.pl
University of Warsaw, Faculty of Economic Sciences
Department of Quantitative Finance, Quantitative Finance Research Group