



UNIVERSITY  
OF WARSAW



FACULTY OF  
ECONOMIC SCIENCES

## WORKING PAPERS

No. 25/2023 (432)

# HEDGING PROPERTIES OF ALGORITHMIC INVESTMENT STRATEGIES USING LONG SHORT-TERM MEMORY AND TIME SERIES MODELS FOR EQUITY INDICES

JAKUB MICHAŃKÓW  
PAWEŁ SAKOWSKI  
ROBERT ŚLEPACZUK

WARSAW 2023



## Hedging Properties of Algorithmic Investment Strategies using Long Short-Term Memory and Time Series models for Equity Indices

Jakub Michańków<sup>a,b</sup>, Paweł Sakowski<sup>b</sup>, Robert Ślepaczuk<sup>b\*</sup>

<sup>a</sup> Cracow University of Economics, Department of Informatics

<sup>b</sup> University of Warsaw, Faculty of Economic Sciences, Quantitative Finance Research Group, Department of Quantitative Finance

Corresponding author: [rslepaczuk@wne.uw.edu.pl](mailto:rslepaczuk@wne.uw.edu.pl)

---

**Abstract:** This paper proposes a novel approach to hedging portfolios of risky assets when financial markets are affected by financial turmoils. We introduce a completely novel approach to diversification activity not on the level of single assets but on the level of ensemble algorithmic investment strategies (AIS) built based on the prices of these assets. We employ four types of diverse theoretical models (LSTM - Long Short-Term Memory, ARIMA-GARCH - Autoregressive Integrated Moving Average - Generalized Autoregressive Conditional Heteroskedasticity, momentum, and contrarian) to generate price forecasts, which are then used to produce investment signals in single and complex AIS. In such a way, we are able to verify the diversification potential of different types of investment strategies consisting of various assets (energy commodities, precious metals, cryptocurrencies, or soft commodities) in hedging ensemble AIS built for equity indices (S&P 500 index). Empirical data used in this study cover the period between 2004 and 2022. Our main conclusion is that LSTM-based strategies outperform the other models and that the best diversifier for the AIS built for the S&P 500 index is the AIS built for Bitcoin. Finally, we test the LSTM model for a higher frequency of data (1 hour). We conclude that it outperforms the results obtained using daily data.

---

**Keywords:** machine learning, recurrent neural networks, long short-term memory, algorithmic investment strategies, testing architecture, loss function, walk-forward optimization, over-optimization

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

**Acknowledgements:** This document is the results of the research project funded by IDUB program: BOB-IDUB-622-233/2022 at the University of Warsaw

**ORCID ID:**

- Jakub Michańków: 0000-0002-0567-6240
- Paweł Sakowski: 0000-0003-3384-3795
- Robert Ślepaczuk: 0000-0003-3384-3795

## 1. Introduction

The main objective of this research is to improve the decision-making process by incorporating energy commodities and other asset classes into the hedging strategy of a diversified portfolio comprised of ensemble algorithmic investment strategies (AIS) constructed for the S&P 500 index. We present novel multidimensional verification of the possibilities of constructing and combining algorithmic investment strategies developed on the basis of 1) the Long Short-Term Memory (LSTM) model, 2) the ARIMA-GARCH class models, as well as concepts of 3) contrarian and 4) momentum strategies for various assets: equity indices, precious metals, energy and soft commodities, and cryptocurrencies. The selection of theoretical models and assets is dictated by the aim to include a set of those which is diverse enough and at the same time highly tested in the literature. We are going to achieve it by:

- testing the efficiency of single strategy and ensemble strategies built with: 1) various types of assets, 2) various theoretical models,
- introducing a walk-forward approach enabling us to test theoretical models on various training, validation, and testing periods with different characteristics of return distributions,
- verifying the diversification potential of various strategies built using different theoretical concepts and different types of assets in hedging investment strategies built on the S&P500 index,
- performing sensitivity analysis in order to check the robustness of final results to various frequencies of data.

Our main contribution to existing literature can be found in a completely novel approach to testing diversification and hedging potential. We focus on the combination of single and ensemble algorithmic investment strategies built for various types of assets in order to maximize risk-adjusted return instead of focusing on just a single combination of new assets with adequate characteristics of returns enabling us to optimize the weights of our portfolio.

In our research, we use a walk-forward procedure on a daily time series with dates ranging from 2004-01-02 to 2022-03-29. In practice, the starting point of data depends on the asset and availability of data for the tested asset and varies between 2004-01-02 and 2010-07-17. In order to accomplish the main aim we decided to formulate the following research questions (RQ):

- RQ1: *Which of the tested groups of assets (energy commodities, cryptocurrencies, gold, or soft commodities) has the largest diversification potential in complex AIS (built with machine learning (ML) models and ARIMA-GARCH models) for equity indices?*
- RQ2: *Are ML techniques more efficient than ARIMA-GARCH models and the concepts of momentum and contrarian in the case of single and complex (ensemble model combining all tested strategies for the given assets - type I) investment strategies.*
- RQ3: *Are complex (ensemble) AIS based on the aggregation of all theoretical models for the single asset (type I) or all assets for a single theoretical model (type II) more efficient than individual strategies?*
- RQ4: *Are results for LSTM models on higher frequencies of data (1h) better than those on daily data.*

The problem analyzed in this research is a fundamental issue not only from the micro, but also from the macro point of view, especially if we realize how much the stability of the financial systems of individual countries, and the state of savings of their citizens, are affected by the efficient and effective asset management in mutual and pension funds, investment funds, hedge funds or insurance companies. Wrong decisions in the allocation of these assets, especially in the context of long-term investment policies and specific investment strategies in the medium-term have very important consequences in the context of financial security and the quality of life of citizens of these countries. A similar approach to the one presented in this paper could also be extended to financial risk or macroeconomic forecasting.

The structure of this paper is as follows. After the introduction in Section 1, we present a comprehensive literature review in Section 2. Then, in Section 3 we describe the details of methodology and data. Finally, Section 4 covers the main results and Section 5 presents conclusions.

## 2. Literature review

In this short literature review, we present a historical background covering the development of (recurrent neural networks) RNN and long short-term memory (LSTM) models and the summary of various empirical papers testing the efficiency of LSTM on various types of assets, frequencies, and studies trying to ensemble it in different ways.

Hochreiter and Schmidhuber (1997) are responsible for the introduction of LSTM. By introducing Constant Error Carousel (CEC) units, LSTM deals with the exploding and vanishing gradient problems. The initial version of the LSTM block included cells, input, and output gates. Gers et al. (2000) introduced the forget gate (also called “keep gate”) into LSTM architecture, enabling the LSTM to reset its own state. They added peephole connections (connections from the cell to the gates) into the architecture. Additionally, the output activation function was omitted. Then, Chung et al. (2014) put forward a simplified variant called Gated Recurrent Unit (GRU).

Chen et al. (2015) implemented the LSTM model to predict the next-day returns for China stocks. Zhang et al. (2019) presented the AT-LSTM model which is the combination of LSTM and Attention-based model. They provided results for three index datasets: Russell 2000, DJIA, and NASDAQ, and argued that their framework for time series prediction is state-of-the-art against the baselines. Kijewski and Ślepaczuk (2020) compared the performance of classical techniques with the LSTM model for the S&P500 index on daily frequency for the last 20 years and showed that LSTM model results are highly dependent on initial hyperparameters assumptions. Siami-Namini et al. (2018) investigate whether and how newly identified deep learning time series forecasting algorithms, such as LSTM, outperform more seasoned ones. It is discovered that LSTM and other deep learning algorithms outperform more traditional algorithms like the ARIMA model. More specifically, LSTM outperformed ARIMA by achieving an average error rate reduction that was between 84 and 87 percent lower.

Castellano Gomez and Ślepaczuk (2021) tested the portfolio of algorithmic investment strategies (TA indicators, calendar anomalies, Macro, and ARIMA models) built on S&P500 and Nasdaq Composite indices in the period of the last 40 years. They revealed that especially ensemble models can beat the benchmark in times of turbulent events as well as during very fast market growth. Di Persio and Honchar (2017) analyze the performance of three different recurrent neural network models—a basic RNN, the LSTM, and the Gated Recurrent Unit (GRU) — using the price of Google stock. The authors also go over the RNN’s hidden dynamics and provide examples. The data clearly show that on a five-day horizon, the LSTM outperformed other versions with a 72 percent accuracy. Grudniewicz and Ślepaczuk (2023) applied several Machine Learning algorithms to technical analysis indicators for the WIG20, DAX, S&P 500, and a few selected CEE indices. The study’s findings reveal that quantitative techniques beat passive strategies in terms of risk-adjusted returns, with the Bayesian Generalized Linear Model and Naive Bayes being the top models for the investigated indices.

Studies additionally make an effort to combine an ensemble or hybrid technique with LSTM. Hossain et al. (2018) created a deep learning hybrid model using the well-known architectures: LSTM, and GRU (2018). The authors train a prediction model using the S&P 500 index time series, which spans over 66 years (1950 to 2016). This method involves passing the input data to the LSTM network, which generates a first-level prediction, and the output of the LSTM layer to the GRU layer, which generates the final prediction. With an MSE of 0.00098 in prediction, the proposed network outperforms earlier neural network methodologies. Michańków et al. (2022) compared the use of the LSTM model in AIS on BTC and S&P500 index on various frequencies. They showed that the efficiency of LSTM in AIS strictly depends on HT and the construction of the model and estimation process. Additionally, they introduced and revealed that proper Loss Function (Mean Absolute Directional Loss - MADL) is crucial in the model estimation process and that the results are dependent on asset classes tested and frequencies used. Their final results were not robust. Shah et al. (2018) provide a good example of how an LSTM-RNN model may deliver exceptional predictions on non-stationary data (2018). They show that the LSTM model not only yields great outcomes for daily forecasts, or predictions made one day in advance but also yields results that are more than satisfactory for predictions made seven days in advance using only the daily price as a feature.

Vo and Ślepaczuk (2022) compared the performance of ARIMA with the combination of ARIMA and

GARCH family models to forecast S&P 500 index log returns in order to construct algorithmic investment strategies on this index. Their main contribution was that the hybrid models outperformed ARIMA and the benchmark (Buy&Hold strategy on S&P 500 index) over the long term. These results were not sensitive to varying window sizes, the type of distribution, and the type of the GARCH model. The current advancements in high-frequency data estimation are related not only to technological issues and the growing processing power of big data but also to the requirement to understand and predict the behavior of variables over shorter time horizons. In their study on high-frequency Bitcoin trading, Lahmiri and Bekiros (2020) used three different kinds of machine learning (ML) models: (i) algorithmic models like regression trees, (ii) statistical ML techniques like support vector regressions (SVR), and (iii) ANN topologies like feedforward (FFNN) or Bayesian regularization (BRNN). Their findings show that artificial neural networks perform better than other types of systems in noisy signal environments. Baranochnikov and Ślepaczuk (2022) presented a walk-forward procedure that is in charge of training models and choosing the best one in order to predict future values of financial assets. They test the algorithms on four financial assets (Bitcoin, Tesla, Brent Oil, and Gold) and discover that LSTM outperforms GRU in the vast majority of cases. In order to compare the performance of random forests and LSTM networks (more specifically, CuDNNLSTM) in predicting the directional movements of the stocks that make up the S&P 500 index out-of-sample from January 1993 to December 2018 for intraday trading, Ghosh et al. (2022) used both training methodologies. In addition to returns relative to closing prices, they also introduced returns relative to opening prices and intraday returns in their multi-feature setting. In the end, they performed better than the benchmark.

Flori and Regoli (2021) used LSTM signals to improve portfolio performances of pairs trading strategies and showed that LSTM signals contain information that goes above and beyond traditional indicators. Moreover, what is important in our study they revealed that LSTM signals allow for the disentangling of the reversal effect from the momentum effect. Another paper that applied long short-term memory networks to financial market predictions was written by Fischer and Krauss (2018). LSTM was benchmarked against deep nets, random forests, and logistic regression. It occurred that Long short-term memory networks exhibit the highest predictional accuracy and returns.

Based on this literature review we can conclude that implementation of the forecasts from LSTM models in buy/sell signals can increase the efficiency of investment strategies. Moreover, we observe a growing number of publications on various types of ensemble models that combine frequencies or assets on the level of the given theoretical models or try to develop new investment techniques by joining many kinds of theoretical models in the process of price forecasting. Finally, we can notice that the type of input variables, the type of normalization, and specifically the architecture of the selected ML model can significantly affect the final results.

### 3. Methodology and Data

#### 3.1. Terminology and Metrics

The investment strategies we use in this work are based on the forecasts obtained from 1) ARIMA-GARCH class models, 2) the Long Short-Term Memory network (LSTM), the concepts of 3) contrarian, and 4) momentum effects. In the case of ARIMA-GARCH models, we apply the concise rolling walk-forward procedure with various Information Criteria (Akaike Information Criterion - AIC, Bayesian Information Criterion - BIC, Hannan–Quinn Information Criterion - HQC, etc.). For the purpose of LSTM modeling, a custom loss function (MADL) was created as the network performance metric and is used during the training process (Michańków et al. (2022)). Buy and sell signals that we use for single investment strategies are based on 1-period ahead forecasts of daily returns. Strategy performance metrics (aRC, ASD, MDD, IR, IR\*, IR\*\*, nObs, nTrades) are calculated using the equity line constructed for each algorithmic investment strategy separately.

The results for the LSTM model were obtained using R (4.1.0) and Python (3.7.10) programming languages. Deep learning libraries used for design, training, and testing the network are Keras 2.5.0 and TensorFlow 2.7.0. The rest of the calculations, as well as graphs and tables, were done using R and RStudio environment. Computer specifications are as follows: AMD Ryzen 7 3700X 3,6GHz, 16GB RAM, NVIDIA

GeForce RTX 2060 Super with 270 tensor cores. One full training (number of iterations  $\times$  300 epochs) lasted around 30 minutes for daily data and around 4-8 hours for hourly data.

### 3.2. ARIMA-GARCH model

We use the combination of ARIMA( $p,d,q$ ) and GARCH( $r,s$ ) models (Tsay (2010)). The ARIMA-GARCH model can be regarded as an extension of the ARMA model which is the combination of the autoregressive AR( $p$ ) and moving average MA( $q$ ) models for stationary time series.

The ARIMA( $p,d,q$ ) process can be written as:

$$\left(1 - \sum_{i=1}^p \phi_i L^i\right) (1 - L)^d y_t = c + \left(1 - \sum_{j=1}^q \theta_j L^j\right) \varepsilon_t \quad (1)$$

where:

- $p$  - is the order of autoregressive terms (AR),
- $\phi_i$  - are coefficients of the autoregressive terms,
- $L$  - is the lag operator, which produces the previous element of the series, eg.  $Ly_t = y_{t-1}$ ,
- $d$  - is the integration order of  $y_t$ ,
- $c$  - is the constant term,
- $q$  - is the order of moving-average terms (MA),
- $\theta_j$  - are the coefficients of the moving-average terms,
- $\varepsilon_t$  - is the IID error term.

In this study, log-returns of assets are described by the ARIMA( $p,0,q$ )-GARCH(1,1) model which is given by:

$$r_t = \mu + \sum_{i=1}^p \phi_i r_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t \quad (2)$$

$$\varepsilon_t = \sqrt{h_t} z_t, \quad z_t \stackrel{\text{IID}}{\sim} N(0, 1) \quad (3)$$

$$h_t = \omega + \alpha \varepsilon_{t-1}^2 + \beta h_{t-1} \quad (4)$$

where  $\mu, \omega, \alpha, \beta$  are parameters,  $z_t$  is the IID error term, and  $h_t$  is the conditional variance function.

We use the following estimation process in order to prepare forecasts based on ARIMA( $p,0,q$ )-GARCH(1,1) model: the parameters of the model are re-estimated every day; ARMA( $p,q$ ) orders are re-optimized every quarter with AIC, SBC, and HQC ( $p_{\max} = 5, q_{\max} = 5$ ); AIC is used for the base case scenario; when the estimation of the model was not possible we use the last available model.

### 3.3. Contrarian and momentum strategies

#### 3.3.1. Contrarian approach

It is one of the simplest investment strategies (Park and Sabourian (2011), Dobrynskaya (2019), Kadoya et al. (2008), and Carta et al. (2022)) assuming a strong mean-reverting process in the analyzed time series, which implies that our next day return forecast is exactly opposite to the previous day's return:

$$\text{Buy}_{\text{signal}} \quad \text{on } P_t \quad \text{if } r_t < 0 \quad (5)$$

$$\text{Sell}_{\text{signal}} \quad \text{on } P_t \quad \text{if } r_t \geq 0 \quad (6)$$

where  $P_t$  is the price at time  $t$ .

3.3.2. Momentum approach

Momentum strategy (Jegadeesh and Titman (2011), Chu et al. (2020), Flori and Regoli (2021), Ong and Herremans (2023), and Pal and Singh (2023)) assumes that financial returns tend to be persistent, which implies that our next-day return forecast is exactly the same with regard to the sign to the previous day's return:

$$\text{Buy}_{\text{signal}} \text{ on } P_t \text{ if } r_t \geq 0 \tag{7}$$

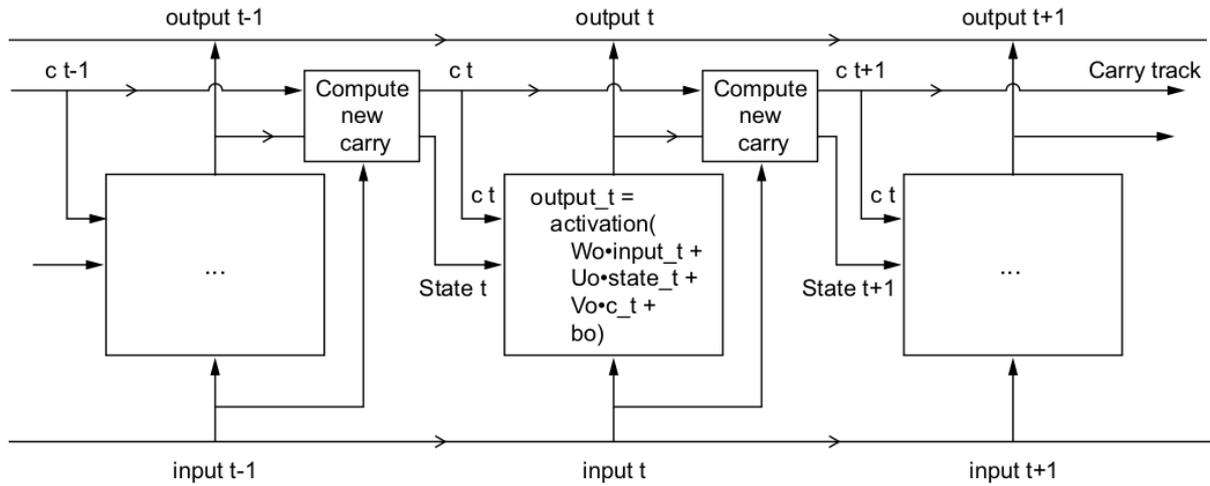
$$\text{Sell}_{\text{signal}} \text{ on } P_t \text{ if } r_t < 0 \tag{8}$$

In the case of contrarian and momentum signals, their values are based on the return from the previous day.

3.4. LSTM model

3.4.1. Architecture of LSTM

LSTM networks (Figure 1) are a type of recurrent neural networks (RNNs) that can keep track of long-term dependencies in data, allowing for partial solving of vanishing gradient problems typical for classic RNNs. It's widely used to model sequential data such as text, speech, and time series. LSTM units are composed of memory cells, with each cell having three types of gates (input gate, output gate and forget gate). These gates use "tanh" and "sigmoid" functions to regulate the flow of information through the cell, deciding how much and which information should be stored in a long-term state, passed on to another step, or discarded.



Note: LSTM cells presented in this figure show the information flow between the main LSTM gates: input, output, and forget. Source: Chollet (2021)

Figure 1: Anatomy of the LSTM model

The LSTM adds a way to carry information ( $c_t$ ) across many timesteps and hence preventing older signals from gradually vanishing during processing. The information  $c_t$  is combined with the input connection and the recurrent connection:

$$\text{output}_t = f(\text{state}_t \bullet U_o + \text{input}_t \bullet W_o + c_t \bullet V_o + b_o) \tag{9}$$

The new value of  $c_{t+1}$  is then calculated as:

$$c_{t+1} = i_t \cdot k_t + c_t \cdot f_t \tag{10}$$

where:

$$i_t = f(\text{state}_t \bullet U_i + \text{input}_t \bullet W_i + b_i) \tag{11}$$

$$f_t = f(\text{state}_t \bullet U_f + \text{input}_t \bullet W_f + b_f) \quad (12)$$

$$k_t = f(\text{state}_t \bullet U_k + \text{input}_t \bullet W_k + b_k) \quad (13)$$

and  $U_i, W_i, b_i, U_f, W_f, b_f, U_k, W_k, b_k$  are matrices with weights,  $f(\cdot)$  is the activation function and  $\bullet$  is the dot product of two vectors.

Our LSTM model consists of three LSTM layers with 512/256/128 neurons and one single neuron dense layer on the output. Each of the LSTM layers is using *tanh* activation function (to retain negative values). L2 regularization (0.000001) and dropout (0.001) are also applied to each of these layers. The first two layers return sequences with the same shape as the input sequence (full sequence), and the last LSTM layer returns only the last output.

To train the model we use the Adam optimizer - a stochastic gradient descent optimizer with momentum (estimating first-order and second-order moments). The learning rate of the optimizer is set to 0.5 (after tuning).

### 3.4.2. Data selection, hyperparameters tuning and LSTM training

We focus primarily on logarithmic returns, using daily data for S&P500, bitcoin (BTC), gold (GLD), natural gas (UNG), and wheat (ZWF) from 2004-01-02<sup>1</sup> and 2022-03-29. We also use hourly data for SPX and UNG, from the same period. Hourly data availability is restricted for extensive time periods, so proprietary data was used in this case. However, daily data for all tested assets is readily accessible.

For the training set, we use an expanding window approach, with the size of the first window set to 252 trading days (one year). The size of the validation set is 33% of the training set. The test set size is always 252 days. The input sequence size for the LSTM network is set to 10. We use the ReLU activation function on the last neuron to obtain only zero or positive values (for Long Only strategies) or inverted ReLU to obtain zero or negative values (for Short Only strategy used for UNG). The output of the model is a single number predicting the next return value. Based on the sign of the predicted return value we assign -1, 0, and 1 signals, depending on the strategy.

During our research, we conduct detailed hyperparameter tuning to ensure the best possible results from our model. The hyperparameters we test are:

- number of layers (1-5) and neurons in each layer (5-512),
- dropout rate (0 - 0.5) and l2 kernel regularization (0 - 0.01),
- the type of optimizer (SGD, RMSProp, and Adam variants),
- learning rate (0.0001 - 0.9) and momentum values (0.1-0.9),
- training and testing window sizes, sequence length, and batch size,
- number of epochs (10-300) and callbacks (early stopping and model checkpoint).

Table 1: Values of hyperparameters selected after network tuning.

Hyperparameter	Selected Value
No. hidden layers	3
No neurons	512/256/128
Activation function	<i>tanh</i>
Dropout rate	0
l2 regularizer	1e-6
Optimizer	Adam
Learning rate	0.5
Train/test size	252-exp. window/252
Batch size	exp. window
Sequence length	10

Note: Hyperparameters used in this study for the LSTM model.

<sup>1</sup>In practice, the starting point of data depends on the asset and varies between 2004-01-02 and 2010-07-17.

In addition, we change the following hyperparameters of the network to optimize it for high-frequency data: train and test sizes are increased to cover one calendar year of data, an additional layer with 252 neurons is added and the number of epochs is changed to 120.

For training and prediction, we use a walk-forward validation/expanding window approach. In the first iteration, the model is trained on one year of data (equal to the train set length) and then used for predictions over the next year (equal to the test set length). After that, the window is expanded by another year of data and the model was retrained. A single return value is predicted each time, using the last 10 (sequence length) values. A single iteration is trained for 300 epochs. The model checkpoint callback function is used to store the best weights (parameters) of the model based on the lowest loss function value in a specific epoch. The weights are then used for prediction.

### 3.4.3. Loss function for LSTM model

We use the loss function proposed by Michańków et al. (2022), who appropriately evaluates the usefulness of the forecasting ability of the LSTM model in algorithmic investment strategies (AIS). The RMSE, MSE, MAE, MAPE, and %OP used in 99.9% of similar research are not the proper error functions for the evaluation of the forecasting ability of the given model in the AIS, mainly because they evaluate the point forecast. These error metrics evaluate only the accuracy of forecasts, which is often confused with the forecasting ability of the given model in AIS.

$$\text{MADL} = \frac{1}{N} \sum_{i=1}^N (-1) \times \text{sign}(R_i \times \hat{R}_i) \times \text{abs}(R_i) \quad (14)$$

where:

- MADL - the Mean Absolute Directional Loss function,
- $R_i$  is the observed return on interval  $i$ ,
- $\hat{R}_i$  is the predicted return on interval  $i$ ,
- $\text{sign}(X)$  is the function which gives the sign of  $X$ ,
- $\text{abs}(X)$  is the function which gives the absolute value of  $X$
- $N$  is the number of forecasts.

This way, the value of the loss function (MADL) is equal to the observed return on the investment with the predicted sign. This allows the model to inform us if its prediction will yield profit or loss and how much this profit or loss will be. MADL was designed specifically for working with AIS's instead of just verification of forecasts in point. The function in our model is minimized, so that if it gives negative values, the strategy will make a profit, and if it gives positive values, the strategy will generate a loss.

### 3.5. Ensemble models

In order to address the research questions we had to create two types of ensemble models:

- type I - built with various theoretical models for the selected type of asset
- type II - built with various types of assets for the selected theoretical model

Therefore, type I ensemble models for a given asset  $j$ , were created according to the following formula:

$$\text{EQline}_j^{(I)} = \frac{1}{n} \sum_{i=1}^n \text{EQline}_{i,j} \quad (15)$$

where:

$n$  - the number of theoretical models,  $i = \{1, \dots, n\}$

$\text{EQline}_j^{(II)}$  - the value of the ensemble equity line on day  $t$  for algorithmic investment strategy on the  $j$ -th asset (S&P 500 index, Bitcoin, Gold, Natural Gas, and Wheat) for all theoretical models (LSTM, ARIMA-GARCH, Momentum, and Contrarian models),

$EQline_{i,j}$  - the value of the single equity line on the day  $t$  for algorithmic investment strategy on the  $j$ -th asset (S&P 500 index, Bitcoin, Gold, Natural Gas, and Wheat) for the  $i$ -th theoretical model (LSTM, ARIMA-GARCH, Momentum, and Contrarian),

On the other hand, type II ensemble models for a given theoretical model  $i$ , were created according to the following formula:

$$EQline_i^{(II)} = \frac{1}{m} \sum_{j=1}^m EQline_{i,j} \quad (16)$$

where:

$m$  - the number of assets,  $j = \{1, \dots, m\}$

$EQline_i^{(I)}$  - the value of the ensemble equity line on day  $t$  for algorithmic investment strategy on all assets (S&P 500 index, Bitcoin, Gold, Natural Gas, and Wheat) for one of the  $i$ -th theoretical model (LSTM, ARIMA-GARCH, Momentum, and Contrarian).

### 3.6. Performance metrics

Based on Kijewski and Ślepaczuk (2020) the following performance metrics were calculated:

- annualized return compounded (aRC):

$$aRC = \prod_{i=1}^n (r_i + 1)^{252/n} - 1 \quad (17)$$

where  $r_i$  is the daily percentage return at time  $i$  and  $n$  is the number of trading days,

- annualized standard deviation (aSD):

$$aSD = \frac{\sqrt{252}}{n-1} \sum_{i=1}^n (r_i - \bar{r})^2 \quad (18)$$

where  $\bar{r}$  is the average daily percentage return,

- Information Ratio\* (IR\*):

$$IR^* = \frac{aRC}{aSD} \quad (19)$$

- Maximum Drawdown (MD):

$$MD = \sup_{0 \leq t_1 \leq t_2 \leq t} \frac{EQ_{t_1} - EQ_{t_2}}{EQ_{t_1}} \quad (20)$$

where  $EQ_t$  is the equity line level at time  $t$ .

- Information Ratio\*\* (IR\*\*)

$$IR^{**} = \frac{aRC * aRC * \text{sign}(ARC)}{aSD * MD} \quad (21)$$

We regard the  $IR^{**}$  as the most important in the evaluation of our final results because this indicator combines the information from two crucial risk metrics: aSD and MD.

- Maximum Loss Duration (MLD): the longest time needed to surpass a maximum value of the strategy returns, measured in years.
- Information Ratio\*\*\* (IR\*\*\*)

$$IR^{***} = \frac{ARC * ARC * ARC}{aSD * MD * MLD} \quad (22)$$

- nObs - the number of observation
- nTrades - the number of trades, which is the number of all changes in position on the analyzed asset

### 3.7. Research description

The detailed research description performed in this research can be summarized as follows:

- tests for two versions of the investment strategies: Long Only and Short Only,
- a new Loss function: MADL, introduced by Michańków et al. (2022)
- hyperparameters tuning, according to details described in Section 3.4.2
- walk-forward optimization:
  - *in-sample*: estimation of the model parameters (LSTM) or optimization of  $p$  and  $q$  orders (ARIMA-GARCH)
    - \* in the *in-sample* period we use the last  $n \times 365$  actual days, where  $n = 1, 2, 3, 4, 5$ ; base case = 3Y,
    - \* in the *in-sample* period we include data for the last 1Y, 2Y, 3Y, 4Y and 5Y years, respectively,
  - *out-of-sample*: re-estimation and re-optimization of models and forecast generations
    - \* the first out-of-sample period starts one year after data start (for all five cases, i.e. 1Y, 2Y, 3Y, 4Y, and 5Y),
    - \* out-of-sample forecasts: 1 day ahead,
- buy/sell signals definitions based on the next day forecasts,
- equity lines and performance metrics according to Ślepaczuk et al. (2018) with DFL =1,
- verification of diversification potential of various asset classes and theoretical models for AIS built for the S&P500 index.
- the construction of ensemble investment strategies based on the combination of signals across different asset classes (S&P500, BTC, UNG, GLD, ZWF) and theoretical models (ARIMA-GARCH, contrarian, momentum, and LSTM).
- the last part is devoted to the sensitivity analysis performed for various data frequencies used in the LSTM model.

## 4. Results

We present results from less complex (single investment models) to more complex (ensemble investment models) while emphasizing their diversification potential. This sequence is not necessarily connected with the order of our research questions in the Introduction.

4.1. Base case scenario

In the first part of the results, we describe individual strategies and type I of the ensemble model where ensembling is defined as an equally weighted portfolio strategy, consisting of different models/strategies for a single asset. Rebalancing is performed on the first available day of Jan, Apr, Jul, and Oct.

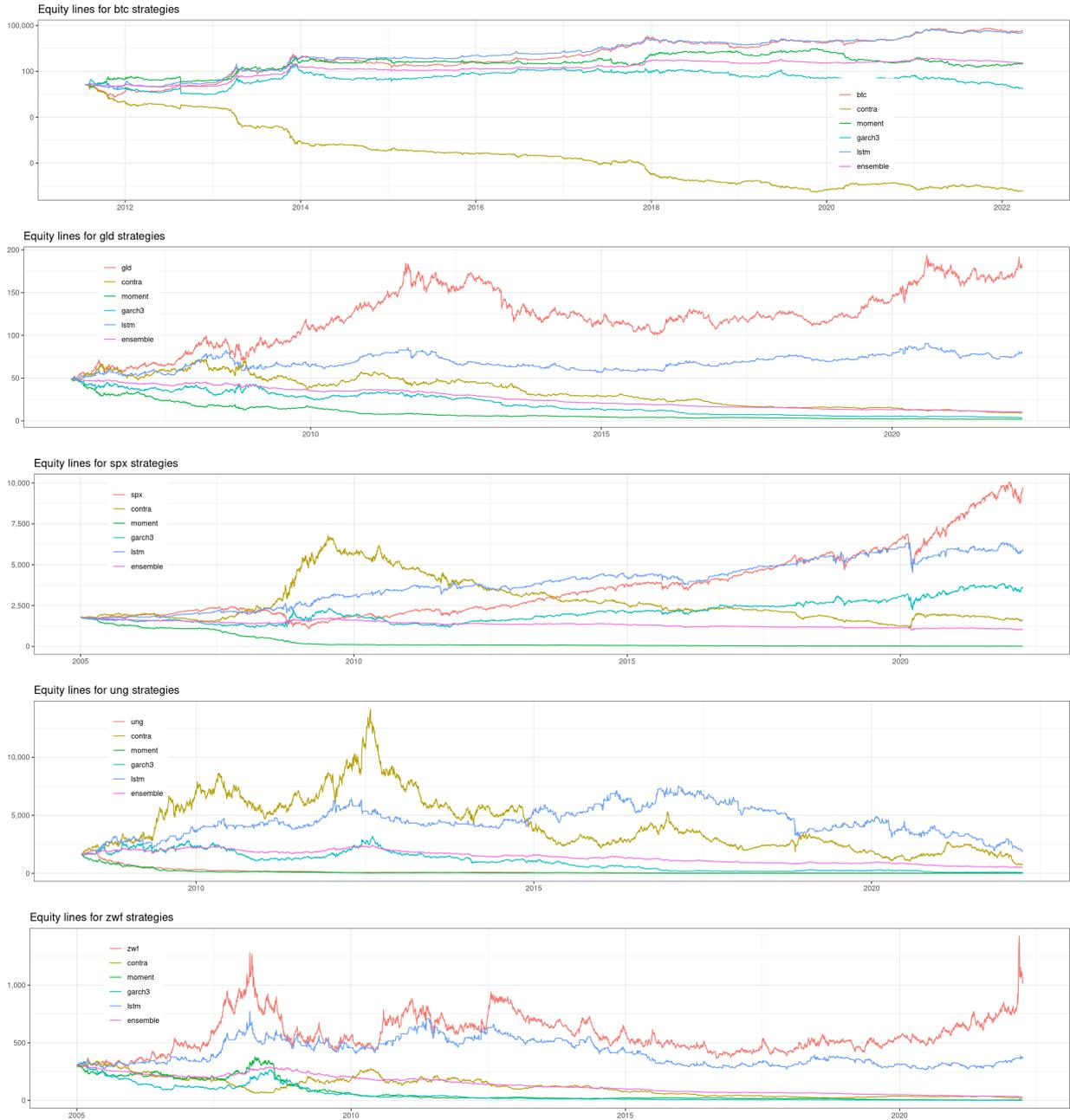
Table 2 shows the performance metrics for tested strategies (individual and ensemble - type I) and the benchmark Buy&Hold strategy. Based on these results we can notice that the LSTM model-based strategy is characterized by the highest IR (IR\*, IR\*\*, and IR\*\*\*) in most cases.

Table 2: Base case scenario results for individual and ensemble strategies, for a single asset

	aRC	aSD	MD	MLD	IR*	IR**	IR***	nObs	nTrades
<b>BTC</b>									
B&H	<b>114.78</b>	88.29	86.67	3.24	1.30	1.722	0.610	3909	2
contra	-77.59	87.78	100.00	10.67	-0.88	-0.686	-0.050	3909	1980
moment	34.66	88.97	94.09	4.01	0.39	0.144	0.012	3909	1984
garch3	-5.10	88.58	97.41	8.30	-0.06	-0.003	0.000	3909	1351
lstm	109.32	64.76	67.19	2.89	<b>1.69</b>	<b>2.747</b>	<b>1.040</b>	3909	525
ensemble	36.57	44.83	68.13	3.99	0.82	0.438	0.040	3909	6012
<b>GLD</b>									
B&H	<b>8.33</b>	18.28	45.56	8.92	<b>0.46</b>	<b>0.083</b>	0.001	4117	2
contra	-9.68	17.74	87.46	14.05	-0.55	-0.060	0.000	4117	2133
moment	-17.83	18.85	96.20	16.21	-0.95	-0.175	-0.002	4117	2143
garch3	-14.29	18.28	92.88	16.27	-0.78	-0.120	-0.001	4117	1593
lstm	3.03	12.63	34.14	8.53	0.24	0.021	0.000	4117	720
ensemble	-8.95	6.10	78.95	16.27	-1.47	-0.166	-0.001	4117	6853
<b>SPX</b>									
B&H	<b>10.35</b>	19.48	55.25	4.48	<b>0.53</b>	0.100	0.002	4340	2
contra	-0.75	18.99	83.62	12.70	-0.04	0.000	0.000	4340	2269
moment	-25.37	19.99	99.40	17.21	-1.27	-0.324	-0.005	4340	2269
garch3	4.24	19.33	49.95	6.96	0.22	0.019	0.000	4340	1274
lstm	7.23	14.92	28.43	1.99	0.48	<b>0.123</b>	<b>0.004</b>	4340	722
ensemble	-3.03	7.45	44.52	17.22	-0.41	-0.028	0.000	4340	6810
<b>UNG</b>									
B&H	-27.52	44.44	99.58	13.73	-0.62	-0.171	-0.003	3512	2
contra	-5.47	43.86	94.69	9.64	-0.12	-0.007	0.000	3512	1803
moment	-30.88	45.05	99.70	13.93	-0.69	-0.212	-0.005	3512	1797
garch3	-19.90	44.40	97.75	9.61	-0.45	-0.091	-0.002	3512	1416
lstm	<b>1.09</b>	31.15	74.79	5.10	<b>0.04</b>	<b>0.001</b>	0.000	3512	612
ensemble	-8.05	14.98	79.28	9.61	-0.54	-0.055	0.000	3512	5852
<b>ZWF</b>									
B&H	<b>7.24</b>	33.22	71.80	14.01	<b>0.22</b>	<b>0.022</b>	0.000	4362	2
contra	-14.61	32.65	95.09	16.94	-0.45	-0.069	-0.001	4362	2240
moment	-21.97	33.82	99.17	13.99	-0.65	-0.144	-0.002	4362	2264
garch3	-29.25	33.21	99.78	17.18	-0.88	-0.258	-0.004	4362	1818
lstm	1.04	22.98	65.99	14.08	0.05	0.001	0.000	4362	778
ensemble	-12.68	10.61	90.69	17.18	-1.20	-0.167	-0.001	4362	7376

Note: Results cover the performance metrics for 4 individual strategies and 1 ensemble model for 5 various assets (BTC, GLD, SPX, UNG, and ZWF). The ensemble model stands for the combination of all theoretical models for the given asset.

Figure 2 presents equity lines for every investment strategy and confirms the results described in Table 2.



Note: Each panel presents five equity lines for each tested asset (BTC, GLD, SPX, UNG, and ZWF). These equity lines represent the results for 4 individual strategies based on the model/concept of LSTM, ARIMA-GARCH, momentum, and contrarian, and one additional equity line for the ensemble model built using these four above-mentioned.

Figure 2: Equity lines for individual and ensemble strategies for single assets

Table 3 contains the performance metrics for all types of ensemble models (type I - ensemble model combining all tested strategies for the given assets: SPX\_all, BTC\_all, GLD\_all, UNG\_all, ZWF\_all, and type II - ensemble model combining all assets for the given tested strategy: contr\_all, momentall, garch3\_all, lstm\_all) and compare it with Buy&Hold strategy for all 5 assets (5\_assets). The important conclusion from this table is that lstm\_all outperforms other strategies and Buy&Hold and that BTC\_all outperforms other

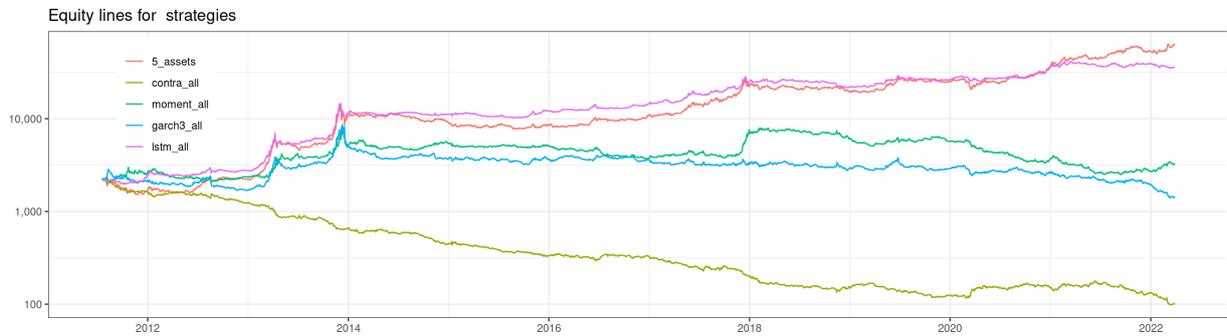
assets. The former could be attributed to the distinctive architecture of LSTM networks, which provides them with the capability to more effectively capture intricate temporal patterns within the data, while the latter to the availability and continuity of BTC data, which is quoted 24/7.

Table 3: Ensemble strategies for single assets and theoretical models

	aRC	aSD	MD	MLD	IR*	IR**	IR***	nObs	nTrades
<b>B&amp;H all assets</b>									
B&H_all	<b>23.947</b>	23.564	44.433	5.024	1.016	0.548	0.026	3908	10
<b>ensembles for single assets</b>									
SPX_all	-3.035	7.454	44.519	17.218	-0.407	-0.028	0.000	4340	6810
BTC_all	36.568	44.834	68.130	3.989	0.816	0.438	0.040	3909	6012
GLD_all	-8.949	6.099	78.950	16.274	-1.467	-0.166	-0.001	4117	6853
UNG_all	-8.053	14.976	79.276	9.611	-0.538	-0.055	0.000	3512	5852
ZWF_all	-12.684	10.608	90.690	17.179	-1.196	-0.167	-0.001	4362	7376
<b>models for all assets</b>									
contra_all	-18.166	15.448	95.657	15.472	-1.176	-0.223	-0.003	3908	10728
moment_all	2.567	22.708	68.705	5.980	0.113	0.004	0.000	3908	10760
garch3_all	-2.862	23.202	83.842	12.028	-0.123	-0.004	0.000	3908	7755
lstm_all	19.674	18.072	30.274	4.460	<b>1.089</b>	<b>0.707</b>	<b>0.031</b>	3908	3660

Note: Each panel presents performance metrics for the Buy&Hold strategy for all assets (5\_assets), for ensemble models for single assets (SPX\_all, BTC\_all, GLD\_all, UNG\_all, ZWF\_all), ensemble models for theoretical concepts (contra\_all, moment\_all, garch3\_all, lstm\_all).

Figure 3 visualizes fluctuations of equity lines for ensemble models and Buy&Hold and confirms the high performance of the LSTM model-based strategy.



Note: 5\_assets stands for Buy&Hold strategy for all assets. Contra\_all, moment\_all, garch3\_all, lstm\_all stand for ensemble models for all assets within one theoretical concept.

Figure 3: Ensemble strategies for all theoretical models

Table 4 contains a summary of the first part of the research which enables us to refer to our research questions.

Table 4: Ensemble strategies for all assets within one theoretical model.

IR**	BTC	GLD	SPX	UNG	ZWF	positive	beat	winner
						IR**	B&H?	
B&H	1.722	<b>0.083</b>	0.100	-0.171	0.022	80%	0%	40%
contrarian	-0.686	-0.060	0.000	-0.007	-0.069	0%	20%	0%
momentum	0.144	-0.175	-0.324	-0.212	-0.144	20%	0%	0%
garch3	-0.003	-0.120	0.019	-0.091	-0.258	20%	20%	0%
lstm	<b>2.747</b>	0.021	<b>0.123</b>	<b>0.001</b>	0.001	100%	60%	60%
ensemble	0.438	-0.166	-0.028	-0.055	-0.167	20%	20%	0%

Note: B&H stands for Buy&Hold strategy for all assets. Contrarian, momentum, garch3, lstm stand for ensemble models for all assets within one theoretical concept. The ensemble stands for ensemble model for all assets and all theoretical models.

Based on the results for the base case scenario, presented in Tables 2, 3, and 4 and Figure 2, and 3 we can refer to RQ2 and RQ3. Referring to RQ2, we can confirm that ML models are more efficient than classical models. In the case of single investment strategies because LSTM was the best strategy in 60% of the cases (3 out of 5 asset classes tested). Moreover, in the case of complex investment strategies (type II) based on the aggregation of all assets for a single theoretical model lstm\_all was the best strategy in comparison to contrarian\_all, momentum\_all, and garch\_all.

Regarding RQ3, the ensemble AIS based on the aggregation of all theoretical models for the single asset (type I) were never better than the LSTM model or the B&H strategy. Moreover, none of the ensemble strategies based on the aggregation of all assets for the single theoretical model (type II): lstm\_all, contrarian\_all, momentum\_all, garch\_all, and B&H\_all were better than the single strategies for the given class of asset.

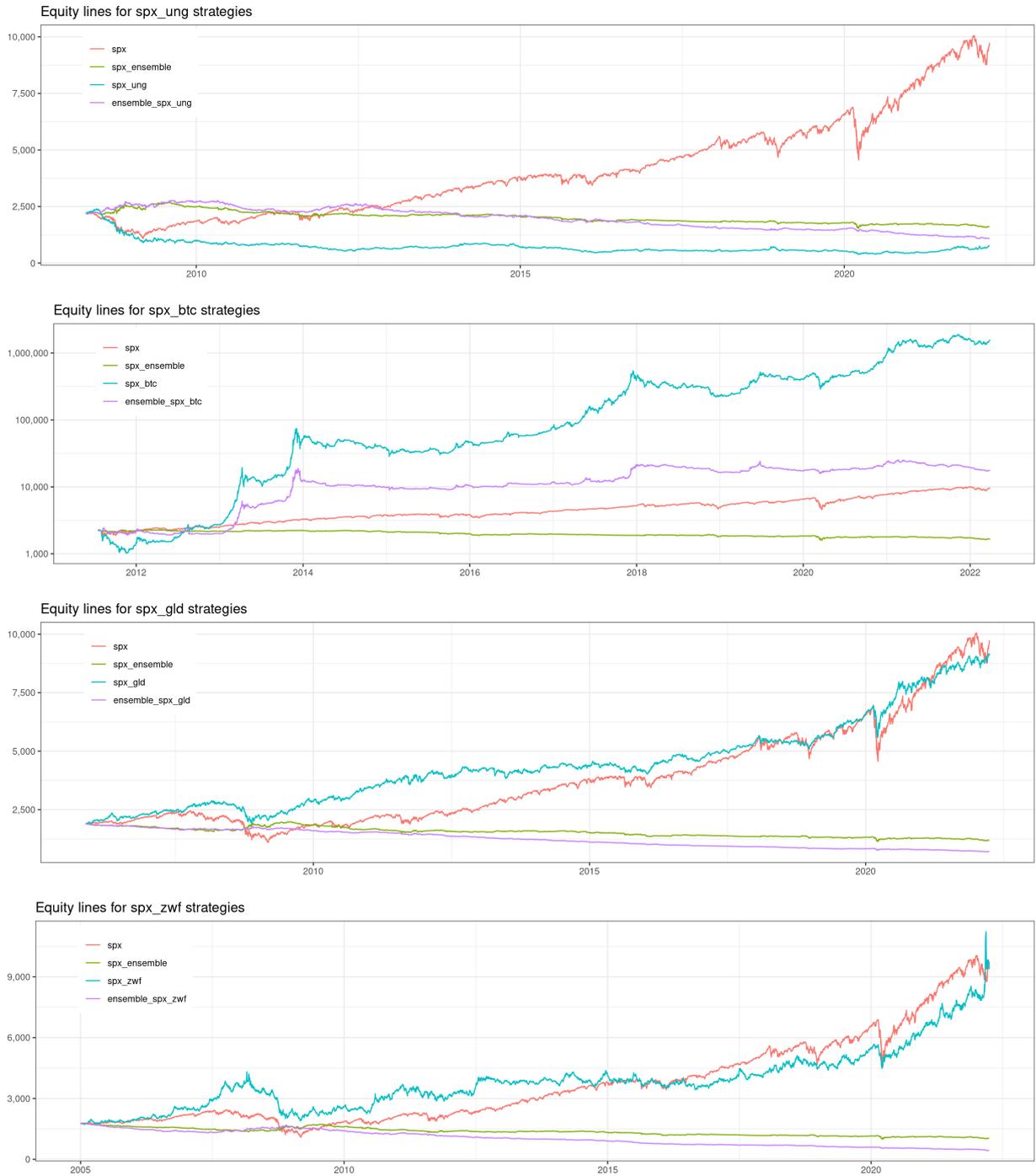
#### 4.2. Base Case Scenario. Ensemble models based on two assets - diversification potential.

Based on the results presented in Table 5 and Figure 4 for the ensemble models of two assets and their diversification potential with regard to strategies based on SPX, we can refer to the RQ1. Looking at the IR\*\* measure, we can state that the only diversification potential can be noticed after adding the ensemble model based on BTC to the ensemble model based on SPX, where the IR\*\* for the ensemble\_spx\_btc increases.

Table 5: Diversification potential of investment models for hedging equity index investment model

	aRC	aSD	MD	MLD	IR*	IR**	IR***	nObs	nTrades
<b>UNG</b>									
spx	<b>11.27</b>	20.59	51.52	2.75	<b>0.55</b>	<b>0.120</b>	0.005	3512	2
spx_ensemble	-2.14	7.90	42.82	12.69	-0.27	-0.014	0.000	3512	5357
spx_ung	-7.39	25.30	84.11	13.77	-0.29	-0.026	0.000	3512	224
ensemble_spx_ung	-4.91	8.25	61.09	12.59	-0.59	-0.048	0.000	3512	11209
<b>BTC</b>									
spx	10.01	14.44	33.79	1.08	0.69	0.205	0.019	3908	2
spx_ensemble	-1.76	5.64	30.19	14.37	-0.31	-0.018	0.000	3908	4003
spx_btc	<b>52.72</b>	42.33	61.87	4.42	<b>1.25</b>	<b>1.061</b>	<b>0.127</b>	3908	172
ensemble_spx_btc	14.36	23.01	52.71	5.82	0.62	0.170	0.004	3908	10012
<b>GLD</b>									
spx	<b>10.58</b>	19.85	55.25	4.48	0.53	0.102	0.002	4117	2
spx_ensemble	-2.72	7.61	42.82	12.69	-0.36	-0.023	0.000	4117	6417
spx_gld	10.17	13.58	34.05	1.72	<b>0.75</b>	<b>0.224</b>	<b>0.013</b>	4117	264
ensemble_spx_gld	-5.80	5.00	63.21	16.31	-1.16	-0.107	0.000	4117	13270
<b>ZWF</b>									
spx	<b>10.30</b>	19.43	55.25	4.48	<b>0.53</b>	<b>0.099</b>	<b>0.002</b>	4362	2
spx_ensemble	-3.02	7.44	44.52	17.31	-0.41	-0.028	0.000	4362	6802
spx_zwf	10.06	20.45	56.04	6.81	0.49	0.088	0.001	4362	276
ensemble_spx_zwf	-7.80	6.54	75.55	17.18	-1.19	-0.123	-0.001	4362	14178

Note: Each of the 4 panels contains the results for 4 strategies: SPX - B&H for S&P 500 index, spx\_ensemble - the ensemble models combining all theoretical models for S&P 500 index, spx\_asset - combined B&H for S&P 500 index and the given asset, ensemble\_spx\_asset - the combination of two ensemble models built for all theoretical models for SPX and the given asset.



Note: Each panel presents the equity lines for 4 different strategies: SPX - B&H for the S&P 500 index, spx\_ensemble - the ensemble models combining all theoretical models for the S&P 500 index, spx\_asset - combined B&H for S&P 500 index and the given asset, ensemble\_spx\_asset - the combination of two ensemble models built for all theoretical models for SPX and the given asset.

Figure 4: Equity lines for hedging strategies for equity index

4.3. Daily versus hourly results for selected assets

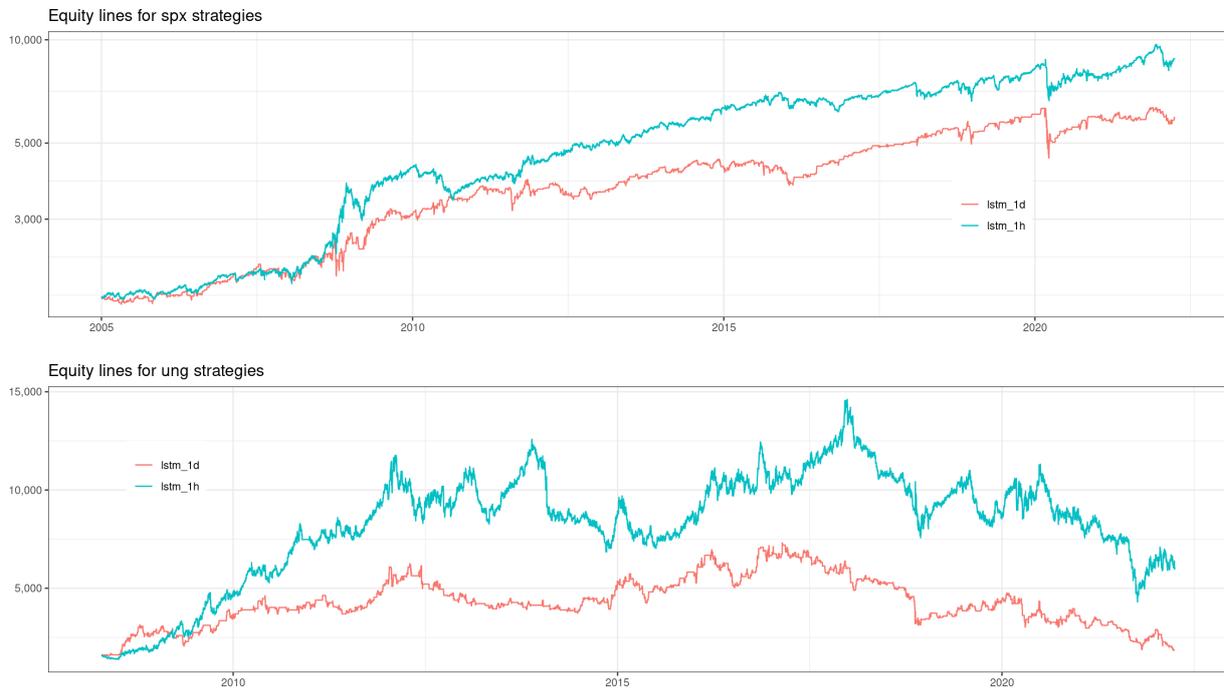
In order to answer RQ4, we repeat training and estimation of the LSTM model for SPX and UNG assets on hourly data in the same period as for the daily data, i.e. from 2008-04-17 to 2022-03-29. The selection of these two assets was dictated by the following reasons. The S&P 500 index was chosen for its wide usage in financial literature, ensuring comparability with other research. UNG represents a distinct dynamic with decreasing asset prices over time and potential diversification benefits during geopolitical stress, such as the Russian-Ukrainian conflict.

Table 6: LSTM model results for S&P 500 index and UNG on daily and hourly data

	aRC	aSD	MD	MLD	IR*	IR**	IR***	nObs	nTrades
<b>S&amp;P 500</b>									
lstm_1d	7.23	14.92	28.43	1.99	0.48	0.123	0.004	4340	722
lstm_1h	<b>9.72</b>	12.34	24.25	1.72	<b>0.79</b>	<b>0.315</b>	<b>0.018</b>	34702	5364
<b>UNG</b>									
lstm_1d	1.09	31.15	74.79	5.1	0.04	0.001	0.000	3512	612
lstm_1h	<b>6.80</b>	24.38	70.54	6.17	<b>0.28</b>	<b>0.027</b>	0.000	122318	16310

Note: lstm\_1d stands for LSTM model-based investment strategy trained and estimated for the S&P 500 index (first panel) and UNG on daily data. lstm\_1h denotes the same strategy test on 1h data.

Table 6 and Figure 5 shows that in each tested case LSTM models on hourly data outperform the ones on daily data in each case of risk-adjusted measures (IR\*, IR\*\*, and IR\*\*\*).



Note: Each panel presents equity lines for SPX and UNG for two different frequencies daily and hourly.

Figure 5: Equity lines for LSTM model results for S&P 500 index and UNG on daily versus hourly data

## 5. Conclusions

The novelty and the main contribution of this paper is an attempt to focus on the problem of diversification from a different perspective than what is usually presented in state-of-the-art research. Based on the results for five different assets (BTC, GLD, SPX, UNG, and ZWF), in the period from 2007 to 2022, we verified a few different research questions focusing on individual and ensemble algorithmic investment strategies using various types of theoretical models. The ensemble process used in this research for the first time focused on 3 different surfaces of single strategies combination, i.e. based on 1) various types of assets, 2) various theoretical models, and 3) a combination of both of them.

We verify the diversification potential of investment strategies for the equity index (S&P 500 index) based on various theoretical concepts against other investment strategies (RQ1). Therefore, referring to RQ1: *Which of the tested groups of assets (energy commodities, cryptocurrencies, gold, or soft commodities) have the largest diversification potential in the complex algorithmic investment strategies, built with machine learning models and ARIMA-GARCH models for equity indices?*, based on the results presented in Table 5 and Figure 4, we can state that only ensemble\_BTC has the diversification potential that increases the efficiency of ensemble models for the equity index. Moreover, taking into account that the distribution of returns for other equity indices is quite similar to that of the S/P 500 we are sure that our conclusions can be extended to them, as well.

Based on the results presented in Table 4 and Figure 3 we can affirmatively address RQ2: *Are machine learning techniques more efficient than ARIMA-GARCH models and the concepts of momentum and contrarian in the case of single and complex (ensemble model combining all tested strategies for the given assets - type I) investment strategies*

After analyzing the results presented in Table 2 and Table 4 we can assert an unfavorable response to RQ3: *Are complex (ensemble) AIS based on the aggregation of all theoretical models for the single asset (type I) or all assets for a single theoretical model (type II) more efficient than individual strategies.*

Finally, based on the results presented in (Table 5 and Figure4) we can provide a positive response to RQ4: *Are results for LSTM models on higher frequencies of data (1h) better than those on daily data.*

Further research extensions of this work should focus on the following: extensive sensitivity analysis with a special focus on alternative loss functions, a larger set of alternative assets and models, different theoretical models in the process of generating buy/sell signals, more careful hyperparameters tuning process, and finally more advanced procedure of selection parameters and hyperparameters in the in-sample period.

## References

- I. Baranochnikov and R. Ślepaczuk. A comparison of lstm and gru architectures with novel walk-forward approach to algorithmic investment strategy. *Working Papers of Faculty of Economic Sciences, University of Warsaw*, WP 21/2022(397), 2022.
- S. Carta, S. Consoli, A. S. Podda, D. R. Recupero, and M. M. Stanciu. Statistical arbitrage powered by explainable artificial intelligence. *Expert Systems with Applications*, 206: 117763, 2022. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2022.117763>. URL <https://www.sciencedirect.com/science/article/pii/S0957417422010405>.
- S. Castellano Gomez and R. Ślepaczuk. Robust optimisation in algorithmic investment strategies. *Working Papers of Faculty of Economic Sciences, University of Warsaw*, WP 27/2021(375), 2021.
- K. Chen, Y. Zhou, and F. Dai. A lstm-based method for stock returns prediction: A case study of china stock market. In *2015 IEEE international conference on big data (big data)*, pages 2823–2824. IEEE, 2015.
- F. Chollet. *Deep Learning with Python, 2nd ed.* Manning Publications Co., 2021.
- J. Chu, S. Chan, and Y. Zhang. High frequency momentum trading with cryptocurrencies. *Research in international business and finance*, 52:101176, 2020.
- J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *arXiv:1412.3555 [cs]*, Dec. 2014. URL <http://arxiv.org/abs/1412.3555>. arXiv: 1412.3555.
- L. Di Persio and O. Honchar. Recurrent neural networks approach to the financial forecast of google assets. *International journal of Mathematics and Computers in simulation*, 11:7–13, 2017.
- V. Dobrynska. Avoiding momentum crashes: Dynamic momentum and contrarian trading. *Journal of International Financial Markets, Institutions and Money*, 63:101141, 2019.
- T. Fischer and C. Krauss. Deep learning with long short-term memory networks for financial market predictions. *European Journal of Operational Research*, 270(2):654–669, 2018. ISSN 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2017.11.054>. URL <https://www.sciencedirect.com/science/article/pii/S0377221717310652>.

- A. Flori and D. Regoli. Revealing pairs-trading opportunities with long short-term memory networks. *European Journal of Operational Research*, 295(2):772–791, 2021. ISSN 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2021.03.009>. URL <https://www.sciencedirect.com/science/article/pii/S0377221721001995>.
- F. A. Gers, J. Schmidhuber, and F. Cummins. Learning to Forget: Continual Prediction with LSTM. *Neural Computation*, 12(10):2451–2471, Oct. 2000. ISSN 0899-7667. doi: 10.1162/089976600300015015. URL <https://doi.org/10.1162/089976600300015015>.
- P. Ghosh, A. Neufeld, and J. K. Sahoo. Forecasting directional movements of stock prices for intraday trading using lstm and random forests. *Finance Research Letters*, 46:102280, 2022. ISSN 1544-6123. doi: <https://doi.org/10.1016/j.flr.2021.102280>. URL <https://www.sciencedirect.com/science/article/pii/S1544612321003202>.
- J. Grudniewicz and R. Ślepaczuk. Application of machine learning in algorithmic investment strategies on global stock markets. *Research in International Business and Finance*, 66:102052, 2023. ISSN 0275-5319. doi: <https://doi.org/10.1016/j.ribaf.2023.102052>. URL <https://www.sciencedirect.com/science/article/pii/S0275531923001782>.
- S. Hochreiter and J. Schmidhuber. Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780, Nov. 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL <https://doi.org/10.1162/neco.1997.9.8.1735>.
- M. A. Hossain, R. Karim, R. K. Thulasiram, N. D. B. Bruce, and Y. Wang. Hybrid deep learning model for stock price prediction. *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 1837–1844, 2018.
- N. Jegadeesh and S. Titman. Momentum. *Annu. Rev. Financ. Econ.*, 3(1):493–509, 2011.
- S. Kadoya, T. Kuroko, and T. Namatame. Contrarian investment strategy with data envelopment analysis concept. *European Journal of Operational Research*, 189(1):120–131, 2008. ISSN 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2007.05.033>. URL <https://www.sciencedirect.com/science/article/pii/S0377221707004730>.
- M. Kijewski and R. Ślepaczuk. Predicting prices of S&P 500 index using classical methods and recurrent neural networks. *Working Papers of Faculty of Economic Sciences, University of Warsaw*, WP 27/2020(333), 2020.
- S. Lahmiri and S. Bekiros. Intelligent forecasting with machine learning trading systems in chaotic intraday bitcoin market. *Chaos, Solitons & Fractals*, 133:109641, 2020. ISSN 0960-0779. doi: <https://doi.org/10.1016/j.chaos.2020.109641>. URL <https://www.sciencedirect.com/science/article/pii/S0960077920300400>.
- J. Michańków, P. Sakowski, and R. Ślepaczuk. LSTM in algorithmic investment strategies on BTC and S&P 500 index. *Sensors*, 22(3), 2022. ISSN 1424-8220. doi: 10.3390/s22030917. URL <https://www.mdpi.com/1424-8220/22/3/917>.
- J. Ong and D. Herremans. Constructing time-series momentum portfolios with deep multi-task learning. *Expert Systems with Applications*, 230:120587, 2023. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2023.120587>. URL <https://www.sciencedirect.com/science/article/pii/S0957417423010898>.
- A. Pal and K. P. Singh. Adamr-grus: Adaptive momentum-based regularized gru for hmer problems. *Applied Soft Computing*, 143:110457, 2023. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2023.110457>. URL <https://www.sciencedirect.com/science/article/pii/S1568494623004751>.
- A. Park and H. Sabourian. Herding and contrarian behavior in financial markets. *Econometrica*, 79(4):973–1026, 2011.
- D. Shah, W. Campbell, and F. H. Zulkernine. A comparative study of lstm and dnn for stock market forecasting. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 4148–4155. IEEE Computer Society, 2018.
- S. Siami-Namini, N. Tavakoli, and A. S. Namin. A comparison of arima and lstm in forecasting time series. In *2018 17th IEEE international conference on machine learning and applications (ICMLA)*, pages 1394–1401. IEEE, 2018.
- R. S. Tsay. *Analysis of Financial Time Series*. John Wiley & Sons, 2010.
- N. Vo and R. Ślepaczuk. Applying hybrid arima-sgarch in algorithmic investment strategies on S&P 500 index. *Entropy*, 24(2), 2022. ISSN 1099-4300. doi: 10.3390/e24020158. URL <https://www.mdpi.com/1099-4300/24/2/158>.
- X. Zhang, X. Liang, A. Li, S. Zhang, R. Xu, and B. Wu. AT-LSTM: An Attention-based LSTM Model for Financial Time Series Prediction. *IOP Conference Series: Materials Science and Engineering*, 569:052037, Aug. 2019. doi: 10.1088/1757-899X/569/5/052037.
- R. Ślepaczuk, P. Sakowski, and G. Zakrzewski. Investment strategies that beat the market. what can we squeeze from the market? *Financial Internet Quarterly*, 14(4):36–55, 2018. doi: doi:10.2478/fiqf-2018-0026. URL <https://doi.org/10.2478/fiqf-2018-0026>.



UNIVERSITY OF WARSAW

FACULTY OF ECONOMIC SCIENCES

44/50 DŁUGA ST.

00-241 WARSAW

[WWW.WNE.UW.EDU.PL](http://WWW.WNE.UW.EDU.PL)