LONG SHORT-TERM MEMORY NETWORKS FOR EQUITY RETURN FORECASTING

By

DAVID JOHAN WAIJERS

Student ID: 478822

A Bachelor's thesis submitted in partial fulfillment of
the requirements for the degree of

INTERNATIONAL BACHELOR ECONOMICS AND BUSINESS ECONOMICS

ERASMUS UNIVERSITY ROTTERDAM
Erasmus School of Economics

Date final version: MAY 10 2020

LONG SHORT-TERM MEMORY NETWORKS FOR EQUITY RETURN FORECASTING

Abstract

by David Johan Waijers, Erasmus University Rotterdam
May 2020

Thesis Advisor: Dr. X. Ma

This thesis deploys Long Short-Term Memory (LSTM) and Gated Recurrent Units (GRU) neural networks to predict out-of-sample equity returns using 51 factor portfolios for 30 of the constituent stocks of the Standard & Poor's 500 index from 2013 to 2019. The results show that both LSTM and GRU neural networks are able to predict directional changes with accuracies significantly higher than 50 percent, with directional accuracies ranging from 54 to 82 percent. Two trading strategies are constructed on the basis of the directional predictions of equity returns. Equally-weighted portfolios of the LSTM strategies were able to generate out-of-sample annualized returns of 21.94 and 35.12 per cent, as well as monthly alphas of 1.22 and 2.46 per cent, respectively, after transaction costs. Moreover, the LSTM and GRU strategies had lower exposure to systematic risk while generating significantly higher returns than holding an equally-weighted buy-and-hold portfolio or the S&P 500 over the out-of-sample period. The LSTM networks outperform the GRU and linear ARIMA models on both out-of-sample return prediction as well as risk-adjusted returns. Predictive performance stays constant over the out-of-sample period, suggesting that the LSTM edge has not been arbitraged away.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter One

Introduction

Both academics and practitioners have made considerable efforts to forecast future stock returns and develop strategies to leverage these predictions into profits. French-Fama's 1993 endeavours to explain the cross-section of equity returns using their three-factor model has inspired other researchers to contribute to a continually expanding list of factors. Whilst these models may provide decent in-sample performance, their out-of-sample performance is usually lacking. Li, W. Zhang, and Kong (2018) show that the linear combination of factors is too restrictive and that a nonlinear combination outperforms. Since neural networks are able to discover nonlinear and chaotic patterns, they may be able to forecast market directions more accurately than linear multi-factor models. This may lead to improved out-of-sample predictive performance compared to traditional methods.

This thesis evaluates the effectiveness of Long-Short Term Memory (LSTM) neural networks models in equity return prediction using a large number of factors, and compares its performance to that of Gated Recurrent Unit neural networks and Autoregressive Integrated Moving Average (ARIMA) models. In conjunction with statistical measures, a trading-based evaluation procedure is used to evaluate the performance of the predictions, as the latter mirrors the economic relevance of LSTM neural networks to practitioners and thus may be more informative. This thesis both contributes to our understanding of the predictive ability of deep learning factor models, as well as their implications for practitioners.

The remaining chapters of this thesis are organized as follows: Chapter 2 provides a literature review of the related studies; Chapter 3 introduces the data and methodology, and Chapter 4 provides the results for the out-of-sample performance of the LSTM, GRU, and ARIMA models. Lastly, Chapter 5 concludes and discusses the results, and recommendations for future research are made.

Chapter Two

Literature Review

2.1 Related Works

2.1.1 Factor Portfolios

Over the last decades, a large body of research has emerged that suggests that financial stock returns may be predictable by the use of financial returns as well as macroeconomic data. Whilst earlier work focused on finding determinants of financial returns in the cross-section by implementing linear regressions, later research incorporates nonlinear methods and machine learning techniques to predict future stock returns. While this section will touch on both subjects, it will be primarily focused on the latter.

Much research has been devoted to uncovering the determinants of financial returns in the cross-section. Fama and French (1993) introduce in the concept of factor portfolios in their seminal and foundational paper on this topic. Fama and French construct two portfolios based on Book-to-Market ratio and firm size. Subsequently, both portfolios are sorted according to their Book-to-Market and firm size, and further partitioned into deciles. Then, the difference in returns between the tenth and first decile portfolio may be interpreted as the risk-premium to that specific factor. This paper essentially lays out the methodology upon which further research on factor portfolios is based. Fama and French's results show that the factor premia for Book-to-Market and firm size in combination with the market factor are able to explain stock returns in the cross-section rather well.

Since Fama and French's seminal paper on factor portfolios, other research has evaluated the abilities of other factors to explain stock returns in the cross-section to a possibly greater extent. Fama and French (2015) themselves introduce two more factors in order to better explain equity returns in the cross-section. Whilst some academics argue that many of these factors and their associated premia are merely compensation for risk, others argue that they can be regarded as anomalous. Over the last decades, a list of over 450 distinct factors or anomalies has emerged in the literature (Hou, Xue, and L. Zhang, 2017). A large replication study by Hou, Xue, and L. Zhang examines the reliability of these factors. The authors find that 82 per cent of the factors under consideration do not pass the hurdle of reaching a significance level of 5 per cent.

2.1.2 Neural Networks and Return Prediction

Neural networks may have several advantages as compared to traditional statistical methods. First, the universal approximation theorem shows that neural networks are able to approximate any nonlinear function (Cybenko, 1989). Second, unlike other nonlinear modelling techniques, neural network models are not specified in advance by the researcher(s), but rather generated by the network itself through model-training (Livingstone, Manallack, and Tetko, 1997). However, it should be noted that while NNs provide important advantages, they also suffer from major drawbacks as compared to other techniques. Neural networks are often regarded as "black boxes", as the weight matrices are difficult to interpret and causal inferences are not easily established (Tu, 1996). Moreover, neural networks are prone to overfitting, meaning that the model may learn to fit the training data too well, thereby losing much of its predictive power over unseen out-of-sample datasets (Tu, 1996).

The above research on the influence on factors on equity returns is mostly performed using linear regression techniques, which impose a linear relationship on the variables. However, Li, W. Zhang, and Kong (2018) show that the linear combination of factors is too restrictive and that a nonlinear combination outperforms. Moreover, Abhyankar, Copeland, and Wong (1997) find evidence for nonlinear dependencies in the returns on four of the world's largest indices: the S&P 500, the DAX, the Nikkei 225, and the FTSE-100. One way in which these nonlinearities may be accounted for is by implementing neural networks. Neural networks are universal approximators of functions and can, therefore, approximate whatever functional form best characterizes a time series (T. Hill, O'Connor, and Remus, 1996). As a result of this, T. Hill, O'Connor, and Remus (1996) find that neural networks significantly outperform traditional statistical methods in a major forecasting competition.

Motiwalla and Wahab (2000) use Artificial Neural Networks to predict equity returns one-step-ahead as well as construct trading strategies based on said predictions. The authors use bond market yield and yield-curve characteristics as input variables. Whilst their neural network only achieves a directional accuracy of 48 per cent, it is able to produce significantly higher cumulative and risk-adjusted returns than buy-and-hold returns on market indices.

Machine learning may also be used to implement more complex trading strategies. Huck (2009) uses Artificial Neural Networks in conjunction with other econometric techniques to forecast the stock direction and magnitude of stock price changes. Subsequently, the authors apply these predictions to form pairs of stocks, in an attempt to perform statistical arbitrage. Such a strategy involves finding co-integrated equities that may whose prices may drift apart in the short run, but which eventually will mean-revert. Hence, shorting the relatively overvalued stock and going long in the undervalued equity may generate positive returns (Huck, 2009). The trading strategies based on a combination of deep neural networks, gradient-boosted trees, and random forests show annualized returns of 73 per cent after transaction costs, with a substantially higher Sharpe ratio of 1.81 as compared to 0.35 for the S&P 500 index. These results clearly challenge the efficient market hypothesis.

More recently, Fischer and Krauss (2018) used Long Short-Term Memory (LSTM) neural networks to forecast the predicting out-of-sample daily directional movements for the constituent stocks of the S&P 500, and implement trading strategies based on these predictions. The authors find that LSTM neural networks outperform other forms of neural networks as well as regression, with annualized returns of 82 per cent after transaction costs and a Sharpe ratio of 2.34 over the out-of-sample period of 1992-2015. Interestingly, Fischer and Krauss find that LSTM returns fluctuate around 0 per cent from 2009 onward, suggesting that the LSTM 'edge', as it is applied in this paper, has been arbitraged away. Nakagawa et al. (2019) use a variation of the LSTM architecture to build a time-varying multi-factor model using 16 factors on the Japanese stock market. The authors find that their LSTM model is able to capture nonlinear and time-varying relationship between the factors and equity returns, resulting in higher annualized returns and a higher Sharpe ratio as compared to a linear model, a Support-Vector Machine, and a Random Forrest.

Besides forecasting returns, neural networks are also frequently used to predict volatility in the context of financial economics. The ability to forecast volatility of returns is crucial to investors given its impact on portfolio optimization and asset valuation, as well as the computation of optimal hedge ratios (Kristjanpoller and Minutolo, 2016). Monfared and Enke (2014) use several hybrid models in which they combine GJR-GARCH models and several types of neural networks to forecast volatility of the NASDAQ over a 40-day period. The authors find that the hybrid model exhibits superior predictive performance in times of extreme turmoil, such as the 2008 financial crash. However, the hybrid model is unnecessarily complex in low-volatility periods, and other less complex econometric techniques show higher performance. Roh (2007) performs a similar volatility forecasting exercise on the Korea Composite Stock Price Index. Roh also finds that neural networks combined with traditional statistical methods outperform either of the individual models in predictive accuracy. Compared to the single neural network model, the hybrid NN-EGARCH model outperforms the former by as much as 36.94 per cent on the basis of Mean Absolute Error. Moreover, the directional accuracy, the percentage of directional changes in volatility correctly predicted, of the NN-EGARCH model is 60.63 per cent while that of the NN-only model is only 43.75 per cent.

2.2 Long Short-Term Memory Neural Networks

Long Short-Term Memory (LSTM) neural networks have been applied in different areas of research, with a particular focus in the field of language processing. LSTM models have shown excellent performance in tasks such as speech synthesis and recognition, machine translation, as well as image-to-text conversion (Zen and Haşim Sak, 2015). Although the LSTM architecture has existed since 1997 (Hochreiter and Schmidhuber, 1997), its application to financial markets has only been recent.

Long Short-Term Memory neural networks is a unique form of a Recurrent Neural Network (RNN). An important distinction of Recurrent Neural Networks as opposed to a Feed-Forward NN is the fact that an RNN is able to retain some of the logic it used to generate previous outputs to generate the current output (Kawakami, 2008). Hence, an RNN is not only influenced by what it learned during supervised training, but also by what it learned in generating previous outputs. In other words, the RNN is able to use contextual information in generating its output. Figure 2.1 shows a schematic representation of a one-input, one-neuron Recurrent Neural Network (Kawakami, 2008). Each individual node represents a different time step. Each input (x_t) is passed onto the neuron, which then passes its output onto the output layer (r_t). The next time step, some information from the previous time step's neuron is passed onto the current neuron, affecting its output. This way, in theory, a RNN should be able to access many of the previous inputs in order to generate its current output.

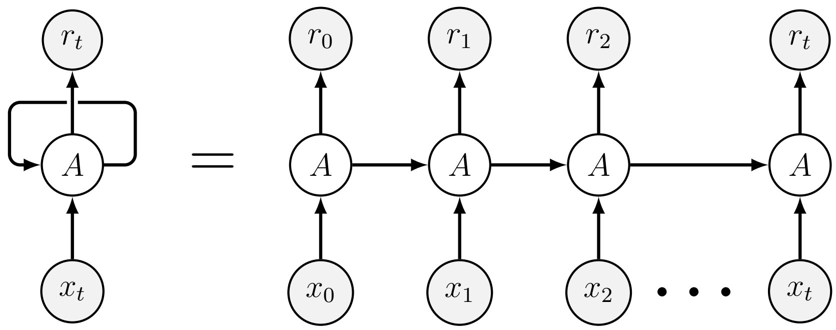

Figure 2.1 Schematic representation of a one-layer Recurrent Neural Network (Kawakami, 2008)

One major drawback of standard Recurrent Neural Networks is that they are somewhat limited in the range of contextual information they are able to access in practice (Kawakami, 2008); RNNs tend to 'forget' contextual information rather quickly. This is referred to as the *vanishing gradient problem* (Hochreiter, 1998). Kawakami (2008) note that the sensitivity of an input decreases over time as new inputs overwrite their activation of the hidden layer; as such, the RNN 'forgets' the inputs many time steps away.

Long Short-Term Memory neural networks are a particular variation within the class of Recurrent Neural Networks that are able to overcome the *vanishing gradient problem*, and hence they are able to learn long-term dependencies. Figure 2.2 shows a visual representation of an LSTM memory block with a single cell and input. The foremost reason why LSTM networks are able to learn long-term dependencies and overcome the vanishing gradient problem is by how the cell-state (C_t in figure 2.2) is manipulated by the memory block and passed onto the memory block of the next time step; this way, it functions as a form of long-term memory. Information is added or removed from the cell-state by the use of three

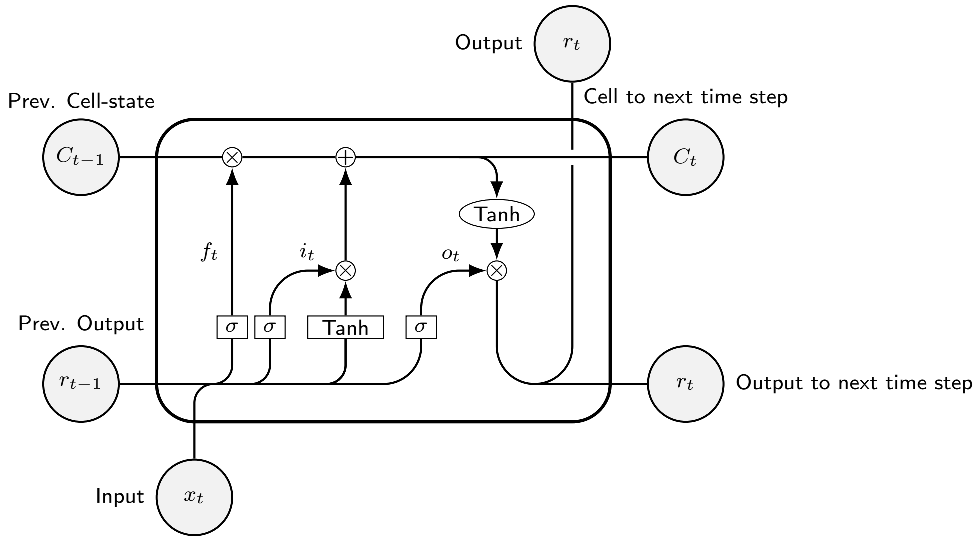

Figure 2.2 Visual representation of an LSTM block with one cell (Olah, 2015)

gates: the input-gate, the forget-gate, and the output-gate (Olah, 2015).

For the input- and forget-gate, the amount of information each of the layers allows through is determined by sigmoid (σ) functions, which take values of between 0 and 1 as to regulate the amount of information they let through (Olah, 2015). A value of 0 means that no information is let through, while a value of 1 implies that all information is passed on. The forget-gate, as the name suggests, regulates what information from the previous cell-state is kept in the current cell-state. Then, a *tanh*-layer determines which of the features (inputs) are to be considered to be added to the cell-state. The input-gate supervises which of these inputs will be added to the cell-state through a sigmoid layer, and incorporated into the cell-state or updates these inputs to their more current value. Lastly, the output-gate determines which what elements of the cell-state will be converted to output; this is subsequently multiplied by a *tanh*-layer of the cell-state, which scales the cell-state contents to values between [-1,1] (Olah, 2015).

2.2.1 LSTM in Equations

The formulae used by LSTM neural networks for the forget-gate and the input-gate are shown in equation 2.1 and 2.2, respectively (Kawakami, 2008; Olah, 2015).

$$f_t = \sigma(w_f \cdot [r_{t-1}, x_t] + b_f) \tag{2.1}$$

$$i_t = \sigma(w_i \cdot [r_{t-1}, x_t] + b_i) \tag{2.2}$$

where:

f_t represents the forget-gate

i_t represents to the input-gate

σ refers to the sigmoid activation function, applied element-wise

w_j = weight for the neurons of the respective gate

r_{t-1} = output of the LSTM memory block of the previous time step

x_t = input at the previous time step

b_j = the bias for the respective gate

As stated above (Section 2.2), a *tanh*-layer determines which inputs to consider for the cell-state by creating a vector of new candidate values as shown in equation 2.3.

$$\tilde{C}_t = \tanh(w_c \cdot [r_{t-1}, x_t] + b_c) \tag{2.3}$$

where:

$\tilde{C}_t$ represents the vector of candidate values

tanh refers to the tanh activation function, applied element-wise

Subsequently, the cell-state is updated by removing some, if any, of the constituents of the previous cell-state through the forget-gate, as well as introducing new and updating values through the input gate. This is shown in equation 2.4.

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t \tag{2.4}$$

where:

C_t = the cell-state memory vector of the LSTM network at time t

$\odot$ refers to the element-wise product

Through a sigmoid layer, the output-layer determines which what elements of the cell-state will be converted to output. Subsequently, the output-layer is multiplied by the vector of the cell-state memory to produce the final output of the LSTM block at time t. These calculations are shown in equations 2.5 and 2.6, respectively. Then, both the cell-state as well as the current output r_t will be passed onto the next LSTM block, to calculate the output for the next time step.

$$o_t = \sigma(w_o \cdot [r_{t-1}, x_t] + b_o) \tag{2.5}$$

$$r_t = o_t \odot \tanh(C_t) \tag{2.6}$$

where:

o_t represents the output-gate

r_t = the output of the LSTM NN at time step t

2.2.2 Gated Recurrent Units

Introduced in 2014 by Cho et al., the Gated Recurrent Unit (GRU) is a closely related variant to Long Short-Term Memory networks that is also able to overcome the vanishing gradient problem. GRU and LSTM are rather similar in architecture in that both use so-called 'gates' to control the information flowing through the neural network. However, GRU networks are less complex than LSTM networks as the former utilizes two gates rather than three. Specifically, the Gated Recurrent Unit does not have an output gate that controls which elements in the cell-state are exposed to the units in the network (see section 2.2.1). As such, the GRU exposes its full cell-state contents without any control (Chung et al., 2014).

In many machine learning applications, GRU and LSTM models show similar performance, despite the GRU being inherently less complex. There is no theoretical guidance as to which model should be chosen for which task; the final decision depends heavily on the dataset and corresponding task (Chung et al., 2014). In comparison to LSTM models, GRU networks have fewer parameters and may therefore generalize better based on smaller datasets (Zhao et al., 2019). However, this comes at a cost: compared to LSTMs, GRUs are able to model fewer temporal relations and thus have lower expressiveness.

Rather than using a cell-state like LSTMs, the GRU operates directly on the output (or hidden state) of the previous time step. Gated Recurrent Units have two different gates to control the flow of information of the hidden state: the update and reset gate. The update gate regulates which new information from the input features and the previous hidden state to incorporate into the current hidden state, and the reset gate decides which information to remove (forget) from the hidden state (Olah, 2015).

The equations used by Gated Recurrent Units for the update and the reset gates are shown in equations 2.7 and 2.8, respectively (Chung et al., 2014).

$$z_t = \sigma(w_z \cdot x_t + U_z \cdot r_{t-1}) \tag{2.7}$$

$$q_t = \sigma(w_q \cdot x_t + U_q \cdot r_{t-1}) \tag{2.8}$$

where:

z_t represents the update gate

q_t represents the reset gate

U_j = weight for the previous hidden state of the respective gate

A *tanh*-layer determines which of the features (inputs) and elements of the previous hidden state are to be considered to be added to the hidden state. This is done by creating a vector of candidate values, as shown in equation 2.9.

$$\tilde{r}_t = \tanh(w_r \cdot x_t + q_t \odot U_q r_{t-1}]) \tag{2.9}$$

where:

$\tilde{r}_t$ represents the vector of candidate values for the hidden state

Subsequently, the hidden state (output) is calculated is by means of the update gate, which allows through elements of the previous hidden state, as well as elements of the vector of candidate values. This is shown in equation 2.10.

$$r_t = (1 - z_t) \odot r_{t-1} + z_t \odot \tilde{r}_t \tag{2.10}$$

Figure 2.3 shows a visual representation of a GRU block with a single cell and input, showing the interaction between the above equations.

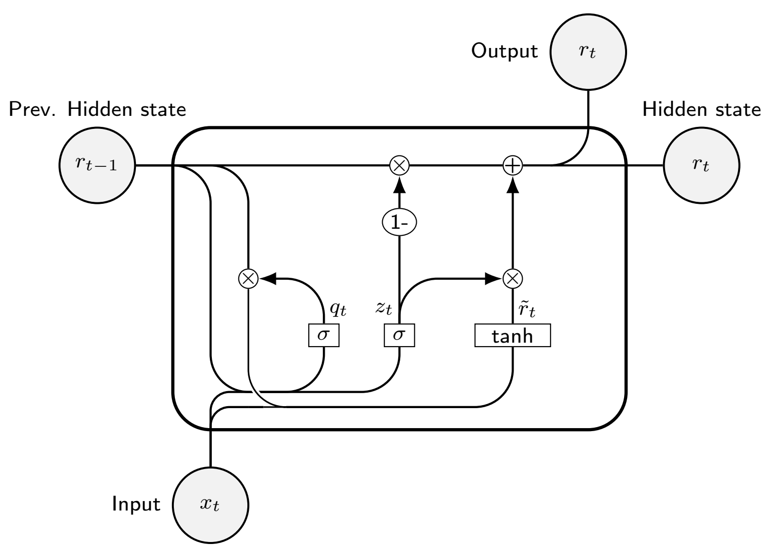

Figure 2.3 Visual representation of a GRU cell (Olah, 2015)

Chapter Three

Data and Methodology

3.1 Data

The datasets on the total monthly returns of 30 of the Standard & Poor's 500 index constituents are obtained from the Center for Research in Security Prices (CRSP). The choice to focus on S&P500 constituent equities is motivated by their relatively high degree of market efficiency and liquidity. Moreover, the sample of public equities has been constructed to reflect a variety of industries and for historic return-data availability. The list of stocks used in this thesis can be found in Table A.2. The size of each of the datasets depends on the stock in question. These datasets are subsequently combined with a dataset of 51 monthly factors returns produced and maintained by Hou, Xue, and L. Zhang (2017). The factor dataset covers the horizon from January 1967 to December 2019. The factors used and their respective descriptions can be found in Table A.1.

The datasets and methodology are constructed to ensure that the input variables used to forecast next month's return were not unobservable at the time the prediction is made. This is done to ensure that the conditions in which the predictions are made are similar to those practitioners experience in the real world.

3.2 Methodology

3.2.1 Recurrent Neural Networks

As stated in section 2.2, Recurrent neural networks (RNN) are neural networks that use contextual information from previous time steps as inputs to influence predictions at the current time step, which in theory would allow them to learn long-term dependencies (Hasim Sak, Senior, and Beaufays, 2014). However, in practice, RNNs are only able to access contextual information between five to ten time steps in the past (Gers, Schmidhuber, and Cummins, 1999). Long short-term memory networks are able to overcome this apparent problem and learn contextual dependencies of over a thousand time steps away. Since Gated Recurrent Units use gates to manipulate the information in a similar way to LSTM networks, they are also able to overcome the *vanishing gradient* problem (Chung et al., 2014).

LSTM neural networks offer excellent performance in analysing sequential data for which an unknown hierarchical decomposition may exist (Gers, Schmidhuber, and Cummins, 1999). Since stock market behaviour may be characterised as such, an LSTM model may offer good performance in equity return prediction. Ryll and Seidens (2019) compiled a comprehensive survey of machine learning algorithms in financial market forecasting. The authors compile evidence of the predictive ability of different types of machine learning algorithms from over 150 published papers on these topics. Ryll and Seidens find that LSTM neural networks, on average, outperform other machine learning techniques such as Support-Vector Machines (SVM), Artificial Neural Networks (ANN), Recurrent Neural Networks (RNN), and Neuro-Fuzzy Networks by a large margin. The choice for Long Short-Term Neural Networks over architectures is specifically motivated by its superior performance over other machine learning algorithms, as well as its ability to overcome the vanishing gradient problem and thus learn long-term dependencies.

This thesis employs Autoregressive Integrated Moving Average (ARIMA) models in order to benchmark the predictive performance of the LSTM models against a traditional econometric statistical technique. ARIMA models are among the most widely used linear models in time series forecasting (Pai and Lin, 2005), and hence serve as a good benchmark. ARIMA models predict the future value of a variable on the basis of a linear combination of previous values and previous prediction errors. Following Ahoniemi (2008), an ARIMA(1,1,1) model is implemented, enhanced with first-lagged values of the factors (Table A.1). This particular ARIMA model specification shows highest directional accuracy on a validation sample. Moreover, the LSTM performance is also compared to that of Gated Recurrent Units. As stated in section 2.2.2, Gated Recurrent Units are a less complex variant of LSTM models that may show higher performance if the datasets are small. Since the datasets used in this thesis are relatively small (n=360 for in-sample training), the GRU architecture is relevant.

3.2.2 Software

This thesis partially follows the methodology as applied by Nakagawa et al. (2019), who use LSTM neural networks in conjunction with 16 accounting variables to forecast 1-month stock returns on the Tokyo Stock Exchange.

Data preparation, model training and model validation is conducted in Python 3.7. To do this, the Python software libraries NumPy (Walt, Colbert, and Gael Varoquaux, 2011), sci-kit learn (Pedregosa et al., 2011), pandas (McKinney et al., 2011), Keras (Chollet et al., 2015), as well as Google TensorFlow revision 2 (Abadi et al., 2016) are used.

First, each individual equity dataset is split into non-overlapping in-sample training and out-of-sample testing sets. The training sample consists of the first 80% of the observations; the test set is the following 20%. For most of the equities in the sample, the training data consists of 360 months, and the out-of-sample test set is 72 months long. Training sets are used to train the LSTM models, which are subsequently evaluated on their out-of-sample performance on the test sets.

3.2.3 Features

LSTM neural networks require sequential time series data of both the target measure as well as the 'features' for training purposes. The target measure, i.e. the measure that the LSTM network attempts to predict, is the monthly stock return, which is defined in equation 3.1. The return is the monthly percentage change in the total value of an investment in a common stock, taking into account stock splits and dividends received.

$$r_{i,t} = \frac{p_{i,t} * f_{i,t} + d_{i,t}}{p_{i,t-1}} \tag{3.1}$$

where:

$p_{i,t}$ = last sale price or closing bid/ask average for stock i at time t

$f_{i,t}$ = price adjustment factor for stock i at time t

$d_{i,t}$ = cash adjustment factor for stock i at time t

The inputs or features used for training are the lagged differences in returns between the 10th decile portfolio and the first decile portfolio for each of the factors, as shown in equation 3.2.

$$F_{i,t} = r_{10,i,t} - r_{1,i,t} \tag{3.2}$$

where:

$r_{10,i,t}$ = the return on the 10th decile portfolio for factor i at time t

$r_{1,i,t}$ = the return on the 1st decile portfolio for factor i at time t

3.2.4 Data Preparation

Before a neural network can be trained on the input data, the data need to be pre-processed and normalized as to improve the efficiency and accuracy of the model (Bishop et al., 1995). Each of the features, as well as the target variable, is normalized by the sklearn MinMaxScaler such that each value lies between zero and one. To avoid any bias, this is done separately for the in-sample training and out-of-sample test samples. After each prediction is made, the LSTM/GRU output is re-scaled back to its 'normal' state.

3.2.5 Model Training

For the training of the LSTM and GRU networks, the RMSprop optimizer is used (Tieleman and Hinton, 2012). Dropout regularization is an effective way to avoid overfitting and thus may improve out-of-sample performance (Srivastava et al., 2014); this means that some nodes are randomly dropped (temporarily removed) from the LSTM or GRU network during training. Following Fischer and Krauss (2018), a dropout rate of 10% is used. Hyperparameter optimization is performed and cross-validated on a sub-sample of the training sample dataset, again as to avoid any data snooping bias.

Hyperparameters are model parameters that have to be selected before the process of training the model, e.g. the number of layers or number of epochs (Lago, De Ridder, and De Schutter, 2018). There are no clear theoretical guidelines as to how one can determine the correct model specification (hyperparameters) of neural networks. In practice, they are often chosen based on rules-of-thumb, or through testing different specifications (Claesen and De Moor, 2015). As stated above, this thesis applies hyperparameter optimization and cross-validation on a validation sample to determine the hyperparameters. As such, the topology of the LSTM and GRU networks are as follows: the first layer is the input layer with 51 features and 48 time steps, the following layer is a hidden layer with 256 neurons, the last layer is a fully-connected dense output layer. These values lead to a high number of observations per parameter, which leads to reduced risk of overfitting and more robust estimates if the training data is noisy (Fischer and Krauss, 2018). Thus, both the LSTM and GRU neural networks take return and factor data of the previous 48 months into account in order to forecast the following month's return.

Each of the models is trained with 200 epochs, meaning that during training, the model passes 200 times over the training set; saving the best fitting model. While there is no technical specification for setting this number of epochs, this number is large enough to find a well-fitting model whilst moderate enough avoiding overfitting.

Since the predictions of the neural networks will be implemented into trading strategies, the accuracy of return direction prediction may be of higher importance than the absolute magnitude of the prediction error. Hence, it may be sensible to implement a custom loss function that penalizes the model when its prediction is not of the same sign as the true outcome. Thus, the loss function, as shown in equation 3.3, is implemented.

$$Loss_{i,t}(r_{i,t}^{true}, r_{i,t}^{pred}) = \begin{cases} \alpha * (r_{i,t}^{pred})^2 - sgn(r_{i,t}^{true}) * r_{i,t}^{pred} + |r_{i,t}^{true}|, & \text{if } r_{i,t}^{pred} * r_{i,t}^{true} < 0 \\ |r_{i,t}^{true} - r_{i,t}^{pred}|, & \text{otherwise} \end{cases} \tag{3.3}$$

where:

$r_{i,t}^{pred}$ = the predicted return for stock i at time t

$r_{i,t}^{true}$ = the predicted return for stock i at time t

α is a parameter that influences the sensitivity of the loss to predictions of the wrong sign

3.2.6 Trading Strategy

Rather than merely considering the statistical accuracy with which the LSTM and GRU models predict returns one month ahead, one should also consider their directional accuracy and the performance of trading strategies based on LSTM output. As such, two trading strategies are implemented: a Long-Only strategy, as well as a Long/Short strategy. The former strategy purchases the respective equity when the neural network model predicts a positive return for the next month and does not invest otherwise. The latter strategy purchases the stock if the forecasted return is positive and shorts the stock when the predicted return is negative. Both strategies will be evaluated against buy-and-hold strategies for each of the equities. Following Avellaneda and Lee (2010), a transaction/slippage cost of 5 basis points (0.05%) is applied to each trade.

In addition to the traditional risk measure of standard deviation of returns, the maximum drawdown is calculated for each of the out-of-sample trading strategies. This risk metric is routinely used by asset managers for fund allocation and redemption decisions (Van Hemert et al., 2020). The maximum drawdown measures downside risk by calculating the largest single percentage drop from the from the peak to the bottom of the value of a portfolio.

The methodology, as specified above, satisfies three important criteria to prevent data snooping bias, as outlined by Motiwalla and Wahab (2000). Firstly, the datasets are split into disjoint training and evaluation sets, as no investor could have obtained parameter estimates based on the entire sample period and trade upon that knowledge. Second, the possible issue of model uncertainty, meaning that the model specification could have been made with the benefit of hindsight, is addressed as LSTM and GRU neural networks form their own model specification after training on the training data. Third, there may be data snooping if the same dataset is used for model specification as well as validation of said specification; since the LSTM and GRU models are trained on a different dataset than they are evaluated upon, this form of data snooping is not present.

3.3 Hypotheses

Hypothesis 1 *The LSTM and GRU neural networks have directional accuracies significantly higher than 50%*

Directional accuracy, as referred to in this thesis, is the percentage of predictions that have the same sign as the realized return in that month. Neural networks may be able to detect nonlinear relationships in return and factor portfolio data, which may improve their ability to predict both the magnitude and sign of financial returns accurately.

Hypothesis 2 *The LSTM and GRU neural network trading strategies significantly outperform a buy-and-hold strategy based on the Sharpe Ratio*

To calculate whether the Neural Network trading strategies produce significantly higher Sharpe ratios than a buy-and-hold strategy of the same equity would, the test of Jobson, Korkie, and Memmel (JKM) is used (Memmel, 2003). Jorion (1992) shows that the power of this test is rather low in small sample sizes, which may be considered a drawback for its application to this thesis. Namely, the length of the out-of-sample test periods used in this thesis are relatively short (n=72 for each public equity). On the other hand, significantly higher Sharpe ratios for the neural network strategies would, therefore, constitute as strong evidence of higher risk-adjusted performance. The z-values for the JKM-test are calculated as shown in equation 3.4, with the Standard Error as calculated in equation 3.5.

$$z_i = \frac{Sh_i^{NN} - Sh_i^{BH}}{SE_i^{diff}} \tag{3.4}$$

where:

Sh_i^{NN} = the Sharpe ratio of the Long-Only or Long-Short NN strategy for stock i

Sh_i^{BH} = the Sharpe ratio of the buy-and-hold strategy for stock i

SE_i^{diff} = the Standard Error of the difference between the Sharpe ratios

$$SE_i^{diff} = (\frac{1}{T}(2 - 2p_{BH,NN} + \frac{1}{2}(Sh_{BH}^2 Sh_{NN}^2 - 2Sh_{BH}Sh_{NN})))^{\frac{1}{2}} \tag{3.5}$$

where:

$p_{BH,NN}$ = the correlation between the the buy-and-hold returns and the NN returns

T = the number of time steps in the out-of-sample test period

Chapter Four

Results

The results are presented in three stages. First, the out-of-sample statistical accuracy of the LSTM, GRU, and ARIMA models is presented and analysed. Following this, the out-of-sample performance of the trading strategies is presented and thoroughly evaluated for each of the three models. Third, it is analysed whether the predictability of returns diminishes over time, which may indicate increasing market efficiency. As a benchmark, the results of the LSTM neural networks are compared to those of Auto Regressive Integrated Moving Average (ARIMA) and Gated Recurrent Unit (GRU) models.

4.1 Statistical Forecast Evaluation

Table 4.1 lists the R^2, Mean Absolute Error (MAE), Mean Square Error (MSE), Root Mean Square Error (RMSE), and the Mean Absolute Percentage Error (MAPE) for each stock over the out-of-sample test period for the LSTM models. The summary statistics for the benchmark GRU and ARIMA models are shown in Table B.1. The statistics in Table 4.1 show how good, on average, the LSTM Neural Networks are at forecasting the magnitude of the monthly returns one step ahead. In addition, figure 4.1 shows the forecasted returns contrasted with the actual returns over the full out-of-sample period for ticker AXP. This figure serves as an example and is similar to those of the other equities.

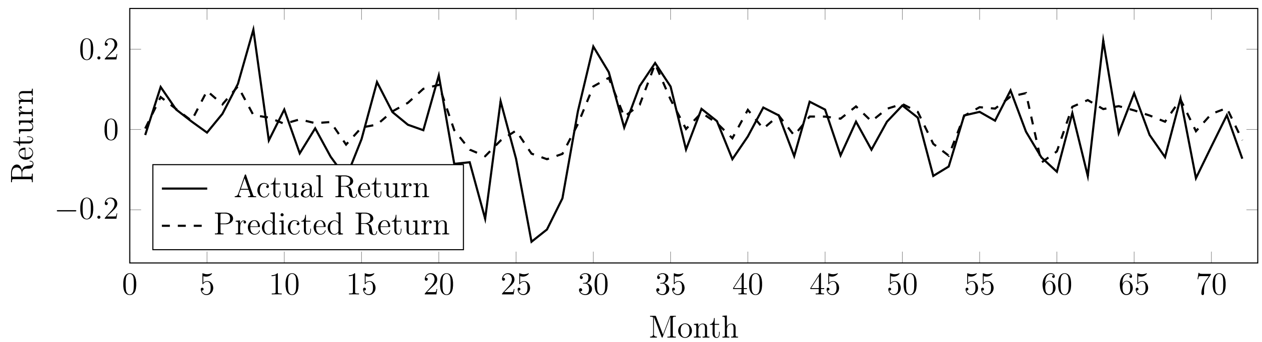

Figure 4.1 LSTM out-of-sample predicted and actual monthly returns for AXP

As shown in 4.1, the LSTM out-of-sample R^2 are rather high in general, with values ranging between 12 to 61 per cent for the stocks under consideration. On average, the LSTM models explain over 14 times as much variation in the cross-section of returns than the benchmark ARIMA models; the GRU models outperform, on average, the LSTM models by 2.6 percentage points as based on R^2. Overall, the LSTM models, in conjunction with the 51 factors, are able to explain relatively high amounts of the variance in out-of-sample equity returns. On the other hand, the summary statistics that measure average prediction error (MAE, RSME, MSE, & MAPE) are rather high for most stocks. This implies that the LSTM models tend to be less precise in forecasting the magnitude of the return. Indeed, from figure 4.1 it is evident that the return predictions are less volatile than the actual returns, resulting in large error statistics. The LSTM models show near identical performance to the GRU models as based on MAE and RMSE, but perform worse as based on MAPE. Overall, the LSTM models outperform the benchmark ARIMA models on the basis of Mean Absolute Error by 69.6 per cent on average.

The error statistics, as reported in Table 4.1, should be interpreted with caution, as they may be higher than they would have been given a different loss function. Namely, the loss function that the LSTM Neural Network utilises for in-sample training not only penalises absolute error but also whether the direction of the return was correctly predicted. This may have adversely affected the statistics reflected in Table 4.1.

Table 4.1 LSTM Out-of-sample R-squared, MAE, RMSE, MSE, and MAPE for each of the equities

Ticker	R-squared	MAE	RMSE	MSE	MAPE
ADM	0.1159	0.0514	0.0727	0.0053	298%
ADP	0.3689	0.0329	0.0414	0.0017	463%
AXP	0.2954	0.0378	0.0520	0.0027	177%
BAX	0.3392	0.0575	0.0654	0.0043	650%
BDX	0.6062	0.0244	0.0324	0.0011	154%
CLX	0.3716	0.0320	0.0414	0.0017	345%
DIS	0.2997	0.0541	0.0627	0.0039	474%
DUK	0.1722	0.0362	0.0458	0.0021	198%
FDX	0.3344	0.0544	0.0713	0.0051	1296%
GIS	0.5181	0.0307	0.0381	0.0014	295%
HON	0.2657	0.0353	0.0470	0.0022	997%
HPQ	0.2929	0.0586	0.0792	0.0063	254%
IFF	0.4146	0.0464	0.0587	0.0034	164%
INTC	0.3035	0.0412	0.0556	0.0031	239%
JNJ	0.5390	0.0209	0.0272	0.0007	115%
JPM	0.3051	0.0408	0.0547	0.0030	178%
LNC	0.4376	0.0540	0.0719	0.0052	130%
MCD	0.3888	0.0318	0.0404	0.0016	287%
MMM	0.3318	0.0450	0.0555	0.0031	515%
NEM	0.3151	0.0604	0.0915	0.0084	334%
PCAR	0.4138	0.0495	0.0647	0.0042	208%
SLB	0.2783	0.0480	0.0667	0.0045	175%
T	0.3534	0.0392	0.0508	0.0026	306%
TAP	0.5474	0.0390	0.0513	0.0026	143%
TGT	0.2762	0.0449	0.0630	0.0040	580%
TXT	0.2141	0.0552	0.0720	0.0052	319%
VFC	0.4530	0.0356	0.0446	0.0020	950%
WBA	0.4831	0.0409	0.0533	0.0028	160%
WMB	0.5189	0.0565	0.0764	0.0058	227%
WMT	0.2970	0.0307	0.0432	0.0019	494%
Mean	0.3617	0.0428	0.0564	0.0034	371%

4.2 Trading Strategy Evaluation

This section will cover the evaluation of the buy-and-hold, LSTM, GRU, and ARIMA trading strategies as specified in section 3.2.6. Each of the respective Long-Only and Long/Short strategies will be evaluated against a buy-and-hold strategy on several aspects. First, the directional accuracy of the different models will be assessed. Second, the annualized returns, standard deviation of the returns, and maximum drawdown of each strategy will be presented and evaluated. Third, the Sharpe ratios of the trading strategies will be evaluated for each of the equities. Lastly, an equally-weighted portfolio of the equities will be created for each of the strategies and subsequently evaluated.

4.2.1 Directional Accuracy

Kendall and A. B. Hill (1953), among others, proposed that stock prices indeed evolve according to a random walk, meaning that it should not be possible to predict the direction of stock price changes consistently. Hence, on average, one should expect an accuracy of 50% in predicting the direction of equity returns.

Table 4.2 shows the out-of-sample directional accuracy of the LSTM models in regards to forecasting next month's return, as well as the fraction of the out-of-sample period in which the LSTM trading strategies take long positions. The reported p-values refer to the binomial probability of achieving at least the reported accuracy under the null hypothesis of 50 per cent. The results show that the overall directional accuracy for the LSTM models is significantly higher than 50 per cent at the 1 per cent level for 24 out of 30 public equities in the sample, with accuracies ranging between 55 and 82 per cent. Table B.3 shows the directional accuracies for the GRU models, of which 21 out of 30 stocks achieve an directional accuracy higher than 50 per cent at the 1 per cent significance level. Overall, the results show that the direction of many of the equities in the sample seem to be predictable.

Table 4.2 Fraction of periods invested as well as out-of-sample directional accuracy of the LSTM Neural Networks

Ticker	Fraction of Periods Long	Directional Accuracy	Upwards Acc.	Downwards Acc.
ADM	0.9028	59.72%**	56.92%	85.71%***
ADP	0.8194	70.83%***	69.49%***	76.92%**
AXP	0.8194	63.89%***	64.41%***	61.54%
BAX	0.1558	54.55%	100.0%***	46.15%
BDX	0.6667	76.39%***	79.17%***	70.83%**
CLX	0.4306	66.67%***	77.42%***	58.54%
DIS	0.9444	58.33%*	55.88%	100.0%***
DUK	0.2778	58.33%*	80.00%***	50.00%
FDX	0.9167	70.83%***	68.18%***	100.0%***
GIS	0.6944	63.89%***	60.00%*	72.73%***
HON	0.8472	70.83%***	72.13%***	63.64%
HPQ	0.7917	76.39%***	73.68%***	86.67%***
IFF	0.1528	58.33%*	100.0%***	50.82%
INTC	0.5694	69.44%***	75.61%***	61.29%*
JNJ	0.7639	81.94%***	80.00%***	88.24%***
JPM	0.7778	69.44%***	67.86%***	75.00%**
LNC	0.3889	75.00%***	92.86%***	63.64%**
MCD	0.4167	66.67%***	83.33%***	54.76%
MMM	0.9167	68.06%***	66.67%***	83.33%**
NEM	0.5972	68.06%***	67.44%***	68.97%**
PCAR	0.2778	69.44%***	90.00%***	61.54%**
SLB	0.4306	79.17%***	74.19%***	82.93%***
T	0.8333	66.67%***	61.67%**	91.67%***
TAP	0.2927	68.29%***	100.0%***	55.17%
TGT	0.5333	76.67%***	84.38%***	67.86%**
TXT	0.8056	62.50%**	60.34%**	71.43%**
VFC	0.4722	72.22%***	82.35%***	63.16%**
WBA	0.6528	73.61%***	65.96%***	88.00%***
WMB	0.7639	75.00%***	69.09%***	94.12%***
WMT	0.5000	68.06%***	69.44%***	66.67%**

Note: * = p < .1, ** = p < .05, *** = p < 0.01, $H_0 = 50\%$

Table B.2 shows the out-of-sample directional accuracy of the benchmark ARIMA models. The results show that the directional accuracy of the ARIMA models is significantly higher than 50 per cent for only 5 out of 30 stocks. On average, the LSTM models outperform the ARIMA models based on directional accuracy by 34.3 per cent (17.7 percentage points); the average LSTM directional accuracy is nearly identical to that of the GRU models.

To evaluate whether the LSTM and GRU models are more accurate in predicting either the upwards or downward movements, the overall directional accuracy is separated into these categories and shown in Table 4.2 and B.3, respectively. Regarding the upwards accuracy of LSTM and GRU models, for 27 and 28 out of 30 equities in the sample, respectively, the upward accuracy is significantly higher than 50 per cent at the 5 per cent level. As for the downwards accuracy, for both the LSTM and GRU models 21 of the 30 equities reach an accuracy significantly higher than 50 per cent at the 5 per cent level. It must be noted that the samples of downward predictions tend to be smaller than those for upward predictions. Overall, there is no evidence to suggest that the LSTM or GRU models are more accurate at forecasting one direction over the other. For the ARIMA models (Table B.2), there does seem a substantial difference in the upward and downward accuracies, with upward accuracies being generally substantially higher than downward accuracies.

4.2.2 Annualized Returns and Risk

Table 4.3 shows the geometric average annualized returns for each of the LSTM trading strategies: the Buy-and-Hold, Long-Only, and Long/Short strategies. As noted in section 3.2.6, these results take trading fees into account. As for annualized returns, the results seem mixed. Whilst some of the LSTM trading strategies clearly outperform the Buy-and-Hold strategy on a return basis, others receive similar or even lower returns. For 17 out of 30 equities in the sample, the LSTM networks outperform the buy-and-hold strategy on a return basis at least at the 5 per cent significance level. This number increases to 22 out of 30 for the GRU models, as shown in Table B.5. Table B.4 shows the annualized returns for the ARIMA models. These results show that for only one equity the benchmark ARIMA models are able to achieve significantly higher annualized returns than the Buy-and-Hold strategy. Moreover, for 24 out of the 30 stocks both the ARIMA Long-Only and Long/Short strategies achieved lower returns than the Buy-and-Hold strategy. As such, the LSTM and GRU models clearly outperform both the Buy-and-Hold strategy as well as the ARIMA models in generating returns over the out-of-sample period.

Table 4.3 LSTM annualized geometric mean returns and standard deviations of monthly returns over the out-of-sample period for each stock

	Annualized Returns			Standard Deviation		
Ticker	Buy&Hold	Long-Only	Long/Short	Buy&Hold	Long-Only	Long/Short
ADM	0.0327	0.1123**	0.1899**	0.0603	0.0537	0.0582
ADP	0.1744	0.2602***	0.347**	0.0476	0.0397	0.0424
AXP	0.0777	0.1947**	0.3019**	0.0553	0.0367	0.0506
BAX	0.1490	0.1647	0.1505	0.0583	0.0322	0.0569
BDX	0.1781	0.2905***	0.4045***	0.0512	0.0400	0.0443
CLX	0.1155	0.1447	0.1608	0.0478	0.0321	0.0469
DIS	0.1285	0.1708**	0.2122**	0.0555	0.0529	0.0539
DUK	0.0925	0.0826	0.0611	0.0404	0.0218	0.0408
FDX	0.0349	0.1529**	0.2618**	0.0701	0.0571	0.0669
GIS	0.0365	0.172***	0.3152***	0.0499	0.0383	0.0438
HON	0.1538	0.1995*	0.2439*	0.0415	0.0364	0.0390
HPQ	0.0834	0.2521***	0.429***	0.0715	0.0593	0.0643
IFF	0.0880	0.1434	0.1598	0.0628	0.0289	0.0618
INTC	0.1832	0.2755*	0.3579*	0.0628	0.0497	0.0586
JNJ	0.0905	0.1911***	0.2958***	0.0388	0.0286	0.0326
JPM	0.1895	0.2499	0.2975	0.0582	0.0457	0.0556
LNC	0.0562	0.4007***	0.7749***	0.0895	0.0562	0.0735
MCD	0.1592	0.1757	0.1832	0.0398	0.0265	0.0391
MMM	0.0733	0.1315*	0.1849*	0.0523	0.0452	0.0505
NEM	0.0808	0.2653*	0.394*	0.1104	0.0814	0.1058
PCAR	0.0919	0.2151*	0.3089	0.0654	0.0345	0.0612
SLB	-0.1365	0.1931***	0.5827***	0.0785	0.0487	0.0673
T	0.0672	0.1254***	0.1859***	0.0466	0.0435	0.0445
TAP	0.1347	0.2717*	0.3979*	0.0644	0.0454	0.0583
TGT	0.1532	0.3178**	0.4788**	0.0731	0.0561	0.0655
TXT	0.0835	0.2571**	0.4191**	0.0766	0.0560	0.0703
VFC	0.1096	0.2401***	0.3722***	0.0590	0.0461	0.0528
WBA	0.0081	0.2341***	0.479***	0.0708	0.0515	0.0615
WMB	-0.0231	0.3886***	0.8697***	0.0990	0.0663	0.0812
WMT	0.1004	0.1941*	0.2777*	0.0509	0.0322	0.0468
Mean	0.0922	0.2156	0.3366	0.0616	0.0448	0.0565

Note: * = p < .1, ** = p < .05, *** = p < 0.01, $H_0 : \overline{R}_{BH} = \overline{R}_{LSTM}$

The risk each of the three strategies entails may be quantified by the standard deviation of monthly returns. The standard deviation of returns for the LSTM, GRU, and ARIMA models are shown in Table 4.3, B.5, and B.4, respectively. Overall, the LSTM, GRU, and ARIMA Long-Only strategies produce lower standard deviations than the Buy-and-Hold strategies, meaning that the former may be considered less risky. It is interesting to note that while one may have expected the Long/Short strategy to be riskier than the Buy-and-Hold strategy, for many of the equities under consideration the former produce lower standard deviations than the latter.

Table 4.4 shows the maximum drawdown (MDD) for the Buy-and-Hold strategy as well as the LSTM, GRU, and ARIMA strategies. Maximum drawdown can be interpreted as a measure of downside risk. It represents the worst possible loss suffered by an investor when investing in a portfolio or stock: buying at the highest point, and selling at the bottom (Van Hemert et al., 2020). The results in Table 4.4 show that the LSTM Long-Only strategy shows substantially lower max drawdowns than the Buy-and-Hold strategy for most of the stocks under consideration. The GRU Long-Only strategies shows similar results, albeit slightly higher than those of the LSTM Long-Only. The LSTM and GRU Long/Short strategies more often show similar MDDs to the Buy-and-Hold strategy. The ARIMA Long-Only and Long/Short strategies both only rarely show substantially lower MDDs than the Buy-and-Hold strategy. Overall, these results suggest that the LSTM and GRU Long-Only strategies seem to be exposed to lower drawdown risk, while the LSTM, GRU Long/Short, and benchmark ARIMA strategies do not.

Table 4.4 Maximum drawdown (MDD) over the out-of-sample period for each stock for the LSTM, GRU, and ARIMA trading strategies

		LSTM MDD		GRU MDD		ARIMA MDD	
Ticker	Buy&Hold MDD	Long-Only	Long/Short	Long-Only	Long/Short	Long-Only	Long/Short
ADM	0.2848	0.2848	0.2848	0.2037	0.2133	0.2093	0.2545
ADP	0.2290	0.1607	0.1607	0.2290	0.2290	0.1627	0.2550
AXP	0.2796	0.1341	0.2360	0.2796	0.3152	0.2801	0.2801
BAX	0.2876	0.1215	0.2443	0.1544	0.2443	0.2300	0.2630
BDX	0.2218	0.1673	0.1673	0.2215	0.2215	0.2219	0.2219
CLX	0.2214	0.1682	0.2060	0.1825	0.1825	0.2010	0.2276
DIS	0.2580	0.2580	0.2580	0.2304	0.2304	0.2027	0.2297
DUK	0.1713	0.1490	0.1635	0.1513	0.1626	0.1415	0.1713
FDX	0.4059	0.3161	0.3698	0.1960	0.2216	0.3564	0.3564
GIS	0.2392	0.1922	0.1922	0.1334	0.1605	0.1885	0.2183
HON	0.1877	0.1815	0.1815	0.1815	0.1815	0.0978	0.1809
HPQ	0.2468	0.2471	0.2584	0.2046	0.2457	0.2432	0.2584
IFF	0.3367	0.1110	0.2132	0.2139	0.2139	0.2147	0.2147
INTC	0.2725	0.2196	0.2196	0.2197	0.2445	0.2147	0.2147
JNJ	0.1956	0.1229	0.1229	0.1070	0.1217	0.1383	0.1658
JPM	0.2415	0.2090	0.2525	0.2112	0.2112	0.2415	0.2415
LNC	0.3802	0.2625	0.2976	0.3760	0.3760	0.3153	0.3991
MCD	0.1959	0.1283	0.1855	0.1354	0.1858	0.1458	0.1704
MMM	0.2330	0.2330	0.2368	0.1780	0.1821	0.2267	0.2267
NEM	0.4476	0.3800	0.4861	0.3800	0.4143	0.3978	0.4389
PCAR	0.2886	0.1363	0.2650	0.1701	0.2269	0.2505	0.2505
SLB	0.3378	0.1935	0.2380	0.1840	0.1999	0.3367	0.3367
T	0.2103	0.2103	0.2103	0.1562	0.1853	0.2100	0.2100
TAP	0.2447	0.1569	0.1470	0.2115	0.2115	0.2115	0.2413
TGT	0.2898	0.2366	0.2808	0.2898	0.2898	0.2445	0.3347
TXT	0.3499	0.2308	0.2559	0.3503	0.3503	0.2942	0.2942
VFC	0.2174	0.2144	0.2144	0.1661	0.1661	0.1810	0.2158
WBA	0.3220	0.2444	0.2444	0.3320	0.3320	0.2887	0.2903
WMB	0.4230	0.2918	0.3089	0.4234	0.4234	0.3535	0.3535
WMT	0.2643	0.1281	0.1833	0.2309	0.2309	0.1221	0.2119
Mean	0.2761	0.2030	0.2362	0.2234	0.2391	0.2308	0.2576

4.2.3 Risk-Return Performance

Table 4.5 shows the Sharpe ratios for both the LSTM and GRU trading strategies, as well as the Buy-and-Hold strategy and benchmark ARIMA models. It should be noted that these Sharpe ratios are calculated with the monthly returns, rather than annual returns. The Sharpe ratios of both the LSTM and GRU Long-Only strategies are significantly higher than the Buy-and-Hold strategy at the 5 per cent level for 28 out of the 30 equities under consideration. On average, the LSTM Long-Only Sharpe ratios are 2.4 times as high as those of the Buy-and-Hold strategy. Hence, it seems that the Long-Only LSTM trading strategy is able to outperform a Buy-and-Hold strategy on a risk-adjusted basis. It is noteworthy that this result is not achieved through solely higher returns, but also by reducing risk as measured by the standard deviation of monthly returns (see Table 4.3). For example, for neither of the equities with tickers IFF and MCD do the Long-Only strategies outperform the Buy-and-Hold strategies on an absolute return basis, but they achieve significantly higher Sharpe ratios through a substantial reduction in the standard deviation of returns.

In contrast to the rather stellar results for the LSTM Long-Only strategy, only 18 out of 30 Sharpe ratios for the LSTM Long/Short strategy are significantly higher than the Buy-and-Hold strategy at the 5 per cent level. This increases to 22 out of 30 stocks for the GRU Long/Short strategies. Notwithstanding, the LSTM Long/Short Sharpe ratios are 2.7 times as high as those of the Buy-and-Hold strategy, on average. Moreover, these results are mostly achieved through higher returns, rather than a reduction in risk. As for the ARIMA benchmark models, only one of the Long-Only and Long/Short strategies achieve a significantly higher Sharpe ratio compared to the Buy-and-Hold strategy. Moreover, the ARIMA models generate negative Sharpe ratios for some of the stocks under consideration. On average, the LSTM Long/Only Sharpe ratio is an astonishing 114 times as high as that of the ARIMA models; the LSTM Long/Short Sharpe is on average 3.7 times as high as the ARIMA Long/Short Sharpe. The Sharpe ratios for the GRU models are similar to those of the LSTM models.

Table 4.5 Sharpe ratios for the Buy-and-Hold, the LSTM, the GRU, and the ARIMA trading strategies for each equity

	Sharpe	LSTM Sharpe		GRU Sharpe		ARIMA Sharpe	
Ticker	Buy&Hold	Long-Only	Long/Short	Long-Only	Long/Short	Long-Only	Long/Short
ADM	0.0755	0.1951***	0.2820**	0.4016***	0.4491**	0.0899	0.0474
ADP	0.3064	0.5138***	0.6168***	0.3026	0.2985	0.2077	-0.0603
AXP	0.1422	0.4288***	0.4655**	0.2571**	0.3200**	0.1260	0.0757
BAX	0.2294	0.3953*	0.2125	0.4877***	0.3688	0.1787	0.0358
BDX	0.2939	0.5613***	0.6749***	0.4416***	0.5299***	0.2981	0.1860
CLX	0.2149	0.3755**	0.2951	0.3560***	0.4377**	0.1864	0.0535
DIS	0.2093	0.2766**	0.3265**	0.3563***	0.4426**	0.2546	0.1913
DUK	0.2032	0.3274	0.1484	0.3088	0.1784	0.2263	0.1315
FDX	0.0782	0.2388***	0.3262**	0.4938***	0.6165***	-0.0368	-0.1387
GIS	0.0848	0.3720***	0.5530***	0.4983***	0.6871***	-0.1116	-0.2570
HON	0.3094	0.4397***	0.4924**	0.4074*	0.3920	0.4084	0.2064
HPQ	0.1296	0.3517***	0.5046***	0.4008***	0.3986*	0.1711	0.1269
IFF	0.1448	0.4066**	0.2332	0.3856**	0.3274	0.1889	0.0726
INTC	0.2557	0.4407***	0.4728*	0.3434	0.2017	0.1301	-0.0893
JNJ	0.2062	0.5348***	0.6922***	0.5467***	0.6807***	0.2702	0.1705
JPM	0.2790	0.4365**	0.4251	0.4656***	0.4863*	0.2367	0.1128
LNC	0.0957	0.5362***	0.7045***	0.2029**	0.2803**	0.1559	0.1199
MCD	0.3308	0.5335**	0.3857	0.6402***	0.7813***	0.0663	-0.2379
MMM	0.1396	0.2532**	0.3088*	0.5265***	0.6852***	0.0610	-0.0380
NEM	0.1121	0.2848**	0.3210*	0.3566***	0.4263**	0.0502	-0.0474
PCAR	0.1449	0.4970***	0.4045*	0.4981***	0.5848***	0.1463	0.1016
SLB	-0.1151	0.3322***	0.6156***	0.4135***	0.7475***	0.0035**	0.1211**
T	0.1399	0.2522***	0.3470***	0.3707***	0.4368**	0.1443	0.1051
TAP	0.1959	0.4703***	0.5165*	0.5567***	0.7854***	0.3271	0.2195
TGT	0.1988	0.4459***	0.5412**	0.4204***	0.5946***	-0.0080	-0.2099
TXT	0.1270	0.3737***	0.4566**	0.2831***	0.3914***	0.2290	0.2038
VFC	0.1765	0.4201***	0.5369***	0.4776***	0.6764***	0.2082	0.1224
WBA	0.0451	0.3726***	0.5724***	0.3284***	0.4597***	0.0648	0.0635
WMB	0.0313	0.4508***	0.6981***	0.3004***	0.4891***	-0.1202	-0.2279
WMT	0.1829	0.4854***	0.4689*	0.3850***	0.4880***	0.0998	-0.0743
Mean	0.1656	0.4001	0.4533	0.4071	0.4881	0.0035	0.1211

Note: * = p < .1, ** = p < .05, *** = p < 0.01, $H_0 : Sh_{BH} = Sh_{NN} \vee Sh_{BH} = Sh_{ARIMA}$

4.2.4 Equally-weighted Portfolio Performance

In this section, equally-weighted portfolios of the Buy-and-Hold, Long-Only, and Long/Short will be created for the LSTM, GRU, and ARIMA models. This more closely resembles a trading scenario in which funds hold and trade multiple equities at the same time.

Figure 4.2 shows the cumulative returns of the equally-weighted LSTM portfolios based on the three strategies, as mentioned in section 3.2.6. Each portfolio has a starting value of 1 US Dollar at the beginning of the out-of-sample period test period. For the first ten months of the out-of-sample period, the cumulative returns between the three portfolios do not seem to differ substantially. However, the portfolio values begin to diverge from the tenth month onward, with the Long-Only and Long/Short portfolios increasing at a substantially higher rate than the Buy-and-Hold portfolio. Figure B.1 shows the cumulative portfolio value for the GRU and ARIMA benchmark portfolios, the latter of which substantially underperformed compared to the Buy-and-Hold portfolio. The GRU Long-Only and Long/Short portfolios performed nearly identical to the LSTM portfolios.

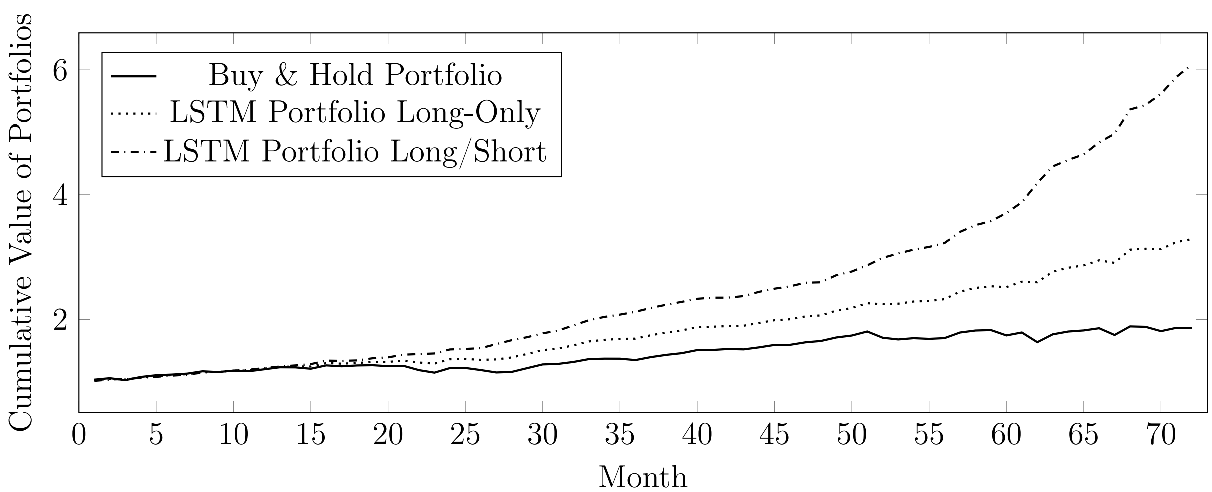

Figure 4.2 Out-of-sample cumulative portfolio value of equally-weighted LSTM portfolios compared to an equity buy-and-hold portfolio

Table 4.6 shows the trading summary statistics for the equally-weighted portfolios, as well as the Standard & Poor 500 Index over the out-of-sample test period. The S&P 500 Index is included as this may be considered a reasonable proxy for the market index. The results show that the equally-weighted portfolios of the LSTM and GRU neural network strategies outperform the Buy-and-Hold portfolio on an absolute return basis, with annualized returns significantly higher at the 1 per cent level. Beta captures the sensitivity of the portfolio to the market index returns and can therefore be used as a measure of sensitivity to systematic risk. Over the entire out-of-sample period, both the LSTM and GRU Long-Only as well as the Long/Short portfolios have lower betas than the Buy-and-Hold portfolio. However, the LSTM model are exposed to lower amounts of systematic risk than the GRU models, while generating near identical annualized returns. Moreover, the monthly Sharpe ratios that both the LSTM and GRU portfolios achieve are both statistically and economically significantly higher than that of the Buy-and-Hold portfolio. This is especially meaningful, as it implies that both the LSTM and GRU portfolios outperform on a risk-adjusted basis. Jensen's alpha is calculated for each of the strategies. This risk-adjusted performance measure calculates ex-post alpha; the mean portfolio return relative to the predicted return by the capital asset pricing model (Jensen, 1968). The LSTM Long-Only and Long/Short portfolios generated an ex-post monthly alpha of 1.22 and 2.46 per cent, respectively; this is similar to the alpha generated by the GRU models. The ARIMA models substantially underperform the Buy-and-Hold strategies on all measures.

Table 4.6 LSTM, GRU, and ARIMA trading evaluation statistics for the equally-weighted portfolios of the Buy-and-Hold, Long-Only, and Long/Short strategies

			LSTM		GRU		ARIMA	
	S&P 500	Buy-and-Hold	Long-Only	Long/Short	Long-Only	Long/Short	Long-Only	Long/Short
Jensen's Alpha	N/A	0.0006	0.0122***	0.0246***	0.0125***	0.0250***	0.0001	0.0003
Beta	1.0000	0.8688	0.4686	0.0967	0.4885	0.1371	0.5170	0.1564
Annualized returns	0.1171	0.1093	0.2194***	0.3512***	0.2252***	0.3639***	0.0613	0.0192
Sharpe	0.2998	0.2986	1.1478***	1.5656***	1.1623***	1.6284***	0.1737	0.0883
Max Drawdown	0.1596	0.1417	0.0708	0.0636	0.0717	0.0538	0.0894	0.0884
MAR Ratio	0.7336	0.7711	2.9414	5.7292	3.1409	6.7639	0.6861	0.2167

Note: * = $p < .1$, ** = $p < .05$, *** = $p < 0.01$

To capture another dimension of risk, the Maximum Drawdown is calculated for each equally-weighted portfolio. The results show that both LSTM portfolios show substantially lower maximum drawdowns of less than half those of the Buy-and-Hold and S&P 500 portfolios. Moreover, the MAR Ratio, a risk-adjusted performance measure commonly used to evaluate hedge funds, is calculated for each portfolio. This is the ratio of the annualized returns since inception over the maximum drawdown since inception. The Long-Only and Long/Short LSTM strategies show MAR Ratios of, respectively, 3.8 and 7.4 times higher than that of the Buy-and-Hold portfolio. This is achieved through both higher annualized returns as well as lower realized maximum drawdowns. The GRU Long-Only portfolio achieve a similar maximum drawdown as the corresponding LSTM portfolio, while the GRU Long/Short portfolio achieves a substantially lower maximum drawdown; both result in slightly higher MAR ratios as compared to the LSTM portfolios; On the contrary, while the benchmark ARIMA portfolios achieve lower maximum drawdowns than the Buy-and-Hold portfolio, they show lower MAR ratios as a result of substantially lower annualized returns.

4.3 LSTM Performance over Time and Market Efficiency

One may expect the LSTM edge to have been arbitraged away with increased usage of machine-learning techniques by stock market participants. If this is indeed the case, then the predictability of stock returns and stock price direction should have diminished over the out-of-sample test period. Figure 4.3 shows the Mean Absolute Error (MAE) and the mean directional inaccuracy, measured in the percentage of predictions that were incorrect. Both graphs do not show a clear trend but rather fluctuations around a constant mean. Hence, it seems that the effectiveness of this particular application of LSTM neural networks in conjunction with factors has not substantially been arbitraged away.

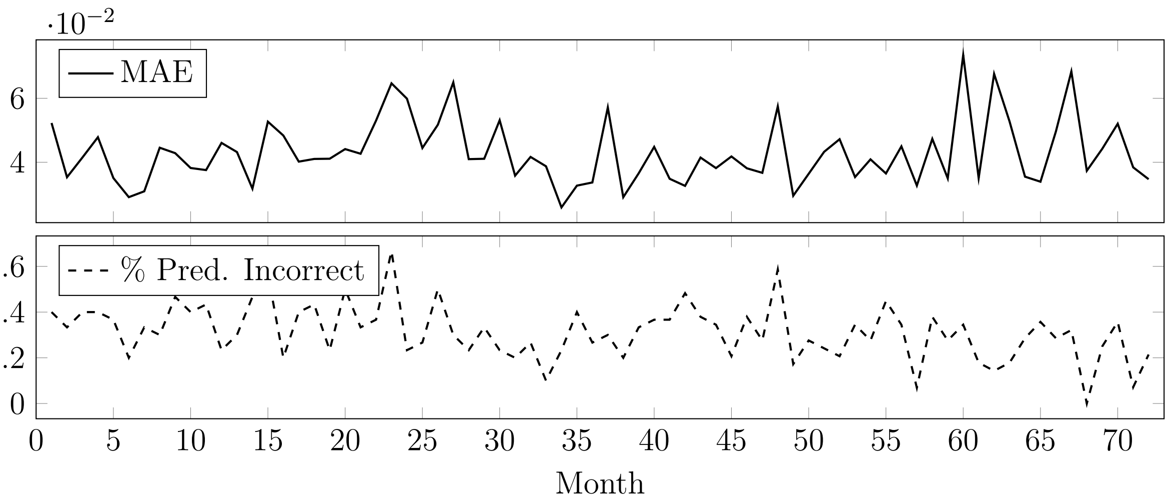

Figure 4.3 MAE & the percentage of predictions incorrect for each month over the entire out-of-sample period

Chapter Five

Conclusion and Discussion

5.1 Conclusion

This thesis applies Long Short-Term Memory and Gated Recurrent Unit neural networks and factor portfolios to the forecasting of stock returns of twenty S&P 500 constituent public equities, with an out-of-sample evaluation period of October 2013 until October 2019. The results suggest that the magnitude of one-month out stock returns cannot be accurately predicted using LSTM neural networks. However, the LSTM and GRU neural networks seem to be able to predict the direction of financial returns rather well, with accuracies well in excess of 50 per cent.

Two straightforward rules-based trading strategies are formulated based on the one-month out return forecast. Both the LSTM and GRU strategies lead to significantly higher risk-adjusted returns than both Buy-and-Hold strategies as well as holding the S&P 500. Namely, the results show that the Long-Only and Long/Short LSTM portfolio strategies produce statistically and economically significant ex-post monthly alpha of 1.22 and 2.46 per cent, respectively, after transaction costs, as well as significantly higher Sharpe ratios. Moreover, the two trading strategies have generally a lower systematic-risk exposure compared to the Buy-and-Hold and S&P 500 portfolios, as well as lower downside risk as measured by the Maximum Drawdown. This thesis is, in essence, an empirical test of the Efficient Market

Hypothesis, and its results challenge the weak form of the EMH.

The LSTM and GRU models substantially outperformed the benchmark ARIMA models on both the statistical evaluation as well as the out-of-sample trading strategy evaluation. These results suggest that there are substantial non-linearities and complex temporal relations present in financial markets, which the LSTM and GRU models were able to capture whilst the ARIMA models did not. Overall, the LSTM and GRU models perform near identically on a risk-adjusted basis. As such, a concrete conclusion as to which gated unit is better at forecasting financial returns cannot be made on the basis of these results.

5.2 Discussion

Further research could evaluate the effectiveness of combining factor portfolios with LSTM or GRU neural networks on equities other than S&P 500 constituents. It may be that the performance of such models is higher for stocks that are less frequently traded and receive less analyst coverage. Moreover, this thesis merely evaluated the effectiveness of LSTM and GRU neural networks over a one-month prediction horizon; further research could focus on the performance of such neural networks over weekly or even daily horizons. It may be that the differences between the two different architectures is more pronounced over different time horizons. Moreover, although a relatively large number of factors is used in this thesis, a different set of factors or the inclusion of other variables such as currency exchange rates, inflation, and interest rates, may also be used as inputs.

Several comments of caution regarding the application of LSTM Neural Networks to trading strategies must be made. First, while the LSTM models as applied in this thesis seem to be able to generate superior out-of-sample performance than a simple Buy-and-Hold strategy or holding the S&P 500, this is no guarantee that these strategies will continue to perform well in the future. Moreover, it may be that the LSTM strategies are exposed to higher amounts of risk not captured by standard deviation, the Sharpe ratio or the Capital Asset Pricing Model. Hence, any alpha generated could merely be compensation for this increased amount of risk.

Furthermore, the use of LSTM neural networks in predicting financial returns has another disadvantage: the lack of transparency and intractability of return forecasts (Nakagawa et al., 2019). Since the model that generates these returns inherently is a 'black-box', due to the complexity of the generated models and the inability to infer why certain inputs are combined and manipulated in the way that the models do. As such, one is not able to extract an interpretable 'reasoning process' from the model. Hence, institutional investors may forgo using machine learning techniques as their lack of transparency may hinder accountability to their customers (Nakagawa et al., 2019).

REFERENCES

Abadi, Martın et al. (2016). "Tensorflow: Large-scale machine learning on heterogeneous distributed systems". In: *arXiv preprint arXiv:1603.04467*.

Abhyankar, Abhay, Laurence S Copeland, and Woon Wong (1997). "Uncovering nonlinear structure in real-time stock-market indexes: the S&P 500, the DAX, the Nikkei 225, and the FTSE-100". In: *Journal of Business & Economic Statistics* 15.1, pp. 1–14.

Ahoniemi, Katja (2008). "Modeling and forecasting the VIX index". In: *Available at SSRN 1033812*.

Avellaneda, Marco and Jeong-Hyun Lee (2010). "Statistical arbitrage in the US equities market". In: *Quantitative Finance* 10.7, pp. 761–782.

Bishop, Christopher M et al. (1995). *Neural networks for pattern recognition*. Oxford university press.

Cho, Kyunghyun et al. (2014). "On the properties of neural machine translation: Encoder-decoder approaches". In: *arXiv preprint arXiv:1409.1259*.

Chollet, François et al. (2015). "keras, GitHub". In: *GitHub repository, https://github. com/fchollet/keras*.

Chung, Junyoung et al. (2014). "Empirical evaluation of gated recurrent neural networks on sequence modeling". In: *arXiv preprint arXiv:1412.3555*.

Claesen, Marc and Bart De Moor (2015). "Hyperparameter search in machine learning". In: *arXiv preprint arXiv:1502.02127*.

Cybenko, George (1989). "Approximation by superpositions of a sigmoidal function". In: *Mathematics of control, signals and systems* 2.4, pp. 303–314.

Fama, Eugene F and Kenneth R French (1993). "Common risk factors in the returns on stocks and bonds". In: *Journal of*.

— (2015). "A five-factor asset pricing model". In: *Journal of financial economics* 116.1, pp. 1–22.

Fischer, Thomas and Christopher Krauss (2018). "Deep learning with long short-term memory networks for financial market predictions". In: *European Journal of Operational Research* 270.2, pp. 654–669.

Gers, Felix A, Jürgen Schmidhuber, and Fred Cummins (1999). "Learning to forget: Continual prediction with LSTM". In:

Hill, Tim, Marcus O'Connor, and William Remus (1996). "Neural network models for time series forecasts". In: *Management science* 42.7, pp. 1082–1092.

Hochreiter, Sepp (1998). "The vanishing gradient problem during learning recurrent neural nets and problem solutions". In: *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 6.02, pp. 107–116.

Hochreiter, Sepp and Jürgen Schmidhuber (1997). "Long short-term memory". In: *Neural computation* 9.8, pp. 1735–1780.

Hou, Kewei, Chen Xue, and Lu Zhang (2017). "Replicating anomalies". In: *The Review of Financial Studies.*

Huck, Nicolas (2009). "Pairs selection and outranking: An application to the S&P 100 index". In: *European Journal of Operational Research* 196.2, pp. 819–825.

Jensen, Michael C (1968). "The performance of mutual funds in the period 1945–1964". In: *The Journal of finance* 23.2, pp. 389–416.

Jorion, Philippe (1992). "Portfolio optimization in practice". In: *Financial Analysts Journal* 48.1, pp. 68–74.

Kawakami, Kazuya (2008). "Supervised sequence labelling with recurrent neural networks". In: *Ph. D. dissertation, PhD thesis. Ph. D. thesis.*

Kendall, Maurice George and A Bradford Hill (1953). "The analysis of economic time-series-part i: Prices". In: *Journal of the Royal Statistical Society. Series A (General)* 116.1, pp. 11–34.

Kristjanpoller, Werner and Marcel C Minutolo (2016). "Forecasting volatility of oil price using an artificial neural network-GARCH model". In: *Expert Systems with Applications* 65, pp. 233–241.

Lago, Jesus, Fjo De Ridder, and Bart De Schutter (2018). "Forecasting spot electricity prices: Deep learning approaches and empirical comparison of traditional algorithms". In: *Applied Energy* 221, pp. 386–405.

Li, Jialiang, Wenyang Zhang, and Efang Kong (2018). "Factor models for asset returns based on transformed factors". In: *Journal of econometrics* 207.2, pp. 432–448.

Livingstone, David J, David T Manallack, and Igor V Tetko (1997). “Data modelling with neural networks: advantages and limitations”. In: *Journal of computer-aided molecular design* 11.2, pp. 135–142.

McKinney, Wes et al. (2011). “pandas: a foundational Python library for data analysis and statistics”. In: *Python for High Performance and Scientific Computing* 14.9.

Memmel, Christoph (2003). “Performance hypothesis testing with the Sharpe ratio”. In: *Finance Letters* 1.1.

Monfared, Soheil Almasi and David Enke (2014). “Volatility forecasting using a hybrid GJR-GARCH neural network model”. In: *Procedia Computer Science* 36, pp. 246–253.

Motiwalla, Luvai and Mahmoud Wahab (2000). “Predictable variation and profitable trading of US equities: a trading simulation using neural networks”. In: *Computers & Operations Research* 27.11-12, pp. 1111–1129.

Nakagawa, Kei et al. (2019). “Deep recurrent factor model: interpretable non-linear and time-varying multi-factor Model”. In: *arXiv preprint arXiv:1901.11493*.

Olah (2015). “Understanding LSTM networks”. In: *Understanding LSTM Networks-Colah’s Blog. Github.*

Pai, Ping-Feng and Chih-Sheng Lin (2005). “A hybrid ARIMA and support vector machines model in stock price forecasting”. In: *Omega* 33.6, pp. 497–505.

Pedregosa, Fabian et al. (2011). “Scikit-learn: Machine learning in Python”. In: *Journal of machine learning research* 12.Oct, pp. 2825–2830.

Roh, Tae Hyup (2007). “Forecasting the volatility of stock price index”. In: *Expert Systems with Applications* 33.4, pp. 916–922.

Ryll, Lukas and Sebastian Seidens (2019). “Evaluating the Performance of Machine Learning Algorithms in Financial Market Forecasting: A Comprehensive Survey”. In: *arXiv preprint arXiv:1906.07786*.

Sak, Hasim, Andrew W Senior, and Françoise Beaufays (2014). “Long short-term memory recurrent neural network architectures for large scale acoustic modeling”. In:

Srivastava, Nitish et al. (2014). “Dropout: a simple way to prevent neural networks from overfitting”. In: *The journal of machine learning research* 15.1, pp. 1929–1958.

Tieleman, Tijmen and Geoffrey Hinton (2012). “Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude”. In: *COURSERA: Neural networks for machine learning* 4.2, pp. 26–31.

Tu, Jack V (1996). "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes". In: *Journal of clinical epidemiology* 49.11, pp. 1225–1231.

Van Hemert, Otto et al. (2020). "Drawdowns". In: *Available at SSRN 3583864*.

Walt, Stéfan van der, S Chris Colbert, and Gael Varoquaux (2011). "The NumPy array: a structure for efficient numerical computation". In: *Computing in Science & Engineering* 13.2, pp. 22–30.

Zen, Heiga and Haşim Sak (2015). "Unidirectional long short-term memory recurrent neural network with recurrent output layer for low-latency speech synthesis". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, pp. 4470–4474.

Zhao, Mingxiong et al. (2019). "Construction of an industrial knowledge graph for unstructured Chinese text learning". In: *Applied Sciences* 9.13, p. 2720.

APPENDIX

Appendix A

Table A.1 Factors used as inputs in the LSTM neural network (Hou, Xue, and L. Zhang, 2017))

#	Category	Factor description
1	**Momentum**	Cumulative abnormal returns (CAR) around earnings announcements, 1-month horizon
2		CAR, 6-month horizon
3		Standardized Unexpected Earnings (SUE), 1-month horizon
4		SUE, 6-month horizon
5		Revisions in Analyst Earnings Forecasts, 1-month horizon
6		Revisions in Analyst Earnings Forecasts, 6-month horizon
7		Prior 6-month returns
8		Prior 11-month returns
9	**Value / Growth**	Book-to-Market equity (B/M)
10		Long-term reversal, 12-month horizon
11		Earnings-to-Price (E/P)
12		Cashflow-to-Price (Cf/P)
13		Payout-Yield (dividends + repurchases)
14		Net payout yield (payouts minus equity issuances)
15		Enterprise multiple (enterprise value/operating income)
16		Sales-to-Price (S/P)
17		Operating Cashflow-to-Price (OCf/P)
18	**Investment**	Investment-to-assets (I/A)
19		Changes in Property, Plant, and Equipment (PPE), and Inventory-to-Assets
20		Net Operating Assets
21		Changes in Net Operating Assets

#	Category	Factor description
22		Investment Growth
23		Net Stock Issues
24		Composite Equity Issuance
25		Operating Accruals
26		Total Accruals
27		Discretionary Accruals
28		Percent Operating Accruals
29	**Profitability**	Return on Equity (ROE), 1-month horizon
30		Return on Equity (ROE), 6-month horizon
31		4-quarter change in ROE, 1-month horizon
32		4-quarter change in ROE, 6-month horizon
33		Operating Profits-to-Book Equity
34		Operating Profits-to-Assets
35		Operating Cashflow-to-Assets
36		Expected Growth
37	**Intangibles**	Organizational Capital-to-Assets
38		Advertising Expense-to-Market
39		R&D Expense-to-Market
40		Industry-Adjusted Real Estate Ratio
41		Seasonality, return in month t-12
42		Seasonality, average returns across months t-24, t-36, t-48, and t-60
43		Seasonality, average returns across months t-72, t-84, t-96, t-108, and t-120
44		Seasonality, average returns across months t-132, t-144, t-156, t-168, and t-180
45		Seasonality, average returns across months t-192, t-204, t-216, t-228, and t-240
46	**Trading Frictions**	Market Equity
47		Total volatility, 1-month horizon

#	Category	Factor description
48		Idiosyncratic volatility estimated from the FF 3-factor model, 1-month horizon
49		Systematic volatility risk, 1-month horizon
50		Market Beta, 1-month horizon
51		Short-term Reversal

Table A.2 List of S&P 500 constituent stocks used for equity return prediction

Firm name	Ticker	Industry (GICS*)
3M Company	MMM	Industrial Conglomerates
American Express Co	AXP	Consumer Finance
Archer-Daniels-Midland Co	ADM	Food Products
AT&T	T	Diversified Telecommunication Services
Automatic Data Processing	ADP	IT Services
Baxter International Inc.	BAX	Health Care Equipment & Supplies
Becton Dickinson	BDX	Health Care Equipment & Supplies
Duke Energy	DUK	Electric Utilities
FedEx Corporation	FDX	Air Freight & Logistics
General Mills	GIS	Food Products
Honeywell Int'l Inc.	HON	Industrial Conglomerates
HP Inc.	HPQ	Technology Hardware, Storage & Peripherals
Intel Corp.	INTC	Semiconductors & Semiconductor Equipment
Intl Flavors & Fragrances	IFF	Chemicals
Johnson & Johnson	JNJ	Pharmaceuticals
JPMorgan Chase & Co.	JPM	Diversified Banks
Lincoln National	LNC	Insurance
McDonald's Corp.	MCD	Hotels, Restaurants & Leisure
Molson Coors Brewing Company	TAP	Beverages
Newmont Corporation	NEM	Metals & Mining
PACCAR Inc.	PCAR	Machinery
Schlumberger Ltd.	SLB	Energy Equipment & Services
Target Corp.	TGT	Multiline Retail
Textron Inc.	TXT	Aerospace & Defense
The Clorox Company	CLX	Household Products
The Walt Disney Company	DIS	Entertainment
V.F. Corp.	VFC	Textiles, Apparel & Luxury Goods
Walgreens Boots Alliance	WBA	Food & Staples Retailing
Walmart	WMT	Food & Staples Retailing
Williams Cos.	WMB	Oil, Gas & Consumable Fuels

*GICS: Global Industry Classification Standard

Appendix B

Table B.1 LSTM, GRU, and ARIMA out-of-sample R-squared, MAE, RMSE, and MAPE for each of the equities

	LSTM				GRU				ARIMA			
Ticker	R^2	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE	R^2	MAE	RMSE	MAPE
ADM	0.1159	0.0514	0.0727	2.9777	0.3445	0.0411	0.0530	1.8312	0.0273	0.0792	0.0957	4.7841
ADP	0.3689	0.0329	0.0414	4.6274	0.1105	0.0792	0.0915	9.9655	0.0102	0.0966	0.1150	7.5887
AXP	0.2954	0.0378	0.0520	1.7720	0.1720	0.0421	0.0593	2.1236	0.0421	0.0587	0.0759	2.7527
BAX	0.3392	0.0575	0.0654	6.5011	0.3484	0.0405	0.0522	3.3689	0.0197	0.0635	0.0809	6.7926
BDX	0.6062	0.0244	0.0324	1.5363	0.5109	0.0392	0.0484	5.1650	0.0664	0.0573	0.0739	4.2026
CLX	0.3716	0.0320	0.0414	3.4474	0.3900	0.0386	0.0464	4.7768	0.0005	0.0636	0.0788	9.6853
DIS	0.2997	0.0541	0.0627	4.7417	0.4381	0.0429	0.0524	3.0634	0.0141	0.0703	0.0861	6.3629
DUK	0.1722	0.0362	0.0458	1.9825	0.4741	0.0376	0.0436	3.8018	0.0203	0.0447	0.0545	3.9192
FDX	0.3344	0.0544	0.0713	12.9576	0.5462	0.0395	0.0521	3.2759	0.0016	0.0919	0.1141	36.1931
GIS	0.5181	0.0307	0.0381	2.9508	0.6666	0.0291	0.0370	3.4120	0.0364	0.0638	0.0801	5.3179
HON	0.2657	0.0353	0.0470	9.9708	0.2879	0.0290	0.0368	2.1447	0.0980	0.0491	0.0614	13.9298
HPQ	0.2929	0.0586	0.0792	2.5389	0.3604	0.0612	0.0736	3.7652	0.0207	0.0810	0.1043	5.0929
IFF	0.4146	0.0464	0.0587	1.6432	0.4224	0.0475	0.0569	2.0057	0.0133	0.0680	0.0867	2.2837
INTC	0.3035	0.0412	0.0556	2.3941	0.2514	0.0596	0.0753	3.0190	0.0013	0.0831	0.1035	4.0787
JNJ	0.5390	0.0209	0.0272	1.1459	0.5492	0.0255	0.0309	1.5473	0.0166	0.0468	0.0588	2.6672
JPM	0.3051	0.0408	0.0547	1.7786	0.3435	0.0369	0.0513	1.9699	0.0125	0.0817	0.1012	4.8921
LNC	0.4376	0.0540	0.0719	1.2955	0.3410	0.0696	0.0893	1.9810	0.0606	0.0853	0.1124	2.2680
MCD	0.3888	0.0318	0.0404	2.8708	0.4330	0.0229	0.0301	3.0325	0.0105	0.0581	0.0711	4.3683
MMM	0.3318	0.0450	0.0555	5.1535	0.5076	0.0275	0.0376	2.6099	0.0100	0.0643	0.0796	10.3217
NEM	0.3151	0.0604	0.0915	3.3367	0.2561	0.0713	0.1019	2.5899	0.0022	0.1449	0.1874	3.9704
PCAR	0.4138	0.0495	0.0647	2.0815	0.4682	0.0420	0.0555	1.8279	0.0370	0.0772	0.0962	3.0285
SLB	0.2783	0.0480	0.0667	1.7544	0.4487	0.0452	0.0602	2.6882	0.0537	0.0781	0.1042	5.4334
T	0.3534	0.0392	0.0508	3.0605	0.3173	0.0312	0.0418	2.3414	0.0017	0.0613	0.0791	4.9305
TAP	0.5474	0.0390	0.0513	1.4316	0.6887	0.0260	0.0370	2.4429	0.0807	0.0719	0.1001	3.2212
TGT	0.2762	0.0449	0.0630	5.8010	0.4073	0.0435	0.0601	9.6958	0.0388	0.0944	0.1202	41.7786
TXT	0.2141	0.0552	0.0720	3.1879	0.3948	0.0434	0.0648	3.3039	0.0626	0.0732	0.1005	3.1993
VFC	0.4530	0.0356	0.0446	9.4954	0.5123	0.0339	0.0428	10.7434	0.0296	0.0606	0.0737	16.3471
WBA	0.4831	0.0409	0.0533	1.6026	0.4532	0.0425	0.0564	1.8729	0.0027	0.0813	0.1048	4.3434
WMB	0.5189	0.0565	0.0764	2.2690	0.5182	0.0715	0.0890	2.8979	0.1009	0.1282	0.1658	4.2653
WMT	0.2970	0.0307	0.0432	4.9398	0.3328	0.0330	0.0451	5.2523	0.0008	0.0636	0.0795	15.7869
Mean	0.3617	0.0428	0.0564	3.7082	0.3877	0.0440	0.0560	3.2725	0.0242	0.0726	0.0911	7.0736

Table B.2 Fraction of periods invested as well as out-of-sample directional accuracy of the ARIMA benchmark models

Ticker	Fraction of Periods Long	Directional Accuracy	Upwards Acc.	Downwards Acc.
ADM	0.4444	50.00%	53.13%	47.50%
ADP	0.5000	43.06%	55.56%	30.56%
AXP	0.5694	55.56%	63.41%**	45.16%
BAX	0.4675	51.95%	63.89%**	41.46%
BDX	0.5833	51.39%	61.90%**	36.67%
CLX	0.5417	47.22%	53.85%	39.39%
DIS	0.5417	51.39%	53.85%	48.48%
DUK	0.5278	63.89%***	71.05%***	55.88%
FDX	0.6528	47.22%	57.45%	28.00%
GIS	0.4861	37.50%	37.14%	37.84%
HON	0.4444	55.56%	75.00%***	40.00%
HPQ	0.5417	59.72%**	69.23%***	48.48%
IFF	0.4444	54.17%	62.50%*	47.50%
INTC	0.4861	50.00%	60.00%*	40.54%
JNJ	0.5556	50.00%	62.50%**	34.38%
JPM	0.5694	48.61%	56.10%	38.71%
LNC	0.5694	56.94%*	63.41%**	48.39%
MCD	0.4306	37.50%	48.39%	29.27%
MMM	0.4861	50.00%	62.86%**	37.84%
NEM	0.4583	48.61%	51.52%	46.15%
PCAR	0.5278	47.22%	50.00%	44.12%
SLB	0.5694	51.39%	43.90%	61.29%*
T	0.5556	47.22%	50.00%	43.75%
TAP	0.5122	65.85%**	76.19%***	55.00%
TGT	0.5333	50.00%	59.38%	39.29%
TXT	0.4861	61.11%**	65.71%**	56.76%
VFC	0.5556	47.22%	55.00%	37.50%
WBA	0.5972	51.39%	48.84%	55.17%
WMB	0.5556	43.06%	47.50%	37.50%
WMT	0.4167	54.17%	56.67%	52.38%

Note: * = p < .1, ** = p < .05, *** = p < 0.01, $H_0 = 50\%$

Table B.3 Fraction of periods invested as well as out-of-sample directional accuracy of the GRU models

Ticker	Fraction of Periods Long	Directional Accuracy	Upwards Acc.	Downwards Acc.
ADM	0.3056	69.44%***	86.36%***	62.00%**
ADP	0.9722	58.33%*	60.00%**	0.000%
AXP	0.8472	63.89%***	63.93%**	63.64%
BAX	0.3472	62.50%**	84.00%***	51.06%
BDX	0.9167	70.83%***	68.18%***	100.0%***
CLX	0.8889	62.50%**	60.94%**	75.00%**
DIS	0.9167	61.11%**	57.58%*	100.0%***
DUK	0.1667	55.56%	91.67%***	48.33%
FDX	0.6944	73.61%***	76.00%***	68.18%**
GIS	0.3056	75.00%***	90.91%***	68.00%***
HON	0.5278	63.89%***	78.95%***	47.06%
HPQ	0.2778	61.11%**	90.00%***	50.00%
IFF	0.2917	61.11%**	80.95%***	52.94%
INTC	0.2083	55.56%	86.67%***	47.37%
JNJ	0.7500	77.78%***	77.78%***	77.78%***
JPM	0.4861	70.83%***	80.00%***	62.16%**
LNC	0.9028	62.50%**	61.54%**	71.43%*
MCD	0.6806	79.17%***	79.59%***	78.26%***
MMM	0.7083	77.78%***	78.43%***	76.19%***
NEM	0.4028	68.06%***	75.86%***	62.79%**
PCAR	0.3194	73.61%***	91.30%***	65.31%**
SLB	0.2778	77.78%***	85.00%***	75.00%***
T	0.5694	65.28%***	65.85%**	64.52%**
TAP	0.5854	78.05%***	83.33%***	70.59%**
TGT	0.8000	76.67%***	72.92%***	91.67%***
TXT	0.7639	63.89%***	61.82%**	70.59%**
VFC	0.6667	75.00%***	75.00%***	75.00%***
WBA	0.3056	69.44%***	77.27%***	66.00%***
WMB	0.8056	68.06%***	63.79%**	85.71%***
WMT	0.7917	63.89%***	59.65%*	80.00%***

Note: * = $p < .1$, ** = $p < .05$, *** = $p < 0.01$, $H_0 = 50\%$

Table B.4 ARIMA annualized geometric mean returns and standard deviations of monthly returns over the out-of-sample period for each stock

	Annualized Returns			Standard Deviation		
Ticker	Buy&Hold	Long-Only	Long/Short	Buy&Hold	Long-Only	Long/Short
ADM	0.0327	0.0320	0.0102	0.0603	0.0412	0.0604
ADP	0.1894	0.0690	-0.0540	0.0479	0.0304	0.0503
AXP	0.0777	0.0561	0.0285	0.0553	0.0479	0.0557
BAX	0.1490	0.0946	0.0261	0.0583	0.0433	0.0581
BDX	0.1781	0.1444	0.1023	0.0512	0.0416	0.0525
CLX	0.1155	0.0694	0.0142	0.0478	0.0346	0.0488
DIS	0.1285	0.1273	0.1129	0.0555	0.0438	0.0557
DUK	0.0925	0.0758	0.0527	0.0404	0.0300	0.0409
FDX	0.0349	-0.0480	-0.1399	0.0701	0.0569	0.0697
GIS	0.0365	-0.0587	-0.1544	0.0499	0.0369	0.0485
HON	0.1538	0.1294	0.0951	0.0415	0.0265	0.0426
HPQ	0.0834	0.0934	0.0773	0.0715	0.0536	0.0715
IFF	0.0880	0.0734	0.0287	0.0628	0.0362	0.0633
INTC	0.1832	0.0503	-0.0938	0.0628	0.0396	0.0646
JNJ	0.0905	0.0838	0.0702	0.0388	0.0271	0.0390
JPM	0.1895	0.1285	0.0587	0.0582	0.0486	0.0601
LNC	0.0562	0.0935	0.0786	0.0895	0.0617	0.0892
MCD	0.1592	0.0145	-0.1216	0.0398	0.0261	0.0408
MMM	0.0733	0.0170	-0.0436	0.0523	0.0435	0.0528
NEM	0.0808	0.0093	-0.1367	0.1104	0.0709	0.1110
PCAR	0.0919	0.0789	0.0532	0.0654	0.0552	0.0657
SLB	-0.1365	-0.0273**	0.0763**	0.0785	0.0670	0.0784
T	0.0672	0.0574	0.0433	0.0466	0.0396	0.0468
TAP	0.1347	0.1575	0.1500	0.0644	0.0408	0.0641
TGT	0.1532	-0.0217	-0.1993	0.0731	0.0489	0.0729
TXT	0.0835	0.1372	0.1580	0.0766	0.0549	0.0757
VFC	0.1096	0.0968	0.0653	0.0590	0.0425	0.0595
WBA	0.0081	0.0221	0.0211	0.0708	0.0593	0.0707
WMB	-0.0231	-0.1449	-0.2812	0.0990	0.0786	0.0966
WMT	0.1004	0.0253	-0.0634	0.0509	0.0274	0.0516
Mean	0.0927	0.0545	0.0012	0.0616	0.0452	0.0619

Note: * = p < .1, ** = p < .05, *** = p < 0.01, $H_0 : \overline{R}_{BH} = \overline{R}_{ARIMA}$

Table B.5 GRU annualized geometric mean returns and standard deviations of monthly returns over the out-of-sample period for each stock

	Annualized Returns			Standard Deviation		
Ticker	Buy&Hold	Long-Only	Long/Short	Buy&Hold	Long-Only	Long/Short
ADM	0.0327	0.1782**	0.3143**	0.0603	0.0365	0.0551
ADP	0.1744	0.1716	0.1691	0.0476	0.0477	0.0477
AXP	0.0777	0.1412**	0.2016*	0.0553	0.0484	0.0532
BAX	0.1508	0.2090	0.2448	0.0566	0.0344	0.0546
BDX	0.1781	0.2524**	0.3265**	0.0512	0.0453	0.0472
CLX	0.1155	0.1813**	0.2465**	0.0478	0.0420	0.0448
DIS	0.1285	0.2115**	0.2928**	0.0555	0.0485	0.0519
DUK	0.0925	0.0907	0.0778	0.0404	0.0250	0.0406
FDX	0.0349	0.2694***	0.5114***	0.0701	0.0429	0.0599
GIS	0.0365	0.2039***	0.3820***	0.0499	0.0327	0.0413
HON	0.1538	0.1748	0.1931	0.0415	0.0352	0.0405
HPQ	0.0834	0.2214*	0.3337*	0.0715	0.0448	0.0669
IFF	0.0880	0.1744	0.2346	0.0628	0.0374	0.0603
INTC	0.1832	0.1742	0.1363	0.0628	0.0421	0.0636
JNJ	0.0905	0.1894***	0.2916***	0.0388	0.0277	0.0327
JPM	0.1895	0.2694*	0.3409*	0.0582	0.0458	0.0544
LNC	0.0562	0.1693**	0.2759**	0.0895	0.0809	0.0865
MCD	0.1592	0.2513***	0.3464***	0.0398	0.0304	0.0330
MMM	0.0733	0.2356***	0.4052***	0.0523	0.0353	0.0436
NEM	0.0808	0.3440**	0.5701**	0.1104	0.0785	0.1022
PCAR	0.0919	0.2737***	0.4526***	0.0654	0.0430	0.0570
SLB	-0.1365	0.2377***	0.6999***	0.0785	0.0463	0.0633
T	0.0672	0.1516**	0.2336**	0.0466	0.0342	0.0431
TAP	0.1347	0.3484***	0.5824***	0.0644	0.0477	0.0516
TGT	0.1532	0.3337***	0.5280***	0.0731	0.0626	0.0640
TXT	0.0835	0.2162***	0.3499***	0.0766	0.0669	0.0719
VFC	0.1096	0.279***	0.4614***	0.0590	0.0460	0.0496
WBA	0.0081	0.1950***	0.3821***	0.0708	0.0499	0.0644
WMB	-0.0231	0.2709***	0.5918***	0.0990	0.0776	0.0890
WMT	0.1004	0.1953***	0.2901**	0.0509	0.0415	0.0465
Mean	0.0923	0.2205	0.3489	0.0616	0.0459	0.0560

Note: * = p < .1, ** = p < .05, *** = p < 0.01, $H_0 : \overline{R}_{BH} = \overline{R}_{GRU}$

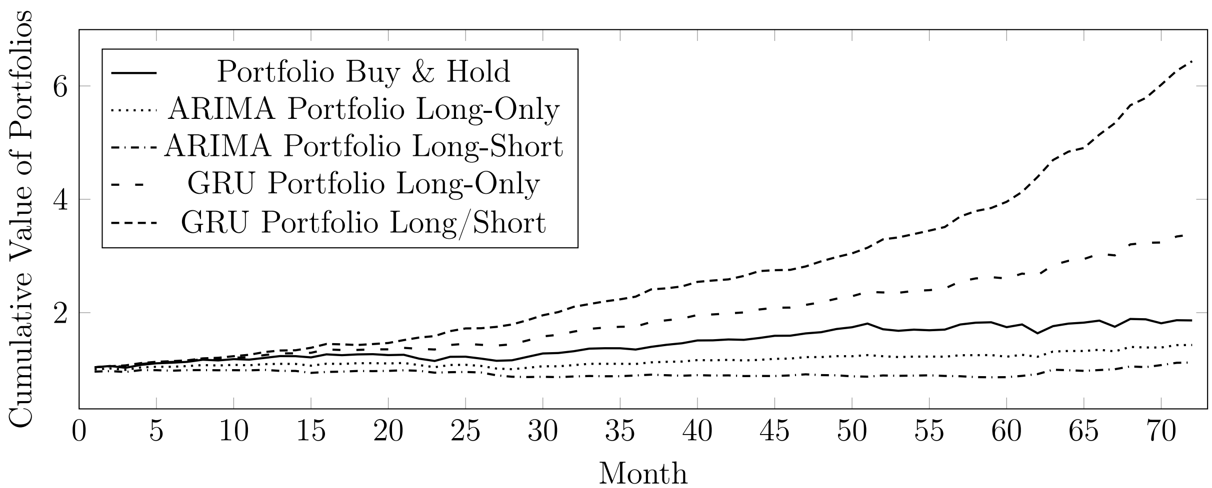

Figure B.1 Equally-weighted ARIMA and GRU portfolios out-of-sample performance compared to an equity buy-and-hold strategy