

LSTM-ARIMA as a Hybrid Approach in Algorithmic Investment Strategies

, Kamil Kashif^{a,1}, Robert Ślepaczuk^{b,2,*}

^aQuantitative Finance Research Group University of Warsaw Faculty of Economic Sciences ul. Długa 44-50 00-241 Warsaw Poland

^bQuantitative Finance Research Group Department of Quantitative Finance Faculty of Economic Sciences University of Warsaw ul. Długa 44-50 00-241 Warsaw Poland

Abstract

This study focuses on building an algorithmic investment strategy employing a hybrid approach that combines LSTM and ARIMA models referred to as LSTM-ARIMA. This unique algorithm uses LSTM to produce final predictions but boost results of this RNN by adding the residuals obtained from ARIMA predictions among other inputs. The algorithm is tested across three equity indices (S&P 500, FTSE 100, and CAC 40) using daily frequency data spanning from January, 2000 to August, 2023. The architecture of testing is based on the walk-forward procedure which is applied for hyperparameter tuning phase that uses using Random Search and backtesting the algorithms. The selection of the optimal model is determined based on adequately selected performance metrics combining focused on risk-adjusted return measures. We considered two strategies for each algorithm: Long-Only and Long-Short in order to present situation of two various groups of investors with different investment policy restrictions. For each strategy and equity index, we compute the performance metrics and visualize the equity curve to identify the best strategy with the highest modified information ratio (IR^{**}). The findings conclude that the LSTM-ARIMA algorithm outperforms all the other algorithms across all the equity indices what confirms strong potential behind hybrid ML-TS (machine learning - time series) models in searching for the optimal algorithmic investment strategies.

Keywords: Deep Learning, Recurrent Neural Networks, Algorithmic Investment Strategy, LSTM, ARIMA, Hybrid/Ensemble Models, Walk-Forward Process,

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

1. Introduction

Predicting the financial market is known to be quite challenging due to factors such as volatility, the complexity of the financial system, and the constantly changing economic landscape. We noticed twice in the past 20 years, the 2008 recession and the COVID-19 pandemic, that there was so much uncertainty on how the markets will progress. Researchers and traders try many approaches to

*Corresponding author

Email address: rslepaczuk@wne.uw.edu.pl (Robert Ślepaczuk)

¹ORCID: <https://orcid.org/0009-0004-0666-581X>

²ORCID: <https://orcid.org/0000-0001-5227-2014>

successfully predict the financial market. Unfortunately, not all are successful as it depends on the economic and political situation of the stock markets. They try to optimize their models ranging from simple linear regression models to advanced machine learning (ML) algorithms and being tested based on all types of invested assets.

High-frequency trading is gaining much more popularity; however, due to its complexity, it may not be available for all users. Therefore, for this research, we will consider only daily data. We believe that it can still give us a general picture of the interactions in the market and enable us to develop an algorithmic investment strategy (AIS). The main focus of this study is to utilize the Auto-regressive Integrated Moving Average (ARIMA) and Long-Short Term Memory (LSTM) models and combine them into a hybrid model called LSTM-ARIMA. The ultimate goal is to apply this hybrid model to develop an efficient AIS. Our main hypothesis states that the LSTM-ARIMA model will outperform other algorithms in most cases*. The ensuing are the research questions that our study aims to explore:

RQ1. Are the algorithmic investment strategies robust to changes in the asset?

RQ2. Does LSTM-ARIMA perform better than the models individually?

RQ3. Are the algorithmic investment strategies robust to changes in the model hyperparameters?

RQ4. Does the Long-Only or Long-Short strategy outperform the Buy&Hold?

To evaluate our algorithmic investment strategies, we have selected three assets, namely, S&P 500 (*GSPC*), FTSE 100 (*FTSE*), and CAC 40 (*FCHI*) equity indices. The motivation behind the choice of assets was to diversify its results across various equity indices to capture the finest capability of the AIS. Therefore, each asset is chosen from stock markets of different regions, from the New York Stock Exchange (NYSE) to the London Stock Exchange (LSE) and Euronext Stock Market (PAR). Our in-sample data begins on *2000-01-03* for S&P 500 and CAC 40 and *2000-01-04* for FTSE 100. The out-of-sample data for the S&P 500 equity index starts on 2005-01-25, for the FTSE 100 equity index on 2005-01-13, and the CAC 40 equity index on 2004-12-28 and lasts until *2023-08-30*. We capture the horizon of approximately 23 years. During this time frame, we capture two extreme times of the market. Considering extreme market conditions while training the model can help them perform well during both stable and volatile conditions.

Our contribution to the existing literature can be summarized in the following sentences. Firstly, we develop algorithmic investment strategies based on the predicted closing prices from ARIMA,

LSTM, and LSTM-ARIMA models and finally, we combine these forecasts into one ensemble model additionally boosting its results. Additionally, we use walk-forward optimization (WFO) as this technique reduces the risk of over-fitting to one specific sample of past returns. For each walk, we divide our data into training, validation, and testing data sets, where the training set equals 1000 trading days and the validation and testing set equals 250 trading days. Moreover, we perform hyperparameter tuning at every walk by performing a random search using a set of parameters explained later on in the study. Finally, the paper is finished with a sensitivity analysis of the most promising model to verify its robustness and potential for using it in real-time investments in financial markets. From a broad literature review, we have concluded, that there are very few papers that cover the process of testing algorithmic investment strategies in such a complex and reliable way.

The structure of the paper is as follows: Section 1 contains an introduction. Section 2 presents a brief overview of the literature. Section 3 provides us with the data description. Section 4 defines the methodology describing ARIMA, LSTM, and LSTM-ARIMA models. It also presents the WFO, performance metrics, research description, and hyperparameter tuning. Section 5 covers the empirical results of the strategies using the equity curves and performance metrics. Section 6 presents the sensitivity analysis where we show the sensitivity of the outcomes to changes in the set of hyperparameters. Section 7 presents the ensembled AIS. The last section concludes and presents a further extension of this paper.

2. Literature

Researchers are continuously searching for ways to build algorithmic investment strategies (AIS) and make higher and less risky profits in their investments This section will focus on the use of time series, machine learning, and hybrid models to forecast the stock market prices and create efficient algorithmic investment strategies.

2.1. Time Series Models

Time series models are considered to be well-performing as they can catch the features of the financial time series data such as the seasonality, trend, and cyclicity of historical data to predict the future values. The time series model used in this study is ARIMA. However, in this section, we will also discuss other types of models used for time series analysis.

ARIMA, introduced by Box and Jenkins (1976), has been one of the main tools of financial time series forecasting for a long time. ARIMA model is derived from the ARMA model, by taking

the first difference of the prices to have a stationary data set. Mondal et al. (2014) studied the effectiveness of ARIMA by forecasting fifty-six stocks from the Indian stock market from different industries. For their predictions, they achieved an accuracy of 85% and the fast-moving consumer goods (FMCG) sector was the most accurately predicted sector by ARIMA. Furthermore, they also concluded that in ARIMA the change in training data size does not influence the accuracy of their models.

Ariyo et al. (2014) used the ARIMA model to predict the prices of stocks from the New York Stock Exchange (NYSE) and the Nigerian Stock Exchange (NSE). They chose Nokia's and Zenith Bank stocks and the time frame was 16 years and 5 years respectively. The authors find the best model using the Bayesian Information Criterion (BIC), the standard error of regression, and the highest adjusted R-squared as the main criterion. They concluded that the best model to predict Nokia's stock was ARIMA(2,1,0) and for Zenith Bank, the best was ARIMA(1,0,1).

Devi et al. (2013) studied the effectiveness of ARIMA for the prediction of stock trends. The authors selected the parameters based on manual examination of ACF, and PACF plots to find the AR and MA orders. The best model was selected based on the AIC and BIC criteria. The paper considered five years of historical data for the analysis. The authors conclude that ARIMA is the most accurate model to predict the stock trend and make investment decisions.

Bui and Ślepaczuk (2022) explores the use of Hurst Exponent for an algorithmic pair trading strategy. The authors also explored the use of correlation and cointegration for their pair trading strategy. The study is focused on 103 stocks from the NASDAQ 100 equity index, covering approximately 18 years with daily frequency. The empirical findings indicate that among all 103 stocks, the correlation method demonstrated superior performance in terms of risk-adjusted return. However, the Buy&Hold strategy outperformed all other strategies in terms of compounded annualized return.

Malladi and Dheeriya (2021) conducted a time series analysis of cryptocurrency returns and volatility using GARCH, VAR, and ARMAX models. ARMAX is an extension of the base ARMA which considers exogenous inputs. They test the algorithm on BTC and XRP. The comparison is done with a standard regression model. The conclusion is that both ARMAX and GARCH perform better than the standard regression and the VAR model, as expected. However, ARMAX showed the best results due to its high accuracy.

Li et al. (2023) introduces spARIMA, a novel time series prediction framework designed with a sequential training approach in batches. Named for its sequential training based on noise levels and

model fit contributions, spARIMA incorporates a self-paced learning (SPL) strategy to effectively mitigate data noise-induced instability. The model's performance is evaluated across twelve diverse datasets, including equity index prices (NASDAQ, RUSSEL, NYSE), as well as traffic and temperature data. Comparisons are made with traditional ARIMA models using two gradient descent algorithms. While spARIMA did not consistently outperform ARIMA across all datasets, it demonstrates promising capabilities against strong noise in time series prediction tasks, demonstrating its potential in enhancing forecasting accuracy.

The Vector Autoregression (VAR) model is also known for its time series modeling capabilities. Suharsono et al. (2017) uses VAR and VECM to model the stock price. They use the ASEAN share price index and perform a manual search to find the best parameters for the models. The criteria to check the performance of the model was based on the Akaike information criterion (AIC). They concluded that in comparison to the VECM model, the VAR model performed the best in modeling.

Castellano Gómez and Ślepaczuk (2021) analyzed four algorithmic strategies and one of them was based on ARIMA. They used S&P 500 equity index data for the predictions and used almost 31 years of historical data. The goal was to create a portfolio strategy using four selected algorithmic strategies. All the strategies were compared with the benchmark buy&hold. The paper showed that ARIMA did not perform well when compared to the Buy&Hold strategy, however, the performance of ARIMA was the highest during the phases of high volatility.

2.2. Machine Learning Models

Machine learning is an advanced approach based on artificial intelligence which can be used to forecast stock market prices. In recent years, Recurrent Neural Networks (RNN) have started to be used more often for time series analysis. Rumelhart (1986) made significant contributions to the field of RNN. Due to issues such as the vanishing gradient problem and the inability to effectively capture long-term dependencies, the development of RNNs faces certain limitations. Hochreiter and Schmidhuber (1997) proposed the architecture of the Long-Short-Term Memory (LSTM) model to tackle the vanishing gradient problem. This gives LSTM a huge advantage over RNN especially when it comes to time series analysis.

Xiong et al. (2014) presents an innovative approach leveraging a firefly algorithm (FA) to optimize multi-output support vector regression (MAVR) parameters in financial forecasting. Their study evaluates this FA-MAVR model across statistical, economic, and computational criteria. Statistical evaluation includes goodness-of-forecast measures and testing methodologies, while economic criteria assess the model's performance using a naive trading strategy. Computational efficiency is also

considered. Testing is conducted on major equity indices: S&P 500, FTSE 100, and Nikkei 225. In comparison to genetic algorithms and particle swarm optimization, FA-MAVR demonstrates superior forecast accuracy and profitability, establishing its effectiveness in equity indices price prediction.

Siarni-Namini et al. (2018) compared the use of LSTM and ARIMA in forecasting time series. They used them to predict the monthly closing prices for eleven stock market indices. In comparison to ARIMA, they conclude that LSTM outperforms the ARIMA model, which results in RMSE measure lower by 85%. Furthermore, they also mentioned that LSTM results were robust to the number of epochs used in the process of estimation.

Grudniewicz and Ślepaczuk (2023) researched various machine learning techniques for creating an AIS. They utilized various machine learning models, including Neural Networks, K Nearest Neighbours, Regression Trees, Random Forest, Naive Bayes classifiers, Bayesian Generalized Linear Models, and Support Vector Machines. These ML models were employed to generate trading signals for WIG20, DAX, S&P500, and six CEE indices over a timeframe spanning approximately 21 years. The authors concluded that in terms of risk-adjusted returns, the Polynomial Support Vector Machine model performed the best in the case of WIG20 and S&P 500 equity indices, while the Linear Support Vector Machine model for DAX and six CEE equity indices.

Roondiwala et al. (2017) presented a study predicting stock prices using LSTM. Five years of historical data on the NIFTY 50 index was used for testing purposes. The training of LSTM models was done by allocating random weights and biases with an architecture of two LSTM layers and two dense layers with ReLU and Linear activation function respectively. Finally, the predicted values were compared with the actual values and evaluated using the RMSE. The best RMSE score was given for the model with High/Low/Open/Close as the inputs with 500 training epochs.

Michańków et al. (2022) presented a study on using LSTM in Algorithmic Investment Strategies (AIS) on BTC and S&P500 Index. The output of their model was a singular value predicting the next day's return value -1, 0, 1. The set of hyperparameters used for the tuning process, relating to this paper, were the number of layers between 1 and 5, the number of neurons in each layer chosen between 5 and 512, dropout rates between 0.001 and 0.2, several types of optimizers including SGD, RMSProp, and Adam variants, learning rates chosen between 0.001 and 0.1, and the batch size ranging from 16 to the length of the test size. After the hyperparameters tuning phase, they selected the model with 3 hidden layers, with 512/256/128 neurons respectively, a dropout rate equal to 0.02, Adam as an optimizer a learning rate of 0.00015, and a batch size of 80. They deduced that when it comes to daily frequency, their model for S&P 500 equity index performed well for the Long-Only

strategy, while the model for BTC performed well for both the Long-Only and Long-Short strategy.

Zhang et al. (2019) wrote an analysis of the Attention-based LSTM model for financial time series prediction (AT-LSTM). Instead of making their prediction of LSTM by inputting the prediction of ARIMA, the authors use the output of the attention model as the input of LSTM. The authors compared the results of AT-LSTM with ARIMA and LSTM. The testing and training were done on three data sets: Russell 2000, DIJA, and NASDAQ indices, and the best model was concluded based on the MAPE (mean absolute percentage error). LSTM performed the best with 2 layers, 8 hidden neurons, 20 training time steps, batch size equal to 50, and 5000 epochs. Finally, the authors summarized that the hybrid AT-LSTM performed better than LSTM and both of them performed way better than ARIMA.

Baranochnikov and Ślepaczuk (2022) analyzed various architectures of LSTM and GRU models in Algorithmic Investment Strategies. Their LSTM model forecasted the rate of return for the period $T+1$. The authors decided to use the set of parameters chosen from the financial literature. Ten model architectures were used during the training process and parameters such as dropout rate, batch size, epochs, and the learning rates were additionally modified. Adam optimizer with the AMSGrad extension was used in all. The authors used the walk-forward process for estimation purposes. The models were tested on Bitcoin, Tesla, Brent Oil, and Gold closing prices. The authors deduced that the LSTM outperformed the traditional Buy&Hold strategy for Bitcoin and Tesla both for daily and hourly frequency.

Another research was conducted in India by Hiransha et al. (2018) who predicted the National Stock Market of India and the New York Stock Exchange using various algorithms such as MLP, RNN, LSTM, and CNN. ARIMA was used as a benchmark. The authors tested three sectors of industries, automobile, finance, and IT from both stock exchanges and measured the accuracy of the predictions using the MAPE metric. The authors concluded that CNN (Convolution Neural Network) outperformed all the other models and that LSTM was better performing than ARIMA due to its useful capability of finding non-linearity.

Kijewski and Ślepaczuk (2020) predicted the S&P 500 equity index prices using the classical models and RNN (Recurrent Neural Networks including ARIMA, MA, momentum and contrarian, volatility breakout, macro factor, and finally LSTM. The models were trained and tested collectively over twenty years. The range of parameters taken for the ARIMA model were, p: 0-5, d: 0-3, and q: 0-5, while in the case of LSTM, a prepared set of hyperparameters was taken from the literature. They mentioned that the best LSTM model had the following configuration: 30 neurons in the

hidden layers with ReLU activation, length of sequence equaling 15, and dropout rate 0.02 by using Adam optimizer with learning rate 0.01 and loss function Mean Square Error. The authors concluded that LSTM outperformed ARIMA and the benchmark buy&hold strategy.

Kryńska and Ślepaczuk (2022) tested several architectures of the LSTM model in AIS based on the S&P 500 equity index and BTC. They tested three frequencies of data: daily, hourly, and 15-minute of S&P 500 and BTC. When the model was considering a regression problem, models on daily data performed better than intraday frequencies. However, in the case of classification problems, the model on intraday data performed the best.

Nelson et al. (2017) used LSTM to predict the stock market direction. The authors use different stocks from the Brazilian stock exchange and some technical indicators as inputs. Furthermore, log-return transformation was performed for the inputs and the frequency of data was 15 minutes. The output of the model is a binary value (1, 0) denoting an increase and decrease in the prices between the time steps. Four metrics were created to evaluate the performance, among which were the accuracy and the precision. The authors concluded that the proposed model of the paper outperforms the benchmark Buy&Hold strategy based on accuracy and offers less risky investment compared to the others.

Mizdrakovic et al. (2024) investigates Bitcoin price dynamics by analyzing factors including Ethereum, S&P 500, VIX, EUR/USD, and GBP/USD. The study introduces a two-tiered methodology: initially employing variational mode decomposition (VMD) enhanced with a variant of the sine cosine algorithm to optimize VMD's control parameters for trend extraction from time series data. Subsequently, LSTM and hybrid Bidirectional LSTM models are utilized to forecast prices over multiple time steps. The authors evaluate their approach across various feature sets, comparing performances with and without VMD. Their findings highlight the outperformance of the V-BiLSTM-HSA-SCA (VMD Bidirectional LSTM with hybrid self-adaptive sine cosine algorithm) model, demonstrating its highest R2 and IA scores, as well as lowest MAE, MSE, and RMSE values among all tested models.

2.3. Hybrid Models

Hybrid models are a combination of two models. A mix of models may work better together as they capture the efficiencies of individual models. For instance, LSTM-ARIMA models can help us capture the linear and non-linear dependencies in the data. Most algorithms can be combined. One of the common combinations is a mixture of ARIMA and GARCH models. Vo and Ślepaczuk (2022) tested a hybrid ARIMA-SGARCH model in algorithmic investment strategies (AIS). Three models

were created during their research: ARIMA, ARIMA-SGARCH, and ARIMA-EGARCH. The models were tested on the S&P 500 equity index and prices in the period of the last 20 years. The authors selected the best ARIMA model using the Akaike Information Criterion (AIC). The results were compared between the models and the benchmark buy&hold strategy. Several performance metrics were used. The authors concluded that the ARIMA-SGARCH model performed the best, followed by ARIMA which performed better than the benchmark.

Senneset and Gultvedt (2020) uses ARIMA and LSTM together to increase their portfolio stability. They used several stocks from the Oslo Stock Exchange for 14 years. The authors used the residual values from ARIMA as an input and the performance was compared using the RMSE and MAE error metrics. During the research, two hybrid models were created: ARIMA-RandomForest and ARIMA-LSTM. The results concluded that ARIMA-RandomForest outperformed all the strategies, while ARIMA-LSTM outperformed just the benchmark strategy.

LSTM-ARIMA was also considered in the approach to forecasting the wind speed by Bali et al. (2020). A few wind parameters such as wind speed, temperature, pressure, etc were used as inputs for LSTM. The authors compared the LSTM-ARIMA model with LSTM and the support vector machine (SVM) using the RMSE. They concluded that the LSTM-ARIMA was the most accurate model compared to LSTM and SVM.

Arnob et al. (2019) forecasted the Dhaka stock exchange (DSE), using the hybrid ARIMA-LSTM approach. The aim was to forecast the correlation coefficient between the assets. They used fifteen companies from DSE to forecast. Several ARIMA orders were chosen using the ACF and PACF plots; however, the best order was chosen based on the lowest AIC. The data was divided into three parts: train, test1, and test2, and the performance was measured using MSE and MAE. The researchers concluded that ARIMA-LSTM performed better than the ARIMA model.

Karim et al. (2022) predicted the stock price of NIFTY-50 stock using a bidirectional LSTM and GRU network hybrid model (Bi-LSTM-GRU). The hybrid model was compared with each model being trained individually and the one with the highest precision was marked as the best. The authors concluded that their proposed hybrid approach outperformed the models individually.

Oyewola et al. (2024) present an approach to stock price prediction within the oil and gas sector, an industry with complex market dynamics and diverse external influences. The study introduces three models: Deep Long Short-Term Memory Q-Learning (DLQL), Deep Long Short-Term Memory Attention Q-Learning (DLAQL), and the state-of-the-art Long Short-Term Memory (LSTM) model. Historical stock prices of CVE, MPLX LP, LNG, and SU are utilized for training and evaluation.

The reinforcement learning technique employed is the Markov Decision Process (MDP) framework. Results indicate that the DLAQL model outperforms all the others in decision-making capabilities, risk management, and most importantly, the profitability, positioning it as a robust choice for stock price prediction in the oil and gas sector.

To summarize, the ARIMA model has been extensively used for financial time series forecasting. Studies show this model’s capabilities in predicting stock prices from various exchanges. The results vary based on the statistical criteria chosen such as AIC and BIC. Other methods were also explored such as Vector Autoregression (VAR) or Vector Error Correction Models (VECM) for stock price modeling purposes. Furthermore, deep learning techniques such as the Long Short-Term Memory model (LSTM), have gained traction for time series forecasting. LSTM is a model designed to address issues like vanishing gradients in traditional Recurrent Neural Networks (RNN) and shows superior performance when compared to ARIMA. Other deep learning techniques, including Support Vector Machines (SVM) or Random Forest (RF), have also been investigated and demonstrated improved predictive capabilities compared to traditional models like ARIMA.

Based on the summary provided, we propose that combining the two top-performing models, namely ARIMA and LSTM, could help overcome the limitations of each model. ARIMA excels at capturing short-term time series patterns, while LSTM is adept at modeling long-term dependencies. By integrating these models, we anticipate outperforming their capabilities and potentially gaining an edge in beating the market.

3. Data Description

We consider three equity indices in our research and the data is taken from Yahoo Finance using their *yfinance* API for *Python*. Table 1 presents the descriptive statistics of the chosen assets. Please note that the difference in the count value may arise due to the difference in the number of trading days in a year in each country.

Table 1: Descriptive Statistics for the closing price series.

	Count	Mean	Standard Deviation	Min	25%	50%	75%	Max
S&P 500	5953	1939	1027	677	1190	1449	2486	4797
FTSE 100	5975	6057	1050	3287	5332	6150	6867	8014
CAC 40	6049	4714	1118	2403	3798	4570	5476	7577

Note: The descriptive statistics for S&P 500, FTSE 100, and CAC 40 are calculated on the closing price in the period from 2000-01-03 for S&P 500 and CAC 40 and 2000-01-04 for FTSE 100 until 2023-08-31.

Figure 1 presents the prices of tested assets and their volatilities. For the S&P 500 equity index, we use the CBOE's VIX as the measure of volatility and for the FTSE 100 and CAC 40 equity index we use the annualized realized volatility with a historical window of 21 days, which is calculated using the following formula:

$$annRV = \sqrt{\frac{\sum_{i=21}^N R_t^2}{N}} \times \sqrt{252} \quad (1)$$

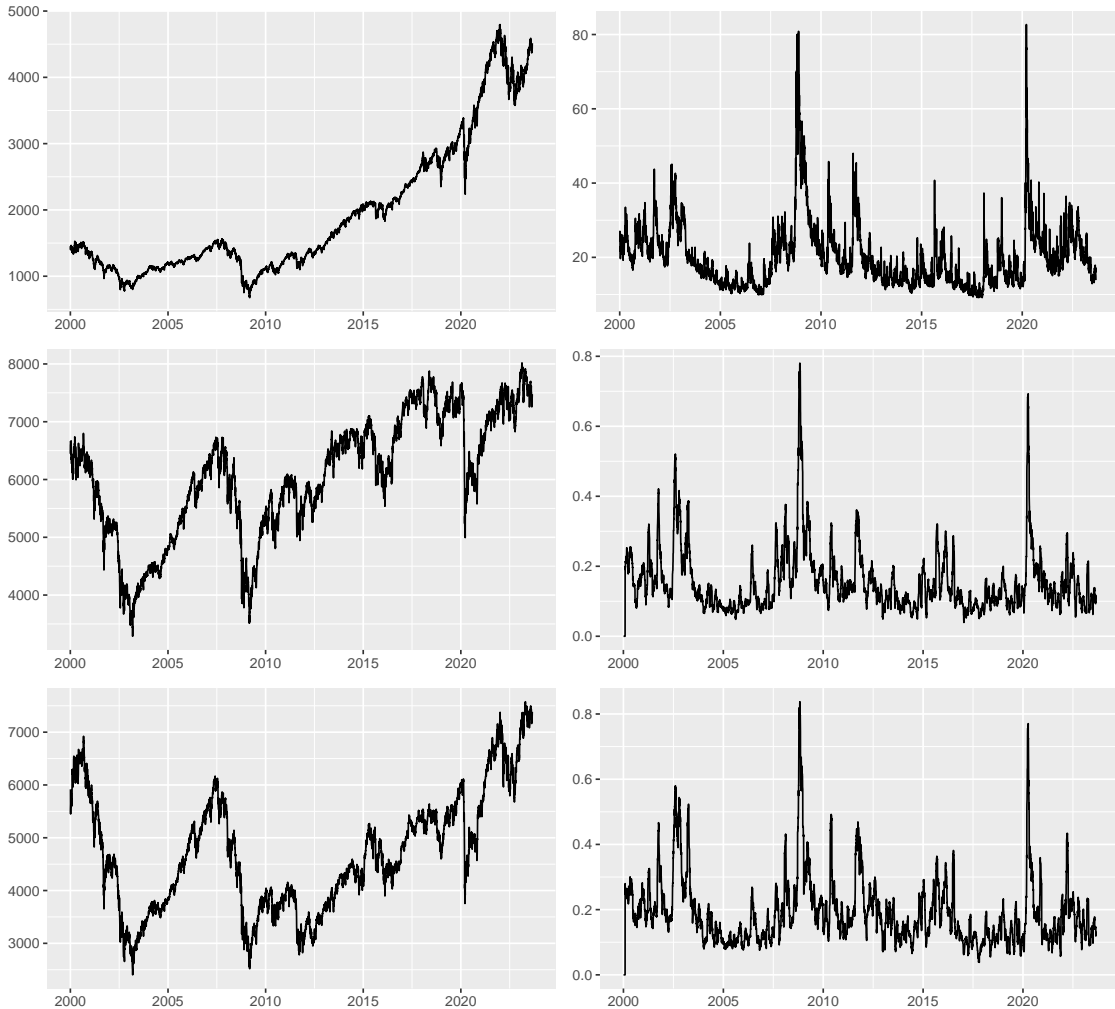
where:

R_t - the daily returns

t - the counter representing each trading day

N - the number of trading days in our time frame

Figure 1: Index Prices and their Volatilities.



Note: Each plot on the left shows the behavior of the index based on their closing price and each plot on the right represents the volatility for the corresponding index. Note that for S&P 500, CBOE's VIX has been used as a measure of volatility, and for the FTSE 100 and CAC 40, the realized volatility was used with a window of 21 trading days.

4. Methodology

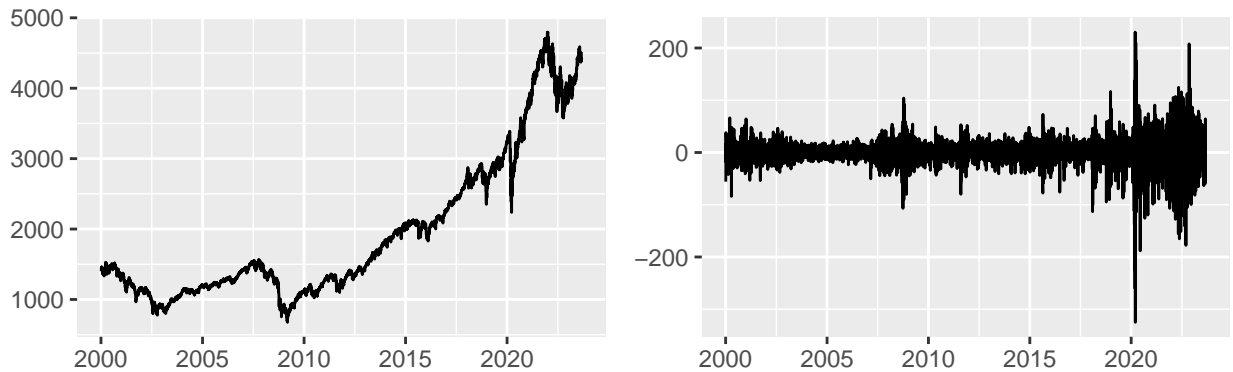
4.1. ARIMA

Autoregressive Integrated Moving Average model (ARIMA), is an econometric model used for forecasting time series data based on some historical data. The model was introduced by George Box and Gwilym Jenkins in 1976, and they initially used it to model changes in financial time series data. The model consists of three parts: Autoregressive (AR), Integrated (I), and Moving Average (MA), where each component has its order. Let's denote p , an order for AR component, d , an order for I component, and q , an order for MA component. Denoting this we can then write the model as $ARIMA(p,d,q)$. The respective orders determine the following properties of the model:

- p - the number of lagged observations
- d - the number of times the data was differenced
- q - the order of the MA process

In terms of stock forecasting, order d is usually set to 1 when we model the prices and 0 when we model the returns as the data is already stationary, and then we have $ARMA(p,q)$. Figure 2 presents the difference between the non-stationary data set which is the closing prices and the stationary data set which are the first differences of prices.

Figure 2: Non-stationary data set compared to stationary data set of S&P 500.



Note: The plot on the left side shows an actual plot of S&P 500 closing price and the plot on the right side shows the differentiated closing price.

Let's now look at the formal side of an ARIMA model. ARIMA generally is an extension of the ARMA model. The $AR(p)$ can be denoted with the following equation:

$$AR(p) : y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t \quad (2)$$

where:

y_t - the value of the time series at a time t

ϵ_t - the error term

ϕ - the coefficients that capture the relationship between the current observation and previous observations at a lag of p

The autoregressive component is responsible for forecasting the chosen variable using the past value of the variable automatically.

The $AR(p)$ model can be written as:

$$AR(p) : (1 - \sum_{i=1}^p \phi_i L^i) y_t = c + \epsilon_t \quad (3)$$

The second component that is responsible for differencing can be denoted as $I(d)$ and presented as:

$$I(d) : (1 - L)^d = \mu + \epsilon_t \quad (4)$$

The third component, moving average, looks as follows:

$$MA(q) : y_t = \mu + \epsilon_t + \theta_1 \epsilon_{t-1} + \dots + \theta_q \epsilon_{t-q} \quad (5)$$

where:

μ - the mean of the given series

$\theta_1 \dots \theta_q$ - the respective weights for each error term $\epsilon_{t-1} \dots \epsilon_{t-q}$.

This represents the moving average procedure with order q . Unlike $AR(p)$, the $MA(q)$ uses the previous error terms for the regression. And using the lag operator, $MA(q)$ may be denoted as:

$$MA(q) : y_t = \mu + (1 + \sum_{i=1}^q \theta_i L^i) \epsilon_t \quad (6)$$

Using the two components explained previously, we can find the $ARIMA(p, \theta, q)$ model which is also known as $ARMA(p, q)$ model. The equation of $ARMA(p, q)$ looks as follows:

$$ARMA(p, q) : y_t = \phi_1 y_{t-1} + \phi_2 y_{t-2} + \dots + \phi_p y_{t-p} + \epsilon_t - \theta_1 \epsilon_{t-1} + \dots - \theta_q \epsilon_{t-q} \quad (7)$$

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

$$ARIMA(p, d, q) : (1 - \sum_{i=1}^p \phi_i L^i)(1 - L)^d y_t = c + (1 + \sum_{i=1}^q \theta_i L^i) \epsilon_t \quad (8)$$

where:

d - the number of times the series was differenced.

4.2. RNN

According to Turing (2023), a job platform, a recurrent neural network (RNN) is a variation of artificial neural networks (ANN). RNN may be used to address various problems such as speech recognition or image captioning. What differentiates RNN from ANN is that ANN just takes the inputs and generates outputs; however, RNN learns from the previously generated outputs to provide results for the next time stamp. Another advantage of RNN is that it has a memory cell that continues the calculations and if the forecast is inaccurate the network auto-learns and executes backpropagation to get the correct result. RNN is very effective for time series forecasting due to its ability to recollect previous inputs. This is where the Long-Short-term memory (LSTM) model comes in.

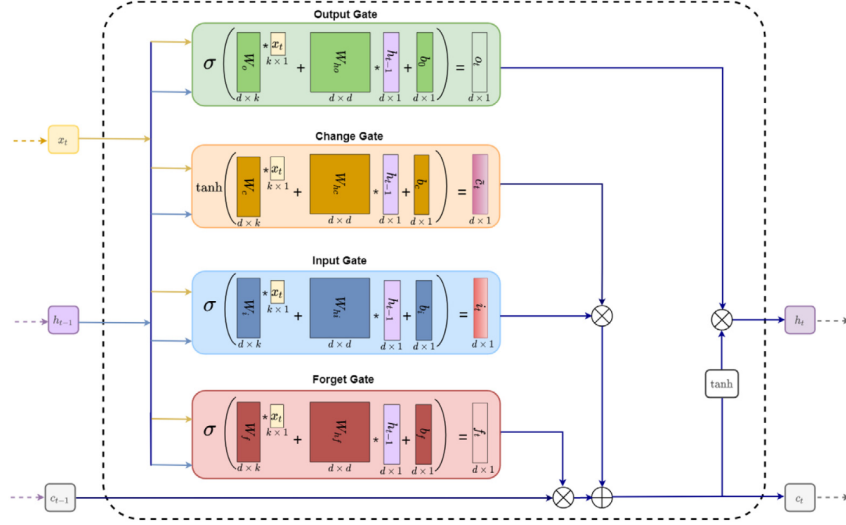
4.2.1. LSTM

LSTM is a special type of RNN. Its main character is the ability to handle long-term data dependencies and push the outcome to the succeeding node more efficiently. It also addresses the vanishing gradient problem, a known issue with RNN, which is tackled by disregarding nugatory information using its forget gate. LSTM also deals well with long-term dependencies i.e. with problems where the output is dependent on the historical inputs. LSTM consists of multiple gates, each having an essential task to be done to have positive results.

Figure 3 presents LSTM which has four gates: input, output, forget, and change. For a sequence in time $x_t - (x_1, x_2, \dots, x_n)$ the forget gate f_t takes the x_t and the hidden state h_{t-1} and produces a binary output 0 and 1 through a sigmoid function and identifies which information should be discarded from the memory cell c_{t-1} . The value equal to 1 is forwarded to the cell with the value equal to 0 and all the other information is forgotten. The input gate i_t identifies what to update from the change gate \hat{c}_t and the output gate o_t decides which information should be taken from the current cell. From the sequence X , two sequences x and y are created, where x is the input sequence and y is the next day closing price. Furthermore, it's worth noting that the memory cell is

responsible for long-term memory and it updates the input gate, forget gate and the change gate. On the other hand, the hidden state is responsible for the short-term memory and is updated by the output gate and the memory cell. The explanation above was influenced by Bhandari et al. (2022) research on ‘Predicting stock market index using LSTM’.

Figure 3: The architecture of Long-Short-term memory.



Note: The architecture of LSTM, source: <https://www.sciencedirect.com/science/article/pii/S2666827022000378>.

The mathematical equations to the previously given terminologies look as follows:

$$i_t = \sigma(W_i x_t + W_{hi} h_{t-1} + b_i) \quad (9)$$

$$f_t = \sigma(W_f x_t + W_{hf} h_{t-1} + b_f) \quad (10)$$

$$o_t = \sigma(W_o x_t + W_{ho} h_{t-1} + b_o) \quad (11)$$

$$\hat{c}_t = \tanh(W_c x_t + W_{hc} h_{t-1} + b_c) \quad (12)$$

$$c_t = f_t \cdot c_{t-1} + i_t \cdot \hat{c}_t \quad (13)$$

$$h_t = o_t \cdot \tanh(c_t) \quad (14)$$

where:

W - weights

b - biases

x_t - sequence of time t

f_t - forget gate at time t

h_{t-1} - hidden state at time $t - 1$

i_t - input gate at time t

\hat{c}_t - change gate at time t

o_t - output gate at time t

4.3. LSTM-ARIMA

In this paper, we introduce a hybrid approach using the ARIMA and LSTM models collectively. This model contains an LSTM input layer which is fed with the residuals of ARIMA predictions and other inputs such as the closing price and the volatility. LSTM-ARIMA is a combination that helps capture both the linear and non-linear properties of the data. Moreover, LSTM is known for its outstanding capability to capture the long-term dependencies in time series data and ARIMA is known for its outstanding capability to capture the short-term dependencies in time series data. Additionally, ARIMA learns from data using statistical methods and LSTM learns by looking at the pattern thanks to the neural networks. Considering all of the above strengths and weaknesses of the models, we believe that collectively they may outperform the performance of both individually. Generally, the process of using the LSTM-ARIMA approach for AIS in this paper can be summarized in the following way:

1. Find the best-fitted ARIMA model using the set of orders based on the smallest Akaike information criterion (AIC).
2. Get the residuals from ARIMA.
3. Perform feature engineering on LSTM, taking into consideration the residuals by ARIMA, the closing price, and the realized volatility of the asset under consideration (in the case of the S&P 500, take VIX).
4. Conduct a random search and choose the *best set of hyperparameters* based on the criteria outlined in the subsequent sections.
5. Fit the best model and execute predictions for buy/sell signals generations.
6. Create the equity curve based on investment signals from the previous point and then compute the performance metrics.

4.4. Walk Forward Optimization

Over-fitting is a big risk in machine learning algorithms, especially in financial time series forecasting. Common cross-validation techniques like *k-fold* are not well suited for financial analysis

and adjusting the hyperparameters may result in over-fitting. Common cross-validation techniques sometimes do not perform as well as intended. Therefore, to have a robust trading strategy it is advised to use the walk-forward optimization (WFO) approach. Carta et al. (2021) stated that walk-forward optimization is one of the most popular validation techniques used by financial researchers to undergo decision-making for trading. There are two types of WFO: anchored, and non-anchored. The difference lies that in anchored WFO each walk has a common beginning point; however, in the non-anchored type each walk has a different starting point but the same length. In this research, we considered the non-anchored type for training, validation, and testing as we believe its robustness is higher than the anchored type. We set the in-sample (IS) window to 1250 trading days where the training set is equal to 1000 trading and the validation set is equal to 250 trading days and we set the out-of-sample (OOS) window to 250 trading days. This is visualized in Figure 4.

Figure 4: Walk forward optimization scheme with 5-years IS and 1-year OOS.



Note: The bars in green color represent the training data set, in yellow color represent the validation data set, in red color represent the out-of-sample testing data set, and the bars in the dark-red color represent the total out-of-sample data. This plot was designed by using the data for the S&P 500 equity index. However, it looks similar for the FTSE 100 and CAC 40 equity indices. The training window is 1000 trading days, and validation and testing windows are 250 trading days each.

4.5. Performance Metrics

To assess and evaluate the robustness of the trading strategies created in this paper, we calculated the performance metrics based on Michańków et al. (2022) and Bui and Ślepaczuk (2021). The details are presented:

Annualized Return Compounded (*ARC*), shows the rate of return that was annualized for the given strategy during the period of $(0, \dots, T)$. It is expressed in percentage.

$$ARC = \left(\prod_{t=1}^N (1 + R_t) \right)^{\frac{252}{N}} - 1 \times 100\% \quad (15)$$

where:

R_t - the percentage rate of return

N - the sample size

R_t is calculated in the following way:

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} \quad (16)$$

where:

P_t - the price at point t

Annualized Standard Deviation (*ASD*) is a risk measure.

$$ASD = \sqrt{252} \times \sqrt{\frac{1}{N-1} \sum_{t=1}^N (R_t - \bar{R})^2} \times 100\% \quad (17)$$

where:

R_t - the percentage rate of return

\bar{R} - the mean rate of return

N - the sample size

\bar{R} is calculated in the following way:

$$\bar{R} = \frac{1}{N} \sum_{t=1}^N R_t \quad (18)$$

Maximum Drawdown (*MD*) gives us the maximum percentage drawdown throughout the invest-

ment and is calculated as follows:

$$MD(T) = \max_{\tau \in [0, T]} (\max_{t \in [0, \tau]} (R_{i, T} - R_{i, \tau})) \times 100\% \quad (19)$$

Maximum Loss Duration (*MLD*) tells us about “the number of years between the previous local maximum to the forthcoming local maximum” (Michańków et al. (2022)) and is calculated as follows:

$$MLD = \max\left(\frac{m_j - m_i}{S}\right) \quad (20)$$

Information Ration (*IR**) describes the risk-adjusted return metric based on the relation between ARC to its ASD and is calculated as follows:

$$IR^* = \frac{ARC}{ASD} \times 100\% \quad (21)$$

Modified Information Ratio (*IR***) is another more complex and comprehensive risk-adjusted return metric which we regard as the **most important** metric for the evaluation of strategies in this research and is calculated as follows:

$$IR^{**} = IR^* \times ARC \times \frac{\text{sign}(ARC)}{MD} \% \quad (22)$$

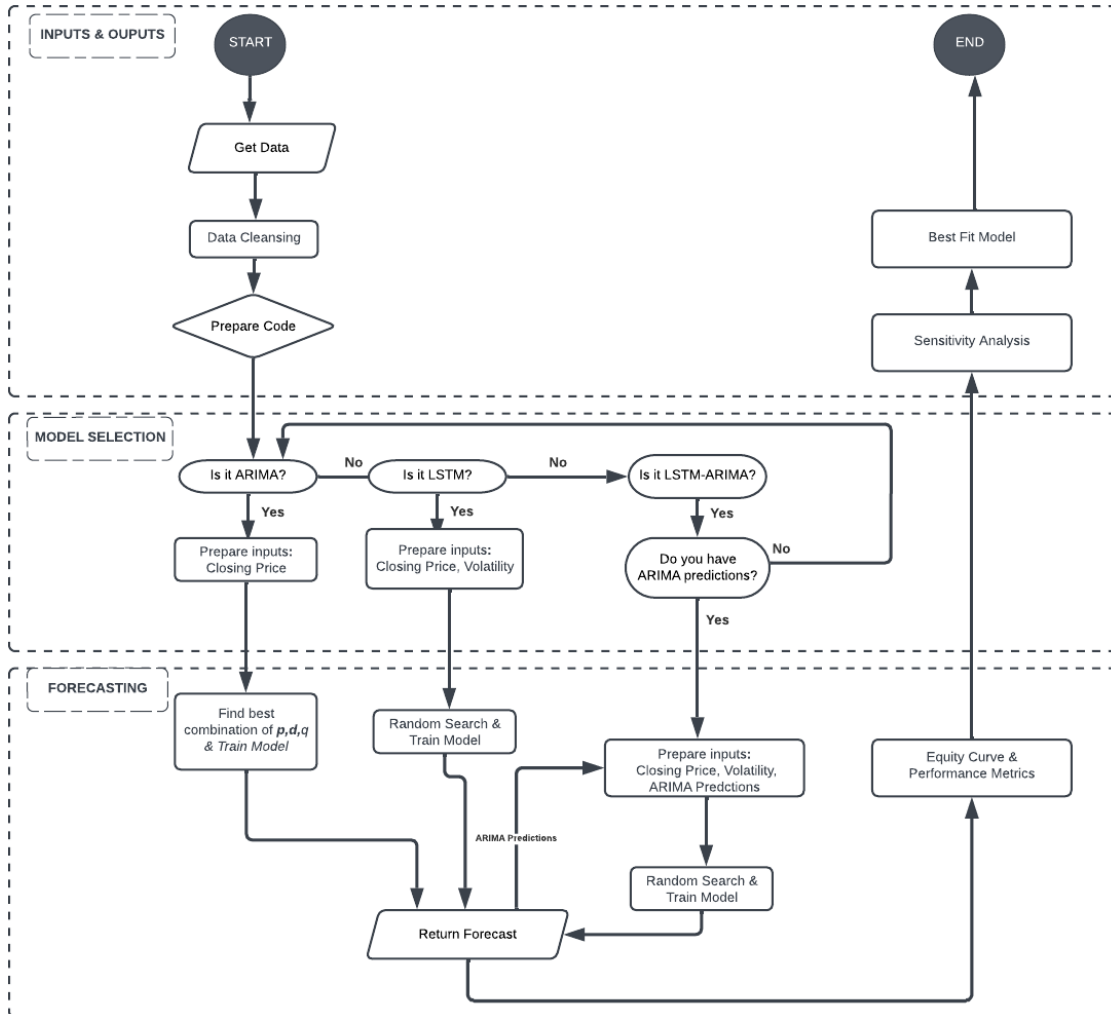
4.6. Research Description

In this study, we use a random search for hyperparameter tuning conducted at each walk of WFO. The steps of the research are presented below:

1. Select the asset and download the data for 1-day frequency using the *yfinance* Python API.
2. Perform data cleansing and prepare the data for feature engineering.
3. Create a code that supports the whole study including the sensitivity analysis.
4. Select the base model scenario for each model.
5. Run a random search and test the strategies.
6. Generate a prediction and take a position based on the criteria presented in the next section.
7. Create equity curves and calculate the performance metrics.
8. Conduct a sensitivity analysis.
9. Summarize the study with the best parameters.

In Diagram 1, the steps can be visually represented to provide a clearer understanding.

Diagram 1: Research Description Flow Chart.



Note: The diagram presents the process of research done in this study.

4.7. Best Set of Hyperparameters Criteria

In our research, we employ random search as a method of hyperparameter tuning. During random search, we select five models with the lowest validation loss. Then we calculate the $IR2$ on the training data set and the validation data set and then calculate the absolute value of their difference. The *best model* is the one with the lowest absolute value of the difference and where the $IR2$ for the validation data set was **NOT** equal to **zero**.

4.8. Strategy

During this research, we considered two kinds of strategies to be evaluated: *Long-Only* and *Long-Short*. *Long-Only* is where we allow to open either a long position (1) or hold no position (0). *Long-Short* is where we allow to open either a long position (1) or a short position (-1). The change for a Long position in both strategies happens whenever the predicted price of $t + 1$ is higher than the price at time t and we take a Short position or hold no position whenever the predicted price of $t + 1$ is lower than the price at time t . Note that at this step we already have the prediction generated and we use them to test the algorithm. Below we present the mathematical notation of the Long-Only and Long-Short strategies:

$$\text{Long - Only : } \begin{cases} \text{Signal} = 1 \text{ if } P_{t+1} > P_t \\ \text{Signal} = 0 \text{ if } P_{t+1} < P_t \end{cases} \quad (23)$$

$$\text{Long - Short : } \begin{cases} \text{Signal} = 1 \text{ if } P_{t+1} > P_t \\ \text{Signal} = -1 \text{ if } P_{t+1} < P_t \end{cases} \quad (24)$$

where:

P_{t+1} - the closing price at $t + 1$

P_t - the closing prices at t

4.9. Hyperparameter Tunning

In this section, we present to you the set of parameters that we use to employ random search. Out of all combinations we conduct 20 trials on a randomly chosen set. We perform the experiments in *python 3.8.15* using the *Tensorflow* library. For ARIMA, the whole process from model training, including all WFO walks, to generating the predictions for a single asset took us approximately 3 minutes, for LSTM approximately 3 hours, and for LSTM-ARIMA approximately 4 hours. Note that for all models, the IS window is equal to 1250 trading days and the OOS window is equal to 250 trading days. By default, we have set the number of epochs to 100. However, using the *Keras EarlyStop* function we optimize the number of epochs based on the validation loss while setting the patience to 10 epochs.

4.9.1. ARIMA

The range of parameters that were selected keeping the other constant are:

- *AR degree (p)*: from 0 to 6
- *Integrated degree (d)*: 1, as we perform model training on closing prices
- *MA degree (p)*: from 0 to 6

The models are chosen based on the Akaike Information Criterion (AIC). The objective of AIC is to find a balanced model that does not lose a lot of information and is also accurate. AIC also penalizes the models with more beta parameters. Therefore, the model with the lowest AIC is chosen. Based on Al-Gounmееin and Ismail (2020), AIC is calculated in the following way,

$$AIC = -2\ln(\hat{l}) + 2k \quad (25)$$

where:

k - the number of the parameters to be estimated

l - the likelihood for the respective model

4.9.2. LSTM

The set of hyperparameters chosen to perform a random search, keeping the other constant, are the following:

- *Neurons*: [25, 50, 75, 100, 250, 500]
- *Number of hidden layers*: [1, 2]
- *Dropout rate*: 0.075
- *Optimizer*: [Adam, Nadam, Adagrad]
- *Learning rates*: [0.01, 0.0001]
- *Loss Function*: Mean Square Error
- *Batch size*: 32
- *Sequence Length*: [7, 14, 21]
- *Input Layer Activation Function*: *sigmoid*
- *Output Layer Activation Function*: *tanh*

The following features were used to predict the closing price at time $t + \text{sequence_length}$:

- Closing price at time t
- Volatility at time t
- Trading volume at time t

4.9.3. LSTM-ARIMA

The hyperparameters for the LSTM-ARIMA hybrid model are the same as those utilized for the individual ARIMA and LSTM models. Random search serves as the search algorithm to tune these hyperparameters. Initially, ARIMA generates predictions, which are subsequently incorporated into the LSTM model. The input layer uses variables such as closing price, volatility t , trading volume at time t , and residuals from the ARIMA model at time t to predict the closing price at time $t + sequence_length$.

5. Empirical Results

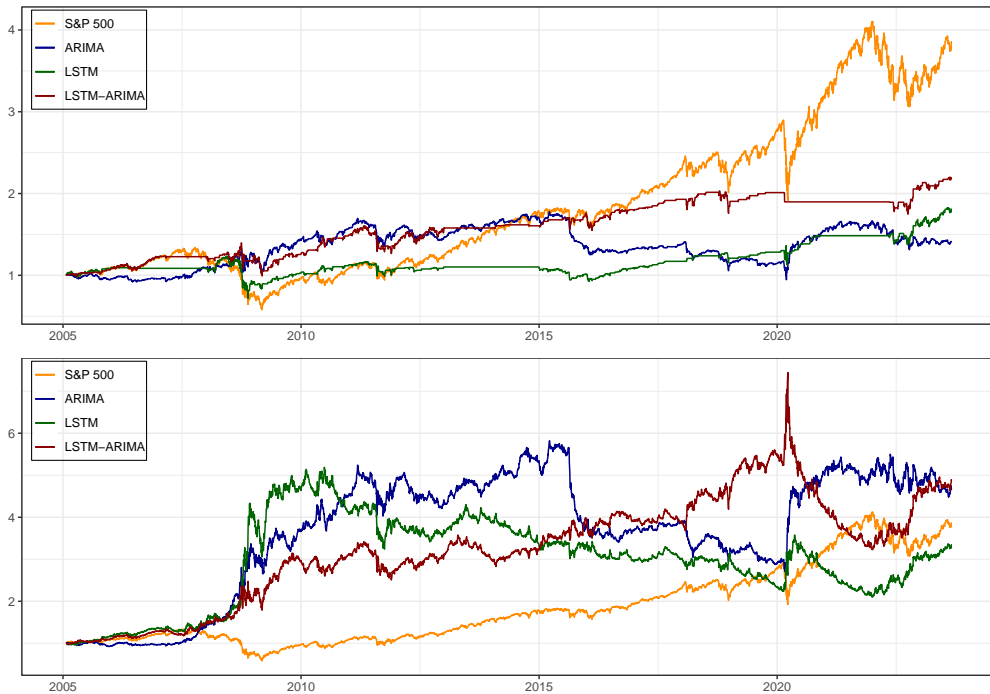
5.1. Base case results

We evaluate the effectiveness of our investment algorithm using out-of-sample data. The S&P 500 equity index began trading on 2005-01-25, the FTSE 100 equity index on 2005-01-13, and the CAC 40 equity index on 2004-12-28, with the trading period continuing until 2023-08-30. Our primary performance evaluation metric is the Modified Information Ratio (IR^{**}), as outlined in Eq. 20. This metric offers a comprehensive assessment, encompassing factors such as annualized return compounded (ARC), return volatility (ASD), and the largest percentage loss experienced by the asset from its peak value before reaching a new peak (MD). This approach allows us to not only assess profitability but also the associated investment risk. For each equity index, we evaluate three algorithms, namely ARIMA, LSTM, and LSTM-ARIMA, by comparing them both among themselves and against the Buy&Hold strategy.

The intervals of the walk-forward optimization process are uniform for all equity indices since they all involve daily frequency. The training period spans 1000 trading days, followed by 250 trading days validation period and a subsequent 250 trading days testing period. Additionally, we consider two strategies: *Long-Only*, where only long positions are allowed, and *Long-Short*, where both long and short positions are permitted. The research flow is also detailed in section 4.6 for reference.

Figure 5, presents the S&P 500 equity index equity curves of all the algorithms for both Long-Only and Long-Short strategies respectively and Table 2 presents the performance metrics for S&P 500 equity index. Based on the evaluation of IR^{**} metrics for both the Long-Only and Long-Short strategies, it can be deduced that the LSTM-ARIMA algorithm outperformed the other algorithms. Additionally, it is noteworthy that all algorithms demonstrated robust performance during the economic downturn of 2008. Furthermore, a remarkable surge in performance is observed during the COVID-19 period in the Long-Short strategy, especially when employing the LSTM-ARIMA algorithm.

Figure 5: The Long-Only and Long-Short Strategy on S&P 500



Note: S&P 500 is the Buy&Hold Strategy. The first plot presents the equity curve for the Long-Only strategy and the second plot presents the equity curve for the Long-Short strategy. The trading starts from 2005-01-25. Each equity curve consists of daily frequency data. The transaction costs are 0.1%.

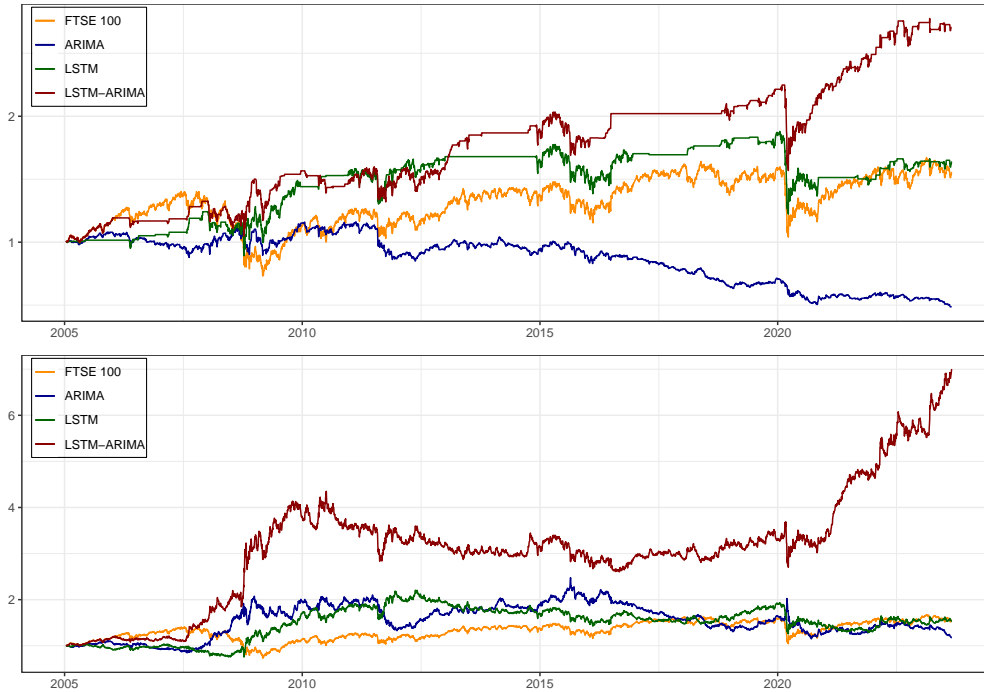
Table 2: Performance metrics for S&P 500

		ARC(%)	ASD(%)	MD(%)	MLD	IR*(%)	IR**(%)
Long Only	S&P 500	7.52	19.58	56.78	1.65	38.43	5.09
	ARIMA	1.89	14.45	46.73	8.45	13.07	0.53
	LSTM	3.26	13.14	41.83	9.8	24.83	1.94
	LSTM-ARIMA	4.32	11.14	28.95	1.67	38.79	5.79
Long Short	S&P 500	7.52	19.58	56.78	1.65	38.43	5.09
	ARIMA	8.66	19.19	54.81	8.44	45.11	7.13
	LSTM	6.71	19.59	59.44	13.16	34.27	3.87
	LSTM-ARIMA	8.92	19.58	56.62	3.44	45.56	7.18

Note: S&P 500 represents the benchmark Buy&Hold Strategy. Trading starts from 2005-01-25. The transaction costs are 0.1%. The best strategy is the one that holds the highest Modified Information Ratio (IR^{**}). Columns with the best corresponding values are denoted in bold.

Figure 6 presents the equity curve of FTSE 100 equity index and Table 3 presents the performance metrics for FTSE 100 equity index. Based on the evaluation of IR^{**} metrics for both the Long-Only and Long-Short strategies, it can be deduced that the LSTM-ARIMA algorithm outperformed the other algorithms for the FTSE 100 equity index. Both the Long-Only and Long-Short strategies, when executed with the LSTM-ARIMA algorithm, display a substantial peak in performance after the year 2020.

Figure 6: The Long-Only and Long-Short Strategy on FTSE 100



Note: FTSE 100 represents the benchmark Buy&Hold Strategy. The first plot presents the equity curve for the Long-Only strategy and the second plot presents the equity curve for the Long-Short strategy. The trading starts from 2005-01-13. Each equity curve consists of daily frequency data. The transaction costs are 0.1%.

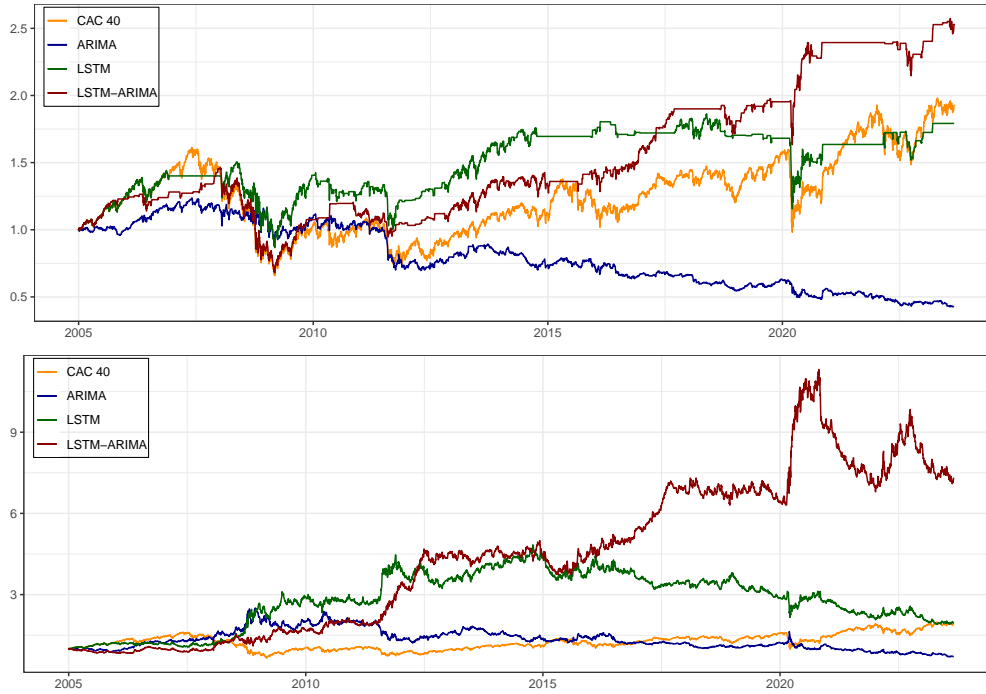
Table 3: Performance metrics for FTSE 100

		ARC(%)	ASD(%)	MD(%)	MLD	IR*(%)	IR**(%)
Long Only	FTSE 100	2.39	18.03	47.83	5.94	13.27	0.66
	ARIMA	-3.78	12.88	58.12	12.55	-29.38	-1.91
	LSTM	2.68	14.32	34.93	3.61	18.75	1.44
	LSTM-ARIMA	5.47	13.79	30.22	0.91	39.71	7.19
Long Short	FTSE 100	2.39	18.03	47.83	5.94	13.27	0.66
	ARIMA	0.84	18.04	53.65	8.03	4.66	0.07
	LSTM	2.28	18.03	42.92	11.3	12.67	0.67
	LSTM-ARIMA	10.98	18.02	40.17	10.89	60.92	16.65

Note: FTSE 100 represents the benchmark Buy&Hold Strategy. Trading starts from 2005-01-13. The transaction costs are 0.1%. The best strategy is the one that holds the highest Modified Information Ratio (IR^{**}). Columns with the best corresponding values are denoted in bold.

Figure 7 presents the equity curve of the CAC 40 equity index and Table 4 presents the performance metrics for CAC 40 equity index. Based on the evaluation of IR^{**} metrics for both the Long-Only and Long-Short strategies, it can be deduced that the LSTM-ARIMA algorithm outperformed the other algorithms for the CAC 40 equity index. Furthermore, we noticed that both Long-Only and Long-Short strategies, when executed with the LSTM-ARIMA algorithm, achieved a high return during the post-Covid time.

Figure 7: The Long-Only and Long-Short Strategy on CAC 40



Note: CAC 40 represents the benchmark Buy&Hold Strategy. The first plot presents the equity curve for the Long-Only strategy and the second plot presents the equity curve for the Long-Short strategy. The trading starts from 2004-12-28. Each equity curve consists of daily frequency data. The transaction costs are 0.1%.

Table 4: Performance statistics for CAC 40

		ARC(%)	ASD(%)	MD(%)	MLD	IR*(%)	IR**(%)
Long Only	CAC 40	3.52	21.44	59.16	14.04	16.43	0.98
	ARIMA	-4.38	15.14	65.53	16.5	-28.9	-1.93
	LSTM	3.12	16.1	42.35	5.38	19.4	1.43
	LSTM-ARIMA	5.02	15.43	53.65	8.33	32.52	3.04
Long Short	CAC 40	3.52	21.44	59.16	14.04	16.43	0.98
	ARIMA	-1.81	21.43	72.02	14.95	-8.45	-0.21
	LSTM	3.56	21.44	60.73	9.01	16.59	0.97
	LSTM-ARIMA	11.06	21.43	39.91	2.91	51.6	14.29

Note: CAC 40 represents the benchmark Buy&Hold Strategy. Trading starts from 2004-12-28. The transaction costs are 0.1%. The best strategy is the one that holds the highest Modified Information Ratio (IR^{**}). Columns with the best corresponding values are denoted in bold.

5.2. Statistical Significance

While LSTM-ARIMA outperformed all the other algorithms for both Long-Only and Long-Short strategies, it would be premature to conclude that the expected values of these strategies' returns distributions surpass those of the benchmarks. Hence, it is prudent to subject them to statistical inference testing to validate their efficacy. We test it using a t-test for paired samples (Devore and Berk (2012)) with the following hypotheses:

$$\begin{cases} H_0 : \mu_d = \mu_{strategy} - \mu_{benchmark} = 0 \\ H_1 : \mu_d > 0 \end{cases} \quad (26)$$

where:

$\mu_{strategy}$ - the expected value of the strategy

$\mu_{benchmark}$ - the expected value of the benchmark

μ_d - the difference between the expected values of the strategy and benchmark returns

Table 5: P-values for the paired t-test

		S&P 500	FTSE 100	CAC 40
Long-Only	ARIMA	0.0362	0.0152	0.0086
	LSTM	0.1248	0.8964	0.6696
	LSTM-ARIMA	0.2420	0.3911	0.9249
Long-Short	ARIMA	0.8730	0.7902	0.4455
	LSTM	0.9046	0.9784	0.9961
	LSTM-ARIMA	0.8569	0.1709	0.3310

Note: The table presents the p-values of the paired t-test. The significance level is set at 10%. P-values less than 0.1 are in bold. Each strategy has been compared with the benchmark Buy&Hold strategy. S&P 500, FTSE 100, and CAC 40 represent the benchmark Buy&Hold Strategy.

Our significance level is set at 10%. If the p-value is lower than 0.1, we reject the null hypotheses; otherwise, we have no grounds to reject it. Based on the p-values provided in Table 5, we infer that the only statistically significant results we attain are observed for the ARIMA model with the Long-Only strategy for all equity indices. In all other scenarios, our findings lack statistical significance. Nonetheless, we opt to conduct an additional test. We formulate a linear regression model as described in eq. 27 and subsequently carry out a right-sided t-test (Wooldridge (2015)) to assess the significance of the intercept, as shown in eq. 28.

$$R_{strategy} = \alpha + \beta \times r_{benchmark} + \epsilon_t \quad (27)$$

$$\begin{cases} H_0 : \alpha = 0 \\ H_1 : \alpha > 0 \end{cases} \quad (28)$$

Table 6: Simple Linear Regression results

		α	SE(α)	t_α	p_α	α	SE(α)	t_β	p_β		
Long-Only	S&P 500	ARIMA	-0.0001	0.0001	-0.9864	0.8380	0.5547	0.0071	77.9073	0.0000	
		LSTM	0.0000	0.0001	-0.0504	0.5201	0.4554	0.0072	63.1896	0.0000	
		LSTM-ARIMA	0.0001	0.0001	0.8697	0.1923	0.3274	0.0068	48.1206	0.0000	
	FTSE 100	ARIMA	-0.0002	0.0001	-2.4608	0.9930	0.5146	0.0072	71.2718	0.0000	
		LSTM	0.0000	0.0001	0.5702	0.2843	0.6344	0.0070	91.1631	0.0000	
		LSTM-ARIMA	0.0002	0.0001	1.9299	0.0268	0.5894	0.0071	83.0157	0.0000	
	CAC 40	ARIMA	-0.0002	0.0001	-2.5518	0.9946	0.5033	0.0072	70.1771	0.0000	
		LSTM	0.0000	0.0001	0.4595	0.3230	0.5647	0.0072	78.8092	0.0000	
		LSTM-ARIMA	0.0001	0.0001	1.2578	0.1043	0.5189	0.0072	71.8728	0.0000	
	Long-Short	S&P 500	ARIMA	0.0004	0.0002	2.0281	0.0213	0.1292	0.0142	9.0957	0.0000
			LSTM	0.0004	0.0002	2.0726	0.0191	-0.1069	0.0145	-7.3498	1.0000
			LSTM-ARIMA	0.0005	0.0002	2.6172	0.0044	-0.1448	0.0145	-10.0100	1.0000
FTSE 100		ARIMA	0.0001	0.0002	0.5591	0.2881	0.0301	0.0146	2.0670	0.0194	
		LSTM	0.0001	0.0002	0.6003	0.2742	0.3932	0.0134	29.3379	0.0000	
		LSTM-ARIMA	0.0005	0.0002	2.8736	0.0020	0.0115	0.0146	0.7874	0.2155	
CAC 40		ARIMA	0.0000	0.0002	0.0855	0.4659	0.0079	0.0145	0.5432	0.2935	
		LSTM	0.0003	0.0002	1.3807	0.0837	-0.1596	0.0143	-11.1765	1.0000	
		LSTM-ARIMA	0.0005	0.0002	2.6859	0.0036	-0.0728	0.0144	-5.0480	1.0000	

Note: The table presents the results of the linear regression as mentioned in eq. 28. The significance level is set at 10%. P-values less than 0.1 are in bold. Each strategy has been compared with the benchmark Buy&Hold strategy. S&P 500, FTSE 100, and CAC 40 represent the benchmark Buy&Hold Strategy.

The findings from Table 6 details the outcomes of the simple linear regression analysis we conducted.} Based on p_α , statistically significant algorithms were observed at a significance level of 10%. Specifically, for the Long-Only strategy, LSTM-ARIMA for the FTSE 100 equity index exhibited statistical significance. Regarding the Long-Short strategy, all algorithms demonstrated statistical significance for the S&P 500 equity index. For the FTSE 100 equity index, only LSTM-ARIMA demonstrated statistical significance. Finally, in the case of the CAC 40 equity index, both LSTM and LSTM-ARIMA exhibited statistical significance.

5.3. Summary

Based on the results presented in Table 2, we see that our novel LSTM-ARIMA algorithm outperformed all the other algorithms for both **Long-Only** and **Long-Short** strategies and all the equity indices. In the case of the S&P 500 equity index, the **Long-Only strategy for LSTM-ARIMA algorithm** obtained a modified information ratio (IR^{**}) of 5.79% and the **Long-Short strategy for LSTM-ARIMA algorithm** of 7.18% . In the case of the FTSE 100 equity index, the **Long-Only strategy for LSTM-ARIMA algorithm** obtained an IR^{**} of 7.19% and the **Long-Short strategy for LSTM-ARIMA algorithm** of 16.65% . Finally, in the case of the CAC 40 equity index, the **Long-Only strategy for LSTM-ARIMA algorithm** obtained an IR^{**} of 3.04% and the **Long-Short strategy for LSTM-ARIMA algorithm** of 14.29% .

Furthermore, based on the results from the previous section, the paired t-test (Table 5) showed that only statistically significant results were observed for the ARIMA model with the Long-Only strategy for all the equity indices. However, summarizing the outcomes of the simple linear regression analysis (Table 6), it becomes evident that the intercept of the LSTM-ARIMA algorithm with the Long-Short strategy returns, significantly exceeded 0 when regressed against the returns from the Buy&Hold strategy.

6. Sensitivity Analysis

This section is specifically focused on addressing the third research question (*RQ3*). Its primary objective is to investigate how changes in specific parameters and hyperparameters influence the output. Through this assessment, we can evaluate the stability and reliability of our investment algorithm. In the case of the ARIMA model, we alter the following parameters:

- The range of ARIMA model order: $(p, d, q) = (0-3, 1, 0-3)$
- The information criterion: Bayesian Information Criterion (BIC)

The following parameters are altered for LSTM and LSTM-ARIMA:

- *the Dropout Rate:* 0.05, 0.1
- *the batch size:* 16, 64

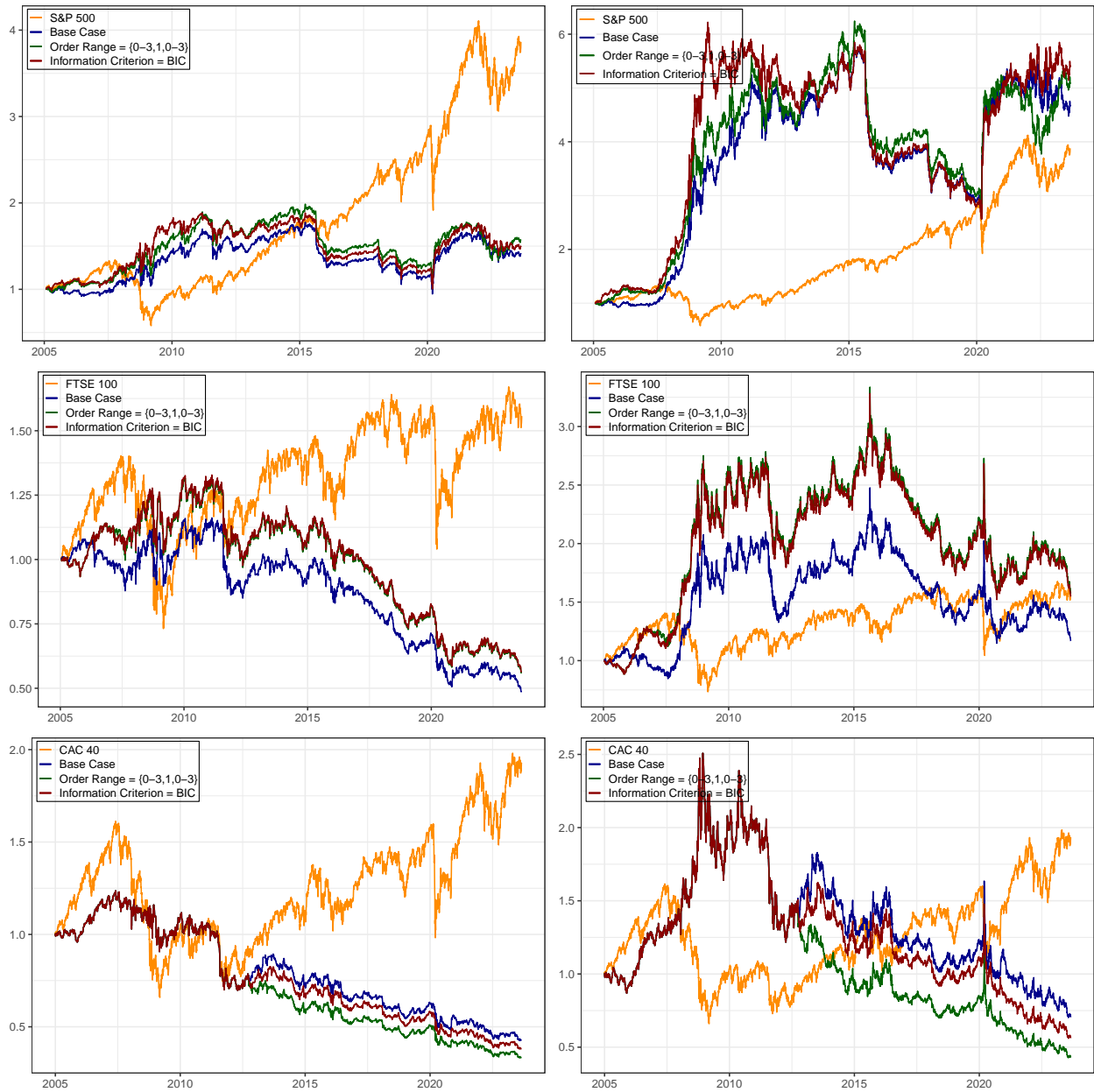
During the sensitivity analysis, only the parameters mentioned are changed and everything else is kept as they were. We continue to use the modified information ratio (IR^{**}) as the main evaluation metric. Additionally, this section is divided into three subsections by our algorithms.

6.1. ARIMA

Figure 8 and Table 7 present the sensitivity analysis results for S&P 500, FTSE 100, and CAC 40 equity indices. In the case of the S&P 500 equity index, based on the IR^{**} metrics, the base case outperforms all the other changes in parameters for the Long-Only strategy. However, under the Long-Short strategy, we notice enhancements in the results when employing the Bayesian Information Criterion (BIC) as the information criterion. In the case of the FTSE 100 equity index, the performance of the ARIMA model appears to be poor. Albeit, for the Long-Short strategy, improvements were seen when narrowing the range of the orders. The model's performance benefited from this adjustment, possibly suggesting that high-order configurations may have been predisposed to overfitting. In the case of the CAC 40 equity index, the changes in the parameters do not yield improved results. The Buy&Hold still outperforms the ARIMA model.

In summary, the results of the ARIMA model used in our algorithmic investment strategy exhibit robustness to changes in the information criterion and the order range setting of the model.

Figure 8: ARIMA Sensitivity Analysis



Note: The figure presents the equity curves for the sensitivity analysis performed on the ARIMA model. S&P 500, FTSE 100, and CAC 40 represent the benchmark Buy&Hold strategy for each index respectively. The base case scenario utilizes the order range $(p,d,q)=0-6, 1, 0-6$ and akaike information criterion (AIC). S&P 500 index trading starts on 2005-01-25, FTSE 100 equity index trading starts on 2005-01-13, and CAC 40 equity index trading starts on 2004-12-28. Each equity curve consists of daily frequency data. The transaction costs are 0.1%. The best values are in bold and are bolded with respect to the base case scenario.

Table 7: ARIMA Sensitivity Analysis performance metrics

		ARC(%)	ASD(%)	MD(%)	MLD	IR*(%)	IR**(%)
S&P 500		7.52	19.58	56.78	1.65	38.43	5.09
Long Only	Base Case	4.32	11.14	28.95	1.67	38.79	5.79
	Order Range = {0-3,1,0-3}	2.47	14.45	46.73	8.45	17.11	0.91
	Information Criterion = BIC	2.21	14.35	46.73	8.45	15.41	0.73
Long Short	Base Case	8.92	19.58	56.62	3.44	45.56	7.18
	Order Range = {0-3,1,0-3}	9.15	19.57	56.24	8.44	46.79	7.62
	Information Criterion = BIC	9.52	19.57	58.85	14.18	48.62	7.86
FTSE 100		2.39	18.03	47.83	5.94	13.27	0.66
Long Only	Base Case	-3.78	12.88	58.12	12.55	-29.38	-1.91
	Order Range = {0-3,1,0-3}	-3.06	12.9	57.55	12.55	-23.71	-1.26
	Information Criterion = BIC	-3.02	12.9	57.55	12.55	-23.4	-1.23
Long Short	Base Case	0.84	18.04	53.65	8.03	4.66	0.07
	Order Range = {0-3,1,0-3}	2.46	18.04	53.65	8.03	13.66	0.63
	Information Criterion = BIC	2.37	18.04	53.65	8.03	13.15	0.58
CAC 40		3.52	21.44	59.16	14.04	16.43	0.98
Long Only	Base Case	-4.38	15.14	65.53	16.5	-28.9	-1.93
	Order Range = {0-3,1,0-3}	-5.62	15.1	73.12	16.5	-37.24	-2.86
	Information Criterion = BIC	-4.95	15.07	69.27	16.5	-32.86	-2.35
Long Short	Base Case	-1.81	21.43	72.02	14.95	-8.45	-0.21
	Order Range = {0-3,1,0-3}	-4.35	21.43	82.97	14.95	-20.28	-1.06
	Information Criterion = BIC	-2.97	21.43	77.67	14.95	-13.86	-0.53

Note: The table shows the performance metrics for the sensitivity analysis performed on the ARIMA model. S&P 500, FTSE 100, and CAC 40 represent the benchmark Buy&Hold strategy for each index respectively. S&P 500 index trading starts on 2005-01-25, FTSE 100 equity index trading starts on 2005-01-13, and CAC 40 equity index trading starts on 2004-12-28. In the base case scenario, the Dropout rate is set to 0.075 and the Batch Size is set to 32. The transaction costs are 0.1%. The best values are in bold and are bolded with respect to the base case scenario.

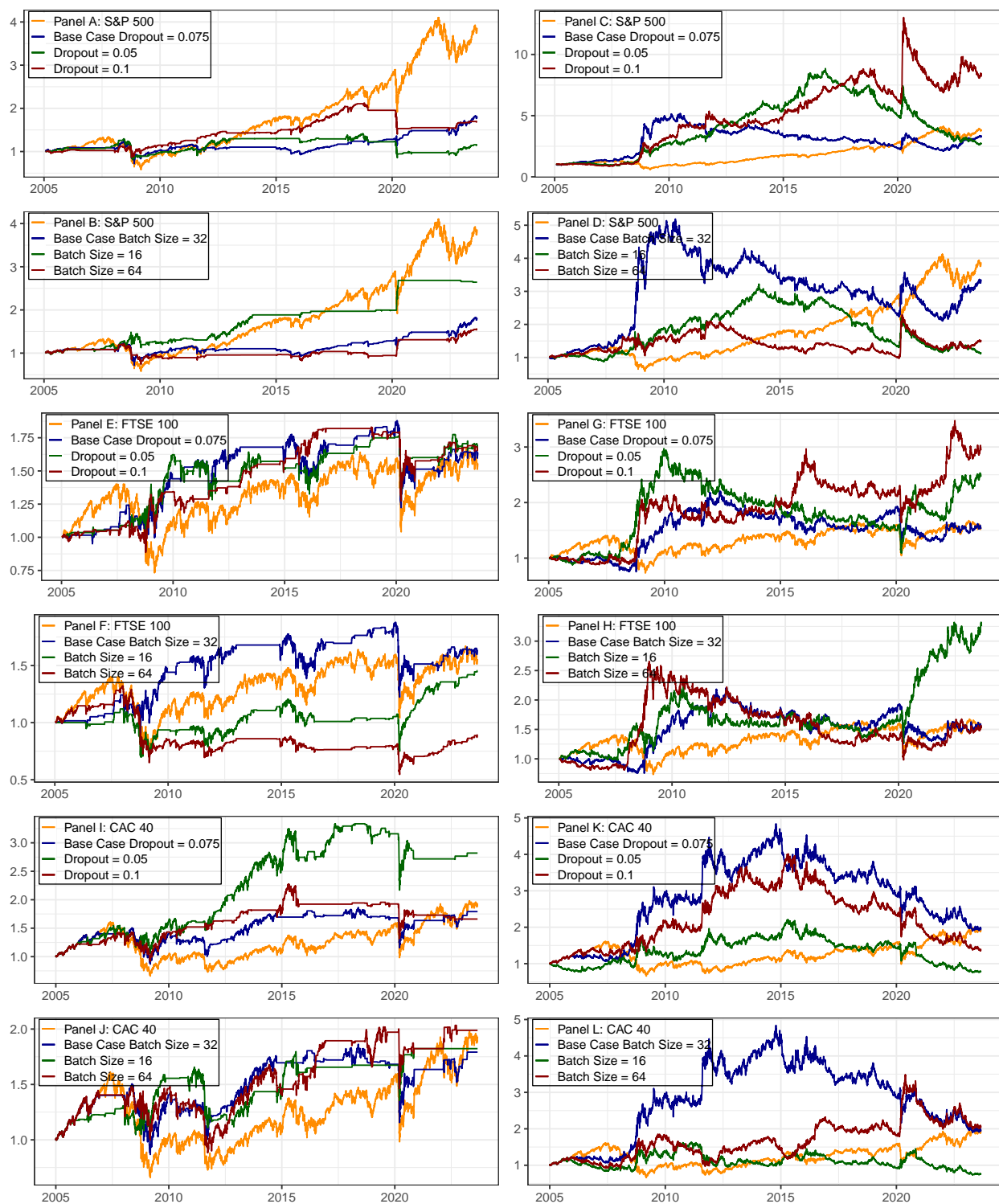
6.2. LSTM

Figure 9 and Table 8 present the results of a sensitivity analysis conducted on the S&P 500, FTSE 100, and CAC 40 equity indices using the LSTM model. In the case of the S&P 500 equity index (presented in Panel A and Panel C), it is observed that an increase in the dropout rate results in a higher modified information ratio (IR^{**}) for both the Long-Only and the Long-Short strategy. Additionally, as indicated in Panel B and Panel D, it is inferred that a smaller batch size yields a higher IR^{**} for the Long-Only strategy, whereas for the Long-Short strategy smaller batch size made it significantly worse.

In the case of the FTSE 100 equity index, indicated in Panel E and Panel G, it is evident that smaller dropout rates yield better results in the case of the Long-Only strategy, while higher dropout rates are more effective for the Long-Short strategy. Furthermore, as indicated in Panel F and Panel H, it is apparent that the Long-Short strategy exhibits higher IR^{**} when using a smaller batch size, while the Long-Only strategy achieves optimal performance when utilizing the base case scenario.

The results for the CAC 40 equity index, presented in Panel I and Panel K, indicate that a smaller dropout rate leads to a higher IR^{**} for the Long-Only strategy. However, this trend is not observed for the Long-Short strategy, as the base case scenario continues to deliver the best results. Furthermore, based on the observations in Panel J and Panel L, it is noticed that a higher batch size results in higher annualized returns compounded (ARC) for both the Long-Only and the Long-Short strategies. However, when considering the IR^{**} metric, the improvement is only evident in the Long-Short strategy, whereas the Long-Only strategy achieves a higher IR^{**} with a smaller batch size.

Figure 9: LSTM Sensitivity Analysis for S&P 500



Note: The figure presents the equity curves for the sensitivity analysis performed on the LSTM model. S&P 500, FTSE 100, and CAC 40 represent the benchmark Buy&Hold strategy for each index respectively. S&P 500 index trading starts on 2005-01-25, FTSE 100 equity index trading starts on 2005-01-13, and CAC 40 equity index trading starts on 2004-12-28. Each equity curve consists of daily frequency data. The transaction costs are 0.1%. The best values are in bold and are bolded with respect to the base case scenario.

Table 8: LSTM Sensitivity Analysis performance metrics

		ARC(%)	ASD(%)	MD(%)	MLD	IR*(%)	IR**(%)
	S&P 500	7.52	19.58	56.78	1.65	38.43	5.09
Long Only	Base Case (Dropout = 0.075)	3.26	13.14	41.83	9.8	24.83	1.94
Panel A: Dropout Rate	Dropout = 0.05	0.79	12.68	39.43	4.94	6.24	0.13
	Dropout = 0.1	2.87	11.4	33.93	4.86	25.2	2.13
Panel B: Batch Size	Base Case (Batch Size = 32)	3.26	13.14	41.83	9.8	24.83	1.94
	Batch Size = 16	5.37	10.45	24.07	3.96	51.37	11.46
	Batch Size = 64	2.42	10.94	38.72	11.77	22.14	1.39
Long Short	Base Case (Dropout = 0.075)	6.71	19.59	59.44	13.16	34.27	3.87
Panel C: Dropout Rate	Dropout = 0.05	5.42	19.59	70.44	6.8	27.66	2.13
	Dropout = 0.1	12.0	19.59	47.23	3.4	61.26	15.57
Panel D: Batch Size	Base Case (Batch Size = 32)	6.71	19.59	59.44	13.16	34.27	3.87
	Batch Size = 16	0.72	19.59	65.51	9.57	3.67	0.04
	Batch Size = 64	2.26	19.58	53.48	8.23	11.55	0.49
	FTSE 100	2.39	18.03	47.83	5.94	13.27	0.66
Long Only	Base Case (Dropout = 0.075)	2.68	14.32	34.93	3.61	18.75	1.44
Panel E: Dropout Rate	Dropout = 0.05	2.85	13.1	27.53	2.45	21.78	2.26
	Dropout = 0.1	2.78	11.99	30.44	4.77	23.17	2.12
Panel F: Batch Size	Base Case (Batch Size = 32)	2.68	14.32	34.93	3.61	18.75	1.44
	Batch Size = 16	2.03	13.84	37.95	5.86	14.66	0.78
	Batch Size = 64	-0.59	14.09	58.72	15.68	-4.2	-0.04
Long Short	Base Case (Dropout = 0.075)	2.28	18.03	42.92	11.3	12.67	0.67
Panel G: Dropout Rate	Dropout = 0.05	4.96	18.03	63.39	13.7	27.49	2.15
	Dropout = 0.1	5.96	18.05	44.81	6.09	33.0	4.39
Panel H: Batch Size	Base Case (Batch Size = 32)	2.28	18.03	42.92	11.3	12.67	0.67
	Batch Size = 16	6.53	18.03	43.77	10.2	36.23	5.4
	Batch Size = 64	2.28	18.04	63.08	14.69	12.63	0.46
	CAC 40	3.52	21.44	59.16	14.04	16.43	0.98
Long Only	Base Case (Dropout = 0.075)	3.12	16.1	42.35	5.38	19.4	1.43
Panel I: Dropout Rate	Dropout = 0.05	5.62	16.79	34.96	5.52	33.48	5.38
	Dropout = 0.1	2.71	14.56	42.18	8.49	18.62	1.2
Panel J: Batch Size	Base Case (Batch Size = 32)	3.12	16.1	42.35	5.38	19.4	1.43
	Batch Size = 16	3.22	13.91	33.09	4.06	23.13	2.25
	Batch Size = 64	3.69	16.98	40.42	3.6	21.74	1.99
Long Short	Base Case (Dropout = 0.075)	3.56	21.44	60.73	9.01	16.59	0.97
Panel K: Dropout Rate	Dropout = 0.05	-1.34	21.43	65.66	8.49	-6.26	-0.13
	Dropout = 0.1	1.75	21.44	66.37	8.49	8.16	0.22
Panel L: Batch Size	Base Case (Batch Size = 32)	3.56	21.44	60.73	9.01	16.59	0.97
	Batch Size = 16	-1.5	21.46	55.1	12.72	-7.0	-0.19
	Batch Size = 64	3.82	21.43	47.78	3.35	17.85	1.43

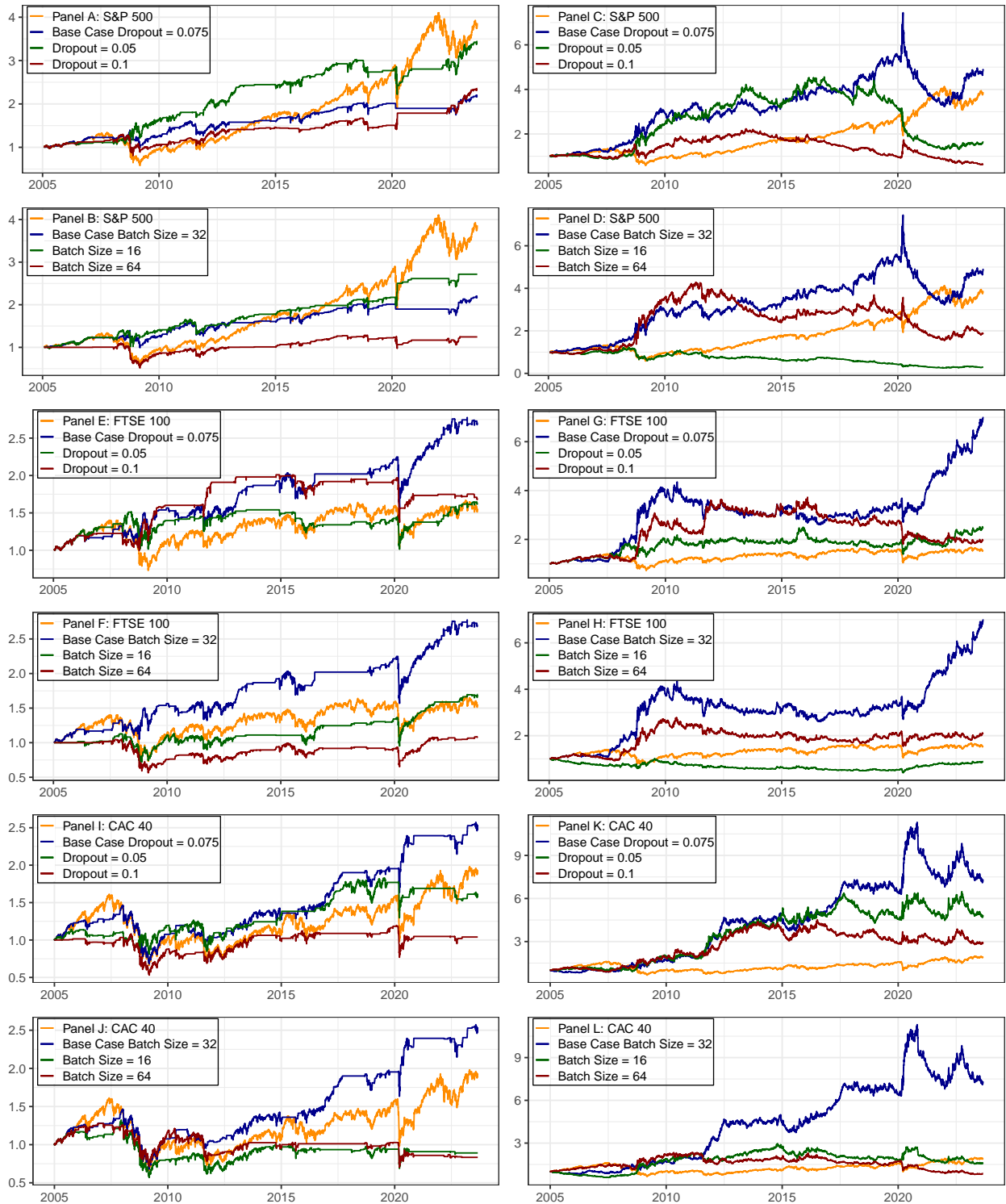
Note: The table shows the performance metrics for the sensitivity analysis performed on the LSTM model. S&P 500, FTSE 100, and CAC 40 represent the benchmark Buy&Hold strategy for each index respectively. S&P 500 index trading starts on 2005-01-25, FTSE 100 equity index trading starts on 2005-01-13, and CAC 40 equity index trading starts on 2004-12-28. In the base case scenario, the Dropout rate is set to 0.075 and the Batch Size is set to 32. The transaction costs are 0.1%. The best values are in bold and are bolded with respect to the base case scenario.

6.3. LSTM-ARIMA

Figure 10 and Table 9 present the results of a sensitivity analysis conducted on the S&P 500, FTSE 100, and CAC 40 equity indices using the LSTM-ARIMA model. The results reveal that the highest values of the modified information ratio (IR^{**}) are achieved in the base case scenario for the FTSE 100 and CAC 40 equity indices, as depicted in Panel E to Panel L.

However, in the case of the S&P 500 equity index, a higher IR^{**} metric is attained for the Long-Only strategy when the dropout rate is decreased (Panel A) and when the batch size is reduced (Panel B). In contrast, for the Long-Short strategy, the base case scenario proves to be the most effective in terms of the IR^{**} metric.

Figure 10: LSTM-ARIMA Sensitivity Analysis for S&P 500



Note: The figure presents the equity curves for the sensitivity analysis performed on the LSTM-ARIMA model. S&P 500, FTSE 100, and CAC 40 represent the benchmark Buy&Hold strategy for each index respectively. S&P 500 index trading starts on 2005-01-25, FTSE 100 equity index trading starts on 2005-01-13, and CAC 40 equity index trading starts on 2004-12-28. Each equity curve consists of daily frequency data. The transaction costs are 0.1%.

Table 9: LSTM-ARIMA Sensitivity Analysis performance metrics

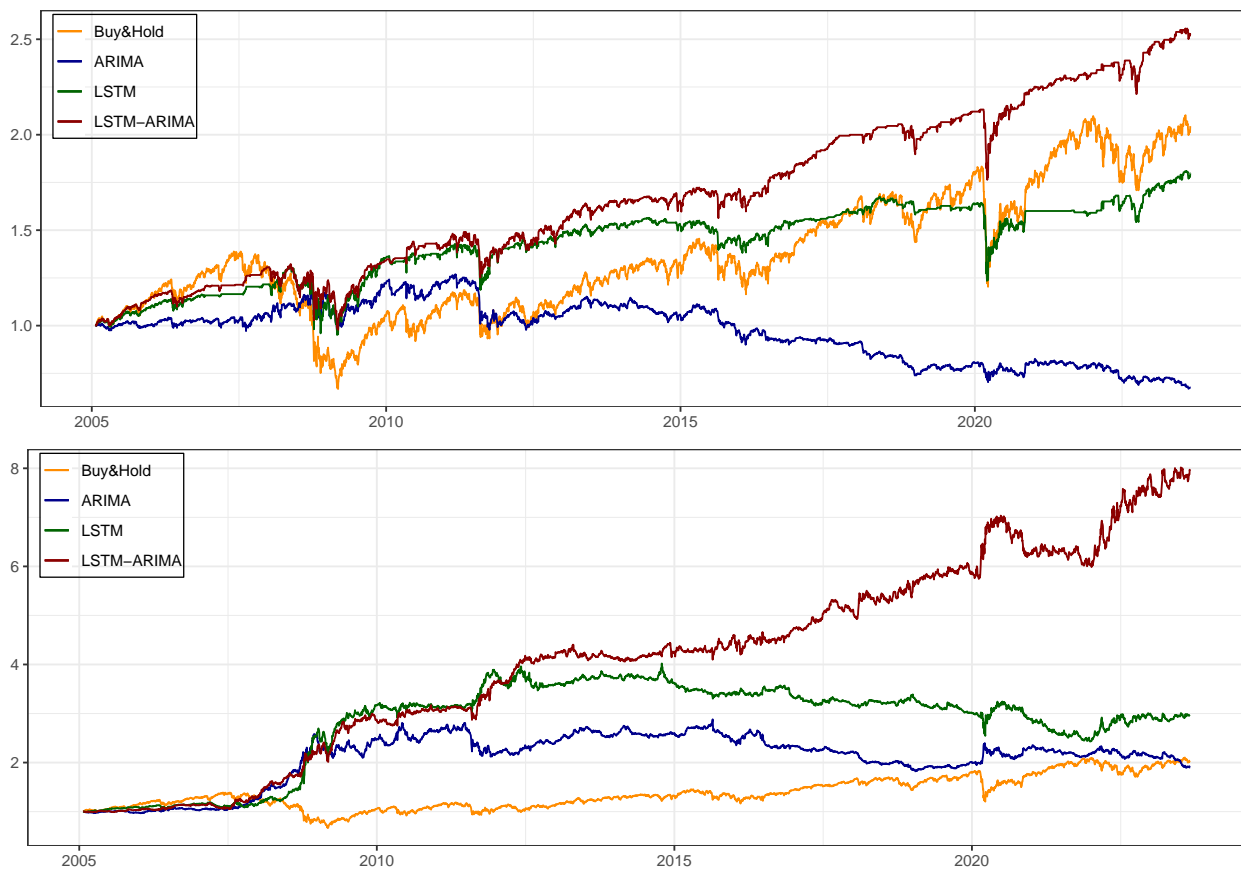
		ARC(%)	ASD(%)	MD(%)	MLD	IR*(%)	IR**(%)
	S&P 500	7.52	19.58	56.78	1.65	38.43	5.09
Long Only	Base Case (Dropout = 0.075)	4.32	11.14	28.95	1.67	38.79	5.79
Panel A: Dropout Rate	Dropout = 0.05	6.88	13.7	26.63	3.92	50.25	12.99
	Dropout = 0.1	4.72	13.5	37.99	3.66	34.99	4.35
Panel B: Batch Size	Base Case (Batch Size = 32)	4.32	11.14	28.95	1.67	38.79	5.79
	Batch Size = 16	5.53	12.03	27.83	1.54	45.92	9.11
	Batch Size = 64	1.19	14.44	52.99	7.54	8.23	0.18
Long Short	Base Case (Dropout = 0.075)	8.92	19.58	56.62	3.44	45.56	7.18
Panel C: Dropout Rate	Dropout = 0.05	2.63	19.6	71.5	6.8	13.4	0.49
	Dropout = 0.1	-2.5	19.6	72.1	10.17	-12.73	-0.44
Panel D: Batch Size	Base Case (Batch Size = 32)	8.92	19.58	56.62	3.44	45.56	7.18
	Batch Size = 16	-6.58	19.58	79.19	15.19	-33.62	-2.79
	Batch Size = 64	3.35	19.59	63.83	12.32	17.12	0.9
	FTSE 100	2.39	18.03	47.83	5.94	13.27	0.66
Long Only	Base Case (Dropout = 0.075)	5.47	13.79	30.22	0.91	39.71	7.19
Panel E: Dropout Rate	Dropout = 0.05	2.66	13.36	34.18	9.02	19.9	1.55
	Dropout = 0.1	2.85	10.94	31.62	8.88	26.1	2.35
Panel F: Batch Size	Base Case (Batch Size = 32)	5.47	13.79	30.22	0.91	39.71	7.19
	Batch Size = 16	2.88	13.63	36.25	0.73	21.15	1.68
	Batch Size = 64	0.44	14.25	46.13	13.54	3.07	0.03
Long Short	Base Case (Dropout = 0.075)	10.98	18.02	40.17	10.89	60.92	16.65
Panel G: Dropout Rate	Dropout = 0.05	4.91	18.02	44.39	7.59	27.24	3.01
	Dropout = 0.1	3.61	18.03	51.81	7.56	20.03	1.4
Panel H: Batch Size	Base Case (Batch Size = 32)	10.98	18.02	40.17	10.89	60.92	16.65
	Batch Size = 16	-0.63	18.03	60.17	14.14	-3.49	-0.04
	Batch Size = 64	3.98	18.04	44.77	13.2	22.08	1.96
	CAC 40	3.52	21.44	59.16	14.04	16.43	0.98
Long Only	Base Case (Dropout = 0.075)	5.02	15.43	53.65	8.33	32.52	3.04
Panel I: Dropout Rate	Dropout = 0.05	2.55	15.34	33.09	4.24	16.61	1.28
	Dropout = 0.1	0.21	16.86	51.01	7.47	1.23	0.01
Panel J: Batch Size	Base Case (Batch Size = 32)	5.02	15.43	53.65	8.33	32.52	3.04
	Batch Size = 16	-0.61	17.15	56.9	15.95	-3.53	-0.04
	Batch Size = 64	-0.95	15.78	45.06	16.98	-6.05	-0.13
Long Short	Base Case (Dropout = 0.075)	11.06	21.43	39.91	2.91	51.6	14.29
Panel K: Dropout Rate	Dropout = 0.05	8.64	21.44	32.86	3.09	40.31	10.6
	Dropout = 0.1	5.7	21.44	39.43	7.27	26.57	3.84
Panel L: Batch Size	Base Case (Batch Size = 32)	11.06	21.43	39.91	2.91	51.6	14.29
	Batch Size = 16	2.4	21.46	48.23	9.01	11.19	0.56
	Batch Size = 64	-0.89	21.44	65.76	12.72	-4.15	-0.06

Note: The table shows the performance metrics for the sensitivity analysis performed on the LSTM-ARIMA model. S&P 500, FTSE 100, and CAC 40 represent the benchmark Buy&Hold strategy for each index respectively. S&P 500 index trading starts on 2005-01-25, FTSE 100 equity index trading starts on 2005-01-13, and CAC 40 equity index trading starts on 2004-12-28. In the base case scenario, the Dropout rate is set to 0.075 and the Batch Size is set to 32. The transaction costs are 0.1%. The best values are in bold and are bolded with respect to the base case scenario.

7. Ensembled AIS

We create an ensemble AIS to diversify the results of our investment algorithms among all the financial instruments. The idea is that we invest 1 dollar in each financial instrument and then test the Long-Only and Long-Short strategy. We assume that the instruments are perfectly divisible and that we assign a weight of $\frac{1}{3}$ to each equity index. The trading in this case starts on *2005-01-25* and goes until *2023-08-30*. Figure 11 and Table 10 present the results for our ensemble AIS. When we aggregate all the equity indices, there is a notable improvement in our results. We achieve a significantly enhanced risk-adjusted return (IR^{**}). The LSTM-ARIMA model combined with the Long-Short strategy outperforms all other approaches, yielding an impressive IR^{**} of 70.54%.

Figure 11: The Long-Only and Long-Short Strategy of our ensemble AIS



Note: The equity curves are a weighted average of the equity curves of all the equity indices. The weight is equal to $1/3$. The first plot presents the equity curve for the Long-Only strategy and the second plot presents the equity curve for the Long-Short strategy. The trading lasts from *2005-01-25* until *2023-08-30*. Each equity curve consists of daily frequency data. The transaction costs are 0.1%.

Table 10: Performance metrics for ensemble AIS

		ARC(%)	ASD(%)	MD(%)	MLD	IR*(%)	IR**(%)
Long Only	S&P 500	3.92	17.43	51.87	7.7	22.48	1.7
	ARIMA	-2.09	10.93	47.24	12.47	-19.09	-0.84
	LSTM	3.21	11.61	27.14	4.0	27.65	3.27
	LSTM-ARIMA	5.12	10.43	26.06	0.42	49.08	9.64
Long Short	S&P 500	3.92	17.43	51.87	7.7	22.48	1.7
	ARIMA	3.51	12.7	36.79	8.02	27.68	2.64
	LSTM	6.0	12.53	39.57	8.86	47.85	7.25
	LSTM-ARIMA	11.82	11.96	16.57	1.87	98.86	70.54

Note: The trading lasts from 2005-01-25 until 2023-08-30. The transaction costs are 0.1%. The best strategy is the one that holds the highest Modified Information Ratio (IR^{**}). Columns with the best corresponding values are denoted in bold.

8. Conclusion

This study aimed to create a strategy using LSTM-ARIMA that performs better than the individual algorithms. To assess the performance of tested approaches, we created three algorithmic investment strategies based on ARIMA, LSTM, and LSTM-ARIMA models. We conducted hyperparameter tuning using a random search. The walk-forward optimization was applied to perform the model training and backtesting. The best model was chosen based on the conditions presented in section 4.7. Next, we generated buy/sell signals using the condition explained in Section 4.8. The algorithmic investment strategy was tested on three different equity indices: S&P 500, FTSE 100, and CAC 40 on daily frequency data between the period of January 2000 and August 2023. The algorithm predicted the next day’s closing price based on the historical data and was classified as a regression problem.

According to our initial hypotheses, the LSTM-ARIMA model was expected to outperform other algorithms in the majority of cases. The LSTM-ARIMA model indeed outperformed all the other algorithms in all the cases, the summary can be read in section 5.2 where it is outlined which strategy performed the best based on the IR^{**} metric. Therefore, we conclude that we have no grounds to reject our hypotheses. The answers to the research questions stated in the *Introduction* are presented below:

RQ1. Are the investment algorithms robust to changes in the asset?

Our hybrid model outperforms all the other models across all equity indices, though with varying performance metrics. The varying performance metrics across different assets indicate that our algorithms are not robust to changes in the asset. This suggests that the algorithms' performance is sensitive to the specific characteristics and dynamics of each asset, requiring further modifications or adaptations to improve their effectiveness across different assets.

RQ2. Does LSTM-ARIMA perform better than the models individually?

Based on the findings presented in Section 5.1 and the performance metrics provided in Tables 2, 3, and 4, it can be concluded that LSTM-ARIMA outperforms the other models examined.

RQ3. Are the algorithmic investment strategies robust to changes in the model hyperparameters?

In section 6, it becomes apparent that modifications to the hyperparameters have an impact on the results, indicating a lack of robustness. This implies that the model's performance is sensitive to the specific choices made for hyperparameter settings.

RQ4. Does the Long-Only or Long-Short strategy outperform the Buy&Hold?

Based on the analysis presented in Section 5.1 and the data provided in Tables 2, 3, and 4, several findings can be observed. In the case of the S&P 500 equity index, it has been observed that both the Long-Only and Long-Short strategies implemented with the LSTM-ARIMA model, along with the Long-Short strategy implemented with the ARIMA model outperform the Buy&Hold strategy. In the case of the FTSE 100 equity index, the Long-Only and Long-Short strategies implemented with the LSTM-ARIMA and LSTM model outperform the Buy&Hold strategy. In the case of the CAC 40 equity index, it has been noticed that both the Long-Only and Long-Short strategies implemented with the LSTM-ARIMA model, along with the Long-Only strategy implemented with the LSTM model outperform the Buy&Hold strategy.

This study has made a valuable contribution to the existing literature by introducing a hybrid approach that combines modern forecasting models, such as LSTM and ARIMA, for algorithmic investment strategies. Previous research has explored this hybrid approach in various domains. For instance, Bali et al. (2020) utilized LSTM-ARIMA to forecast wind speed, Fan et al. (2021) optimized the hybrid model for well production forecasting, Dave et al. (2021) employed it to

forecast exports in Indonesia, and Arnob et al. (2019) used the hybrid approach for forecasting the Dhaka stock exchange (DSE). However, our study stands out by applying this model specifically to algorithmic investment strategies, a relatively uncommon application for this hybrid approach.

In the base case scenario outlined in Section 5, our hybrid approach outperformed all the models, aligning with our expectations. However, through the course of our sensitivity analysis, it became apparent that there is room for further enhancements by adjusting the dropout rate and the batch size.

There are several potential directions to expand upon this research. Firstly, it would be valuable to investigate whether utilizing returns as inputs instead of the closing price impacts the outcomes. Secondly, the sensitivity analysis revealed noteworthy improvements in the results by reducing the dropout rate and the batch size S&P 500 equity index Long-Only strategy. Therefore, conducting a more comprehensive sensitivity analysis by examining a broader range of dropout rates and sizes would be beneficial. Thirdly, it is worth evaluating the changes in results when considering a binary cross-entropy problem. Finally, it is important to explore the use of a specific threshold that indicates when to change the position in both the Long-Only and Long-Short strategies.

References

- [1] Al-Gounmeem, R. S., and Ismail, M. T. (2020). Forecasting the exchange rate of the jordanian dinar versus the US dollar using a box-jenkins seasonal ARIMA model. *International Journal of Mathematics and Computer Science*, 15(1), 27–40.
- [2] Ariyo, A. A., Adewumi, A. O., and Ayo, C. K. (2014). Stock price prediction using the ARIMA model. *2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation*, 106–112.
- [3] Arnob, R. I., Alam, R., and Alam, A. E. (2019). *Dhaka stock market analysis with ARIMA-LSTM hybrid model* [PhD thesis]. Brac University.
- [4] Bali, V., Kumar, A., and Gangwar, S. (2020). A novel approach for wind speed forecasting using LSTM-ARIMA deep learning models. *International Journal of Agricultural and Environmental Information Systems (IJAEIS)*, 11(3), 13–30.
- [5] Baranochnikov, I., and Ślepaczuk, R. (2022). *A comparison of LSTM and GRU architectures with the novel walk-forward approach to algorithmic investment strategy*.
- [6] Bhandari, H. N., Rimal, B., Pokhrel, N. R., Rimal, R., Dahal, K. R., and Khatri, R. K. (2022). Predicting stock market index using LSTM. *Machine Learning with Applications*, 9, 100320.
- [7] Box, G. E. P., and Jenkins, G. M. (1976). *Time series analysis: „Forecasting and control”*.
- [8] Bui, Q., and Ślepaczuk, R. (2022). Applying hurst exponent in pair trading strategies on nasdaq 100 index. *Physica A: Statistical Mechanics and Its Applications*, 592, 126784.
- [9] Castellano Gómez, S., and Ślepaczuk, R. (2021). *Robust optimisation in algorithmic investment strategies*.
- [10] Dave, E., Leonardo, A., Jeanice, M., and Hanafiah, N. (2021). Forecasting indonesia exports using a hybrid model ARIMA-LSTM. *Procedia Computer Science*, 179, 480–487.

- [11] Devi, B. U., Sundar, D., and Alli, P. (2013). An effective time series analysis for stock trend prediction using ARIMA model for fifty midcap-50. *International Journal of Data Mining & Knowledge Management Process*, 3(1), 65.
- [12] Devore, J., and Berk, K. (2012). *Modern mathematical statistics with applications*. 285.
- [13] Fan, D., Sun, H., Yao, J., Zhang, K., Yan, X., and Sun, Z. (2021). Well production forecasting based on ARIMA-LSTM model considering manual operations. *Energy*, 220, 119708.
- [14] Grudniewicz, J., and Ślepaczuk, R. (2023). Application of machine learning in algorithmic investment strategies on global stock markets. *Research in International Business and Finance*, 66, 102052.
- [15] Hiransha, M., Gopalakrishnan, E. A., Menon, V. K., and Soman, K. (2018). NSE stock market prediction using deep-learning models. *Procedia Computer Science*, 132, 1351–1362.
- [16] Hochreiter, S., and Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780.
- [17] Karim, M. E., Foysal, M., and Das, S. (2022). Stock price prediction using bi-LSTM and GRU-based hybrid deep learning approach. *Proceedings of Third Doctoral Symposium on Computational Intelligence: DoSCI 2022*, 701–711.
- [18] Kijewski, M., and Ślepaczuk, R. (2020). Predicting prices of s&P500 index using classical methods and recurrent neural networks. *Work. Pap. Fac. Econ. Sci. Univ. Wars.*
- [19] Kryńska, K., and Ślepaczuk, R. (2022). *Daily and intraday application of various architectures of the LSTM model in algorithmic investment strategies on bitcoin and the s&P 500 index*.
- [20] Li, Y., Wu, K., and Liu, J. (2023). Self-paced ARIMA for robust time series prediction. *Knowledge-Based Systems*, 269, 110489.
- [21] Malladi, R. K., and Dheeriyaa, P. L. (2021). Time series analysis of cryptocurrency returns and volatilities. *Journal of Economics and Finance*, 45(1), 75–94.
- [22] Michańków, J., Sakowski, P., and Ślepaczuk, R. (2022). LSTM in algorithmic investment strategies on BTC and s&P500 index. *Sensors*, 22(3), 917.
- [23] Mizdrakovic, V., Kljajic, M., Zivkovic, M., Bacanin, N., Jovanovic, L., Deveci, M., and Pedrycz, W. (2024). Forecasting bitcoin: Decomposition aided long short-term memory based time series modeling and its explanation with shapley values. *Knowledge-Based Systems*, 299.
- [24] Mondal, P., Shit, L., and Goswami, S. (2014). Study of effectiveness of time series modeling (ARIMA) in forecasting stock prices. *International Journal of Computer Science, Engineering and Applications*, 4(2), 13.
- [25] Nelson, D. M., Pereira, A. C., and De Oliveira, R. A. (2017). Stock market's price movement prediction with LSTM neural networks. *2017 International Joint Conference on Neural Networks (IJCNN)*, 1419–1426.
- [26] Oyewola, D. O., Akinwunmi, S. A., and Omotehinwa, T. O. (2024). Deep LSTM and LSTM-attention q-learning based reinforcement learning in oil and gas sector prediction. *Knowledge-Based Systems*, 284, 111290.
- [27] Roondiwala, M., Patel, H., and Varma, S. (2017). Predicting stock prices using LSTM. *International Journal of Science and Research (IJSR)*, 6(4), 1754–1756.
- [28] Rumelhart, D. (1986). Hinton g. E and williams rj, ". *Learning Representations by Backpropagating Errors*, 533–536.
- [29] Senneset, K., and Gultvedt, M. (2020). *Something old, something new: A hybrid approach with ARIMA and LSTM to increase portfolio stability* [Master's thesis].
- [30] Siami-Namini, S., Tavakoli, N., and Namin, A. S. (2018). A comparison of ARIMA and LSTM in forecasting time series. *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, 1394–1401.

- [31] Suharsono, A., Aziza, A., and Pramesti, W. (2017). Comparison of vector autoregressive (VAR) and vector error correction models (VECM) for index of ASEAN stock price. *AIP Conference Proceedings*, 1913, 020032.
- [32] Turing. (2023). *Recurrent neural networks (RNN) and LSTM: Overview and uses*. Turing. <https://www.turing.com/kb/recurrent-neural-networks-and-lstm>
- [33] Vo, N., and Ślepaczuk, R. (2022). Applying hybrid ARIMA-SGARCH in algorithmic investment strategies on s&P500 index. *Entropy*, 24(2), 158.
- [34] Wooldridge, J. M. (2015). *Introductory econometrics: A modern approach*.
- [35] Xiong, T., Bao, Y., and Hu, Z. (2014). Multiple-output support vector regression with a firefly algorithm for interval-valued stock price index forecasting. *Knowledge-Based Systems*, 55.
- [36] Zhang, X., Liang, X., Zhiyuli, A., Zhang, S., Xu, R., and Wu, B. (2019). At-lstm: An attention-based lstm model for financial time series prediction. *IOP Conference Series: Materials Science and Engineering*, 569, 052037.