Event Study vs Machine Learning:

Challenging the Fundamentals of the Event Study Methodology

Florian Miedema

Thesis for Master Data Science & Society

Academic year: 2021-2022

Tilburg University

School of Humanities and Digital Sciences

Department of Cognitive Science and Artificial Intelligence

Supervisor: Dr. Gonzalo Nápoles

Second reader: Dr. Afra Alishahi

SNR: 2047951

Email: f.s.miedema@tilburguniversity.edu

Word count: 8712

Abstract

The event study methodology is among the most influential models in the field of finance. Numerous studies implemented this technique to evaluate the impact of a given event on stock prices. However, the traditional linear approach of the event study methodology lacks the flexibility and predictive power to generate valid estimations of the normal returns. The research question in this study aims to identify a superior model for this task using state-of-the-art machine learning techniques. The current literature lacks research in this field, since the event study methodology has not been revised in over 50 years.

This study has been conducted using historical stock returns of 1,000 European companies between 2010 and 2020. The main finding is the significant predictive performance of the LSTM model, which predicted normal returns with an RMSE of 0.2661. In comparison, the linear event study methodology showed an RMSE of over 13 times higher. The difference in performance persisted across multiple event window sizes. Other recurrent neural network architectures, namely SRN and GRU, also performed better than the event study methodology, yet failed to outperform the LSTM in terms of RMSE. However, bear in mind that model selection is more than solely predictive power. It is a trade-off between accuracy and transparency. As machine learning is increasingly adopted to make more important decisions than ever before, society seeks to shed light on the “black box”.

Table of Contents

Chapter 1: Introduction

This section provides an overview of the problem statement, research goal and research questions, as well as a brief summary of the main findings.

1.1 Problem Statement & Research Goal

The world might be on a verge of revolutionizing one of the most influential financial models of all time. The goal of this research is to improve the predictive performance of the traditional event study methodology using recurrent neural networks.

New information moves stock prices. This was the finding of the breakthrough Efficient Market Hypothesis (Fama, 1970). The event study methodology is the main model to quantify the impact of new information on stock prices. Widely used among various fields, event studies are meant to isolate the effect of mergers, annual reports, regulatory transitions or any other potentially impactful events. In addition to insight on the historical impact, the event study reveals important information about how securities are likely to react to a given event in the future (Bash & Alsaifi, 2018).

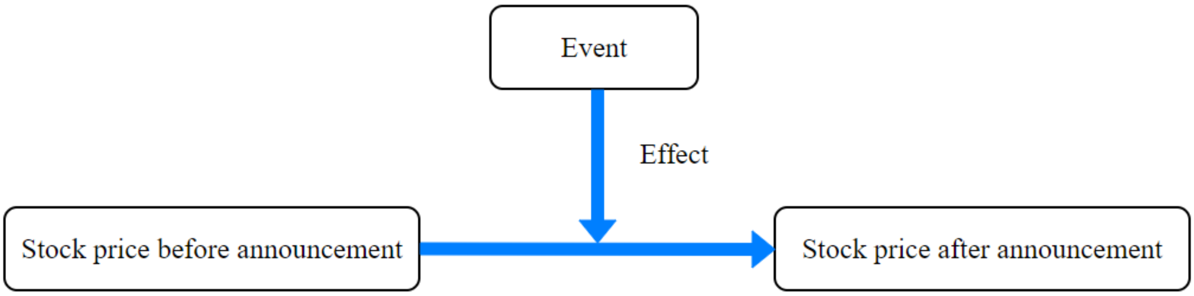

Figure 1: Simplified overview of the event study methodology. Created by F.S. Miedema.

Stock prices are characterized as dynamic, chaotic, noisy and non-linear in nature. The continuous flow of news, fluctuating economies and conflicting investor's perception make stock returns difficult to predict. Therefore, the problem statement is as follows. *Given the complexity of stock returns, the linear approach of the traditional event study methodology lacks flexibility and predictive power to produce valid estimations of event impact on stock prices*. The existing methodology is deficient in accuracy, which leads to unreliable estimations. Therefore, the research goal of this study is to improve the predictive performance of the event study.

From a scientific point of view, this problem is worth addressing because it is an opportunity to improve an established and widely used methodology. This study provides a proposal to revolutionize one of the most prominent models in the financial scientific field, making it both important and interesting. This study is relevant, because it fills a gap in the current scientific knowledge, as the present state of literature lacks research in this field. This is addressed extensively in Chapter 2: Literature Review. From a societal and practical point of view, this problem is worth addressing because society continuously seeks advanced empirical analysis methods to optimize stock price predictions. At the moment, the community is deficient in predictive power when estimating the impact of an event on stock prices. Stock markets are an important area of asset allocation for investors worldwide. This study adds significant value for all kinds of investors, such as companies, pension funds and households. Understanding the relationship between a given event and stock prices provides insight in the dynamics of the market. This allows investors to anticipate on likely stock price reactions whenever a given event occurs, and thus gain control over the price movements within their portfolio. This study provides a progressive contribution to the worldwide scientific field and society by proposing a modernized approach for event impact estimation on stock prices.

1.2 Research Questions

This subsection presents the main research question and sub-research questions (SRQ). The sub-research questions contribute to answering the main research question by improving the robustness of the results (SRQ1), performing an analysis of the errors (SRQ2), and comparing the performance across several suitable models (SRQ3). Common analysis techniques such as disparate biases across groups and feature selection were unavailable given the nature of the data.

Main research question:

"To what extent does an LSTM outperform the predictive power of the event study methodology, in terms of RMSE, when estimating the normal returns of European stocks between 2010 and 2020?"

The current literature contains a clear gap in the field of event study development, as described in Chapter 2: Literature Review. The methodology lacks predictive power and has not been revised in over 50 years. This main research question examines a design to fill this gap in the literature by using the state-of-the-art LSTM recurrent neural network, and solve the research goal.

This study aims to identify a superior model in the field of estimating normal returns for European stocks, as the literature does not contain research in this geographical area. This research design allows the literature to gain concrete, contextual, in-depth knowledge about a specific stock market. The forecasting task in this thesis provides insight in the market dynamics of one of the world's largest stock exchanges.

Sub-research question 1:

"How does the performance, in terms of RMSE, of the event study methodology and LSTM change when implementing several multi-day event windows?"

The Efficient Market Hypothesis (Fama, 1970) states that stock price reactions happen as soon as new information becomes available for investors, which nowadays means a matter of minutes. However, several studies suggest that markets might need more time to digest the information. The implementation of this time-lagging effect provides more robust results of the main research question (Wagner et al., 2018; Nisar & Yeung, 2018). This first sub-research question will examine the predictive performance of the event study and the LSTM while taking into account this time-lagging effect.

Sub-research question 2:

"To what extent are the cumulative average abnormal returns in the event window statistically significant, measured by hypothesis testing?"

This second sub-research question provides a clear insight in the predictive power of the event study regression. Namely, statistically significant cumulative average abnormal returns show a substantial difference between the predicted returns and the observed returns, which provides a valuable insight regarding the research question. The effect size in this sub-research question will be answered by hypothesis testing, as described in Chapter 3: Methodology & Experimental Setup.

Sub-research question 3:

"What is the difference in performance, in terms of RMSE, of other types of recurrent neural networks such as SRN and GRU for this prediction task?"

The literature is not decisive and unambiguous in establishing the superior model between LSTM and GRU in stock price prediction tasks. Assessing several models, in addition to the main research question, provides a more comprehensive contribution to the current literature in this field. The literature does show shortcomings for the SRN, with respect to the LSTM and GRU. Still, it might be interesting to assess its abilities in terms of predictive performance. This third research question will be answered by computing the RMSE of the SRN and GRU, and comparing it with the RMSE of the LSTM and event study, as described in Chapter 3: Methodology & Experimental Setup. In addition, a comparison in terms of computational complexity and transparency will be made, since these are important considerations when deciding on a model.

1.3 Findings

The main finding is this thesis is the significant difference in performance of the LSTM compared to the event study methodology. The RMSE of the LSTM is over 13 times lower than the RMSE of the event study methodology, which shows that the LSTM is more suited for estimating normal returns. The LSTM also outperformed both the SRN and the GRU, in terms of RMSE. These results remained valid across all event window sizes. Another major finding were the statistically significant cumulative average abnormal returns of the event study methodology at a 1% level across all event window sizes.

Chapter 2: Literature Review

This section provides a comprehensive exploration and analysis of the current literature related to the research problem. The literature review aims to define the scientific context, and identify gaps and shortcomings in the existing theoretical framework.

2.1 Event Study

The event study methodology was first used to estimate the stock price reaction for stock splits (Fama et al., 1969). Surprisingly enough, not much has changed since then. Kothari and Warner stated that "the basic statistical format of event studies has not changed over time" (Kothari & Warner, 2004, p. 8). In addition, the study of Vijh et al. (2020), almost fifteen years later, concluded the same. They identified that the only notable difference is that nowadays (intra-)daily stock returns are often used rather than monthly stock returns. One could perceive this 50 year-stationarity of development as a gap in the literature. The main research question in this study aims to tackle this stagnation and provide evidence for a more suitable alternative, namely an LSTM recurrent neural network.

Over the last decades, the event study methodology has been the predominant model to evaluate the impact of an event on stock prices. The event study is a widely used technique in various fields. Kothari & Warner (2004) found that between 1974-2000 over 500 papers have been published in the top five financial journals that used an event study. All the necessary steps, computations and considerations are presented comprehensively in various studies, e.g., Kirtzman (2018) and MacKinlay (1997). Hypothesis testing establishes the predictive power of the event study regression, which the second sub-research question aims to assess.

However, the event study methodology also has its limitations. Fisch et al. (2018) and Baker (2016) stated that the event study is not always reliable. The normal returns do not reflect stock prices in a realistic manner when volatility in the market suddenly shifts. Both of these researches were based on the financial crisis period of 2007-2009. A study by Kothari & Warner (2004) acknowledges this conclusion. Moreover, they stated that variance increases induce a bias in the t statistic of the hypothesis testing, which causes the null hypothesis to be rejected too often.

The current state of literature does not contain suggestions that the underlying linear model for computing the normal returns in an event study is insufficiently flexible or lacks predictive power, as the problem statement indicates. However, model accuracy is of vital importance in this task. A recent research of event studies stated that "the accuracy of the abnormal returns assessment is highly dependent on the accuracy of the preceding expected return model" (Enginar & Atici, 2021, p.1).

2.2 Non-Linearity in Stock Price Predictions

More than 20 years ago, Abhyankar et al. (1997) found that stocks returns worldwide contain a pattern of nonlinear dependence. As a follow-up on this research, Qi (1999) applied this theory, and found that nonlinear models such as neural networks outperform linear models in stock return forecasts.

Since then, the literature about machine learning models for stock price predictions is rich in methods and practical applications. Numerous studies have pursued to approximate the optimal machine learning method for stock price predictions, e.g., Gu et al. (2020), Ma et al. (2021), Naik & Mohan (2019), Leipold et al. (2021) and Leung et al. (2021). Support vector machines and neural networks are the most commonly used machine learning techniques for predicting stock prices (Henrique et al., 2019). Controversially, a leading paper from Gu et al (2020) stated that regression trees outperform support vector machines in a stock price prediction task. More importantly, both Henrique et al. (2019) and Gu et al. (2020) agree on neural networks having the highest predictive power for asset pricing. An important limitation of these models is their tendency to overfit. Regularization is key when implementing these techniques.

In addition, Gu et al. (2020) showed that traditional regression-based methods lack the flexibility to accommodate nonlinear interactions. This, in combination with the mentioned findings of Abhyankar et al. (1997) and Qi (1999), inspired the research problem of this study and lead to the research question.

2.3 Recurrent Neural Networks

Humans do not start thinking from scratch every second. They perceive each observation based on their understanding of previous experiences. Hence, memory is important in grasping relationships. The simple recurrent network (SRN), first introduced by Elman (1991), is a deep learning model that specializes in sequential data, such as stock prices. The dynamic hidden layers allow the model to capture long distance dependencies in the sequential input data. However, the SRN is rarely used in practice today due to exploding and vanishing gradients during backpropagation when task complexity increases (Ribeiro et al., 2020). This limitation was examined by Hochreiter & Schmidhuber (1997), who introduced the long short-term memory recurrent neural network (LSTM). Although it was initially not acknowledge to its full potential, nowadays LSTM is the state of the art in the field of stock price predictions (Premanand et al., 2021; Chen & Ge, 2018). These findings played a vital role in the definition of the main research question of this thesis.

In addition, the gated recurrent unit recurrent neural network (GRU) is also a popular model for stock market forecasting. The GRU, introduced by Cho et al. (2014), is also designed to reduce the problem of exploding and vanishing gradients of the traditional RNN when learning long-term dependencies. Several studies indicate the similarity in performance between the LSTM and GRU for stock price predictions, e.g. Shahi et al. (2020), Berradi et al. (2020) and Fang et al. (2021). However, a few studies disagree. Yamak et al. (2019) found that GRU outperforms LSTM in terms of RMSE when comparing Bitcoin price predictions. In addition, a study by Li & Pan (2021) states that GRU outperforms LSTM in terms of convergence in CPU time.

The literature contains contradicting findings regarding the performance of LSTM and GRU. The third sub-research question examines the (dis)similarity in performance between these two state-of-the-art models given the research problem.

2.4 Machine Learning in Event Studies

As mentioned, a few studies have researched a topic somewhat related to this thesis. Yamashita & Miura (2019) found a machine learning model that outperforms the event study analysis for several selected Japanese stocks. They conducted an event study using a topological model with a self-organising map (SOM), also known as the Kohonen network, instead of the traditional linear regression model. This is an unsupervised learning neural network that applies dimension reduction while holding the underlying topological relation of the data, and then determining correlation between the event and the abnormal returns. This event study was focused specifically on the impact of the Government Pension Investment Fund’s holding disclosure in 2015 on the Japanese stock market. In addition, Yamashita & Miura (2019) identified a potential bias in the results of the event study mechanism when applied to an event-clustered market situation.

Next to this Japanese market event study, Dogra et al (2021) conducted research on the effect of national financial news, such as fraud and mergers, on stock prices of Indian banks between 2018 and 2021. This effect was evaluated by a transformer model instead of the classic linear regression model. Similar to a recurrent neural network, the transformer model uses an encoder-decoder architecture, temporal memory and sequential input data. The main difference between the two is that recurrent neural networks require ordered data, where transformer models do not. Unlike the study by Yamashita & Miura (2019), this Indian-based research did not compare the performance of their model with the traditional event study model. Another major difference between these two studies is that Yamashita & Miura (2019) did not consider the time-lagging effect in the event window, since their study only uses a single event window size. This is a limitation, because negligence of multiple event windows make the results unreliable (Wagner et al., 2018; Nisar & Yeung, 2018). The first sub-research question of this thesis is incorporated to control for the time-lagging effect, as described in Section 1.2: Research Questions.

All in all, the use of machine learning in event studies is sparsely present in the current literature. Moreover, these few studies have been fixated on rather specific stocks, which makes their findings deficient in generalizability. The use of machine learning methods to estimate normal returns for an event study experiment is a unique research design, as the literature only sporadically contains research in this field, especially given the focus on the European stock markets. A research that challenges the fundamental methodology used in event studies is a gap in the literature. This gap has led to the defined research problem and research goal of this thesis.

Chapter 3: Methodology & Experimental Setup

This chapter includes a description of the methodology used in this study to answer the research question, including scientific motivation and respective comparison. In addition, this chapter provides as much transparency as possible about the dataset and experimental procedure.

3.1 Methodology

This section describes and justifies the algorithms used in the research. The first sub-section includes all elements of the event study methodology, after which the SRN, LSTM, and GRU are presented. The final sub-section contains a justification for the methods used, as well as a comparison with alternative techniques.

3.1.1 Event Study

The event study computes the difference between theoretical returns, excluding the event, and observed returns, including the event. Appendix 1 provides a visual overview of the event study methodology. This model serves as the baseline in this study.

The estimation window starts 180 trading days before the event, and ends 5 trading days before the event. The chosen length of the estimation window is sufficient to make valid estimations of the normal returns (Pesaran & Pick, 2011). The 5 trading days gap between the end of the estimation window and the event date has been implemented because it prevents potential information leakage of the event to affect the estimation window (Ding et al., 2018). The event study is conducted using event windows of three different sizes, namely 1 trading day, 3 trading days and 10 trading days. This application ensures that the market has a sufficient period of time to digest the new information of the event and act upon it accordingly (Wagner et al., 2018; Nisar & Yeung, 2018).

The theoretical normal returns (NR) are predicted in the estimation window using the OLS regression Equation (1)

$$R_{i,\tau} = \alpha_i + \beta_i Rm_\tau + u_{i,\tau}, \tag{1}$$

where $R_{i,\tau}$ = excess returns of stock i at time t, and Rm_τ = excess market returns.

This linear model regresses the excess return of each firm on a constant term and the excess market index within the estimation window, where the event is still unknown to the market. The normal returns in the event window are the predictions of this linear model. These normal returns NR are the approximated returns of a given firm in a situation where the event would not have happened.

The abnormal returns ($R_{i,\tau} - NR_{i,\tau}$) of a given firm *i* show the impact of the event on the stock returns in a certain time period τ. In order to obtain the overall impact of an event on the stock returns, the abnormal returns of a firm need to be summed within the event window. Next, now that each individual firm has its own cumulative abnormal return, the overall effect needs to be quantified. The overall impact, which is known as the cumulative average abnormal return (CAAR), is calculated by computing the mean of all firm's cumulative abnormal returns, as shown in Equation (2)

$$CAAR = \frac{1}{N} \sum_{i=1}^{N} \sum_{i=1}^{\tau} R_{i,\tau} - NR_{i,\tau}, \tag{2}$$

where $NR_{i,\tau}$ = normal returns of firm i at time t.

3.1.2 SRN

The SRN is designed to model sequential input data. The states of the hidden layers are dynamic, as opposed to traditional feed-forwards neural networks, which means that the state depends on the previous version of itself. Appendix 1 provides a visual overview of the SRN architecture.

The SRN has a short-term memory for the output of the previous hidden state. The inputs of the network are therefore both the input units and the hidden units from the previous hidden state. The hidden states of the SRN are updated by Equation (3)

$$h_t = \sigma(U x_t + W h_{t-1} + b), \tag{3}$$

where h_t = hidden state at time t.

3.1.3 LSTM

In order to solve the research problem and answer the research question, the performance of the event study methodology is compared to the performance of an LSTM, in terms of RMSE. The LSTM is a specific architecture based on the traditional recurrent neural network, which is able to capture long distance dependencies in the data. The input of the LSTM model passes the gates in Equation (4), Equation (5), and Equation (6)

$$f_t = \sigma\left(U_f x_t + W_f h_{t-1} + b_f\right), \tag{4}$$

$$i_t = \sigma(U_i x_t + W_i h_{t-1} + b_i), \tag{5}$$

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

where f_t = forget gate at time t, i_t = input gate at time t, and o_t = output gate at time t.

The information passes through the three gates. These gates determine which and when information flows through the architecture. The gate activations are learnable parameters that are optimized by the

LSTM during training. The example below provides an insight in the functionality of a gate, which is also known as masking.

$$\begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix} \odot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} a \\ 0 \\ c \end{bmatrix}$$

Important to note is that the parameters in the gate vectors are often not precisely 0 or 1, but the gate activations approach these values. This is due to the implementation of the nonlinear sigmoid activation function.

The information from the input x_t, previous hidden state h_{t-1}, and previous memory cell c_{t-1} enter the LSTM architecture. The forget gate f_t determines which information from the previous state is kept for the remainder of the cell. Next, the input gate i_t selects the information that is saved in the memory cell. Lastly, the output gate o_t determines which information goes on to the next timestep. The generated output of this state, namely the memory cell c_t and the current hidden state h_t are the input for the next timestep. Note that the weight matrices and bias vectors of the gates are independent of each other, and remain constant across timesteps. Appendix 1 provides a visual overview of the LSTM architecture.

The output of the current state is the memory cell and the hidden state. The memory cell is a combination of the previous memory cell and a function of inputs and hidden states, modulated by the forget gate and input gate respectively. The hidden state is a function of the current memory cell, which is modulated by the output gate. Both computations are Equation (7) and Equation (8)

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(U_c x_t + W_c h_{t-1} + b_c), \tag{7}$$

$$h_t = o_t \odot \tanh(c_t), \tag{8}$$

where c_t = memory cell at time t and h_t = hidden state at time t.

This architecture allows the model to learn from long distance dependencies within the sequential input data. A comparison between the predictive power of the LSTM and event study provides the answer to the research question of this study.

3.1.4 GRU

Similar to the LSTM, the GRU is a method to retain short-term memory of sequential input data. The GRU is an advancement of the standard recurrent neural network, which overcomes the traditional vanishing gradient problem. Similar to the LSTM architecture, the GRU uses gate mechanisms to manage the information flow within the states, as is shown in Equation (9) and Equation (10), and hidden states to retain information across timesteps, which are presented in Equation (11) and Equation (12)

$$r_t = \sigma(U_r x_t + W_r h_{t-1} + b_r), \tag{9}$$

$$z_t = \sigma(U_z\, x_t \; + \; W_z\, h_{t-1} \; + \; b_z), \tag{10}$$

$$h'_t = \; \tanh\big(U_n\, x_t + \; b_n + \; r_t \; \odot (W_n\, h_{t-1} + \; b_n)\big), \tag{11}$$

$$h_t = \; (1 - z_t) \; \odot \; h'_t + \; z_t \; \odot \; h_{t-1}, \tag{12}$$

where r_t = reset gate at time t and z_t = update gate at time t.

The predictive performance of the GRU is compared to the predictive performance of both the LSTM and the event study methodology, in terms of RMSE. This comparison identifies the superior model given the task of stock price predictions. Appendix 1 provides a visual overview of the GRU architecture.

3.1.5 RMSE

The difference in performance between the event study methodology and the recurrent neural networks answers the research question. The model performance is measured by the RMSE. Numerous studies use RMSE as the evaluation metric, as this is a robust metric to measure model performance in stock market forecasting (Reddy, 2019; Henrique et al., 2018).

3.1.5 Motivation of the Methods

The event study methodology is the most suited and most popular method to examine the impact of a given event on stock returns (Singh et al., 2020; Kothari & Warner, 2004). Besides a few studies, as addressed in Chapter 2: Literature Review, the literature does not contain alternatives for quantifying the effect of an event on stock returns. Therefore, the event study methodology is a logical choice towards the research goal.

The literature review of this study established a clear overview of both appropriate methodologies to answer the research question, as well as inappropriate methods given this problem. Figure 2 provides a heatmap of several methods with their corresponding predictive power, computational efficiency and robustness to overfitting, given this research problem.

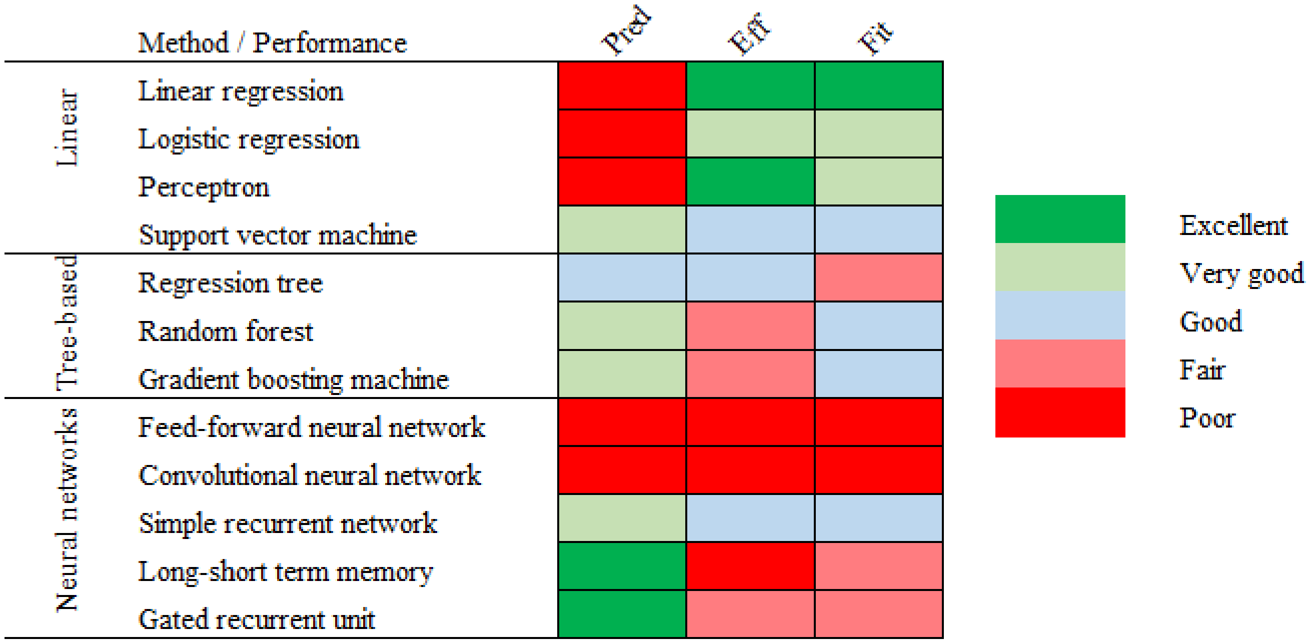

Figure 2: Heatmap of performance per method. Pred = predictive power, Eff = computational efficiency, and Fit = robustness to overfitting. Created by: F.S. Miedema. This heatmap shows the expectations of the author.

Linear models lack predictive power to forecast stock returns, since these contain a pattern of nonlinear dependence (Abhyankar et al., 1997; Qi, 1999). Despite the beneficial computational efficiency and high robustness to overfitting, the linear regression, logistic regression, and perceptron are not suited for this study.

The support vector machine, which is a linear model in nature, is able to capture nonlinear patterns using the kernel trick (Ike et al., 2019). The support vector machine and the tree-based methods are among the most popular models for stock price predictions (Henrique et al., 2019). However, as the research question aims to maximize the predictive power, the support vector machine, regression tree, random forest and gradient boosting machine are not selected for this study. Both Gu et al (2020) and Henrique et al. (2019) found that neural networks have the highest predictive power for asset pricing, given the highly complex structure of stock price data.

Neural networks can take various forms and specialized architectures. Figure 2 displays ‘Poor’ for both the feed-forward neural network and the convolutional neural network, because both of these networks are not suitable for sequential input data. Both of these models have invariance to position, which is crucial when prediction stock returns (Tashiro et al., 2019). Recurrent neural networks are specifically designed to handle sequential data. The traditional simple recurrent network however is not able to capture long distance dependencies in the data due to its exploding and vanishing gradient problem, which affects its predictive power (Ribeiro et al., 2020). This problem is reduced by the LSTM and GRU, which both have an significant predictive power given a stock return forecasting task, as can be seen in Figure 2.

Both Henrique et al. (2019) and Gu et al. (2020) agree on neural networks having the highest predictive power for asset pricing. Moreover, the LSTM is the state of the art in the field of stock price predictions (Premanand et al., 2021; Chen & Ge, 2018). These findings played a vital role in the definition of the main research question of this thesis. In addition, several studies found that the performance of the GRU is similar to the LSTM for stock price predictions (Shahi et al., 2020; Berradi et al., 2020; Fang et al., 2021). The third sub-research question identifies which of the two is superior in terms of predictive power.

3.2 Experimental Setup

This subsection aims to provide as much transparency as possible about the dataset and experimental procedure. Given this section, other researchers are granted the tools and insights to replicate the findings in this study. The Data Source/Code/Ethics Statement can be found in Appendix 2, together with the publicly available code and dataset.

3.2.1 Dataset

The raw dataset was collected on 7 March 2020 from Datastream, which is a global financial time-series database from Thomas Reuters. Datastream is directly accessible as an Excel add-in via the Tilburg University credentials of the Financial Data Support department. The daily stock returns from 1,000 randomly selected European stocks are collected between 1 January 2010 until 31 December 2019. Appendix 3 provides a detailed roadmap with instructions to replicate the exact same dataset as the one that was used in this study, as well as a list of the 1,000 randomly selected European stocks, in the original and ready-to-use Datastream format.

In addition, the Stoxx Europe 600 returns are collected from Datastream in the same period, as a proxy for the European market return. The one-month German government bond returns are selected as a proxy for the European risk-free rate, given its high liquidity and low credit risk Ejsing et al. (2015).

The variables of the raw dataset are: stock id, date, event date, stock return, market return and risk free rate. The dataset contains 2,395,710 rows, namely 10 years of trading days times 1,000 stocks. Appendix 4 provides an overview of the performed exploratory data analysis.

3.2.2 Pre-processing

Firstly, the dataset contained an error in the form of full sample missing data for roughly 5% of the companies. These 53 stocks have been dropped from the dataset. In addition, 15 stocks had insufficient data points, as their IPO was after 4 May 2019. Each stock in the sample must contain at least 190 trading days, given the described window sizes in Chapter 3: Methodology & Experimental Setup. The dataset contained 932 companies after discarding all stocks with insufficient data availability.

Second, both the stock returns and the market returns were transformed to excess returns and excess market returns respectively. Excess returns are a required pre-processing step for conducting an event study (Kirtzman 2018; MacKinlay 1997). This transformation has been performed by subtracting the corresponding risk-free rate from the stock returns and the market returns (Rasekhschaffe & Jones, 2019).

Third, the occasional missing data points in the dataset have been imputed accordingly, since neural networks are unable to handle missing data (Kia et al., 2022). This study assumes the sporadic missing values in the dataset to be caused by an absence of trade rather than human error or incomplete data entry. Therefore, the missing values are imputed with a value of 0, since the stock price did not move, for example because of differences in holidays between European stock markets. A study by Sontag et al. (2018) applied this imputation strategy in a similar prediction task using GRU to forecast time series data.

Lastly, each individual stock return was normalized by the minimum and maximum value in the training set. All elements of the data were scaled between 0 and 1 in order to set a common scale for the dataset without distorting the patterns in the data. Data normalization is an essential aspect in the process of stock price predictions, which aims to improve both the model performance and runtime (Kumari & Swarnkar, 2021). The normalization has been performed by the MinMaxScaler of Scikit-learn with a feature range from 0 to 1. Figure 3 provides a visual overview of the described pre-processing pipeline.

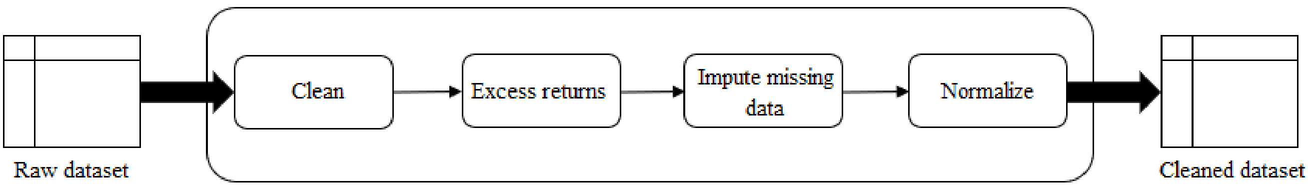

Figure 3: Pre-processing pipeline. Created by F.S. Miedema.

3.2.3 Experimental Procedure

In order to create a robust comparison between the event study methodology and the recurrent neural networks, both models are trained on an equal portion of the sample. The event study uses 185 trading days in total to quantify the effect for each stock, namely an estimation window of 175 trading days and an event window of 1, 3, and 10 trading days. Therefore, the recurrent neural networks are set up using an equivalent split of the data. The TimeSeriesSplit function of Scikit-learn ensures a valid cross-validation procedure for the sequential input data. The train size of the split is set to 175 and the gap is set to 5, similar to the event study composition. Also, the predicted output sequence is set to 1, 3 and 10 trading days respectively. Figure 4 provides a visualisation of the TimeSeriesSplit architecture. As mentioned, the train size remained fixed across splits in order to generate a robust comparison between the event study and the recurrent neural networks.

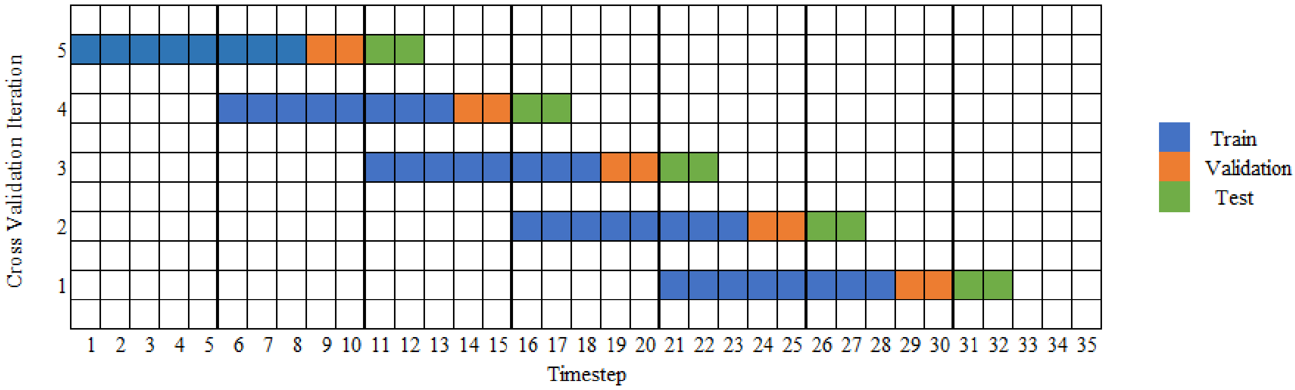

Figure 4: TimeSeriesSplit visualisation. Created by F.S. Miedema, inspired by Godoy-Rojas et al., 2022.

Several techniques have been implemented in order to strive for optimal model performance. The adaptive moment estimation (Adam) is implemented as the optimizer, as this is the state of the art in stock price prediction tasks (Kamalov et al., 2020). The mean squared error (MSE) has been selected as the loss function, as this error measurement is suitable for stock price predictions with LSTM and GRU (Hossain et al., 2018). A study by Fazard & Mashayeki (2019) compared the performance of several activation functions for recurrent neural networks, and found that the sigmoid activation function and hyperbolic tangent activation function (tanh) generate the best results. Therefore, the hyperbolic tangent activation function has been selected in the hidden layers of the recurrent neural networks. The output layer of the networks contained a linear activation function, since the model predicts stock returns. As stock returns can be either positive or negative, the linear output layer grants the freedom for the stock returns to take any value. Lastly, in order to enhance model robustness and improve generalization of the models, the inputs and hidden states are equipped with a dropout mask of 20% (Roberts et al. 2019). The architectures are presented in detail in Appendix 5.

In order to provide robust results and out-of-sample model evaluation, the research included stratification using randomized event dates, as well as hyperparameter tuning during training on a 20% validation set. The hyperparameters that were tuned are the learning rate of the Adam optimizer [0.01, 0.001, 0.0001] and the number of epochs [10, 50, 100]. The selected upper level of 100 epochs is set because of feasibility, following the research of Siami-Namini et al. (2019). The optimal number of nodes and hidden layers are 50 and 3 respectively. These parameters have been tested manually, again for feasibility reasons. The number of tuned hyperparameters is relatively limited, given the large size of the dataset. The total CPU training time of the models amounted to over 90 hours. The hyperparameter tuning has been performed on the training data and validation data only, in order to ensure that the model is tested on unseen data.

Figure 5 presents the implemented research technique in the form of a flowchart. This research followed the guidelines of the CRoss-Industry Standard Process for Data Mining (CRISP-DM), which is “the de-facto standard process model in data mining projects” (Schröer et al., 2021, p. 533).

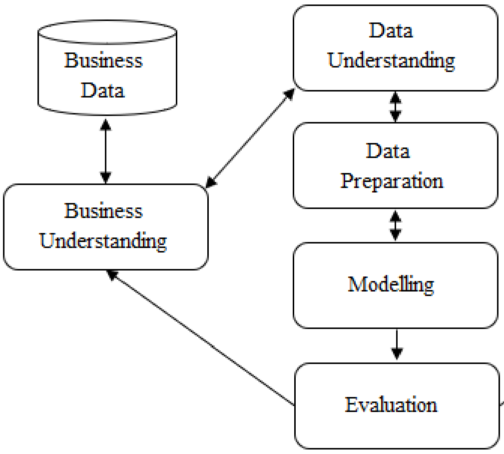

Figure 5: Methodology flowchart. Created by F.S. Miedema, inspired by Schröer et al. (2021) .

The CRISP-DM is the golden thread that runs through the empirical analysis as a whole. This research design provides a structured yet flexible approach to implementing a robust data mining study. It captures an iterative process from a thorough understanding of the research task until the final deployment of the models. Especially the exploratory data analysis in combination with the data cleaning are an important share of the work. The arrows in Figure 9 indicate the most important dependencies between phases.

Figure 6 displays an extensive diagram of the implemented methodology, which has been described in Chapter 3: Methodology & Experimental Setup.

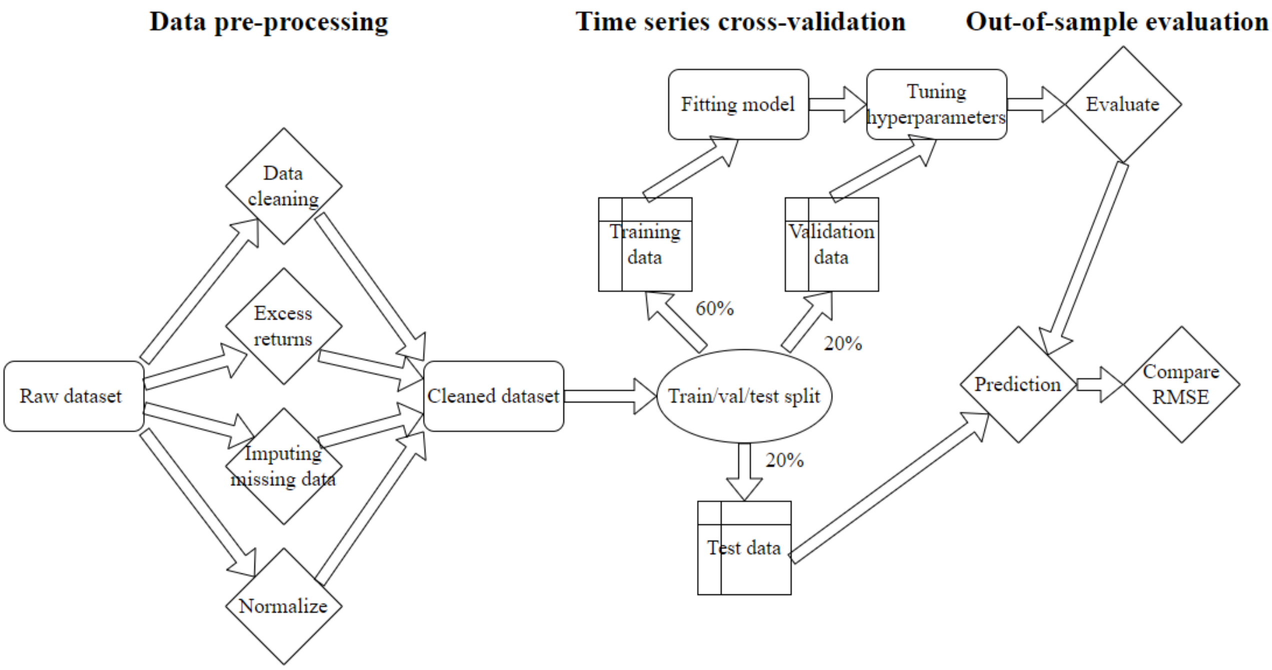

Figure 6: Methodology diagram. Created by: F.S. Miedema.

As a result of the exploratory data analysis, the raw data from Datastream is transformed accordingly using the described pipeline. After pre-processing, the data is split into a training set, validation set, and test set. The models are fitted and tuned in the training set and validation set respectively, without access to the test data. During this process, the Adam optimizer minimizes the MSE loss function. Lastly, the models perform an out-of-sample evaluation based on the unseen test data. The model with the lowest RMSE has the highest predictive performance.

The programming languages, packages and corresponding versions used in this study are disclosed in Appendix 6.

Chapter 4: Empirical Results

This section provides the results and findings of the experiments, and is structured in alignment with the research questions. The first subsection provides the error analysis of the training data, validation data, and test data, after which the results of the comprehensive comparison between the event study methodology and the LSTM for the single day event window is presented. The second subsection is devoted to the analysis of the time-lagging effect on the performance, by including several event window sizes. Next, the predictive power of the event study is defined using hypothesis testing, as described in Section 3.1.1: Event Study. The following subsection presents the assessment of predictive performance of additional recurrent neural network models, namely SRN and GRU. Lastly, an overall comparison is conducted between all models across all event window sizes, in order to provide a convenient overview on model performance in terms of RMSE. In addition, several key characteristics of the models are contrasted to provide a more comprehensive contribution to the current academic literature.

4.1 Error Analysis

This subsection provides an extensive analysis of the errors in the training data, validation data, and test data of all four models.

Figure 7 provides an overview of the observed errors in the training data, validation data, and test data in the different event window sizes.

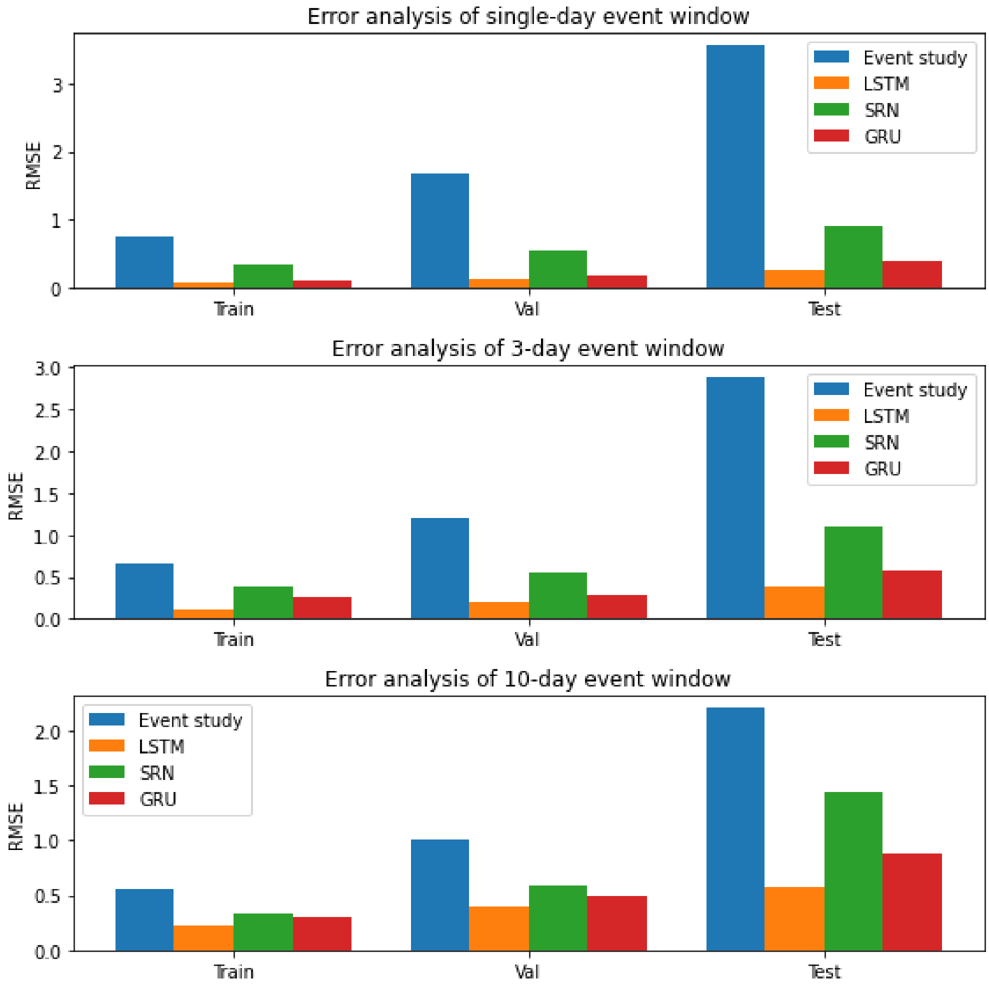

Figure 7: Error analysis in the training, validation, and test set for the event study, LSTM, SRN, and GRU across the single-day, 3-day, and 10-day event window. Created by F.S. Miedema.

The error analysis displays highest RMSE across each model in the test data, as opposed to the training data, where the RMSE is lowest. The models perform significantly worse when predicting the unseen test data. Interesting to observe is the increase in RMSE for the recurrent neural network architectures as the event window size increases, whereas the linear event study shows a decreasing trend across time.

Figure 8 shows the residual plot and distribution of the prediction errors of the models.

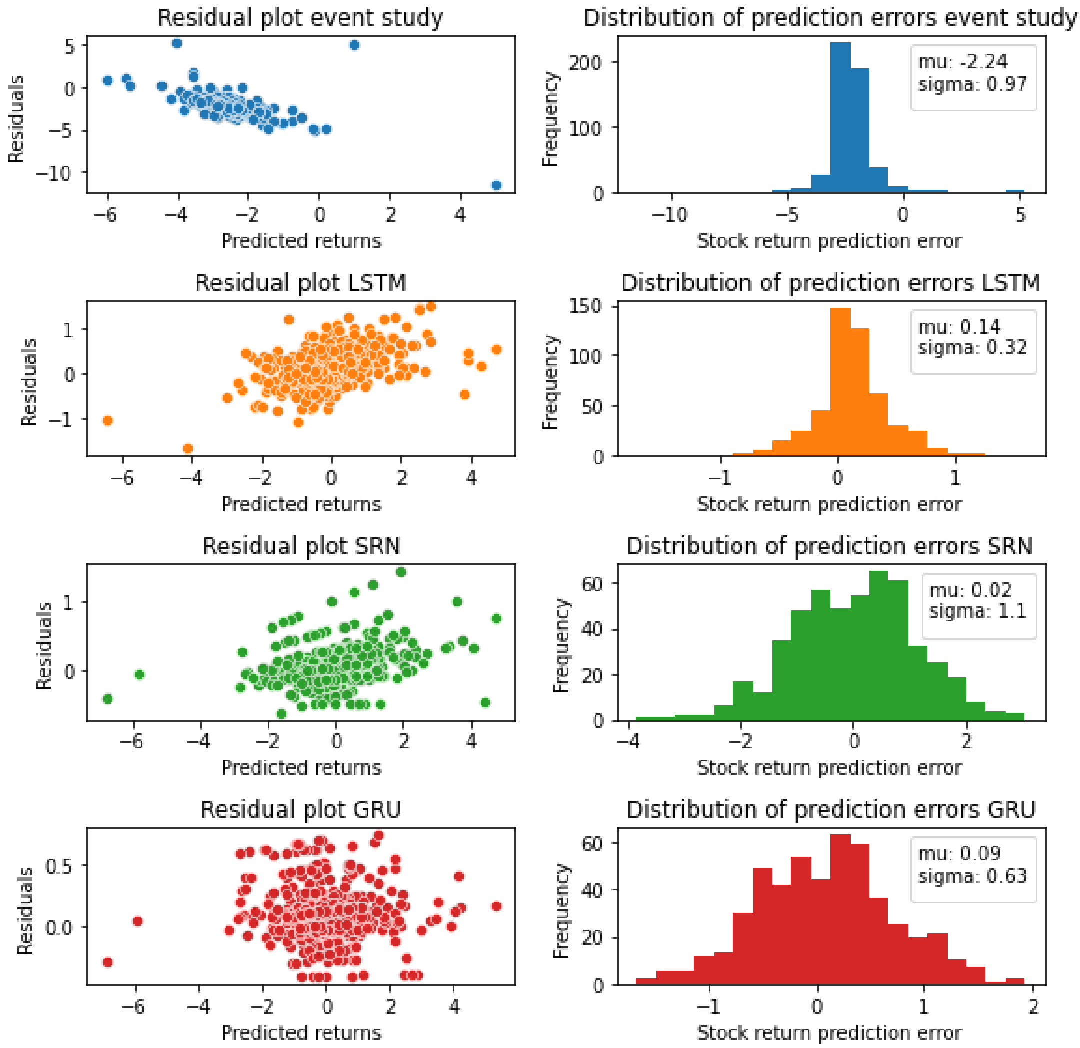

Figure 8: Prediction error analysis of the event study, LSTM, SRN, and GRU. The left hand side displays the residual plots for each model, whereas the right hand side shows the distribution of the prediction errors per model. Created by F.S. Miedema.

The residual plots of the recurrent neural networks do not show a clear trend, which demonstrates that the errors are independent and normally distributed. The event study methodology however shows a linear relationship in the residual plot, in addition to the observed convergent pattern which relates to heteroscedasticity (Li, 2014). The prediction error distributions of the recurrent neural networks centre around zero, whereas the event study shows an average prediction error of -2.24, which indicates that the predicted returns of the model are too high on average. The prediction error distribution of the LSTM has the lowest standard deviation, namely 0.32.

4.2 Main Research Question: Event Study vs LSTM

Table 1 shows the out-of-sample RMSE in the single-day event window of both the event study methodology and the LSTM, given the task of predicting stock return of European stocks between 2010 and 2020.

Table 1: Event study vs LSTM RMSE in single day event window.

Model / Performance measure	RMSE
Event study	3.5568
LSTM	0.2661

The RMSE of the event study methodology is over 13 times higher than the RMSE of the LSTM. As shown in Table 1, the 0.2661 RMSE of the LSTM outperforms the baseline 3.5568 RMSE of the event study methodology excessively.

4.3 Sub-Research Question 1: Performance Considering the Time-Lagging Effect

Table 2 presents the RMSE of the event study methodology and the LSTM in three different event window sizes, namely 1, 3, and 10.

Table 2: Event study vs LSTM RMSE in several event window sizes.

	RMSE		
Model / Event window size	Single day	3 days	10 days
Event study	3.5568	2.8882	2.2805
LSTM	0.2661	0.5783	0.7035

The LSTM outperformed the baseline event study methodology, in terms of RMSE, across all three event window sizes. The event study RMSE is around 13, 5, and 3 times larger than the LSTM RMSE, in the single-day event window, 3-day event window, and 10-day event window respectively.

Interesting to observe is the contrasting behaviour of both models across time. The RMSE of the LSTM increases as the output sequence enlarges. This relationship is due to the increasing uncertainty of the predictions as the forecasting advances per timestep, as illustrated in Figure 9. Larger output sequences include more predictions based on predicted values, which capture more uncertainty. In comparison, predicting tomorrow's weather is easier than predicting the weather a month from now. This mechanism increases the prediction errors, leading to a higher RMSE in the 10-day event window compared to the shorter output sequence of the single-day event window.

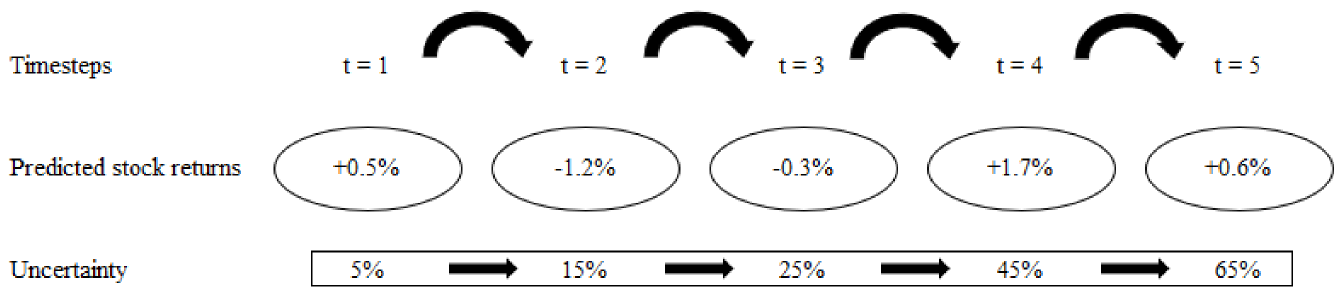

Figure 9: Prediction uncertainty increases by output length. Both the stock returns and uncertainty measures in this figure are illustrative. Created by F.S. Miedema.

Contrastingly, the RMSE of the event study decreases as output sequence size enlarges. As the underlying mechanism of the event study differs substantially from the LSTM, this behaviour might be logical. The LSTM generates the predictions as sequential timesteps, whereas the event study computes the predictions based on a linear relationship between the stock returns and the market return (within the estimation window). For example, if the beta coefficient of the event study's OLS regression is 0.5, and the market return at a given timestep in the event window is 2%, than the event study predicts a normal return of 1% at this timestep. Whether this market return is in the first day of the event window or in the tenth day of the event window is indifferent. The event study predicts stock returns completely independent of the timestep, as opposed to the LSTM which deals with prediction uncertainty across time. However, a clear explanation for the decreasing RMSE of the event study across timesteps is not found in the literature. Figure 10 provides a visual presentation of the RMSE of both models for the different event window sizes.

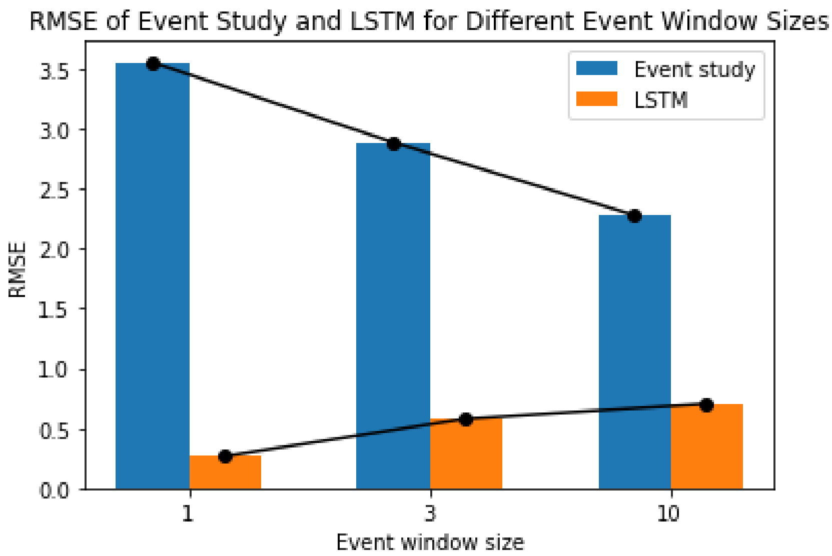

Figure 10: Visual overview event study vs LSTM in several event window size. Created by F.S. Miedema.

Figure 10 clearly presents the increasing and decreasing RMSE trend of the LSTM and the event study methodology respectively, as well as the substantial difference in predictive performance.

4.4 Sub-Research Question 2: Event Study Statistical Significance

Table 3 shows the effect size result of the hypothesis testing on the cumulative average abnormal returns of the event study in the event window with size 1, 3, and 10.

Table 3: The computed CAARs in different event window sizes with the corresponding significance levels and standard errors. The robust standard errors of the CAARs are displayed between parentheses. The level of statistical significance are shown by the number of asterisks. One asterisk (*) represents a p-value that is significant at a 10% level, two asterisks represent a significance level of 5%, and three asterisks represent a significance level of 1%.

	Event window		
	Single day	3 days	10 days
CAAR	1.5260***	2.5246***	3.8362***
	(0.1913)	(0.2280)	(0.2459)

The p-values are statistically significant at a 1% level for all three event window sizes. This indicates strong evidence against the null hypothesis, namely that the CAARs are zero. Since the threshold is set at 5%, as described in Section 3.1.1: Event Study, the null hypothesis is rejected. The statistically significant CAARs show that the predicted normal returns of the event study differ significantly from the actual observed returns. Since the event windows in this experiment are set at randomized dates, there is no event affecting the stock returns in the event window. This implicates that the generated normal returns should be equal to the actual returns.

Since the CAARs in Table 3 are statistically significant at a 1% level, the event study methodology failed to produce valid estimations of the stock returns. This is in line with the high RMSE scores of the event study methodology in Table 2 and Figure 10.

4.5 Sub-Research Question 3: Predictive Performance SRN & GRU

Table 4 provides the out-of-sample RMSE of the event study, LSTM, SRN and GRU, in the single-day, 3-day, and 10-day event window respectively.

Table 4: RMSE of the Event Study, LSTM, SRN, and GRU Across Three Different Event Window Sizes.

	RMSE		
Model / Event window size	Single day	3 days	10 days
Event study	3.5568	2.8882	2.2805
LSTM	0.2661	0.5783	0.7035
SRN	0.9096	1.4366	1.6992
GRU	0.3837	0.8725	1.0263

The LSTM outperformed the event study, SRN, and GRU across all event window sizes, in terms of predictive performance measured in RMSE, in estimating the normal returns for European stocks between 2010 and 2020. The GRU approaches the performance of the LSTM most closely, as the difference in RMSE is relatively limited. The event study methodology has the lowest predictive power compared to the other models. Figure 11 provides a visual overview of the RMSE of the models in all three event windows.

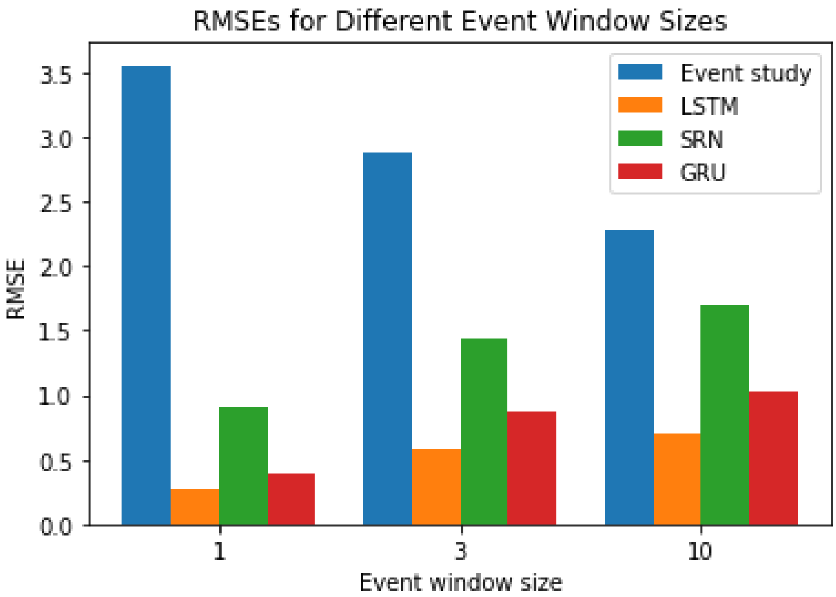

Figure 11: Visual overview of RMSE of the event study, LSTM, SRN, and GRU across three different event window sizes. Created by F.S. Miedema.

As mentioned, the event study shows a decreasing trend across timesteps, whereas the LSTM, SRN, and GRU show an increasing trend when event window size enlarges. Extrapolating this trend for larger event windows might even result in an event study RMSE that outperforms the SRN, or even the GRU and LSTM. Following this logic, one might state that the traditional event study methodology is superior when implementing larger event windows. However, larger event windows are strongly discouraged in the literature, as this leads to invalid results due to the sensitivity of standard errors (Kothari & Warner, 2004; MacKinlay, 1997).

4.6 Key Characteristics of the Models

Lastly, the research questions in this study are focussed on the predictive power of the models given the task of predicting normal returns. Yet, it might be interesting to consider additional characteristics of the models.

Computational complexity plays an important role when deciding on an experimental setup design. Figure 12 shows the RMSE and total runtime per model given the task of this study.

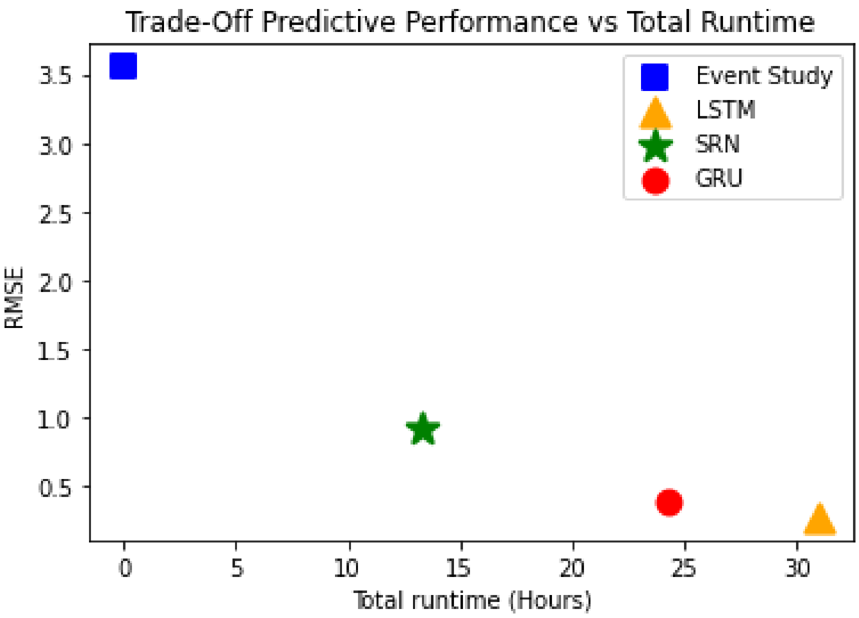

Figure 12: RMSE vs total runtime per model. Created by F.S. Miedema.

The high-performance recurrent neural networks show a relatively low RMSE. Yet, this high predictive performance comes at the cost of relatively long runtime. The LSTM, SRN, and GRU have a total runtime given this task of around 32, 13, and 24 hours respectively, whereas the event study methodology has a total runtime of only 1.5 minutes.

Chapter 5: Discussion

This section evaluates the results regarding the research questions. The goal of this study was to provide a comprehensive academic resolution for the identified gap in the literature, namely the absence of a revolutionized alternative for the traditional linear event study methodology. This study implemented a machine learning approach to improve the estimations of normal returns, which is a fundamental component of the event study methodology. This study found that the event study methodology lacks the predictive power to estimate valid stock returns, and proposes a significantly more effective model to solve the given task.

5.1 Main Research Question

"To what extent does an LSTM outperform the predictive power of the event study methodology, in terms of RMSE, when estimating the normal returns of European stocks between 2010 and 2020?"

The main research question aimed to assess whether the LSTM outperforms the linear event study approach in the task of estimating normal returns. This research question was selected, because Gu et al. (2020) found that linear regression models lack the flexibility to accommodate nonlinear interactions, which are a fundamental property of stock price data (Abhyankar et al., 1997; Qi, 1999). This thesis found that the LSTM significantly outperformed the event study, which was the baseline, with an out-of-sample RMSE of 0.2661 and 3.5568 respectively. These results are congruent with the findings of Premanand et al. (2020) and Chen & Ge (2018), who stated that the LSTM is the state of the art in the field of stock price predictions. This application is implemented in a crucial component of the event study methodology, namely the estimation of normal returns. This approach is sparsely present in the current literature, especially regarding European stock markets.

5.2 Sub-Research Question 1

"How does the performance, in terms of RMSE, of the event study methodology and LSTM change when implementing several multi-day event windows?"

The first sub-research question aimed to identify the difference in performance between the LSTM and event study when considering several event window sizes. This approach controls for the time-lagging effect, and increases the robustness of the results (Wagner et al., 2018; Nisar & Yeung, 2018). This thesis found that the LSTM outperformed the event study across all three event windows. The LSTM showed an RMSE of 0.2661, 0.5783, and 0.7035, and the event study showed an RMSE of 3.5568, 2.8882, and 2.2805, for the single-day, 3-day, and 10-day event window respectively. As mentioned, the difference in performance was as expected. However, the RMSE of the LSTM reveals an increasing trend across timesteps, whereas the event window shows a decreasing trend in RMSE as the event window size enlarges. This pattern contradicts prior research of Li et al. (2018, p. 2), who state that "the LSTM enforces a constant error flow over timesteps". The decreasing RMSE of the event study across timestep is plausible, in the sense that the OLS regression approach varies fundamentally from the LSTM methodology. Yet, despite extensive exploration of the current literature, no convincing evidence was found that supported the observed RMSE trend of the event study.

5.3 Sub-Research Question 2

"To what extent are the cumulative average abnormal returns in the event window statistically significant, measured by hypothesis testing?"

The second sub-research question evaluated the predictive performance of the event study methodology. The hypothesis testing showed that the CAARs are statistically significant at a 1% level, for all three event windows. This finding is in line with the research of Gu et al. (2020), who found that linear regression models are not suited for predicting stock returns. On the other hand, this finding is quite surprising, because the event study methodology is widely used among the top journals in the financial field (Kritzman, 2018; Kothari & Warner, 2004). The results indicate that the approach of the event study methodology is not suitable for its purpose. This new insight might lead practitioners to rethink their model of choice when evaluating event impact on stock prices.

5.4 Sub-Research Question 3

"What is the difference in performance, in terms of RMSE, of other types of recurrent neural networks such as SRN and GRU for this prediction task?"

The third sub-research question aimed to identify a superior model for this problem, as the literature is not decisive in this area. As expected, the LSTM outperforms the SRN significantly, as the latter suffers from exploding and vanishing gradients during backpropagation for complex tasks, such as predicting stock prices (Ribeiro et al., 2020). The LSTM also slightly outperforms the GRU, which is somewhat surprising given the current literature. Several studies emphasize the similarity in predictive performance when forecasting stock returns, e.g. Shahi et al. (2020), Berradi et al. (2020) and Fang et al. (2021). A study by Yamak et al. (2019) even states that the GRU outperforms the LSTM in terms of RMSE. On the other hand, the differences in terms of computational efficiency between the LSTM, SRN, and GRU were as suggested in the literature (Li & Pan, 2021). The SRN performed best with approximately 13 hours of total runtime, second was the GRU with 24 hours, and worst was the LSTM with nearly 32 hours of total runtime.

In addition to computational efficiency, the interpretability and transparency of models is of increasing importance. Understanding the mechanisms behind the scenes are fundamental in order to establish trust of society. Figure 13 shows an overview of the trade-off between model interpretability and model performance.

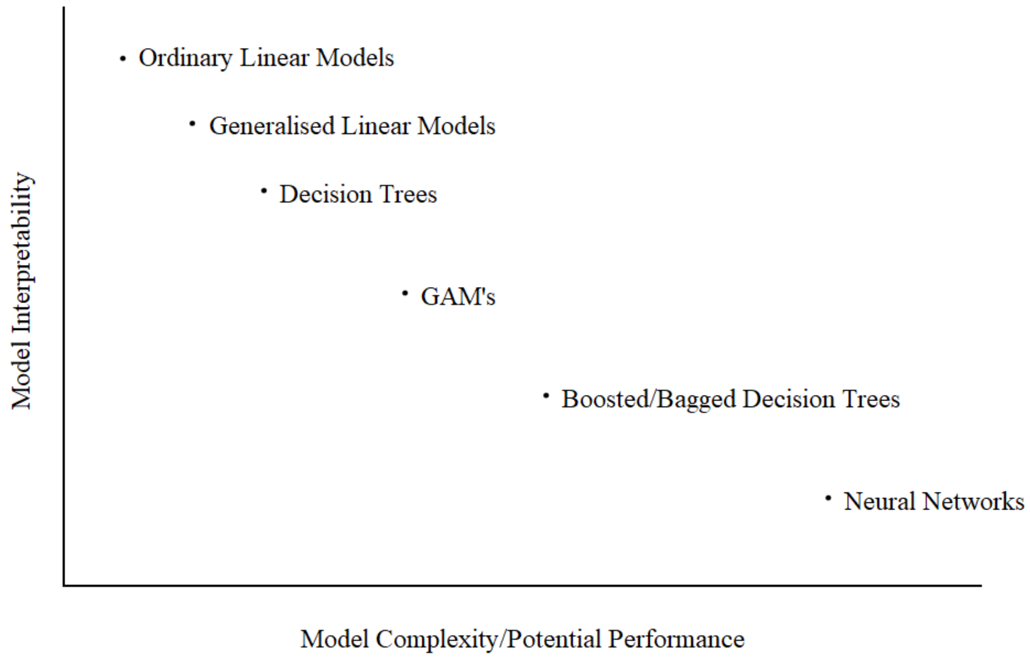

Figure 13: Model Interpretability vs Performance. Created by F.S. Miedema, inspired by Lazaridis (2021).

The linear model of the event study methodology shows high interpretability, because regression coefficients and significance tests provide direct insight in feature importance. Yet, the linear model lacks the flexibility to capture complex relationships within data structures. On the contrary, recurrent neural networks are the state of the art within the field of time series forecasting performance. The transparency of neural networks however remains limited, and debatably problematic. The input data is subjected to stacked, non-linear transformations inside the many hidden layers of the network, which make the mechanisms within the model architecture difficult to grasp. Transparency is fundamental to mitigate biases or other adverse factors within networks, especially in automated decision-making from an economic, medical and legal point of view (Meyes et al. 2020).

Chapter 6: Conclusion

This chapter provides a brief summary of the key findings of this thesis, in relation to both the research goal and the scientific contribution, as well as the limitations and potential areas for future research.

6.1 Summary

This thesis aimed to improve the predictive performance for estimating normal returns of the traditional event study methodology using several recurrent neural network architectures. The research question and sub-research questions examined a design to fill the gap in the literature by using a state-of-the-art LSTM model. The predictive performance, in terms of RMSE, of the LSTM was significantly higher compared to the event study methodology, which fulfils the research goal. On the other hand, the linear event study excelled in terms of computational efficiency and transparency, which are also important considerations when deciding on a model. Still, when one is mainly interested in predictive power for stock return forecasting, the LSTM is the superior model, also compared to the SRN and the GRU.

The event study methodology has not been revised in over 50 years. This thesis provides a progressive contribution to the scientific field and society by proposing a modernized approach that significantly outperforms one of the most influential financial models of all time. The current literature is missing alternatives for the traditional event study methodology, which is shown to lack predictive power to estimate valid normal returns. The novel contribution to the existing literature is the significant improvement of a crucial component of the event study methodology, namely the estimation of normal returns. This thesis provides both the proof and the tools for practitioners to estimate more accurate stock price reactions than ever before, for example a trader that wants to quantify the impact of the war in Ukraine on European car manufacturers.

6.2 Limitations and Recommendations

Although this research provides valuable insights in the assessment of event impact on stock prices, it should be emphasized that the number of tuned hyperparameters is relatively limited. The large size of the dataset in combination with the present hardware resources formed a constraint in terms of computational complexity.

Another limitation, although this is in the nature of the event study methodology, is the use of univariate stock data. It is generally known that certain stock characteristics, such as book-to-market ratio and market capitalization, provide a significant contribution in predicting the price of a stock. It would be interesting to perform a follow up study with a similar methodology on multivariate stock data, and compare the results.

In addition, revising one of the many event study based researches might lead to new insights, when implementing the presented methodology in this thesis. This would be interesting to demonstrate. On the other hand, it is important to recognize that implementing an LSTM model requires more statistical expertise than conducting a traditional linear event study.

Lastly, the research goal has been accomplished with a significant performance of the LSTM model. However, the weakness of this model, in comparison with the event study methodology, is its high computational complexity and limited transparency, as is shown in Figure 14 and Figure 15 respectively. Future research in this area might reveal practical applications to mitigate these disadvantages, which is of increasing importance.

References

Abhyankar, A., Copeland, L.S., & Wong, W. (1997). Uncovering Nonlinear Structure in Real-Time Stock-Market Indexes: The S&P 500, the DAX, the Nikkei 225, and the FTSE-100. *Journal of Business & Economic Statistics*, 15(1), 1-14. doi: 10.2307/1392068

Baker, A.C., (2016). Single-Firm Event Studies, Securities Fraud, and Financial Crisis: Problems of Inference. *Stanford Law Review*, 68(5), 1207-1266. Retrieved from: https://review.law.stanford.edu/wp-content/uploads/sites/3/2016/06/68_Baker_-_Stan._L._Rev._1207_Appx.pdf

Bash, A., & Alsaifi, K. (2019). Fear for uncertainty: An Event Study of Khashoggi and Stock Market Returns, *Journal of Behavioural and Experimental Finance*, 23, 54-58. doi: 10.1016/j.jbef.2019.05.004

Berradi, Z., Lazaar, M., Omara, H., & Mahboub, O. (2020). Effect of Architecture in Recurrent Neural Network Applied on the Prediction of Stock Price, *International Journal of Computer Science*, 47(3), 11-17. Retrieved from: http://www.iaeng.org/IJCS/issues_v47/issue_3/IJCS_47_3_11.pdf

Chen, S., & Ge, L. (2018). Exploring the Attention Mechanism in LSTM-Based Hong Kong Stock Price Movement Prediction, *Journal of Quantitative Finance*, 19(9), 1507-1515. doi: 10.1080/14697688.2019.1622287

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder-Decoder for Statistical Machine Translation, *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, 1724-1734. doi: 10.48550/arXiv.1406.1078

Ding, L., Lam, H.K.S., Cheng, T.C.E., & Zhou, H. (2018). A Review of Short-Term Event Studies in Operations and Supply Chain Management, *Journal of Production Economics*, 200, 329-342. doi: 10.1016/j.ijpe.2018.04.006

Dogra, V., Singh, A., Verma, S., Alharbi, A., & Alosaimi, W. (2021). Event Study: Advanced Machine Learning and Statistical Technique for Analyzing Sustainability in Banking Stocks, *Journal of Mathematics*, 9(24), 3319. doi: 10.3390/math9243319

Ejsing, J., Grothe, M., & Grothe, O. (2015). Liquidity and Credit Premia in the Yields of Highly-Rated Sovereign Bonds, *Journal of Empirical Finance*, 33, 160-173. doi: 10.1016/j.jempfin.2015.04.001

Elman, J.L. (1991). Distributed Representations, Simple Recurrent Networks, and Grammatical Structure, *Journal of Machine Learning*, 7(2), 195-225. doi: 10.1007/BF00114844

Enginar, O., & Atici, K.B. (2021). Optimal Forecast Error as an Unbiased Estimator of Abnormal Return: A Proposition, *Journal of Forecasting*, 41(1), 158-166. doi: 10.1002/for.2798

Fama, E.F. (1970). Efficient Capital Markets: A Review of Theory and Empirical Work. *Journal of Finance*, 25(2), 383-417. doi: 10.2307/2325486

Fama, E.F., Fisher, L., Jensen, M.C. & Roll, R. (1969). The Adjustment of Stock Prices to New Information, *International Economic Review,* 10(1), 1-21. doi: 10.2307/2525569

Fang, W., Chen, Y., Xue, Q. (2021). Survey on Research of RNN-Based Spatio-Temporal Sequence Prediction Algorithms, *Journal of Big Data,* 3(3), 97-110. doi: 10.32604/jbd.2021.016993

Fazard, A., & Mashayekhi, H. (2017). A Comparative Performance Analysis of Different Activation Functions in LSTM Networks for Classification, *Journal of Neural Computing and Applications*, 31, 2507-2521. doi: 10.1007/s00521-017-3210-6

Fisch, J. E., Gelbach, J. B., Klick, J. (2018). The Logic and Limits of Event Studies in Securities Fraud Litigation, *Texas Law Review,* 96(3), 553-621. Retrieved from: https://www.law.upenn.edu/live/files/8942-96texlrev553pdf

Godoy-Rojas, D.F., Leon-Medina, J.X., Rueda, B., Vargas, W., Romero, J., Pedraza, C., Pozo, F., & Tibaduiza, D.A. (2022). Attention-Based Deep Recurrent Neural Network to Forecast the Temperature Behavior of an Electric Arc Furnace Side-Wall, *Sensors*, 22(4), 1418. doi: doi.org/10.3390/s22041418

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical Asset Pricing via Machine Learning, *The Review of Financial Studies,* 33(5), 2223-2273. doi: 10.1093/rfs/hhaa009

Henrique, B.M., Sobreiro, V.A., & Kimura, H. (2018). Stock Price Prediction Using Support Vector Regression on Daily and Up to the Minute Prices, *Journal of Finance and Data Science*, 4(3), 183-201. doi: 10.1016/j.jfds.2018.04.003

Henrique, B.M., Sobreiro, V.A., & Kimura, H. (2019). Literature review: Machine Learning Techniques Applied to Financial Market Prediction, *Expert Systems with Application*, 124, 226-251. doi: 10.1016/J.ESWA.2019.01.012

Hochreiter, S. & Schmidhuber, J. (1997). Long Short-Term Memory, *Journal of Neural Computations*, 9(8), 1735-1780. doi: 10.1162/neco.1997.9.8.1735

Hossain, M.A., Karim, R., Thulasiram, R., Bruce, N.D.B., & Wang, Y. (2018). Hybrid Deep Learning Model for Stock Price Prediction, *2018 IEEE Symposion Series on Computational Intelligence*. doi: 10.1109/SSCI.2018.8628641

Kamalov, F., Smail, L., & Gurrib, I. (2020). Stock Price Forecast with Deep Learning, *2020 International Conference on Decision Aid Sciences and Application (DASA),* 1098-1102. doi: 10.1109/DASA51403.2020.9317260

Kia, S.M., Rad, N.M., Van Opstal, D., Van Schie, B., Cahn, W., Marquand, A., Pluim, J.P.W., & Schnack, H.G. (2022). Pruning Missing Values in Neural Networks, *International Conference on Learning Representations 2022*. Retrieved from: https://openreview.net/pdf?id=M_o5E088xO5

Kothari, S.P. & Warner, J.B. (2004). Econometrics in Event Studies, *Handbook of Corporate Finance: Empirical Corporate Finance*, 1, 1-51. doi: 10.2139/ssrn.608601

Kritzman, M.P. (2018). What Practitioners Need to Know About Event Studies, *Financial Analysts Journal*, 50(6), 17-20. doi: 10.2469/faj.v50.n6.17

Kumari, B., & Swarnkar, T. (2021). Stock Movement Prediction Using Hybrid Normalization Technique and Artificial Neural Network, *International Journal of Advanced Technology and Engineering Exploration,* 83(8), 1336-1350. doi: 10.19101/IJATEE.2021.874387

Lazaridis, D. (2021). Explainable AI and Interpretable Machine Learning Models, *Ambiata: Scientific Return-on-Data*. Retrieved from: https://ambiata.com/blog/2021-04-12-xai-part-1/

Leipold, M., Wang, Q., & Zhou, W. (2021). Machine Learning in the Chinese Stock Market, *To appear in Journal of Financial Economics.* doi: 10.1016/j.jfineco.2021.08.017

Leung, E., Lohre, H., Mischlich, D., Shea, Y., & Stroh, M. (2021). The Promises and Pitfalls of Machine Learning for Predicting Stock Returns, *Journal of Financial Data Science,* 3(2), 21-50. doi: 10.2139/ssrn.3546725

Li, K. (2014). Cognitive Models for Developing Students' Graphical Understanding of Residual Plots, *The 3rd International Conference on Education and Management Innovation,* 38-42. doi: 10.7763/IPEDR.2014.V70.8

Li, S., Li, W., Cook, C., & Gao, Y. (2018). Independently Recurrent Neural Networks: Building a Longer and Deeper RNN, *Journal of Computer Science*, 2018(1), 31-48. doi: 10.48550/arXiv.1803.04831

Li, W., Wu. H., Zhu, N., Jiang, Y., Tan, J., & Guo, Y. (2021). Prediction of Dissolved Oxygen in a Fishery Pond Based on Gated Recurrent Unit (GRU), *Information Processing in Agriculture,* 8(1), 185-193. doi: 10.1016/j.inpa.2020.02.002

Li, Y., & Pan, Y. (2021). A Novel Ensemble Deep Learning Model for Stock Prediction Based on Stock Prices and News, *International Journal of Data Science and Analytics*, 13, 139-149. doi: 10.1007/s41060-021-00279-9

Ma, Y., Han, R., & Wang, W. (2021). Portfolio Optimization with Return Prediction Using Deep Learning and Machine Learning, *Expert Systems with Applications*, 165(3). doi: 10.1016/j.eswa.2020.113973

MacKinlay, A.C. (1997). Event Studies in Economics and Finance, *Journal of Economic Literature*, 35(1), 13-39. Retrieved from: http://www.business.unr.edu/faculty/liuc/files/BADM742/MacKinlay_1997.pdf

Meyes. R., De Puiseau, C.W., Posada-Moreno, A., & Meisen, T. (2020). Under the Hood of Neural Networks: Characterizing Learned Representations by Functional Neuron Populations and Network Ablations, *Journal of Computer Science*, 2020(1), 12-54.

Naik, N., & Mohan, B.R. (2019). Study of Stock Return Predictions Using Recurrent Neural Networks with LSTM, *International Conference on Engineering Applications of Neural Networks*, 453-459. doi: 10.1007/978-3-030-20257-6_39

Nisar, T.M., & Yeung, M. (2018). Twitter as a Tool for Forecasting Stock Market Movements: a Short-Window Event Study, *The Journal of Finance and Data Science*, 4(2), 101-119. doi: 10.1016/j.jfds.2017.11.002

Ike, J.C., Anireh, V.I.E., & Bennett, E.O. (2019). Prediction of Foreign Exchange (Forex) Market Using Support Vector Machines, *Journal of Advances in Mathematical & Computational Sciences*, 7(2), 19-32. doi: 10.22624/AIMS/MATHS/V7N2P1

Pesaran, M.H., & Pick, A. (2011). Forecast Combination Across Estimation Windows, *Journal of Business & Economic Statistics*, 29(2), 307-318. doi: 10.1198/jbes.2010.09018

Premanand, G., Raghav, S., Ishaan, K., Param, M., & Rahul, C. (2021). Multi-Day Window for Stock Price Prediction and Financial News Classification for Predicting Market Sentiments, *International Journal of Next-Generation Computing*, 12(5), 534-543. doi: 10.47164/ijngc.v12i5.428

Qi, M. (1999). Nonlinear Predictability of Stock Returns Using Financial and Economic Variables, *Journal of Business & Economic Statistics*, 17(4), 419-429. doi: 10.2307/1392399

Ren, J., Karagoz, K., Gatza, M.L., Singer, E., Sadimin, E., Foran, D., & Qi, X. (2018). Recurrence Analysis on Prostate Cancer Patients With Gleason Score 7 Using Integrated Histopathology Whole-Slide Images and Genomic Data Through Deep Neural Networks, *Journal of Medical Imaging*, 5(4), 47-50. doi: 10.1117/1.JMI.5.4.047501

Rasekhschaffe, K.C., & Jones, R.C. (2019). Machine Learning for Stock Selection, *Financial Analysts Journal*, 75(3), 70-88. doi: 10.1080/0015198X.2019.1596678

Reddy, C.V. (2019). Predicting the Stock Market Index using Stochastic Time Series ARIMA Modelling: The Sample of BSE and NSE, *Indian Journal of Finance*, 13(8), 7-25. doi: 10.17010/ijf%2F2019%2Fv13i8%2F146301

Ribeiro, A.H., Tiels, K., Aguirre, L.A., & Schöne, T. (2020). Beyond Exploding and Vanishing Gradients: Analysing RNN Training using Attractors and Smoothness, *International Conference on Artificial Intelligence and Statistics*, 108, 2370-2380. doi: 10.48550/arXiv.1906.08482

Roberts, S., Zohren, S., & Lim, B. (2019). Enhancing Time-Series Momentum Strategies Using Deep Neural Networks, *Journal of Financial Data*, 1(4), 19-38. doi: 10.3905/jfds.2019.1.015

Schröer, C., Krusse, F., & Gómez, J.M. (2021). A Systematic Literature Review on Applying CRISP-DM Process Model, *Procedia Computer Science*, 181, 526-534. doi: 10.1016/j.procs.2021.01.199

Shahi, T.B., Shrestha, A., Neupane, A., & Guo, W. (2020). Stock Price Forecasting with Deep Learning: A Comparative Study, *Journal of Mathematics*, 8(9), 1441. doi: 10.3390/math8091441

Siami-Namini, S., Tavakoli, N., & Namin, A.S. (2019). A Comparison of ARIMA and LSTM in Forecasting Time Series, *IEEE International Conference on Machine Learning and Applications*, 17, 1394-1401. doi: 10.1109/ICMLA.2018.00227

Singh, B., Dhall, R., Narang, S., & Rawat, S. (2020). The Outbreak of COVID-19 and Stock Market Responses: An Event Study and Panel Data Analysis for G-20 Countries, *Global Business Review*. doi: 10.1177/0972150920957274

Sontag, D., Che, Z., Purushotham, S., Cho, K., & Liu, Y. (2018). Recurrent Neural Networks for Multivariate Time Series with Missing Values, *Scientific Reports,* 8(1), 1-11. doi: 10.1038/s41598-018-24271-9

Tashiro, D., Matsushima, H., Izumi, K., & Sakaji, H. (2019). Encoding of High-Frequency Order Information and Prediction of Short-Term Stock Price by Deep Learning, *Journal of Quantitative Finance*, 19(9), 1499-1506. doi: 10.1080/14697688.2019.1622314

Vijh, M., Chandola, D., Tikkiwal, V.A. & Kumar, A. (2020). Stock Closing Price Prediction using Machine Learning Techniques, *Procedia Computer Science,* 167, 599-606. doi: 10.1016/j.procs.2020.03.326

Wagner, A., Zeckhauser, R. & Ziegler, A. (2018). Company Stock Price Reactions to the 2016 Election Shock: Trump, Taxes, and Trade. *Journal of Financial Economics*, 130(2), 428-451. doi: 10.1016/j.jfineco.2018.06.013

Yamak. P.T., Yujian, L., & Gadosey, P.K. (2019). A Comparison between ARIMA, LSTM, and GRU for Time Series Forecasting, *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, 49-55. doi: 10.1145/3377713.3377722

Yamashita, T., & Miura, R. (2019). Improving Regression-based Event Study Analysis Using a Topological Machine-learning Method, *To appear in Journal of Quantitative Finance*. doi: 10.48550/arXiv.1905.06536

Appendix 1

This appendix provides a visual overview of the model architecture that are used in this study.

(1) Event study

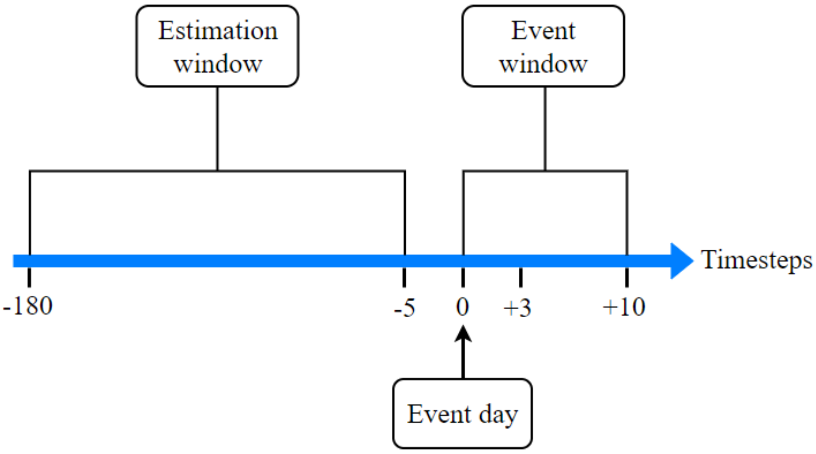

Figure 14: Event study diagram. Created by F.S. Miedema.

(2) SRN

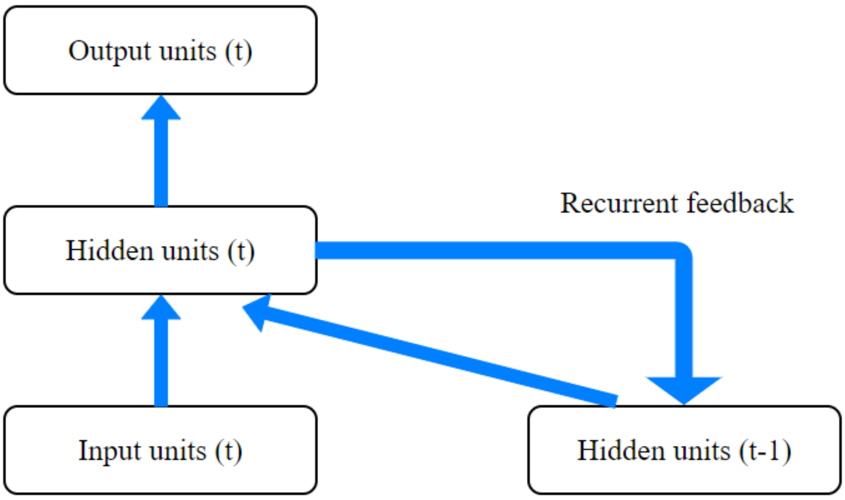

Figure 15: SRN diagram. Created by F.S. Miedema, inspired by Elman (1991).

(3) LST

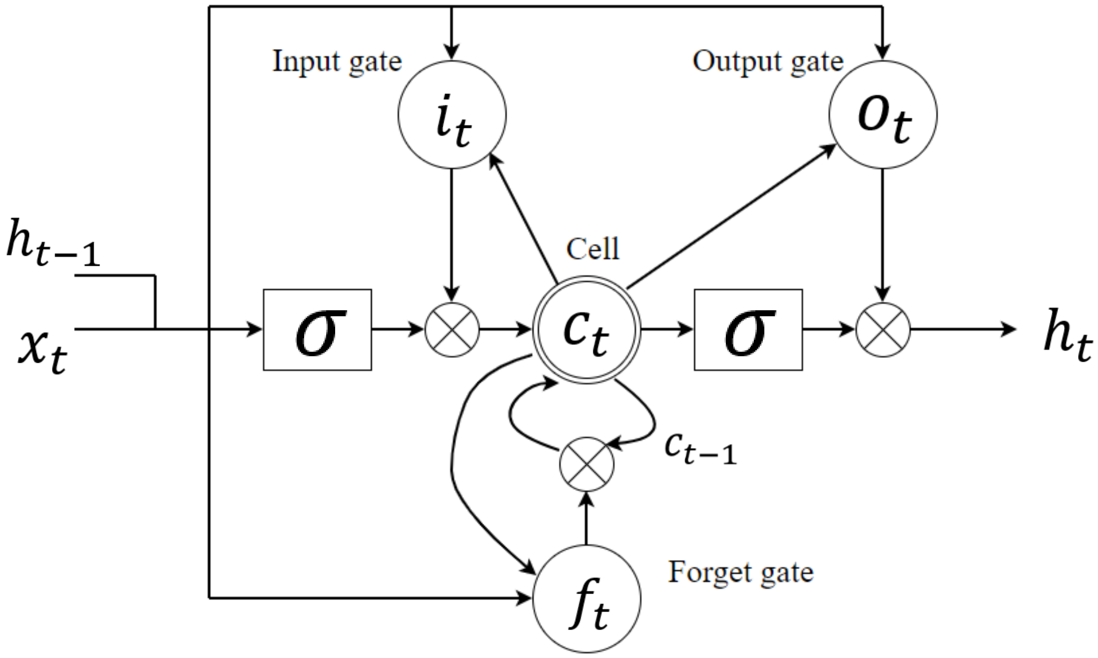

Figure 16: LSTM diagram. x_t *is the input vector,* c_t *is the memory cell, and* c_{t-1} *denotes the previous memory cell.* h_t *is the hidden state, and* h_{t-1} *is the previous hidden state.* f_t *is the forget gate,* i_t *is the input gate, and* o_t *is the output gate. Lastly,* σ *denotes the sigmoid activation function, and* $\times$ *is a pointwise multiplication. Created by F.S. Miedema, inspired by [Ren et al., 2018](#).*

(4) GRU

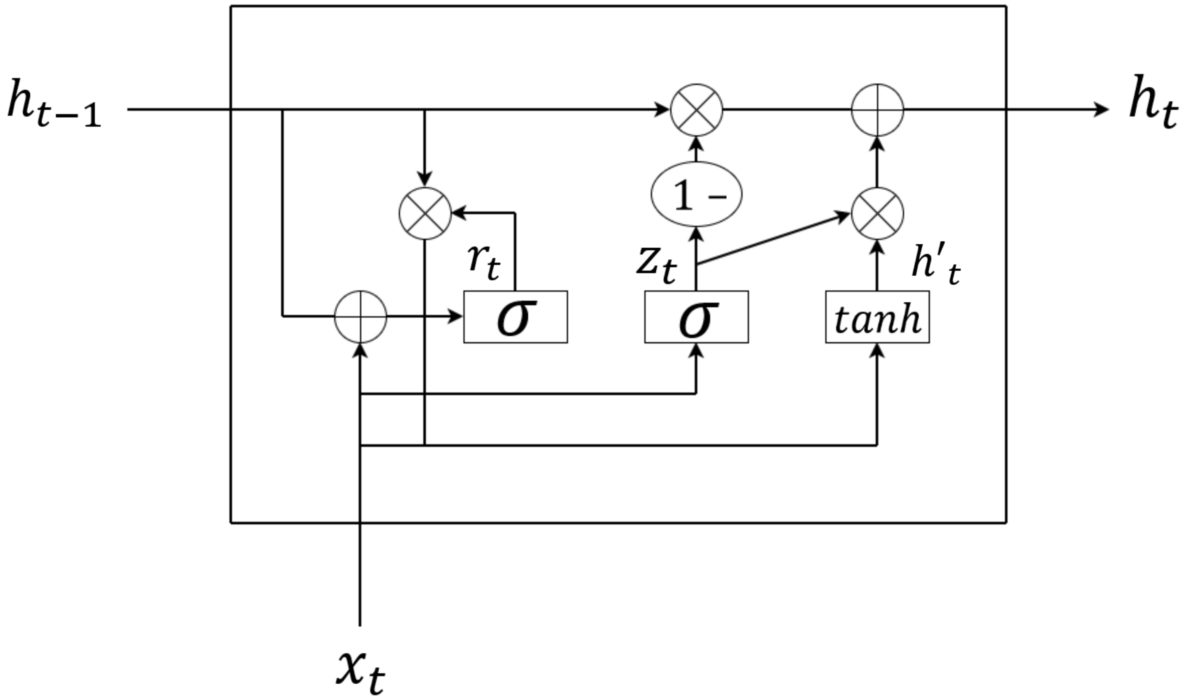

Figure 17: GRU diagram. x_t *is the input vector,* z_t *is the update gate, and* r_t *is the reset gate.* h_{t-1} *denotes the previous hidden state,* h'_t *is the candidate hidden state, and* h_t *denotes the current hidden state.* σ *and* $\tanh$ *are the activation functions, namely sigmoid and hyperbolic tangent respectively. Lastly,* × *is a pointwise multiplication, and + is a pointwise addition. Created by F.S. Miedema, inspired by [Li et al., 2021](#).*

Appendix 2

This appendix provides the Data Source/Code/Ethics Statement.

The dataset has been obtained from Datastream, which is a comprehensive financial historical database of Thomson Reuters. Tommy Klein Breteler, who is a representative of the Financial Data Support department of Tilburg University, provided the licence under which the data for this thesis has been collected. In addition, no data from human participants or animals have been included in this research.

The project code has not been obtained from an external source, and is the original work of the author. The code and dataset are publicly available in the GitHub repository of the author, under the name 'MSc Data Science RNNs.py' and 'MSc Data Science Event Study.do', for the recurrent neural networks and event study respectively. The script for the visualisations can be found in the same public repository as 'MSc Data Science Visualisations.py'.

Lastly, Figure 3, Figure 4, Figure 5, Figure 8, Figure 9, and Figure 15 were inspired by the work of Elman (1991), Ren et al. (2018), Li et al. (2021), Godoy-Rojas et al. (2022), Schröer et al. (2021), and Lazaridis (2021) respectively. The design of these figures was based on the insights of these studies, which are referenced in the corresponding figure captions as well.

Appendix 3

This appendix provides a guideline of the steps needed to replicate the dataset that was used in this study, as well as the Datastream code to obtain an exact copy of the stock returns.

First, select ‘Total Returns’ as the measure. The description of the measure is as follows: “The total return incorporates the price change and any relevant dividends for the specified period. Compounded daily return for the specified period is used to calculate Total Return and it’s effectively the dividend reinvested Total Return methodology. The most recently completed trading day is set as the default period. The Dividend type used is the most widely reported Dividend for a market and it is either Gross or Net.”

Second, set the date range from 2010-01-01 until 2019-12-31.

Third, in the ‘Layout’ section, select ‘Instrument’ for the columns, and select ‘Date’ for the rows. Set the sorting order to ‘Date (ASC)’.

Next, set the following filters in order to assure that the dataset only contains European stock: (1) Stock Exchange: London Stock Exchange, Frankfurt Stock Exchange, Milan Stock Exchange, Euronext Paris, Euronext Brussels, Euronext Amsterdam & Euronext Lisbon, and (2) Type of Equity: Only Ordinary Shares & Active Shares Only.

Lastly, select randomized stocks and set the amount to 1,000 in order to shrink the total of 34,164 stocks to 1,000 stocks. This prevents the dataset from becoming excessively large. Note that the data has to be imported in batches of 100, because Datastream returns a Timeout Error when attempting to download all the data at once.

Below is the code that is used in Excel to collect the data. This code, after installing the Datastream add-in, returns the exact dataset. In addition, this code provides a complete overview of the stocks that have been included in this study.

```
=@TR("TTEF.PA;ASML.AS;SHEL.L;LVMH.PA;BNPP.PA;BP.L;INGA.AS;AIR.PA;GLEN.L;AZN
.L;SASY.PA;MT.AS;AXAF.PA;AAL.L;SOGN.PA;HSBA.L;ABI.BR;RIO.L;OREP.PA;SCHN.PA;L
LOY.L;SGEF.PA;ADYEN.AS;BATS.L;AIRP.PA;DGE.L;PRX.AS;PRTP.PA;ULVR.L;BAES.L;EN
GIE.PA;BARC.L;CAGR.PA;SAF"&".PA;VOD.L;HRMS.PA;TCFP.PA;GSK.L;AD.AS;SGOB.PA;E
SLX.PA;DANO.PA;ASMI.AS;PRU.L;STM.PA;ORAN.PA;FERG.L;PHG.AS;RR.L;PERP.PA;RKT.
L;NG.L;CAPP.PA;REL.L;SSE.L;MICP.PA;RENA.PA;VIE.PA;DAST.PA;DSMN.AS;ICAG.L;KBC.
BR;HEIN.AS;STAN.L;TKWY.AS;NWG.L;AKZO.AS;NN.AS;WLSNc"&".AS;AV.L;IHG.L;TSCO.
L;ABNd.AS;AEGN.AS;ALSO.PA;TEPRF.PA;AHT.L;CPG.L;EXPN.L;FLTRF.L;LEGD.PA;CARR.
PA;WPP.L;LGEN.L;EDP.LS;PUBP.PA;AVST.L;WLN.PA;LSEG.L;BOUY.PA;CRH.L;GALP.LS;K
PN.AS;EZJ.L;EUFI.PA;VLOF.PA;IMB.L;MNDI.L;BT.L;BESI.AS;UCB.BR;UMG.AS;III.L;ATOS.P
```

"&"A;VIV.PA;EDF.PA;CCH.L;ENT.L;SN.L;FOUG.PA;EPED.PA;UMI.BR;SGRO.L;POLYP.L;CL
G.L;BVI.PA;EDPR.LS;SMT.L;AUTOA.L;BRBY.L;ORP.PA;WTB.L;MGGT.L;BNZL.L;INF.L;CRD
A.L;HWDN.L;LOIM.PA;RAND.AS;SGE.L;ACCP.PA;ABF.L;PSN.L;IMCD.AS;RTO.L;ANTO.L;E
DEN.PA;CRDI.MI;KGF.L;UBIP."&"PA;ISP.MI;BBOXT.L;MNZS.L;AKE.PA;LIGHT.AS;SOLB.B
R;SJP.L;EXHO.PA;NXT.L;WIZZ.L;VLLP.PA;AGES.BR;PSON.L;SVT.L;BDEV.L;ASRNL.AS;PH
NX.L;TW.L;AIRF.PA;UU.L;TATE.L;JMAT.L;ITM.L;CNA.L;BEFB.BR;GFCP.PA;HLMA.L;ENI.M
I;HIK.L;SBRY.L;RXL.PA;ITRK.L;S30.PA;GBLB.BR;RMG.L;"&"JD.L;DCC.L;BIOX.PA;AM.PA;P
ROX.BR;BCP.LS;SXS.L;LAND.L;ARGX.BR;ADML.L;ABDN.L;LMPL.L;SCOR.PA;DRX.L;MRO
N.L;ALFEN.AS;BMEB.L;SMDS.L;STDM.PA;RMV.L;CCL.L;SPX.L;DARK.L;MNG.L;EVRE.L;C
ASP.PA;OXIG.L;OCDO.L;HRGV.L;FRES.L;STLA.MI;YCA.L;SNNS.L;JMT.LS;AMUN.PA;CNPP.
P"&"A;PTNL.AS;IMI.L;BLND.L;BKGH.L;TPK.L;DPH.L;SMIN.L;ADP.PA;GLPG.AS;GETP.PA;A
ALB.AS;CWR.L;WEIR.L;IPN.PA;SOIT.PA;PNN.L;AVV.L;VOPA.AS;TE.PA;SFOR.L;AMG.AS;J
DEP.AS;CVO.PA;SDR.L;BEZG.L;GFTU_u.L;HBR.L;MKS.L;ENEI.MI;RCOP.PA;IGG.L;EUA.L;E
RMT.PA;ITV.L;ENX.PA;EMG."&"L;ICP.L;SEBF.PA;DLGD.L;APAM.AS;CNE.L;CEY.L;HAYS.L;
FUTR.L;VLS.PA;JET2.L;GTT.PA;ELI.BR;GEPH.PA;INCH.L;FDJ.PA;DLN.L;FXPO.L;ECM.L;EU
AV.BR;HEIO.AS;INPST.AS;INVP.L;OCI.AS;BOOH.L;GAW.L;UTG.L;COLR.BR;ASOS.L;HSX.L;
TLW.L;SMWH.L;MNKS.L;COFB.BR;NEOEN.PA;EURA.PA;"&"KORI.PA;PHP.L;BRWM.L;ULE.
L;PRSMB.L;RUBF.PA;SQZ.L;ROR.L;BWY.L;THG.L;SCTS.L;FLOW.AS;SAFE.L;SSON.L;CLINC
.L;AKA.PA;NEXS.PA;WISEa.L;SBMO.AS;KNOS.L;PANR.L;UKWG.L;INPP.L;WOSG.L;LDOF.
MI;WDPP.BR;VMUK.L;ABIO.PA;ABCA.L;FEVR.L;HSV.L;BAB.L;ELIS.PA;SOF.BR;LTEN.PA;"
&"TRIG.L;GRG.L;QQ.L;AGRP.L;INDV.L;COFA.PA;FAN.L;HICL.L;IETB.BR;PLUSP.L;KBCA.B
R;BYG.L;FUGR.AS;ETL.PA;DOCS.L;CSPC.L;ALSS.LS;PTEC.L;BAMN.AS;GASI.MI;CTEC.L;F
NAC.PA;QLT.L;AOO.BR;ACKB.BR;BVIC.L;GBGP.L;DIOR.PA;ICAD.PA;MWDP.PA;MCPHY.P
A;AAF.L;PETSP.L;OSBO.L;AML"&".L;ARDS.AS;MCRO.L;CORB.AS;RDW.L;PCT.L;CAPCC.L;
ACCG.AS;VGP1.BR;DPLM.L;BFIT.AS;SRP.L;ONT.L;NEXI.PA;BOLL.PA;JTC.L;GRI.L;HOCM.L
;PAGPA.L;SOPR.PA;SPIE.PA;GROW.L;CBRO.L;YSO.LS;GNS.L;NEX.L;LXIL.L;RACE.MI;NVG
R.LS;CHG.L;JUP.L;BALF.L;RCH.L;TRST.L;CTY.L;CCC.L;CI"&"NE.L;EPGT.L;RCP.L;RWS.L;G
KP.L;LAGA.PA;ROO.L;3IN.L;GENG.L;BOOMA.L;LRE.L;ALESK.PA;ISOS.PA;VVO.L;TRMR.L;
BPOST.BR;TRIA.PA;VTYV.L;MLXS.BR;LOTB.BR;SYNTS.L;DNLM.L;ALLFG.AS;BAMI.MI;V
RLA.PA;SRET.L;CVSG.L;BEKB.BR;RENE.LS;TNET.BR;PLOF.PA;ASHM.L;BREE.L;SVS.L;VC
P"&".L;IWG.L;ATST.L;HUDN.AS;BPTB.L;KWS.L;ATT.L;FGT.L;GCC.L;VOF.L;SSPG.L;TENR.
MI;EBOX.L;MERY.PA;HMSO.L;UKCM.L;CWK.L;HVPEa.L;FRAS.L;WG.L;WEHA.AS;CNHI.MI
;DEC.L;PAGE.L;DBG.PA;SHED.L;LTGL.L;TWKNc.AS;HGT.L;VCTX.L;LIO.L;SUPR.L;BOSN.A
S;SRG.MI;ELIOR.PA;RSW.L;MONY"&".L;PHAR.AS;IMTP.PA;IPO.L;MONC.MI;TRN.MI;WWH.
L;PFC.L;CAML.L;BGEO.L;SONG.L;GNC.L;ASCL.L;FCIT.L;AFX.L;RAT.L;USAB.L;IEM.L;CUR
Y.L;BAR.BR;ARB.L;SHUR.BR;JAM.L;TEM.L;MDBI.MI;BURF.L;INTER.AS;CDL.L;JCDX.PA;W
HRW.L;OVH.PA;TFFP.PA;PNL.L;BSIF.L;NOS.LS;BCPT.L;GGPL."&"L;XFAB.PA;ALDA.PA;BIC

P.PA;SPT.L;TOM2.AS;MGNS.L;ENOG.L;AFEN.L;CGT.L;NEXII.MI;HYVE.L;RECT.BR;FDM.L;
TLIT.MI;JDW.L;POLR.L;SAGA.L;PFG.L;KIST.L;AMSU.L;WTAN.L;SHB.L;VIRB.PA;BMNB.L;C
PRI.MI;BYIT.L;REDD.L;ONTEX.BR;IPX.L;DOM.L;VLAN.AS;NESF.L;CTPNV.AS;MSLH.L;FBK
"&".MI;PRY.MI;AVON.L;MDCM.L;BBH.L;GPEG.L;EMII.MI;GAMA.L;EDIN.L;ECMPA.AS;FCS
S.L;888.L;CMCOM.AS;SIRE.L;SEQI.L;LWDB.L;CPI.L;JMG.L;PST.MI;ROTH.PA;CHRY.L;CGEO
.L;MRCM.L;BBGIB.L;ENQ.L;SYNG.L;ASL.L;BNKR.L;WKP.L;TRY.L;KIPO.BR;BRW.L;IBST.L;
TMPL.L;MAUP.PA;OXB.L;NF"&"C.L;KAPE.L;ALLDL.PA;BERI.L;SGC.L;BHMG.L;THRG.L;KE
TL.L;CFEB.BR;YOU.L;CRST.L;GVOLT.LS;EWI.L;CTT.LS;EMISG.L;BOY.L;RHIM.L;FDPF.L;M
TO.L;XPP.L;MYI.L;IPAR.PA;FAGRO.BR;FORT.L;EUCAR.PA;ATG.L;FASTN.AS;AMPF.MI;PRE
M.L;FFARM.AS;MMTP.PA;AJBA.L;NETW.L;CORA.LS;PANI."&"L;ORDN.AS;CSH.L;BENB.L;
VEILV.L;GCPI.L;MDM.PA;MADE.L;STEMS.L;HRI.L;RTN.L;VLTSA.PA;MOTA.LS;HILS.L;SEP
L.L;VLX.L;OTB.L;JLEN.L;BGN.MI;CHBE.PA;ATYM.L;AWE.L;VCTP.PA;PDL.L;MAB.L;MONT
E.BR;AZMT.MI;ALCYB.PA;PIRC.MI;HEIJ.AS;MRCH.L;TRNT.L;I3E.L;SEIT.L;JSG.L;JRS.L"&";
CATGR.PA;TCAPI.L;PSH.AS;DIAS.MI;FEV.L;PHLL.L;ABS.PA;MUT.L;CLDN.L;BIFF.L;SPI.L;S
DRY.L;EOTE.L;SLPL.L;FSV.L;WINEW.L;M2Z.L;FGP.L;BGS.L;MMIT.L;AVQ.PA;FSFL.L;IMAF
.PA;BZU.MI;ATL.MI;RETE.BR;BRSC.L;MRLM.L;HSL.L;SOLG.L;HUR.L;PHI.L;JGGI.L;ALGAU
.PA;BGFD.L;HTWS."&"L;CKN.L;IHPI.L;A2.MI;DAMA.PA;CTH.L;DOTD.L;AGT.L;WIX.L;NCC
G.L;EXOR.MI;ALCRB.PA;RWI.L;HFEL.L;TESB.BR;OCIO.L;CGL.L;RSTP.L;BMED.MI;WJG.L;T
FIF.L;TED.L;TM17.L;IVG.MI;MOONM.L;HLAN.AS;HFD.L;ERM.L;JFJ.L;JUSTJ.L;TIFS.L;GIMV
.BR;HTG.L;PAFR.L;SAIN.L;UNPI.MI;DLAR."&"L;IDEA.L;EBUS.AS;HOMEH.L;POG.L;PFD.L;B
PCR.L;CMCX.L;MITRA.BR;GOG.L;XIOR.BR;IQE.L;CRIP.PA;JEMI.L;AZE.BR;SPMI.MI;SDY.L;
PZC.L;ELM.L;AO.L;ARGAN.PA;PCFT.L;SMSS.L;LUCEL.L;MARS.L;MONT.L;TCH.PA;SMCP.P
A;SDP.L;THRLT.L;ESP.L;N91.L;ALAMG.PA;ASLI.L;TUB.BR;CLSH.L;V"&"TU.L;BRGE.L;RGL
R.L;TEP.L;ESNT.L;CARM.PA;INWT.MI;TET.L;KCT.L;VSVS.L;SYNCS.L;JSE.L;BGEU.L;ALNO
V.PA;LECS.PA;MGAMM.L;ECONB.BR;MCKS.L;APTD.L;APAX.L;AGLE.L;CBLP.PA;BSGR.AS
;FDEV.L;APF.L;ROBF.PA;EUNZ.L;IDLA.PA;LSAA.L;BRUN.AS;CARDC.L;TBCG.L;QDT.PA;SE
SL.PA;ALA"&"PH.L;RNWH.L;MTRO.L;CHARC.L;SOHO.L;KIE.L;IOG.L;TYMN.L;DGI9.L;SLIG
R.AS;MANP.PA;CRN.L;PEUG.PA;NRRT.L;IBAB.BR;LNA.PA;ITPG.MI;ERGO.L;ALLA.L;ICGT.
L;ORIT.L;NSTEc.AS;CORD.L;METEX.PA;DORE.L;ALCLS.PA;FOUR.L;JARA.L;NBPE.L;GFRD.
L;RECI.MI;PRIF.BR;BCG.L;CRW.L;COA."&"L;SAVES.L;DECB.BR;ABDP.L;MIDW.L;IPH.PA;
MACF.L;JCGI.L;ACSO.L;ALBIO.PA;ALERS.PA;ANTIN.PA;SRPG.PA;FEML.L;AUSC.L;JOUL.L
;SIFB.BR;JCQ.PA;VETO.PA;HLCL.L;JLP.L;SLPE.L;PRSR.L;NCYF.L;GSF.L;KOF.PA;ERG.MI;VI
LM.PA;BGUK.L;BOWL.L;BIOGW.L;FDEL.PA;BOKU.L;CPINV.BR;AIR"&"A.L;ANIM.MI;AVT
X.AS;CRE.L;HFG.L;SCIN.L;DDDD.L;JEDT.L;DSCV.L;ODET.PA;BGSC.L;GRID.L;HOTC.L;GNF
T.PA;WAVE.PA;IG.MI;AEWU.L;ALMDT.PA;AIE.L;MTCM.MI;AVCT.L;AAIF.L;ALCAR.PA;FA
LG.PA;TRCS.L;AFM.L;SBRE.L;ARAMI.PA;BREI.L;SHAN.L;US.MI;UEM.L;JCH.L;RCN.L;SWO
R.PA;SNR"&".L;LDG.L;SHI.L;RSER.L;RMM.L;PODP.L;BRFI.L;HRA.MI;LDSV.PA;CNIC.L;NAI

T.L;KPCK.L;VTC.L;PHAI.PA;ROOFA.L;ALVMG.PA;SIGHT.PA;PCTN.L;AGL.MI;KGHK.L;JHD.
L;ADVT.L;SRC.L;CREI.L;MAJ.AS;VIC.L;CGDM.PA;DIG.L;ALQGC.PA;EVSB.BR;EXMR.BR;JA
GIJ.L;BUT.L;CDAF.PA;JII.L;NAVYA"&".PA;HSW.L;BITI.PA;GREENY.BR;LOOK.L;BGBN.PA;
EGL.L;JUSC.L;ALREW.PA;TUNE.L;SLI.L;SRS.MI;THHG.PA;BGCG.L;CPH2.L;IOMG.L;KLR.L;
JMI.L;SOM.L;ATRAS.L;ESYS.L;CBOX.L;BVC.L;ALFOC.PA;CYNL.L;BOND.PA;MTECM.L;LW
I.L;ESCT.L;ASCC.BR;ARCMA.L;AUBT.PA;QTX.L;PAYP.L;TTG.L;VN"&"Hq.L;GOEG.PA;FEET
.L;WIN.L;BLV.PA;PSDL.L;OPEN.PA;VVY.AS;ZPHR.L;NXFIL.AS;DIVI.L;CAPD.L;UJO.L;HON
Y.L;AKW.PA;HZM.L;LOCAL.PA;NACON.PA;DBV.PA;LOK.L;JOG.L;GLTN.PA;LSL.L;JDG.L;
MTU.L;AXI.L;EPICE.L;ADVIC.PA;HEIT.L;ALLIX.PA;GTCN.PA;HDIV.L;TEKT.L;CSN.L;MAT
D.L;BA"&"G.L;IHR.L;ARBN.AS;GYM.L;SREI.L;AGFB.BR;NTAS.L;IISP.PA;BIDS.L;GLEG.L;Y
NGa.L;RESI.BR;REVB.L;ALBPS.PA;MCLSM.L;HE1H.L;EGID.PA";"TR.TotalReturn";"Frq=D
SDate=2000-01-03 EDate=2019-12-31 CH=date RH=IN SORTA=date CONVERTCODE=NO";A1)

Appendix 4

This appendix shows an overview of the exploratory data analysis.

Common EDA visualisations in the form of boxplots are not as informative because they are difficult to interpret given the large size of the dataset. Therefore, this appendix provides the descriptive statistics of the stock returns, market return, and risk free rate in the dataset, between 2010-2020. Note that the stock returns are expressed in percentage points, so for example 2.5% is represented in the data as the float 2.5, rather than 0.025.

	Stock returns		Market returns		Risk free rate
Count	2,051,232	Count	2,395,710	Count	2,395,710
Mean	0.05834	Mean	0.0022	Mean	0.0002
Standard deviation	2.3045	Standard deviation	0.9936	Standard deviation	0.0021
Minimum	-97.7484	Minimum	-7.0315	Minimum	-0.0019
25^{th} percentile	-0.7798	25^{th} percentile	-0.4483	25^{th} percentile	-0.0016
Median	0.0011	Median	0.0383	Median	0.0002
75^{th} percentile	0.8404	75^{th} percentile	0.5336	75^{th} percentile	0.0013
Maximum	485.7159	Maximum	7.1487	Maximum	0.0064

Descriptive Statistics of the three numeric variables in the dataset, namely (1) Stock returns, (2) Market return, and (3) Risk free rate.

The stock returns show a lower count compared to the market return and risk free rate, because the missing data imputations from the pre-processing pipeline had not been performed at this stage yet. The stock returns contain 344,478 missing data points in total, which is roughly 10% of the total dataset. The handling of missing values is described in Chapter 3: Methodology & Experimental Setup.

Both the stock returns and the market return show a mean and median around 0%. This is surprising, because visually inspecting any given European stock price chart shows an overall increasing trend between 2010 and 2020. The mean and median were expected to be somewhat higher. The mean and median of the risk free rate are as expected. The standard deviations of all three look plausible as well, just like the minimum and maximum values for all three returns.

Appendix 5

This appendix presents a summary of the model architecture of the (1) SRN, (2) LSTM, and (3) GRU.

(1) SRN

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 simple_rnn (SimpleRNN)      (None, 1, 50)             2600

 dropout_54 (Dropout)        (None, 1, 50)             0

 simple_rnn_1 (SimpleRNN)    (None, 1, 50)             5050

 dropout_55 (Dropout)        (None, 1, 50)             0

 simple_rnn_2 (SimpleRNN)    (None, 50)                5050

 dropout_56 (Dropout)        (None, 50)                0

 dense_18 (Dense)            (None, 1)                 51

=================================================================
Total params: 12,751
Trainable params: 12,751
Non-trainable params: 0
_________________________________________________________________
```

(2) LSTM

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm_27 (LSTM)              (None, 1, 50)             10400

 dropout_27 (Dropout)        (None, 1, 50)             0

 lstm_28 (LSTM)              (None, 1, 50)             20200

 dropout_28 (Dropout)        (None, 1, 50)             0

 lstm_29 (LSTM)              (None, 50)                20200

 dropout_29 (Dropout)        (None, 50)                0

 dense_9 (Dense)             (None, 1)                 51

=================================================================
Total params: 50,851
Trainable params: 50,851
Non-trainable params: 0
_________________________________________________________________
```

(3) GRU

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 gru (GRU)                   (None, 1, 50)             7950

 dropout_81 (Dropout)        (None, 1, 50)             0

 gru_1 (GRU)                 (None, 1, 50)             15300

 dropout_82 (Dropout)        (None, 1, 50)             0

 gru_2 (GRU)                 (None, 50)                15300

 dropout_83 (Dropout)        (None, 50)                0

 dense_27 (Dense)            (None, 1)                 51

=================================================================
Total params: 38,601
Trainable params: 38,601
Non-trainable params: 0
_________________________________________________________________
```

Appendix 6

This appendix provides an overview of the software used in the research.

Software	Sub-module	Package	Version	Functionality
pandas			1.4.1	Data manipulation
numpy			1.20.3	Data manipulation
keras	models	Sequential	2.7.0	Model initialization
keras	layers	Dense	2.7.0	Linear activation
keras	layers	LSTM	2.7.0	LSTM layers
keras	layers	GRU	2.7.0	GRU layers
keras	layers	SimpleRNN	2.7.0	SRN layers
sklearn	preprocessing	MinMaxScaler	1.0.2	Data normalization
sklearn	model_selection	TimeSeriesSplit	1.0.2	Cross-validation
sklearn	model_selection	RandomizedSearchCV	1.0.2	Hyperparameter tuning
sklearn	metrics	mean_squared_error	1.0.2	Performance measure
math		sqrt	3.10.4	Performance measure
datetime		now	4.4.0	Runtime measure
matplotlib	pyplot		3.5.1	Visualization

The training has been performed using Google Colaboratory. This platform enables one to run several sessions simultaneously, thus reducing runtime. During the implementation of this research, the dataset was divided into batches of 100 stocks at a time.

The recurrent neural networks are implemented using Python (version 3.9.7) in a Spyder environment (version 5.1.5) through Anaconda (version 4.12.0), and the event study is conducted using Stata (version 17). The overview above displays Python packages, as Stata does not require the importation of additional packages.