Zürcher Hochschule für Angewandte Wissenschaften
School of Management and Law

MSc Banking and Finance

Master Thesis

The Impact of Fake News on Financial Markets: Deep Learning and Stock Market Reaction

submitted by
Visar Dauti

submitted to
Dr. Bledar Fazlija

Winterthur, 05.12.2022

Management Summary

Intentionally false or misleading information or commonly described as “fake news” (FN), has received a great deal of attention in recent years and can potentially affect political or social domains. The rapid expansion of social media platforms has further driven the dissemination of misleading information across all channels. Associated with the U.S. 2016 presidential election the prevalence of FN articles may have been decisive for the election.

Despite the general popularity of FN as a research focus in recent years, the amount of available detailed studies on how FN specifically affects financial securities is limited.The tenets of the efficient market hypothesis states that financial securities reflect all publicly available information and that false or misleading information should not persist in efficient financial markets. The primary objective of this thesis is to measure the price response of FN articles on financial securities and to examine whether an efficient market, based on the principles of Fama, already incorporating false or misleading news in the security price.

First, financial news is retrieved from one of the largest crowd-sourced platforms. Then, the financial texts are classified into legitimate and fake articles using state-of-the-art natural language processing and Deep Learning. The algorithm is trained on a unique dataset covering different types of news. After fine-tuning the algorithm, it is applied to Motley Fool news articles. To measure the impact of FN articles on the corresponding company, an event study is conducted using the Five-Factor Model by Fama.

In conclusion, the results indicate the release of FN articles is associated with negatives cumulative abnormal returns for small-cap firms. The data do not reveal in detail whether the downward trend in the market is an act of pump-and-dump scheme to manipulate the stock price or a signal of underlying poor corporate performance. The influence of FN on the price of large companies remains absent. There is a considerable return response for mid-size firms. The cumulative abnormal returns reaches a high of 7.9% after 20 days. As a result, the market is not perfectly efficient and fake information can influence the price of securities.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

AR	Abnormal Return
BoW	Bag of Words
CAR	Cumulative Abnormal Return
CBoW	Continuous Bag of Words
DL	Deep Learning
FN	Fake News
ML	Machine Learning
NLP	Natural Language Processing
OLS	Ordinary Least Squares
SEC	Securities and Exchange Commission

1. Introduction and problem statement

False or misleading information has received a great deal of attention in recent years and can potentially affect social, political, or economic relationships (Niessner et al., 2018, pp. 1–2). The rapid proliferation of Facebook, Twitter, and other social media platforms paved the way for the dissemination of information across all media channels without any significant third-party filtering or monitoring (Allcott & Gentzkow, 2017, p. 211). Although disinformation, now commonly described as “fake news” (FN), has long existed (Burkhardt, 2017), it has recently gained more prominence as the new digital age has enabled its diffusion. Commonly linked to major events such as the 2016 U.S. presidential election or Brazil’s 2018 presidential election, FN has had a dominant influence and might have been pivotal in these races (Allcott & Gentzkow, 2017, p. 232; Tomaz & Tomaz, 2020, pp. 500–509).
Despite the general popularity of FN as a research focus in recent years, the amount of available detailed studies on how FN specifically affects financial securities is limited. Chen et al. (2014b) and Cookson and Niessner (2019) provided initial insights by analyzing the extent to which investor opinions or stock market news disseminated via investment-related social media sites affect financial securities. They demonstrated that news shared on these financial social media sites could influence the volume of trading, the price of shares, and earnings surprises (Chen et al., 2014a, pp. 1398–1400; Cookson & Niessner, 2019, pp. 34–35).

However, the first evidence of the impact of FN on financial markets in particular comes from studies by Clarke et al. (2019) and Niessner et al. (2018), who indicated that the presence of FN could distort stock trading volumes and returns and affect both the social welfare of financial social media platforms and the perceived fairness of market competition. The FN used in these two studies came from a Securities and Exchange Commission (SEC) investigation and the whistleblower Richard Pearson. In 2014, Pearson worked for Seeking Alpha, which is one of the largest and best-known crowd-sourced websites for financial news (Niessner et al., 2018, p. 2). He was compensated by a PR firm to write bullish articles about public companies (Niessner et al., 2018, p. 2) under the guise of impartiality and independence (Clarke et al., 2019, p. 1). Although disclosure of such a compensation arrangement is required by Section 17(b) of the Securities Act of 1933 and the terms of Seeking Alpha, the author did not disclose the payment. Instead of declining the offer from the PR firm, he went undercover and started to investigate how rampant this practice was on these crowd-sourced platforms. He ultimately turned over to the SEC 171 articles about 47 companies written by 20 authors who had intentionally published false information (Niessner et al., 2018, p. 2). The SEC initiated an investigation, which was made public in March 2014 and eventually led to legal action against the PR firms and the authors of the articles (Niessner et al., 2018, p. 9).

There is no consensus in the literature as to whether misinformation has an significant effect on the stock market. On the one hand, Niessner et al. (2018, pp. 41–42) argue that FN can have a temporary impact on trading volumes and prices, especially for small caps. In contrast, Clarke et al. (2019, pp. 24–25) find that fake publications cannot significantly affect stock prices and contend that the market appears to discount fake news articles.

According to the tenets of the efficient market hypothesis (Fama, 1970), FN should not be capable of influencing the price of securities because it conveys misinformation that would be rejected by rationality in a perfectly efficient market regardless of the equilibrium model for asset pricing (Fong, 2021, pp. 2–5). This implies that observed effects of FN should not persist in financial markets, as they contradict the principles of the efficient market hypothesis. However, recent studies in the area of sentiment analysis of financial and social media news claim that FN could arguably affect the prices of financial securities (Mejia, 2022; Oliveira et al., 2013; Souza et al., 2015; Tetlock, 2007; Yu et al., 2015; Zhang & Skiena, 2010; Zheludev et al., 2014). Thus, this paper questions the market efficiency hypothesis and investigates whether there is empirical evidence to the contrary.

1.1. Motivation

The primary goal of this thesis is to measure the impact of FN on the stock market and to empirically show whether an efficient market, based on the principles of Fama, already takes FN into account when determining the stock price or whether the market is influenced by price reactions. Unfortunately, the FN from the SEC's investigation cannot be used because it has not been disclosed. Therefore, this work classifies financial texts from a novel, self-generated data set derived from the Motley Fool website using state-of-the-art natural language processing (NLP). Motley Fool is one of the largest shared platforms for financial news. Deriving from the initial situation, the following research questions were defined:

1. *Can we classify news articles from a crowd-founded financial website with a high degree of accuracy using state-of-the-art natural language processing trained on open FN data sest?*
2. *What influence does fake news articles have on the stock market?*

The following subsections describe the structure of this work.

1.2. Methodology and structure of the work

This thesis consists of six chapters. In section one, the initial situation, the problem statement, and the aim of this work have already been introduced. The second chapter provides an insight into the relevant literature on FN and its implications for the financial market. The third chapter covers machine learning (ML), deep learning (DL), and binary text classification, which are essential for an understanding of the rest of the thesis. Subsequently, the aim is to combine the findings of the second and third parts to implement an algorithm for binary text classification. Derived from these findings, the "Empirical research" chapter focuses on model building and evaluation as well as describing the data set. Furthermore, in this chapter the trained algorithm will be used to classify financial news articles from the website Motley Fool into legitimate or fake news. In this section, the first research question will be answered. To answer the second research question, an event study is conducted to examine return responses to the release of FN in the fifth chapter of this thesis. Chapter six discusses the results of this work and suggests recommendations for further studies.

1.3. Scope

This thesis measures the impact of FN on the stock market during the time frame 2011–2022, using stock news from the website Motley Fool. Chapter 4.1 presents in detail the selected data sets. Other financial news sources were not considered. Because the field of NLP is broad, it is not feasible to address all aspects of it in this thesis. The practical implementation of NLP for binary classification and measuring the impact of FN on financial securities is the focus of this work. The open-source libraries Keras, Tensorflow, and Sklearn were used, and no individual algorithms were built from scratch.

2. Literature review and related work

The pronounced lack of literature interpreting the impact of FN in the context of financial markets limits the relevance of conclusions that can be derived directly from previous studies. This chapter therefore attempts to highlight certain general characteristics of FN and make additional references to existing studies on the impact of FN on financial securities.

2.1. Fake news and financial markets

Financial news and investors' opinions transmitted through financial social media platforms can substantially forecast earnings surprises and influence stock prices. However, the current state of knowledge about financial disinformation on crowd-sourced platforms such as Seeking Alpha, Motley Fool, and StockTwits and their impact on the stock market is limited. The presence of FNs published on these financial social media platforms could distort trading volumes and affect returns on financial securities. For instance, Clarke et al. (2019, p. 16) emphasized in their study that investors' attention to FN is higher than to truthful news, especially in the first two days after publication. This finding is supported by Vosoughi et al. (2018, pp. 4–5), who stated that FN can spread faster and further than real news, in part because FN is often more extreme and exaggerated to increase its spread.

Previous studies in the literature have demonstrated that it is indeed possible to rely on linguistic stylistics to detect deception (Newman et al., 2003; Purda & Skillicorn, 2015). To better identify FN, studies by Clarke et al. (2019) and Niessner et al. (2018) used a Linguistic inquiry word count (LIWC) model to examine characteristics of expression to assess the differences between fake and real news. LIWC is a linguistic instrument that focuses on writing or speaking style and is uniquely useful for measuring individuals' cognitive and emotional states across domains (Pennebaker et al., 2015). They found that FN is generally longer, contains more words per sentence, and is more about money and numbers than legitimate news (Clarke et al., 2019, p. 22). These results are consistent with the evidence from a study by Zhou and Zhang (2008, pp. 121–122), which concluded that liars use more words per sentence and more words overall. Furthermore, people who tell lies tend to use fewer self-referential words, fewer insight and relative words, and more discrepancy verbs, as reported in the research of (Pennebaker, 2011, pp. 142–160).

The results of Clarke et al. (2019) and Niessner et al. (2018) are divergent, although they have similar data and aim to answer a common question, namely whether FN published on financial social media platforms influences the stock market and trading volume. Therefore, it is even more important to examine the findings of both studies in detail and possibly find disparities in their approaches to measuring abnormal return, trading volume, or dissimilarities in the data set. For instance, Clarke et al. (2019, p. 12) gathered abnormal returns data for 346 FN and 51,051 legitimate articles, with 12 different stock tickers associated with FN in their sample. For this reason, FN only affected a small number of companies they studied. To measure the impact of FN on stock price, Clarke et al. (2019, p. 12) used the standard event study with the market model to evaluate the excess return for firm i on day t:

$$ARet_{i,t} = Ret_{i,t} - Ret_{m,t}$$

where $ARet_{i,t}$ is the abnormal return, $Ret_{i,t}$ is the actual return, and $Ret_{m,t}$ is the return on the CRSP value-weighted index. This involved reviewing short- and long-term outcomes over four time periods. The short-term performances were measured over the $[0, +1]$ and $[0, +2]$ windows and the long-term abnormal return over six-month $[+3, +120]$ and one-year $[+3, +242]$ windows, where 0 corresponds to the publication date of the article on Seeking Alpha. They also examined the abnormal return over the window $[-30, -1]$ relative to the release date of the FN to control for run-up in the stock prior to publication (Clarke et al., 2019, p. 13). In addition, they evaluated the abnormal trading volume around the release date of the FN, which is represented with the following equation

$$\frac{Vol_{i,t}}{\overline{Vol}_{i,t} - 5, t - 65}$$

where for each stock i and day t, the abnormal trading volume $Vol_{i,t}$ is divided by the average trading volume $\overline{Vol}_{i,t}$ over the time window $[t\ -5]$ to $[\ t\ -65]$. They computed abnormal trading volumes over longer timeframes by cumulating the log of daily abnormal trading volumes. As above, abnormal trading was also calculated over the four windows. However, it is not clear for how many different tickers there are trading volume data available. The researchers only stated that they gathered abnormal returns for 322 FN and 49,900 legitimate news articles.

In their result, they compared abnormal returns and volumes associated with fake and legitimate news. They determined that FN generates significantly lower abnormal trading volume than legitimate news and that initial price effects are smaller than for corresponding genuine news, implying that agents partially discount FN (Clarke et al., 2019, pp. 24–25).

To acquire a better understanding of these studies in order to make reference to them, it is necessary to examine Niessner et al. (2018)'s approach to measuring abnormal return and volume. The effect of abnormal trading is measured as follows (Niessner et al., 2018, p. 21):

$$Vol\ (i,t)/^{\frac{1}{T}} \sum_{k=1}^{251} Vol\ (i,t-k)$$

"The trading volume for stock i on the day t relative to the average daily trading volume in stock i over the last year (250 trading days)." In addition, they sum abnormal volume over days $t = 0, t + 1, t + 2$ where $t = 0$ is the date of publication of the article on the website and then regress the logarithm of abnormal volume for any article on these platforms about a company.

Niessner et al. (2018, p. 21) reported a significant 12.1% increase in trading volume in the first three days after the publication date, which could mean that investors were trading in immediate response to news articles. Trading volume was six times larger for small firms than for large firms, which could mean that small firms traded more with retail investors and may have had a more opaque information environment (Niessner et al., 2018, p. 22). In contrast, Clarke et al. (2019, p. 24) did not see a significant difference in terms of trading volume generated by FN compared to legitimate news.

Based on the principles of the efficient market hypothesis from Fama (1970), in a perfectly efficient market FN should not have an impact on prices even if investors are attracted more by FN and trade on it. However, the key question is whether this hypothesis is still applicable more than 50 years after Fama's research or whether it is possible to observe an inefficiency of the market in the new age of rapid information distribution on various finance news platforms.

To capture the effect of FN on stock price using data from the SEC crackdown (171 FN-articles), Niessner et al. (2018, p. 34) used Fama's three-factor model, in which the cumulative abnormal returns are calculated as equally weighted residuals, complemented with a momentum factor, starting from the day after the article's publication until 251 trading days later. The authors expressed their approach to measurement as follows:

> Using equal-weighted portfolios of the market, size, value, and momentum factors (RMRF, SMB, HML,UMD) from Ken French's website, we estimate betas for each stock i for day t using past daily returns from $t-252$ to $t-1$. We then use those betas to calculate the residual (abnormal) cumulative returns, relative to the same four factors, for stock i for days $t+1$ to $t+251$(Niessner et al., 2018, p. 34).

Their results indicated that earnings, especially for small firms, increased by a total of 15% from the time of publication until after 60 days, before losing all gains and ending the year at -10%. The negative price impact of -10% results from the fact that the market may detect FN, and investors may see the release of misinformation as a negative signal for the company or that FN has temporarily delayed its decline (Niessner et al., 2018, p. 34).

In a further step, the authors used the same algorithm as Clarke et al. (2019), LIWC, to classify FN. They applied the LIWC to the data set they generated, which contained about 203,000 articles from the website Seeking Alpha and 147,000 from Motley Fool (Niessner et al., 2018, p. 11). The results were comparable to those obtained with the SEC's sample (171 FN); the yields for returns from misinformation increased by 8% over 6 months for small firms following the release date, but reverted back to their initial price level (Niessner et al., 2018, p. 35). This paper attempts to highlight potential differences that may have caused the disparity in the results of the above studies. First, to have a parameter for comparability, both studies would need to have identical approach to measure the expected returns for a given ticker.

Second, in the study by Clarke et al. (2019), the data sample includes a total of 383 FN articles with 27 different authors, whereas (Niessner et al., 2018) study examined a sample of 171 FN articles and 12 different authors. Third, using a sample with a total of 383 FN articles, Clarke et al. (2019) examined the impact on only 12 different firms, although they had a larger data set compared to Niessner et al. (2018), whose sample of 171 FN articles covers 46 publicly traded companies. This conclusion could also affect the results of the two studies, possibly leading to different findings.

Based on the articles mentioned above and their findings, certain aspects need to be critically questioned. The sample from the SEC contains only 171 FN articles. Both articles validate the LIWC on this sample and then calibrate it for a model to detect FN. However, the sample is rather small and unbalanced to be used as a stable sample for training an algorithm and applying it to a larger data set. To be more specific, the sample that Niessner et al. (2018) used to train the LIWC-algorithm contains 171 FN and 334 legitimate articles, and they applied it to a data set of around 350,000 articles. The Clarke et al. (2019) study also addressed machine learning. Using the linguistic features they generated from LIWC, the researchers presented six different machine learning models with the aim of detecting FN. The gradient-boosting classifier based on linguistic features achieved the highest accuracy of 87.1%. The current state of knowledge in text classification with DL and NLP already achieves accuracy levels of 93–97%. Therefore, the accuracy of the gradient-boosting algorithm is not adequate to derive subsequent insights with this approach.

3. Machine and Deep Learning

This chapter provides an insight into machine learning (ML), natural language processing (NLP), and deep learning (DL) with the aim of providing a foundation for the empirical part of this paper. This thesis will present findings from various works that have addressed the classification of texts using ML or DL. The aim is to use these references for the model building in Chapter four. Since the field of ML and DL is quite extensive, only those aspects that are of importance and of relevance to this work will be addressed. It is assumed that the reader already has an in-depth understanding of the concepts of ML, DL, NLP, and financial markets. The first subchapter is a brief introduction into ML. The second subchapter discusses DL approaches and highlights their benefits compared to classical ML models. In a next section, the different performance parameters are shown. The fourth subchapter describes the novel concept of using Word2Vec as an NLP technique. Findings from previous work and different DL models closely related to text classification are presented in the last chapter.

3.1. Classical machine learning

Chollet (2017, p. 6) defines machine learning as follows: “A machine learning model transforms its input data into meaningful outputs, a process that is ‘learned’ from exposure to known examples of inputs and outputs." ML can be divided into two main subtopics: supervised learning and unsupervised learning (Figure 1). A third type of ML, called reinforcement learning, is less frequently used (P. Murphy, 2012, p. 2).

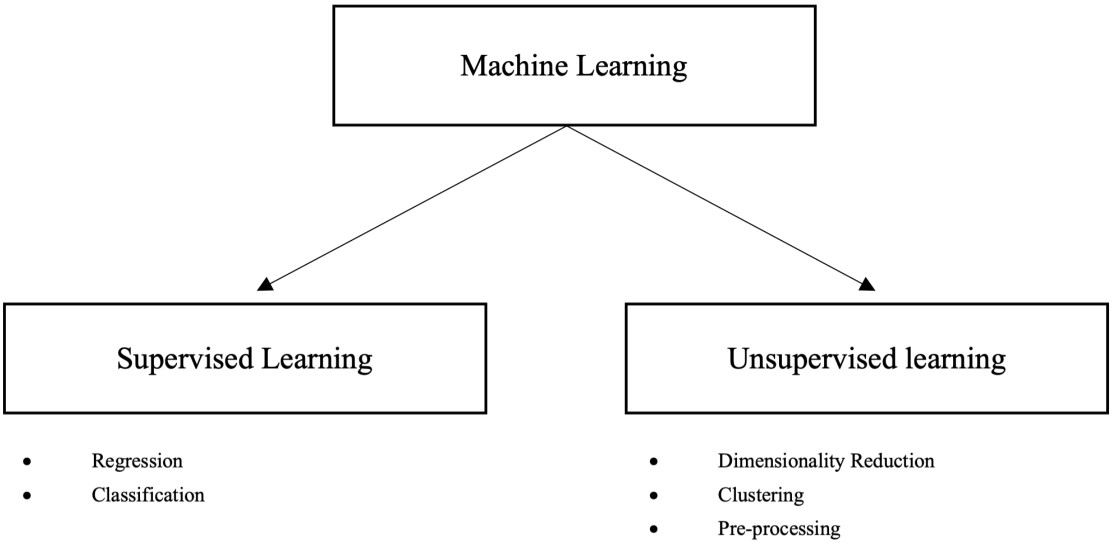

Figure 1: Machine learning (Fazlija, 2022)

Supervised learning takes a task-oriented approach and can be further divided into regression and classification problems. For instance, classification aims to obtain from the mapping of inputs x to outputs y, where $y \in \{1, \ldots, \mathrm{C}\}$ with C being the number of classes. If $\mathrm{C} = 2$, this is called binary classification (in this instance, it is frequently assumed that $y \in \{1,0\}$); if $\mathrm{C} > 2$, this is called multiclass classification (P. Murphy, 2012, p. 3). This thesis focuses on binary classification because articles are divided into two classes of news, legitimate $\{1\}$ or fake $\{0\}$. For the function $y = f(x)$, an unknown function f is assumed, and the main objective of this process and of supervised learning is to estimate the function f given a labeled training set. The next step, called generalization, is to make predictions using $\hat{y} = f(x)$ on novel inputs (i.e., ones not seen before); predicting the answer to the training set is easy as one can simply check the label (P. Murphy, 2012, p. 3). Chapter 4.1 describes in more detail the training set that is necessary for the training of the algorithm and the subsequent classification of Motley Fool articles with the trained algorithm. Regression is essentially the same as classification, with the difference that the response variable is continuous. It is applied widely to predict stock prices, the age of a viewer watching a particular video, or some indicators in the field of medicine (P. Murphy, 2012, pp. 8–9). In contrast to supervised learning, unsupervised learning tasks do not contain any input data. The purpose of this approach is to explore the uniqueness in the structure of the data (P. Murphy, 2012, p. 9). Unsupervised learning is often necessary to gain a better understanding of a data set before proceeding and solving a supervised learning problem. Pre-processing, dimensionality reduction, and clustering are well-established concepts of unsupervised learning (Chollet, 2017, p. 94). Below is a list of the 10 most used machine learning (ML) algorithms:

- Linear regression
- Logistic regression
- Decision tree
- Support vector machines (SVM)
- Naïve Bayes
- KNN
- K-means
- Random forest
- Gradient boosting and AdaBoosting

3.2. Deep learning

This thesis assumes that the reader already has a solid understanding of DL; if not, the reader is encouraged to consult the relevant literature. Chollet (2017, p. 8) defines DL as follows: "Deep learning is a specific subfield of machine learning: a new take on learning representations from data that puts an emphasis on learning successive *layers* of increasingly meaningful representations." These layered representations are trained by neural networks that are built in layers literally stacked on top of one another. See Appendix 8.1 for a neural network dictionary that provides a better understanding of the various terms used in DL and ML. The term neural networks dates back to the 1940s and is a reference to neurobiology (Fazlija, 2022). DL models are not models of the brain, even though some of the core concepts of DL were developed partly based on our understanding of the brain (Chollet, 2017, p. 8). The analogy between a biological neuron (left) and a mathematical model (right) is illustrated in Figure 2.

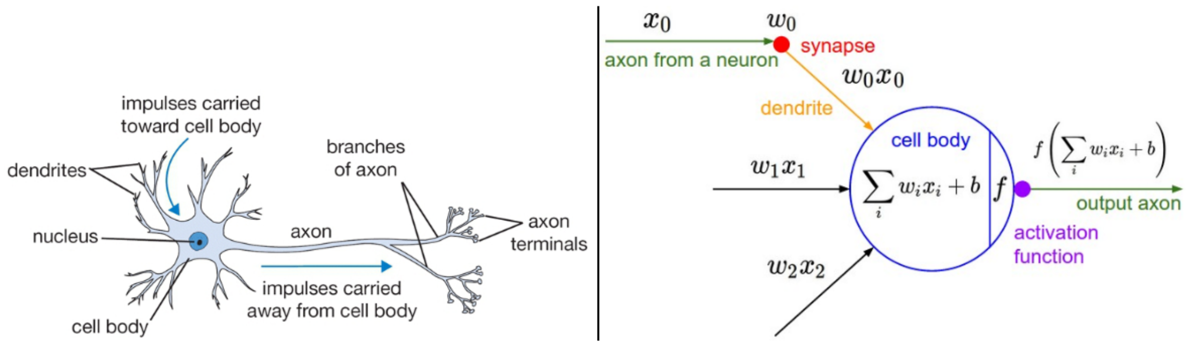

Figure 2: Neuron (Stanford, 2022)

From a mathematical point of view, each neuron performs a dot product with the input x and its weights w, adds the bias b, and applies the non-linearity (activation) function (Stanford, 2022). The activation function takes an input and performs a certain fixed mathematical operation on it. There are several activation functions, such as sigmoid, ReLU, leaky ReLU, tanh, or maxout (Stanford, 2022).

Neural Network architectures

Figure 3 illustrates the neural network architecture of a three-layer network with three inputs, two hidden layers of four neurons each, and one output layer (Stanford, 2022).

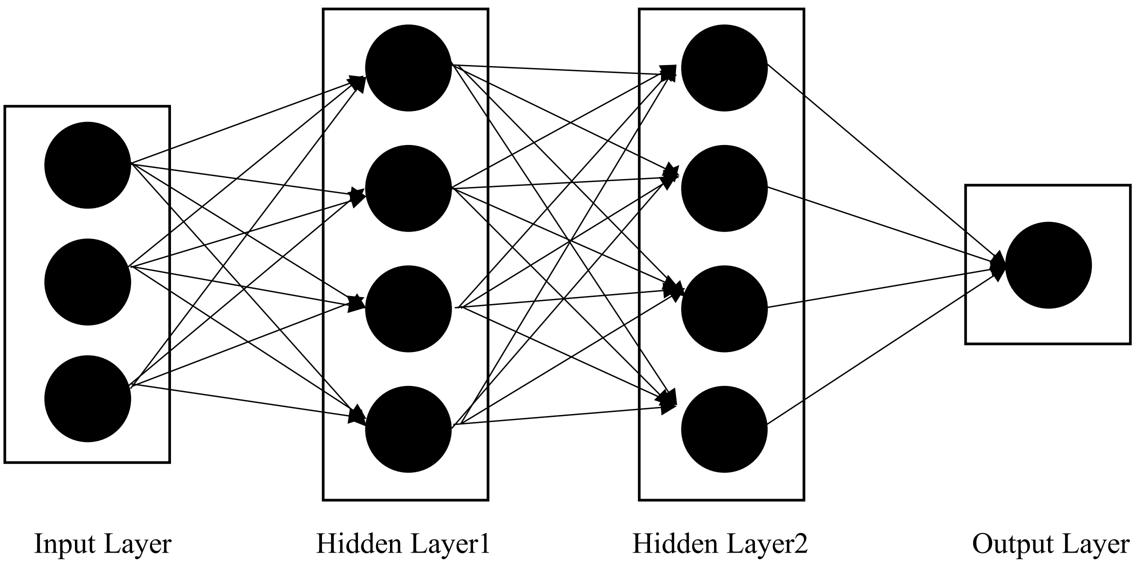

Figure 3: Neural network architecture (Stanford, 2022)

The network has 4 + 4 + 1 = 9 neurons, [3 x 4] + [4 x 4] + [4 x 1] = 12 + 16 + 4 = 32 weights, and 4 + 4 + 1 = 9 biases, for a total of 41 learnable parameters. As the size and number of layers in a neural network increases, so does the capacity of the network. This means that the space of representable functions increases because neurons can interact to express many different functions (Stanford, 2022). Neural networks with more neurons can render more complex functions, but this could quickly lead to *overfitting*. Overfitting occurs when a high-capacity model corresponds to the noise in the data rather than the (assumed) fundamental relationship. In practice, it is suggested to use *regularization* techniques to prevent overfitting instead of recuing the number of neurons (Stanford, 2022). Regularization is the general term for modifications to counteract overfitting. One of the simplest approaches to regularization is the *early stopping* technique, in which neural network training is stopped at the point where developmental perplexity is lowest, but this approach might not prevent overfitting (Charniak, 2019, p. 80). *L2 regularization* and *dropout* are better approaches to avoid overfitting, as suggested by Charniak (2019, p. 80). The dropout technique is applied to a layer with the aim of randomly dropping out (setting to zero) a number of output features during the training stage of the algorithm (Chollet, 2017, p. 109). After applying the dropout technique, the layer would return a vector with several zeros distributed at random. The dropout rate is commonly set between 0.2 and 0.5 and is the proportion of features that are set to zero (Chollet, 2017, p. 109). Charniak (2019, p. 81) provide an example in which dropout overcomes overfitting without any reversal in the complexity; even at 30 epochs, complexity is still decreasing. The question arises as to how this technique can reduce overfitting due to randomization to zero (Chollet, 2017, p. 109).Chollet (2017, p. 110) quotes Geoffrey E. Hinton in his book regarding dropout as follows:

> I went to my bank. The tellers kept changing and I asked one of them why. He said he didn't know but they got moved around a lot. I figured it must be because it would require cooperation between employees to successfully defraud the bank. This made me realize that randomly removing a different subset of neurons on each example would prevent conspiracies and thus reduce overfitting (Chollet (2017, p. 110).

The key concept is that introducing noise into the output values of a layer can break up random insignificant patterns (what Hinton calls conspiracies) that the network begins to store when there is no noise (Chollet, 2017, p. 110). The second technique is L2 regularization, commonly known as weight decay (Goodfellow et al., 2017, p. 231). This method aims to adjust the weights closer to the origin by adding a regularization term $\Omega(\theta) = \frac{1}{2} \parallel \omega \parallel \frac{2}{2}$ to the object function (Goodfellow et al., 2017, p. 231). Since describing each DL model is not feasible, only the model in use will be discussed in detail in the empirical part of this paper. Below is a list of the 10 most used machine learning (ML) algorithms:

1. Convolutional neural networks (CNN)
2. Long short-term memory networks (LSTM)
3. Recurrent neural networks (RNN)
4. Generative adversarial networks (GAN)
5. Radial basis function networks (RBFN)
6. Multilayer perceptrons (MLP)
7. Self-organizing maps (SOM)

8. Deep belief networks (DBN)
9. Restricted Boltzmann machines (RBM)
10. Autoencoders

The Keras library in Python provides a convenient approach to defining and training any kind of DL model (Chollet, 2017, p. 60).

3.3. Performance measurement

This subsection describes the different metrics used to evaluate an ML or DL model. There are several scores, such as accuracy, precision, recall or the F_1 score, that can be used as performance measures to evaluate a model in the context of a classification task. After the algorithm has predicted the data, a confusion matrix is generated from which several metrics can be derived. Each metric has a different importance depending on the problem. The metrics used in this paper are listed below:

Confusion Matrix

		Predicted Class	
		(+)	**(-)**
True Class	**(+)**	True positive (TP)	False negative (FN) **Type II Error**
	(-)	False positive (FP) **Type I Error**	True negative (TN)

Table 1: Confusion matrix (Fazlija, 2022)

Accuracy $$accuracy = \frac{true\ positive + true\ negative}{true\ positive + false\ negative + false\ positive + false\ negative}$$

Accuracy quantifies the proportion of predictions that our model evaluated correctly (Google, 2022). This metric is appropriate when the significance of the correct predictions of the classes is equal and when the classes appear homogeneous (Guttag, 2017, p. 406).

Recall $recall = \frac{true\ positives}{true\ positives + false\ negatives}$

Recall represents the ratio of correctly predicted true positives out of all true positive values. Recall is appropriate for instances where it is essential to maximize the identification of positive observations and the classification of the negative class is secondary (Guttag, 2017, p. 406).

Specificity $specificity = \frac{true\ negative}{true\ negative + false\ positive}$

Specificity represents the proportion of negatives that are correctly identified as such (Guttag, 2017, p. 406).

Precision $positive\ predicted\ values = \frac{true\ positives}{true\ positives + false\ positive}$

The positive predictive value is the probability that an example classified as positive is actually positive (Guttag, 2017, p. 406).

Precision $negative\ predicted\ values = \frac{true\ negatives}{true\ negatives + false\ negatives}$

The negative predictive value is the probability that an example classified as negative is actually negative (Guttag, 2017, p. 406).

F_1 Score $F_1 = 2 \times \frac{precision \times recall}{precision + r\ ecall}$

The F_1 score is the harmonic mean of recall and precision, with equal weighting. The F_1 score is intended for cases in which it is important for both the recall and the precision to be high (Chi Nhan & Oliver, 2022, p. 132).

3.4. Natural language processing

This subchapter aims to provide an overview of various traditional natural language processing (NLP) methods. Since there are many different approaches to NLP, the most common methods are described briefly. The Word2Vec approach will be described in detail, as it was chosen for this work based on the results derived in Section 3.5. An ML algorithm performs on a numeric feature space with inputs as a two-dimensional array in which the rows are instances and columns are features. To apply ML to text, it is necessary to transform the text into vector representation, which is an essential first step toward language-aware analysis (Bengfort et al., 2018, p. 55). The general preprocessing steps are listed below (Kulkarni & Shivananda, 2021, pp. 31–62):

1. **Text cleaning**
   Remove stopwords, HTML tags, punctuation, and emoji.

2. **Spelling correction**
   Use TextBlob library

3. **Tokenization**
   Tokenize the text.

4. **Lemmatizing or stemming**
   Normalizing text with stemming and lemmatization to reduce the number of features.

5. **Exploratory data analysis**
   Compute frequency distribution of words, word cloud, and length of text.

Bag of Words (BoW)

The most basic encoding of semantic space is the *Bag of Words (BoW)* model, whose most important insight is that meaning and similarity are encoded in vocabulary (Bengfort et al., 2018, pp. 55–56). Specifically, this model ranks the frequency of co-occurrence of words with themselves and other words in each limited context. Co-occurrence indicates which words are likely to follow each other, and by drawing inferences from limited pieces of text, large amounts of meaning can be captured (Bengfort et al., 2018, p. 13). A drawback of the BoW representation is that the word order is completely disregarded, and this could lead to a loss of the structure of the input text or the semantic content (Guido & Müller, 2016, pp. 327–339).
The process of performing the BoW representation for a corpus of documents consists of the following three steps and is illustrated in Figure 4 (Guido & Müller, 2016, p. 327):

1. **Tokenization**
   Split each document into the words that appear in it (called tokens).
2. **Vocabulary building**
   Gather a vocabulary of all words that appear in any of the documents and number them.

3. **Encoding**

For each document, count how often each of the words in the vocabulary appear in this document.

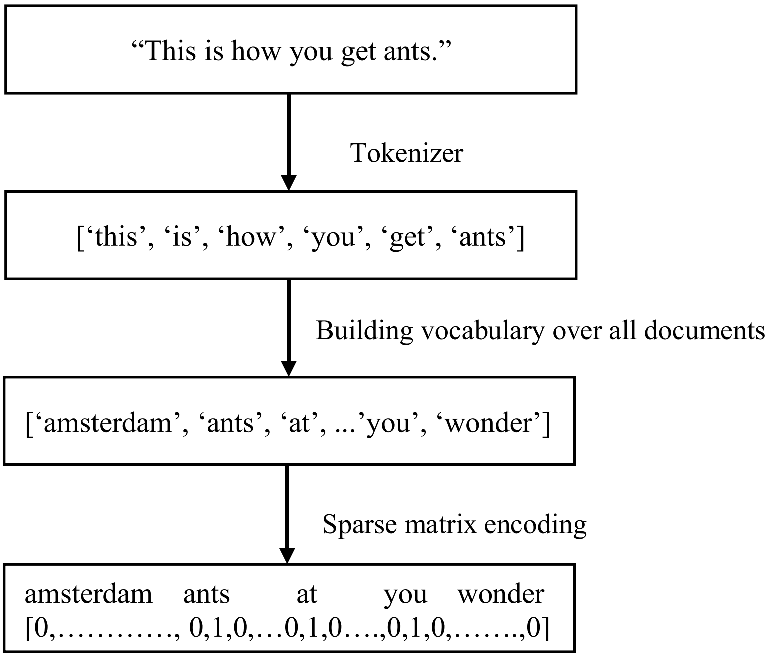

Figure 4: Bag of words processing (Guido & Müller, 2016, p. 327)

Term Frequency–Inverse document frequency (TF-IDF)

BoW representations describe a document in an independent way, without considering the context of the text. A more appropriate method would be to consider the relative frequency or rarity of tokens in the document compared to their frequency in other documents (Bengfort et al., 2018, p. 62). The TF-IDF method normalizes the frequency of tokens in a document compared to the rest of the corpus. This encoding approach highlights terms that are highly relevant to a particular instance (Bengfort et al., 2018, pp. 62–63). However, the TF-IDF approach has several limitations. For instance, it does not include the semantic similarities between words.

Word2Vec

To counteract the classical problem associated to the field of NLP as described above, this paper describes the Word2Vec algorithm, which was introduced by Google and Mikolov et al. (2013). Word2Vec is a recent breakthrough in the application of NLP and one of the most popular techniques for learning word embeddings by using neural networks. It can capture the context of a word in a document, the semantic and syntactic of the word, and its relationship to other words (Mikolov et al., 2013, pp. 4–6). In the process of word embedding, each individual word is transformed into a numerical representation of the word (vector), similar to the BoW approach. This vector is then trained based on either a continuous bag of words (CBOW) or skip-gram model, both of which have neural network architectures (Bengfort et al., 2018, p. 66). Word2Vec's effectiveness is based on its underlying ability to group vectors of similar words and provide estimates of a word's meaning based on its occurrences in the corpus (Mikolov et al., 2013, pp. 4–6).

Continuous bag of words (CBOW)

Figure 5 illustrates the architecture of the CBOW method. Similarly to a feedforward neural network, this architecture essentially takes the context of each word as the input and tries to predict a target word which corresponds to the context. The order of words does not affect the projection. Unlike the standard BoW model, this approach uses continuous distributed representation of the context (Mikolov et al., 2013, p. 4). The weight matrix between the input and projection layers is divided in an identical way for all word positions.

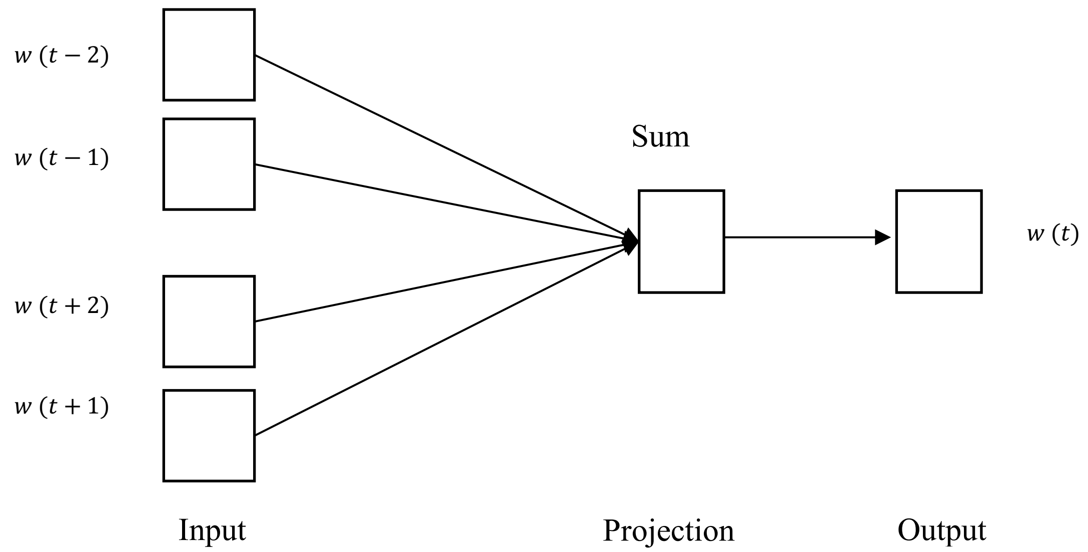

Figure 5: Continuous bag of words (CBOW) based on Mikolov et al. (2013, p. 5)

Continuous skip-gram model

The second model aims to maximize word classification based on another word in the sentence. The architecture of the model is shown in Figure 6. It takes the current word as input and tries to correctly predict the words before and after the input word. Experiments to evaluate the model's accuracy have shown that higher numbers of word vectors increase predictive quality but are also more computationally expensive (Mikolov et al., 2013, p. 4).

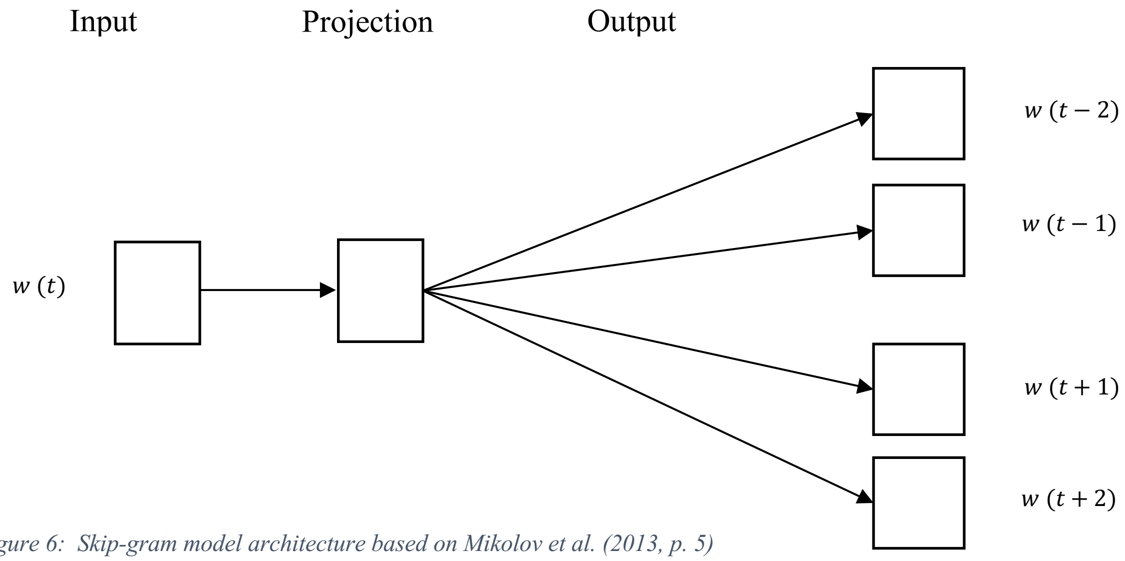

Figure 6: Skip-gram model architecture based on Mikolov et al. (2013, p. 5)

3.5. Fake news classification with Deep Learning

This chapter aims to briefly review the literature on classifying fake news using DL. The intention is to gain insights from the studies that are already available with a view to the application of these insights in the empirical part of Chapter 4.

Over the years, several studies have been carried out on the characteristics of FN and how to detect it. Rubin et al. (2015, p. 2) defined three type of FN—large-scale hoaxes, serious fabrications, and humorous fakes—and classified as FN articles that were intentionally and verifiably false and could mislead readers (Allcott & Gentzkow, 2017, p. 213). This strict definition is helpful in the sense that it can clarify the ambiguity between FN and related concepts such as hoax and satire. For the purposes of this thesis, only news articles that have been labeled as fake will be used. Other specific types of FN are not considered. Several studies have focused on the detection of FN by employing different DL models. The use of traditional ML models for text classification is widespread in the literature. However, in the last few years, it has been found that DL models can achieve higher levels of accuracy and better performance scores than traditional ML models. For instance, in earlier studies (Ahmad et al., 2020; Ahmed et al., 2017; Ruchansky et al., 2017), traditional ML models were outperformed by CNN, LSTM, Bi-LSTM, and BERT models. These findings have been supported by more recent evidence from multiple studies (Ali et al., 2022; Goldani et al., 2021; Khan et al., 2021; Mishra et al., 2022; Sastrawan et al., 2022). A comparison of several models proposed in different research papers is presented in Table 2.

Author	Embedding	Model	Accuracy	Precision	Recall	F1-Score
(Ali et al., 2022)	RoBERTa	CNN	97%	98%	96%	97%
(Ahmad et al., 2020)	LIWC	RF	95%	98%	93%	95%
(Khan et al., 2021)	Glove	C-LSTM	95%	95%	95%	95%
(Mishra et al., 2022)	Word2vec	Bi-LSTM	95%	95%	95%	95%
(Rai et al., 2022)	BERT	LSTM	89%	91%	90%	90%
(Sastrawan et al., 2022)	Glove	Bi-LSTM	95%	95%	95%	95%

Table 2: Comparison of fake news detection models

Even though the performance of the respective algorithms appears promising at first glance, it should be noted that most of the studies focused on recognizing news of a particular type (e.g., political). Based on the data set, they developed their models and designed features for these specific types of news. Most of the studies referred to above used the ISOT corpus, the Fake or Real corpus, the LIAR corpus, or a combination of these three sets. Such approaches could potentially be prone to a bias in the data set and a negative evaluation of news items on a distinct topic. This finding is clearly evident in the results of Khan et al. (2021, p. 8), where the Bi-LSTM model achieves 95% accuracy in the combined corpus, but only 54% in the LIAR data set. Therefore, it is advisable to examine whether ML or DL models are appropriate for different types of news by evaluating them on different data sets and benchmarking their performance. To conclude, it is important to have a data set that covers several different news categories. To address this issue, the next chapter presents the self-generated data set used for the training of a DL model. With the findings of this chapter as a starting point, this paper uses an LSTM model to classify FN. Based on a recurrent neural network (RNN) architecture, long short-term memory neural networks (LSTM) are widely used for NLP -tasks (Goodfellow et al., 2017, p. 18).

4. Empirical research

Thus far, this work has reviewed the literature on FN recognition and discussed ML, DL, and NLP. To provide a better understanding of the various tasks involved in this work, Figure 7 illustrates the different steps in the process, from data collection to the final task of measuring the impact of FN on the share price. Two different data sets were generated. The first, referred to in this paper as the labeled FN-RN data set, contained both fake and legitimate news. Subsequently, the DL model was trained on this set of data. Once the DL model had been trained and evaluated, the next step was the classification of Motley Fool articles into fake or legitimate news. Finally, the identified FN articles with the corresponding tickers are examined using an event study approach to measure the response to the return.

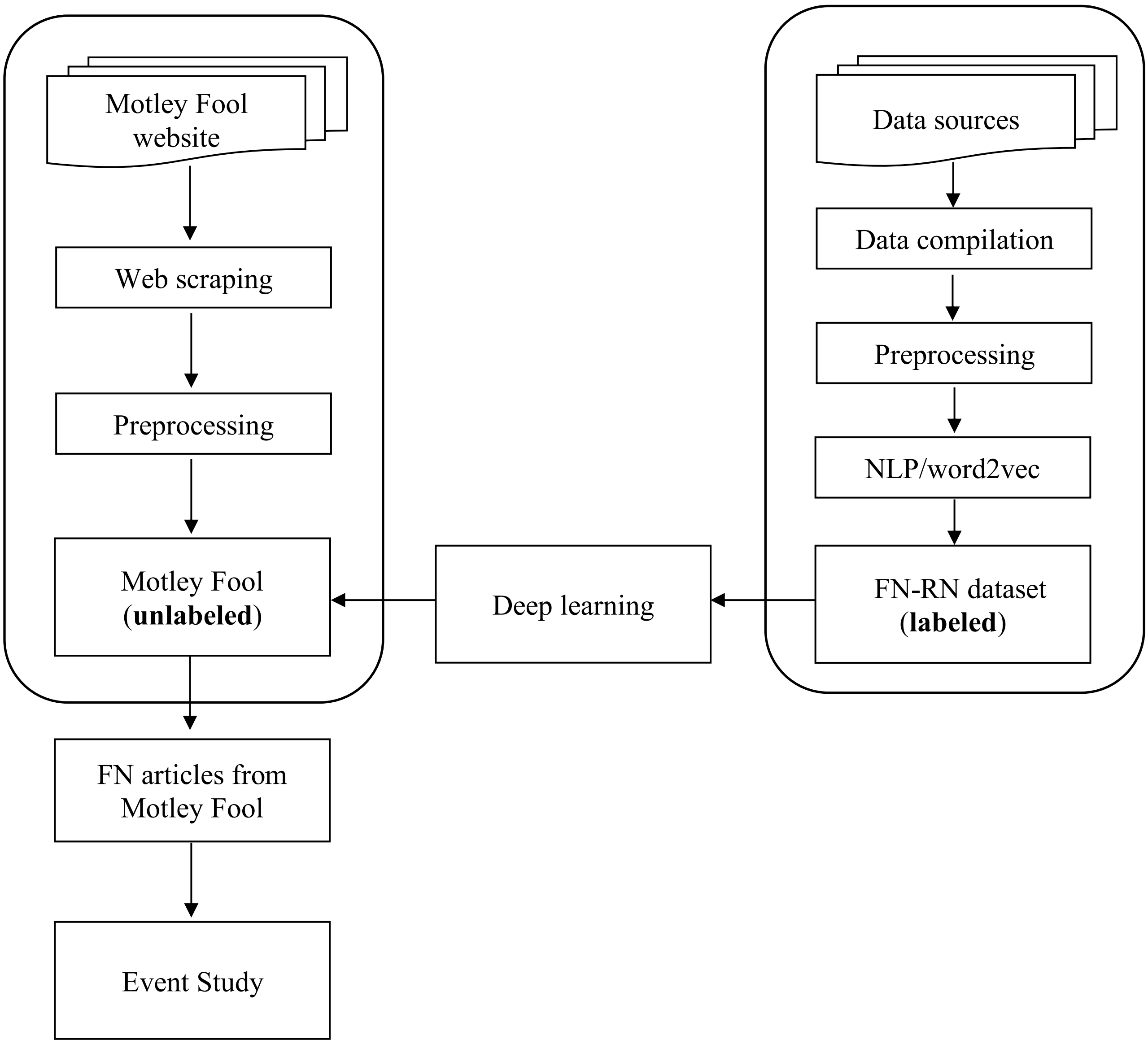

Figure 7: Overview of the empirical research

4.1. Data

The different data sets used in this thesis and their descriptive statistics are described in this chapter. This chapter also details the preprocessing steps required before using the data as input for a machine learning model.

4.1.1. Motley Fool data set

Recent studies Chen et al. (2014) and Niessner (2015) indicate that financial platforms on social media forecast trading volume, earnings surprises, and stock price. News articles from the one of the largest crowd-sourced sites, Motley Fool, are thereby used to measure whether FN has a significant impact on the financial market. The Motley Fool is a multimedia financial services company that offers financial investment advice to investors through a shared knowledge platform (NASDAQ, 2022).

Descriptive Statistic

The Beautiful Soup Package in Python can be used to directly extract vast amounts of information from web sites (Kulkarni & Shivananda, 2021, p. 24), which are generally XML or HTML documents (Richardson, 2022). This Python library was used to extract news articles from the Motley Fool website. The generated data set contained a total of 404,947 news articles as well as 10 different types of information over the period 2011 to 2022. Information extracted specifically from The Motley Fool is presented in Table 3.

Author	John Ballard
Author Link	https://www.fool.com/author/16707/
Date	September 8, 2022
Article-Title	Why Intel Stock Was Down Earlier This Morning
Article-Text	Intel CEO Pat Gelsinger sees better revenue stability in the fourth quarter. What happened? Shares of Intel (INTC 0.85%) were down as low as 2.4% after [..]
Link	https://www.fool.com/investing/2022/09/08/why-intel-stock-was-down-earlier-this-morning/
Name	Intel Corporation
Ticker	INTC
Market Cap	$117B
Price	$28.07

Table 3: Data items Motley Fool

In total, the data set contained more than 5,579 different tickers. With about 11,587 news articles, Apple was the company with the most articles written about it, followed by Amazon, Netflix, Meta, and Tesla. The top 10 list is shown in Table 4.

Ticker	**Name**	**Numbers of Articles**
AAPL	Apple	11,587
^DJI	Dow Jones	7,499
AMAZN	Amazon	5,045
NFLX	Netflix	4,132
META	Meta/Facebook	3,761
DIS	Walt Disney	3,573
GOOGL	Alphabet	3,380
MSFT	Microsoft	3,098
F	Ford	2,815
INTC	Intel	2,612

Table 4: Top 10 tickers with the most articles

The 10 authors who wrote the most articles on Motley Fool are listed in Table 5. These 10 authors cumulatively published 116,445 news articles between 2011 and 2022, 28% of the total data set. In total, 4,367 different authors published articles on the platform. To express the relationship in numbers, 0.0022% of authors were responsible for 28% of the articles written on the Motley Fool.

Author Name	**Numbers of Published Articles**
Seth Jayson	25,541
Dan Caplinger	16,366
Sean Williams	11,579
Rich Smith	11,372
Maurie Backman	11,068
Travis Hoium	8,377
Matthew DiLallo	8,277
Rich Duprey	8,251
Rick Munarriz	8,035
Keith Speights	7,579

Table 5: Top 10 authors with most written articles

To illustrate the characteristics of the texts, a column was added with a measure of the number of words per text. The average length of an article was 672 words per article. The shortest text contained three words, and the longest text had 6,190 words.

Feature	**Text length**
mean	671
Std	482
min	3
25%	405
50%	650
75%	814
Max	6,190

Table 6: Text length Motley Fool

4.1.2. Data analysis and preprocessing

The previous section briefly described the characteristics of the data set. In this section, the focus is on the preprocessing as well as the adaptation of the data set so that it can be used as input in machine learning. The five preprocessing steps introduced in Chapter 3.4 and the exploratory data analysis to obtain insights are therefore described in the next subchapter.

1. Extended stopword list and text cleaning

A common procedure in NLP and in preprocessing is the removal of stopwords to better represent the semantics and context of a text. Python comes with two different stopword lists: sklearn, containing 318 stopwords, and nltk, with 179. A novel stopword list was created from the two libraries, with a total of unique 378 stopwords, in an attempt to exclude as many stopwords as possible from the articles. Furthermore, the most frequent words in the data set were analyzed to see if other words occurred frequently (e.g., “images,” “sources”), and these were added to our list of stopwords. The word "images" or "source" appeared very often in the text because the articles also contained HTML tags at this point and this word refers to those tags. The next step was to write a custom function in Python to remove stopwords, HTML tags, emoticons, and punctuation from the articles. The text was set to lower case.

2. **Dropping rows containing NaN values and crypto news**

The Motley Fool data set had numerous NaN values in various columns. If a row contained a NaN value in the title, article body, author, or post date columns, the row was deleted. This step was taken because empty texts or texts without an author were not considered for this work. Publication date is vital for the evaluation in Section 5, because the influence of FN should be measurable in a specific timeframe. Of the 404,947 articles, 110,441 had no ticker or firm name, and 154,493 did not contain a market cap. However, these rows were not deleted in advance because not all articles have a primary ticker and an article may cover a variety of stocks. Over the period covered in the data set, authors also published an increasing number of articles about cryptocurrencies. These articles were deleted from the data set as they were not relevant to this study. Rows with duplicates were removed.

3. **Threshold for text length**

For an article to have adequate informational content, it must be of a certain length. Based on the statistics in Table 6, it is important to set an appropriate threshold for the number of words per text to exclude articles that are too short while not losing too many articles from the data set. Therefore, the percentiles were used to determine whether exclusion above a given threshold would result in the loss of an excessive number of texts. There were a total of 126 articles that contained less than 150 words each, which corresponds to approximately the 5th percentile of text length. If the threshold were set to more than 150 words per article, fewer articles would be removed from the data set. The 99th percentile of the text length is 2,395.

Percentile	**Numbers of Articles**
5th percentile of text length	127
10th percentile of text length	267
25th percentile of text length	405
50th percentile of text length	650
75th percentile of text length	814
90th percentile of text length	992
95th percentile of text length	1,143
99th percentile of text length	2,395

Table 7: Percentiles of text length Motley Fool

After the application of the criteria described above, our data set contained 396,642 rows and 12 columns. Figure 8 illustrates the distribution of text lengths and their frequency in the data set.

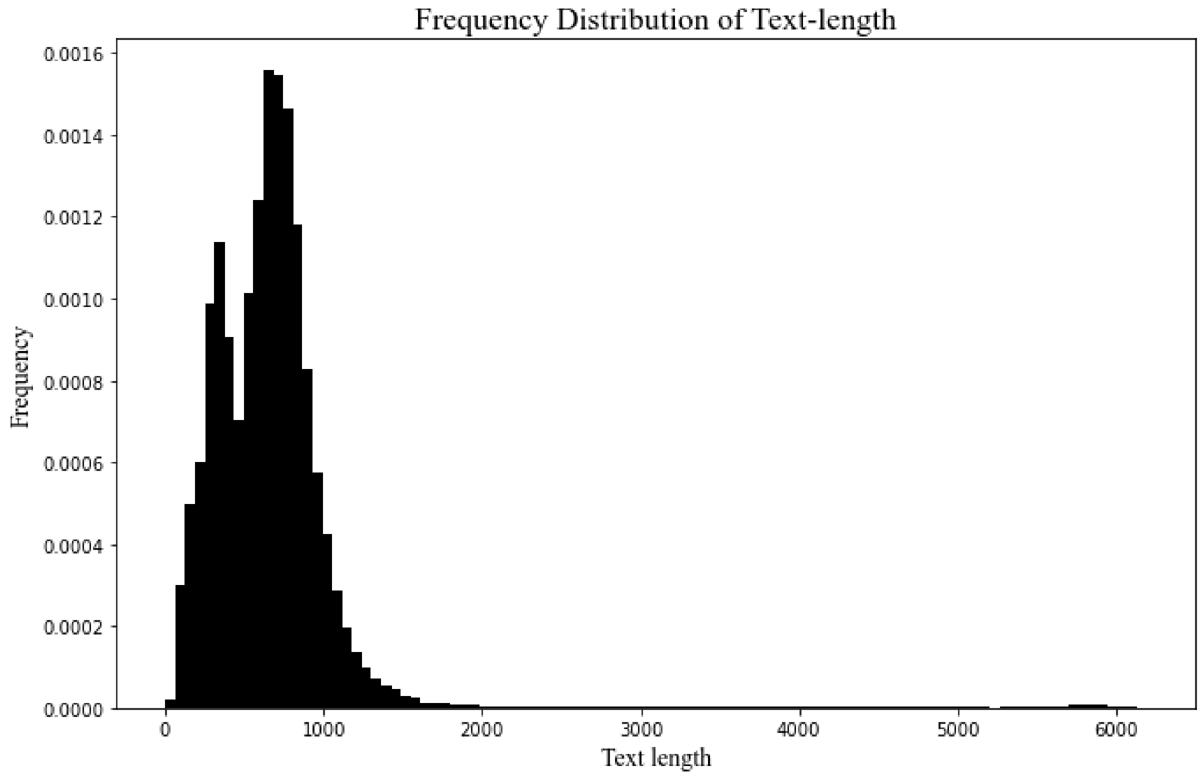

Figure 8: Frequency distribution of text length Motley Fool

4.2. FN-RN data set

The underlying issue in ML or DL is the inherent relationship between *optimization* and *generalization*. The aim of optimization is to adjust the model to achieve the best performance scores. On the other hand, the term generalization refers on how well our trained model predicts on novel data. Different generalization techniques such as dropout or L2 regularization can be implemented, but it is not possible to control generalization because the algorithm can only be adjusted based on its training data (Chollet, 2017, p. 104). For this reason, it is important to have a data set that is large enough to allow the algorithm to learn about the different patterns in the data structure.

Descriptive Statistics

As described in the introduction, Clarke et al. (2019) and Niessner et al. (2018) reviewed only 171 different FN articles. This sample is considered small and could lead to the model learning deceptive or irrelevant patterns in the training data. The best way to prevent overfitting is to gather more training data for the model (Chollet, 2017, p. 104). Therefore, a unique data set of labeled news articles from various sources was compiled and processed in this work. Table 8 lists the different data sets and their characteristics.

Data set	Category	Fake	True	Total entries
1: Fake news	Politics, sports, entertainment	2,120	1,868	3,988
2: ISOT data set	Economy, political, entertainment	23,481	21,417	44,898
3: LIAR data set	Economy, politics, healthcare	2,833	3,638	6,471
4: GM data set	Economy, politics	3,164	3,171	6,335
5: Fake or real	Economy, politics, entertainment	10,387	10,413	20,800
6: FN-Set	Economy, politics	44,459	96,024	140,483
7: Guardian data set	Economy, politics, sports, art, technology, culture	52,462	0	52,462
8: FakeNews Net	Politics, social media	5,755	17,441	23,196
Total		***144,661***	***153,972***	***298,633***

Table 8: Data set FN-RN

The first data set is available at Kaggle.[1] It includes a total of 3,988 articles, both fake and legitimate. True articles come from sources such as The New York Times, Reuters, and CNN, whereas FN articles are from various other websites. This data set was used in the paper from Ahmad et al. (2020). The second in the table is the ISOT data set,[2] which contains 44,989 real news items obtained by crawling articles from the Reuters website and FN articles checked by Politifact (a fact-checking organization in the US) and Wikipedia. Most of the articles in this data set focus on political news, but there are also general news articles that cover a variety of different news categories. This data set was used in the studies from Ahmed et al. (2017); Mishra et al. (2022) and Samadi et al. (2021). The LIAR[3] data set contains 10,239 articles. However, this data set has a total of six different labels for the articles: “true, “mostly true,” “half true,” “barely true,” “false,” and “pants on fire.” As the classifications of “barely true” or

[1] *https://www.kaggle.com/datasets/jruvika/fake-news-detection*
[2] *https://www.kaggle.com/datasets/clmentbisaillon/fake-and-real-news-dataset?resource=download*
[3] *https://www.kaggle.com/datasets/csmalarkodi/liar-fake-news-dataset*

"half true" are not very clear, the texts with these labels have been left out. Therefore, only the articles labeled as "mostly true," "true," "false," and "pants on fire" are used. Different categories of news, such as business, politics, and health, are covered in this data set. This data set has also been used as the basis for many studies, including those published by D'Ulizia et al. (2021), Khan et al. (2021), Samadi et al. (2021), and Wang (2017). The GM data set[4] contains 6,335 different articles and was used in the paper of (Khan et al., 2021; Sastrawan et al., 2022) and includes articles in the areas of economics and politics. The articles in the Fake or Real[5] data set are not limited to a single field such as politics but include articles from various other fields. These data have been used in a number of studies on the classification of fake news (Ahmed et al., 2017; Mishra et al., 2022). The next data set, FN-Set,[6] contains approximately 426,00 unique articles. Although the size of this data set appears to be extensive at first glance, it contains a high proportion of articles written in other languages. After removing all non-English articles, 140,483 remained in the data set. Another issue with this data set is that the news items are classified into 12 different labels. Therefore, only the unique labels *fake* and *reliable* from this data set are considered. Labels such as unknown, satire or hate are not taken into further account. The Guardian Dataset[7] can also be found on Kaggle and contains 52,462 articles from the newspaper website *The Guardian*. It includes several news categories. FakeNews Net[8] can be retrieved through the Harvard database and is a multidimensional data repository that currently contains two data sets of news content: social context and spatiotemporal information (Harvard, 2022; Shu et al., 2019). After the collection of all the data from the different sources, it is evident that there is no inherent issue with the imbalance ratio between the fake news and the legitimate news. Furthermore, the new concatenated data set of approximately 300,000 articles from different sources contains data in a diverse set of categories. Not only political news was considered, but also other topics such as the economy, sports, the arts, social media content, and health care are represented. The next section focuses on the exploration and preprocessing of the data set to obtain a more detailed insight into the new data set. The author of this paper calls the newly generated data set "FN-RN."

[4] *https://github.com/joolsa/fake_real_news_dataset*
[5] *https://www.kaggle.com/c/fake-news/data*
[6] *https://www.kaggle.com/datasets/pontes/fake-news-sample?resource=download*
[7] *https://www.kaggle.com/datasets/sameedhayat/guardian-news-dataset*
[8] *https://dataverse.harvard.edu/dataset.xhtml?persistentId=doi:10.7910/DVN/UEMMHS*

4.2.1. FN-RN analysis and preprocessing

The different stages of pre-processing have already been described briefly in Chapter 4.1.2; therefore, they are not described in detail here. This chapter highlights the characteristics of the newly generated FN-RN data set and provide an insight into some of its features.

1. Text cleaning and dropping non-English articles

As already noted, the FN-Set[9] contained articles that were not in English. Therefore, the entire data set was first examined to see if any foreign language articles were present. The function `detect` from the library langdetect can evaluate the language of each text and return its abbreviation. All non-English articles were removed. Second, all rows containing duplicates were excluded. A new column was added to the data set to represent the text length for fake and legitimate articles. Table 9 presents the text length of articles, with RN standing for legitimate news and FN for fake news.

Features	Text length of RN	Text length of FN
count	150,742	133,825
mean	440	644
std	562	1,064
min	1	0
25%	89	167
50%	271	416
75%	641	774
max	32,551	30,114

Table 9: Text length for true and fake news articles in FN-RN

2. Threshold for text length

The descriptive statistics indicate that fake news tends to be longer than legitimate news. This finding is consistent with the evidence from the literature review and with the results of research by Zhou & Zhang (2008) showing that liars use more words and more words per phrase. As with the Motley Fool data, a threshold value needed to be defined for FN-RN. The percentiles listed in Table 10 were used to determine whether an given exclusion threshold would result in the loss of an excessive number of texts.

[9] *https://www.kaggle.com/datasets/pontes/fake-news-sample?resource=download*

Percentile	Number of articles
5th percentile of text length	12
10th percentile of text length	22
25th percentile of text length	123
50th percentile of text length	332
75th percentile of text length	710
90th percentile of text length	1,091
95th percentile of text length	1,435
99th percentile of text length	3,869

Table 10: Percentile of text length of articles in FN-RN

This data set contained more articles with fewer words than the Motley Fool data set. The threshold was set at more than 30 words per article to preserve articles with sufficient content while avoiding removing an excessive number of articles from the data set. After applying these criteria, the data set contained a total of 251,240 articles. Of these, 129,192 were legitimate news items and 122,048 were FN items. Figure 9 illustrates the distribution of the number of words per text and their frequency in the FN-RN set.

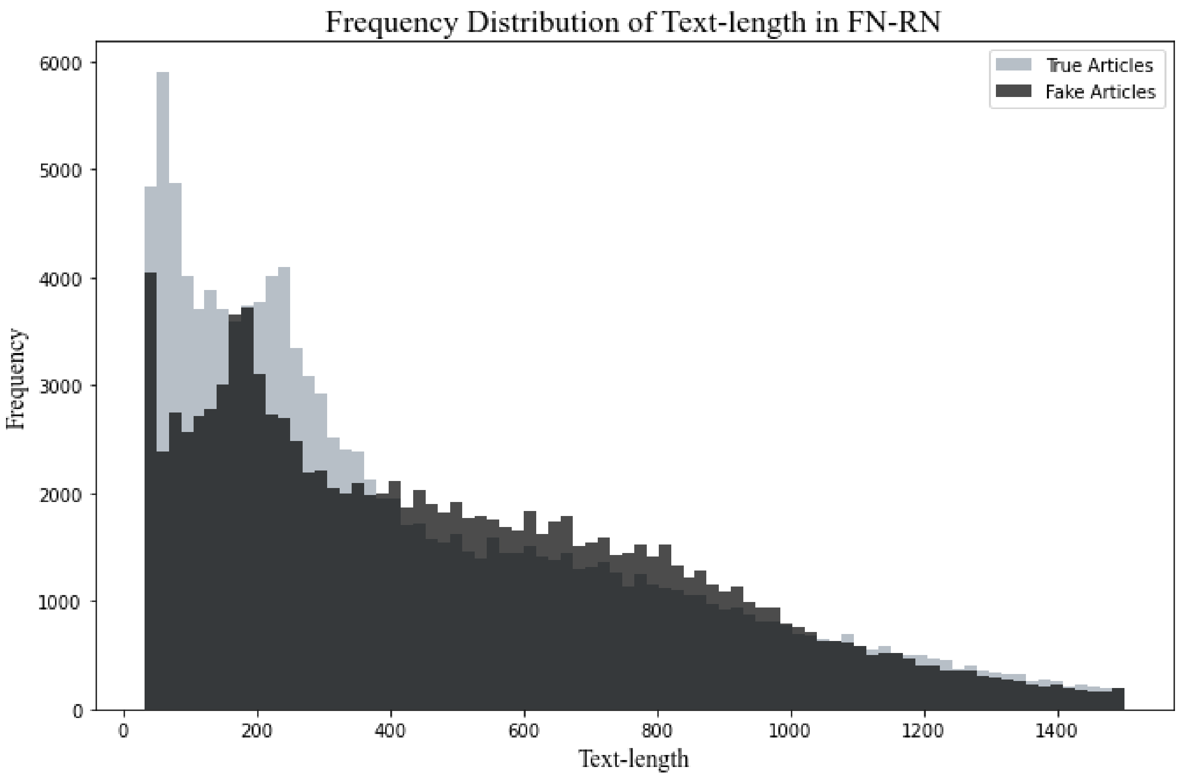

Figure 9: Frequency distribution of text length of articles in the FN-RN data set

Table 11 shows two examples from the FN-RN data set after removing stopwords, HTML tags, emoticons, and punctuation. The binary labels 0 and 1 stand for false and true, respectively. The text length column denotes the total number of words in the article.

Text	Label	Text length of RN	Text length of FN
donald trump sends embarrassing new year's eve message [..]	0	NaN	267
trump's fight with corker jeopardizes his legislative agenda [..]	1	1,094	NaN

Table 11: FN-RN data set

4.3. Model building

Starting from the findings of Section 3.5 and from works by Chollet (2017), Goodfellow et al. (2017), and P. Murphy (2012) on text classification with DL, this thesis uses an LSTM model and Word2Vec as word embedding. In the following subsections, the adopted model is described in detail along with the parameters used.

Word2Vec

The parameters for the Word2Vec word embeddings are presented below. This thesis uses Word2Vec for the embedding layer of the LSTM model with an CBOW method.

Parameter	Input	Description
Dimensionality	200	Dimensionality of the feature vector
Windows	5	Maximum distance between the current and predicted word within a sentence
Min_count	1	Ignores word with a total frequency lower than this
max_len	1,000	Takes the first 1,000 words from a text, based on the frequency distribution
Size of vocabulary	768,597	The total size of the vocabulary in the data set

Table 12: Word2Vec word embeddings parameters

4.3.1. LSTM

The FN-RN data set was used to train the model. To emphasize the importance and relevance of the algorithm, only the parameters that can have a significant impact on the model are described. Therefore, the steps of tokenizing and vectorizing the text so that the algorithm could process the data as numerical vectors are not explained in detail. The split into training and test sets is 0.80 and 0.20, respectively. The LSTM model was trained with 200-dimensional Word2Vec embeddings. The network consists of five layers. The first layer contains the embedding vectors as weights which are generated by Word2Vec; the input length is set to 1000 and the output dimension to 200. Layer 3 is a dense layer consisting of 128 hidden units. Layers 2 and 4 are dropout layers using a rate of 0.2 for regularization and to avoid overfitting for generalization. The last layer is a two-way *softmax* layer, which returns an array of two probabilities. Each score represents the probability of an article being fake or true. The compilation setup contains three parameters—*loss*, *optimizer,* and *metric*—to monitor performance during the training of the model. The loss function *sparse_categorical_crossentropy* was implemented. The stochastic gradient descent algorithms modification, commonly referred to as *adam* optimizer, was applied to update the weights and reduce the learning error. *Accuracy* was chosen as a metric. The architecture of the model is illustrated in Figure 10.

LSTM Model

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 1000, 200)         153719600

dropout_2 (Dropout)          (None, 1000, 200)         0

lstm_1 (LSTM)                (None, 128)               168448

dropout_3 (Dropout)          (None, 128)               0

dense_1 (Dense)              (None, 2)                 258

=================================================================
```

Figure 10: LSTM model architecture

In the first run, the model iterates on the training data in batches of 512 samples with 30 *epochs*. At each epoch, the model computes the gradients of the weights with respect to the loss of the batch and updates the weights (Chollet, 2017, p. 53). Figure 11 represents the accuracy of the LSTM model on the training set as well on the validation set. After 10 epochs, the algorithm tends to overfit. The loss on the validation is higher than on the training set.

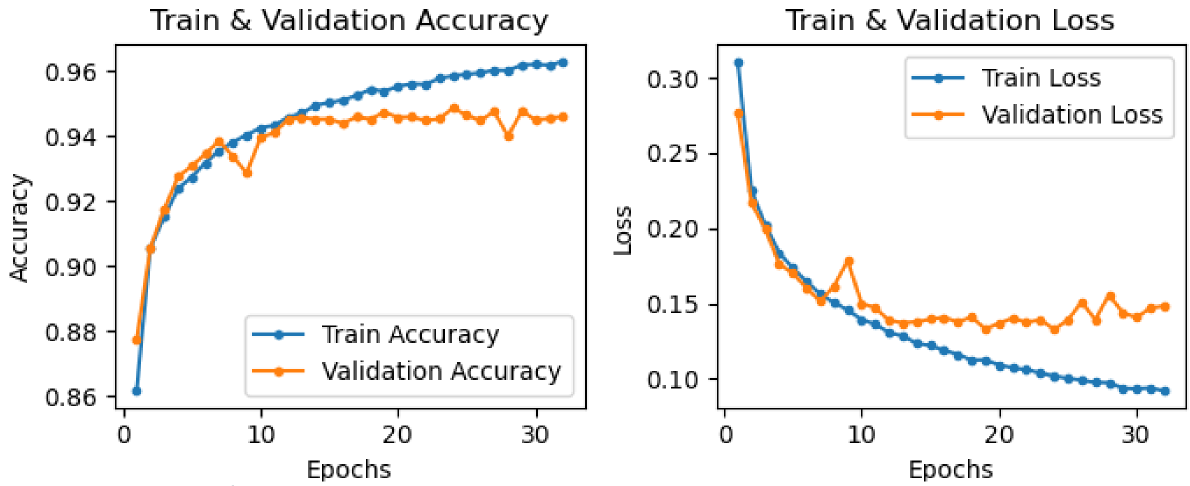

Figure 11: Accuracy and Loss - LSTM

The classification report provides insights into how well the algorithm evaluates fake or real news. It is notable that the algorithm with Word2Vec as word embeddings and the configurations described above already yields high performance scores.

	Precision	**Recall**	**F1-Score**
Fake	94%	95%	95%
True	96%	94%	95%
Accuracy			95%
Macro avg	95%	95%	95%
Weighted avg	95%	95%	95%

Table 13: Classification report of LSTM

The results were comparable to those of the FN classification studies in Chapter 3.5. The next section focuses on the further improvement of the presented model in terms of accuracy and reduction of overfitting.

4.3.2. Regularizing and Hyperparameter optimization

There are seemingly arbitrary decisions involved in the process of building a deep learning model. Questions arise regarding the size of the layers, how many units each layer should contain, what activating function should be implemented, or how much dropout should be used (Chollet, 2017, p. 263). These architecture-level parameters are referred to as hyperparameters to differentiate them from the parameters of a model that are trained using backpropagation. In practice, skilled researchers have developed intuition about the configuration of model structure and inputs (Chollet, 2017, p. 263). However, most of the time is taken up by building a model, testing it, and repeating the process with adjusted parameters. In this chapter, several techniques for model optimization and overfitting are discussed and further implemented in the model with the aim of improving the algorithm in terms of accuracy and overfitting. Table 14 lists the different techniques and their associated properties.

Feature	**Settings**
Regularization	L2-Regularization
Epochs	10, 20, 30, 40
Batch Size	64, 128, 256, 512
Optimizer	Adam, ReLU, Rmsprop
Layers	Convolutional Layer
Early Stopping	4
Callbacks	Save the best model

Table 14: Hyperparameter optimization

According to Chollet (2017, pp. 107–108) and Goodfellow et al. (2017, p. 231) L2 regularization—commonly known as weight decay—can help to reduce overfitting in a neural network. In addition, the model was adapted with the early stopping technique, which interrupts the training of the algorithm after four epochs when the validation loss no longer improves. The dropout technique, the most efficient and widely used regularization technique, has already been presented in the last model. As part of the hyperparameter optimization, a one-dimensional convnet (`Conv1D`) layer with a window size of 7 and 32 filters was set, as well as a `Maxpooling1D` layer of pool size 2. This approach was proposed by (Chollet, 2017, pp. 253–254; Khan et al., 2021, p. 8) for binary text classification. The algorithm was trained with different batch sizes of 64, 128, 256, and 512 and different epochs. The process of optimizing the algorithm was extensive. Numerous combinations of different techniques and settings were tested. As it would be impractical to present all possible combinations, the model that achieved the best results with its corresponding features is described below.

4.4. Model evaluation

4.4.1. C-LSTM

Figure 12 illustrates the architecture of the new proposed C-LSTM model with a convnet layer.

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 1000, 200)         153719600

 dropout_1 (Dropout)         (None, 1000, 200)         0

 conv1d_1 (Conv1D)           (None, 1000, 32)          19232

 max_pooling1d_1 (MaxPooling  (None, 500, 32)          0
 1D)

 lstm_1 (LSTM)               (None, 128)               82432

 dense_1 (Dense)             (None, 2)                 258

=================================================================
```

Figure 12: C-LSTM model architecture

The final parameters are listed in Table 15.

Feature	Settings
Epochs	30
Batch Size	128
Optimizer	Adam
Loss	Sparse categorical crossentropy
Activation	Softmax
Metrics	Accuracy
Layers	Convolutional Layer
Early Stopping	4
Callbacks	Save the best model

Table 15: Hyperparameter C-LSTM

By adjusting the parameters, the validation loss could be significantly improved compared to the first model. After 10 epochs, the algorithm stops by implementing early stopping. As a result, the validation loss is approximately 0.07.

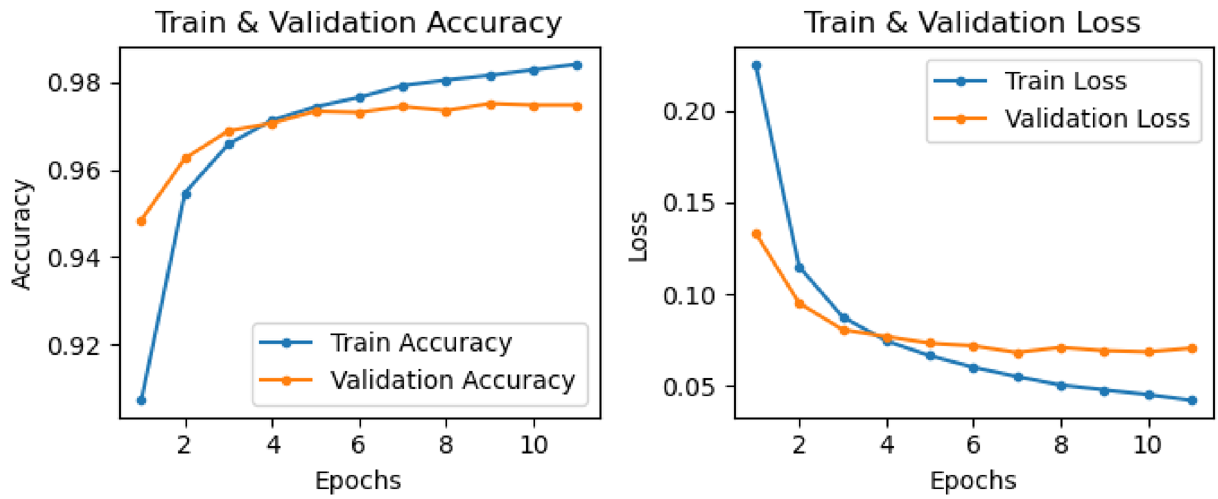

Figure 13: Accuracy and loss – C-LSTM

The classification report shows that the algorithm achieved significant improvements in the different performance metrics. The construction of the LSTM model using Word2Vec for the word embeddings and the newly added convnet layer clearly demonstrates that this approach can achieve state-of-the-art results in binary text classification.

	Precision	**Recall**	**F1-Score**
Fake	96%	98%	97%
True	98%	96%	97%
Accuracy			97%
Macro avg	97%	97%	97%
Weighted avg	97%	97%	97%

Table 16: Classification report C-LSTM

The comparison of the confusion matrices of both models is shown in the next figure. It can be observed that the C-LSTM can predict more accurately than the LSTM, and it also has a lower rate of type I and type II errors, which is preferred in the context of this work. A type I error, or false positives (FP), means that the model has classified a news article as true even though it is fake. For instance, the LSTM predicted 1,093 news as true even though they were fake. In comparison, the C-LSTM model produced only 442 FPs. A 2% percentage point improvement in the accuracy of the C-LSTM model leads to a significant improvement in the classification of the FN. It is important to highlight these findings in relation to the prediction of the Motley

Fool data set of around 400,000 news articles in total. A higher proportion of type I or II errors could potentially bias the results.

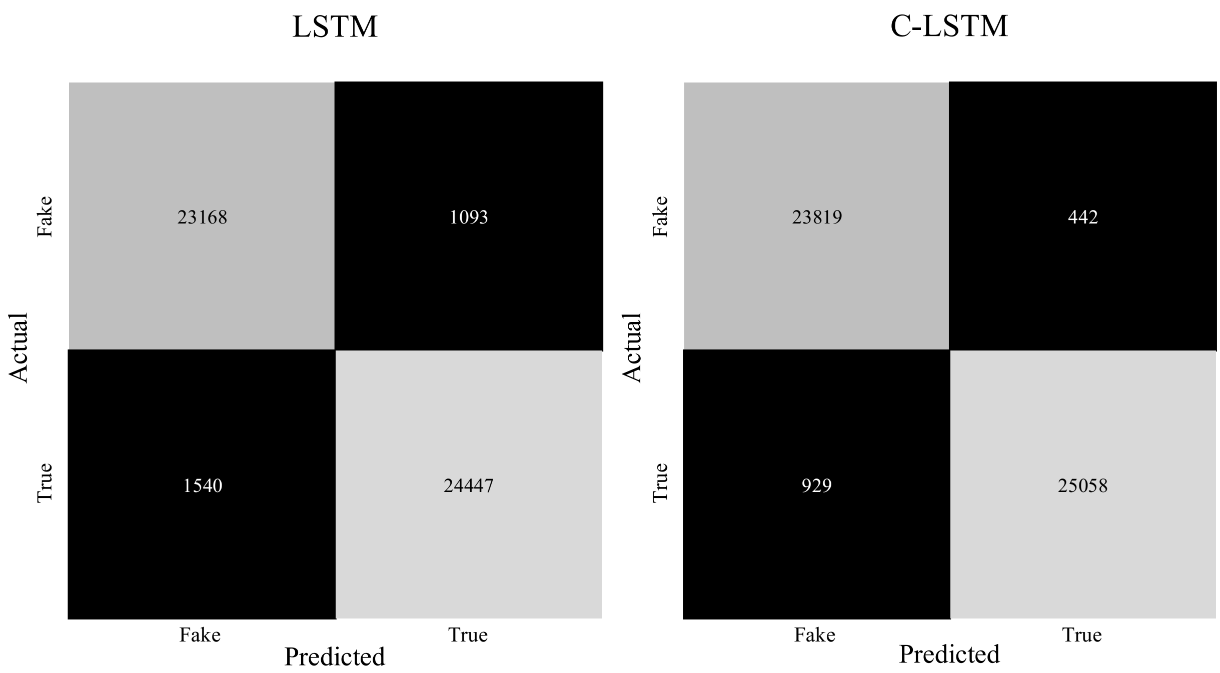

Figure 14: Comparison of confusion matrices

4.5. Model prediction

As illustrated at the beginning of this chapter, once the model has been trained and evaluated, the next step is to predict the Motley Fool news. Chapter 5 provides an insight into the unique attributes of the predicted articles.

5. Results

The results of the thesis are described in detail in this chapter. First, the classified Motley Fool data set is discussed. The different characteristics of fake and legitimate news are contrasted and highlighted. In the second part of this section, the focus has been placed on the impact of FN on the share price.

5.1. Motley Fool results

From 396,642 articles in all, the algorithm classifies 338,790 as legitimate news and 57,852 as FN. This includes articles that do not contain a ticker or are not listed. The next step is the examination of 50 real and fake articles without market capitalization or ticker symbol to gather a better understanding for this type of news. After a subjective evaluation of these articles, it can be concluded that these financial articles do not contain specific information about a certain firm. In order to assess whether FN articles have a distorting effect on the share price, a ticker is necessary. As a result, all news items that do not contain a ticker or a market capitalization will be removed. After this step, we have in total 201,039 legitimate and 49,695 fake articles and represent 12 % of the total dataset. The classification of Motley Fool articles is demonstrateg in Figure 15

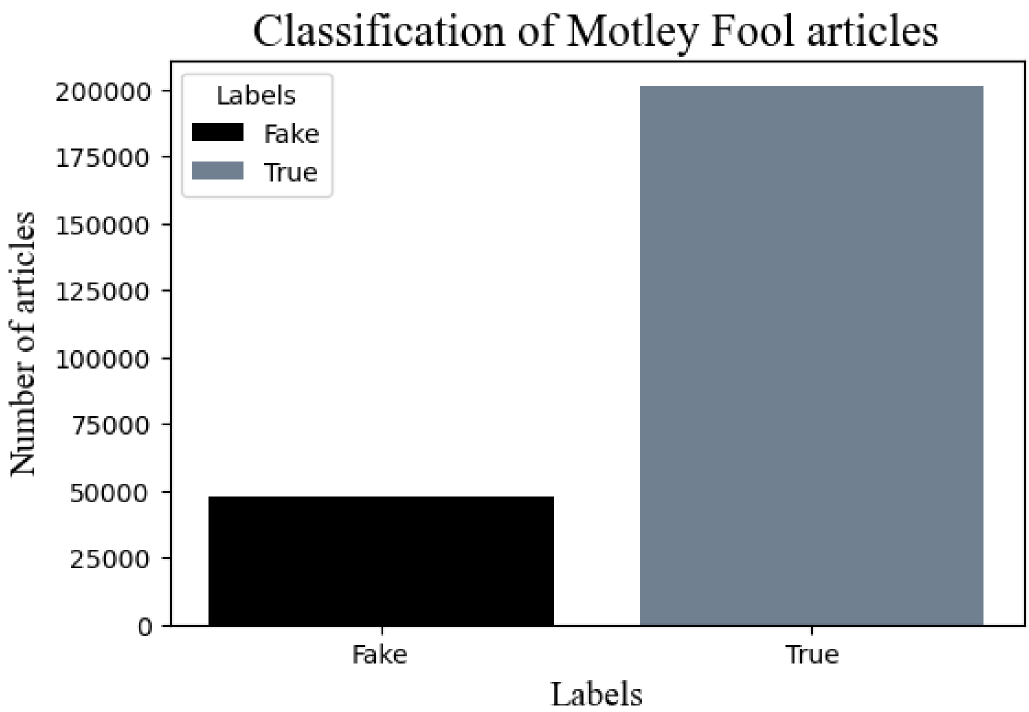

Figure 15: Classification of Motley Fool articles

5.1.1. Fake news and legitimate news

This sub-section is a brief comparison of the characteristics of fake news and legitimate news. Figure 16 shows the top 10 tickers receiving the most news in each case. This graph shows: Both fake news and legitimate news about large companies are reported. As can be seen in Table 4 of this thesis, Apple contains the largest number of written articles with a total of 11,587 in the sample, followed by DownJones, Amazon, Netflix and Meta. The top 10 companies include 34,773 reliable news and 9,588 fake news. There are 5'376 different tickers for reliable news and 2,995 for FN. The ratio and the number of tickers for FN seem high at first sight.However, as we will see in the next section, there are more than 1,000 tickers that contain only one fake article.

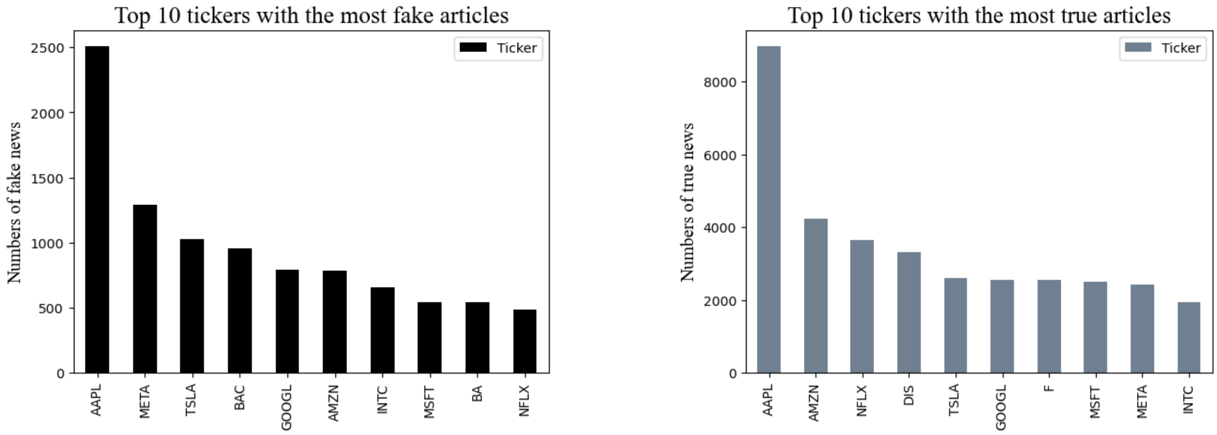

Figure 16: Comparison of Tickers

Another issue is whether there is a pattern in terms of authorship of articles. Figure 17 shows that FN articles and real news can be written by the same authors.

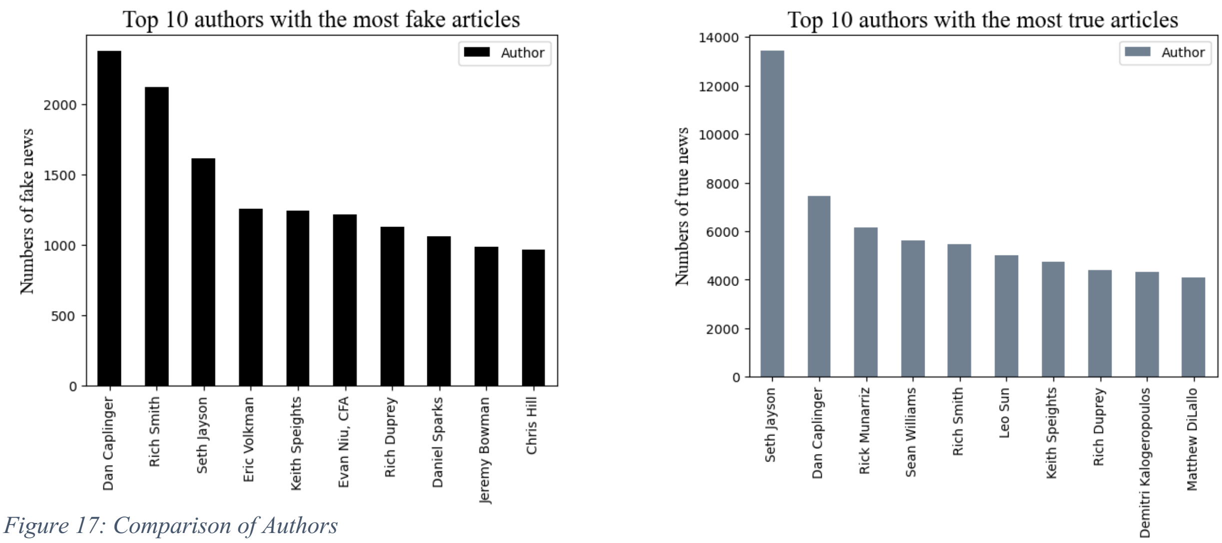

Figure 17: Comparison of Authors

Figure 18 also illustrates the distribution of FN articles over the period from 2012 to 2022, with a peak of 8,749 FN in 2013. An important indicator in this figure has been added by showing how many different companies are affected by this news in each year. Even though 2013 has the highest number of FN articles, it is worth noting that in 2012, a total of 1,081 FN report on 378 different companies. This indicates that FN covers a broader spectrum of different firms with less FN written per Firm.

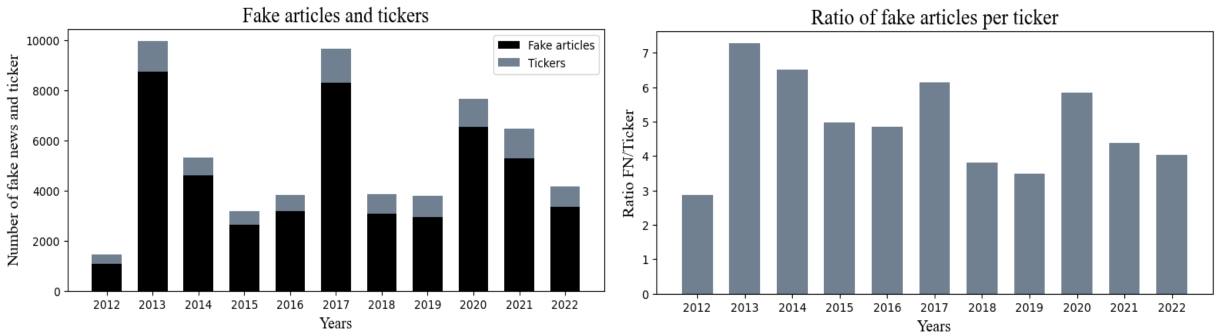

Figure 18: Fake News and Ticker between 2012-2022

Based on this initial situation and derived from Figure 16, it is necessary to examine in detail the distribution of tickers and the number of FN written per ticker. It was found that the top 10 companies with the most misinformation written were large companies. However, since the distribution is highly skewed and therefore cannot be displayed graphically, Figure 19 shows the frequency with defined classes. This is explained in more detail with an example to avoid possible ambiguity. Class 1 shows that a total of 834 different companies have one FN article. Class 2 shows that a total of 397 different companies have two fake posts. There are only 9 companies in total that have more than 500 FN articles. On the right side of the figure, it can be seen that 26% of the tickers contain only one false article. About 55% of the tickers contain less than 6 published misinformation.

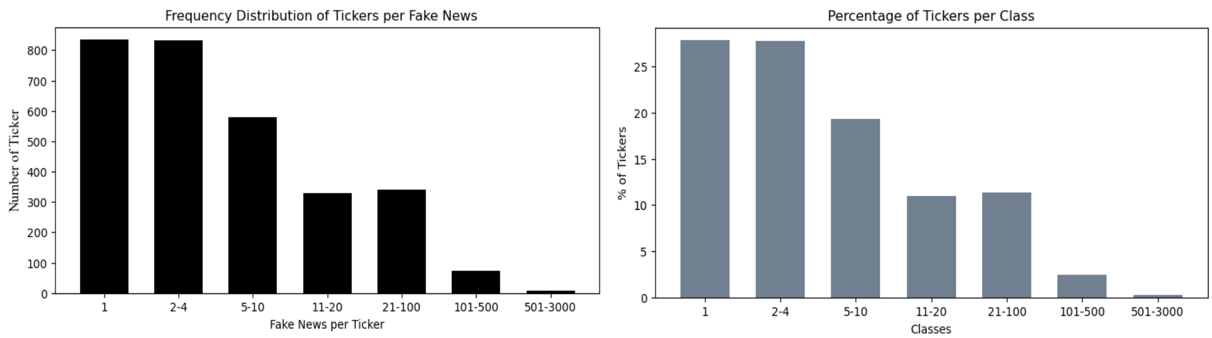

Figure 19: Distribution of Tickers

5.2. Impact of fake news on returns

The previous section demonstrated the characteristics of the FN and their representation in the data set. In this chapter, the focus is to set to analyze whether stock prices respond to FN. Therefore, an event study can be conducted to measure the impact of specific events on the price of securities (MacKinlay, 1997, p. 13). The analysis of the FN and its impact on returns provides new evidence on whether the principles of the efficient market hypothesis are still applicable or whether there is a contrary inference. The subchapters of this section will discuss the event study approach, the model and the results of this thesis.

5.2.1. Event Study

Derived from the insights in Section 2.1, this thesis will examine the influence of FN with an event study which is based on the methodology of (MacKinlay, 1997). As a preliminary step, the parameters for the three different windows must be defined. Figure 20 illustrate the timeline of an event study with the three different windows.

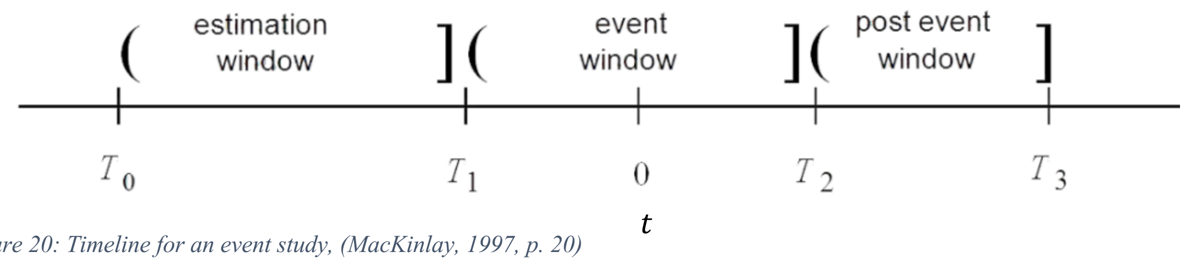

Figure 20: Timeline for an event study, (MacKinlay, 1997, p. 20)

The estimation window $[T_0, T_1]$ also called L1 is used to estimate the model parameters (average returns, betas, et.) prior to the event (MacKinlay, 1997, p. 15). With the parameters estimates, the event window $[T_1, T_2]$ calculates the abnormal returns. In the timeline t refers as the event date or in the context of this thesis the release date of the article on the website Motley Fool.

5.2.2. Fama & French Five-Factor Modell

Numerous models have been developed to measure the abnormal returns for a particular stock (MacKinlay, 1997, p. 17). For instance, research typically use the market model (CAPM), constant mean returns or the multi-factor model to calculate the expected returns. The expected returns are needed to compute the abnormal returns or the cumulative abnormal returns (CAR).

The abnormal return (AR) is defined as

$$AR_{i,t} = R_{i,t} - (\alpha_i + \beta_i R_{m,t})$$

where $AR_{i,t}$ for a given stock i on day t can be calculated as the realized return minus the expected return. Furthermore, the AR cumulated over time result in cumulative abnormal returns (CAR) and is expressed as follows:

$$CAR(t_{1,}t_2) = \sum_{t=t_1}^{t_2} AR_{i,t}$$

Derived from the insights in Section 2.1, this thesis will measure the CAR as equal-weighted residuals from the Fama & French (1992) Five-Factor Model (RMRF, SMB, HML, RMW, CMA). The expected return with the Five-Factor Model is as follows

$$R_{i,t} - R_{F,t} = \alpha_i + b_i(R_{Mt} - R_{Ft}) + s_i SMB_t + h_i HML_t + r_i RMW_t + c_i CMA_t + e_{i,t})$$

where $R_{i,t}$ is the return for a stock on day t , $R_{F,t}$ is the riskfree return, R_{Mt} is the return on the value-weight (VW) market portfolio, $SMB_t$is the return on a diversified portfolio of small stocks minus the return on a diversified portfolio of big stocks, HML_t is the difference between the returns of high and low B/M securities, RMW_t is the difference between the returns on diversified portfolios of stocks with robust and weak probability, CMA_t stands for the difference between the returns on diversified portfolios of low and high investment securities and $e_{i,t}$ is a zero-mean residual (Fama & French, 1992, pp. 4–5). In the estimation window, the parameter beta is estimated for every stock i and day t using the 251 daily returns prior to the event. The betas are then used to calculate the cumulative abnormal returns relative to the same five factors in the event window. Since the purpose of this thesis is to measure the reaction of returns to the publication of FN articles, it is necessary to examine in a subsequent step how many unique publications dates each ticker has. Furthermore, this thesis will introduce an additional event study in which the cumulative abnormal returns for a legitimate and fake article are published on the identical release date for a given security.

5.2.3. Publication dates and tickers

Based on the references in Section 5.1, the first step is to collect the financial data for the different tickers from 2012 and 2022 that contain at least one fake article. In a further step, the total quantity of 2,995 different tickers is examined to determine whether they remain available. A total of 2,794 tickers were found and downloaded on Bloomberg with the data of returns and market capitalization. No data were available for 201 tickers. Companies are divided into small caps with market capitalization of less than \$2 billion, mid-caps with market capitalization of \$2 billion to \$10 billion, and large caps with market capitalization of more than \$10 billion to better examine their respective returns as a function of market capitalization. This subdivision of firm size is identical to Niessner et al. (2018, p. 34) study, with the aim of better examining the impact of FN on profits for each of the three categories and thus obtaining more concise findings. In contrast, the study by Clarke et al. (2019, p. 10) did not make any subdivision based on market capitalization. There are a total of 49,695 FN articles and each of them contain a publication date. If a ticker contains two identical publication dates of a FN article, the duplicates are removed. For instance, if the ticker AAPL has two FN article on 2022-01-01, there will be one unique input date for the event study model. As a result, the sample of FN with a total of 49,695 entries covers 2,794 different tickers and 41,993 different publication data. The following figure presents the distribution of the respective tickers and the number of publication dates based on the size of the company. As already shown in chapter four, since large caps contain the largest number of FN articles, it is not surprising that the large-cap firms also include more publication dates. Nevertheless, the figure on the right shows that in the event study most of the companies involved are represented by small caps. To be more specific, there are in total 8,221 unique release dates of FN on the website Motley Fool for small caps with 1'327 different tickers.

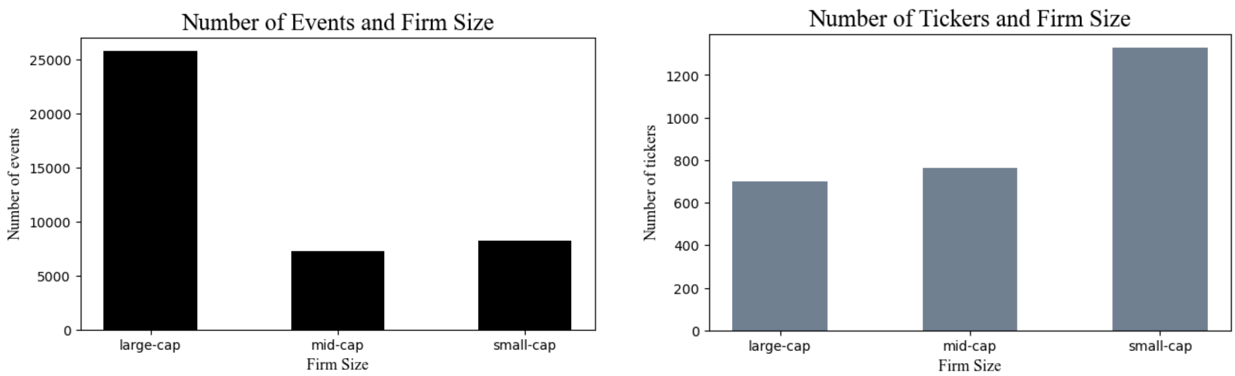

Figure 21: Number of event dates and tickers for fake news

While the article above represents only FN articles, this thesis will examine additionally the response on returns for a given company that has a publication of a trusted article and a false article on the same date on Motley Fool. This approach provides a degree of comparability of results with the findings of Clarke et al. (2019) and Niessner et al. (2018). Therefore, a proprietary data set is first created that contains only legitimate articles and FN articles with an identical release date t for a given security i. Although they were a total of 41,993 unique releases dates for FN in the first example of this section, Figure 22 on the left illustrate, that there are a cumulative 11,125 unique publication dates of articles on Motley Fool, that contain a fake and trustful news for a specific ticker i on a given day t. Comparable findings can also be observed regarding the amount of news, which are mostly written about large caps firms.

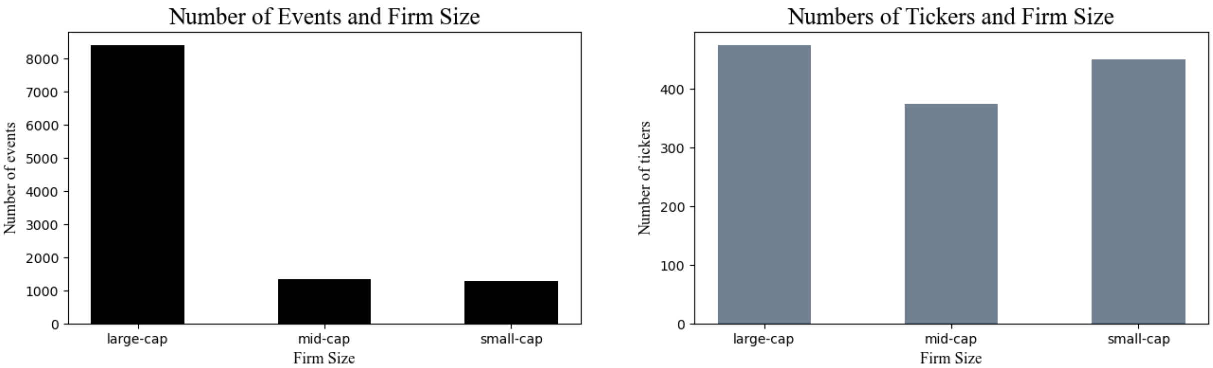

Figure 22: Number of event dates and tickers for fake and true news

The difference to the approach described above, however, is that there are fewer companies in the small-cap class. Based on the findings from MacKinlay (1997,pp. 22–26) in section 5.2.1 , the event study model is conducted with an 41-day event window and a post event window of $[-20, +20]$. For each release date of the article the 251-trading day prior to the publication date is used as the estimation window. Contrary to Niessner et al. (2018), which used an event window of [+1, +250], this thesis will set the event window shorter, with the aim to prevent bias as suggested by (Kothari & Warner, 2007, p. 8). According to Fama (1991, p. 1602) event studies represent the cleanest evidence on the efficiency of the market.

To conclude briefly, the abnormal returns are examined for an event study of the release of only fake articles as well an additional event study with the proprietary data set which includes release dates of true and FN articles. The first sample consists of a total of 41,993 releases of FN for 2,794 companies for the 10-year period September 2012 to October 2022. The releases of FN are categorized into three different classes of firm size. The second sample contain in total 11,125 publication dates of true and fake articles with the identical categorization of firm size and time span. In the next subsection, the result of the event study is presented.

5.2.4. Return reaction

The properties and features of the model have been presented in detail in the previous chapters. In this section, the results are outlined and compared with the findings from the literature review. This subsection first discusses the findings of the event study with the FN articles. Then, the findings of the second event study are subsequently explained.

Cumulative Abnormal Returns for Fake News

Returns for small-cap firms reach a cumulative abnormal return (CAR) of 3% in the first two days after the release. The downward trend is pronounced as the CAR decreases significantly after the third day and ends negative at -1% after 20 days. The abnormal return on the event day 0 is 0.008 % with a t-statistic of 2.02. Table 1 in Appendix 2 provides the abnormal and CAR returns in tabular form. This is consistent with the findings from MacKinlay (1997, pp. 25–26), which demonstrated that bad news about a company's earnings announcement leads to a negative cumulative abnormal return of -1% after 20 days. The permanent downward trend in securities prices for small caps and the publication of the FN could possibly be explained by the fact that the market recognizes the FN articles and investor perceives the signal of FN as negative. Contrary to (Niessner et al., 2018), no positive cumulative abnormal returns are observed during the 20-day period. Figure 23 illustrates the CAR.

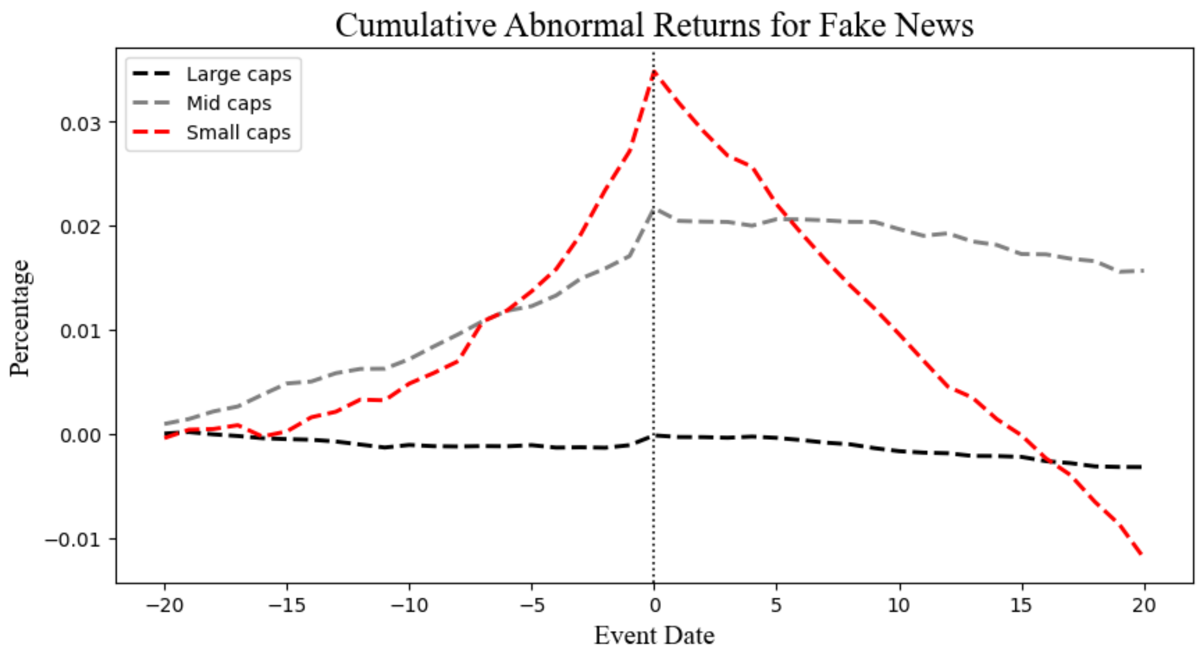

Figure 23: Cumulative Abnormal Returns for Fake News

This could also be due to the fact that more micro-cap companies are represented in the study by Niessner et al. (2018), than in this thesis. As the example of Galena Biopharm shows, the impact of the FN on the share price of micro-cap companies is significantly higher. The security price of Galena Biopharm with the ticker GALE rose from \$2 to \$7.50 during the releases of FN articles between August 2013 and January 2014, more than tripling for six months (Niessner et al., 2018, pp. 17–18). The CAR for mid-size firms is for the first 5 days about 2% and then decreases continuously until day 20 and reaches 1.5%. This result provides new evidence compared to the studies mentioned in chapter 2.1 that FN articles are potentially able to influence the share price of mid-cap companies. Even if the effect of the price reaction is not significant high, these findings still show that the market cannot corporate the information to some extent. The CAR for each event date are statistically significant at the 1% level. Not surprisingly, FN has no impact on the security prices of the large companies. AR and CAR are constant at around 0.0%. This is in consistency with the findings from Fama (1991, pp. 1602–1603) and with the existing literacy interpretation that large companies are not affected by false information and that the market for this firm size is efficient.

In conclusion, it is evident that the release of FN is associated with long-term negatives for small-cap firms. The data do not reveal in detail whether the downward trend in the market is an act of pump-and-dump scheme to manipulate the stock price or a signal of underlying poor corporate performance. In contrast, there is a marginal increase in CAR for mid-cap companies, but this decreases again slightly over the measured period. The influence of FN on the price of large companies remains absent.

Cumulative Abnormal Returns for True and Fake News

In subsequent step, the results of price responses for true and fake articles are examined. Figure 24 depicts the difference between the CAR for true and fake articles and represent the findings separately for the three different firm sizes. As illustrated in Figure 24, the CAR decreases significantly for small-cap firms and ends to 0% after 20 days. Similar to the results of the first event study, there is no evidence that large companies are influenced by news articles. There is a considerable return response for mid-size firms. The CAR reaches a high of 7.9% after 20 days with a standard error of 0.0006 and is statistically significant at the 1% level. Appendix 3 and Table 2 provides the AR and CAR in tabular form for the second event study. The results from the first event study are supported that FN articles published on the same day with an trustfull article can lead to an increase of price reaction for mid-cap firms. From this point of view, it could be argued that the investor cannot distinguish between legitimate and FN articles to a certain extent. This dispersion of news could potentially cause investors to overreact or that the market temporarily delay the price reaction.

A remarkable resemblance between these underlying trends - small-cap firms that experience immediate decline in the security price or mid-size firms that can cumulate significant abnormal returns for 20 days following publication or the non-effect of news for large companies – are similar to the first event study. The similarity of the results confirms to some extent the methodology of the event study.

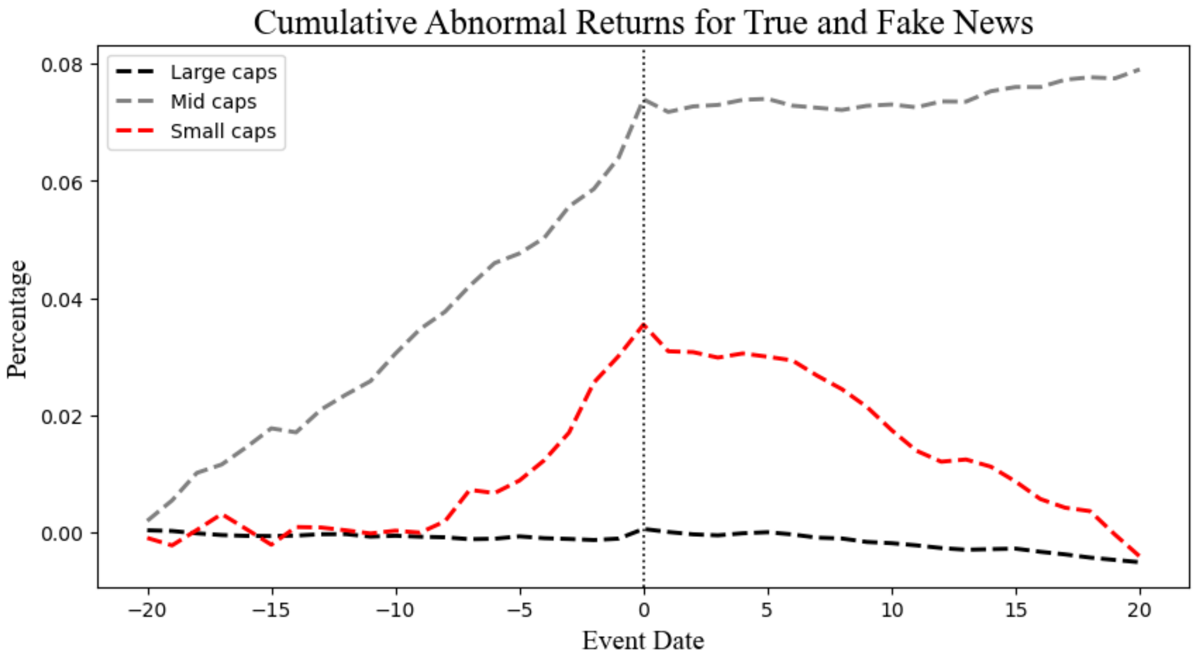

Figure 24: Cumulative abnormal returns for fake and true news

6. Conclusion and outlook

The final part of this thesis will summarize the resulsts and answers the research questions. In addition, the results are critically reviewed. Recommendations are provided for further research in this area.

6.1. Summary and discussion of results

This work used a unique and self-generated dataset to optimally train an algorithm with texts from different domains to classify financial texts from the website Motley Fool with a high degree of accuracy. The first part of this work therefore focused strongly on the implementation and development of a DL model. As a result, the algorithm using NLP and the self-generated dataset achieved state-of-the art results for binary text classification, which also answers the first research question of this thesis. The second part of the paper dealt with the event study and the price reactions of FN articles. The impact of the FN on large companies is essentially unprecedented. A negative effect is noticeable for small cap companies, which is supported by findings from existing studies (Fama, 1991, p. 1602; MacKinlay, 1997, pp. 25–26). This paper presents new evidence that FN articles published on social media financial platforms have a significant impact on medium-sized firms. This finding represent to some extent the first documented influence of FN articles on the security prices for mid-sized firms and is consistent with the theory that crowd-sourced financial platforms could impact stock prices (Chen et al., 2014b). This also answers the second question of this thesis by stating that the FN articles published on these financial platforms can have an impact on the price of a security for medium-sized companies.

6.2. Limitation

This master thesis shows that the influence of FN as well as the approach to measure the price reaction can be complex. However, the evidence resulting above must also be considered critically. There are no open data sets with FN from the field of finance. Since the trained algorithm has been trained with texts from different domains, such as sports, politics, social media or economics, it might lead to the fact that it cannot specifically classify financial texts. Financial texts can be distinguished in that they are mostly recorded objectively and corroborated with numbers. After training the algorithm, transferlearning is applied. It is not exactly clear whether the classified article as fake on Motley fool have been classified correctly by the algorithm.

6.3. Recommendation for further research

Recommendations for further research in this area can be made at the level of the dataset and event study design. Thus, a labeled dataset containing only finance-related news can be created with the goal of training an algorithm specifically adapted to these articles and their linguistic characteristics. Further efforts can be made in the area of event studies by adapting different models, estimation, event or post-event windows.

7. Biblioraphy

Ahmad, I., Yousaf, M., Yousaf, S., & Ahmad, M. O. (2020). Fake News Detection Using Machine Learning Ensemble Methods. *Complexity*, *2020*, e8885861. https://doi.org/10.1155/2020/8885861

Ahmed, H., Traore, I., & Saad, S. (2017). Detecting opinion spams and fake news using text classification. *Security and Privacy*, *1*, e9. https://doi.org/10.1002/spy2.9

Ali, A. M., Ghaleb, F. A., Al-Rimy, B. A. S., Alsolami, F. J., & Khan, A. I. (2022). Deep Ensemble Fake News Detection Model Using Sequential Deep Learning Technique. *Sensors*, *22*(18), Article 18. https://doi.org/10.3390/s22186970

Allcott, H., & Gentzkow, M. (2017). Social Media and Fake News in the 2016 Election. *Journal of Economic Perspectives*, *31*(2), 211–236.

Bengfort, B., Bilbro, R., & Ojeda, T. (2018). *Applied Text Analysis with Python Enabling Language-Aware Data Products with Machine Learning* (First Edition). O-Reilly.

Burkhardt, J. M. (2017). Chapter 1. History of Fake News. *Library Technology Reports*, *53*(8), Article 8.

Charniak, E. (2019). *Introduction to Deep Learning*. The MIT Press.

Chen, H., De, P., Hu, Y. (Jeffrey), & Hwang, B.-H. (2014a). Wisdom of Crowds: The Value of Stock Opinions Transmitted Through Social Media. *Review of Financial Studies*, *27*(5), 1367–1403.

Chen, H., De, P., Hu, Y. (Jeffrey), & Hwang, B.-H. (2014b). Wisdom of Crowds: The Value of Stock Opinions Transmitted Through Social Media. *Review of Financial Studies*, *27*(5), 1367–1403. https://doi.org/10.1093/rfs/hhu001

Chi Nhan, N., & Oliver, Z. (2022). *Machine Learning – kurz & gut • Buch oder E-Book kaufen • dpunkt.verlag* (2. Auflage). O'Reilly. https://dpunkt.de/produkt/machine-learning-kurz-gut-2/

Chollet, F. (2017). *Deep Learning with Python*. Manning Publications.

Clarke, J., Chen, H., Du, D., & Hu, Y. J. (2019). *Fake News, Investor Attention, and Market Reaction* (SSRN Scholarly Paper No. 3213024). https://doi.org/10.2139/ssrn.3213024

Cookson, J. A., & Niessner, M. (2019). *Why Don't We Agree? Evidence from a Social Network of Investors* (SSRN Scholarly Paper No. 2754086). https://doi.org/10.2139/ssrn.2754086

D'Ulizia, A., Caschera, M. C., Ferri, F., & Grifoni, P. (2021). Fake news detection: A survey of evaluation datasets. *PeerJ Computer Science*, *7*, e518. https://doi.org/10.7717/peerj-cs.518

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

Fama, E. F. (1991). Efficient Capital Markets: II. *The Journal of Finance*, *46*(5), 1575–1617. https://doi.org/10.1111/j.1540-6261.1991.tb04636.x

Fama, E. F., & French, K. R. (1992). The Cross-Section of Expected Stock Returns. *The Journal of Finance*, *47*(2), 427–465. https://doi.org/10.1111/j.1540-6261.1992.tb04398.x

Fazlija, B. (2022). *Class materials from the Machine Learning course. Fall semester 2022*. University of Applied Science Zurich, School of Management and Law.

Fong, B. (2021). Analysing the behavioural finance impact of "fake news" phenomena on financial markets: A representative agent model and empirical validation. *Financial Innovation*, *7*(1), 53. https://doi.org/10.1186/s40854-021-00271-z

Goldani, M. H., Momtazi, S., & Safabakhsh, R. (2021). Detecting fake news with capsule neural networks. *Applied Soft Computing*, *101*, 106991. https://doi.org/10.1016/j.asoc.2020.106991

Goodfellow, I., Bengio, Y., & Courville, aaron. (2017). *Deep Learning*. MIT Press.

Google. (2022). *Klassifizierung: Genauigkeit | Machine Learning*. Google Developers. https://developers.google.com/machine-learning/crash-course/classification/accuracy?hl=de

Guido, S., & Müller, A. C. (2016). *Introduction to Machine Learning with Python von Sarah Guido, Andreas C. Müller. Bücher | Orell Füssli* (First Edition). O'Reilly. https://www.orellfuessli.ch/shop/home/artikeldetails/A1033701861

Guttag, J. V. (2017). *Introduction to Computation and Programming Using Python with Application to Understanding Data* (2nd ed.). The MIT Press.

Harvard. (2022, September 24). *KaiDMML/FakeNewsNet*. https://github.com/KaiDMML/FakeNewsNet

Khan, J. Y., Khondaker, Md. T. I., Afroz, S., Uddin, G., & Iqbal, A. (2021). A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications*, *4*, 100032. https://doi.org/10.1016/j.mlwa.2021.100032

Kothari, S. P., & Warner, J. B. (2007). Econometrics of event studies. *Handbook of Corporate Finance ; Vol. 1*.

Kulkarni, A., & Shivananda, A. (2021). *Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python*. Apress. https://doi.org/10.1007/978-1-4842-4267-4

MacKinlay, A. C. (1997). Event Studies in Economics and Finance. *Journal of Economic Literature*, *35*(1), 13–39.

Mejia, N. (2022). BEATING THE MARKET: STOCK PRICE PREDICTION THROUGH ARTIFICIAL NEURAL NETWORKS, SUPPORT VECTOR MACHINES, AND SENTIMENT ANALYSIS. *Innovatus*, *5*(1), 14–14.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). *Efficient Estimation of Word Representations in Vector Space* (arXiv:1301.3781). arXiv. https://doi.org/10.48550/arXiv.1301.3781

Mishra, S., Shukla, P., & Agarwal, R. (2022). Analyzing Machine Learning Enabled Fake News Detection Techniques for Diversified Datasets. *Wireless Communications and Mobile Computing*, *2022*, e1575365. https://doi.org/10.1155/2022/1575365

NASDAQ. (2022). *Nasdaq*. https://www.nasdaq.com/publishers/the-motley-fool

Newman, M. L., Pennebaker, J. W., Berry, D. S., & Richards, J. M. (2003). Lying Words: Predicting Deception from Linguistic Styles. *Personality and Social Psychology Bulletin*, *29*(5), 665–675. https://doi.org/10.1177/0146167203029005010

Niessner, M. (2015). *Strategic Disclosure Timing and Insider Trading* (SSRN Scholarly Paper No. 2439040). https://doi.org/10.2139/ssrn.2439040

Niessner, M., Kogan, S., & Moskowitz, T. (2018, September 22). Fake News: Evidence from Financial Markets. *The Harvard Law School Forum on Corporate Governance*. https://corpgov.law.harvard.edu/2018/09/22/fake-news-evidence-from-financial-markets/

Oliveira, N., Cortez, P., & Areal, N. (2013). On the Predictability of Stock Market Behavior Using StockTwits Sentiment and Posting Volume. In L. Correia, L. P. Reis, & J. Cascalho (Eds.), *Progress in Artificial Intelligence* (pp. 355–365). Springer. https://doi.org/10.1007/978-3-642-40669-0_31

P. Murphy, K. (2012). *Machine Learning A Probabilistic Perspective*. MIT Press.

Pennebaker, J., Booth, R., Boyd, R., & Franchis, M. (2015). *Linguistic Inquiry and Word Count: LIWC2015*. https://www.researchgate.net/publication/337731895_Linguistic_Inquiry_and_Word_Count_LIWC2015

Pennebaker, J. W. (2011). *The Secret Life of Pronouns: What Our Words Say About Us*. https://de.b-ok.xyz/book/1184413/49c038

Purda, L., & Skillicorn, D. (2015). Accounting Variables, Deception, and a Bag of Words: Assessing the Tools of Fraud Detection. *Contemporary Accounting Research*, *32*(3), 1193–1223. https://doi.org/10.1111/1911-3846.12089

Rai, N., Kumar, D., Kaushik, N., Raj, C., & Ali, A. (2022). Fake News Classification using transformer based enhanced LSTM and BERT. *International Journal of Cognitive Computing in Engineering*, *3*, 98–105. https://doi.org/10.1016/j.ijcce.2022.03.003

Richardson, L. (2022). *beautifulsoup4: Screen-scraping library* (4.11.1) [Python]. https://www.crummy.com/software/BeautifulSoup/bs4/

Rubin, V. L., Chen, Y., & Conroy, N. K. (2015). Deception detection for news: Three types of fakes. *Proceedings of the Association for Information Science and Technology*, *52*(1), 1–4. https://doi.org/10.1002/pra2.2015.145052010083

Ruchansky, N., Seo, S., & Liu, Y. (2017). CSI: A Hybrid Deep Model for Fake News Detection. *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, 797–806. https://doi.org/10.1145/3132847.3132877

Samadi, M., Mousavian, M., & Momtazi, S. (2021). Deep contextualized text representation and learning for fake news detection. *Information Processing & Management*, *58*(6), 102723. https://doi.org/10.1016/j.ipm.2021.102723

Sastrawan, I. K., Bayupati, I. P. A., & Arsa, D. M. S. (2022). Detection of fake news using deep learning CNN–RNN based methods. *ICT Express*, *8*(3), 396–408. https://doi.org/10.1016/j.icte.2021.10.003

Shu, K., Mahudeswaran, D., Wang, S., Lee, D., & Liu, H. (2019). *FakeNewsNet: A Data Repository with News Content, Social Context and Spatialtemporal Information for Studying Fake News on Social Media* (arXiv:1809.01286). arXiv. https://doi.org/10.48550/arXiv.1809.01286

Souza, T. T. P., Kolchyna, O., Treleaven, P. C., & Aste, T. (2015). *Twitter Sentiment Analysis Applied to Finance: A Case Study in the Retail Industry* (arXiv:1507.00784). arXiv. https://doi.org/10.48550/arXiv.1507.00784

Stanford. (2022). *CS231n Convolutional Neural Networks for Visual Recognition*. https://cs231n.github.io/neural-networks-1/

Tetlock, P. C. (2007). Giving Content to Investor Sentiment: The Role of Media in the Stock Market. *The Journal of Finance*, *62*(3), 1139–1168. https://doi.org/10.1111/j.1540-6261.2007.01232.x

Tomaz, R. M., & Tomaz, J. M. T. (2020). The Brazilian Presidential Election of 2018 and the relationship between technology and democracy in Latin America. *Journal of Information, Communication and Ethics in Society*, *18*(4), 497–509. https://doi.org/10.1108/JICES-12-2019-0134

University of Toronto. (2022). *DEEP BELIEF NETWORKS WITH APPLICATIONS TO FAST DOCUMENT RETRIEVAL CSC2515*.

Vosoughi, S., Roy, D., & Aral, S. (2018). The spread of true and false news online. *Science*, *359*, 1146–1151. https://doi.org/10.1126/science.aap9559

Wang, W. Y. (2017). "Liar, Liar Pants on Fire": A New Benchmark Dataset for Fake News Detection. *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 422–426. https://doi.org/10.18653/v1/P17-2067

Yu, X., Mitra, G., Arbex-Valle, C., & Sayer, T. (2015). *An Impact Measure for News: Its Use in Daily Trading Strategies* (SSRN Scholarly Paper No. 2702032). https://doi.org/10.2139/ssrn.2702032

Zhang, W., & Skiena, S. (2010). Trading Strategies to Exploit Blog and News Sentiment. *Proceedings of the International AAAI Conference on Web and Social Media*, *4*(1), Article 1. https://doi.org/10.1609/icwsm.v4i1.14075

Zheludev, I., Smith, R., & Aste, T. (2014). When Can Social Media Lead Financial Markets? *Scientific Reports*, *4*(1), Article 1. https://doi.org/10.1038/srep04213

Zhou, L., & Zhang, D. (2008). Following Linguistic Footprints: Automatic Deception Detection in Online Communication. *Commun. ACM*, *51*, 119–122. https://doi.org/10.1145/1378727.1389972

8. Appendix

Appendix 1: Neural network dictionary – based on Fazlija (2022)

- **Backpropagation:** Gradient-based iterative algorithm to learn parameters of neural networks.
- **Hyperparameter:** Parameters of a model that can be changed. The following are examples:
    - learning rate
    - epoch
    - batch size
    - number of hidden layers etc.
- **Learning rate:** Hyperparameter which controls how fast weights are updated.
- **Chain rule:** A technique for computing the derivative of compositions of functions.
- **Epoch:** One training cycle through the entire training data set.
- **Batch size:** Number of samples that are propagated through the network at once.
- **Gradient:** Vector-valued function whose value at point pp is the partial derivative of function computed at point pp.
- **Gradient descent:** An iterative optimization algorithm for finding extreme values of differentiable functions.
- **Stochastic gradient descent:** Stochastic approximation of gradient descent where instead of using all data for update, a randomly selected subset of the data is used.
- **Loss:** A scalar which tells how well/poorly the predictions of the neural networks fit with respect to the ground truth.
- **Random initialization:** Randomly setting the weights of a neural network before the training process.
- **Regularization:** A technique to prevent overfitting.
- **Parameters:** Weights of the neural networks.
- **Activation function:** Non-linear function that is applied to neurons elementwise.
- **Overfitting:** Happens when neural networks perform well on training data but not on the test data.
- **Optimizer**: Algorithm that implements the form the weights are adapted (e.g., rmsprop, root mean square propagation).
- **Underfitting:** Happens when neural networks perform poorly on both the training and the test data.
- **Input layer:** Input data.
- **Output layer:** Predictions of the neural networks.
- **Hidden layers:** Layers between input and output layers, also called intermediate layers.

Appendix 2: Table 1 – CAR and AR for Fake News

Table 1 – Fake News

	Five-Factor Model Fama					
	Large caps		Mid caps		Small caps	
Event Day	AR	CAR	AR	CAR	AR	CAR
-20	0.00001	0.00001	0.00094	0.00094 **	-0.00042	-0.00042
-19	0.00015	0.00016	0.00047	0.00141 **	0.00080	0.00038
-18	-0.00022	-0.00006	0.00072	0.00214 ***	0.00008	0.00045
-17	-0.00016	-0.00022	0.00048	0.00262 ***	0.00037	0.00082
-16	-0.00021	-0.00043	0.00113	0.00375 ***	-0.00109	-0.00027
-15	-0.00010	-0.00052 *	0.00108	0.00483 ***	0.00049	0.00022
-14	-0.00006	-0.00058 *	0.00019	0.00502 ***	0.00136	0.00158
-13	-0.00017	-0.00075 **	0.00080	0.00582 ***	0.00052	0.00211
-12	-0.00029	-0.00104 ***	0.00042	0.00623 ***	0.00117	0.00328
-11	-0.00029	-0.00133 ***	0.00002	0.00625 ***	-0.00007	0.00321
-10	0.00024	-0.00108 **	0.00092	0.00717 ***	0.00161	0.00482
-9	-0.00012	-0.0012 ***	0.00122	0.00839 ***	0.00102	0.00584
-8	-0.00003	-0.00123 ***	0.00117	0.00956 ***	0.00114	0.00698
-7	0.00004	-0.00119 **	0.00126	0.01082 ***	0.00381	0.01079
-6	-0.00002	-0.00121 **	0.00101	0.01183 ***	0.00107	0.01186
-5	0.00011	-0.0011 **	0.00044	0.01227 ***	0.00184	0.01370
-4	-0.00024	-0.00134 **	0.00103	0.0133 ***	0.00213	0.01583
-3	0.00003	-0.00131 **	0.00161	0.01491 ***	0.00332	0.01916
-2	-0.00005	-0.00135 **	0.00098	0.01589 ***	0.00427	0.02343
-1	0.00026	-0.0011 **	0.00120	0.01709 ***	0.00376	0.02718
0	0.00094	-0.00016	0.00460	0.02169 ***	0.00766	0.03484 **
1	-0.00016	-0.00032	-0.00122	0.02047 ***	-0.00300	0.03184 *
2	-0.00001	-0.00033	-0.00007	0.0204 ***	-0.00271	0.02913
3	-0.00006	-0.00039	-0.00003	0.02037 ***	-0.00238	0.02675
4	0.00011	-0.00028	-0.00037	0.02001 ***	-0.00107	0.02568
5	-0.00012	-0.00040	0.00063	0.02063 ***	-0.00361	0.02207
6	-0.00021	-0.00061	-0.00002	0.02061 ***	-0.00276	0.01931
7	-0.00025	-0.00086	-0.00009	0.02052 ***	-0.00261	0.01670
8	-0.00016	-0.00102	-0.00014	0.02038 ***	-0.00244	0.01427
9	-0.00037	-0.00139 **	-0.00001	0.02037 ***	-0.00223	0.01204
10	-0.00028	-0.00168 **	-0.00069	0.01968 ***	-0.00245	0.00959
11	-0.00014	-0.00182 **	-0.00064	0.01904 ***	-0.00256	0.00703
12	-0.00007	-0.00188 **	0.00023	0.01927 ***	-0.00251	0.00452
13	-0.00026	-0.00214 ***	-0.00078	0.01849 ***	-0.00106	0.00347
14	-0.00001	-0.00215 ***	-0.00035	0.01815 ***	-0.00207	0.00139
15	-0.00008	-0.00223 ***	-0.00087	0.01728 ***	-0.00155	-0.00015
16	-0.00041	-0.00264 ***	-0.00003	0.01725 ***	-0.00222	-0.00237
17	-0.00019	-0.00283 ***	-0.00042	0.01683 ***	-0.00160	-0.00397
18	-0.00031	-0.00314 ***	-0.00023	0.0166 ***	-0.00257	-0.00655
19	-0.00006	-0.0032 ***	-0.00102	0.01558 ***	-0.00218	-0.00873
20	0.00001	-0.00321 ***	0.00010	0.01568 ***	-0.00323	-0.01196

Appendix 3: Table 2 – CAR and AR for Fake and Real News

Table 2 – Fake and Real News

	Five-Factor Model Fama					
	Large cap		Mid cap		Small cap	
Event Day	AR	CAR	AR	CAR	AR	CAR
-20	0.00036	0.00036 *	0.00197	0.00197 *	-0.00094	-0.00094
-19	-0.00012	0.00024	0.00349	0.00546 ***	-0.00128	-0.00222
-18	-0.0004	-0.00016	0.00471	0.01016 ***	0.00263	0.00041
-17	-0.00029	-0.00045	0.00143	0.01159 ***	0.00273	0.00314
-16	-0.00015	-0.00059	0.00297	0.01456 ***	-0.00259	0.00055
-15	-0.00002	-0.00062	0.00321	0.01777 ***	-0.00265	-0.0021
-14	0.00008	-0.00054	-0.00069	0.01708 ***	0.00302	0.00092
-13	0.00023	-0.00031	0.00388	0.02096 ***	-0.00008	0.00084
-12	0.00001	-0.00029	0.00254	0.0235 ***	-0.00048	0.00036
-11	-0.00043	-0.00072	0.00233	0.02583 ***	-0.00054	-0.00018
-10	0.00016	-0.00057	0.00467	0.03051 ***	0.00047	0.00029
-9	-0.00017	-0.00074	0.00422	0.03472 ***	-0.00033	-0.00004
-8	-0.00010	-0.00084	0.00295	0.03767 ***	0.00194	0.0019
-7	-0.00032	-0.00116	0.00443	0.0421 ***	0.00537	0.00727
-6	0.00009	-0.00108	0.00387	0.04598 ***	-0.00056	0.00671
-5	0.00039	-0.00068	0.00164	0.04762 ***	0.00214	0.00885
-4	-0.00031	-0.00099	0.00265	0.05027 ***	0.00346	0.01231 **
-3	-0.00013	-0.00112	0.00539	0.05565 ***	0.00474	0.01704 ***
-2	-0.00018	-0.0013	0.00298	0.05863 ***	0.00854	0.02558 ***
-1	0.00025	-0.00106	0.00535	0.06398 ***	0.00453	0.03012 ***
0	0.00165	0.00059	0.00995	0.07393 ***	0.00536	0.03547 ***
1	-0.00052	0.00007	-0.00214	0.07178 ***	-0.00455	0.03092 ***
2	-0.00040	-0.00033	0.00094	0.07273 ***	-0.00014	0.03078 ***
3	-0.00017	-0.0005	0.00025	0.07298 ***	-0.00094	0.02984 ***
4	0.00035	-0.00015	0.00088	0.07385 ***	0.00072	0.03055 ***
5	0.00018	0.00003	0.00017	0.07402 ***	-0.00057	0.02999 ***
6	-0.00037	-0.00034	-0.00116	0.07286 ***	-0.00062	0.02937 ***
7	-0.00056	-0.0009	-0.00036	0.0725 ***	-0.00260	0.02676 ***
8	-0.00014	-0.00104	-0.00039	0.07212 ***	-0.00232	0.02445 ***
9	-0.00059	-0.00163	0.00071	0.07283 ***	-0.00301	0.02144 ***
10	-0.00021	-0.00184	0.00023	0.07306 ***	-0.00397	0.01747 **
11	-0.00037	-0.00221 *	-0.00049	0.07257 ***	-0.00351	0.01395 *
12	-0.00045	-0.00267 **	0.00100	0.07358 ***	-0.00189	0.01207
13	-0.00029	-0.00296 **	-0.00004	0.07354 ***	0.00038	0.01245
14	0.00009	-0.00286 **	0.00178	0.07532 ***	-0.00122	0.01122
15	0.00009	-0.00277 **	0.00072	0.07604 ***	-0.00254	0.00868
16	-0.00054	-0.00331 **	-0.00002	0.07603 ***	-0.00298	0.00569
17	-0.00045	-0.00376 ***	0.00127	0.07729 ***	-0.00148	0.00421
18	-0.00053	-0.0043 ***	0.00041	0.0777 ***	-0.00058	0.00363
19	-0.00040	-0.0047 ***	-0.00019	0.07751 ***	-0.00399	-0.00036
20	-0.00040	-0.00509 ***	0.00153	0.07904 ***	-0.00374	-0.00409